



A guia AWS CDK de camadas

AWS Orientação prescritiva



AWS Orientação prescritiva: A guia AWS CDK de camadas

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

Introdução	1
Construções da camada 1	3
O AWS CDK— CloudFormation ciclo de vida das construções L1	3
A especificação AWS CloudFormation do recurso	4
Construções de camada 2	6
Propriedades padrão	8
Estruturas, tipos e interfaces	9
Métodos estáticos	9
Métodos auxiliares	10
Enumerações	12
Classes auxiliares	12
Construções de camada 3	14
Interações de recursos	15
Extensões de recursos	17
Recursos personalizados	18
Práticas recomendadas	27
Perguntas frequentes	29
Não posso usar as camadas AWS CDK sem entender?	29
Posso fazer construções L2 a partir de L1 da mesma forma que faço construções L3 a partir de L2?	29
Quais AWS recursos ainda não têm construções L2 oficiais?	29
Posso criar uma construção L2 ou L3 em qualquer idioma compatível? AWS CDK	29
Onde posso encontrar construções L3 existentes fora do? AWS CDK	30
Recursos	31
Histórico do documento	32
Glossário	33
#	33
A	34
B	37
C	39
D	42
E	46
F	48
G	50

H	51
eu	53
L	55
M	56
O	61
P	63
Q	66
R	67
S	70
T	74
U	75
V	76
W	76
Z	77
.....	lxxix

A guia AWS CDK de camadas

Steven Guggenheimer, da Amazon Web Services (AWS)

Dezembro de 2023 ([histórico do documento](#))

Um dos principais conceitos por trás do AWS Cloud Development Kit (AWS CDK) é muito parecido com o conceito por trás de se manter aquecido em um dia frio. Esse conceito é chamado de camadas. Em um dia frio, você veste uma camisa, uma jaqueta e, às vezes, uma jaqueta ainda maior, dependendo do frio. Então, se você entrar e o aquecedor estiver em chamas, você pode tirar uma ou as duas camadas da jaqueta para não ficar muito quente. O AWS CDK usa camadas para fornecer diferentes níveis de abstração para o uso de componentes de nuvem. A disposição em camadas garante que você nunca precise escrever muito código ou ter pouco acesso às propriedades dos recursos ao implantar sua infraestrutura como pilhas de código (IAC).

Se você não usa o AWS CDK, precisa escrever seus [AWS CloudFormation](#) modelos manualmente; ou seja, você está aproveitando apenas uma única camada que o força a escrever muito mais código do que normalmente é necessário. Por outro lado, se AWS CDK eles abstraíssem tudo o CloudFormation que você normalmente não precisa escrever, você não seria capaz de lidar com nenhum caso extremo.

Para resolver esse problema, o AWS CDK divide o provisionamento de recursos em três camadas separadas e distintas:

- Camada 1 — A CloudFormation camada: A camada mais básica em que o CloudFormation recurso e o AWS CDK recurso são quase idênticos.
- Camada 2 — A camada com curadoria: a camada em que CloudFormation os recursos são abstraídos em classes programáticas que simplificam grande parte da sintaxe padronizada CloudFormation . Essa camada compõe a maior parte do AWS CDK.
- Camada 3 — A camada padrão: a camada mais abstrata na qual você pode usar os blocos de construção fornecidos pelas camadas 1 e 2 para personalizar o código para seu caso de uso específico.

Cada item de cada camada é uma instância de uma AWS CDK classe especial chamada `Construct`. De acordo com a [AWS documentação](#), as construções são “os blocos básicos de construção dos AWS CDK aplicativos. Uma construção representa um 'componente de nuvem' e encapsula tudo o que é AWS CloudFormation necessário para criar o componente.” As construções

dentro dessas camadas são conhecidas como construções L1, L2 e L3, dependendo da camada à qual elas pertencem. Neste guia, faremos um tour por cada AWS CDK camada para descobrir para que elas são usadas e por que são importantes.

Este guia é destinado a gerentes técnicos, leads e desenvolvedores interessados em se aprofundar nos principais conceitos que compõem o AWS CDK trabalho. AWS CDK É uma ferramenta popular, mas é muito comum que as equipes percam uma grande parte do que ela tem a oferecer. Ao começar a entender os conceitos descritos neste guia, você pode desbloquear um novo mundo de possibilidades e otimizar os processos de provisionamento de recursos de suas equipes.

Neste guia:

- [Construções da camada 1](#)
- [Construções de camada 2](#)
- [Construções de camada 3](#)
- [Práticas recomendadas](#)
- [PERGUNTAS FREQUENTES](#)
- [Recursos](#)

Construções da camada 1

As [construções L1](#) são os blocos de construção do AWS CDK e são facilmente distinguidas de outras construções pelo prefixo. Cfn Por exemplo, o pacote Amazon DynamoDB no AWS CDK contém `Table` uma construção, que é uma construção L2. A construção L1 correspondente é chamada `CfnTable` e representa diretamente um CloudFormation `Table` DynamoDB. É impossível usar o AWS CDK sem acessar essa primeira camada, embora um AWS CDK aplicativo normalmente nunca use uma construção L1 diretamente. No entanto, na maioria dos casos, as construções L2 e L3 que os desenvolvedores estão acostumados a usar dependem muito das construções L1. Portanto, você pode pensar nas construções L1 como a ponte entre CloudFormation e o AWS CDK.

O único objetivo do AWS CDK é gerar CloudFormation modelos usando linguagens de codificação padrão. Depois de executar o comando `cdk synth CLI` e os CloudFormation modelos resultantes serem gerados, o AWS CDK trabalho será concluído. O comando `cdk deploy` existe apenas por conveniência, mas o que você está fazendo ao executar esse comando acontece inteiramente nele CloudFormation. A peça do quebra-cabeça que traduz o AWS CDK código para o formato que CloudFormation compreende é a construção L1.

O AWS CDK— CloudFormation ciclo de vida das construções L1

O processo para criar e usar construções L1 consiste nestas etapas:

1. O processo de AWS CDK construção converte CloudFormation especificações em código programático na forma de construções L1.
2. Os desenvolvedores escrevem código que faz referência direta ou indireta às construções L1 como parte de um aplicativo. AWS CDK
3. Os desenvolvedores executam o comando `cdk synth` para converter o código programático novamente no formato ditado pelas CloudFormation especificações (modelos).
4. Os desenvolvedores executam o comando `cdk deploy` para implantar as CloudFormation pilhas desses modelos nos ambientes AWS da conta.

Vamos fazer um pequeno exercício. Acesse o [repositório de código AWS CDK aberto](#) em GitHub, escolha um AWS serviço aleatório e, em seguida, acesse o AWS CDK pacote desse serviço (localizado na `pastapackages`, `aws-cdk-libaws-<servicename>,lib`). Neste exemplo, vamos escolher o Amazon S3, mas isso funciona para qualquer serviço. Se você olhar o [arquivo `index.ts`](#) principal desse pacote, verá uma linha que diz:

```
export * from './s3.generated';
```

No entanto, você não verá o `s3.generated` arquivo em nenhum lugar do diretório correspondente. Isso ocorre porque as construções L1 são geradas automaticamente a partir da [especificação do CloudFormation recurso](#) durante o AWS CDK processo de construção. Portanto, você verá `s3.generated` no pacote somente depois de executar o comando AWS CDK build do pacote.

A especificação AWS CloudFormation do recurso

A especificação do AWS CloudFormation recurso define a infraestrutura como código (IAC) para AWS e determina como o código nos CloudFormation modelos é convertido em recursos em uma AWS conta. Essa especificação define AWS recursos no [formato JSON](#) em um nível por região. Cada recurso recebe um [nome de tipo de recurso](#) exclusivo que segue o `formatoprovider::service::resource`. Por exemplo, o nome do tipo de recurso para um bucket do Amazon S3 seria `AWS::S3::Bucket`, e o nome do tipo de recurso para um ponto de acesso do Amazon S3 seria `AWS::S3::AccessPoint`. Esses tipos de recursos podem ser renderizados em um CloudFormation modelo usando a sintaxe definida na especificação do AWS CloudFormation recurso. Quando o processo de AWS CDK criação é executado, cada tipo de recurso também se torna uma construção L1.

Conseqüentemente, cada construção L1 é uma imagem espelhada programática de seu recurso correspondente CloudFormation. Cada propriedade que você aplicaria em um CloudFormation modelo está disponível quando você instancia uma construção L1, e cada CloudFormation propriedade necessária também é exigida como argumento quando você instancia a construção L1 correspondente. A tabela a seguir compara um bucket S3 conforme representado em um CloudFormation modelo com o mesmo bucket S3, conforme definido como uma AWS CDK construção L1.

CloudFormation modelo

```
"amzn-s3-demo-bucket": {
  "Type": "AWS::S3::Bucket",
  "Properties": {
    "BucketName": "amzn-s3-demo-
bucket",
    "BucketEncryption": {
```

Construção L1

```
new CfnBucket(this, "amzn-s3-de
mobucket", {
  bucketName: "amzn-s3-demo-bucket",
  bucketEncryption: {
    serverSideEncryptionConfigu
ration: [
  {
```

```

    "ServerSideEncryptionConfig
uration": [
      {
        "ServerSideEncrypt
ionByDefault": {
          "SSEAlgorithm": "AES256"
        }
      }
    ],
    "MetricsConfigurations": [
      {
        "Id": "myConfig"
      }
    ],
    "OwnershipControls": {
      "Rules": [
        {
          "ObjectOwnership":
"BucketOwnerPreferred"
        }
      ]
    },
    "PublicAccessBlockConfigura
tion": {
      "BlockPublicAcls": true,
      "BlockPublicPolicy": true,
      "IgnorePublicAcls": true,
      "RestrictPublicBuckets": true
    },
    "VersioningConfiguration": {
      "Status": "Enabled"
    }
  }
}

```

```

serverSideEncryptionByDefau
lt: {
  sseAlgorithm: "AES256"
}
],
metricsConfigurations: [
  {
    id: "myConfig"
  }
],
ownershipControls: {
  rules: [
    {
      objectOwnership: "BucketOw
nerPreferred"
    }
  ]
},
publicAccessBlockConfiguration: {
  blockPublicAcls: true,
  blockPublicPolicy: true,
  ignorePublicAcls: true,
  restrictPublicBuckets: true
},
versioningConfiguration: {
  status: "Enabled"
}
});

```

Como você pode ver, a construção L1 é a manifestação exata no código do CloudFormation recurso. Não há atalhos ou simplificações, então a quantidade de texto padronizado que deve ser escrita é praticamente a mesma. No entanto, uma das grandes vantagens de usar o AWS CDK é que ele ajuda a eliminar grande parte dessa sintaxe padronizada CloudFormation . Então, como isso acontece? É aí que entra a construção L2.

Construções de camada 2

O [repositório de código AWS CDK aberto](#) é escrito principalmente usando a linguagem de [TypeScript](#) programação e consiste em vários pacotes e módulos. A biblioteca de pacotes principal, chamada `aws-cdk-lib`, é aproximadamente dividida em um pacote por AWS serviço, embora isso nem sempre seja o caso. Conforme discutido anteriormente, as construções L1 são geradas automaticamente durante o processo de construção. Então, qual é todo esse código que você vê quando olha dentro do repositório? Essas são [construções L2](#), que são abstrações das construções L1.

Os pacotes também contêm uma coleção de TypeScript tipos, enums e interfaces, bem como classes auxiliares que adicionam mais funcionalidades, mas todos esses itens servem a construções L2. Todas as construções L2 chamam suas construções L1 correspondentes em seus construtores após a instanciação, e a construção L1 resultante que é criada pode ser acessada da camada 2 da seguinte forma:

```
const role = new Bucket(this, "amzn-s3-demo-bucket", {/*...BucketProps*/});
const cfnBucket = role.node.defaultChild;
```

A construção L2 pega as propriedades padrão, os métodos de conveniência e outros açúcares sintáticos e os aplica à construção L1. Isso elimina grande parte da repetição e da verbosidade necessárias para provisionar recursos diretamente. CloudFormation

Todas as construções L2 constroem suas construções L1 correspondentes sob o capô. No entanto, as construções L2 na verdade não estendem as construções L1. [As construções L1 e L2 herdam uma classe especial chamada Construct](#). Na versão 1 AWS CDK da Construct classe foi incorporada ao kit de desenvolvimento, mas na versão 2 é um [pacote independente](#) separado. Isso é para que outros pacotes, como o [Cloud Development Kit for Terraform \(CDKTF\)](#), possam incluí-lo como uma dependência. Qualquer classe que herda a Construct classe é uma construção L1, L2 ou L3. As construções L2 estendem essa classe diretamente, enquanto as construções L1 estendem uma classe chamada `CfnResource`, conforme mostrado na tabela a seguir.

Árvore de herança L1

Construção L1

→ aula [CfnResource](#)

Árvore de herança L2

Construção L2

→ classe [Construct](#)

→ classe abstrata [CfnRefElement](#)

→→→ classe abstrata [CfnElement](#)

→→→→ [construção da classe](#)

Se as construções L1 e L2 herdam a `Construct` classe, por que as construções L2 simplesmente não estendem L1? Bem, as classes entre a `Construct` classe e a camada 1 bloqueiam a construção L1 como uma imagem espelhada do CloudFormation recurso. Eles contêm métodos abstratos (métodos que as classes posteriores devem incluir) `_toCloudFormation`, como, que forçam a construção a gerar CloudFormation sintaxe diretamente. As construções L2 ignoram essas classes e estendem a `Construct` classe diretamente. Isso lhes dá a flexibilidade de abstrair grande parte do código necessário para construções L1, construindo-as separadamente em seus construtores.

A seção anterior apresentou uma side-by-side comparação entre um bucket S3 de um CloudFormation modelo e esse mesmo bucket S3 renderizado como uma construção L1. Essa comparação mostrou que as propriedades e a sintaxe são quase idênticas, e a construção L1 economiza apenas três ou quatro linhas em comparação com a CloudFormation construção. Agora vamos comparar a construção L1 com a construção L2 para o mesmo bucket S3:

Construção L1 para bucket S3

```
new CfnBucket(this, "amzn3demobucket", {
  bucketName: "amzn-s3-demo-bucket",
  bucketEncryption: {
    serverSideEncryptionConfiguration: [
      {
        serverSideEncryptionByDefault: {
          sseAlgorithm: "AES256"
        }
      }
    ]
  },
  metricsConfigurations: [
    {
```

Construção L2 para bucket S3

```
new Bucket(this, "amzn3demobucket",
{
  bucketName: "amzn-s3-demo-bucket",
  encryption: BucketEncryption.S3_MANAGED,
  metrics: [
    {
      id: "myConfig"
    }
  ],
  objectOwnership: ObjectOwnership.BUCKET_OWNER_PREFERRED,
  blockPublicAccess: BlockPublicAccess.BLOCK_ALL,
  versioned: true
});
```

```
        id: "myConfig"
    }
],
ownershipControls: {
    rules: [
        {
            objectOwnership: "BucketOwnerPreferred"
        }
    ]
},
publicAccessBlockConfiguration: {
    blockPublicAcls: true,
    blockPublicPolicy: true,
    ignorePublicAcls: true,
    restrictPublicBuckets: true
},
versioningConfiguration: {
    status: "Enabled"
}
});
```

Como você pode ver, a construção L2 tem menos da metade do tamanho da construção L1. As construções L2 usam várias técnicas para realizar essa consolidação. Algumas dessas técnicas se aplicam a uma única construção L2, mas outras podem ser reutilizadas em várias construções para que sejam separadas em sua própria classe para reutilização. As construções L2 consolidam a CloudFormation sintaxe de várias maneiras, conforme discutido nas seções a seguir.

Propriedades padrão

A maneira mais simples de consolidar o código para provisionar um recurso é transformar as configurações de propriedade mais comuns em padrões. O AWS CDK tem acesso a linguagens de programação poderosas e CloudFormation não tem, então esses padrões geralmente são de natureza condicional. Às vezes, várias linhas de CloudFormation configuração podem ser eliminadas do AWS CDK código porque essas configurações podem ser inferidas dos valores de outras propriedades que são passadas para a construção.

Estruturas, tipos e interfaces

Embora o AWS CDK esteja disponível em várias linguagens de programação, ele é escrito nativamente TypeScript, de modo que o sistema de tipos da linguagem é usado para definir os tipos que compõem as construções L2. Mergulhar profundamente nesse sistema de tipos está além do escopo deste guia; consulte a [TypeScript documentação](#) para obter detalhes. Para resumir, a TypeScript `type` descreve o tipo de dados que uma determinada variável contém. Podem ser dados básicos, como `umstring`, ou dados mais complexos, como `umobject`. A TypeScript `interface` é outra forma de expressar o tipo de TypeScript objeto e a `struct` é outro nome para uma interface.

TypeScript não usa o termo estrutura, mas se você olhar na [Referência da AWS CDK API](#), verá que uma estrutura é, na verdade, apenas outra TypeScript interface dentro do código. A Referência da API também se refere a determinadas interfaces como interfaces. Se estruturas e interfaces são a mesma coisa, por que a AWS CDK documentação faz uma distinção entre elas?

O que AWS CDK eles chamam de estruturas são interfaces que representam qualquer objeto usado por uma construção L2. Isso inclui os tipos de objeto dos argumentos de propriedade que são passados para a construção L2 durante a instanciação, como `BucketProps` para a construção do S3 Bucket e para `TableProps` a construção da tabela do DynamoDB, além de outras interfaces usadas no. TypeScript AWS CDK Resumindo, se for uma TypeScript interface dentro do AWS CDK e seu nome não for prefixado pela letra `I`, eles a AWS CDK chamarão de estrutura.

Por outro lado, ele AWS CDK usa o termo interface para representar os elementos básicos que um objeto simples precisaria ser considerado uma representação adequada de uma determinada construção ou classe auxiliar. Ou seja, uma interface descreve quais devem ser as propriedades públicas de uma construção L2. Todos os nomes de AWS CDK interface são nomes de construções ou classes auxiliares existentes prefixadas pela letra `I`. Todas as construções L2 estendem a `Construct` classe, mas também implementam a interface correspondente. Portanto, a construção L2 `Bucket` implementa a `IBucket` interface.

Métodos estáticos

Cada instância de uma construção L2 também é uma instância de sua interface correspondente, mas o inverso não é verdadeiro. Isso é importante ao examinar uma estrutura para ver quais tipos de dados são necessários. Se uma estrutura tiver uma propriedade chamada `bucket`, que requer o tipo de dados `IBucket`, você pode passar um objeto que contenha as propriedades listadas na `IBucket` interface ou uma instância de uma `Bucket` L2. Qualquer um funcionaria. No entanto,

se essa bucket propriedade exigisse um `L2Bucket`, você poderia passar somente uma `Bucket` instância nesse campo.

Essa distinção se torna muito importante quando você importa recursos pré-existentes para sua pilha. Você pode criar uma construção L2 para qualquer recurso nativo da sua pilha, mas se precisar referenciar um recurso que foi criado fora da pilha, precisará usar a interface dessa construção L2. Isso porque criar uma construção L2 cria um novo recurso, caso ainda não exista um dentro dessa pilha. As referências aos recursos existentes devem ser objetos simples que estejam em conformidade com a interface dessa construção L2.

Para facilitar isso na prática, a maioria das construções L2 tem um conjunto de métodos estáticos associados a elas que retornam a interface dessa construção L2. Esses métodos estáticos geralmente começam com a palavra `from`. Os dois primeiros argumentos passados para esses métodos são os mesmos `scope` e `id` os argumentos necessários para uma construção L2 padrão. No entanto, o terceiro argumento não é `props` senão um pequeno subconjunto de propriedades (ou às vezes apenas uma propriedade) que define uma interface. Por esse motivo, quando você passa uma construção L2, na maioria dos casos, somente os elementos da interface são necessários. Isso é para que você também possa usar recursos importados, sempre que possível.

```
// Example of referencing an external S3 bucket
const preExistingBucket = Bucket.fromBucketName(this, "external-bucket", "name-of-
bucket-that-already-exists");
```

No entanto, você não deve depender muito de interfaces. Você deve importar recursos e usar interfaces diretamente somente quando for absolutamente necessário, porque as interfaces não fornecem muitas das propriedades, como métodos auxiliares, que tornam uma construção L2 tão poderosa.

Métodos auxiliares

Uma construção L2 é uma classe programática em vez de um objeto simples, portanto, ela pode expor métodos de classe que permitem manipular a configuração do recurso após a instanciação. Um bom exemplo disso é a construção AWS Identity and Access Management (IAM) L2 [Role](#). Os trechos a seguir mostram duas maneiras de criar a mesma função do IAM usando a construção `Role` L2.

Sem um método auxiliar:

```
const role = new Role(this, "my-iam-role", {
```

```

    assumedBy: new FederatedPrincipal('my-identity-provider.com'),
    managedPolicies: [
      ManagedPolicy.fromAwsManagedPolicyName("ReadOnlyAccess")
    ],
    inlinePolicies: {
      lambdaPolicy: new PolicyDocument({
        statements: [
          new PolicyStatement({
            effect: Effect.ALLOW,
            actions: [ 'lambda:UpdateFunctionCode' ],
            resources: [ 'arn:aws:lambda:us-east-1:123456789012:function:my-
function' ]
          })
        ]
      })
    }
  });

```

Com um método auxiliar:

```

const role = new Role(this, "my-iam-role", {
  assumedBy: new FederatedPrincipal('my-identity-provider.com')
});

role.addManagedPolicy(ManagedPolicy.fromAwsManagedPolicyName("ReadOnlyAccess"));
role.attachInlinePolicy(new Policy(this, "lambda-policy", {
  policyName: "lambdaPolicy",
  statements: [
    new PolicyStatement({
      effect: Effect.ALLOW,
      actions: [ 'lambda:UpdateFunctionCode' ],
      resources: [ 'arn:aws:lambda:us-east-1:123456789012:function:my-function' ]
    })
  ]
}));

```

A capacidade de usar métodos de instância para manipular a configuração de recursos após a instanciação dá às construções L2 muita flexibilidade adicional em relação à camada anterior. As construções L1 também herdam alguns métodos de recursos (como `addPropertyOverride`), mas é somente na camada dois que você obtém métodos projetados especificamente para esse recurso e suas propriedades.

Enumerações

CloudFormation A sintaxe geralmente exige que você especifique muitos detalhes para provisionar um recurso adequadamente. No entanto, a maioria dos casos de uso geralmente é coberta por apenas algumas configurações. Representar essas configurações usando uma série de valores enumerados pode reduzir consideravelmente a quantidade de código necessária.

Por exemplo, no exemplo de código L2 do bucket S3 do início desta seção, você precisa usar a `bucketEncryption` propriedade do CloudFormation modelo para fornecer todos os detalhes, incluindo o nome do algoritmo de criptografia a ser usado. Em vez disso, AWS CDK fornece o `BucketEncryption` enum, que usa as cinco formas mais comuns de criptografia de bucket e permite que você expresse cada uma usando nomes de variáveis únicas.

E quanto aos casos extremos que não são cobertos pelos enums? Um dos objetivos de uma construção L2 é simplificar a tarefa de provisionar um recurso de camada 1, de modo que certos casos extremos que são menos usados podem não ser suportados na camada 2. Para dar suporte a esses casos extremos, o AWS CDK permite manipular as propriedades do CloudFormation recurso subjacente diretamente usando o [addPropertyOverride](#) método. Para saber mais sobre substituições de propriedades, consulte a seção [Melhores práticas](#) deste guia e a seção [Abstrações e escotilhas de escape](#) na documentação. AWS CDK

Classes auxiliares

Às vezes, um enum não consegue realizar a lógica programática necessária para configurar um recurso para um determinado caso de uso. Nessas situações, AWS CDK geralmente oferece uma classe auxiliar. Um enum é um objeto simples que oferece uma série de pares de valores-chave, enquanto uma classe auxiliar oferece todos os recursos de uma classe. TypeScript Uma classe auxiliar ainda pode agir como uma enumeração expondo propriedades estáticas, mas essas propriedades poderiam então ter seus valores definidos internamente com lógica condicional no construtor da classe auxiliar ou em um método auxiliar.

Portanto, embora a `BucketEncryption` enumeração possa reduzir a quantidade de código necessária para definir um algoritmo de criptografia em um bucket do S3, essa mesma estratégia não funcionaria para definir durações de tempo porque simplesmente há muitos valores possíveis para escolher. Criar uma enumeração para cada valor seria muito mais problemático do que vale a pena. Por esse motivo, uma classe auxiliar é usada para as configurações padrão do S3 Object Lock de um bucket do S3, conforme representado pela classe. [ObjectLockRetention](#) `ObjectLockRetention` contém dois métodos estáticos: um para retenção de conformidade e

outro para retenção de governança. Ambos os métodos usam uma instância da [classe auxiliar Duration](#) como argumento para expressar a quantidade de tempo para a qual o bloqueio deve ser configurado.

Outro exemplo é a classe AWS Lambda auxiliar [Runtime](#). À primeira vista, pode parecer que as propriedades estáticas associadas a essa classe podem ser tratadas por um enum. No entanto, nos bastidores, cada valor de propriedade representa uma instância da própria Runtime classe, portanto, a lógica executada no construtor da classe não poderia ser alcançada em um enum.

Construções de camada 3

Se as construções L1 realizam uma tradução literal de CloudFormation recursos em código programático e as construções L2 substituem grande parte da CloudFormation sintaxe detalhada por métodos auxiliares e lógica personalizada, o que as construções L3 fazem? A resposta para isso é limitada apenas pela sua imaginação. Você pode criar a camada 3 para se adequar a qualquer caso de uso específico. Se seu projeto precisar de um recurso que tenha um subconjunto específico de propriedades, você poderá criar uma construção L3 reutilizável para atender a essa necessidade.

As construções L3 são chamadas de padrões dentro do AWS CDK. Um padrão é qualquer objeto que estende a `Construct` classe no AWS CDK (ou estende uma classe que estende a `Construct` classe) para realizar qualquer lógica abstrata além da camada 2. Ao usar a AWS CDK CLI para executar `cdk init` para iniciar um novo AWS CDK projeto, você deve escolher entre três tipos de AWS CDK aplicativos: `app`, `lib` e `sample-app`.

```
Available templates:
* app: Template for a CDK Application
  └─ cdk init app --language=[csharp|fsharp|go|java|javascript|python|typescript]
* lib: Template for a CDK Construct Library
  └─ cdk init lib --language=typescript
* sample-app: Example CDK Application with some constructs
  └─ cdk init sample-app --language=[csharp|fsharp|go|java|javascript|python|typescript]
```

`app` e `sample-app` ambos representam AWS CDK aplicativos clássicos nos quais você cria e implanta CloudFormation pilhas em ambientes da AWS. Ao escolher `lib`, você está optando por construir uma nova construção L3. `app` e `sample-app` permitem que você escolha qualquer idioma AWS CDK compatível, mas você só pode escolher TypeScript com `lib`. Isso ocorre porque o AWS CDK é escrito de forma nativa TypeScript e usa um sistema de código aberto chamado [JSII](#) para traduzir o código original para os outros idiomas suportados. Quando você `lib` opta por iniciar seu projeto, você está optando por criar uma extensão para o AWS CDK.

Qualquer classe que estenda a `Construct` classe pode ser uma construção L3, mas os casos de uso mais comuns da camada 3 são interações de recursos, extensões de recursos e recursos personalizados. A maioria das construções L3 usa um ou mais desses três casos para estender AWS CDK a funcionalidade.

Interações de recursos

Uma solução normalmente emprega vários serviços da AWS que funcionam juntos. Por exemplo, uma CloudFront distribuição da Amazon geralmente usa um bucket S3 como origem e AWS WAF como proteção contra explorações comuns. AWS AppSync e o Amazon API Gateway costumam usar tabelas do Amazon DynamoDB como fontes de dados para suas APIs. Um pipeline AWS CodePipeline geralmente usa o Amazon S3 como fonte e AWS CodeBuild para seus estágios de construção. Nesses casos, geralmente é útil criar uma única construção L3 que manipule o provisionamento de duas ou mais construções L2 interconectadas.

Aqui está um exemplo de uma construção L3 que provisiona uma CloudFront distribuição junto com sua origem S3, um AWS WAF registro do Amazon Route 53 e um certificado AWS Certificate Manager (ACM) para adicionar um endpoint personalizado com criptografia em trânsito, tudo em uma construção reutilizável:

```
// Define the properties passed to the L3 construct
export interface CloudFrontWebsiteProps {
  distributionProps: DistributionProps
  bucketProps: BucketProps
  wafProps: CfnWebAclProps
  zone: IHostedZone
}

// Define the L3 construct
export class CloudFrontWebsite extends Construct {
  public distribution: Distribution

  constructor(
    scope: Construct,
    id: string,
    props: CloudFrontWebsiteProps
  ) {
    super(scope, id);

    const certificate = new Certificate(this, "Certificate", {
      domainName: props.zone.zoneName,
      validation: CertificateValidation.fromDns(props.zone)
    });

    const defaultBehavior = {
      origin: new S3Origin(new Bucket(this, "bucket", props.bucketProps))
    }
  }
}
```

```
const waf = new CfnWebACL(this, "waf", props.wafProps);
this.distribution = new Distribution(this, id, {
  ...props.distributionProps,
  defaultBehavior,
  certificate,
  domainNames: [this.domainName],
  webAclId: waf.attrArn,
});
}
}
```

Observe que o Amazon S3 CloudFront, o Route 53 e o ACM usam construções L2, mas a ACL da web (que define regras para lidar com solicitações da web) usa uma construção L1. Isso ocorre porque AWS CDK é um pacote de código aberto em evolução que não está totalmente completo e ainda não existe uma construção L2. `WebAcl` No entanto, qualquer pessoa pode contribuir com a criação AWS CDK de novas construções L2. Então, até que AWS CDK ofereça uma construção L2 para `WebAcl`, você precisa usar uma construção L1. Para criar um novo site usando a construção `L3CloudFrontWebsite`, você usa o seguinte código:

```
const siteADotCom = new CloudFrontWebsite(stack, "siteA", siteAProps);
const siteBDotCom = new CloudFrontWebsite(stack, "siteB", siteBProps);
const siteCDotCom = new CloudFrontWebsite(stack, "siteC", siteCProps);
```

Neste exemplo, a construção `CloudFront Distribution L2` é exposta como uma propriedade pública da construção L3. Ainda haverá casos em que você precisará expor propriedades L3 como essa, conforme necessário. Na verdade, veremos `Distribution` novamente mais tarde, na seção [Recursos personalizados](#).

AWS CDK Isso inclui alguns exemplos de padrões de interação de recursos, como este. Além do `aws-ecs` pacote que contém as construções L2 para o Amazon Elastic Container Service (Amazon ECS), AWS CDK ele tem um pacote chamado. [aws-ecs-patterns](#) Esse pacote contém várias construções L3 que combinam o Amazon ECS com balanceadores de carga de aplicativos, balanceadores de carga de rede e grupos-alvo, oferecendo diferentes versões predefinidas para o Amazon Elastic Compute Cloud (Amazon) e. EC2 AWS Fargate Como muitos aplicativos sem servidor usam o Amazon ECS somente com o Fargate, essas construções L3 oferecem uma conveniência que pode economizar o tempo dos desenvolvedores e o dinheiro dos clientes.

Extensões de recursos

Alguns casos de uso exigem que os recursos tenham configurações padrão específicas que não sejam nativas da construção L2. No nível da pilha, isso pode ser tratado usando [aspectos](#), mas outra maneira conveniente de dar novos padrões a uma construção L2 é estendendo a camada 2. Como uma construção é qualquer classe que herda a `Construct` classe, e as construções L2 estendem essa classe, você também pode criar uma construção L3 estendendo diretamente uma construção L2.

Isso pode ser especialmente útil para uma lógica de negócios personalizada que ofereça suporte às necessidades singulares de um cliente. Suponhamos que uma empresa tenha um repositório que armazene todo o seu código de AWS Lambda função em um único diretório chamado `src/lambda` e que a maioria das funções do Lambda reutilize sempre o mesmo tempo de execução e nome de manipulador. Em vez de configurar o caminho do código toda vez que você configura uma nova função Lambda, você pode criar uma nova construção L3:

```
export class MyCompanyLambdaFunction extends Function {
  constructor(
    scope: Construct,
    id: string,
    props: Partial<FunctionProps> = {}
  ) {
    super(scope, id, {
      handler: 'index.handler',
      runtime: Runtime.NODEJS_LATEST,
      code: Code.fromAsset(`src/lambda/${props.functionName || id}`),
      ...props
    });
  }
}
```

Você poderia então substituir a `Function` construção L2 em qualquer lugar no repositório da seguinte forma:

```
new MyCompanyLambdaFunction(this, "MyFunction");
new MyCompanyLambdaFunction(this, "MyOtherFunction");
new MyCompanyLambdaFunction(this, "MyThirdFunction", {
  runtime: Runtime.PYTHON_3_11
});
```

Os padrões permitem que você crie novas funções do Lambda em uma única linha, e a construção L3 é configurada para que você ainda possa substituir as propriedades padrão, se necessário.

Estender construções L2 diretamente funciona melhor quando você deseja apenas adicionar novos padrões às construções L2 existentes. Se você também precisar de outra lógica personalizada, é melhor estender a `Construct` classe. A razão para isso decorre do `super` método, que é chamado dentro do construtor. Em classes que estendem outras classes, o `super` método é usado para chamar o construtor da classe mãe, e essa deve ser a primeira coisa que acontece em seu construtor. Isso significa que qualquer manipulação dos argumentos passados ou de outra lógica personalizada só pode acontecer após a criação da construção L2 original. [Se você precisar executar alguma dessas lógicas personalizadas antes de instanciar sua construção L2, é melhor seguir o padrão descrito anteriormente na seção Interações de recursos.](#)

Recursos personalizados

Os [recursos personalizados](#) são um recurso poderoso CloudFormation que permite executar a lógica personalizada a partir de uma função Lambda que é ativada durante a implantação da pilha. Sempre que precisar de algum processo durante a implantação que não seja diretamente suportado CloudFormation, você pode usar um recurso personalizado para que isso aconteça. AWS CDK Oferece classes que também permitem criar recursos personalizados de forma programática. Ao usar recursos personalizados em um construtor L3, você pode criar uma construção a partir de quase tudo.

Uma das vantagens de usar a Amazon CloudFront são seus fortes recursos globais de armazenamento em cache. Se você quiser redefinir manualmente esse cache para que seu site reflita imediatamente as novas alterações feitas em sua origem, você pode usar uma [CloudFront invalidação](#). No entanto, as invalidações são processos executados em uma CloudFront distribuição em vez de serem propriedades de uma CloudFront distribuição. Eles podem ser criados e aplicados a uma distribuição existente a qualquer momento, portanto, não são parte nativa do processo de provisionamento e implantação.

Nesse cenário, talvez você queira criar e executar uma invalidação após cada atualização na origem de uma distribuição. Por causa dos recursos personalizados, você pode criar uma construção L3 mais ou menos assim:

```
export interface CloudFrontInvalidationProps {
  distribution: Distribution
  region?: string
}
```

```

    paths?: string[]
  }

export class CloudFrontInvalidation extends Construct {
  constructor(
    scope: Construct,
    id: string,
    props: CloudFrontInvalidationProps
  ) {
    super(scope, id);
    const policy = AwsCustomResourcePolicy.fromSdkCalls({
      resources: AwsCustomResourcePolicy.ANY_RESOURCE
    });
    new AwsCustomResource(scope, `${id}Invalidation`, {
      policy,
      onUpdate: {
        service: 'CloudFront',
        action: 'createInvalidation',
        region: props.region || 'us-east-1',
        physicalResourceId:
PhysicalResourceId.fromResponse('Invalidation.Id'),
        parameters: {
          DistributionId: props.distribution.distributionId,
          InvalidationBatch: {
            Paths: {
              Quantity: props.paths?.length || 1,
              Items: props.paths || ['/*']
            },
            CallerReference: crypto.randomBytes(5).toString('hex')
          }
        }
      }
    }
  }
}

```

Usando a distribuição que criamos anteriormente na construção `CloudFrontWebsite L3`, você pode fazer isso com muita facilidade:

```

new CloudFrontInvalidation(this, 'MyInvalidation', {
  distribution: siteADotCom.distribution
});

```

Essa construção L3 usa uma construção AWS CDK L3 chamada [AwsCustomResource](#) para criar um recurso personalizado que executa a lógica personalizada. `AwsCustomResource` é muito conveniente quando você precisa fazer exatamente uma chamada do AWS SDK, pois permite que você faça isso sem precisar escrever nenhum código Lambda. Se você tiver requisitos mais complexos e quiser implementar sua própria lógica, poderá usar a [CustomResource](#) classe básica diretamente.

Outro bom exemplo do AWS CDK uso de uma construção L3 de recurso personalizada é a implantação do [bucket do S3](#). A função Lambda criada pelo recurso personalizado dentro do construtor dessa construção L3 adiciona funcionalidades que não CloudFormation seriam capazes de manipular de outra forma: ela adiciona e atualiza objetos em um bucket do S3. Sem a implantação do bucket do S3, você não conseguiria colocar conteúdo no bucket do S3 que acabou de criar como parte da sua pilha, o que seria muito inconveniente.

O melhor exemplo de como AWS CDK eliminar a necessidade de escrever uma grande quantidade de CloudFormation sintaxe é esse básico: `S3BucketDeployment`

```
new BucketDeployment(this, 'BucketObjects', {
    sources: [Source.asset('./path/to/amzn-s3-demo-bucket')],
    destinationBucket: amzn-s3-demo-bucket
});
```

Compare isso com o CloudFormation código que você precisaria escrever para realizar a mesma coisa:

```
"lambdapolicyA5E98E09": {
  "Type": "AWS::IAM::Policy",
  "Properties": {
    "PolicyDocument": {
      "Statement": [
        {
          "Action": "lambda:UpdateFunctionCode",
          "Effect": "Allow",
          "Resource": "arn:aws:lambda:us-east-1:123456789012:function:my-function"
        }
      ],
      "Version": "2012-10-17"
    },
    "PolicyName": "lambdaPolicy",
    "Roles": [
      {
```

```

    "Ref": "myiamroleF09C7974"
  }
]
},
"Metadata": {
  "aws:cdk:path": "CdkScratchStack/lambda-policy/Resource"
}
},
"BucketObjectsAwsCliLayer8C081206": {
  "Type": "AWS::Lambda::LayerVersion",
  "Properties": {
    "Content": {
      "S3Bucket": {
        "Fn::Sub": "cdk-hnb659fds-assets-${AWS::AccountId}-${AWS::Region}"
      },
      "S3Key": "e2277687077a2abf9ae1af1cc9565e6715e2ebb62f79ec53aa75a1af9298f642.zip"
    },
    "Description": "/opt/awscli/aws"
  },
  "Metadata": {
    "aws:cdk:path": "CdkScratchStack/BucketObjects/AwsCliLayer/Resource",
    "aws:asset:path":
"asset.e2277687077a2abf9ae1af1cc9565e6715e2ebb62f79ec53aa75a1af9298f642.zip",
    "aws:asset:is-bundled": false,
    "aws:asset:property": "Content"
  }
},
"BucketObjectsCustomResourceB12E6837": {
  "Type": "Custom::CDKBucketDeployment",
  "Properties": {
    "ServiceToken": {
      "Fn::GetAtt": [
        "CustomCDKBucketDeployment8693BB64968944B69Aafb0CC9EB8756C81C01536",
        "Arn"
      ]
    }
  },
  "SourceBucketNames": [
    {
      "Fn::Sub": "cdk-hnb659fds-assets-${AWS::AccountId}-${AWS::Region}"
    }
  ],
  "SourceObjectKeys": [
    "f888a9d977f0b5bdbbc04a1f8f07520ede6e00d4051b9a6a250860a1700924f26.zip"
  ],

```

```

    "DestinationBucketName": {
      "Ref": "amzn-s3-demo-bucket77F80CC0"
    },
    "Prune": true
  },
  "UpdateReplacePolicy": "Delete",
  "DeletionPolicy": "Delete",
  "Metadata": {
    "aws:cdk:path": "CdkScratchStack/BucketObjects/CustomResource/Default"
  }
},
"CustomCDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756CServiceRole89A01265": {
  "Type": "AWS::IAM::Role",
  "Properties": {
    "AssumeRolePolicyDocument": {
      "Statement": [
        {
          "Action": "sts:AssumeRole",
          "Effect": "Allow",
          "Principal": {
            "Service": "lambda.amazonaws.com"
          }
        }
      ],
      "Version": "2012-10-17"
    },
    "ManagedPolicyArns": [
      {
        "Fn::Join": [
          "",
          [
            "arn:",
            {
              "Ref": "AWS::Partition"
            },
            ":iam::aws:policy/service-role/AWSLambdaBasicExecutionRole"
          ]
        ]
      }
    ]
  }
},
"Metadata": {
  "aws:cdk:path": "CdkScratchStack/
Custom::CDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756C/ServiceRole/Resource"

```

```

    }
  },

  "CustomCDKBucketDeployment8693BB64968944B69Aafb0cc9EB8756CServiceRoleDefaultPolicy88902FDF":
  {
    "Type": "AWS::IAM::Policy",
    "Properties": {
      "PolicyDocument": {
        "Statement": [
          {
            "Action": [
              "s3:GetBucket*",
              "s3:GetObject*",
              "s3:List*"
            ],
            "Effect": "Allow",
            "Resource": [
              {
                "Fn::Join": [
                  "",
                  [
                    "arn:",
                    {
                      "Ref": "AWS::Partition"
                    },
                    ":s3::",
                    {
                      "Fn::Sub": "cdk-hnb659fds-assets-${AWS::AccountId}-${AWS::Region}"
                    },
                    "/*"
                  ]
                ]
              }
            ],
          },
          {
            "Fn::Join": [
              "",
              [
                "arn:",
                {
                  "Ref": "AWS::Partition"
                },
                ":s3::",
                {
                  "Fn::Sub": "cdk-hnb659fds-assets-${AWS::AccountId}-${AWS::Region}"
                }
              ]
            ]
          }
        ]
      }
    }
  }
}

```

```
    }
  ]
]
}
]
},
{
  "Action": [
    "s3:Abort*",
    "s3:DeleteObject*",
    "s3:GetBucket*",
    "s3:GetObject*",
    "s3:List*",
    "s3:PutObject",
    "s3:PutObjectLegalHold",
    "s3:PutObjectRetention",
    "s3:PutObjectTagging",
    "s3:PutObjectVersionTagging"
  ],
  "Effect": "Allow",
  "Resource": [
    {
      "Fn::GetAtt": [
        "amzn3demobucket77F80CC0",
        "Arn"
      ]
    },
    {
      "Fn::Join": [
        "",
        [
          {
            "Fn::GetAtt": [
              "amzn3demobucket77F80CC0",
              "Arn"
            ]
          },
          "/"
        ]
      ]
    }
  ]
}
],
```

```

    "Version": "2012-10-17"
  },
  "PolicyName":
"CustomCDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756CServiceRoleDefaultPolicy88902FDF",
  "Roles": [
    {
      "Ref":
"CustomCDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756CServiceRole89A01265"
    }
  ],
  "Metadata": {
    "aws:cdk:path": "CdkScratchStack/
Custom::CDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756C/ServiceRole/DefaultPolicy/
Resource"
  }
},
"CustomCDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756C81C01536": {
  "Type": "AWS::Lambda::Function",
  "Properties": {
    "Code": {
      "S3Bucket": {
        "Fn::Sub": "cdk-hnb659fds-assets-${AWS::AccountId}-${AWS::Region}"
      },
      "S3Key": "9eb41a5505d37607ac419321497a4f8c21cf0ee1f9b4a6b29aa04301aea5c7fd.zip"
    },
    "Role": {
      "Fn::GetAtt": [
        "CustomCDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756CServiceRole89A01265",
        "Arn"
      ]
    },
    "Environment": {
      "Variables": {
        "AWS_CA_BUNDLE": "/etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem"
      }
    },
    "Handler": "index.handler",
    "Layers": [
      {
        "Ref": "BucketObjectsAwsCliLayer8C081206"
      }
    ],
    "Runtime": "python3.9",

```

```
    "Timeout": 900
  },
  "DependsOn": [

    "CustomCDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756CServiceRoleDefaultPolicy88902FDF",
    "CustomCDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756CServiceRole89A01265"
  ],
  "Metadata": {
    "aws:cdk:path": "CdkScratchStack/
Custom::CDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756C/Resource",
    "aws:asset:path":
"asset.9eb41a5505d37607ac419321497a4f8c21cf0ee1f9b4a6b29aa04301aea5c7fd",
    "aws:asset:is-bundled": false,
    "aws:asset:property": "Code"
  }
}
```

4 linhas versus 241 linhas é uma grande diferença! E esse é apenas um exemplo do que é possível quando você aproveita a camada 3 para personalizar suas pilhas.

Práticas recomendadas

Construções L1

- Nem sempre é possível evitar o uso direto de construções L1, mas você deve evitá-las sempre que possível. Se uma construção L2 específica não suportar seu caso extremo, você pode explorar essas duas opções em vez de usar a construção L1 diretamente:
 - `AccessDefaultChild`: Se a CloudFormation propriedade de que você precisa não estiver disponível em uma construção L2, você poderá acessar a construção L1 subjacente usando `L2Construct.node.defaultChild`. Você pode atualizar qualquer propriedade pública da construção L1 acessando-as por meio dessa propriedade, em vez de se dar ao trabalho de criar a construção L1 sozinho.
 - Use substituições de propriedade: e se a propriedade que você deseja atualizar não for pública? A melhor saída de emergência que permite AWS CDK fazer qualquer coisa que um CloudFormation modelo possa fazer é usar um método que está disponível em cada construção L1: [addPropertyOverride](#). Você pode manipular sua pilha no nível do CloudFormation modelo passando o nome e o valor da CloudFormation propriedade diretamente para esse método.

Construções L2

- Lembre-se de aproveitar os métodos auxiliares que as construções L2 geralmente oferecem. Com a camada 2, você não precisa passar todas as propriedades na instanciação. Os métodos auxiliares L2 podem tornar o provisionamento de recursos exponencialmente mais conveniente, especialmente quando a lógica condicional é necessária. Um dos métodos auxiliares mais convenientes é derivado da classe [Grant](#). Essa classe não é usada diretamente, mas muitas construções L2 a usam para fornecer métodos auxiliares que tornam as permissões muito mais fáceis de implementar. Por exemplo, se você quiser dar permissão a uma função Lambda L2 para acessar um bucket L2 S3, você pode `s3Bucket.grantReadWrite(lambdaFunction)` chamar em vez de criar uma nova função e política.

Construções L3

- Embora as construções L3 possam ser muito convenientes quando você deseja tornar suas pilhas mais reutilizáveis e personalizáveis, recomendamos que você as use com cuidado. Considere qual tipo de construção L3 você precisa ou se você realmente precisa de uma construção L3:

- Se você não estiver interagindo diretamente com AWS os recursos, geralmente é mais apropriado criar uma classe auxiliar em vez de estender a classe. `Construct` Isso ocorre porque a `Construct` classe executa muitas ações por padrão que são necessárias somente se você estiver interagindo diretamente com AWS os recursos. Portanto, se você não precisa que essas ações sejam executadas, é mais eficiente evitá-las.
- Se você determinar que a criação de uma nova construção L3 é apropriada, na maioria dos casos você desejará estender a `Construct` classe diretamente. Estenda outras construções L2 somente quando quiser atualizar as propriedades padrão da construção. Se outras construções L2 ou lógica personalizada estiverem envolvidas, estenda `Construct` diretamente e instancie todos os recursos dentro do construtor.

Perguntas frequentes

Não posso usar as camadas AWS CDK sem entender?

Você com certeza pode. Mas, como acontece com as ferramentas mais poderosas, elas AWS CDK se tornam mais poderosas quanto mais você as conhece. Aprender como as AWS CDK camadas interagem libera um novo nível de compreensão que ajuda a simplificar suas implantações de pilha muito além do que você pode fazer com apenas o conhecimento básico. AWS CDK

Posso fazer construções L2 a partir de L1 da mesma forma que faço construções L3 a partir de L2?

Se um recurso já tiver uma construção L2, recomendamos que você use essa construção e faça suas personalizações na camada 3. Isso ocorre porque muitas pesquisas já foram feitas para descobrir as melhores maneiras de configurar construções L2 existentes para um recurso específico. No entanto, existem várias construções L1 cujas construções L2 ainda não existem. Nesses casos, incentivamos você a criar suas próprias construções L2 e compartilhá-las com outras pessoas, tornando-se um colaborador da biblioteca de AWS CDK código aberto. Você pode encontrar tudo o que precisa para começar nas [diretrizes de contribuição](#) do AWS CDK.

Quais AWS recursos ainda não têm construções L2 oficiais?

[O número de AWS recursos que não têm construções L2 está diminuindo a cada dia, mas se você estiver interessado em ajudar a criar uma construção L2 para um desses recursos, visite a AWS CDK Referência da API.](#) Veja a lista de recursos no painel esquerdo. Os recursos que têm o sobrescrito 1 ao lado de seus nomes não têm construções L2 oficiais.

Posso criar uma construção L2 ou L3 em qualquer idioma compatível? AWS CDK

O AWS CDK suporta várias linguagens de programação TypeScript JavaScript, incluindo Python, Java, C# e Go. Você pode criar suas construções L3 pessoais usando o AWS CDK código compilado na linguagem relevante. No entanto, se você quiser contribuir AWS CDK ou criar AWS CDK construções nativas, você deve usar TypeScript. Isso ocorre porque TypeScript é o único

idioma nativo do AWS CDK. As AWS CDK versões para outras linguagens são criadas a partir do TypeScript código nativo usando uma AWS biblioteca chamada [JSii](#).

Onde posso encontrar construções L3 existentes fora do? AWS CDK

Há muitos locais para compartilhar aqui, mas você pode encontrar muitas das construções mais populares no site da [AWS Solutions Constructs](#) e na AWS CDK seção do [Construct Hub](#).

Recursos

- [AWS CDK API Reference](#)
- [AWS CloudFormation especificação de recursos](#)
- [AWS CDK constrói documentação](#)
- [AWS CDK abstrações e escotilhas de escape](#)
- [Aproveite as construções L2 para reduzir a complexidade do seu AWS CDK aplicativo](#) (AWS postagem no blog)
- [AWS CloudFormation recursos personalizados](#)
- [AWS Soluções e construções](#)
- [Construct Hub](#)
- [AWS CDK Exemplos](#) (GitHub repositório)

Histórico do documento

A tabela a seguir descreve alterações significativas feitas neste guia. Se desejar receber notificações sobre futuras atualizações, inscreva-se em um [feed RSS](#).

Alteração	Descrição	Data
Publicação inicial	—	4 de dezembro de 2023

AWS Glossário de orientação prescritiva

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

Números

7 Rs

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

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

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

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

A

ABAC

Consulte controle de [acesso baseado em atributos](#).

serviços abstratos

Veja os [serviços gerenciados](#).

ACID

Veja [atomicidade, consistência, isolamento, durabilidade](#).

migração ativa-ativa

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

migração ativa-passiva

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

função agregada

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

AI

Veja a [inteligência artificial](#).

AIOps

Veja as [operações de inteligência artificial](#).

anonimização

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

antipadrões

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

controle de aplicativos

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

portfólio de aplicações

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

inteligência artificial (IA)

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

operações de inteligência artificial (AIOps)

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

criptografia assimétrica

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

atomicidade, consistência, isolamento, durabilidade (ACID)

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

controle de acesso por atributo (ABAC)

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

fonte de dados autorizada

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

Zona de disponibilidade

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

AWS Estrutura de adoção da nuvem (AWS CAF)

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

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

Uma ferramenta que avalia as cargas de trabalho de migração do banco de dados, recomenda estratégias de migração e fornece estimativas de trabalho. AWS O WQF está incluído com AWS

Schema Conversion Tool (AWS SCT). Ela analisa esquemas de banco de dados e objetos de código, código de aplicações, dependências e características de performance, além de fornecer relatórios de avaliação.

B

bot ruim

Um [bot](#) destinado a perturbar ou causar danos a indivíduos ou organizações.

BCP

Veja o [planejamento de continuidade de negócios](#).

gráfico de comportamento

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

sistema big-endian

Um sistema que armazena o byte mais significativo antes. Veja também [endianness](#).

classificação binária

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

filtro de bloom

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

blue/green deployment (implantação azul/verde)

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

bot

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

botnet

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

ramo

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

acesso em vidro quebrado

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

estratégia brownfield

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

cache do buffer

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

capacidade de negócios

O que uma empresa faz para gerar valor (por exemplo, vendas, atendimento ao cliente ou marketing). As arquiteturas de microsserviços e as decisões de desenvolvimento podem

ser orientadas por recursos de negócios. Para obter mais informações, consulte a seção [Organizados de acordo com as capacidades de negócios](#) do whitepaper [Executar microsserviços containerizados na AWS](#).

planejamento de continuidade de negócios (BCP)

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

C

CAF

Consulte [Estrutura de adoção da AWS nuvem](#).

implantação canária

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

CCoE

Veja o [Centro de Excelência em Nuvem](#).

CDC

Veja [a captura de dados de alterações](#).

captura de dados de alterações (CDC)

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

engenharia do caos

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

CI/CD

Veja a [integração e a entrega contínuas](#).

classificação

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

criptografia no lado do cliente

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

Centro de excelência em nuvem (CCoE)

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

computação em nuvem

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

modelo operacional em nuvem

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

estágios de adoção da nuvem

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

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

Esses estágios foram definidos por Stephen Orban na postagem do blog [The Journey Toward Cloud-First & the Stages of Adoption](#) no blog de estratégia Nuvem AWS empresarial. Para obter

informações sobre como eles se relacionam com a estratégia de AWS migração, consulte o [guia de preparação para migração](#).

CMDB

Consulte o [banco de dados de gerenciamento de configuração](#).

repositório de código

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

cache frio

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

dados frios

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

visão computacional (CV)

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

desvio de configuração

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

banco de dados de gerenciamento de configuração (CMDB)

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

pacote de conformidade

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

integração contínua e entrega contínua (CI/CD)

O processo de automatizar os estágios de origem, criação, teste, preparação e produção do processo de lançamento do software. CI/CD is commonly described as a pipeline. CI/CD pode ajudá-lo a automatizar processos, melhorar a produtividade, melhorar a qualidade do código e entregar com mais rapidez. Para obter mais informações, consulte [Benefícios da entrega contínua](#). CD também pode significar implantação contínua. Para obter mais informações, consulte [Entrega contínua versus implantação contínua](#).

CV

Veja [visão computacional](#).

D

dados em repouso

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

classificação de dados

Um processo para identificar e categorizar os dados em sua rede com base em criticalidade e confidencialidade. É um componente crítico de qualquer estratégia de gerenciamento de riscos de segurança cibernética, pois ajuda a determinar os controles adequados de proteção e retenção para os dados. A classificação de dados é um componente do pilar de segurança no AWS Well-Architected Framework. Para obter mais informações, consulte [Classificação de dados](#).

desvio de dados

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

dados em trânsito

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

malha de dados

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

minimização de dados

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

perímetro de dados

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

pré-processamento de dados

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

proveniência dos dados

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

titular dos dados

Um indivíduo cujos dados estão sendo coletados e processados.

data warehouse

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

linguagem de definição de dados (DDL)

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

linguagem de manipulação de dados (DML)

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

DDL

Consulte a [linguagem de definição de banco](#) de dados.

deep ensemble

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

Aprendizado profundo

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

defense-in-depth

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

administrador delegado

Em AWS Organizations, um serviço compatível pode registrar uma conta de AWS membro para administrar as contas da organização e gerenciar as permissões desse serviço. Essa conta é chamada de administrador delegado para esse serviço. Para obter mais informações e uma lista de serviços compatíveis, consulte [Serviços que funcionam com o AWS Organizations](#) na documentação do AWS Organizations .

implantação

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

ambiente de desenvolvimento

Veja o [ambiente](#).

controle detectivo

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

mapeamento do fluxo de valor de desenvolvimento (DVSM)

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

gêmeo digital

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

tabela de dimensões

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

desastre

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

Recuperação de desastres (RD)

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

DML

Veja a [linguagem de manipulação de banco](#) de dados.

design orientado por domínio

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

DR

Veja a [recuperação de desastres](#).

detecção de deriva

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

DVSM

Veja o [mapeamento do fluxo de valor do desenvolvimento](#).

E

EDA

Veja a [análise exploratória de dados](#).

EDI

Veja [intercâmbio eletrônico de dados](#).

computação de borda

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

intercâmbio eletrônico de dados (EDI)

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

Criptografia

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

chave de criptografia

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

endianismo

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

endpoint

Veja o [endpoint do serviço](#).

serviço de endpoint

Um serviço que pode ser hospedado em uma nuvem privada virtual (VPC) para ser compartilhado com outros usuários. Você pode criar um serviço de endpoint com AWS PrivateLink e conceder permissões a outros diretores Contas da AWS ou a AWS Identity and Access Management (IAM). Essas contas ou entidades principais podem se conectar ao serviço de endpoint de maneira privada criando endpoints da VPC de interface. Para obter mais informações, consulte [Criar um serviço de endpoint](#) na documentação do Amazon Virtual Private Cloud (Amazon VPC).

planejamento de recursos corporativos (ERP)

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

criptografia envelopada

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

ambiente

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

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

epic

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

ERP

Veja o [planejamento de recursos corporativos](#).

análise exploratória de dados (EDA)

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

F

tabela de fatos

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

falham rapidamente

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

limite de isolamento de falhas

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

ramificação de recursos

Veja a [filial](#).

recursos

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

importância do recurso

O quanto um recurso é importante para as previsões de um modelo. Isso geralmente é expresso como uma pontuação numérica que pode ser calculada por meio de várias técnicas, como Shapley Additive Explanations (SHAP) e gradientes integrados. Para obter mais informações, consulte [Interpretabilidade do modelo de aprendizado de máquina com AWS](#).

transformação de recursos

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

solicitação rápida

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

FGAC

Veja o [controle de acesso refinado](#).

Controle de acesso refinado (FGAC)

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

migração flash-cut

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

FM

Veja o [modelo da fundação](#).

modelo de fundação (FM)

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

G

IA generativa

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

bloqueio geográfico

Veja as [restrições geográficas](#).

restrições geográficas (bloqueio geográfico)

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

Fluxo de trabalho do GitFlow

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

imagem dourada

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

estratégia greenfield

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

barreira de proteção

Uma regra de alto nível que ajuda a governar recursos, políticas e conformidade em todas as unidades organizacionais (OUs). Barreiras de proteção preventivas impõem políticas para garantir o alinhamento a padrões de conformidade. Elas são implementadas usando políticas de controle de serviço e limites de permissões do IAM. Barreiras de proteção detectivas detectam violações de políticas e problemas de conformidade e geram alertas para remediação. Eles são implementados usando AWS Config, AWS Security Hub, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector e verificações personalizadas AWS Lambda .

H

HA

Veja a [alta disponibilidade](#).

migração heterogênea de bancos de dados

Migrar seu banco de dados de origem para um banco de dados de destino que usa um mecanismo de banco de dados diferente (por exemplo, Oracle para Amazon Aurora). A migração heterogênea geralmente faz parte de um esforço de redefinição da arquitetura, e converter

o esquema pode ser uma tarefa complexa. [O AWS fornece o AWS SCT](#) para ajudar nas conversões de esquemas.

alta disponibilidade (HA)

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

modernização de historiador

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

dados de retenção

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

migração homogênea de bancos de dados

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

dados quentes

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

hotfix

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

período de hipercuidados

Imediatamente após a substituição, o período em que uma equipe de migração gerencia e monitora as aplicações migradas na nuvem para resolver quaisquer problemas. Normalmente,

a duração desse período é de 1 a 4 dias. No final do período de hiper cuidados, a equipe de migração normalmente transfere a responsabilidade pelas aplicações para a equipe de operações de nuvem.

eu

laC

Veja a [infraestrutura como código](#).

Política baseada em identidade

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

aplicação ociosa

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

IloT

Veja a [Internet das Coisas industrial](#).

infraestrutura imutável

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

VPC de entrada (admissão)

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

migração incremental

Uma estratégia de substituição na qual você migra a aplicação em pequenas partes, em vez de realizar uma única substituição completa. Por exemplo, é possível mover inicialmente

apenas alguns microsserviços ou usuários para o novo sistema. Depois de verificar se tudo está funcionando corretamente, mova os microsserviços ou usuários adicionais de forma incremental até poder descomissionar seu sistema herdado. Essa estratégia reduz os riscos associados a migrações de grande porte.

Indústria 4.0

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

infraestrutura

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

Infraestrutura como código (IaC)

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

Internet industrial das coisas (IIoT)

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

VPC de inspeção

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

Internet das Coisas (IoT)

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

interpretabilidade

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

IoT

Consulte [Internet das Coisas](#).

Biblioteca de informações de TI (ITIL)

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

Gerenciamento de serviços de TI (ITSM)

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

ITIL

Consulte [a biblioteca de informações](#) de TI.

ITSM

Veja o [gerenciamento de serviços de TI](#).

L

controle de acesso baseado em etiqueta (LBAC)

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

zona de pouso

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

informações sobre zonas de pouso, consulte [Configurar um ambiente da AWS com várias contas seguro e escalável](#).

modelo de linguagem grande (LLM)

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

migração de grande porte

Uma migração de 300 servidores ou mais.

LBAC

Veja controle de [acesso baseado em etiquetas](#).

privilégio mínimo

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

mover sem alterações (lift-and-shift)

Veja [7 Rs](#).

sistema little-endian

Um sistema que armazena o byte menos significativo antes. Veja também [endianness](#).

LLM

Veja [um modelo de linguagem grande](#).

ambientes inferiores

Veja o [ambiente](#).

M

machine learning (ML)

Um tipo de inteligência artificial que usa algoritmos e técnicas para reconhecimento e aprendizado de padrões. O ML analisa e aprende com dados gravados, por exemplo, dados da

Internet das Coisas (IoT), para gerar um modelo estatístico baseado em padrões. Para obter mais informações, consulte [Machine learning](#).

ramificação principal

Veja a [filial](#).

malware

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

serviços gerenciados

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

sistema de execução de manufatura (MES)

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

MAP

Consulte [Migration Acceleration Program](#).

mecanismo

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

conta de membro

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

MES

Veja o [sistema de execução de manufatura](#).

Transporte de telemetria de enfileiramento de mensagens (MQTT)

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

microsserviço

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

arquitetura de microsserviços

Uma abordagem à criação de aplicações com componentes independentes que executam cada processo de aplicação como um microsserviço. Esses microsserviços se comunicam por meio de uma interface bem definida usando leveza. APIs Cada microsserviço nessa arquitetura pode ser atualizado, implantado e escalado para atender à demanda por funções específicas de uma aplicação. Para obter mais informações, consulte [Implementação de microsserviços em. AWS](#)

Programa de Aceleração da Migração (MAP)

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

migração em escala

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

fábrica de migração

Equipes multifuncionais que simplificam a migração de workloads por meio de abordagens automatizadas e ágeis. As equipes da fábrica de migração geralmente incluem operações,

analistas e proprietários de negócios, engenheiros de migração, desenvolvedores e DevOps profissionais que trabalham em sprints. Entre 20 e 50% de um portfólio de aplicações corporativas consiste em padrões repetidos que podem ser otimizados por meio de uma abordagem de fábrica. Para obter mais informações, consulte [discussão sobre fábricas de migração](#) e o [guia do Cloud Migration Factory](#) neste conjunto de conteúdo.

metadados de migração

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

padrão de migração

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

Avaliação de Portfólio para Migração (MPA)

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

Avaliação de Preparação para Migração (MRA)

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

estratégia de migração

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

ML

Veja o [aprendizado de máquina](#).

modernização

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

avaliação de preparação para modernização

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

aplicações monolíticas (monólitos)

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

MAPA

Consulte [Avaliação do portfólio de migração](#).

MQTT

Consulte Transporte de [telemetria de enfileiramento de](#) mensagens.

classificação multiclasse

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

infraestrutura mutável

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

O

OAC

Veja o [controle de acesso de origem](#).

CARVALHO

Veja a [identidade de acesso de origem](#).

OCM

Veja o [gerenciamento de mudanças organizacionais](#).

migração offline

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

OI

Veja a [integração de operações](#).

OLA

Veja o [contrato em nível operacional](#).

migração online

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

OPC-UA

Consulte [Comunicação de processo aberto — Arquitetura unificada](#).

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

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

acordo de nível operacional (OLA)

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

análise de prontidão operacional (ORR)

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

tecnologia operacional (OT)

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

integração de operações (OI)

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

trilha organizacional

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

gerenciamento de alterações organizacionais (OCM)

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

controle de acesso de origem (OAC)

Em CloudFront, uma opção aprimorada para restringir o acesso para proteger seu conteúdo do Amazon Simple Storage Service (Amazon S3). O OAC oferece suporte a todos os buckets

S3 Regiões da AWS, criptografia do lado do servidor com AWS KMS (SSE-KMS) e solicitações dinâmicas ao bucket S3. PUT DELETE

Identidade do acesso de origem (OAI)

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

ORR

Veja a [análise de prontidão operacional](#).

OT

Veja a [tecnologia operacional](#).

VPC de saída (egresso)

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

P

limite de permissões

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

Informações de identificação pessoal (PII)

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

PII

Veja as [informações de identificação pessoal](#).

manual

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

PLC

Consulte [controlador lógico programável](#).

AMEIXA

Veja o gerenciamento [do ciclo de vida do produto](#).

política

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

persistência poliglota

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

avaliação do portfólio

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

predicado

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

pressão de predicados

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

controle preventivo

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

principal (entidade principal)

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

privacidade por design

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

zonas hospedadas privadas

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

controle proativo

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

gerenciamento do ciclo de vida do produto (PLM)

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

ambiente de produção

Veja o [ambiente](#).

controlador lógico programável (PLC)

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

encadeamento imediato

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

pseudonimização

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

publish/subscribe (pub/sub)

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

Q

plano de consulta

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

regressão de planos de consultas

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

R

Matriz RACI

Veja [responsável, responsável, consultado, informado \(RACI\)](#).

RAG

Consulte [Geração Aumentada de Recuperação](#).

ransomware

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

Matriz RASCI

Veja [responsável, responsável, consultado, informado \(RACI\)](#).

RCAC

Veja o [controle de acesso por linha e coluna](#).

réplica de leitura

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

rearquiteta

Veja [7 Rs](#).

objetivo de ponto de recuperação (RPO).

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

objetivo de tempo de recuperação (RTO)

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

refatorar

Veja [7 Rs](#).

Região

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

regressão

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

redefinir a hospedagem

Veja [7 Rs](#).

versão

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

realocar

Veja [7 Rs](#).

redefinir a plataforma

Veja [7 Rs](#).

recomprar

Veja [7 Rs](#).

resiliência

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

política baseada em recurso

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

matriz responsável, accountable, consultada, informada (RACI)

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

controle responsivo

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

reter

Veja [7 Rs](#).

aposentar-se

Veja [7 Rs](#).

Geração Aumentada de Recuperação (RAG)

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

alternância

O processo de atualizar periodicamente um [segredo](#) para dificultar o acesso das credenciais por um invasor.

controle de acesso por linha e coluna (RCAC)

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

RPO

Veja o [objetivo do ponto de recuperação](#).

RTO

Veja o [objetivo do tempo de recuperação](#).

runbook

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

S

SAML 2.0

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

SCADA

Veja [controle de supervisão e aquisição de dados](#).

SCP

Veja a [política de controle de serviços](#).

secret

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

segurança por design

Uma abordagem de engenharia de sistemas que leva em conta a segurança em todo o processo de desenvolvimento.

controle de segurança

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

fortalecimento da segurança

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

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

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

automação de resposta de segurança

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

Criptografia do lado do servidor

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

política de controle de serviços (SCP)

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

service endpoint (endpoint de serviço)

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

acordo de serviço (SLA)

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

indicador de nível de serviço (SLI)

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

objetivo de nível de serviço (SLO)

Uma métrica alvo que representa a integridade de um serviço, conforme medida por um indicador de [nível de serviço](#).

modelo de responsabilidade compartilhada

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

SIEM

Veja [informações de segurança e sistema de gerenciamento de eventos](#).

ponto único de falha (SPOF)

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

SLA

Veja o contrato [de nível de serviço](#).

ESGUIO

Veja o indicador [de nível de serviço](#).

SLO

Veja o objetivo do [nível de serviço](#).

split-and-seed modelo

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

CUSPE

Veja [um único ponto de falha](#).

esquema de estrelas

Uma estrutura organizacional de banco de dados que usa uma grande tabela de fatos para armazenar dados transacionais ou medidos e usa uma ou mais tabelas dimensionais menores para armazenar atributos de dados. Essa estrutura foi projetada para uso em um [data warehouse](#) ou para fins de inteligência comercial.

padrão strangler fig

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

sub-rede

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

controle de supervisão e aquisição de dados (SCADA)

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

symmetric encryption (criptografia simétrica)

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

testes sintéticos

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

prompt do sistema

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

T

tags

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

variável-alvo

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

lista de tarefas

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

ambiente de teste

Veja o [ambiente](#).

treinamento

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

gateway de trânsito

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

fluxo de trabalho baseado em troncos

Uma abordagem na qual os desenvolvedores criam e testam recursos localmente em uma ramificação de recursos e, em seguida, mesclam essas alterações na ramificação principal. A ramificação principal é então criada para os ambientes de desenvolvimento, pré-produção e produção, sequencialmente.

Acesso confiável

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

tuning (ajustar)

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

equipe de duas pizzas

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

U

incerteza

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

tarefas indiferenciadas

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

ambientes superiores

Veja o [ambiente](#).

V

aspiração

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

controle de versões

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

emparelhamento da VPC

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

Vulnerabilidade

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

W

cache quente

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

dados mornos

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

função de janela

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

workload

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

workstreams

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

MINHOCA

Veja [escrever uma vez, ler muitas](#).

WQF

Consulte [Estrutura de qualificação AWS da carga de](#) trabalho.

escreva uma vez, leia muitas (WORM)

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

Z

exploração de dia zero

Um ataque, geralmente malware, que tira proveito de uma vulnerabilidade de [dia zero](#).

vulnerabilidade de dia zero

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

aviso zero-shot

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

eficácia da solicitação zero depende da complexidade da tarefa e da qualidade da solicitação. Veja também a solicitação [de algumas fotos](#).

aplicação zumbi

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

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