



Projetando para HA e resiliência em aplicativos Amazon EKS

AWS Orientação prescritiva



AWS Orientação prescritiva: Projetando para HA e resiliência em aplicativos Amazon EKS

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

As marcas comerciais e imagens de marcas 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
Design de HA e resiliência	2
Divida as cargas de trabalho	2
Use restrições de dispersão da topologia do pod	3
Afinidade e antiafinidade do pod	7
Orçamento para interrupção do pod	9
Sondas e verificações de saúde	9
Sonda de inicialização	10
Sonda de vivacidade	10
Sonda de prontidão	10
Verificações de integridade do recurso de entrada e do balanceador de carga	11
Ganchos para o ciclo de vida do contêiner	12
Entenda o despejo de lagoas durante interrupções zonais	14
Implementando a mudança de zona do Amazon EKS para melhorar a resiliência	14
Entendendo o mecanismo de mudança zonal	14
Métodos de ativação por mudança zonal	15
Pré-requisitos para uma mudança zonal efetiva	16
Recomendações para resiliência à interrupção zonal	16
Conclusão e recuperação do turno	17
Conclusão	18
Recursos	19
Histórico do documento	20
Glossário	21
#	21
A	22
B	25
C	27
D	30
E	34
F	36
G	38
H	39
eu	41
L	43

M	44
O	49
P	51
Q	54
R	55
S	58
T	62
U	63
V	64
W	64
Z	65
.....	lxvii

Projetando para alta disponibilidade e resiliência em aplicativos Amazon EKS

Haofei Feng, Frank Fan e Rus Kalakutskiy, da Amazon Web Services (AWS)

Outubro de 2025 ([histórico do documento](#))

Garantir alta disponibilidade (HA) e resiliência no design do aplicativo é crucial para alcançar objetivos de ponto de recuperação (RPO) e objetivos de tempo de recuperação (RTO) quase nulos. À medida que as organizações migram e modernizam cada vez mais seus aplicativos para ambientes Kubernetes, a demanda por soluções robustas e escaláveis continua aumentando. O Amazon Elastic Kubernetes Service (Amazon EKS) ajuda você a gerenciar com eficiência aplicativos em contêineres em grande escala.

Este guia aborda um conjunto de recomendações e melhores práticas amplamente reconhecidas para projetar e gerenciar aplicativos de microsserviços do Amazon EKS. Com base em ampla experiência e implantações no mundo real, esses insights oferecem orientação valiosa para arquitetos e desenvolvedores. Implemente essas recomendações para obter alto desempenho, confiabilidade e escalabilidade de seus aplicativos baseados em Kubernetes para obter operações robustas.

Considerações sobre design de alta disponibilidade e resiliência

O modelo de responsabilidade compartilhada se torna mais complexo com o Kubernetes. A disponibilidade e a resiliência do plano de controle do Amazon EKS são gerenciadas pela Amazon Web Services (AWS). Sua organização gerencia o plano de dados, o que pode afetar significativamente o desempenho e a disponibilidade de seus aplicativos de microsserviços.

Ao criar um aplicativo altamente disponível e resiliente no Amazon EKS, considere os seguintes componentes:

- O aplicativo de microsserviços: seus pods e contêineres
- O plano de dados da carga de trabalho: controlador de entrada, pod, componentes do sistema, como a [interface de rede de contêineres \(CNI\) da Amazon Virtual Private Cloud \(Amazon VPC\)](#), sidecars de service mesh e kube-proxy
- A camada de gerenciamento da carga de trabalho: controladores, controladores de admissão, mecanismos de política de rede e armazenamento persistente de dados para esses componentes
- O plano de controle do Kubernetes
- Infraestrutura: nós, rede e dispositivos de rede

Para as três primeiras considerações, que se referem aos componentes que são executados em um cluster Kubernetes, este guia aborda os seguintes tópicos:

- [Distribuindo cargas de trabalho entre nós e zonas de disponibilidade](#)
- [Protegendo cargas de trabalho críticas com um PDB](#)
- [Configurando sondas e verificações de saúde](#)
- [Configurando ganchos do ciclo de vida do contêiner](#)
- [Entendendo o despejo de lagoas durante interrupções zonais](#)

Distribua cargas de trabalho entre nós e zonas de disponibilidade

A distribuição de uma carga de trabalho em [domínios de falha](#), como zonas de disponibilidade e nós, melhora a disponibilidade dos componentes e diminui as chances de falha em aplicativos escaláveis

horizontalmente. As seções a seguir apresentam formas de distribuir cargas de trabalho entre nós e zonas de disponibilidade.

Use restrições de dispersão da topologia do pod

As [restrições de dispersão da topologia do pod do Kubernetes instruem](#) o programador do Kubernetes a distribuir pods que são gerenciados por ReplicaSet ou StatefulSet entre diferentes domínios de falha (zonas de disponibilidade, nós e tipos de hardware). Ao usar restrições de dispersão da topologia do pod, você pode fazer o seguinte:

- Distribua ou concentre os pods em diferentes domínios de falha, dependendo dos requisitos do aplicativo. Por exemplo, você pode distribuir pods para resiliência e concentrar pods para desempenho de rede.
- Combine condições diferentes, como distribuição entre zonas de disponibilidade e distribuição entre nós.
- Especifique a ação preferida se as condições não puderem ser atendidas:
 - Use `whenUnsatisfiable: DoNotSchedule` com uma combinação de `maxSkew` e `minDomains` para criar requisitos rígidos para o agendador.
 - Use `whenUnsatisfiable: ScheduleAnyway` para reduzir `maxSkew`.

Se uma zona de falha ficar indisponível, os pods dessa zona ficarão insalubres. O Kubernetes reprograma os pods enquanto segue a restrição de propagação, se possível.

O código a seguir mostra um exemplo do uso de restrições de distribuição da topologia de pod entre zonas de disponibilidade ou entre nós:

```
...
spec:
  selector:
    matchLabels:
      app: <your-app-label>
  replicas: 3
  template:
    metadata:
      labels: <your-app-label>
    spec:
      serviceAccountName: <ServiceAccountName>
...
  topologySpreadConstraints:
```

```
- labelSelector:
  matchLabels:
    app: <your-app-label>
  maxSkew: 1
  topologyKey: topology.kubernetes.io/zone # <---spread those pods evenly over
all availability zones
  whenUnsatisfiable: ScheduleAnyway
- labelSelector:
  matchLabels:
    app: <your-app-label>
  maxSkew: 1
  topologyKey: kubernetes.io/hostname # <---spread those pods evenly over all
nodes
  whenUnsatisfiable: ScheduleAnyway
```

Restrições de dispersão da topologia padrão em todo o cluster

Por padrão, o Kubernetes fornece um [conjunto de restrições de distribuição de topologia para distribuição de pods entre nós e zonas de disponibilidade](#):

```
defaultConstraints:
- maxSkew: 3
  topologyKey: "kubernetes.io/hostname"
  whenUnsatisfiable: ScheduleAnyway
- maxSkew: 5
  topologyKey: "topology.kubernetes.io/zone"
  whenUnsatisfiable: ScheduleAnyway
```

Note

Aplicativos que precisam de diferentes tipos de restrições de topologia podem substituir a política em nível de cluster.

As restrições padrão definem um valor `maxSkew`, o que não é útil para implantações com um pequeno número de pods. No momento, [não KubeSchedulerConfiguration pode ser alterado](#) no Amazon EKS. Se você precisar aplicar outros conjuntos de restrições de dispersão de topologia, considere usar o controlador de admissão mutante, como na seção abaixo. Você também pode controlar as restrições de dispersão da topologia padrão se você executar um agendador alternativo. No entanto, o gerenciamento de agendadores personalizados aumenta a complexidade e pode ter

implicações na resiliência do cluster e na HA. Por esses motivos, não recomendamos o uso de um programador alternativo somente para restrições de dispersão de topologia.

A política do Gatekeeper para restrições de dispersão de topologia

[Outra opção para impor restrições de dispersão de topologia é usar uma política do projeto Gatekeeper](#). As políticas do Gatekeeper são definidas no nível do aplicativo.

Os exemplos de código a seguir mostram o uso de uma Gatekeeper OPA política para implantação. Você pode modificar a política de acordo com suas necessidades. Por exemplo, aplique a política somente às implantações que tenham o rótulo `HA=true` ou escreva uma política semelhante usando um controlador de política diferente.

Este primeiro exemplo mostra o `ConstraintTemplate` uso `comk8stopologyspreadrequired_template.yml`:

```
apiVersion: templates.gatekeeper.sh/v1
kind: ConstraintTemplate
metadata:
  name: k8stopologyspreadrequired
spec:
  crd:
    spec:
      names:
        kind: K8sTopologySpreadRequired
      validation:
        openAPIV3Schema:
          type: object
          properties:
            message:
              type: string
  targets:
    - target: admission.k8s.gatekeeper.sh
      rego: |
        package k8stopologyspreadrequired

        get_message(parameters, _default) =3D msg {
          not parameters.message
          msg :=_default
        }

        get_message(parameters, _default) =3D msg {
```

```

    msg := parameters.message
  }

  violation[{"msg": msg}] {
    input.review.kind.kind = "Deployment"
    not input.review.object.spec.template.spec.topologySpreadConstraint
    def_msg : "Pod Topology Spread Constraints are required for Deployments"
    msg : get_message(input.parameters, def_msg)
  }
}

```

O código a seguir mostra o `k8stopologyspreadrequired_constraint.yml` manifesto constraints YAML:

```

apiVersion: constraints.gatekeeper.sh/v1beta1
kind: K8sTopologySpreadRequired
metadata:
  name: require-topologyspread-for-deployments
spec:
  match:
    kinds:
      - apiGroups: ["apps"]
        kinds: ["Deployment"]
    namespaces: ## Without theses two lines will apply to the whole cluster
      - "example"

```

Quando usar restrições de dispersão de topologia

Considere o uso de restrições de dispersão de topologia para os seguintes cenários:

- Qualquer aplicativo escalável horizontalmente (por exemplo, serviços web sem estado)
- Aplicativos com réplicas ativo-ativas ou ativas-passivas (por exemplo, bancos de dados NoSQL ou caches)
- Aplicativos com réplicas em espera (por exemplo, controladores)

Os componentes do sistema que podem ser usados para o cenário escalável horizontalmente, por exemplo, incluem o seguinte:

- [Cluster Autoscaler](#) e [Karpenter](#) (com e) `replicaCount > 1 leader-elect = true`
- [AWS Controlador de balanceador de carga](#)

- [CoreDNS](#)

Afinidade e antiafinidade do pod

Em alguns casos, é vantajoso garantir que não mais do que um pod de um tipo específico esteja sendo executado em um nó. Por exemplo, para evitar o agendamento de vários pods com muita rede no mesmo nó, você pode usar a regra de antiafinidade com o rótulo `ingress-network-heavy`. Ao usar `anti-affinity`, você também pode usar uma combinação do seguinte:

- Manchas em nós otimizados para rede
- Tolerâncias correspondentes em cápsulas de rede pesada
- Afinidade de nós ou seletor de nós para garantir que pods com muita rede usem instâncias otimizadas para rede

Os pods com muita rede são usados como exemplo. Você pode ter requisitos diferentes, como GPU, memória ou armazenamento local. Para ver outros exemplos de uso e opções de configuração, consulte a documentação do [Kubernetes](#).

Rebalancear cápsulas

Esta seção discute duas abordagens para rebalancear pods em um cluster Kubernetes. O primeiro usa o `Descheduler` for Kubernetes. O `Descheduler` ajuda a manter a distribuição dos pods aplicando estratégias para remover os pods que violam as restrições de dispersão da topologia ou as regras de antiafinidade. A segunda abordagem usa o recurso de consolidação e empacotamento do `Karpenter`. A consolidação avalia e otimiza continuamente o uso de recursos consolidando cargas de trabalho em menos nós compactados com mais eficiência.

Recomendamos usar o `Descheduler` se você não estiver usando o `Karpenter`. Se você estiver usando o `Karpenter` e o `Autoescalador de Cluster` juntos, poderá usar o `Desagendador` com o `Autoescalador de Cluster` para grupos de nós.

Desagendador para nós sem grupos

Não há garantia de que as restrições de topologia permaneçam satisfeitas quando os pods são removidos. Por exemplo, reduzir uma implantação pode resultar em uma distribuição desequilibrada de pods. No entanto, como o Kubernetes usa restrições de dispersão da topologia do pod somente no estágio de agendamento, os pods ficam desbalanceados em todo o domínio da falha.

Para manter uma distribuição equilibrada de pods nesses cenários, você pode usar o [Descheduler](#) for Kubernetes. O Descheduler é uma ferramenta útil para vários propósitos, como impor a idade máxima do pod ou o tempo de vida (TTL) ou para melhorar o uso da infraestrutura. No contexto de resiliência e alta disponibilidade (HA), considere as seguintes estratégias do Descheduler:

- [RemovePodsViolatingTopologySpreadConstraint](#)
- [RemovePodsViolatingInterPodAntiAffinity](#)
- [RemoveDuplicates](#)

Recurso de consolidação e empacotamento de caixas do Karpenter

Para cargas de trabalho que usam o Karpenter, você pode usar a funcionalidade de consolidação e empacotamento para otimizar a utilização de recursos e reduzir custos em clusters Kubernetes. O Karpenter avalia continuamente o posicionamento dos pods e a utilização dos nós e tenta consolidar as cargas de trabalho em menos nós compactados com mais eficiência, sempre que possível. Esse processo envolve analisar os requisitos de recursos, considerar restrições como regras de afinidade de pods e potencialmente mover pods entre os nós para melhorar a eficiência geral do cluster. O código a seguir fornece um exemplo:

```
apiVersion: karpenter.sh/v1beta1
kind: NodePool
metadata:
  name: default
spec:
  disruption:
    consolidationPolicy: WhenUnderutilized
    expireAfter: 720h
```

Para `consolidationPolicy`, você pode usar `WhenUnderutilized` ou `WhenEmpty`:

- Quando `consolidationPolicy` está definido como `WhenUnderutilized`, o Karpenter considera todos os nós para consolidação. Quando o Karpenter descobre um nó vazio ou subutilizado, o Karpenter tenta remover ou substituir o nó para reduzir o custo.
- Quando `consolidationPolicy` está definido como `WhenEmpty`, o Karpenter considera para consolidação somente nós que não contêm pods de carga de trabalho.

As decisões de consolidação do Karpenter não se baseiam apenas nas porcentagens de utilização da CPU ou da memória que você pode ver nas ferramentas de monitoramento. Em vez disso, o

Karpenter usa um algoritmo mais complexo baseado em solicitações de recursos do pod e possíveis otimizações de custos. Para obter mais informações, consulte a documentação do [Karpenter](#).

Proteja cargas de trabalho críticas com um PDB

Um orçamento de interrupção de pod (PDB) é um recurso essencial para manter a alta disponibilidade dos aplicativos em um cluster. O PDB especifica um tamanho alvo, que é a disponibilidade mínima para um tipo específico de pod. Isso significa que um número mínimo de réplicas de um determinado tipo de pod deve estar em execução a qualquer momento. Se o número de réplicas em execução ficar abaixo do tamanho desejado, o Kubernetes evitará mais interrupções nas réplicas restantes até que o tamanho desejado seja atingido. PDBs ajudam a garantir que as cargas de trabalho não sejam afetadas por esses eventos e possam continuar funcionando sem interrupções. Quando ocorre uma interrupção, o Kubernetes tenta expulsar os pods dos nós afetados com facilidade, mantendo o número de réplicas especificado no PDB.

Você pode usar um PDB para declarar o `maxUnavailable` número `minAvailable` e o número de réplicas. Por exemplo, se você quiser que pelo menos três cópias do seu aplicativo estejam disponíveis, crie um PDB semelhante ao exemplo a seguir:

```
apiVersion: policy/v1beta1
kind: PodDisruptionBudget
metadata:
  name: my-svc-pdb
spec:
  minAvailable: 3
  selector:
    matchLabels:
      app: my-svc
```

A configuração PDBs correta de seus aplicativos ajuda a minimizar a interrupção durante eventos planejados ou não planejados. Você pode usar a regra de antiafinidade para programar os pods de uma implantação em nós diferentes e evitar atrasos no PDB durante as atualizações dos nós.

Configurar sondas e verificações de integridade do balanceador de carga

O Kubernetes fornece várias maneiras de realizar verificações de integridade do aplicativo, além das verificações de integridade do balanceador de carga. Você pode executar os seguintes testes

integrados do Kubernetes junto com a verificação de integridade do balanceador de carga como um comando no contexto do pod ou como um teste para o kubelet ou o HTTP/TCP endereço IP do host.

As sondas de vivacidade e prontidão devem ser diferentes e independentes (ou, pelo menos, com valores de tempo limite diferentes). Se um aplicativo tiver um problema temporário, a sondagem de prontidão marcará o pod como não pronto até que o problema seja resolvido. Se as configurações da sonda liveness não estiverem corretas, a sonda liveness poderá encerrar o pod.

Sonda de inicialização

Use sondas de inicialização para proteger aplicativos com ciclos de inicialização longos. Até que a sondagem de inicialização seja bem-sucedida, as outras sondas serão desativadas.

Você pode definir o tempo máximo que o Kubernetes deve esperar pela inicialização do aplicativo. Se, após o tempo máximo configurado, o pod ainda falhar nos testes de inicialização, o aplicativo será encerrado e um novo pod será criado.

Use sondas de inicialização quando o tempo de inicialização de um aplicativo for imprevisível. Se você sabe que seu aplicativo precisa de 10 segundos para ser iniciado, use uma sonda de vivacidade ou uma sonda de prontidão. `initialDelaySeconds`

Sonda de vivacidade

Use sondas de atividade para detectar problemas no aplicativo ou se o processo está sendo executado sem problemas. Uma sonda de atividade pode detectar condições de impasse em que o processo continua em execução, mas o aplicativo deixa de responder. Ao usar uma sonda de vivacidade, faça o seguinte:

- Use `initialDelaySeconds` para atrasar a primeira sonda.
- Não defina a mesma especificação para sondas de vivacidade e prontidão.
- Não configure uma sonda de atividade para depender de um fator externo ao seu pod (por exemplo, um banco de dados).
- Defina a sonda de vivacidade para um específico. `terminationGracePeriodSeconds` Para obter mais informações, consulte [a documentação do Kubernetes](#).

Sonda de prontidão

Use uma sonda de prontidão para detectar o seguinte:

- Se o aplicativo está pronto para aceitar tráfego
- Disponibilidade parcial, em que o aplicativo pode estar temporariamente indisponível, mas espera-se que volte a funcionar após a conclusão de uma determinada operação

Os testes de prontidão ajudam a garantir que a configuração e as dependências do aplicativo sejam executadas sem problemas ou erros, para que o aplicativo possa atender ao tráfego. No entanto, uma sonda de prontidão mal configurada pode causar uma interrupção em vez de evitá-la. Sondagens de prontidão que dependem de fatores externos, como conectividade com um banco de dados, podem fazer com que todos os pods falhem na sondagem. Essas falhas podem resultar em uma interrupção e levar a uma falha em cascata de um serviço de back-end para outros serviços que usaram os pods com falha.

Verificações de integridade do recurso de entrada e do balanceador de carga

O Application Load Balancer e o Kubernetes `ingress` fornecem recursos de verificação de integridade. Para as verificações de integridade do Application Load Balancer, especifique as portas e o caminho de destino.

Note

Para o Kubernetes `ingress`, haverá uma latência de cancelamento de registro. O padrão para o Application Load Balancer é 300 segundos. Considere configurar seu recurso de entrada ou a verificação de integridade do balanceador de carga usando os mesmos valores que você usou para sua análise de prontidão.

O NGINX também fornece uma verificação de integridade. Para obter mais informações, consulte a documentação do [NGINX](#).

Os gateways de entrada e saída do Istio não têm um mecanismo de verificação de integridade comparável à verificação de integridade HTTP do NGINX. No entanto, você pode obter uma funcionalidade semelhante usando o [disjuntor ou a detecção de DestinationRule valores discrepantes do Istio](#).

Para obter mais informações, consulte [Disponibilidade e ciclo de vida do pod](#) no Guia de melhores práticas do Amazon EKS.

Configurar ganchos do ciclo de vida do contêiner

Durante um desligamento normal do contêiner, seu aplicativo deve responder a um SIGTERM sinal iniciando o desligamento para que os clientes não tenham nenhum tempo de inatividade. Seu aplicativo deve executar procedimentos de limpeza como os seguintes:

- Salvando dados
- Fechando descritores de arquivo
- Fechando conexões de banco de dados
- Concluindo solicitações durante o voo com elegância
- Saindo em tempo hábil para atender à solicitação de encerramento do pod

Defina um período de carência que seja longo o suficiente para que a limpeza termine. Para saber como responder ao SIGTERM sinal, consulte a documentação da linguagem de programação que você usa para seu aplicativo.

Os [ganchos do ciclo de vida do contêiner](#) permitem que os contêineres estejam cientes dos eventos em seu ciclo de vida de gerenciamento. Os contêineres podem executar código implementado em um manipulador quando o gancho de ciclo de vida correspondente é executado. Os ganchos do ciclo de vida do contêiner fornecem uma solução alternativa para a natureza assíncrona do Kubernetes e da nuvem. Essa abordagem pode evitar a perda de conexões que são encaminhadas para o pod de encerramento antes do recurso de entrada e iptables são atualizadas para não enviar novo tráfego para o pod.

Ciclo de vida do contêiner, Endpoint, e EndpointSlice fazem parte de diferentes APIs. É importante orquestrá-los. No entanto, quando um pod está sendo encerrado, a API Kubernetes notifica simultaneamente o kubelet (para o ciclo de vida do contêiner) e o controlador. Para obter mais informações, incluindo um diagrama, consulte [Lidar com elegância com as solicitações do cliente no Guia de melhores práticas do Amazon EKS](#).

Quando kubelet envia SIGTERM para o pod, o EndpointSlice controlador está encerrando o EndpointSlice objeto. Esse encerramento notifica os servidores da API Kubernetes para notificarem cada nó a ser kube-proxy atualizado. Embora essas ações ocorram ao mesmo tempo, não há dependências ou sequências entre elas. Há uma grande chance de o contêiner receber o SIGKILL sinal muito antes de cada nó atualizar as iptables regras locais. Nesse caso, os cenários possíveis incluem o seguinte:

- Se sua solicitação cancelar imediatamente e sem rodeios as solicitações e conexões em voo após o recebimento SIGTERM, os clientes verão erros. 500
- Se seu aplicativo garantir que todas as solicitações e conexões em andamento sejam processadas completamente após o recebimento SIGTERM, durante o período de carência, as novas solicitações do cliente ainda serão enviadas ao contêiner do aplicativo, pois iptables as regras talvez ainda não tenham sido atualizadas. Até que o procedimento de limpeza feche o soquete do servidor no contêiner, essas novas solicitações resultarão em novas conexões. Quando o período de carência termina, as novas conexões que foram estabelecidas após o SIGTERM envio são descartadas incondicionalmente.

Para abordar os cenários anteriores, você pode implementar a integração no aplicativo ou o gancho do PreStop ciclo de vida. Para obter mais informações, incluindo um diagrama, consulte [Encerrar aplicativos normalmente no Guia de melhores práticas do Amazon EKS](#).

Note

Independentemente de o aplicativo ser encerrado normalmente ou do resultado de um problema, os contêineres do preStop aplicativo acabam sendo encerrados no final do período de carência. SIGKILL

Use o preStop gancho com um sleep comando para atrasar o envio SIGTERM. Isso ajudará a continuar aceitando as novas conexões enquanto o objeto de entrada as encaminha para o pod. Teste o valor temporal do sleep comando para garantir que qualquer latência do Kubernetes e de outras dependências do aplicativo seja levada em consideração, conforme mostrado no exemplo a seguir:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  containers:
    - name: nginx
      lifecycle:
        # This "sleep" preStop hook delays the Pod shutdown until
        # after the Ingress Controller removes the matching Endpoint or EndpointSlice
        preStop:
          exec:
```

```
command:  
- /bin/sleep  
- "20"  
# This period should be turned to Ingress/Service Mesh update latency
```

Para obter mais informações, consulte [Container hooks](#) na documentação do Kubernetes e [desligar aplicativos normalmente no Amazon EKS Best Practices](#) Guide.

Entenda o despejo de lagoas durante interrupções zonais

Quando ocorre uma interrupção total da zona de disponibilidade, ou seja, quando todos os nós dessa zona de disponibilidade perdem a conectividade com o plano de controle do Kubernetes, o [controlador do ciclo de vida do nó](#) no Kubernetes detecta a situação e expulsa os pods da zona afetada. Os pods em nós inacessíveis são marcados como `Terminating` e novos pods são programados em nós íntegros nas zonas de disponibilidade disponíveis. Durante esse período, os nós afetados exibem um `NotReady` status, o agendador impede que novos pods sejam colocados nesses nós e o `EndpointSlice` controlador remove os endpoints associados à Zona de Disponibilidade prejudicada do roteamento do serviço até que a conectividade seja restaurada.

Para cenários que envolvem falhas parciais de nós em uma zona, em que somente um subconjunto de nós se torna inacessível, o controlador do ciclo de vida do nó aplica diferentes comportamentos de despejo. Se a interrupção persistir além do período de tolerância configurado (por padrão, cinco minutos), os pods nos nós desconectados serão marcados como `Terminating` e os novos pods serão programados nos nós íntegros nas zonas de disponibilidade disponíveis.

Implementando a mudança de zona do Amazon EKS para melhorar a resiliência

A [mudança de zona do Amazon EKS, que se integra ao Amazon Application Recovery Controller \(ARC\)](#), fornece um mecanismo para gerenciar proativamente o tráfego durante deficiências na zona de disponibilidade. Esse recurso permite o redirecionamento temporário do tráfego de rede de uma zona de disponibilidade insalubre para zonas íntegras dentro da mesma, a fim de minimizar Região da AWS a interrupção do serviço.

Entendendo o mecanismo de mudança zonal

A mudança de zona do Amazon EKS aborda o tráfego leste-oeste (comunicação entre pods dentro do cluster). Quando a mudança de zona é configurada com `Application Load Balancers` ou `Network Load Balancers`, ela também oferece suporte ao roteamento de tráfego de entrada. O

mecanismo opera coordenando vários componentes do Kubernetes e do plano de AWS controle para redirecionar o tráfego com segurança sem interromper as cargas de trabalho em execução. Durante uma mudança de zona ativa, o Amazon EKS executa automaticamente as seguintes ações coordenadas:

- **Isolamento de nós:** todos os nós na Zona de Disponibilidade comprometida são isolados. Isso impede que o programador do Kubernetes coloque novos pods nos nós enquanto mantém as cargas de trabalho existentes.
- **Suspensão do rebalanceamento da zona de disponibilidade:** para grupos de nós gerenciados, as operações de rebalanceamento da zona de disponibilidade são suspensas e os grupos do Auto Scaling são atualizados para iniciar novos nós do plano de dados exclusivamente em zonas de disponibilidade saudáveis. Isso garante que a nova capacidade não seja provisionada na zona prejudicada.
- **Remoção do endpoint:** o EndpointSlice controlador remove os endpoints do pod na zona de disponibilidade prejudicada de todos os itens relevantes. EndpointSlices Isso garante que os mecanismos de descoberta de serviços e balanceamento de carga direcionem o tráfego somente para pods que estão sendo executados em zonas de disponibilidade saudáveis.
- **Preservação da carga de trabalho:** o Amazon EKS se abstém de encerrar nós ou remover pods na zona de disponibilidade afetada. Ele mantém a capacidade total na zona prejudicada para que, quando a mudança zonal expirar ou for cancelada, o tráfego possa retornar com segurança sem exigir operações adicionais de escalonamento.

Métodos de ativação por mudança zonal

Você pode escolher entre duas abordagens para iniciar mudanças zonais, dependendo do seu modelo operacional:

- O [deslocamento zonal manual](#) fornece controle orientado pelo operador quando problemas específicos da zona de disponibilidade são detectados por meio de monitoramento, alertas ou relatórios de clientes. Esse método requer ação explícita por meio do console ARC, AWS Command Line Interface (AWS CLI) ou mudança APIs zonal, em que os operadores especificam a zona de disponibilidade prejudicada e definem um prazo de expiração para o turno. Os turnos manuais são apropriados quando as equipes têm recursos dedicados de monitoramento e plantão e preferem manter o controle direto sobre as decisões de gerenciamento de tráfego.
- O [deslocamento automático zonal](#) AWS autoriza o início automático de turnos quando o ARC detecta possíveis falhas ou deficiências na zona de disponibilidade com base em telemetria

interna e sinais de saúde em vários, Serviços da AWS incluindo métricas de rede, Amazon Elastic Compute Cloud (Amazon EC2) e Elastic Load Balancing. AWS encerra automaticamente um deslocamento automático quando os indicadores mostram que o problema foi resolvido. Se você deseja a postura de maior disponibilidade com o mínimo de intervenção manual, recomendamos essa abordagem, pois ela permite uma resposta em menos de um minuto às deficiências detectadas na Zona de Disponibilidade.

Pré-requisitos para uma mudança zonal efetiva

Para que a mudança zonal proteja com sucesso os aplicativos durante deficiências na Zona de Disponibilidade, você deve arquitetar seus clusters para resiliência Multi-AZ antes de ativar o recurso de mudança zonal:

- Distribuição de nós Multi-AZ: provisione nós de trabalho em pelo menos três zonas de disponibilidade para garantir redundância suficiente quando uma zona ficar indisponível.
- Planejamento de capacidade: pré-provisione capacidade computacional suficiente em zonas de disponibilidade saudáveis para acomodar toda a carga de trabalho quando uma zona de disponibilidade é removida do serviço, pois as operações de escalabilidade durante uma interrupção ativa podem encontrar capacidade insuficiente.
- Distribuição e pré-escalonamento de pods: [implante várias réplicas de cada aplicativo em todas as zonas de disponibilidade e pré-escale componentes críticos do sistema, como o CoreDNS, em todas as zonas](#). Isso ajuda a garantir que a capacidade suficiente permaneça após a mudança de uma zona.

Recomendações para resiliência à interrupção zonal

- Habilite a mudança zonal na criação do cluster: Para novos clusters EKS, habilite a integração do deslocamento zonal com o ARC durante o provisionamento inicial por meio do console Amazon EKS ou de ferramentas de infraestrutura como código (IaC) AWS CLI, como. AWS CloudFormation Os [clusters do EKS Auto Mode](#) criados com configuração rápida têm a mudança de zona ativada por padrão.
- Selecione o método de ativação apropriado: escolha o deslocamento automático zonal para ambientes de produção que exigem disponibilidade máxima com resposta automatizada, especialmente para aplicativos voltados para o cliente, nos quais minutos de inatividade durante um comprometimento da zona de disponibilidade podem ter um impacto significativo nos negócios. Use a mudança de zona manual para ambientes em que as equipes de operações preferem

fornecer aprovação explícita antes das mudanças de tráfego ou onde os testes e a validação de aplicativos ainda estão em andamento.

- Teste a resiliência antes da implantação na produção: valide o comportamento do cluster sob perda de Single-AZ iniciando manualmente as mudanças zonais de teste ou habilitando a prática de mudança automática zonal para verificar se os aplicativos mantêm a disponibilidade, o desempenho permanece aceitável e a capacidade é suficiente ao operar com uma contagem reduzida de zonas de disponibilidade. É altamente recomendável esse teste para que você possa identificar lacunas na configuração antes que ocorram deficiências reais na Zona de Disponibilidade.
- Coordene com a configuração do balanceador de carga: para aplicativos que recebem tráfego externo, ative a mudança zonal ARC nos balanceadores de carga de aplicativos e balanceadores de carga de rede associados para garantir que tanto o tráfego de entrada quanto o tráfego no cluster leste-oeste mudem juntos durante deficiências na zona de disponibilidade. Essa coordenação evita cenários em que solicitações externas cheguem a pods saudáveis, mas esses pods não conseguem se comunicar com dependências na zona afastada.
- Monitore as operações de turno: depois de ativar o turno zonal, configure o monitoramento e o alerta para eventos de turno, incluindo ativações de turnos automáticos, iniciações manuais de turnos e vencimentos de turnos, para manter a visibilidade operacional das ações de gerenciamento de tráfego e seu impacto no comportamento do aplicativo.

Conclusão e recuperação do turno

Quando um deslocamento zonal expira com base em sua duração configurada ou é cancelado manualmente após a resolução do comprometimento da Zona de Disponibilidade, o EndpointSlice controlador atualiza automaticamente tudo EndpointSlices para reincorporar os endpoints na Zona de Disponibilidade restaurada. O tráfego retorna gradualmente à zona anteriormente afetada à medida que os clientes atualizam as informações do endpoint e estabelecem novas conexões. Isso permite a utilização total da capacidade do cluster sem exigir intervenção manual ou reprogramação do pod.

Conclusão

Ao projetar sua arquitetura para alta disponibilidade e resiliência de aplicativos, considere os seguintes componentes:

- O aplicativo de microsserviços (seus pods e contêineres)
- O plano de dados da carga de trabalho (Ingress Controller, pod, componentes do sistema, como Amazon [VPC CNI](#), service mesh sidecars e kube-proxy)
- A camada de gerenciamento da carga de trabalho (controladores, controladores de admissão, mecanismos de políticas de rede e armazenamento persistente de dados para esses componentes)
- O plano de controle do Kubernetes
- Infraestrutura (nós, rede e dispositivos de rede)

Para abordar essas considerações de componentes, use as seguintes estratégias principais:

- Para ajudar a garantir alta disponibilidade e tolerância a falhas, distribua as cargas de trabalho entre nós e zonas de disponibilidade.
- Para proteger cargas de trabalho críticas, mantenha a estabilidade do aplicativo durante interrupções usando orçamentos de interrupção do pod (). PDBs
- Para ajudar a garantir que os pods estejam funcionando e veiculando o tráfego corretamente, configure testes de inicialização, testes de atividade, testes de prontidão e verificações de integridade do balanceador de carga.
- Para gerenciar as transições de estado do contêiner com eficiência, configure os ganchos do ciclo de vida do contêiner.
- Para fornecer controle sobre o processo de despejo durante falhas ou manutenção do nó, configure o tempo de despejo do pod.

Ao implementar essas práticas, você pode melhorar significativamente a confiabilidade e a resiliência dos aplicativos executados no Amazon EKS, garantindo desempenho robusto e alta disponibilidade.

Recursos

- [Restrições de dispersão da topologia do pod Kubernetes](#) (documentação do Kubernetes)
- [Karpenter FAQs](#) (documentação do Karpenter)
- [Desagendador](#) para Kubernetes (repositório) GitHub
- Guia de melhores práticas do Amazon EKS [para disponibilidade e ciclo de vida do pod](#)
- [Encerre aplicativos normalmente | Guia de melhores práticas do Amazon EKS](#)
- [\[EKS\] \[solicitação\]: capacidade de configuração pod-eviction-timeout e soluções alternativas](#) (repositório Containers Roadmap)

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
Atualização	Revisou a seção sobre despejos de cápsulas durante interrupções zonais .	29 de outubro de 2025
Atualização	A seção Usar restrições de dispersão da topologia do pod foi revisada.	27 de janeiro de 2025
Publicação inicial	—	23 de outubro de 2024

AWS Glossário de orientação prescritiva

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

Números

7 Rs

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

- Refatorar/rearquitetar: mova uma aplicação e modifique sua arquitetura aproveitando ao máximo os recursos nativos de nuvem para melhorar a agilidade, a performance e a escalabilidade. Isso normalmente envolve a portabilidade do sistema operacional e do banco de dados. Exemplo: migrar seu banco de dados Oracle on-premises para o Amazon Aurora Edição Compatível com 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: migrar seu banco de dados Oracle on-premises para o Amazon Relational Database Service (Amazon RDS) para Oracle na Nuvem AWS.
- Recomprar (drop and shop): mude para um produto diferente, normalmente migrando de uma licença tradicional para um modelo SaaS. Exemplo: migrar seu sistema de gerenciamento de relacionamento com o cliente (CRM) para o 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: migrar seu banco de dados Oracle on-premises para o Oracle em uma instância do EC2 na 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 on-premises para um serviço de nuvem para a mesma plataforma. Exemplo: migrar um Microsoft Hyper-V aplicativo para o AWS
- Reter (revisitar): mantenha as aplicações em seu ambiente de origem. Isso pode incluir aplicações que exigem grande refatoração, e você deseja adiar esse trabalho para um

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

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

A

ABAC

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

serviços abstraídos

Veja [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 em que 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.

AGGREGATE FUNCTION

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 [inteligência artificial](#).

AIOps

Veja [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 aplicações

Uma abordagem de segurança que permite o uso somente de aplicações aprovadas 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 malicioso

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

BCP

Veja [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 da aplicação em um ambiente (azul) e a nova versão da aplicação no outro ambiente (verde). Essa estratégia ajuda você a reverter rapidamente com o mínimo de impacto.

bot

Uma aplicação de software que executa tarefas automatizadas na internet e simula a atividade ou interação humana. Alguns bots são úteis ou benéficos, como crawlers da web que indexam informações na internet. Outros bots, conhecidos como bots maliciosos, têm como objetivo causar interrupção ou danos a indivíduos ou organizações.

botnet

Redes de [bots](#) infectadas por [malware](#) e sob o controle de uma única parte, conhecidas como bot herder ou operador de bots. Os botnets 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 de emergência

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 [Implement break-glass procedures](#) nas orientações do AWS Well-Architected.

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

Veja [AWS Cloud Adoption Framework](#).

implantação canário

O lançamento lento e incremental de uma versão para usuários finais. Quando estiver confiante, você implanta a nova versão e substitui a versão atual por completo.

CCoE

Veja [Centro de Excelência da Nuvem](#).

CDC

Veja [captura de dados de alteração](#).

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 [integração e 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 corporativa.

computação em nuvem

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

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 ao migrar para a 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

Veja [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 o GitHub ou o 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 de [IA](#) que usa machine learning para analisar e extrair informações de formatos visuais, como vídeos e imagens digitais. Por exemplo, a Amazon SageMaker AI fornece algoritmos de processamento de imagem para CV.

desvio de configuração

Em uma workload, uma alteração de configuração em relação ao estado esperado. Isso pode fazer com que a workload 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

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

integração contínua e entrega contínua (CI/CD)

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

CV

Veja [visão computacional](#).

D

dados em repouso

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

classificação de dados

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

desvio de dados

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

dados em trânsito

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

data mesh

Um framework de arquitetura 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 compatível com business intelligence, como analytics. 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

Veja [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 [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 normalmente são 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 [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 [recuperação de desastres](#).

Deteção da oscilação

Rastreamento de desvios de uma configuração de linha de base. 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 [mapeamento do fluxo de valor de desenvolvimento](#).

E

EDA

Veja [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 com a [computação em nuvem](#), a computação de borda 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 EDI \(Intercâmbio eletrônico de dados\)?](#).

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 [endpoint de 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 empresariais (ERP)

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

criptografia envelopada

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

ambiente

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

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

epic

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

ERP

Veja [planejamento de recursos empresariais](#).

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](#). Ela armazena dados quantitativos sobre as operações comerciais. Normalmente, uma tabela de fatos contém dois tipos de colunas: as que contêm medidas e as que contêm uma chave externa para uma tabela de dimensões.

Antecipar-se à falha

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

delimitação de isolamento contra 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 [AWS Fault Isolation Boundaries](#).

ramificação de recursos

Veja [ramificação](#).

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.

prompt few shot

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 em contexto, em que os modelos aprendem com exemplos (shots) incorporados aos prompts. Prompts few-shot podem ser eficazes para tarefas que exigem formatação, raciocínio ou conhecimento de domínio específicos. Veja também [prompts zero-shot](#).

FGAC

Veja [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 via [captura de dados de alteração](#) para migrar os 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 [modelo de base](#).

modelo de base (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 de base?](#).

G

IA generativa

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

bloqueio geográfico

Veja [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 trunk-based](#) é a abordagem moderna e preferencial.

golden image

Um snapshot de um sistema ou software usado como modelo para implantar novas instâncias desse sistema ou software. Por exemplo, na manufatura, uma golden image 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 CSPM, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector e verificações personalizadas AWS Lambda .

H

HA

Veja [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 hold-out

Uma parte dos dados históricos rotulados que são retidos de um conjunto de dados usado para treinar um modelo de [machine learning](#). Você pode usar dados de hold-out para avaliar a performance do modelo comparando as previsões do modelo com os dados de retenção.

migração homogênea de bancos de dados

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

dados quentes

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

hotfix

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

período de hipercuidados

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

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

eu

laC

Veja [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 [Internet das Coisas Industrial](#).

infraestrutura imutável

Um modelo que implanta uma nova infraestrutura para workloads de produção em vez de atualizar, aplicar patches ou modificar a infraestrutura existente. Infraestruturas imutáveis são inerentemente mais consistentes, confiáveis e preditivas do que [infraestruturas mutáveis](#). Para obter mais informações, consulte a prática recomendada [Implantar usando infraestrutura imutável](#) no AWS Well-Architected 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 manufatura por meio de avanços em conectividade, dados em tempo real, automação, analytics 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

Veja [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

Veja [biblioteca de informações de TI](#).

ITSM

Veja [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](#).

grande modelo de linguagem (LLM)

Um modelo de [IA](#) de aprendizado profundo pré-treinado em uma grande quantidade de dados. Um LLM pode realizar várias tarefas, como responder a 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 rótulo](#).

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 [grande modelo de linguagem](#).

ambientes inferiores

Veja [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 [ramificação](#).

Malware

Software projetado para comprometer a segurança ou a privacidade do computador. O malware pode interromper os sistemas do computador, vaziar informações sensíveis ou obter acesso não autorizado. Exemplos de malware incluem vírus, worms, ransomware, cavalos de Troia, 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 abstraídos.

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

Veja [Programa de Aceleração da Migração](#).

mecanismo

Um processo completo em que 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 [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 o Amazon EC2 AWS com o 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 a 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 workload para a Nuvem AWS. Para obter mais informações, veja a entrada [7 Rs](#) neste glossário e consulte [Mobilize sua organização para acelerar migrações em grande escala](#).

ML

Veja [machine learning](#).

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 [Strategy for modernizing applications in the 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 [Evaluating modernization readiness for applications in the 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](#).

MPA

Veja [Avaliação do Portfólio para Migração](#).

MQTT

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

OAI

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

OCM

Veja [gerenciamento de alterações 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 [integração de operações](#).

Ola

Veja [acordo de 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

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

Open Process Communications - Unified Architecture (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 práticas recomendadas 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 AWS Well-Architected Framework.

tecnologia operacional (TO)

Sistemas de hardware e software que trabalham com o ambiente físico para controlar operações, equipamentos e infraestrutura industriais. Na manufatura, a integração dos sistemas de tecnologia da informação (TI) e tecnologia operacional (TO) 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 [análise de prontidão operacional](#).

OT

Veja [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 [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

Veja [controlador lógico programável](#).

PLM

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

política

Um objeto que pode definir permissões (veja [política baseada em identidade](#)), especificar condições de acesso (veja [política baseada em recurso](#)) ou definir as permissões máximas para todas as contas em uma organização no AWS Organizations (veja [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.

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 cláusula `WHERE`.

pushdown de predicados

Uma técnica de otimização de consultas 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 a performance 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 em engenharia de sistemas 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](#) desenvolvido para evitar a implantação de recursos não conformes. Esses controles verificam 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 a concepção, o desenvolvimento e o lançamento, passando pelo crescimento e maturidade, até o declínio e a remoção.

ambiente de produção

Veja [ambiente](#).

controlador lógico programável (PLC)

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

encadeamento de prompts

Uso da 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 em que outros microsserviços possam assinar. 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, aprovador, consultado, informado \(RACI\)](#).

RAG

Veja [geração aumentada via 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, aprovador, consultado, informado \(RACI\)](#).

RCAC

Veja [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.

Redefinir arquitetura

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 informações, consulte [Specify which Regiões da AWS your account can use](#).

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 uma aplicação de resistir ou se recuperar de interrupções. [Alta disponibilidade](#) e [recuperação de desastres](#) são considerações comuns ao planejar a resiliência na Nuvem AWS. Para obter mais informações, consulte [Nuvem AWS Resilience](#).

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](#).

Retirada

Veja [7 Rs](#).

Geração Aumentada de Recuperação (RAG)

Uma tecnologia de [IA generativa](#) em que 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 \(geração aumentada via recuperação\)?](#).

alternância

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

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 [objetivo de ponto de recuperação](#).

RTO

Veja [objetivo de 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 no Console de gerenciamento da AWS 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 [política de controle de serviço](#).

secret

Em AWS Secrets Manager, informações confidenciais ou restritas, como uma senha ou credenciais de usuário, que você armazena de forma criptografada. 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 [What's in a Secrets Manager secret?](#) na documentação do Secrets Manager.

segurança desde a concepção

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

controle de segurança

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

hardening 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 aplicação de patches em uma instância do Amazon EC2 ou a alternância 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 avaliação de um aspecto de performance de um serviço, como taxa de erro, disponibilidade ou throughput.

objetivo de nível de serviço (SLO)

Uma métrica alvo que representa a integridade de um serviço, conforme avaliado 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 [sistema de gerenciamento de eventos e informações de segurança](#).

ponto único de falha (SPOF)

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

SLA

Veja [acordo de serviço](#).

SLI

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

SLO

Veja [objetivo de 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 [Phased approach to modernizing applications in the Nuvem AWS](#).

SPOF

Veja [ponto único de falha](#).

esquema em estrela

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 ser usada 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 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 a performance. Você pode usar o [Amazon CloudWatch Synthetics](#) para criar esses testes.

prompt do sistema

Uma técnica para fornecer contexto, instruções ou orientações a um [LLM](#) a fim de direcionar seu comportamento. Os prompts do sistema ajudam a definir o contexto e a 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 da . 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 [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 [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 backend.

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.

WORM

Veja [gravação única e várias leituras](#).

WQF

Veja [AWS Workload Qualification Framework](#).

gravação única e várias leituras (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, normalmente malware, que tira proveito de uma [vulnerabilidade zero-day](#).

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.

prompt zero shot

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

eficácia dos prompts zero-shot depende da complexidade da tarefa e da qualidade do prompt.

Veja também [prompts few-shot](#).

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.