Estrutura do ciclo de vida de resiliência

AWS Orientação prescritiva

AWS Orientação prescritiva: Estrutura do ciclo de vida de resiliência

Table of Contents

Estrutura do ciclo de vida da resiliência: uma abordagem contínua para a melhoria da resiliência

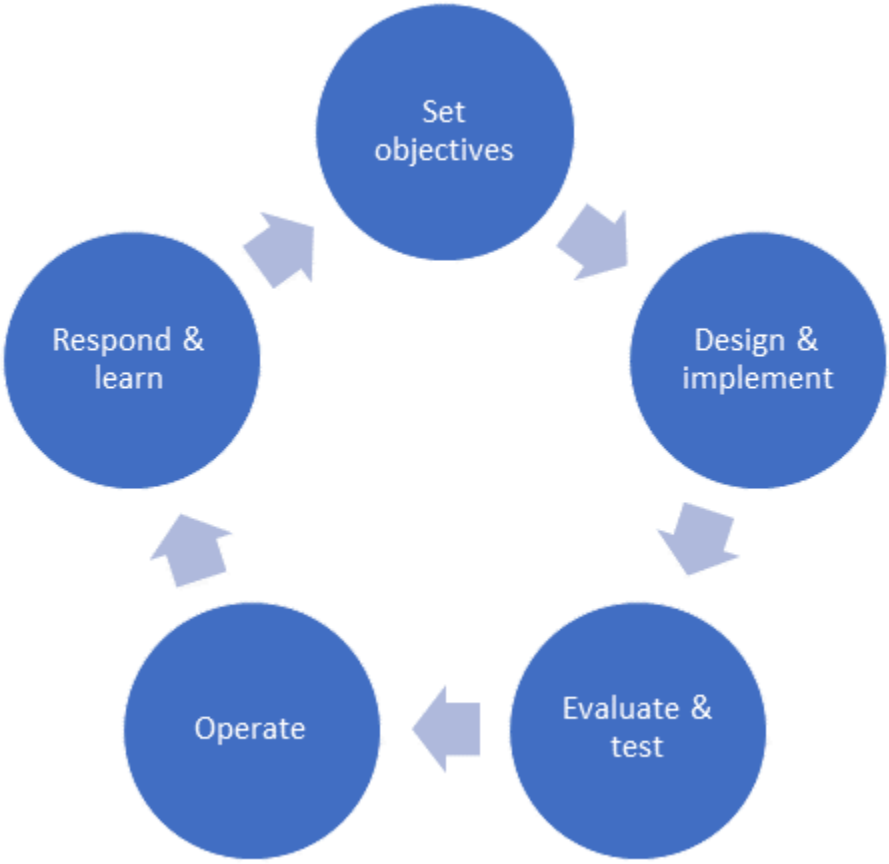
Amazon Web Services (colaboradores)

Outubro de 2023 (histórico do documento)

Atualmente, as organizações modernas enfrentam um número cada vez maior de desafios relacionados à resiliência, especialmente quando as expectativas dos clientes mudam para uma mentalidade sempre ativa e sempre disponível. Equipes remotas e aplicativos complexos e distribuídos estão associados a uma necessidade crescente de lançamentos frequentes. Como resultado, uma organização e seus aplicativos precisam ser mais resilientes do que nunca.

AWS define resiliência como a capacidade de um aplicativo resistir ou se recuperar de interrupções, incluindo aquelas relacionadas à infraestrutura, serviços dependentes, configurações incorretas e problemas transitórios de rede. (Consulte Resiliência e os componentes da confiabilidade na documentação do AWS Well-Architected Framework Reliability Pillar.) No entanto, para alcançar o nível desejado de resiliência, muitas vezes são necessárias compensações. A complexidade operacional, a complexidade da engenharia e o custo precisarão ser avaliados e ajustados adequadamente.

Com base em anos de trabalho com clientes e equipes internas, AWS desenvolveu uma estrutura de ciclo de vida de resiliência que captura os aprendizados e as melhores práticas de resiliência. A estrutura descreve cinco estágios principais que são ilustrados no diagrama a seguir. Em cada estágio, você pode usar estratégias, serviços e mecanismos para melhorar sua postura de resiliência.



Esses estágios são discutidos nas seções a seguir deste guia:

- Etapa 1: Estabeleça objetivos
- Etapa 2: Projetar e implementar
- Etapa 3: avaliar e testar
- Estágio 4: Operar
- Etapa 5: responda e aprenda

Termos e definições

Os conceitos de resiliência de cada estágio são aplicados em diferentes níveis, desde componentes individuais até sistemas inteiros. A implementação desses conceitos requer uma definição clara de vários termos:

- Um componente é um elemento que executa uma função e consiste em recursos de software e tecnologia. Exemplos de componentes incluem configuração de código, infraestrutura, como rede,

ou até mesmo servidores, armazenamentos de dados e dependências externas, como dispositivos de autenticação multifator (MFA).

- Um aplicativo é uma coleção de componentes que agregam valor comercial, como uma loja virtual voltada para o cliente ou o processo de back-end que aprimora os modelos de aprendizado de máquina. Um aplicativo pode consistir em um subconjunto de componentes em uma única AWS conta ou pode ser uma coleção de vários componentes que abrangem várias Contas da AWS regiões.
- Um sistema é uma coleção de aplicativos, pessoas e processos necessários para gerenciar uma determinada função comercial. Ela engloba o aplicativo necessário para executar uma função; processos operacionais como integração e entrega contínuas (CI/CD), observabilidade, gerenciamento de configuração, resposta a incidentes e recuperação de desastres; e os operadores que gerenciam essas tarefas.
- Uma interrupção é um evento que impede que seu aplicativo forneça sua função comercial adequadamente.
- A deficiência é o efeito que uma interrupção tem em um aplicativo se não for atenuada. Os aplicativos podem ser prejudicados se sofrerem um conjunto de interrupções.

Resiliência contínua

O ciclo de vida da resiliência é um processo contínuo. Mesmo dentro da mesma organização, suas equipes de aplicativos podem atuar em diferentes níveis de integridade em cada estágio, dependendo dos requisitos do seu aplicativo. No entanto, quanto mais completo for cada estágio, maior será o nível de resiliência do seu aplicativo.

Você deve pensar no ciclo de vida da resiliência como um processo padrão que sua organização pode operacionalizar. AWS modelou intencionalmente o ciclo de vida de resiliência para ser semelhante ao ciclo de vida de desenvolvimento de software (SDLC), com o objetivo de incorporar planejamento, teste e aprendizado em todos os processos operacionais enquanto você desenvolve e opera seus aplicativos. Como acontece com muitos processos de desenvolvimento ágil, o ciclo de vida da resiliência pode ser repetido a cada iteração do processo de desenvolvimento. Recomendamos que você aprofunde as práticas em cada estágio do ciclo de vida progressivamente ao longo do tempo.

Etapa 1: Estabeleça objetivos

Entender qual nível de resiliência é necessário e como você o medirá é a base para o estágio de objetivos definidos. É difícil melhorar algo se você não tem um objetivo e não consegue medi-lo.

Nem todos os aplicativos precisam do mesmo nível de resiliência. Ao definir objetivos, considere o nível necessário para fazer os investimentos e compensações corretos. Uma boa analogia para isso é um carro: ele tem quatro pneus, mas carrega apenas um pneu sobressalente. A chance de furar vários pneus durante uma viagem é baixa, e ter peças extras pode prejudicar outras características, como espaço de carga ou eficiência de combustível, portanto, essa é uma compensação razoável.

Depois de definir os objetivos, você implementa os controles de observabilidade em estágios posteriores (Estágio 2: projetar e implementar e Estágio 4: Operar) para entender se os objetivos estão sendo atingidos.

Mapeando aplicativos críticos

Definir objetivos de resiliência não deve ser exclusivamente uma conversa técnica. Em vez disso, comece com um foco voltado para os negócios para entender o que o aplicativo deve oferecer e as consequências da deficiência. Essa compreensão dos objetivos de negócios então se espalha para áreas como arquitetura, engenharia e operações. Qualquer objetivo de resiliência que você definir pode ser aplicado a todos os seus aplicativos, mas a forma como os objetivos são medidos geralmente varia de acordo com a função do aplicativo. Você pode estar executando um aplicativo essencial para os negócios e, se esse aplicativo estiver danificado, sua organização poderá perder uma receita significativa ou sofrer danos à reputação. Como alternativa, você pode ter outro aplicativo que não seja tão crítico e possa tolerar algum tempo de inatividade sem afetar negativamente a capacidade de sua organização de fazer negócios.

Como exemplo, pense em um aplicativo de gerenciamento de pedidos para uma empresa de varejo. Se os componentes do aplicativo de gerenciamento de pedidos estiverem danificados e não funcionarem adequadamente, as novas vendas não serão realizadas. Essa empresa de varejo também tem uma cafeteria para seus funcionários localizada em um de seus edifícios. A cafeteria tem um menu on-line que os funcionários podem acessar em uma página estática. Se essa página ficar indisponível, alguns funcionários poderão reclamar, mas isso não necessariamente causará danos financeiros à empresa. Com base nesse exemplo, a empresa provavelmente escolheria ter metas de resiliência mais agressivas para o aplicativo de gerenciamento de pedidos, mas não faria um investimento significativo para garantir a resiliência do aplicativo web.

Identificar os aplicativos mais críticos, onde aplicar mais esforço e onde fazer concessões é tão importante quanto ser capaz de medir a resiliência de um aplicativo na produção. Para entender melhor o impacto da deficiência, você pode realizar uma análise de impacto nos negócios (BIA). Um BIA fornece uma abordagem estruturada e sistemática para identificar e priorizar aplicativos comerciais essenciais, avaliar possíveis riscos e impactos e identificar dependências de suporte. O BIA ajuda a quantificar o custo do tempo de inatividade dos aplicativos mais importantes da sua organização. Essa métrica ajuda a descrever quanto custará se um aplicativo específico for prejudicado e incapaz de concluir sua função. No exemplo anterior, se o aplicativo de gerenciamento de pedidos estiver danificado, o negócio de varejo poderá perder uma receita significativa.

Mapeando histórias de usuários

Durante o processo de BIA, você pode descobrir que um aplicativo é responsável por mais de uma função comercial ou que uma função comercial exige vários aplicativos. Usando o exemplo anterior de uma empresa de varejo, a função de gerenciamento de pedidos pode exigir aplicativos separados para finalização de compra, promoção e preços. Se um aplicativo falhar, o impacto poderá ser sentido pela empresa e pelos usuários que interagem com a empresa. Por exemplo, talvez a empresa não consiga adicionar novos pedidos, fornecer acesso a promoções e descontos ou atualizar o preço de seus produtos. Essas diferentes funções exigidas pela função de gerenciamento de pedidos podem depender de vários aplicativos. Essas funções também podem ter várias dependências externas, o que torna o processo de obtenção de resiliência puramente focada em componentes muito complexo. A melhor maneira de lidar com esse cenário é focar nas histórias de usuários, que descrevem a experiência que os usuários esperam ao interagir com um aplicativo ou um conjunto de aplicativos.

O foco nas histórias de usuários ajuda você a entender quais partes da experiência do cliente são mais importantes, para que você possa criar mecanismos de proteção contra ameaças específicas. No exemplo anterior, uma história de usuário poderia ser checkout, que envolve o aplicativo de checkout e depende do aplicativo de preços. Outra história de usuário pode ser a visualização de promoções, que envolve o aplicativo de promoção. Depois de mapear os aplicativos mais importantes e suas histórias de usuário, você pode começar a definir as métricas que usará para medir a resiliência dessas histórias de usuários. Essas métricas podem ser aplicadas em um portfólio inteiro ou em histórias de usuários individuais.

Definindo medidas

Objetivos de ponto de recuperação (RPOs), objetivos de tempo de recuperação (RTOs) e objetivos de nível de serviço (SLOs) são medidas padrão do setor usadas para avaliar a resiliência de um determinado sistema. O RPO se refere à quantidade de perda de dados que a empresa pode tolerar em caso de falha, enquanto o RTO é uma medida da rapidez com que um aplicativo deve estar disponível novamente após uma interrupção. Essas duas métricas são medidas em unidades de tempo: segundos, minutos e horas. Você também pode medir a quantidade de tempo durante o qual o aplicativo está funcionando corretamente; ou seja, ele executa suas funções conforme projetado e está acessível aos usuários. Esses SLOs detalham o nível esperado de serviço que os clientes receberão e são medidos por métricas como a porcentagem (%) de solicitações atendidas sem erros em um tempo de resposta inferior a um segundo (por exemplo, 99,99% das solicitações receberão uma resposta a cada mês). O RPO e o RTO estão relacionados às estratégias de recuperação de desastres, supondo que haverá interrupções na operação do aplicativo e nos processos de recuperação que vão desde a restauração de backups até o redirecionamento do tráfego do usuário. Os SLOs são resolvidos por meio da implementação de controles de alta disponibilidade, que tendem a reduzir o tempo de inatividade de um aplicativo.

As métricas de SLO são comumente usadas na definição de contratos de nível de serviço (SLAs), que são contratos entre provedores de serviços e usuários finais. Os SLAs geralmente vêm com compromissos financeiros e descrevem penalidades que precisam ser pagas pelo provedor se esses acordos não forem cumpridos. No entanto, um SLA não é uma medida de sua postura de resiliência, e aumentar um SLA não torna seu aplicativo mais resiliente.

Você pode começar a definir seus objetivos com base em SLOs, RPOs e RTOs. Depois de definir seus objetivos de resiliência e obter uma compreensão clara de suas metas de RPO e RTO, você pode usar AWS Resilience Hubpara executar uma avaliação de sua arquitetura para descobrir possíveis pontos fracos relacionados à resiliência. AWS Resilience Hub avalia uma arquitetura de aplicativo em relação às melhores práticas do AWS Well-Architected Framework e compartilha orientações de remediação no contexto do que especificamente precisa ser aprimorado para atender às suas metas definidas de RTO e RPO.

Criação de medidas adicionais

RPO, RTO e SLOs são bons indicadores de resiliência, mas você também pode pensar nas metas do ponto de vista comercial e definir objetivos em torno das funções do seu aplicativo. Por exemplo, seu objetivo pode ser: Pedidos bem-sucedidos por minuto permanecerão acima de 98% se a

latência entre meu front-end e back-end aumentar em 40%.Ou: Os fluxos iniciados por segundo permanecerão dentro de um desvio padrão da média, mesmo se um componente específico for perdido. Você também pode criar objetivos para reduzir o tempo médio de recuperação (MTTR) em todos os tipos de falha conhecidos; por exemplo: os tempos de recuperação serão reduzidos em x% se algum desses problemas conhecidos ocorrer. A criação de objetivos alinhados a uma necessidade comercial ajuda você a antecipar os tipos de falhas que seu aplicativo deve tolerar. Também ajuda a identificar abordagens para reduzir a probabilidade de comprometimento do seu aplicativo.

Se você pensar no objetivo de continuar operando se perder 5% das instâncias que alimentam seu aplicativo, você pode determinar que seu aplicativo deve ser pré-escalado ou ter a capacidade de escalar com rapidez suficiente para suportar o tráfego adicional causado durante esse evento. Ou você pode determinar que deve aproveitar diferentes padrões de arquitetura, conforme descrito na seção Etapa 2: Projeto e implementação.

Você também deve implementar medidas de observabilidade para seus objetivos comerciais específicos. Por exemplo, você pode acompanhar a taxa média do pedido, o preço médio do pedido, o número médio de assinaturas ou outras métricas que podem fornecer informações sobre a saúde da empresa com base no comportamento do seu aplicativo. Ao implementar recursos de observabilidade para seu aplicativo, você pode criar alarmes e agir se essas métricas excederem seus limites definidos. A observabilidade é abordada com mais detalhes na seção Estágio 4: Operar.

Etapa 2: Projetar e implementar

Na etapa anterior, você define seus objetivos de resiliência. Agora, no estágio de projeto e implementação, você tenta antecipar os modos de falha e identificar as opções de projeto, conforme orientado pelos objetivos definidos no estágio anterior. Você também define estratégias para gerenciamento de mudanças e desenvolve código de software e configuração de infraestrutura. As seções a seguir destacam as AWS melhores práticas que você deve considerar ao considerar compensações como custo, complexidade e sobrecarga operacional.

AWS Estrutura Well-Architected

Ao arquitetar seu aplicativo com base nos objetivos de resiliência desejados, você precisa avaliar vários fatores e fazer concessões na arquitetura mais ideal. Para criar um aplicativo altamente resiliente, você deve considerar aspectos de design, construção e implantação, segurança e operações. O AWS Well-Architected Framework fornece um conjunto de melhores práticas, princípios de design e padrões arquitetônicos para ajudá-lo a projetar aplicativos resilientes. AWS Os seis pilares do AWS Well-Architected Framework fornecem as melhores práticas para projetar e operar sistemas resilientes, seguros, eficientes, econômicos e sustentáveis. A estrutura fornece uma maneira de medir consistentemente suas arquiteturas em relação às melhores práticas e identificar áreas de melhoria.

Veja a seguir exemplos de como o AWS Well-Architected Framework pode ajudá-lo a projetar e implementar aplicativos que atendam aos seus objetivos de resiliência:

- O pilar da confiabilidade: O pilar da confiabilidade enfatiza a importância de criar aplicativos que possam operar de forma correta e consistente, mesmo durante falhas ou interrupções. Por exemplo, o AWS Well-Architected Framework recomenda que você use uma arquitetura de microsserviços para tornar seus aplicativos menores e mais simples, para que você possa diferenciar as necessidades de disponibilidade de diferentes componentes em seu aplicativo. Você também pode encontrar descrições detalhadas das melhores práticas para criar aplicativos usando limitação, repetição com recuo exponencial, falha rápida (redução de carga), idempotência, trabalho constante, disjuntores e estabilidade estática.
- Análise abrangente: O AWS Well-Architected Framework incentiva uma revisão abrangente de sua arquitetura em relação às melhores práticas e princípios de design. Ele fornece uma maneira de medir consistentemente suas arquiteturas e identificar áreas de melhoria.

- Gerenciamento de riscos: O AWS Well-Architected Framework ajuda você a identificar e gerenciar riscos que podem afetar a confiabilidade do seu aplicativo. Ao abordar cenários de possíveis falhas de forma proativa, você pode reduzir sua probabilidade ou o comprometimento resultante.
- Melhoria contínua: a resiliência é um processo contínuo, e o AWS Well-Architected Framework enfatiza a melhoria contínua. Ao revisar e refinar regularmente sua arquitetura e processos com base na orientação do AWS Well-Architected Framework, você pode garantir que seus sistemas permaneçam resilientes diante dos desafios e requisitos em evolução.

Entendendo as dependências

Compreender as dependências de um sistema é fundamental para a resiliência. As dependências incluem as conexões entre componentes dentro de um aplicativo e conexões com componentes fora do aplicativo, como APIs de terceiros e serviços compartilhados de propriedade da empresa. Compreender essas conexões ajuda a isolar e gerenciar interrupções, pois uma deficiência em um componente pode afetar outros componentes. Esse conhecimento ajuda os engenheiros a avaliar o impacto das deficiências, planejar adequadamente e garantir que os recursos sejam usados de forma eficaz. Compreender as dependências ajuda você a criar estratégias alternativas e coordenar os processos de recuperação. Também ajuda a determinar casos em que você pode substituir uma dependência física por uma dependência flexível, para que seu aplicativo possa continuar cumprindo sua função comercial quando houver uma deficiência de dependência. As dependências também influenciam as decisões sobre balanceamento de carga e escalabilidade de aplicativos. Compreender as dependências é vital quando você faz alterações em seu aplicativo, pois pode ajudá-lo a determinar possíveis riscos e impactos. Esse conhecimento ajuda você a criar aplicativos estáveis e resilientes, auxiliando no gerenciamento de falhas, avaliação de impacto, recuperação de deficiências, balanceamento de carga, escalabilidade e gerenciamento de mudanças. Você pode rastrear dependências manualmente ou usar ferramentas e serviços AWS X-Raypara entender as dependências de seus aplicativos distribuídos.

estratégias de recuperação de desastres

Uma estratégia de recuperação de desastres (DR) desempenha um papel fundamental na criação e operação de aplicativos resilientes, principalmente garantindo a continuidade dos negócios. Isso garante que as operações comerciais cruciais possam persistir com o mínimo de prejuízo possível, mesmo durante eventos catastróficos, minimizando assim o tempo de inatividade e a potencial perda de receita. As estratégias de DR são essenciais para a proteção de dados porque geralmente incorporam backups e replicação de dados regulares em vários locais, o que ajuda a

proteger informações comerciais valiosas e a evitar a perda total durante um desastre. Além disso, muitos setores são regulados por políticas que exigem que as empresas tenham uma estratégia de DR para proteger dados confidenciais e garantir que os serviços permaneçam disponíveis durante um desastre. Ao garantir o mínimo de comprometimento do serviço, uma estratégia de DR também reforça a confiança e a satisfação do cliente. Uma estratégia de DR bem implementada e praticada com frequência reduz o tempo de recuperação após um desastre e ajuda a garantir que os aplicativos sejam rapidamente colocados novamente on-line. Além disso, os desastres podem gerar custos substanciais, não apenas pela perda de receita devido ao tempo de inatividade, mas também pelas despesas de restauração de aplicativos e dados. Uma estratégia de DR bem projetada ajuda a se proteger contra essas perdas financeiras.

A estratégia escolhida depende das necessidades específicas do seu aplicativo, do seu RTO e RPO e do seu orçamento. AWS Elastic Disaster Recoveryé um serviço de resiliência desenvolvido especificamente que você pode usar para ajudar a implementar sua estratégia de DR para aplicativos locais e baseados em nuvem.

Para obter mais informações, consulte Recuperação de desastres de cargas de trabalho AWS e Fundamentos AWS multirregionais no site. AWS

Definindo estratégias de CI/CD

Uma das causas comuns de falhas no aplicativo é o código ou outras alterações que alteram o aplicativo de um estado de funcionamento conhecido anteriormente. Se você não abordar o gerenciamento de mudanças com cuidado, isso pode causar deficiências frequentes. A frequência das mudanças aumenta a oportunidade de impacto. No entanto, fazer alterações com menos frequência resulta em conjuntos de mudanças maiores, que têm muito mais probabilidade de resultar em prejuízos devido à sua alta complexidade. As práticas de integração contínua e entrega contínua (CI/CD) são projetadas para manter as mudanças pequenas e frequentes (resultando em maior produtividade), ao mesmo tempo em que submetem cada alteração a um alto nível de inspeção por meio da automação. Algumas das estratégias fundamentais são:

- Automação total: O conceito fundamental de CI/CD é automatizar os processos de construção e implantação o máximo possível. Isso inclui criação, teste, implantação e até mesmo monitoramento. As tubulações automatizadas ajudam a reduzir a possibilidade de erro humano, garantem a consistência e tornam o processo mais confiável e eficiente.
- Desenvolvimento orientado a testes (TDD): escreva testes antes de escrever o código do aplicativo. Essa prática garante que todo código tenha testes associados, o que melhora a

confiabilidade do código e a qualidade da inspeção automatizada. Esses testes são executados no pipeline de CI para validar as alterações.

- Confirmações e integrações frequentes: incentive os desenvolvedores a confirmarem códigos com frequência e realizarem integrações com frequência. Alterações pequenas e frequentes são mais fáceis de testar e depurar, o que reduz o risco de problemas significativos. A automação reduz o custo de cada confirmação e implantação, possibilitando integrações frequentes.
- Infraestrutura imutável: trate seus servidores e outros componentes da infraestrutura como entidades estáticas e imutáveis. Substitua a infraestrutura em vez de modificá-la o máximo possível e crie uma nova infraestrutura por meio de código testado e implantado em seu pipeline.
- Mecanismo de reversão: sempre tenha uma maneira fácil, confiável e frequentemente testada de reverter as alterações se algo der errado. Ser capaz de retornar rapidamente ao bom estado anterior conhecido é essencial para a segurança da implantação. Isso pode ser um botão simples para reverter ao estado anterior ou pode ser totalmente automatizado e iniciado por alarmes.
- Controle de versão: mantenha todo o código, a configuração e até mesmo a infraestrutura do aplicativo como código em um repositório com controle de versão. Essa prática ajuda a garantir que você possa rastrear facilmente as alterações e revertê-las, se necessário.
- Implantações Canary e implantações em azul/verde: implantar primeiro novas versões do seu aplicativo em um subconjunto de sua infraestrutura ou manter dois ambientes (azul/verde) permite verificar o comportamento de uma mudança na produção e revertê-la rapidamente, se necessário.

O CI/CD não diz respeito apenas às ferramentas, mas também à cultura. Criar uma cultura que valorize a automação, os testes e o aprendizado com as falhas é tão importante quanto implementar as ferramentas e os processos certos. As reversões, se feitas muito rapidamente com impacto mínimo, não devem ser consideradas uma falha, mas uma experiência de aprendizado.

Conduzindo ORRs

Uma revisão de prontidão operacional (ORR) ajuda a identificar lacunas operacionais e processuais. Na Amazon, criamos ORRs para transformar o aprendizado de décadas operando serviços de alta escala em perguntas selecionadas com orientação de melhores práticas. Um ORR captura as lições aprendidas anteriormente e exige que novas equipes garantam que tenham considerado essas lições em seus aplicativos. Os ORRs podem fornecer uma lista de modos de falha ou causas de falha que podem ser incluídos na atividade de modelagem de resiliência descrita na seção de modelagem de resiliência abaixo. Para obter mais informações, consulte Operational Readiness Reviews (ORRs) no site Well-Architected AWS Framework.

Entendendo os limites de isolamento de AWS falhas

AWS fornece vários limites de isolamento de falhas para ajudá-lo a atingir seus objetivos de resiliência. Você pode usar esses limites para aproveitar o escopo previsível de contenção de impacto que eles fornecem. Você deve estar familiarizado com a forma como os AWS serviços são projetados usando esses limites para poder fazer escolhas intencionais sobre as dependências selecionadas para seu aplicativo. Para entender como usar limites em seu aplicativo, consulte Limites de isolamento de AWS falhas no AWS site.

Seleção de respostas

Um sistema pode responder de várias maneiras a um alarme. Alguns alarmes podem exigir uma resposta da equipe de operações, enquanto outros podem acionar mecanismos de autorrecuperação dentro do aplicativo. Você pode decidir manter respostas que possam ser automatizadas como operações manuais para controlar os custos da automação ou gerenciar restrições de engenharia. É provável que o tipo de resposta a um alarme seja selecionado em função do custo de implementação da resposta, da frequência prevista do alarme, da precisão do alarme e da possível perda comercial de não responder ao alarme.

Por exemplo, quando um processo do servidor falha, o processo pode ser reiniciado pelo sistema operacional, ou um novo servidor pode ser provisionado e o antigo encerrado, ou um operador pode ser instruído a se conectar remotamente ao servidor e reiniciá-lo. Essas respostas têm o mesmo resultado — reiniciar o processo do servidor de aplicativos — mas têm níveis variados de custos de implementação e manutenção.

> **Note**
>
> Você pode selecionar várias respostas para adotar uma abordagem de resiliência aprofundada. Por exemplo, no cenário anterior, a equipe de aplicativos pode optar por implementar todas as três respostas com um intervalo de tempo entre cada uma. Se o indicador de processo do servidor com falha ainda estiver em estado de alarme após 30 segundos, a equipe pode presumir que o sistema operacional falhou ao reiniciar o servidor de aplicativos. Portanto, eles podem criar um grupo de escalonamento automático para criar um novo servidor virtual e restaurar o processo do servidor de aplicativos. Se o indicador ainda estiver em estado de alarme após 300 segundos, um alerta poderá ser enviado à equipe operacional para se conectar ao servidor original e tentar restaurar o processo.

A resposta que a equipe de aplicativos e a empresa selecionam deve refletir o apetite da empresa em compensar a sobrecarga operacional com um investimento inicial em tempo de engenharia. Você deve escolher uma resposta, um padrão de arquitetura, como estabilidade estática, um padrão de software, como um disjuntor, ou um procedimento operacional, considerando cuidadosamente as restrições e a manutenção prevista de cada opção de resposta. Algumas respostas padrão podem existir para orientar as equipes de aplicativos, para que você possa usar as bibliotecas e os padrões gerenciados pela função de arquitetura centralizada como uma entrada para essa consideração.

Modelagem de resiliência

A modelagem de resiliência documenta como um aplicativo responderá às diferentes interrupções previstas. Ao antecipar interrupções, sua equipe pode implementar processos de observabilidade, controles automatizados e recuperação para mitigar ou evitar deficiências apesar das interrupções. AWS criou diretrizes para o desenvolvimento de um modelo de resiliência usando a estrutura de análise de resiliência. Essa estrutura pode ajudá-lo a antecipar interrupções e seu impacto em seu aplicativo. Ao antecipar as interrupções, você pode identificar as mitigações necessárias para criar um aplicativo resiliente e confiável. Recomendamos que você use a estrutura de análise de resiliência para atualizar seu modelo de resiliência a cada iteração do ciclo de vida do seu aplicativo. O uso dessa estrutura em cada iteração ajuda a reduzir incidentes, antecipando interrupções durante a fase de design e testando o aplicativo antes e depois da implantação da produção. Desenvolver um modelo de resiliência usando essa estrutura ajuda a garantir que você atenda aos seus objetivos de resiliência.

Falhando com segurança

Se você não conseguir evitar interrupções, falhe com segurança. Considere criar seu aplicativo com um modo de operação padrão à prova de falhas, no qual nenhuma perda significativa de negócios possa ocorrer. Um exemplo de estado à prova de falhas para um banco de dados seria usar como padrão as operações somente para leitura, nas quais os usuários não têm permissão para criar ou alterar nenhum dado. Dependendo da sensibilidade dos dados, você pode até querer que o aplicativo adote como padrão um estado de desligamento e nem mesmo realize consultas somente para leitura. Considere qual deve ser o estado à prova de falhas do seu aplicativo e use como padrão esse modo de operação sob condições extremas.

Etapa 3: avaliar e testar

Durante a fase de avaliação e teste do ciclo de vida, o aplicativo ou as alterações em um aplicativo existente foram projetados, mas ainda não foram lançados para produção. Nesse estágio, você implementa atividades para testar as práticas que foram realizadas nas etapas anteriores e avaliar os resultados. O aplicativo ainda pode estar em desenvolvimento ativo ou o desenvolvimento primário pode estar concluído e o aplicativo pode estar sendo testado antes de ser lançado para produção. Durante esse estágio, você se concentra em desenvolver e executar testes que confirmem ou refutem as expectativas de que o aplicativo atenderá aos objetivos definidos de resiliência. Além disso, você desenvolve e testa os procedimentos operacionais do sistema. Os procedimentos de implantação que você desenvolveu no estágio 2: estágio de projeto e implementação são colocados em prática e os resultados são avaliados. Embora essas atividades de teste e avaliação comecem durante essa parte do ciclo de vida, elas não terminam aqui. Os testes e a avaliação continuam à medida que você avança para o estágio 4: estágio de operação.

A etapa de avaliação e teste é dividida em duas fases: atividades de pré-implantação e atividades de pós-implantação. As atividades de pré-implantação consistem em tarefas que devem ser concluídas antes da implantação do aplicativo em qualquer ambiente, incluindo a implantação de novas versões do software, bem como a implantação inicial em um ambiente de teste. As atividades de pós-implantação ocorrem após a implantação do software em um ambiente de teste ou produção. As seções a seguir discutem essas fases com mais detalhes.

Atividades de pré-implantação

Design de ambiente

O ambiente no qual você testa e avalia seu aplicativo afeta a profundidade com que você pode testá-lo e a confiança que você tem de que esses resultados refletem com precisão o que acontecerá na produção. Talvez você possa realizar alguns testes de integração localmente em máquinas de desenvolvedores usando serviços como o Amazon DynamoDB (consulte Como configurar o DynamoDB local na documentação do DynamoDB). No entanto, em algum momento, você precisará testar em um ambiente que replique seu ambiente de produção para obter a maior confiança em seus resultados. Esse ambiente gerará custos, por isso recomendamos que você adote uma abordagem em etapas ou em pipeline em seus ambientes, em que ambientes semelhantes aos de produção apareçam posteriormente.

Teste de integração

O teste de integração é o processo de testar se um componente bem definido de um aplicativo executa suas funções corretamente quando opera com dependências externas. Essas dependências externas podem ser outros componentes desenvolvidos de forma personalizada, AWS serviços que você usa para seu aplicativo, dependências de terceiros e dependências locais. Este guia se concentra em testes de integração que demonstram a resiliência do seu aplicativo. Ele pressupõe que já existam testes unitários e de integração que demonstrem a precisão funcional do seu software.

Recomendamos que você projete testes de integração que testem especificamente os padrões de resiliência que você implementou, como padrões de disjuntores ou redução de carga (consulte Etapa 2: Projeto e implementação). Os testes de integração orientados à resiliência geralmente envolvem a aplicação de uma carga específica ao aplicativo ou a introdução intencional de interrupções no ambiente usando recursos como ().AWS Fault Injection ServiceAWS FIS Idealmente, você deve executar todos os testes de integração como parte do seu pipeline de CI/CD e garantir a execução de testes sempre que o código for confirmado. Isso ajuda você a detectar e reagir rapidamente a quaisquer alterações no código ou nas configurações que resultem em violações de seus objetivos de resiliência. Aplicativos distribuídos em grande escala são complexos, e até mesmo pequenas alterações podem afetar significativamente a resiliência de partes aparentemente não relacionadas do seu aplicativo. Tente executar seus testes em cada commit. AWS fornece um excelente conjunto de ferramentas para operar seu pipeline de CI/CD e outras DevOps ferramentas. Para obter mais informações, consulte Introdução ao DevOps AWS on no AWS site.

Pipelines de implantação automatizados

A implantação e o teste em seus ambientes de pré-produção é uma tarefa repetitiva e complexa que é melhor deixar para a automação. A automação desse processo libera recursos humanos e reduz a oportunidade de erro. O mecanismo para automatizar esse processo geralmente é chamado de pipeline. Ao criar seu pipeline, recomendamos que você configure uma série de ambientes de teste que se aproximem cada vez mais da sua configuração de produção. Você usa essa série de ambientes para testar repetidamente seu aplicativo. O primeiro ambiente fornece um conjunto de recursos mais limitado do que o ambiente de produção, mas tem um custo significativamente menor. Ambientes subsequentes devem adicionar serviços e escalabilidade para espelhar mais de perto o ambiente de produção.

Comece testando no primeiro ambiente. Depois que suas implantações passarem por todos os testes no primeiro ambiente de teste, deixe o aplicativo ser executado sob certa quantidade de

carga por um período de tempo para ver se algum problema ocorre ao longo do tempo. Confirme se você configurou a observabilidade corretamente (consulte Precisão do alarme mais adiante neste guia) para que você possa detectar quaisquer problemas que surjam. Quando esse período de observação for concluído com êxito, implante seu aplicativo em seu próximo ambiente de teste e repita o processo, adicionando testes ou cargas adicionais conforme suportado pelo ambiente. Depois de testar suficientemente seu aplicativo dessa forma, você pode usar os métodos de implantação configurados anteriormente para implantar o aplicativo na produção (consulte Definir estratégias de CI/CD anteriormente neste guia). O artigo Automatizando implantações seguras e sem intervenção na Amazon Builders' Library é um excelente recurso que descreve como a Amazon automatiza a implantação de código. O número de ambientes que precedem sua implantação de produção variará, dependendo da complexidade do seu aplicativo e dos tipos de dependências que ele tem.

Testes de carga

Superficialmente, o teste de carga se assemelha ao teste de integração. Você testa uma função discreta do seu aplicativo e suas dependências externas para verificar se ele funciona conforme o esperado. O teste de carga então vai além do teste de integração para se concentrar em como o aplicativo funciona sob cargas bem definidas. O teste de carga exige a verificação da funcionalidade correta, portanto, ele deve ocorrer após um teste de integração bem-sucedido. É importante entender o quão bem o aplicativo responde sob as cargas esperadas e como ele se comporta quando a carga excede as expectativas. Isso ajuda a verificar se você implementou os mecanismos necessários para garantir que seu aplicativo permaneça resiliente sob carga extrema. Para obter um guia abrangente sobre testes de carga em AWS, consulte Teste de carga distribuído AWS ativado na Biblioteca de AWS soluções.

Atividades de pós-implantação

A resiliência é um processo contínuo e a avaliação da resiliência do seu aplicativo deve continuar após a implantação do aplicativo. Os resultados de suas atividades pós-implantação, como avaliações contínuas de resiliência, podem exigir que você reavalie e atualize algumas das atividades de resiliência realizadas anteriormente no ciclo de vida da resiliência.

Conduzindo avaliações de resiliência

A avaliação da resiliência não para depois que você implanta seu aplicativo na produção. Mesmo que você tenha pipelines de implantação bem definidos e automatizados, às vezes as mudanças

podem ocorrer diretamente em um ambiente de produção. Além disso, pode haver fatores que você ainda não tenha levado em consideração na verificação de resiliência pré-implantação. AWS Resilience Hubfornece um local central onde você pode avaliar se sua arquitetura implantada atende às suas necessidades definidas de RPO e RTO. Você pode usar esse serviço para executar avaliações sob demanda da resiliência do seu aplicativo, automatizar avaliações e até mesmo integrá-las às suas ferramentas de CI/CD, conforme discutido na postagem do AWS blog Avaliando continuamente a resiliência do aplicativo com e. AWS Resilience HubAWS CodePipeline Automatizar essas avaliações é uma prática recomendada porque ajuda a garantir que você esteja avaliando continuamente sua postura de resiliência na produção.

teste de DR

Na Etapa 2: Projeto e implementação, você desenvolveu estratégias de recuperação de desastres (DR) como parte do seu sistema. Durante a Etapa 4, você deve testar seus procedimentos de DR para garantir que sua equipe esteja totalmente preparada para um incidente e que seus procedimentos funcionem conforme o esperado. Você deve testar regularmente todos os seus procedimentos de DR, incluindo failover e failback, e analisar os resultados de cada exercício para determinar se e como os procedimentos do seu sistema devem ser atualizados para obter o melhor resultado possível. Ao desenvolver seu teste de DR inicialmente, agende o teste com bastante antecedência e garanta que toda a equipe entenda o que esperar, como os resultados serão medidos e qual mecanismo de feedback será usado para atualizar os procedimentos com base no resultado. Depois de se tornar proficiente na execução de testes de DR agendados, considere executar testes de DR sem aviso prévio. Desastres reais não ocorrem de acordo com um cronograma, então você precisa estar preparado para exercitar seu plano a qualquer momento. No entanto, sem aviso prévio não significa não planejado. As principais partes interessadas ainda precisam planejar o evento para garantir que o monitoramento adequado esteja em vigor e que os clientes e os aplicativos essenciais não sejam afetados adversamente.

Detecção de desvios

Mudanças imprevistas na configuração em aplicativos de produção podem ocorrer mesmo quando há automação e procedimentos bem definidos. Para detectar alterações na configuração do seu aplicativo, você deve ter mecanismos para detectar desvios, que se referem a desvios de uma configuração básica. Para saber como detectar desvios em suas AWS CloudFormation pilhas, consulte Detecção de alterações de configuração não gerenciadas em pilhas e recursos na documentação. AWS CloudFormation Para detectar desvios no AWS ambiente do seu aplicativo, consulte Detectar e resolver desvios AWS Control TowerAWS Control Tower na documentação.

Teste sintético

O teste sintético é o processo de criação de software configurável que é executado em produção, de forma programada, para testar as APIs do seu aplicativo de uma forma que simule a experiência do usuário final. Esses testes às vezes são chamados de canários, em referência ao uso original do termo na mineração de carvão. Os testes sintéticos geralmente podem fornecer avisos antecipados quando um aplicativo sofre uma interrupção, mesmo que a deficiência seja parcial ou intermitente, como geralmente acontece com falhas cinzentas.

Engenharia do caos

A engenharia do caos é um processo sistemático que envolve submeter deliberadamente um aplicativo a eventos disruptivos de forma mitigada pelos riscos, monitorando de perto sua resposta e implementando as melhorias necessárias. Seu objetivo é validar ou desafiar suposições sobre a capacidade do aplicativo de lidar com essas interrupções. Em vez de deixar esses eventos ao acaso, a engenharia do caos capacita os engenheiros a orquestrar experimentos em um ambiente controlado, normalmente durante períodos de baixo tráfego e com suporte de engenharia prontamente disponível para uma mitigação eficaz.

A engenharia do caos começa com a compreensão das condições operacionais normais, conhecidas como estado estacionário, da aplicação em questão. A partir daí, você formula uma hipótese que detalha o comportamento bem-sucedido do aplicativo na presença de interrupções. Você executa o experimento, que envolve a injeção deliberada de interrupções, incluindo, mas não se limitando a, latência da rede, falhas no servidor, erros no disco rígido e comprometimento de dependências externas. Em seguida, você analisa os resultados do experimento e aprimora a resiliência do aplicativo com base em seus aprendizados. O experimento serve como uma ferramenta valiosa para melhorar várias facetas do aplicativo, incluindo seu desempenho, e revela problemas latentes que poderiam ter permanecido ocultos de outra forma. Além disso, a engenharia do caos ajuda a revelar deficiências nas ferramentas de observabilidade e alarme e ajuda a refiná-las. Também contribui para reduzir o tempo de recuperação e aprimorar as habilidades operacionais. A engenharia do caos acelera a adoção das melhores práticas e cultiva uma mentalidade de melhoria contínua. Em última análise, permite que as equipes desenvolvam e aprimorem suas habilidades operacionais por meio da prática regular e da repetição.

AWS recomenda que você inicie seus esforços de engenharia do caos em um ambiente que não seja de produção. Você pode usar AWS Fault Injection Service (AWS FIS) para executar experimentos de engenharia de caos com falhas de uso geral, bem como falhas exclusivas de. AWS

Esse serviço totalmente gerenciado inclui alarmes de condições de parada e controles completos de permissão para que você possa adotar facilmente a engenharia do caos com segurança e confiança.

Estágio 4: Operar

Depois de concluir a Etapa 3: avaliar e testar, você estará pronto para implantar o aplicativo na produção. No estágio Operate, você implanta seu aplicativo na produção e gerencia a experiência de seus clientes. O design e a implementação de seu aplicativo determinam muitos de seus resultados de resiliência, mas esse estágio se concentra nas práticas operacionais que seu sistema usa para manter e melhorar a resiliência. Construir uma cultura de excelência operacional ajuda a criar padrões e consistência nessas práticas.

Observabilidade

A parte mais importante para entender a experiência do cliente é por meio de monitoramento e alarmes. Você precisa instrumentar seu aplicativo para entender seu estado e precisa de perspectivas diversas, o que significa que você precisa medir tanto do lado do servidor quanto do lado do cliente, normalmente com canários. Suas métricas devem incluir dados sobre as interações do seu aplicativo com suas dependências e dimensões que se alinham aos limites de isolamento de falhas. Você também deve produzir registros que forneçam detalhes adicionais sobre cada unidade de trabalho executada pelo seu aplicativo. Você pode considerar combinar métricas e registros usando uma solução como o formato de métrica CloudWatch incorporada da Amazon. Você provavelmente descobrirá que sempre quer mais observabilidade, então considere as compensações de custo, esforço e complexidade necessárias para implementar o nível desejado de instrumentação.

Os links a seguir fornecem as melhores práticas para instrumentar seu aplicativo e criar alarmes:

- Serviços de monitoramento de produção na Amazon (apresentação do AWS re:Invent 2020)
- Amazon Builders' Library: excelência operacional na Amazon (apresentação do re:Invent 2021)AWS
- Melhores práticas de observabilidade na Amazon (apresentação do AWS re:Invent 2022)
- Instrumentando sistemas distribuídos para visibilidade operacional (artigo da Amazon Builders' Library)
- Criação de painéis para visibilidade operacional (artigo da Amazon Builders' Library)

Gerenciamento de eventos

Você deve ter um processo de gerenciamento de eventos para lidar com deficiências quando seus alarmes (ou pior, seus clientes) informam que algo está errado. Esse processo deve incluir a contratação de um operador de plantão, a escalada de problemas e o estabelecimento de runbooks para abordagens consistentes de solução de problemas que ajudem a remover erros humanos. No entanto, as deficiências geralmente não ocorrem isoladamente; um único aplicativo pode afetar vários outros aplicativos que dependem dele. Você pode resolver problemas rapidamente entendendo todos os aplicativos afetados e reunindo operadores de várias equipes em uma única teleconferência. No entanto, dependendo do tamanho e da estrutura da sua organização, esse processo pode exigir uma equipe de operações centralizada.

Além de configurar um processo de gerenciamento de eventos, você deve revisar regularmente suas métricas por meio de painéis. As avaliações regulares ajudam você a entender a experiência do cliente e as tendências de longo prazo no desempenho do seu aplicativo. Isso ajuda você a identificar problemas e gargalos antes que eles causem um impacto significativo na produção. Analisar as métricas de forma consistente e padronizada oferece benefícios significativos, mas exige uma adesão de cima para baixo e um investimento de tempo.

Os links a seguir fornecem as melhores práticas na criação de painéis e análises de métricas operacionais:

- Criação de painéis para visibilidade operacional (artigo da Amazon Builders' Library)
- A abordagem da Amazon para falhar com sucesso (apresentação do AWS re:Invent 2019)

Resiliência contínua

Durante a Etapa 2: Projeto e implementação e Etapa 3: Avaliação e teste, você iniciou as atividades de revisão e teste antes de implantar seu aplicativo na produção. Durante a fase de operação, você deve continuar iterando essas atividades na produção. Você deve revisar periodicamente a postura de resiliência do seu aplicativo por meio de análises do AWS Well-Architected Framework, Operational Readiness Reviews(ORRs) e da estrutura de análise de resiliência. Isso ajuda a garantir que seu aplicativo não se desvie das linhas de base e dos padrões estabelecidos e mantém você atualizado com orientações novas ou atualizadas. Essas atividades de resiliência contínua ajudam você a descobrir interrupções anteriormente imprevistas e a criar novas mitigações.

Você também pode considerar realizar dias de jogo e experimentos de engenharia do caos na produção depois de executá-los com sucesso em ambientes de pré-produção. Os dias de jogo

simulam eventos conhecidos que você criou mecanismos de resiliência para mitigar. Por exemplo, um dia de jogo pode simular uma falha no serviço AWS regional e implementar um failover multirregional. Embora a implementação dessas atividades possa exigir um nível significativo de esforço, ambas as práticas ajudam a criar confiança de que seu sistema é resiliente aos modos de falha que você o projetou para suportar.

Ao operar seus aplicativos, enfrentar eventos operacionais, revisar métricas e testar seu aplicativo, você encontrará inúmeras oportunidades de responder e aprender.

Etapa 5: responda e aprenda

A forma como seu aplicativo responde a eventos disruptivos influencia sua confiabilidade. Aprender com a experiência e como seu aplicativo reagiu às interrupções no passado também é fundamental para melhorar sua confiabilidade.

O estágio Responder e aprender se concentra nas práticas que você pode implementar para responder melhor a eventos disruptivos em seus aplicativos. Também inclui práticas para ajudá-lo a extrair o máximo de aprendizado das experiências de suas equipes de operações e engenheiros.

Criação de relatórios de análise de incidentes

Quando ocorre um incidente, a primeira ação é evitar mais danos aos clientes e à empresa o mais rápido possível. Depois que o aplicativo for recuperado, a próxima etapa é entender o que aconteceu e identificar as etapas para evitar a recorrência. Essa análise pós-incidente geralmente é capturada como um relatório que documenta o conjunto de eventos que levaram ao comprometimento do aplicativo e os efeitos da interrupção no aplicativo, nos clientes e nos negócios. Esses relatórios se tornam valiosos artefatos de aprendizado e devem ser amplamente compartilhados em toda a empresa.

> **Note**
>
> É fundamental realizar a análise de incidentes sem atribuir nenhuma culpa. Suponha que todos os operadores tenham adotado o melhor e mais adequado curso de ação, com base nas informações de que dispunham. Não use os nomes dos operadores ou engenheiros em um relatório. Citar o erro humano como motivo da deficiência pode fazer com que os membros da equipe sejam protegidos para se protegerem, resultando na captura de informações incorretas ou incompletas.

Um bom relatório de análise de incidentes, como o documentado no processo Amazon Correction of Error (COE), segue um formato padronizado e tenta capturar, com o máximo de detalhes possível, as condições que levaram ao comprometimento do aplicativo. O relatório detalha uma série de eventos com data e hora e captura dados quantitativos (geralmente métricas e capturas de tela de painéis de monitoramento) que descrevem o estado mensurável do aplicativo ao longo do tempo. O relatório deve capturar os processos de pensamento dos operadores e engenheiros que agiram e as informações que os levaram a suas conclusões. O relatório também deve detalhar o

desempenho de diferentes indicadores – por exemplo, quais alarmes foram acionados, se esses alarmes refletiram com precisão o estado do aplicativo, o intervalo de tempo entre os eventos e os alarmes resultantes e o tempo para resolver o incidente. A linha do tempo também captura os runbooks ou automações que foram iniciados e como eles ajudaram o aplicativo a recuperar um estado útil. Esses elementos do cronograma ajudam sua equipe a entender a eficácia das respostas automatizadas e do operador, incluindo a rapidez com que resolveram o problema e a eficácia com que foram na mitigação da interrupção.

Essa imagem detalhada de um evento histórico é uma ferramenta educacional poderosa. As equipes devem armazenar esses relatórios em um repositório central que esteja disponível para toda a empresa, para que outras pessoas possam analisar os eventos e aprender com eles. Isso pode melhorar a intuição de suas equipes sobre o que pode dar errado na produção.

Um repositório de relatórios detalhados de incidentes também se torna uma fonte de material de treinamento para operadores. As equipes podem usar um relatório de incidentes para inspirar um dia de jogo de mesa ou ao vivo, em que as equipes recebem informações que reproduzem a linha do tempo capturada no relatório. Os operadores podem percorrer o cenário com informações parciais da linha do tempo e descrever quais ações eles tomariam. O moderador do dia do jogo pode então fornecer orientação sobre como o aplicativo respondeu com base nas ações do operador. Isso desenvolve as habilidades de solução de problemas dos operadores, para que eles possam antecipar e solucionar problemas com mais facilidade.

Uma equipe centralizada responsável pela confiabilidade do aplicativo deve manter esses relatórios em uma biblioteca centralizada que toda a organização possa acessar. Essa equipe também deve ser responsável por manter o modelo de relatório e treinar as equipes sobre como preencher o relatório de análise de incidentes. A equipe de confiabilidade deve revisar periodicamente os relatórios para detectar tendências em toda a empresa que possam ser abordadas por meio de bibliotecas de software, padrões de arquitetura ou alterações nos processos da equipe.

Conduzindo análises operacionais

Conforme discutido na Etapa 4: Operação, as análises operacionais são uma oportunidade de analisar lançamentos recentes de recursos, incidentes e métricas operacionais. A análise operacional também é uma oportunidade de compartilhar os aprendizados sobre lançamentos de recursos e incidentes com a comunidade mais ampla de engenharia da sua organização. Durante a análise operacional, as equipes analisam as implantações de recursos que foram revertidas, os incidentes que ocorreram e como eles foram tratados. Isso dá aos engenheiros de toda a organização a oportunidade de aprender com as experiências de outras pessoas e fazer perguntas.

Abra suas análises operacionais para a comunidade de engenharia da sua empresa para que eles possam aprender mais sobre os aplicativos de TI que administram os negócios e os tipos de problemas que podem encontrar. Eles levarão esse conhecimento à medida que projetam, implementam e implantam outros aplicativos para a empresa.

Analisando o desempenho do alarme

Os alarmes, conforme discutido na fase de operação, podem resultar na criação de alertas no painel, na criação de tickets, no envio de e-mails ou na chamada de operadores. Um aplicativo terá vários alarmes configurados para monitorar vários aspectos de sua operação. Com o tempo, a precisão e a eficácia desses alarmes devem ser revisadas para aumentar a precisão do alarme, reduzir os falsos positivos e consolidar alertas duplicados.

Precisão do alarme

Os alarmes devem ser tão específicos quanto possível para reduzir o tempo gasto interpretando ou diagnosticando a interrupção específica que causou o alarme. Quando um alarme é acionado em resposta a uma falha no aplicativo, os operadores que recebem e respondem ao alarme devem primeiro interpretar as informações que o alarme transmite. As informações podem ser um código de erro simples que mapeia um curso de ação, como um procedimento de recuperação, ou podem incluir linhas dos registros do aplicativo que você precisa revisar para entender por que o alarme foi disparado. À medida que sua equipe aprende a operar um aplicativo com mais eficiência, ela deve refinar esses alarmes para torná-los o mais claros e concisos possível.

Você não pode prever todas as possíveis interrupções em um aplicativo, então sempre haverá alarmes gerais que exigem que um operador analise e diagnostique. Sua equipe deve trabalhar para reduzir o número de alarmes gerais a fim de melhorar os tempos de resposta e diminuir o tempo médio de reparo (MTTR). Idealmente, deve haver uma one-to-one relação entre um alarme e uma resposta automatizada ou executada por humanos.

Falsos positivos

Os alarmes que não exigem nenhuma ação dos operadores, mas produzem alertas como e-mails, páginas ou tickets, serão ignorados pelos operadores ao longo do tempo. Periodicamente, ou como parte de uma análise de incidentes, revise os alarmes para identificar aqueles que geralmente são ignorados ou que não exigem nenhuma ação dos operadores (falsos positivos). Você deve trabalhar para remover o alarme ou melhorá-lo para que ele emita um alerta acionável aos operadores.

Falsos negativos

Durante um incidente, os alarmes configurados para alertar durante o incidente podem falhar, talvez devido a um evento que afeta o aplicativo de forma inesperada. Como parte de uma análise de incidentes, você deve revisar os alarmes que deveriam ter sido acionados, mas não foram. Você deve trabalhar para melhorar esses alarmes para que eles reflitam melhor as condições que podem surgir de um evento. Como alternativa, talvez seja necessário criar alarmes adicionais mapeados para a mesma interrupção, mas gerados por um sintoma diferente da interrupção.

Alertas duplicativos

Uma interrupção que prejudique seu aplicativo provavelmente causará vários sintomas e resultará em vários alarmes. Periodicamente, ou como parte de uma análise de incidentes, você deve revisar os alarmes e alertas emitidos. Se os operadores receberam alertas duplicados, crie alarmes agregados para consolidá-los em uma única mensagem de alerta.

Conduzindo análises de métricas

Sua equipe deve coletar métricas operacionais sobre seu aplicativo, como o número de incidentes por gravidade por mês, o tempo para detectar o incidente, o tempo para identificar a causa, o tempo para remediar e o número de tickets criados, alertas enviados e páginas levantadas. Analise essas métricas pelo menos uma vez por mês para entender a carga da equipe operacional, a signal-to-noise proporção com a qual ela lida (por exemplo, alertas informativos versus alertas acionáveis) e se a equipe está melhorando sua capacidade de operar os aplicativos sob seu controle. Use essa análise para entender as tendências nos aspectos mensuráveis da equipe de operações. Solicite ideias da equipe sobre como melhorar essas métricas.

Fornecendo treinamento e capacitação

É difícil capturar uma descrição detalhada de um aplicativo e de seu ambiente que provocou um incidente ou comportamento inesperado. Além disso, modelar a resiliência do seu aplicativo para antecipar esses cenários nem sempre é simples. Sua organização deve investir em materiais de treinamento e capacitação para que suas equipes operacionais e desenvolvedores participem de atividades como modelagem de resiliência, análise de incidentes, dias de jogos e experimentos de engenharia do caos. Isso aumentará a fidelidade dos relatórios que suas equipes produzem e o conhecimento que elas capturam. As equipes também estarão mais bem equipadas para antecipar

falhas sem depender de um grupo menor e mais experiente de engenheiros, que precisará fornecer suas ideias por meio de revisões programadas.

Criação de uma base de conhecimento sobre incidentes

Um relatório de incidentes é uma saída padrão de uma análise de incidentes. Você deve usar o mesmo relatório ou um semelhante para documentar cenários em que detectou um comportamento anômalo do aplicativo, mesmo que o aplicativo não tenha sido prejudicado. Use a mesma estrutura de relatórios padronizada para capturar o resultado de experimentos de caos e dias de jogo. O relatório representa um instantâneo do aplicativo e de seu ambiente que levou a um incidente ou a um comportamento inesperado. Você deve armazenar esses relatórios padronizados em um repositório central que todos os engenheiros da empresa possam acessar.

As equipes de operações e os desenvolvedores podem então pesquisar essa base de conhecimento para entender o que interrompeu os aplicativos no passado, quais tipos de cenários poderiam ter causado interrupções e o que evitou o comprometimento dos aplicativos. Essa base de conhecimento se torna um acelerador para aprimorar as habilidades de suas equipes operacionais e de seus desenvolvedores e permite que eles compartilhem seus conhecimentos e experiências. Além disso, você pode usar os relatórios como material de treinamento ou cenários para dias de jogos ou experimentos de caos para melhorar a intuição e a capacidade da equipe operacional de solucionar interrupções.

> **Note**
>
> Um formato de relatório padronizado também fornece aos leitores uma sensação de familiaridade e os ajuda a encontrar as informações que estão procurando com mais rapidez.

Implementando resiliência em profundidade

Conforme discutido anteriormente, uma organização avançada implementará várias respostas a um alarme. Não há garantia de que uma resposta será eficaz, portanto, ao agrupar as respostas em camadas, um aplicativo estará mais bem equipado para falhar normalmente. Recomendamos que você implemente pelo menos duas respostas para cada indicador para garantir que uma resposta individual não se torne um único ponto de falha que possa levar a um cenário de DR. Essas camadas devem ser criadas em ordem serial, para que uma resposta sucessiva seja executada somente se a resposta anterior for ineficaz. Você não deve executar várias respostas em camadas

para um único alarme. Em vez disso, use um alarme que indique se uma resposta não foi bem-sucedida e, em caso afirmativo, inicie a próxima resposta em camadas.

Conclusão e atributos

Este guia apresenta um ciclo de vida que ajuda você a melhorar continuamente a resiliência de seus aplicativos implementando as melhores práticas em cinco estágios: definir objetivos, projetar e implementar, avaliar e testar, operar, responder e aprender.

Para obter mais informações sobre os serviços e conceitos discutidos neste guia, consulte os recursos a seguir.

AWS serviços:

- AWS Backup
- AWS Elastic Disaster Recovery
- AWS Fault Injection Service (AWS FIS)
- AWS Resilience Hub
- Controlador de recuperação de aplicativos Amazon (ARC)
- AWS X-Ray

Postagens e artigos no blog:

- Disponibilidade e muito mais: entendendo e melhorando a resiliência de sistemas distribuídos em AWS
- AWS Limites de isolamento de falhas
- AWS Fundamentos de várias regiões
- Engenharia do caos na nuvem
- Avaliando continuamente a resiliência do aplicativo com e AWS Resilience HubAWS CodePipeline
- Recuperação de desastres de aplicativos locais para AWS
- Pilar de confiabilidade — AWS Well-Architected Framework
- Framework de análise de resiliência

Colaboradores

Os colaboradores deste guia incluem:

- Bruno Emer, arquiteto principal de soluções, AWS
- Clark Richey, arquiteto principal de soluções, AWS
- Elaine Harvey, gerente geral de serviços de confiabilidade, AWS
- Jason Barto, arquiteto principal de soluções, AWS
- John Formento, arquiteto principal de soluções, AWS
- Lisi Lewis, gerente sênior de marketing de produto, AWS
- Michael Haken, arquiteto principal de soluções, AWS
- Neeraj Kumar, arquiteto principal de soluções, AWS
- Wangechi Doble, arquiteto principal de soluções, AWS

Histórico do documento

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

Alteração	Descrição	Data
Publicação inicial	—	6 de outubro de 2023

AWS Glossário de orientação prescritiva

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

Números

7 Rs

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

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

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

A

ABAC

Consulte controle de acesso baseado em atributos.

serviços abstratos

Veja os serviços gerenciados.

ACID

Veja atomicidade, consistência, isolamento, durabilidade.

migração ativa-ativa

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

migração ativa-passiva

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

função agregada

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

AI

Veja a inteligência artificial.

AIOps

Veja as operações de inteligência artificial.

anonimização

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

antipadrões

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

controle de aplicativos

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

portfólio de aplicações

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

inteligência artificial (IA)

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

operações de inteligência artificial (AIOps)

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

criptografia assimétrica

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

atomicidade, consistência, isolamento, durabilidade (ACID)

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

controle de acesso por atributo (ABAC)

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

fonte de dados autorizada

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

Zona de disponibilidade

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

AWS Estrutura de adoção da nuvem (AWS CAF)

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

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

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

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

B

bot ruim

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

BCP

Veja o planejamento de continuidade de negócios.

gráfico de comportamento

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

sistema big-endian

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

classificação binária

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

filtro de bloom

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

blue/green deployment (implantação azul/verde)

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

bot

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

botnet

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

ramo

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

acesso em vidro quebrado

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

estratégia brownfield

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

cache do buffer

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

capacidade de negócios

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

ser orientadas por recursos de negócios. Para obter mais informações, consulte a seção Organizados de acordo com as capacidades de negócios do whitepaper Executar microsserviços conteinerizados na AWS.

planejamento de continuidade de negócios (BCP)

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

C

CAF

Consulte Estrutura de adoção da AWS nuvem.

implantação canária

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

CCoE

Veja o Centro de Excelência em Nuvem.

CDC

Veja a captura de dados de alterações.

captura de dados de alterações (CDC)

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

engenharia do caos

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

CI/CD

Veja a integração e a entrega contínuas.

classificação

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

criptografia no lado do cliente

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

Centro de excelência em nuvem (CCoE)

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

computação em nuvem

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

modelo operacional em nuvem

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

estágios de adoção da nuvem

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

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

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

informações sobre como eles se relacionam com a estratégia de AWS migração, consulte o guia de preparação para migração.

CMDB

Consulte o banco de dados de gerenciamento de configuração.

repositório de código

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

cache frio

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

dados frios

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

visão computacional (CV)

Um campo da IA que usa aprendizado de máquina para analisar e extrair informações de formatos visuais, como imagens e vídeos digitais. Por exemplo, AWS Panorama oferece dispositivos que adicionam CV às redes de câmeras locais, e a Amazon SageMaker AI fornece algoritmos de processamento de imagem para CV.

desvio de configuração

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

banco de dados de gerenciamento de configuração (CMDB)

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

pacote de conformidade

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

integração contínua e entrega contínua (CI/CD)

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

CV

Veja visão computacional.

D

dados em repouso

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

classificação de dados

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

desvio de dados

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

dados em trânsito

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

malha de dados

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

minimização de dados

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

perímetro de dados

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

pré-processamento de dados

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

proveniência dos dados

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

titular dos dados

Um indivíduo cujos dados estão sendo coletados e processados.

data warehouse

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

linguagem de definição de dados (DDL)

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

linguagem de manipulação de dados (DML)

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

DDL

Consulte a linguagem de definição de banco de dados.

deep ensemble

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

Aprendizado profundo

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

defense-in-depth

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

administrador delegado

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

implantação

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

ambiente de desenvolvimento

Veja o ambiente.

controle detectivo

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

mapeamento do fluxo de valor de desenvolvimento (DVSM)

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

gêmeo digital

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

tabela de dimensões

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

desastre

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

Recuperação de desastres (RD)

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

DML

Veja a linguagem de manipulação de banco de dados.

design orientado por domínio

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

DR

Veja a recuperação de desastres.

detecção de deriva

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

DVSM

Veja o mapeamento do fluxo de valor do desenvolvimento.

E

EDA

Veja a análise exploratória de dados.

EDI

Veja intercâmbio eletrônico de dados.

computação de borda

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

intercâmbio eletrônico de dados (EDI)

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

Criptografia

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

chave de criptografia

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

endianismo

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

endpoint

Veja o endpoint do serviço.

serviço de endpoint

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

planejamento de recursos corporativos (ERP)

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

criptografia envelopada

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

ambiente

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

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

epic

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

ERP

Veja o planejamento de recursos corporativos.

análise exploratória de dados (EDA)

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

F

tabela de fatos

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

falham rapidamente

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

limite de isolamento de falhas

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

ramificação de recursos

Veja a filial.

recursos

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

importância do recurso

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

transformação de recursos

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

solicitação de alguns instantes

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

FGAC

Veja o controle de acesso refinado.

Controle de acesso refinado (FGAC)

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

migração flash-cut

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

FM

Veja o modelo da fundação.

modelo de fundação (FM)

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

G

IA generativa

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

bloqueio geográfico

Veja as restrições geográficas.

restrições geográficas (bloqueio geográfico)

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

Fluxo de trabalho do GitFlow

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

imagem dourada

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

estratégia greenfield

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

barreira de proteção

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

H

HA

Veja a alta disponibilidade.

migração heterogênea de bancos de dados

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

o esquema pode ser uma tarefa complexa. O AWS fornece o AWS SCT para ajudar nas conversões de esquemas.

alta disponibilidade (HA)

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

modernização de historiador

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

dados de retenção

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

migração homogênea de bancos de dados

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

dados quentes

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

hotfix

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

período de hipercuidados

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

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

eu

IaC

Veja a infraestrutura como código.

Política baseada em identidade

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

aplicação ociosa

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

IIoT

Veja a Internet das Coisas industrial.

infraestrutura imutável

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

VPC de entrada (admissão)

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

migração incremental

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

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

Indústria 4.0

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

infraestrutura

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

Infraestrutura como código (IaC)

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

Internet industrial das coisas (IIoT)

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

VPC de inspeção

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

Internet das Coisas (IoT)

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

interpretabilidade

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

IoT

Consulte Internet das Coisas.

Biblioteca de informações de TI (ITIL)

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

Gerenciamento de serviços de TI (ITSM)

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

ITIL

Consulte a biblioteca de informações de TI.

ITSM

Veja o gerenciamento de serviços de TI.

L

controle de acesso baseado em etiqueta (LBAC)

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

zona de pouso

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

informações sobre zonas de pouso, consulte Configurar um ambiente da AWS com várias contas seguro e escalável.

modelo de linguagem grande (LLM)

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

migração de grande porte

Uma migração de 300 servidores ou mais.

LBAC

Veja controle de acesso baseado em rótulos.

privilégio mínimo

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

mover sem alterações (lift-and-shift)

Veja 7 Rs.

sistema little-endian

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

LLM

Veja um modelo de linguagem grande.

ambientes inferiores

Veja o ambiente.

M

machine learning (ML)

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

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

ramificação principal

Veja a filial.

malware

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

serviços gerenciados

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

sistema de execução de manufatura (MES)

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

MAP

Consulte Migration Acceleration Program.

mecanismo

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

conta de membro

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

MES

Veja o sistema de execução de manufatura.

Transporte de telemetria de enfileiramento de mensagens (MQTT)

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

microsserviço

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

arquitetura de microsserviços

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

Programa de Aceleração da Migração (MAP)

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

migração em escala

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

fábrica de migração

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

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

metadados de migração

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

padrão de migração

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

Avaliação de Portfólio para Migração (MPA)

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

Avaliação de Preparação para Migração (MRA)

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

estratégia de migração

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

ML

Veja o aprendizado de máquina.

modernização

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

avaliação de preparação para modernização

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

aplicações monolíticas (monólitos)

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

MAPA

Consulte Avaliação do portfólio de migração.

MQTT

Consulte Transporte de telemetria de enfileiramento de mensagens.

classificação multiclasse

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

infraestrutura mutável

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

O

OAC

Veja o controle de acesso de origem.

CARVALHO

Veja a identidade de acesso de origem.

OCM

Veja o gerenciamento de mudanças organizacionais.

migração offline

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

OI

Veja a integração de operações.

OLA

Veja o contrato em nível operacional.

migração online

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

OPC-UA

Consulte Comunicação de processo aberto — Arquitetura unificada.

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

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

acordo de nível operacional (OLA)

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

análise de prontidão operacional (ORR)

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

tecnologia operacional (OT)

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

integração de operações (OI)

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

trilha organizacional

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

gerenciamento de alterações organizacionais (OCM)

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

controle de acesso de origem (OAC)

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

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

Identidade do acesso de origem (OAI)

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

ORR

Veja a análise de prontidão operacional.

OT

Veja a tecnologia operacional.

VPC de saída (egresso)

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

P

limite de permissões

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

Informações de identificação pessoal (PII)

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

PII

Veja as informações de identificação pessoal.

manual

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

PLC

Consulte controlador lógico programável.

AMEIXA

Veja o gerenciamento do ciclo de vida do produto.

política

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

persistência poliglota

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

avaliação do portfólio

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

predicado

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

pressão de predicados

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

controle preventivo

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

principal (entidade principal)

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

privacidade por design

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

zonas hospedadas privadas

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

controle proativo

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

gerenciamento do ciclo de vida do produto (PLM)

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

ambiente de produção

Veja o ambiente.

controlador lógico programável (PLC)

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

encadeamento imediato

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

pseudonimização

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

publish/subscribe (pub/sub)

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

Q

plano de consulta

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

regressão de planos de consultas

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

R

Matriz RACI

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

RAG

Consulte Geração Aumentada de Recuperação.

ransomware

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

Matriz RASCI

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

RCAC

Veja o controle de acesso por linha e coluna.

réplica de leitura

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

rearquiteta

Veja 7 Rs.

objetivo de ponto de recuperação (RPO).

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

objetivo de tempo de recuperação (RTO)

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

refatorar

Veja 7 Rs.

Região

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

regressão

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

redefinir a hospedagem

Veja 7 Rs.

versão

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

realocar

Veja 7 Rs.

redefinir a plataforma

Veja 7 Rs.

recomprar

Veja 7 Rs.

resiliência

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

política baseada em recurso

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

matriz responsável, accountable, consultada, informada (RACI)

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

controle responsivo

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

reter

Veja 7 Rs.

aposentar-se

Veja 7 Rs.

Geração Aumentada de Recuperação (RAG)

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

alternância

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

controle de acesso por linha e coluna (RCAC)

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

RPO

Veja o objetivo do ponto de recuperação.

RTO

Veja o objetivo do tempo de recuperação.

runbook

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

S

SAML 2.0

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

SCADA

Veja controle de supervisão e aquisição de dados.

SCP

Veja a política de controle de serviços.

secret

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

segurança por design

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

controle de segurança

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

fortalecimento da segurança

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

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

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

automação de resposta de segurança

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

Criptografia do lado do servidor

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

política de controle de serviços (SCP)

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

service endpoint (endpoint de serviço)

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

acordo de serviço (SLA)

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

indicador de nível de serviço (SLI)

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

objetivo de nível de serviço (SLO)

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

modelo de responsabilidade compartilhada

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

SIEM

Veja informações de segurança e sistema de gerenciamento de eventos.

ponto único de falha (SPOF)

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

SLA

Veja o contrato de nível de serviço.

ESGUIO

Veja o indicador de nível de serviço.

SLO

Veja o objetivo do nível de serviço.

split-and-seed modelo

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

CUSPE

Veja um único ponto de falha.

esquema de estrelas

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

padrão strangler fig

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

sub-rede

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

controle de supervisão e aquisição de dados (SCADA)

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

symmetric encryption (criptografia simétrica)

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

testes sintéticos

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

prompt do sistema

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

T

tags

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

variável-alvo

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

lista de tarefas

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

ambiente de teste

Veja o ambiente.

treinamento

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

gateway de trânsito

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

fluxo de trabalho baseado em troncos

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

Acesso confiável

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

tuning (ajustar)

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

equipe de duas pizzas

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

U

incerteza

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

tarefas indiferenciadas

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

ambientes superiores

Veja o ambiente.

V

aspiração

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

controle de versões

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

emparelhamento da VPC

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

Vulnerabilidade

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

W

cache quente

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

dados mornos

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

função de janela

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

workload

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

workstreams

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

MINHOCA

Veja escrever uma vez, ler muitas.

WQF

Consulte Estrutura de qualificação AWS da carga de trabalho.

escreva uma vez, leia muitas (WORM)

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

Z

exploração de dia zero

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

vulnerabilidade de dia zero

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

aviso zero-shot

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

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

aplicação zumbi

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

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