Escolhendo a GitOps ferramenta certa para seu cluster Amazon EKS

AWS Orientação prescritiva

AWS Orientação prescritiva: Escolhendo a GitOps ferramenta certa para seu cluster Amazon EKS

Table of Contents

Escolhendo a GitOps ferramenta certa para seu cluster Amazon EKS

Pradip Kumar Pandey e Pratap Kumar Nanda, da Amazon Web Services ()AWS

Abril de 2025 (histórico do documento)

No cenário em rápida evolução das tecnologias nativas da nuvem, GitOps surgiu como uma metodologia poderosa para gerenciar e implantar aplicativos e infraestrutura. Se você estiver usando o Amazon Elastic Kubernetes Service (Amazon EKS) GitOps , os princípios de implementação podem aprimorar significativamente seus processos de implantação, melhorar a confiabilidade e simplificar as operações. Uma variedade de GitOps ferramentas está disponível, e escolher a correta para seu cluster EKS é uma decisão crítica que pode afetar a eficiência de sua equipe e o sucesso geral de suas DevOps práticas.

A seleção de uma GitOps ferramenta apropriada para seu ambiente Amazon EKS envolve uma análise cuidadosa de vários fatores, incluindo seus requisitos específicos, a experiência da equipe, as necessidades de escalabilidade e os recursos de integração com os existentes Serviços da AWS. Cada ferramenta vem com seu próprio conjunto de recursos, pontos fortes e possíveis limitações, por isso é essencial alinhar sua escolha às metas e ao contexto operacional da sua organização.

Este guia explora as principais considerações na seleção de GitOps ferramentas para o Amazon EKS, compara as opções usadas com frequência e fornece informações para ajudar você a tomar uma decisão informada. Ele abrange nove GitOps ferramentas populares:

- Argo CD
- Fluxo
- Tecer GitOps
- Jenkins X
- GitLab CI/CD
- Spinnaker
- Frota de fazendeiros
- Codefresh
- Pulumi

Resultados de negócios desejados

A lista a seguir discute possíveis metas e resultados quando você escolhe uma ferramenta para implementar GitOps princípios em seus processos de desenvolvimento e operações.

Integração perfeita com o Amazon EKS

Sua GitOps ferramenta deve se integrar perfeitamente ao Amazon EKS e fornecer compatibilidade com recursos e otimizações específicos do Amazon EKS.

- Suporte nativo ao Amazon EKS: procure ferramentas que ofereçam suporte integrado para o Amazon EKS, incluindo fácil conexão e gerenciamento de clusters.
- AWS service (Serviço da AWS) integração: certifique-se de que a ferramenta possa interagir com outras, Serviços da AWS como AWS Identity and Access Management (IAM), Amazon Elastic Container Registry (Amazon ECR) e Amazon. CloudWatch
- Compatibilidade do complemento Amazon EKS: confirme se a ferramenta oferece suporte aos complementos do Amazon EKS e pode gerenciá-los de forma eficaz.

Escalabilidade e desempenho

Sua GitOps ferramenta deve ser capaz de lidar com a escala de suas operações do Amazon EKS, desde pequenos clusters até grandes ambientes com vários clusters.

- Eficiência de recursos: avalie o consumo de recursos da ferramenta e seu impacto no desempenho do cluster.
- Operações em grande escala: avalie a capacidade da ferramenta de gerenciar vários aplicativos e clusters simultaneamente.
- Desempenho sob carga: considere o desempenho da ferramenta durante atualizações de alta frequência e implantações em grande escala.

Segurança e conformidade

Os recursos de segurança e conformidade são cruciais, especialmente em setores regulamentados ou quando você lida com dados confidenciais.

- Controle de acesso: procure recursos robustos de controle de acesso baseado em funções (RBAC) que se integrem ao IAM.
- Gerenciamento de segredos: avalie como a ferramenta lida com informações confidenciais e se integra com AWS Secrets Manageroutras soluções.
- Trilhas de auditoria: certifique-se de que a ferramenta forneça recursos abrangentes de registro e auditoria para conformidade e solução de problemas.
- Verificação de segurança: considere ferramentas que ofereçam verificação de segurança integrada para vulnerabilidades nas implantações.

Facilidade de uso e curva de aprendizado

A ferramenta deve ser fácil de usar e estar alinhada às habilidades da sua equipe para garantir a rápida adoção e o uso eficiente.

- Interface do usuário: avalie a intuitividade dos recursos da interface de linha de comando (CLI) e da interface gráfica do usuário (GUI).
- Qualidade da documentação: procure up-to-date documentação e tutoriais abrangentes.
- Recursos de aprendizagem: considere a disponibilidade de materiais de treinamento, cursos e recursos comunitários.

Suporte comunitário e de rede

Uma comunidade e uma rede fortes podem fornecer recursos valiosos, plug-ins e sustentabilidade a longo prazo.

- Desenvolvimento ativo: verifique a frequência das atualizações e a capacidade de resposta dos mantenedores.
- Tamanho da comunidade: considere o tamanho e a atividade da comunidade de usuários para suporte e compartilhamento de conhecimento.
- Integrações de terceiros: avalie a disponibilidade de plug-ins e integrações com outras ferramentas em sua pilha.

Recursos de gerenciamento de vários clusters

Se você tiver vários clusters EKS, a capacidade de gerenciá-los com eficiência é crucial.

- Gerenciamento centralizado: procure recursos que permitam gerenciar vários clusters a partir de um único plano de controle.
- Federação de clusters: considere ferramentas que oferecem suporte à federação Kubernetes para aplicativos de vários clusters.
- Paridade de ambiente: avalie o quão bem a ferramenta mantém a consistência em diferentes ambientes, como desenvolvimento, preparação e produção.

Observabilidade e monitoramento

A ferramenta deve fornecer informações claras sobre o estado de suas implantações e a integridade do cluster.

- Visibilidade da implantação: procure recursos que ofereçam uma visão clara do status e do histórico da implantação.
- Integração com ferramentas de monitoramento: considere o quão bem a ferramenta se integra com soluções de monitoramento populares, como Prometheus e Grafana.
- Recursos de alerta: avalie a capacidade da ferramenta de configurar e gerenciar alertas para problemas de implantação ou desvios.

Flexibilidade e personalização

A capacidade de adaptar a ferramenta aos seus fluxos de trabalho e requisitos específicos é importante para a satisfação a longo prazo.

- Extensibilidade: procure arquiteturas de plug-ins ou APIs que permitam ampliar a funcionalidade da ferramenta.
- Suporte de recursos personalizados: confirme se a ferramenta pode lidar com recursos personalizados do Kubernetes de forma eficaz.
- Personalização do fluxo de trabalho: avalie com que facilidade você pode adaptar os GitOps fluxos de trabalho às necessidades da sua equipe.

Entrega contínua e suporte progressivo à implantação

Estratégias avançadas de implantação geralmente são cruciais para minimizar riscos e garantir atualizações sem problemas.

- Implantações do Canary: procure suporte integrado para versões do Canary.
- Blue/green deployments: Assess the tool's capabilities for blue/greenestratégias de implantação.
- Mecanismos de reversão: garanta recursos robustos e de easy-to-use reversão para recuperação rápida de implantações com falha.

Custo-efetividade e utilização de recursos

Considere o custo geral da adoção e manutenção da ferramenta, incluindo custos diretos e indiretos.

- Custos de licenciamento: compare as opções de código aberto com as soluções comerciais e considere o suporte e os recursos corporativos.
- Despesas operacionais: avalie os custos operacionais adicionais em termos de gerenciamento e manutenção.
- Consumo de recursos: avalie a eficiência da ferramenta em termos de recursos de computação e armazenamento que seriam necessários.

Ao considerar cuidadosamente esses resultados e seus aspectos, você pode tomar uma decisão informada sobre a GitOps ferramenta mais adequada para seu cluster EKS e garantir que a ferramenta esteja alinhada às necessidades, capacidades e estratégia de longo prazo da sua organização.

GitOps ferramentas para clusters EKS

Atualmente, existem várias GitOps ferramentas para Kubernetes disponíveis no mercado. Aqui está uma lista de algumas das opções mais usadas:

- Argo CD
- Fluxo
- Tecer GitOps
- Jenkins X
- GitLab CI/CD
- Spinnaker
- Frota de fazendeiros
- Codefresh
- Pulumi

Siga os links para ver informações detalhadas sobre como essas ferramentas implementam GitOps práticas. Cada ferramenta tem pontos fortes e casos de uso. A escolha depende de fatores como seus requisitos específicos, infraestrutura existente, experiência da equipe e recursos desejados. É importante avaliar essas ferramentas com base nas necessidades da sua organização e na complexidade do seu ambiente Kubernetes.

Argo CD

O Argo CD é uma ferramenta de entrega GitOps contínua (CD) amplamente usada para Kubernetes que está em conformidade com vários princípios fundamentais. GitOps

GitOps apoio

Área	Capacidades da ferramenta
Configuração declarativa	O Argo CD usa configurações declarativas que são armazenadas nos repositórios Git. O estado desejado do aplicativo e da infraestr utura é definido nos arquivos YAML. Essas

Área	Capacidades da ferramenta
	configurações descrevem o que deve ser implantado, não como implantá-las.
Sistema de controle de versão como fonte única de verdade	Os repositórios Git servem como a única fonte confiável para todo o sistema. Todas as alterações no aplicativo e na infraestrutura são feitas por meio do Git. Isso garante uma trilha de auditoria completa e a capacidade de reverter para qualquer estado anterior.
Sincronização automatizada	O Argo CD monitora continuamente o repositór io Git em busca de alterações. Quando as alterações são detectadas, ele sincroniza automaticamente o estado real do cluster com o estado desejado definido no Git. Isso garante que o cluster sempre reflita o estado descrito no repositório.
Nativo do Kubernetes	O Argo CD foi projetado especificamente para ambientes Kubernetes. Ele aproveita a natureza declarativa e os recursos personali zados do Kubernetes para gerenciar aplicativ os.
Autocura e detecção de desvios	O Argo CD compara regularmente o estado ativo do cluster com o estado desejado no Git. Se detectar algum desvio (diferenças entre os estados real e desejado), ele poderá corrigir automaticamente essas discrepâncias.
Suporte a vários clusters e multilocação	O Argo CD pode gerenciar vários clusters Kubernetes a partir de uma única instância. Ele suporta multilocação, para que diferentes equipes possam gerenciar seus aplicativos de forma independente.

Área	Capacidades da ferramenta
Definição da aplicação	Os aplicativos no Argo CD são definidos usando o Application CRD (definição personali zada de recursos). Isso permite uma forma nativa do Kubernetes de definir o que deve ser implantado e como.
Separação entre implantação e lançamento	O Argo CD separa a implantação do código de seu lançamento para os usuários. Isso é obtido por meio de várias estratégias de implantação, como blue/green implantações canárias.
Observabilidade e auditabilidade	O Argo CD fornece uma interface de usuário da web e uma CLI para observar o estado dos aplicativos e clusters. Todas as ações são registradas para fornecer uma trilha de auditoria clara das mudanças e implantações.
Segurança e RBAC	O Argo CD se integra ao controle de acesso baseado em funções (RBAC) do Kubernetes. Ele oferece suporte à integração de login único para autenticação e autorização.
Arquitetura conectável	O Argo CD é compatível com vários sistemas de gerenciamento de controle de origem, gráficos Helm, Kustomize e outros formatos de manifesto do Kubernetes. Essa flexibili dade permite que ele se encaixe em diversos ambientes e fluxos de trabalho.
Entrega contínua (CD)	Embora o Argo CD se concentre na entrega contínua, ele pode ser integrado às ferrament as de integração contínua (CI) para criar um CI/ CD pipeline completo.

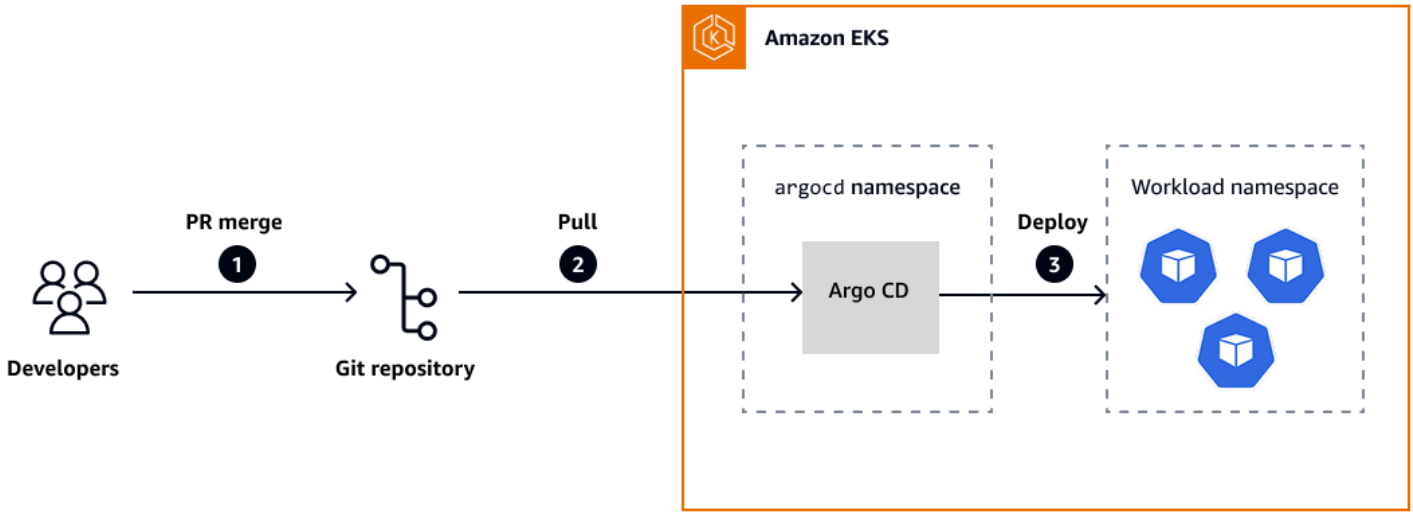
Ao aderir a esses GitOps princípios, o Argo CD fornece uma maneira robusta, escalável e segura de gerenciar as implantações do Kubernetes. Ele garante que o estado operacional do seu sistema esteja sempre sincronizado com o estado desejado definido no seu repositório Git e promove consistência, confiabilidade e facilidade de gerenciamento em ambientes complexos do Kubernetes.

Para cenários e requisitos que o Argo CD pode abordar, consulte os casos de uso do Argo CD posteriormente neste guia. Para uma comparação entre o Argo CD e o Flux, consulte Comparação de recursos posteriormente neste guia.

Para obter informações adicionais, consulte a documentação do CD Argo.

Arquitetura

O diagrama a seguir ilustra um fluxo de trabalho de CD GitOps orientado por CD que usa o Argo CD em um cluster EKS. Para obter informações detalhadas, consulte a documentação do CD Argo.



em que:

- Etapa 1: mesclagem do Pull Request (PR). Um desenvolvedor confirma as alterações nos manifestos do Kubernetes ou nos gráficos do Helm que são armazenados em um repositório Git. Quando o PR é revisado e incorporado à ramificação principal, o estado desejado do aplicativo é atualizado no controle de origem.
- Etapa 2: Sincronização do repositório. O Argo CD é executado em um namespace dedicado (`argocd`) no cluster EKS e monitora continuamente o repositório Git configurado. Quando detecta alterações, ele obtém as atualizações mais recentes para reconciliar o estado declarado.

- Etapa 3: Implantação no namespace de destino. O Argo CD compara o estado desejado do Git com o estado ativo no cluster. Em seguida, ele aplica as alterações necessárias ao namespace da carga de trabalho de destino para que o aplicativo seja implantado ou atualizado adequadamente. Isso inclui gerenciar recursos do Kubernetes, como implantações, serviços e segredos ConfigMaps, para manter a consistência do cluster com a fonte confiável do Git.

Fluxo

O Flux é outra ferramenta para Kubernetes que implementa GitOps princípios de uma forma única.

GitOps apoio

Área	Capacidades da ferramenta
Git como a única fonte da verdade	O Flux usa repositórios Git como a fonte definitiva para definir o estado desejado do sistema. Todas as configurações de aplicativos e infraestrutura são armazenadas no Git.
Configuração declarativa	O Flux trabalha com descrições declarativas do estado desejado do seu cluster. Essas descrições geralmente são manifestos do Kubernetes, gráficos do Helm ou sobreposi ções do Kustomize.
Sincronização automatizada	O Flux monitora continuamente o repositório Git em busca de alterações. Quando detecta alterações, ele as aplica automaticamente ao cluster.
Nativo do Kubernetes	O Flux é construído como um conjunto de controladores Kubernetes e recursos personali zados. Ele usa os mecanismos de extensão no Kubernetes para fornecer recursos. GitOps
Modelo de implantação baseado em pull	Ao contrário dos CI/CD sistemas tradicionais baseados em push, o Flux usa um modelo

Área	Capacidades da ferramenta
	baseado em pull. O cluster extrai o estado desejado do Git em vez de usar um sistema externo para enviar alterações.
Reconciliação contínua	O Flux compara constantemente o estado real do cluster com o estado desejado no Git. Ele corrige automaticamente qualquer desvio detectado entre esses estados.
Multilocação	O Flux oferece suporte à multilocação por meio de seus conceitos de personalizações e. HelmReleases Equipes diferentes podem gerenciar suas próprias partes da configuração de forma independente.
Entrega progressiva	O Flux oferece suporte a estratégias avançadas de implantação, como versões e A/B testes canários, por meio de seu componente Flagger.
Integração com o Helm	O Flux inclui suporte nativo para o Helm, para que você possa gerenciar facilmente as versões do Helm por meio do. GitOps
Automação de atualização de imagem	O Flux pode atualizar automaticamente as imagens do contêiner no Git quando novas versões estão disponíveis no registro do contêiner.
Personalize o suporte	Você pode usar o suporte nativo fornecido pelo Flux for Kustomize para personalizar e corrigir manifestos do Kubernetes.

Área	Capacidades da ferramenta
Segurança e RBAC	O Flux se integra ao Kubernetes RBAC para controle de acesso. Ele suporta o gerenciam ento de segredos por meio de vários back-ends .
Observabilidade	O Flux fornece informações de status e métricas sobre reconciliação e operações. Ele se integra às ferramentas de monitoramento para melhorar a observabilidade.
Arquitetura orientada a eventos	O Flux usa uma abordagem orientada por eventos para implementar reconciliações e atualizações.
Extensibilidade	A ferramenta foi projetada para ser extensível, para que você possa adicionar controladores e recursos personalizados.
Sincronização entre clusters	O Flux suporta o gerenciamento de vários clusters a partir de um único conjunto de repositórios.
Gerenciar dependências	Ele permite definir dependências entre diferentes partes do sistema e garante a ordem correta das operações.
Receptores de webhook	Você pode configurar o Flux para receber webhooks de provedores Git ou outros sistemas para iniciar a reconciliação imediata.

Ao implementar esses GitOps princípios, o Flux fornece um sistema robusto e flexível para gerenciar clusters e aplicativos Kubernetes. Ele garante que sua infraestrutura e aplicativos estejam sempre sincronizados com seus repositórios Git e fornece consistência, confiabilidade e facilidade de gerenciamento em ambientes complexos do Kubernetes. A abordagem nativa do Kubernetes e o

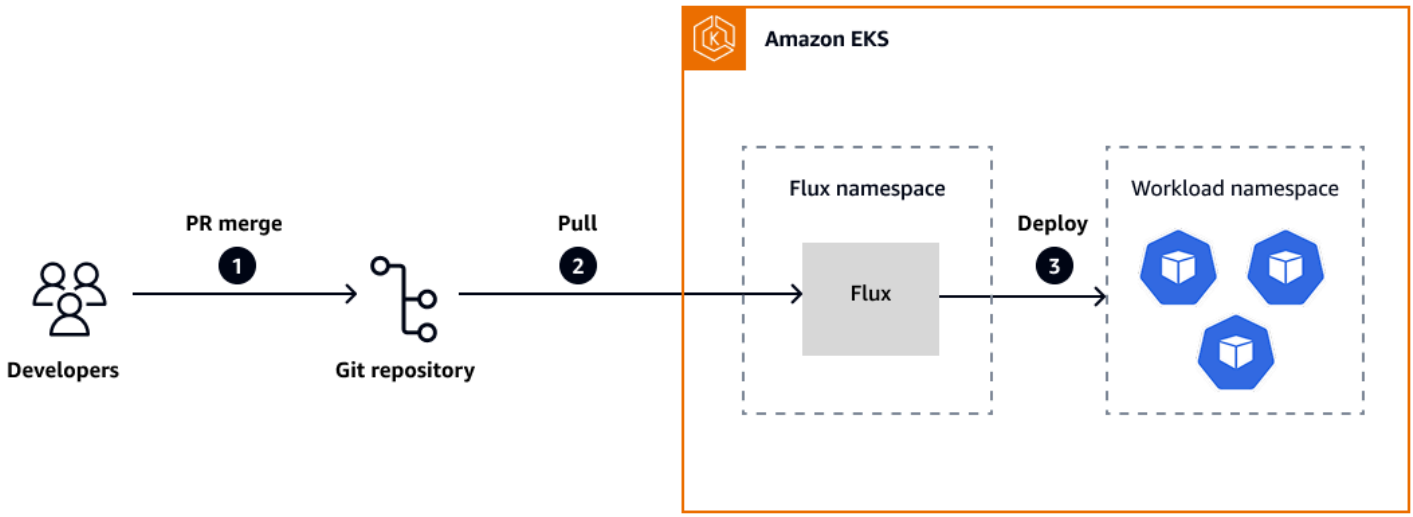
foco na automação da ferramenta a tornam particularmente adequada para ambientes nativos da nuvem.

Para cenários e requisitos que o Flux pode abordar, consulte os casos de uso do Flux posteriormente neste guia. Para uma comparação entre o Argo CD e o Flux, consulte Comparação de recursos posteriormente neste guia.

Para obter informações adicionais, consulte a documentação do Flux.

Arquitetura

O diagrama a seguir ilustra um fluxo de trabalho de CD GitOps orientado por CD que usa o Flux em um cluster EKS. Para obter informações detalhadas, consulte a documentação do Flux.



em que:

- Etapa 1: mesclagem do Pull Request (PR). Um desenvolvedor confirma as alterações nos manifestos do Kubernetes ou nos gráficos do Helm que são armazenados em um repositório Git. Quando o PR é revisado e incorporado à ramificação principal, o estado desejado do aplicativo é atualizado no controle de origem.
- Etapa 2: Sincronização do repositório. O Flux é executado em um namespace dedicado no cluster EKS e monitora continuamente o repositório Git configurado. Quando detecta alterações, ele obtém as atualizações mais recentes para reconciliar o estado declarado.
- Etapa 3: Implantação no namespace de destino. O Flux compara o estado desejado do Git com o estado ativo no cluster. Em seguida, ele aplica as alterações necessárias ao namespace da carga de trabalho de destino para que o aplicativo seja implantado ou atualizado adequadamente.

Tecer GitOps

O Weave GitOps foi desenvolvido pela Weaveworks, empresa que introduziu o termo. GitOps Essa ferramenta fornece uma GitOps solução abrangente que se baseia nos GitOps princípios fundamentais.

GitOps apoio

Área	Capacidades da ferramenta
Git como a única fonte da verdade	O Weave GitOps usa repositórios Git como fonte autorizada para definir o estado desejado do sistema. Todas as configurações, incluindo manifestos de aplicativos, definições de infraestrutura e políticas, são armazenadas no Git.
Configuração declarativa	O sistema se baseia em descrições declarati vas de todo o estado do sistema. Essas descrições geralmente são manifestos do Kubernetes, gráficos do Helm ou outros formatos declarativos.
Sincronização automatizada	O Weave monitora GitOps continuamente os repositórios Git em busca de alteraçõe s. Quando detecta alterações, ele as aplica automaticamente ao ambiente de destino.
Arquitetura nativa do Kubernetes	O Weave GitOps foi criado como um conjunto de controladores Kubernetes e recursos personalizados. Ele usa os mecanismos de extensão no Kubernetes para fornecer recursos. GitOps
Reconciliação contínua	Essa ferramenta compara constantemente o estado real do cluster com o estado desejado definido no Git. Ele corrige automaticamente

Área	Capacidades da ferramenta
	qualquer desvio detectado entre esses estados.
Gerenciamento de vários clusters	O Weave GitOps suporta o gerenciamento de vários clusters Kubernetes a partir de um único plano de controle. Ele permite a implantaç ão consistente de aplicativos em diferentes ambientes.
Política como código	O Weave GitOps incorpora o conceito de política como código para aplicar as regras de segurança e conformidade. As políticas são controladas por versão junto com o código do aplicativo e as definições de infraestrutura.
Entrega progressiva	Essa ferramenta oferece suporte a estratégias avançadas de implantação, como lançament os e blue/green implantações canary. Ele se integra ao Flagger para entrega automatizada e progressiva.
Observabilidade e painéis	O Weave GitOps fornece painéis integrado s para monitorar o estado dos aplicativos e clusters. Ele oferece informações sobre os processos de reconciliação e a integridade do cluster.
Seguro por design	A ferramenta implementa as melhores práticas de segurança, incluindo integração com RBAC e gerenciamento de segredos. Ele oferece suporte a vários métodos de autenticação e se integra aos provedores de identidade corporati va.

Área	Capacidades da ferramenta
Extensibilidade e integração	A ferramenta foi projetada para funcionar com uma ampla variedade de ferramentas nativas da nuvem. Ele suporta ferramentas populares como Flux, Helm e Kustomize.
Plataformas de autoatendimento para desenvolvedores	O Weave GitOps permite a criação de plataformas de autoatendimento para desenvolvedores. Ele fornece modelos e grades de proteção para implantação de aplicativos.
GitOps Automation	A ferramenta automatiza muitos aspectos do GitOps fluxo de trabalho, incluindo a geração de pull requests para atualizações.
Pipelines de entrega contínua	Ele se integra aos CI/CD sistemas para criar canais end-to-end de entrega.
Auditoria e conformidade	O Weave DevOps fornece uma trilha de auditoria completa de todas as mudanças e ações. Ele ajuda você a atender aos requisito s de conformidade por meio de controle de versão e processos automatizados.
Escalabilidade	A ferramenta foi projetada para escalar desde pequenos projetos até grandes implantações de nível corporativo.
Colaboração em equipe	O Weave GitOps facilita a colaboração entre as equipes de desenvolvimento e operações por meio de fluxos de trabalho baseados em Git.
GitOps como um serviço	Essa ferramenta é oferecida GitOps como um serviço gerenciado, o que simplifica a adoção e o gerenciamento.

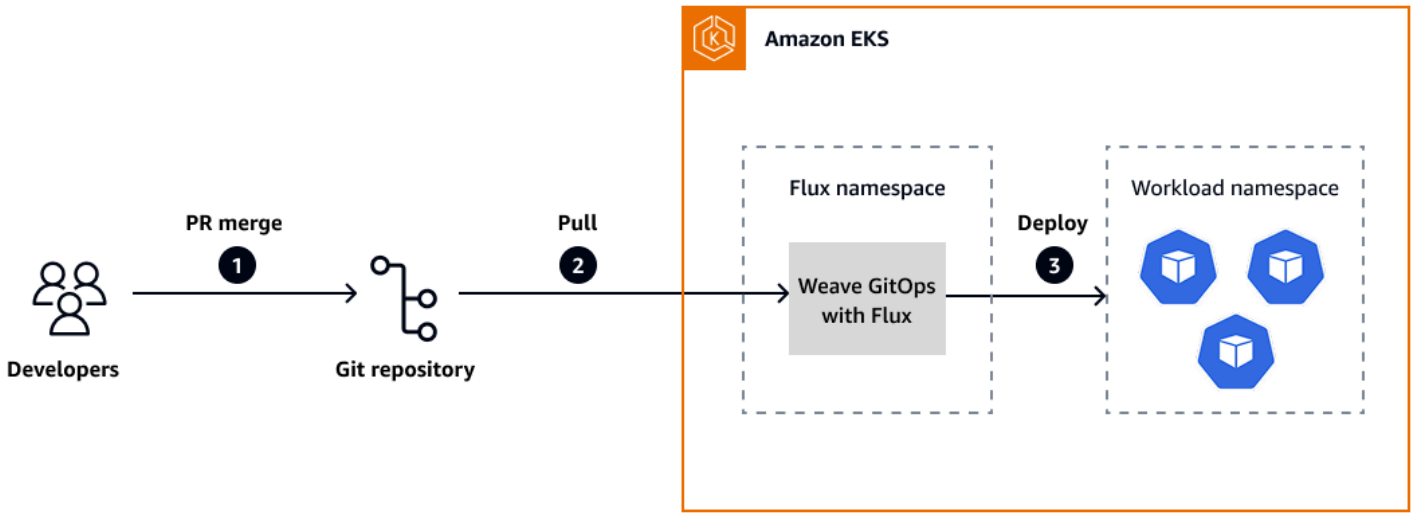
Área	Capacidades da ferramenta
Suporte híbrido e multinuvem	O Weave GitOps permite o gerenciamento consistente em diferentes provedores de nuvem e ambientes locais.
Segurança contínua	A ferramenta integra a verificação de segurança e a aplicação de políticas em todo o processo de implantação.

O Weave GitOps implementa esses princípios para fornecer uma GitOps solução abrangente que vai além da automação básica de implantação. O objetivo é criar um modelo operacional completo para aplicativos nativos da nuvem que se concentre em segurança, escalabilidade e facilidade de uso. Ao aderir a esses GitOps princípios, o Weave GitOps ajuda as organizações a obter um gerenciamento consistente, auditável e eficiente de seus ambientes Kubernetes em vários clusters e provedores de nuvem.

Para obter mais informações, consulte a GitOpsdocumentação do Weave.

Arquitetura

O diagrama a seguir ilustra um fluxo de trabalho de CD GitOps orientado por CD que usa o Weave GitOps em um cluster EKS. Para obter informações detalhadas, consulte o GitOpsrepositório Weave.



em que:

- Etapa 1: mesclagem do Pull Request (PR). Um desenvolvedor confirma as alterações nos manifestos do Kubernetes ou nos gráficos do Helm que são armazenados em um repositório Git. Quando o PR é revisado e incorporado à ramificação principal, o estado desejado do aplicativo é atualizado no controle de origem.
- Etapa 2: Sincronização do repositório. O Weave GitOps é executado no namespace Flux no cluster EKS e monitora continuamente o repositório Git configurado. Quando detecta alterações, ele obtém as atualizações mais recentes para reconciliar o estado declarado.
- Etapa 3: Implantação no namespace de destino. O Weave GitOps compara o estado desejado do Git com o estado ativo no cluster. Em seguida, ele aplica as alterações necessárias ao namespace da carga de trabalho de destino para que o aplicativo seja implantado ou atualizado adequadamente.

Jenkins X

O Jenkins X é uma CI/CD plataforma de código aberto nativa da nuvem que implementa GitOps princípios para ambientes Kubernetes. Embora o Jenkins X não seja exclusivamente uma GitOps ferramenta como o Argo CD ou o Flux, ele incorpora GitOps práticas em seus fluxos de trabalho.

GitOps apoio

Área	Capacidades da ferramenta
Fluxo de trabalho centrado no Git	O Jenkins X usa repositórios Git como a principal fonte confiável tanto para o código quanto para a configuração do aplicativo. Todas as alterações nos aplicativos e na infraestrutura são feitas por meio do Git.
Ambiente como código (EaC)	Ambientes (como preparação e produção) são definidos como código nos repositórios Git. Isso permite o controle de versão e a revisão das configurações do ambiente.
CI/CD Pipelines automatizados	O Jenkins X configura automaticamente CI/CD pipelines para projetos. Esses pipelines são

Área	Capacidades da ferramenta
	definidos como código (pipeline como código) e armazenados no Git.
Nativo do Kubernetes	O Jenkins X foi criado especificamente para ambientes Kubernetes. Ele usa recursos do Kubernetes e definições de recursos personali zadas (). CRDs
Ambientes de pré-visualização	O Jenkins X cria automaticamente ambientes temporários para pull requests. Ele permite a fácil revisão e o teste das alterações antes das mesclagens.
Promoção entre ambientes	O Jenkins X usa uma GitOps abordagem para promover aplicativos entre ambientes (por exemplo, da preparação à produção). As promoções são gerenciadas usando pull requests para garantir processos adequados de revisão e aprovação.
Gerenciamento de gráficos do Helm	O Jenkins X usa gráficos do Helm para empacotar e implantar aplicativos. Os gráficos são controlados por versão nos repositórios Git.
Controle de versão automatizado	O Jenkins X gerencia automaticamente o controle de versões de aplicativos e versões. Ele usa versionamento semântico e gera notas de lançamento.
ChatOps integração	O Jenkins X oferece suporte ChatOps para operações comuns. Isso se alinha aos GitOps princípios de automação e colaboração.

Área	Capacidades da ferramenta
Extensibilidade	Essa ferramenta fornece um sistema de plug-ins para estender a funcionalidade. Ele permite a integração com várias ferramentas nativas da nuvem.
Infraestrutura como código (IaC)	O Jenkins X é compatível com Terraform, CloudFormation, AWS Cloud Development Kit (AWS CDK), e outras ferramentas de IaC para definir e gerenciar a infraestrutura. As definiçõe s de infraestrutura são controladas por versão junto com o código do aplicativo.
Reversões automatizadas	O Jenkins X oferece suporte a reversões automatizadas se problemas forem detectados após a implantação.
Gerenciamento de segredos	A ferramenta se integra a soluções externas de gerenciamento de segredos para lidar com informações confidenciais com segurança.
Observabilidade	O Jenkins X fornece integração com ferrament as de monitoramento e registro para observabi lidade.
Suporte multinuvem	O Jenkins X foi projetado para funcionar em diferentes provedores de nuvem e ambientes locais.
Colaboração em equipe	Essa ferramenta incentiva a colaboração por meio de fluxos de trabalho e pull requests baseados em Git.
Feedback contínuo	A ferramenta fornece feedback rápido sobre as mudanças por meio de ambientes automatiz ados de teste e pré-visualização.

Área	Capacidades da ferramenta
DevOps melhores práticas	O Jenkins X implementa as DevOps melhores práticas por padrão, incluindo GitOps princípio s.
Configuração declarativa	A ferramenta usa configurações declarativas para definir aplicativos e ambientes.
Atualizações automatizadas	O Jenkins X fornece ferramentas para automatizar as atualizações da própria plataforma Jenkins X.

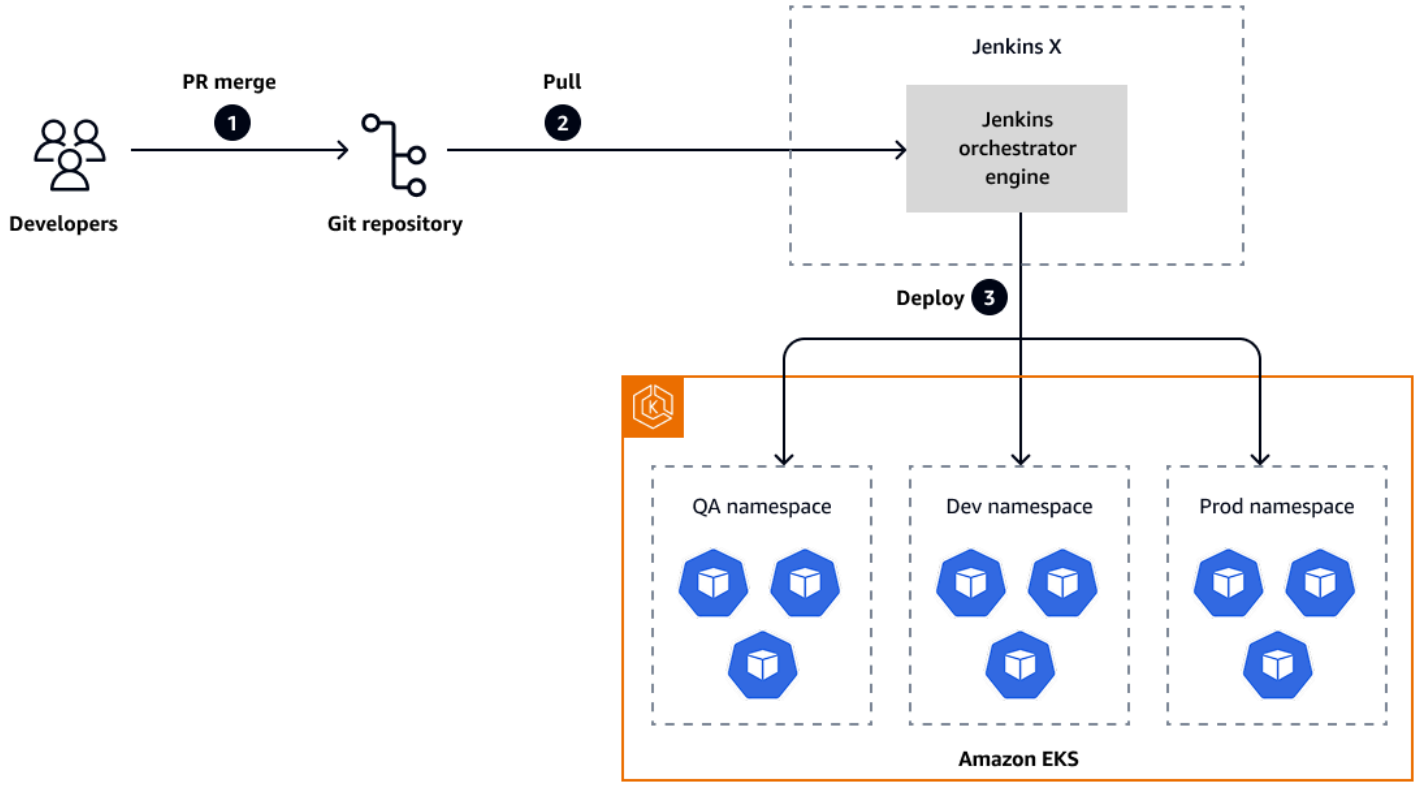
O Jenkins X implementa esses GitOps princípios para criar uma solução abrangente de CI/CD para Kubernetes. O objetivo é automatizar e agilizar todo o processo de entrega de software, desde a confirmação do código até a implantação da produção, ao mesmo tempo em que segue as práticas. GitOps Ao fazer isso, ajuda as equipes a obter implantações mais rápidas, confiáveis e consistentes em ambientes nativos da nuvem.

A principal diferença entre o Jenkins X e ferramentas como o Argo CD ou o Flux é que o Jenkins X fornece uma CI/CD solução mais abrangente, incluindo automação de construção e gerenciamento de pipeline, ao mesmo tempo em que incorpora GitOps princípios para implantação e gerenciamento do ambiente. Isso o torna particularmente adequado para equipes que precisam de uma all-in-one solução que abranja os aspectos de CI e CD em uma única GitOps estrutura.

Para obter mais informações, consulte a documentação do Jenkins X.

Arquitetura

O diagrama a seguir ilustra um fluxo de trabalho GitOps controlado por CD que usa o Jenkins X. Para obter informações detalhadas, consulte a documentação do Jenkins X.



em que:

- Etapa 1: mesclagem do Pull Request (PR). Um desenvolvedor cria uma pull request que inclui alterações nos manifestos do Kubernetes, nos gráficos do Helm ou no código do aplicativo armazenado em um repositório Git. Após análise e aprovação, o PR é incorporado à filial principal e atualiza o estado desejado no controle de origem.
- Etapa 2: Sincronização do repositório. O Jenkins X aciona automaticamente um CI/CD pipeline quando detecta a alteração. O pipeline cria, testa e promove o aplicativo em diferentes ambientes (por exemplo, preparação e produção) usando GitOps princípios.
- Etapa 3: Implantação nos namespaces de destino. O Jenkins X atualiza os repositórios do ambiente (preparação e produção) com as novas versões do aplicativo. O cluster reconcilia automaticamente as alterações retirando os manifestos mais recentes do Git e implantando o aplicativo nos namespaces apropriados.

GitLab CI/CD

GitLab CI/CD is an integrated part of the GitLab platform that provides continuous integration, delivery, and deployment capabilities. Although GitLab CI/CDnão é exclusivamente uma GitOps

ferramenta, você pode configurá-la para implementar GitOps princípios, especialmente quando você a usa para implantações do Kubernetes.

GitOps apoio

Área	Capacidades da ferramenta
Git como a única fonte da verdade	GitLab O CI/CD usa repositórios Git para armazenar o código do aplicativo e as configurações de infraestrutura. Todas as alterações no sistema são feitas por meio do Git, o que garante um histórico completo e uma trilha de auditoria.
Configuração declarativa	GitLab Os pipelines de CI/CD são definidos em um arquivo.gitlab-ci.yml, que é uma configura ção declarativa armazenada no repositório Git. Manifestos do Kubernetes, gráficos do Helm ou outros arquivos de infraestrutura como código (IaC) podem ser armazenados no mesmo repositório para definir o estado desejado da infraestrutura.
Pipelines automatizados	GitLab O CI/CD aciona automaticamente os pipelines quando as alterações são enviadas para o repositório. Esses pipelines podem incluir estágios para criar, testar e implantar aplicativos.
Integração com o Kubernetes	GitLab O CI/CD fornece integração nativa com o Kubernetes e oferece suporte a implantações no estilo GitOps -em clusters do Kubernetes. Ele pode criar e gerenciar automaticamente os recursos do Kubernetes com base na configura ção no Git.
Gestão ambiental	GitLab O CI/CD suporta a definição de vários ambientes (como preparação e produção)

Área	Capacidades da ferramenta
	como código. As implantações nesses ambientes podem ser automatizadas ou podem exigir aprovação manual, em conformidade com GitOps as práticas.
Analise os aplicativos	GitLab pode criar automaticamente ambientes temporários para solicitações de mesclagem , semelhantes aos ambientes de visualização em outras GitOps ferramentas. Isso facilita a revisão e o teste das alterações antes das mesclagens.
Implantação contínua	GitLab O CI/CD pode ser configurado para implantar automaticamente as alterações nos clusters do Kubernetes quando as alterações são mescladas em filiais específicas.
IaC	GitLab O CI/CD oferece suporte à integraçã o com ferramentas como o Terraform e CloudFormation ao gerenciamento da infraestr utura como código. As definições de infraestr utura podem ser controladas por versão junto com o código do aplicativo.
Observabilidade e monitoramento	GitLab O CI/CD fornece recursos integrados de monitoramento e observabilidade, incluindo integração com o Prometheus e o Grafana.
Escaneamento de segurança	GitLab CI/CD includes built-in security scanning tools that can be integrated into the CI/CDpipe line para reforçar a segurança como parte do GitOps fluxo de trabalho.

Área	Capacidades da ferramenta
Registro de contêiner	GitLab O CI/CD inclui um registro de contêiner integrado para integração perfeita do gerenciamento de imagens de contêineres no fluxo de trabalho. GitOps
Automático DevOps	O DevOps recurso Auto em GitLab CI/CD can automatically configure CI/CD pipelines que seguem GitOps os princípios para implantaç ões do Kubernetes.
Fluxos de trabalho de aprovação	GitLab O CI/CD suporta processos de aprovação para implantações, que fornecem promoções controladas entre ambientes.
Gerenciamento de segredos	GitLab CI/CD provides features to securely manage and use secrets within CI/CDoleo dutos.
Versionamento e lançamentos	GitLab CI/CD supports automatic versioning and release management as part of the CI/ CDprocesso.
Reversões	GitLab O CI/CD permite reversões fáceis para versões anteriores se problemas forem detectados após a implantação.
Logs de auditoria	GitLab O CI/CD fornece registros de auditoria abrangentes para todas as ações para apoiar o aspecto de rastreabilidade do. GitOps
Pipelines de vários projetos	GitLab O CI/CD suporta GitOps fluxos de trabalho complexos que abrangem vários projetos ou repositórios.

Área	Capacidades da ferramenta
ChatOps	GitLab O CI/CD suporta ChatOps integrações, que fornecem colaboração e operações por meio de interfaces de bate-papo.
Gerenciamento de clusters Kubernetes	GitLab O CI/CD fornece recursos para gerenciar clusters Kubernetes diretamente da interface. GitLab

Porém GitLab CI/CD is not exclusively designed for GitOps, it can be used effectively to implement GitOps practices, especially for teams that already use GitLab as their primary development platform. Its integrated approach, which combines source control, CI/CD, o gerenciamento do Kubernetes o torna uma ferramenta poderosa para implementar fluxos de trabalho. GitOps

A principal diferença entre os GitLab CI/CD and dedicated GitOps tools such as Argo CD or Flux is that GitLab provides a more comprehensive platform that includes source control management, issue tracking, and other development tools along with its CI/CD recursos. Isso o torna particularmente adequado para equipes que precisam de uma all-in-one solução que possa implementar GitOps práticas em um sistema de desenvolvimento mais amplo.

Para obter mais informações sobre o GitLab CI/CD e sua arquitetura, consulte a documentação do GitLab CI/CD.

Spinnaker

Embora o Spinnaker não tenha sido projetado exclusivamente como uma GitOps ferramenta, você pode configurá-lo para implementar GitOps princípios, especialmente quando você o usa para implantações nativas da nuvem e do Kubernetes.

GitOps apoio

Área	Capacidades da ferramenta
Configuração declarativa	O Spinnaker usa definições declarativas de pipeline, que normalmente são armazenad as como arquivos JSON ou YAML. Essas

Área	Capacidades da ferramenta
	definições de pipeline podem ser controladas por versão nos repositórios Git, de acordo com as práticas. GitOps
IaC	O Spinnaker suporta a definição de configura ções de infraestrutura e implantação como código. Essas definições podem ser armazenadas nos repositórios Git e podem servir como a única fonte confiável.
Implantações em várias nuvens	O Spinnaker foi projetado para funcionar em vários provedores de nuvem e clusters Kubernetes. Ele permite GitOps práticas consistentes em diversos ambientes.
Pipeline como código	Os pipelines do Spinnaker podem ser definidos como código e armazenados em repositór ios Git. Isso permite o controle de versão e a revisão dos processos de implantação.
Implantações automatizadas	Você pode configurar o Spinnaker para iniciar automaticamente as implantações com base nas alterações nos repositórios Git. A ferrament a oferece suporte a práticas de implantação contínua que são fundamentais para GitOps o.
Infraestrutura imutável	A Spinnaker promove o uso de infraestrutura imutável, que é um conceito-chave em. GitOps Ela incentiva a implantação de novas instância s em vez de modificar as existentes.
Reversões e controle de versão	O Spinnaker fornece recursos robustos de reversão e rápida reversão aos bons estados anteriores conhecidos. Ele suporta o controle de versões de implantações, em alinhamento com GitOps os princípios de rastreabilidade.

Área	Capacidades da ferramenta
Fluxos de trabalho de aprovação	O Spinnaker inclui estágios de julgamento manual em pipelines para apoiar promoções controladas entre ambientes. Isso suporta GitOps práticas de separação entre implantaç ões e versões.
Canary e implantações blue/green	O Spinnaker oferece suporte a estratégias avançadas de implantação que se alinham GitOps às práticas de lançamentos seguros e controlados.
Integração com sistemas de controle de versão	O Spinnaker pode se integrar a vários provedores de Git para iniciar pipelines com base nos eventos do repositório.
Integração com o Kubernetes	O Spinnaker fornece suporte nativo para o Kubernetes e oferece suporte ao gerenciam ento em estilo dos recursos do Kubernetes. GitOps
Gerenciamento de Artefatos	O Spinnaker oferece suporte ao gerenciam ento de artefatos e ao controle de versões, que são cruciais para manter um fluxo de trabalho. GitOps
Observabilidade e monitoramento	O Spinnaker oferece integração com ferrament as de monitoramento para apoiar o aspecto de observabilidade do. GitOps
Trilha de auditoria	O Spinnaker fornece registros e histórico detalhados de implantação, que apoiam o princípio de auditabilidade do. GitOps

Área	Capacidades da ferramenta
Regras de controle de acesso com base em função (RBAC)	Essa ferramenta implementa o RBAC para um controle refinado sobre quem pode realizar quais ações, de acordo com as práticas de segurança. GitOps
Modelagem e parametrização	O Spinnaker oferece suporte à modelagem em definições de pipeline para permitir implantaç ões reutilizáveis e parametrizadas.
Promoção do meio ambiente	O Spinnaker facilita a promoção de aplicações entre ambientes (por exemplo, da preparação à produção) de forma controlada.
Integração com ferramentas de CI	O Spinnaker pode se integrar a várias ferramentas de integração contínua (CI) para fornecer um CI/CD pipeline completo que segue os princípios. GitOps
Estágios e extensões personalizados	Essa ferramenta oferece suporte a estágios e extensões personalizados, para que as equipes possam implementar GitOps fluxos de trabalho adaptados às suas necessidades.
Gerenciamento centralizado	O Spinnaker fornece uma plataforma centraliz ada para gerenciar implantações em vários ambientes e provedores de nuvem.

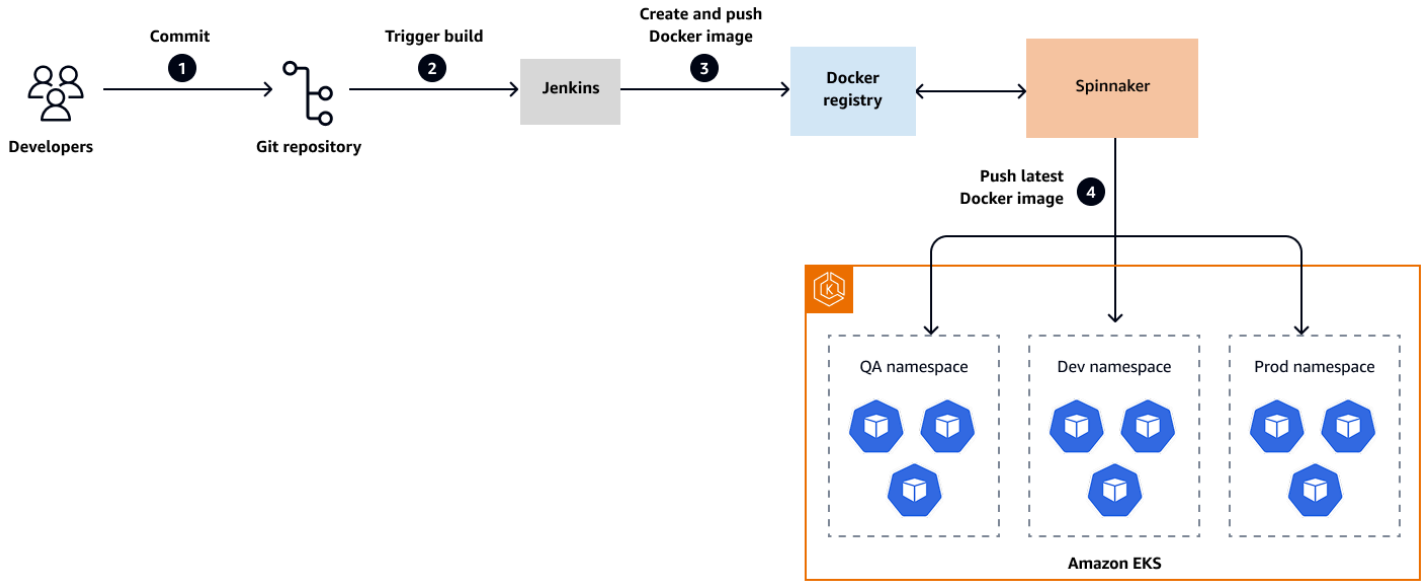
Embora o Spinnaker não seja comercializado principalmente como uma GitOps ferramenta, sua flexibilidade e seu conjunto robusto de recursos o tornam capaz de implementar GitOps fluxos de trabalho, especialmente em ambientes complexos com várias nuvens. A principal diferença entre o Spinnaker e GitOps ferramentas dedicadas, como o Argo CD ou o Flux, é que o Spinnaker oferece uma plataforma de entrega contínua mais abrangente com estratégias avançadas de implantação e suporte a várias nuvens.

A força da Spinnaker está em sua capacidade de lidar com cenários complexos de implantação em vários provedores de nuvem e em seu suporte a estratégias avançadas de implantação. Quando o Spinnaker está configurado corretamente, ele pode implementar GitOps princípios de forma eficaz. Isso o torna uma ferramenta poderosa para organizações que desejam adotar GitOps práticas em ambientes diversos e complexos.

Para obter mais informações, consulte a documentação do Spinnaker.

Arquitetura

O diagrama a seguir ilustra um fluxo GitOps de trabalho baseado em CD que usa o Spinnaker e o Jenkins X. Para obter informações detalhadas, consulte a documentação do Spinnaker.



em que:

- Etapa 1: confirmação do código. Os desenvolvedores confirmam as alterações no código do aplicativo em um repositório Git. Essas mudanças podem incluir atualizações no próprio aplicativo, nos Dockerfiles ou nos manifestos do Kubernetes.
- Etapa 2: construção do Jenkins e criação da imagem. O Jenkins é acionado automaticamente pelo repositório Git por meio de um webhook ou enquete. O Jenkins cria o aplicativo, cria uma imagem do Docker e envia a imagem criada para um registro do Docker configurado (como Amazon ECR ou Docker Hub).
- Etapa 3: monitoramento de imagem do Spinnaker e acionamento do pipeline. O Spinnaker monitora continuamente o registro do Docker em busca de novas imagens. Quando uma nova

versão da imagem é detectada, o Spinnaker aciona automaticamente um pipeline para iniciar o processo de implantação.

- Etapa 4: Implantação nos namespaces de destino. A Spinnaker implanta a nova imagem do Docker no Amazon EKS. Com base nas configurações do pipeline, a imagem é implantada nos namespaces de destino no cluster. O Spinnaker garante que a versão mais recente do aplicativo seja implantada enquanto segue estratégias de implantação definidas, como blue/green implantações canárias.

Frota de fazendeiros

O Rancher Fleet é uma GitOps-at-scale solução projetada especificamente para gerenciar vários clusters Kubernetes. Ele segue rigorosamente GitOps os princípios enquanto se concentra na escalabilidade e no gerenciamento de vários clusters.

GitOps apoio

Área	Capacidades da ferramenta
Git como a única fonte da verdade	O Fleet usa repositórios Git como fonte autorizada para definir o estado desejado de aplicativos e recursos em vários clusters. Todas as configurações, incluindo manifesto s do Kubernetes, gráficos do Helm e recursos personalizados, são armazenadas no Git.
Configuração declarativa	O Fleet trabalha com descrições declarativas do estado desejado para aplicativos e recursos. Eles podem ser YAML brutos do Kubernete s, gráficos do Helm, arquivos Kustomize ou recursos personalizados específicos do Fleet.
Sincronização automatizada	O Fleet monitora continuamente os repositór ios Git em busca de alterações. Ele aplica automaticamente as alterações aos clusters de destino quando detecta diferenças entre o estado Git e o estado do cluster.

Área	Capacidades da ferramenta
Gerenciamento de vários clusters	O Fleet foi projetado especificamente para gerenciar implantações em vários clusters Kubernetes. Ele pode lidar com milhares de clusters a partir de um único plano de controle.
Arquitetura nativa do Kubernetes	O Fleet é construído como um conjunto de recursos e controladores personalizados do Kubernetes. Ele usa os mecanismos de extensão no Kubernetes para operações. GitOps
Reconciliação contínua	O Fleet compara constantemente o estado real dos clusters com o estado desejado definido no Git. Ele corrige automaticamente qualquer desvio detectado entre esses estados.
Agrupamento e segmentação de clusters	O Fleet permite agrupar clusters e direciona r implantações para grupos específicos ou clusters individuais. Ele oferece suporte à implantação consistente de aplicativos em diferentes ambientes e tipos de cluster.
Configurações em camadas	O Fleet suporta configurações em camadas, que fornecem configurações básicas com sobreposições específicas do ambiente. Isso se alinha às GitOps práticas de gerenciamento eficiente de vários ambientes.
Integração com o Helm	O Fleet fornece suporte nativo para gráficos do Helm e fornece fácil gerenciamento de aplicativos complexos. Ele pode criar versões e gerenciar versões do Helm por meio de GitOps fluxos de trabalho.

Área	Capacidades da ferramenta
Definições personalizadas de recursos (CRDs)	O Fleet usa recursos personalizados, como o GitRepo Bundle, para definir implantaç ões. Eles CRDs fornecem uma forma nativa do Kubernetes de definir fluxos de trabalho. GitOps
Segurança e RBAC	O Fleet se integra ao Kubernetes RBAC para controle de acesso. Ele oferece suporte ao gerenciamento seguro de informações e credenciais confidenciais.
Observabilidade	O Fleet fornece informações de status sobre o estado de sincronização de clusters e aplicativos. Ele oferece insights sobre GitOps processos em toda a frota de clusters.
Escalabilidade	O Fleet foi projetado para ser escalado para gerenciar milhares de clusters com eficiênci a. Ele oferece suporte a GitOps operações de grande escala em ambientes corporativos.
Gerenciar dependências	Você pode definir dependências entre diferente s recursos e aplicativos. A frota garante que a ordem correta das operações seja seguida em implantações complexas.
Personalização e extensibilidade	O Fleet oferece suporte a scripts personali zados e ganchos de ciclo de vida para personalização avançada de implantações. Ele permite a integração com ferramentas e fluxos de trabalho existentes.

Área	Capacidades da ferramenta
Suporte off-line e sem fio	O Fleet pode operar em ambientes com pouca ou nenhuma conectividade com a Internet. Ele suporta GitOps fluxos de trabalho em ambientes regulamentados ou de alta segurança.
Lançamentos progressivos	O Fleet suporta implementações graduais em clusters, o que permite estratégias de implantação controladas e graduais.
Interface de gerenciamento unificada	O Fleet fornece uma interface única para gerenciar GitOps fluxos de trabalho em todos os clusters. Ele simplifica as operações em ambientes complexos de vários clusters.
Integração com outras ferramentas do Rancher	O Fleet se integra a outras ferramentas do Rancher para fornecer uma solução abrangent e de gerenciamento de Kubernetes.
Trilha de auditoria e conformidade	O Fleet mantém uma trilha de auditoria clara de todas as mudanças e implantações. Ele ajuda você a atender aos requisitos de conformid ade por meio de operações baseadas em Git e controladas por versão.

O Rancher Fleet implementa esses GitOps princípios com um forte foco na escalabilidade e no gerenciamento de vários clusters. Seu design é particularmente adequado para organizações que gerenciam um grande número de clusters Kubernetes em diferentes ambientes, data centers ou provedores de nuvem.

O principal diferencial do Fleet é sua capacidade de lidar GitOps em grande escala. Esse recurso o torna especialmente valioso para grandes empresas ou provedores de serviços gerenciados que gerenciam vários clusters. Ferramentas como Argo CD ou Flux são frequentemente usadas para gerenciamento de clusters individuais, enquanto o Fleet é projetado para gerenciar uma grande frota de GitOps clusters.

Ao aderir a esses GitOps princípios, o Rancher Fleet fornece uma solução para organizações que desejam implementar o gerenciamento consistente, escalável e automatizado de aplicativos e recursos em um ambiente Kubernetes diversificado e de grande escala.

Para obter mais informações, consulte a documentação do Fleet.

Arquitetura

Para obter informações sobre arquitetura e fluxo de trabalho, consulte o repositório Fleet.

Codefresh

O Codefresh é uma CI/CD plataforma moderna que suporta GitOps princípios, especialmente para implantações de Kubernetes. O Codefresh oferece um conjunto abrangente de CI/CD recursos e seus GitOps recursos são notáveis.

GitOps apoio

Área	Capacidades da ferramenta
Git como a única fonte da verdade	O Codefresh usa repositórios Git como fonte autorizada para código de aplicativo, definiçõe s de infraestrutura e configurações de pipeline. Todas as alterações no sistema são feitas por meio do Git, o que garante um histórico completo e uma trilha de auditoria.
Configuração declarativa	O Codefresh suporta definições declarativas de pipeline usando arquivos YAML armazenados no Git. Os manifestos do Kubernetes, gráficos do Helm, CloudFormation modelos e outros arquivos IaC podem ser controlados por versão nos mesmos repositórios.
GitOps painel	O Codefresh fornece um GitOps painel dedicado para visualizar e gerenciar fluxos de trabalho. GitOps Ele oferece uma visão clara

Área	Capacidades da ferramenta
	do status de sincronização entre o Git e os estados do cluster.
Sincronização automatizada	O Codefresh monitora continuamente os repositórios Git em busca de alterações. Ele inicia automaticamente os pipelines para aplicar alterações nos ambientes de destino quando detecta diferenças.
Integração com o Kubernetes	O Codefresh oferece integração profunda com o Kubernetes para oferecer suporte GitOps a implantações no estilo -em vários clusters. Ele oferece suporte a vários recursos do Kubernete s e definições de recursos personalizadas (). CRDs
Gestão ambiental	Você pode definir e gerenciar vários ambientes (como desenvolvimento, preparação e produção) como código. O Codefresh apóia a promoção entre ambientes usando práticas. GitOps
Integração com Argo CD	O Codefresh se integra ao Argo CD para recursos aprimorados. GitOps Ele combina seus recursos de CI com os pontos fortes de CD do Argo CD para fornecer uma solução completa GitOps.
Suporte de leme	O Codefresh suporta gráficos Helm e fornece fácil gerenciamento de aplicações complexas por meio de. GitOps Ele também oferece controle de versão e promoção do Helm Chart.

Área	Capacidades da ferramenta
Entrega progressiva	O Codefresh oferece suporte a estratégias avançadas de implantação, como canary e implantações. blue/green Você pode implement ar e gerenciar essas estratégias por meio de GitOps fluxos de trabalho.
Reversões e controle de versão	O Codefresh permite reversões fáceis para versões anteriores se problemas forem detectados após a implantação. Ele mantém o controle de versão da implantação para rastreabilidade.
Fluxos de trabalho de aprovação	O Codefresh suporta processos de aprovação manuais e automatizados para implantações. Ele permite promoções controladas entre ambientes, em conformidade com GitOps as práticas.
IaC	O Codefresh oferece suporte à integração com ferramentas de IaC, como o Terraform . CloudFormation Ele permite o controle de versão das definições de infraestrutura junto com o código do aplicativo.
Observabilidade e monitoramento	O Codefresh fornece recursos integrados de monitoramento e observabilidade. Ele também oferece integrações com ferramentas externas de monitoramento para melhorar a visibilidade.
Escaneamento de segurança	O Codefresh inclui recursos de verificação de segurança que podem ser integrados aos fluxos de trabalho. GitOps As verificações de segurança fazem parte do processo de implantação automatizada.

Área	Capacidades da ferramenta
Trilhas de auditoria	A Codefresh mantém registros de auditoria abrangentes para todas as ações e mudanças. Ele suporta os aspectos de rastreabilidade e conformidade do. GitOps
RBAC e controle de acesso	O Codefresh implementa o controle de acesso baseado em funções (RBAC) para gerenciam ento refinado de permissões. Isso ajuda a garantir GitOps operações seguras entre equipes e ambientes.
GitOps Automation	O Codefresh oferece recursos para automatizar vários aspectos dos GitOps fluxos de trabalho, incluindo criação e mesclagem de pull request (PR).
Implantações multinuvem e híbridas	O Codefresh oferece suporte a GitOps fluxos de trabalho em vários provedores de nuvem e ambientes locais.
Modelagem e parametrização	O Codefresh oferece suporte a modelos em configurações de pipeline e implantação. Isso permite fluxos de trabalho reutilizáveis e GitOps parametrizados.
Gerenciamento integrado de imagens	O Codefresh fornece recursos integrados de gerenciamento de imagens de contêineres. Ele integra construções e implantações de imagens em fluxos de trabalho. GitOps
GitOps para gerenciamento de segredos	O Codefresh oferece maneiras seguras de gerenciar segredos nos fluxos de trabalho. GitOps Ele se integra a soluções externas de gerenciamento de segredos.

Área	Capacidades da ferramenta
Recursos de colaboração	O Codefresh fornece recursos para colaboração em equipe dentro dos processos. GitOps Esses recursos incluem comentários, notificações e painéis compartilhados.

A abordagem Codefresh GitOps é notável por sua integração dos recursos de CI/CD com as práticas. GitOps O objetivo é fornecer uma plataforma abrangente que cubra todo o ciclo de vida da entrega de software, respeitando os princípios. GitOps

O principal diferencial do Codefresh na GitOps área é sua abordagem de plataforma unificada, que combina recursos de CI com CD e recursos. GitOps Isso o torna particularmente adequado para equipes que desejam uma all-in-one solução que possa lidar com CI/CD cenários complexos enquanto implementa GitOps práticas.

A Codefresh oferece uma plataforma para organizações que desejam adotar GitOps metodologias dentro de um CI/CD contexto mais amplo, especialmente quando trabalham com Kubernetes e tecnologias nativas da nuvem.

Para obter mais informações, consulte a documentação do Codefresh.

Pulumi

Pulumi é uma plataforma IaC que não foi projetada exclusivamente para. GitOps No entanto, ele pode ser usado de forma eficaz para implementar GitOps princípios, especialmente para infraestrutura em nuvem e implantações de Kubernetes.

GitOps apoio

Área	Capacidades da ferramenta
IaC	O Pulumi permite que você defina sua infraestrutura usando linguagens de programação de uso geral, como Python e Go. TypeScript Essa abordagem baseada em código se alinha à

Área	Capacidades da ferramenta
	GitOps ênfase em configurações declarativas e versionadas.
Git como a única fonte da verdade	O código de infraestrutura no Pulumi pode ser armazenado e controlado por versão nos repositórios Git. Isso garante que o Git sirva como a única fonte confiável para as definições de infraestrutura.
Estado declarativo desejado	Embora o Pulumi use linguagens de programaç ão, ele ainda descreve o estado desejado da infraestrutura de forma declarativa. O código define a aparência da infraestrutura, não o step-by-step processo para criá-la.
Sincronização automatizada	O Pulumi pode ser integrado a CI/CD pipelines para aplicar alterações automaticamente quando o código é atualizado no Git. Isso permite a implantação contínua de mudanças na infraestrutura, o que é um GitOps princípio fundamental.
Suporte para várias nuvens e Kubernetes	O Pulumi oferece suporte a uma ampla variedade de provedores de nuvem e Kubernetes, para que você possa seguir GitOps as práticas em diversos ambientes. A ferramenta permite o gerenciamento consisten te de recursos em diferentes plataformas.
Gerenciamento de estados	O Pulumi gerencia o estado da infraestrutura, que pode ser armazenada remotamente e com segurança. Esse gerenciamento de estado é crucial para GitOps as práticas e garante a consistência entre o estado definido e o estado real de sua infraestrutura.

Área	Capacidades da ferramenta
Detecção e reconciliação de desvios	O Pulumi pode detectar diferenças entre o estado desejado (em código) e o estado real da infraestrutura. Ele concilia essas diferença s de alinhamento com o GitOps princípio da reconciliação contínua.
Política como código	Você pode definir e aplicar políticas como código usando o CrossGuard Pulumi. Isso permite um gerenciamento de GitOps estilo controlado por versão das políticas de conformidade e segurança.
Gerenciamento de segredos	O Pulumi fornece maneiras seguras de gerenciar informações confidenciais dentro do código de infraestrutura. Ele suporta a integração com sistemas externos de gerenciamento de segredos, o que é crucial para as práticas GitOps de segurança.
Componentes modulares e reutilizáveis	O Pulumi suporta a criação de componentes e módulos reutilizáveis. Essa modularidade se alinha às GitOps práticas de gerenciamento de implantações complexas em vários ambientes.
Pré-visualizar e planejar	O Pulumi oferece a capacidade de visualizar as alterações antes de aplicá-las. Isso apóia o GitOps princípio de mudanças seguras e previsíveis na infraestrutura.
Reversões e histórico	O Pulumi mantém um histórico de implantaç ões e oferece suporte a reversões para estados anteriores. Isso se alinha aos GitOps princípios de rastreabilidade e reversibilidade.

Área	Capacidades da ferramenta
Entrega contínua para infraestrutura	O Pulumi pode ser integrado a CI/CD tubulaçõe s para entrega contínua de mudanças na infraestrutura. Ele suporta testes automatiz ados e validação do código de infraestrutura.
RBAC e controle de acesso	O Pulumi fornece controle de acesso baseado em funções para gerenciar quem pode fazer alterações na infraestrutura. Isso apoia as práticas GitOps de segurança e governança.
Observabilidade e registro	O Pulumi oferece recursos de registro e monitoramento para mudanças na infraestr utura. Esses recursos apoiam o aspecto de observabilidade das GitOps práticas.
Integração com outras ferramentas	O Pulumi pode se integrar com várias ferramentas na nuvem. Essa flexibilidade permite GitOps fluxos de trabalho abrangentes.
Gestão ambiental	O Pulumi suporta o gerenciamento de vários ambientes (desenvolvimento, preparação, produção) usando a mesma base de código com configurações diferentes. Isso se alinha às GitOps práticas de gerenciamento consistente de vários ambientes.
Gerenciar dependências	O Pulumi lida com dependências entre recursos e garante a ordem correta das operações. Isso é crucial para GitOps implantações complexas que envolvem componentes interdependentes.

Área	Capacidades da ferramenta
Provedores de recursos personalizados	O Pulumi permite que você crie provedores personalizados para gerenciar qualquer serviço orientado por API. Isso estende GitOps as práticas a uma ampla variedade de recursos além das ofertas de nuvem padrão.
Recursos de colaboração	O Pulumi suporta a colaboração em equipe por meio de controles compartilhados de estado e acesso. Isso facilita os GitOps fluxos de trabalho em ambientes de equipe.

Ao usar esses recursos do Pulumi, as organizações podem implementar GitOps práticas para sua infraestrutura, especialmente em cenários em que precisam de controle refinado ou lógica complexa, ou desejam gerenciar um conjunto diversificado de recursos locais e na nuvem em uma estrutura única e consistente.

A abordagem da Pulumi GitOps é única porque traz o poder e a flexibilidade das linguagens de programação de uso geral para o gerenciamento de infraestrutura, ao mesmo tempo em que segue os princípios. GitOps Isso pode ser particularmente vantajoso para equipes que preferem trabalhar com linguagens de programação familiares e desejam aplicar as melhores práticas de engenharia de software ao gerenciamento da infraestrutura.

O principal diferencial do Pulumi em GitOps é o uso de linguagens de programação padrão para definir a infraestrutura. GitOps As ferramentas tradicionais geralmente usam YAML ou linguagens específicas de domínio, enquanto o Pulumi permite uma lógica mais complexa, melhor reutilização de código e integração mais fácil com fluxos de trabalho de desenvolvimento existentes.

Para obter mais informações, consulte a documentação do Pulumi.

GitOps comparação de ferramentas

Aqui está uma comparação das nove GitOps ferramentas que foram discutidas nas seções anteriores. Ao escolher uma ferramenta, considere seus requisitos específicos, a infraestrutura existente, a experiência da equipe e o nível desejado de controle e personalização.

Facilidade de uso

- Argo CD, Flux e Rancher Fleet geralmente são mais fáceis de configurar.
- Spinnaker e Jenkins X têm curvas de aprendizado mais acentuadas.
- O Weave GitOps pode exigir mais configurações para recursos avançados.
- GitLab O CI/CD e o Codefresh oferecem experiências integradas.

Integração com o Kubernetes

- Argo CD, Flux e Rancher Fleet são muito centrados no Kubernetes.
- O Jenkins X e o Weave GitOps oferecem recursos mais DevOps amplos.
- As outras ferramentas oferecem suporte ao Kubernetes sem um foco exclusivo nele.

Capacidades de CI/CD

- Jenkins X, GitLab CI/CD, and Codefresh offer complete CI/CD soluções.
- Argo CD, Flux e Weave GitOps se concentram mais no aspecto de CD do fluxo de trabalho e geralmente exigem integração com ferramentas de CI separadas.

GitOps pureza

- Argo CD e Flux são ferramentas que se concentram especificamente em. GitOps
- As outras ferramentas incorporam GitOps princípios em graus variados.

Suporte multinuvem

- A Spinnaker e a Pulumi se destacam em cenários de várias nuvens.
- As outras ferramentas podem funcionar em várias nuvens, mas podem exigir configuração adicional.

Suporte a vários clusters

- Todas as ferramentas oferecem suporte a implantações em vários clusters.
- O Argo CD e o Weave GitOps têm recursos mais avançados de gerenciamento de vários clusters.

Integração

- A Flux tem um forte apoio da Cloud Native Computing Foundation (CNCF).
- O Argo CD tem uma comunidade grande e ativa.
- O Argo CD e o Flux têm uma forte integração com o Kubernetes.
- O Jenkins X usa o sistema Jenkins mais amplo.
- O Weave GitOps é mais novo, mas está crescendo com forte apoio comercial.
- GitLab O CI/CD se integra perfeitamente com o. GitLab
- O Rancher Fleet funciona bem dentro do sistema Rancher.

Comunidade e suporte

- O Flux tem um forte suporte de CNCF.
- Argo CD, GitLab, e Spinnaker têm grandes comunidades.
- O suporte comercial está disponível para a maioria das ferramentas.

Funcionalidades corporativas

- Por padrão, o Weave GitOps e o Jenkins X oferecem mais recursos voltados para empresas.
- O Argo CD e o Flux têm ofertas corporativas ou podem ser estendidos para uso corporativo.

Flexibilidade e extensibilidade

- O Flux é altamente modular e extensível.
- O Argo CD oferece boas opções de personalização.
- O Jenkins X é muito extensível, mas pode exigir mais esforço.
- O GitOps objetivo do Weave é fornecer uma solução completa com menos necessidade de extensibilidade.

Escalabilidade

- O Spinnaker e o GitLab CI/CD são conhecidos pela escalabilidade corporativa.
- O Argo CD e o Flux lidam bem com implantações de Kubernetes em grande escala.

Gerenciamento de infraestrutura

- A Pulumi se concentra no gerenciamento de infraestrutura.
- O Weave GitOps e o Flux oferecem bons recursos de IaC.

Suporte a modelos e linguagens de programação

- No Pulumi, você pode definir a infraestrutura usando linguagens de programação de uso geral, como Python, Go, TypeScript C# e Java. O uso de linguagens padrão pela Pulumi permite a integração do código de infraestrutura com fluxos de trabalho de desenvolvimento familiares, práticas de teste e lógica complexa.
- O Terraform usa a linguagem HashiCorp de configuração (HCL).
- CloudFormation usa modelos JSON e YAML.
- Argo CD, Flux, Rancher Fleet, Weave GitOps, Spinnaker e GitLab CI/CD gerenciam principalmente arquivos YAML ou de configuração declarativa.
- O Jenkins X gerencia YAML e pipelines baseados em scripts, mas não oferece nativamente programação de uso geral para IaC.

Casos de uso do Argo CD e do Flux

Esta seção se concentra em duas ferramentas, Argo CD e Flux, que fornecem funcionalidade pura GitOps. Nesse contexto, pure GitOps se refere a um modelo em que um repositório Git serve como a única fonte confiável para o estado desejado dos aplicativos e da infraestrutura. Todas as alterações são feitas por meio de commits do Git, e o sistema sincroniza automaticamente o ambiente ativo para corresponder ao estado definido no repositório. Nenhuma intervenção manual é necessária fora das operações do Git.

Considerações gerais

- Talvez você prefira usar o Argo CD em ambientes em que o gerenciamento visual e os fluxos de trabalho centrados em aplicativos são importantes.
- Você pode escolher o Flux se precisar de soluções mais leves, multilocação forte ou integração profunda com a rede mais ampla do Cloud Native Computing Foundations (CNCF).
- O Argo CD geralmente atrai equipes que estão migrando do CI/CD tradicional para devido à GitOps sua interface de usuário intuitiva.
- O Flux geralmente é preferido em ambientes nativos da nuvem, onde fluxos de trabalho baseados em CLI e práticas de IaC já estão estabelecidos.

Em última análise, a escolha entre o Argo CD e o Flux geralmente depende de suas necessidades organizacionais específicas, das ferramentas existentes e das preferências da equipe. Ambas as ferramentas são capazes de lidar com a maioria dos GitOps cenários, por isso recomendamos que você as avalie com base em seus casos de uso e requisitos específicos.

Casos de uso do Argo CD

Gerenciamento visual:

- Quando você precisa de uma interface de usuário fácil de usar para gerenciar implantações e visualizar estados de aplicativos.
- Para equipes que preferem uma interface gráfica para monitoramento e solução de problemas.

Abordagem centrada no aplicativo:

- Quando você quiser gerenciar implantações no nível do aplicativo em vez de gerenciar recursos individuais.
- Para organizações que estruturam suas implantações em torno de conceitos de aplicativos.

Gerenciamento de vários clusters:

- Quando gerenciar implantações em vários clusters é um requisito primário.
- Para ambientes complexos e distribuídos com muitos clusters.

Ondas de reversão e sincronização:

- Quando você precisa de um controle refinado sobre o processo de implantação, incluindo ondas de sincronização e intervenções manuais.
- Para cenários que exigem estratégias complexas de reversão.

Integração com ferramentas existentes:

- Quando você já está usando outras ferramentas no projeto Argo, como Argo Workflows e Argo Events.

Ambientes corporativos:

- Para grandes empresas que precisam de RBAC robusto e integração de login único por padrão.

Casos de uso do Flux

Implantações leves:

- Quando você precisa de uma solução mais leve e que consuma menos recursos GitOps.
- Para cenários de computação de ponta ou IoT em que os recursos podem ser restritos.

Atualizações automatizadas de imagens:

- Quando a detecção e a implantação automáticas de novas imagens de contêiner são um requisito fundamental.

- Para equipes que se concentram na implantação contínua com atualizações frequentes de imagens.

Multilocação:

- Quando é necessário um forte suporte multilocatário, especialmente em ambientes de cluster compartilhados.
- Para provedores de serviços ou grandes organizações que têm separações estritas entre equipes ou projetos.

IaC:

- Ao gerenciar aplicativos e infraestrutura por meio do mesmo GitOps fluxo de trabalho, é importante.
- Para equipes que investem fortemente no paradigma do IaC.

Integração com o Helm:

- Quando o uso extensivo de gráficos do Helm faz parte de sua estratégia de implantação.
- Para ambientes com implantações complexas baseadas em Helm.

Integração do projeto CNCF:

- Quando a integração estreita com outros projetos de CNCF é importante.
- Para organizações que se alinham às tecnologias e princípios da CNCF.

Arquitetura modular:

- Quando você precisa de flexibilidade para usar somente componentes específicos do GitOps kit de ferramentas.
- Para equipes que desejam criar GitOps fluxos de trabalho personalizados usando componentes modulares.

Entrega progressiva:

- Quando estratégias avançadas de implantação, como versões canárias ou A/B testes, são requisitos essenciais.

Comparação de recursos

Área	Argo CD	Fluxo
Support aos GitOps princípios fundamentais	 S	 Sim
Arquitetura	End-to-end aplicativo para implementar fluxos de trabalho do Kubernetes GitOps	Fornece Kubernetes CRDs e controladores para GitOps
Configuração	Simples	Complexo
Suporte de leme	 S	 Sim
Personalize o suporte	 S	 Sim
GUI integrada	CLI e interface de usuário web completa	CLI e interface web leve opcional
Suporte RBAC	Controle granular	RBAC nativo do Kubernetes
Suporte para multilocação e vários clusters	Excelente suporte para vários clusters	Excelente suporte para multilocação
Autenticação de login único	 S	 Sim

Área	Argo CD	Fluxo
Automação de sincronização	Capacidade de sincronizar janelas	Capacidade de definir intervalos de reconciliação
Sincronização parcial	 S	 Não
Processo de reconciliação	Suporta sincronizações manuais e automáticas. Várias estratégias diferentes estão disponíveis.	Suporta sincronizações manuais e automáticas.
Extensibilidade	Suporta plug-ins personali zados. Opções limitadas de personalização.	Suporta controlador personali zado. Boa extensibilidade e integrações de terceiros.
Apoio comunitário	Comunidade grande e ativa.	Comunidade menor, mas em crescimento.
Escalabilidade	Boa escalabilidade, mas limitada pela taxa de busca de dados da interface do usuário da web. A análise da comunidade sugere suporte para dezenas de milhares de aplicativos.	Guias claros para escalabil idade horizontal e vertical, até dezenas de milhares de aplicativos.

Práticas recomendadas para escolher uma GitOps ferramenta

Esta seção fornece considerações, dicas e melhores práticas para escolher uma GitOps ferramenta para seu cluster EKS. A escolha certa depende do seu contexto específico, dos requisitos e da estratégia de longo prazo. Geralmente, é benéfico realizar uma prova de conceito com suas melhores escolhas antes de tomar uma decisão final.

Avalie as necessidades e capacidades da sua organização:

- Considere o conjunto de habilidades atual da sua equipe e a disposição de aprender novas ferramentas.
- Avalie a complexidade do seu ambiente Amazon EKS. (Por exemplo, você está usando um único cluster ou vários clusters?)
- Determine seus requisitos específicos de conformidade, segurança e escalabilidade.

> **Prática recomendada**
>
> Crie um documento de requisitos detalhado que descreva os recursos necessários e os recursos úteis, mas não obrigatórios.

Avalie a maturidade e a adoção da ferramenta:

- Pesquise a maturidade de GitOps ferramentas em potencial e suas taxas de adoção no setor.
- Procure ferramentas que tenham um histórico comprovado em ambientes Amazon EKS.

> **Prática recomendada**
>
> Priorize ferramentas que foram amplamente adotadas e têm uma forte presença na rede Cloud Native Computing Foundation (CNCF).

Considere a integração com sua cadeia de ferramentas existente:

- Avalie o quão bem a GitOps ferramenta se integra ao seu CI/CD pipeline atual, às soluções de monitoramento e a outras ferramentas operacionais.
- Procure integrações nativas com, Serviços da AWS como IAM, Amazon ECR e. CloudWatch

> **Prática recomendada**
>
> Crie uma prova de conceito para testar os recursos de integração antes de tomar uma decisão final.

Avalie os recursos de segurança:

- Priorize ferramentas que tenham recursos robustos de controle de acesso baseado em funções (RBAC) e se integrem bem ao IAM.
- Procure recursos que ofereçam suporte ao gerenciamento seguro de segredos e à aplicação de políticas.

> **Prática recomendada**
>
> Escolha uma ferramenta que ofereça suporte a práticas de segurança GitOps baseadas em políticas, incluindo políticas como código e verificações automatizadas de conformidade.

Avalie a escalabilidade e o desempenho:

- Considere o desempenho da ferramenta com um grande número de aplicativos e clusters.
- Avalie seu impacto no desempenho do cluster e no consumo de recursos.

> **Prática recomendada**
>
> Realize testes de desempenho com cargas de trabalho semelhantes ao seu ambiente de produção para garantir que a ferramenta possa lidar com sua escala.

Considere o suporte a vários clusters e vários ambientes:

- Se você tem ou planeja ter vários clusters EKS, priorize ferramentas que tenham fortes recursos de gerenciamento de vários clusters.
- Procure recursos que ofereçam suporte a implantações consistentes em diferentes ambientes (como desenvolvimento, preparação e produção).

> **Prática recomendada**
>
> Escolha uma ferramenta que permita o gerenciamento centralizado de vários clusters, mantendo as configurações específicas do ambiente.

Avalie os recursos de observabilidade e monitoramento:

- Procure ferramentas que forneçam visibilidade clara do estado de suas implantações e da integridade do cluster.
- Considere o quão bem a ferramenta se integra às suas soluções existentes de monitoramento e registro.

> **Prática recomendada**
>
> Priorize ferramentas que ofereçam painéis personalizáveis e mecanismos de alerta para a detecção proativa de problemas.

Avalie a curva de aprendizado e a documentação:

- Avalie a qualidade e a abrangência da documentação da ferramenta.
- Considere a disponibilidade de recursos de treinamento e apoio da comunidade.

> **Prática recomendada**
>
> Escolha uma ferramenta que tenha documentação bem mantida, fóruns comunitários ativos e programas oficiais de treinamento ou certificações.

Considere o custo e a utilização de recursos:

- Avalie os custos diretos (como licenciamento e suporte) e os custos indiretos (como despesas gerais operacionais e custos de treinamento) da adoção da ferramenta.
- Avalie a eficiência da ferramenta em termos de consumo de recursos de computação e armazenamento.

> **Prática recomendada**
>
> Execute uma análise do custo total de propriedade (TCO) que inclua custos de curto e longo prazo.

Avalie as opções de flexibilidade e personalização:

- Procure ferramentas que permitam personalizar fluxos de trabalho para atender às suas necessidades específicas.
- Considere a extensibilidade da ferramenta por meio de plug-ins ou. APIs

> **Prática recomendada**
>
> Escolha uma ferramenta que equilibre a funcionalidade padrão com a capacidade de personalizar de acordo com seus requisitos exclusivos.

Avalie os recursos de entrega contínua e implantação progressiva:

- Procure ferramentas que ofereçam suporte a estratégias avançadas de implantação, como lançamentos e blue/green implantações canary.
- Avalie a facilidade de implementar e gerenciar essas estratégias.

> **Prática recomendada**
>
> Priorize ferramentas que oferecem suporte integrado para padrões de entrega progressivos para minimizar o risco em suas implantações.

Considere a dependência de um fornecedor e a portabilidade:

- Avalie as dependências da ferramenta em provedores ou tecnologias de nuvem específicos.
- Considere a facilidade de migrar para uma ferramenta diferente no futuro, se necessário.

> **Prática recomendada**
>
> Prefira ferramentas que usem padrões abertos e forneçam recursos de exportação para suas GitOps configurações.

Avalie o suporte e as extensões da comunidade:

- Veja o tamanho e a atividade da comunidade de usuários.
- Avalie a disponibilidade de integrações e plug-ins de terceiros.

> **Prática recomendada**
>
> Participe de fóruns comunitários ou grupos de usuários para obter experiências em primeira mão de outros usuários antes de tomar uma decisão.

Considere os requisitos de conformidade e auditoria:

- Avalie o quão bem a ferramenta atende às suas necessidades de conformidade, incluindo trilhas de auditoria e relatórios.
- Procure recursos que ajudem a manter e demonstrar a conformidade.

> **Prática recomendada**
>
> Escolha uma ferramenta que forneça registros de auditoria abrangentes e ofereça suporte à geração de relatórios de conformidade.

Avalie os recursos de reversão e recuperação de desastres:

- Avalie a facilidade e a confiabilidade dos mecanismos de reversão.
- Considere como a ferramenta suporta cenários de recuperação de desastres.

Prática recomendada

Teste minuciosamente os processos de reversão e recuperação como parte de sua avaliação.

Perguntas frequentes

P: Quais são as GitOps ferramentas mais populares para o Amazon EKS?

R: As GitOps ferramentas mais populares para o Amazon EKS incluem Argo CD, Flux, Jenkins X e CI/CD. GitLab Cada ferramenta tem pontos fortes, mas o Argo CD e o Flux são particularmente conceituados por sua abordagem nativa do Kubernetes e pelo forte apoio da comunidade.

P: Como GitOps melhorar o gerenciamento de clusters do EKS?

R: GitOps melhora o gerenciamento do cluster EKS fornecendo controle de versão para infraestrutura, implantações automatizadas, segurança aprimorada por meio de configurações declarativas, reversões mais fáceis e melhor auditabilidade. Também aprimora a colaboração e reduz o erro humano nas implantações.

P: Quais recursos principais devo procurar em uma GitOps ferramenta para o Amazon EKS?

R: Os principais recursos a serem procurados incluem: integração perfeita com o Amazon EKS, RBAC robusto, suporte a vários clusters, recursos de observabilidade, suporte para estratégias de entrega progressiva, escalabilidade e integração com IAM e Amazon ECR. Serviços da AWS

P: Como faço para garantir a segurança ao implementar GitOps no Amazon EKS?

R: Para garantir a segurança, escolha uma ferramenta que tenha uma forte integração do RBAC com o IAM, gerenciamento seguro de segredos, suporte para repositórios Git criptografados e a capacidade de implementar políticas de segurança como código. Além disso, verifique se a ferramenta fornece registros de auditoria abrangentes.

P: GitOps As ferramentas podem lidar com ambientes Amazon EKS de vários clusters?

R: Sim, GitOps ferramentas como Argo CD e Flux têm recursos robustos de gerenciamento de vários clusters. Eles permitem que você gerencie vários clusters EKS a partir de um único plano de controle, o que garante a consistência em todos os ambientes.

P: Como GitOps as ferramentas se integram aos pipelines de CI/CD existentes?

R: GitOps as ferramentas normalmente se integram aos pipelines de CI/CD existentes, atuando como o estágio de implantação do pipeline. Eles podem ser acionados por ferramentas de CI quando as alterações são enviadas para o repositório Git e automatizam o processo de implantação nos clusters EKS.

P: Quais são os desafios da implementação GitOps no Amazon EKS?

R: Os desafios comuns incluem gerenciar segredos com segurança, garantir controles de acesso adequados, lidar com aplicativos com estado, gerenciar a variação entre o Git e o estado do cluster e adaptar os fluxos de trabalho da equipe ao modelo. GitOps

P: Como GitOps as ferramentas lidam com reversões no Amazon EKS?

R: GitOps as ferramentas normalmente lidam com reversões revertendo para um commit anterior no repositório Git. Isso aciona automaticamente uma implantação do estado anterior em boas condições, o que resulta em reversões rápidas e confiáveis.

P: GitOps As ferramentas podem gerenciar os complementos e outros AWS recursos do Amazon EKS?

R: Muitas GitOps ferramentas podem gerenciar complementos e alguns AWS recursos do Amazon EKS, especialmente quando combinadas com ferramentas de IaC, como Terraform ou. CloudFormation No entanto, a extensão desse recurso pode variar; consulte a seção de GitOps ferramentas para obter informações específicas sobre cada ferramenta.

P: Como GitOps as ferramentas oferecem suporte aos requisitos de conformidade no Amazon EKS?

R: GitOps as ferramentas apoiam a conformidade fornecendo uma trilha de auditoria clara de todas as alterações, aplicando processos de aprovação, implementando políticas como código para verificações automatizadas de conformidade e oferecendo recursos detalhados de registro e geração de relatórios.

P: Qual é a curva de aprendizado para implementação GitOps no Amazon EKS?

R: A curva de aprendizado pode variar dependendo da ferramenta e do conhecimento existente da sua equipe. Geralmente, as equipes que estão familiarizadas com o Git, o Kubernetes e o Amazon EKS se adaptarão mais rapidamente do que outras. As ferramentas mais populares oferecem ampla documentação e recursos de treinamento para facilitar a adoção.

P: Como GitOps as ferramentas lidam com o gerenciamento de segredos no Amazon EKS?

R: GitOps as ferramentas geralmente se integram a soluções externas de gerenciamento de segredos, como AWS Secrets Manager o HashiCorp Vault. Algumas ferramentas também oferecem criptografia integrada para segredos armazenados nos repositórios Git.

P: GitOps As ferramentas podem funcionar com aplicativos sem estado e com monitoramento de estado no Amazon EKS?

R: Sim, GitOps as ferramentas podem funcionar com aplicativos sem estado e com monitoramento de estado. No entanto, o gerenciamento de aplicativos com estado geralmente exige considerações adicionais, como lidar com volumes persistentes e garantir a consistência dos dados durante as atualizações.

P: Como GitOps as ferramentas oferecem suporte ao canary ou às blue/green implantações no Amazon EKS?

R: Muitas GitOps ferramentas oferecem suporte integrado para estratégias avançadas de implantação. Eles podem gerenciar a implantação gradual de novas versões, monitorar problemas e reverter automaticamente se forem detectados problemas. Todas essas operações são definidas como código no repositório Git.

P: Qual é a diferença entre usar uma GitOps ferramenta e usar **`kubectl apply`** com um CI/CD pipeline?

R: GitOps as ferramentas oferecem vantagens sobre `kubectl apply` comandos simples, incluindo detecção e reconciliação automatizadas de desvios, segurança aprimorada por meio de implantações baseadas em pull, melhor auditabilidade e estratégias de implantação mais sofisticadas. Eles também fornecem uma abordagem mais abrangente para gerenciar todo o estado do cluster.

Recursos

Os recursos a seguir fornecem documentação oficial, guias práticos, estudos de caso e análises detalhadas que podem ajudá-lo a tomar uma decisão informada ao escolher uma GitOps ferramenta para seu cluster EKS. Eles abrangem vários aspectos GitOps, incluindo estratégias de implementação, melhores práticas, comparações entre diferentes ferramentas e experiências do mundo real.

AWS recursos:

- Documentação do Amazon EKS
- Automatizando o Amazon EKS com GitOps (AWS publicação no blog)
- Introdução ao GitOps EKS com Weaveworks (workshop)AWS
- Laboratório Flux (workshop do Amazon EKS)
- Laboratório Argo CD (workshop do Amazon EKS)

GitOps e documentação da ferramenta:

- GitOps Melhores práticas para implantação contínua e segurança progressiva (webinar on-demand DevOps .com)
- Documentação do Kubernetes
- Documentação do Argo CD
- Documentação do Flux
- Documentação do Weave GitOps
- Documentação do Jenkins X
- GitLab CI/CD documentação
- Documentação do Spinnaker
- Documentação da Rancher Fleet
- Documentação Codefresh

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	—	30 de abril de 2025

AWS Glossário de orientação prescritiva

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

Números

7 Rs

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

- Refatorar/rearquitetar: mova uma aplicação e modifique sua arquitetura aproveitando ao máximo os recursos nativos de nuvem para melhorar a agilidade, a performance e a escalabilidade. Isso normalmente envolve a portabilidade do sistema operacional e do banco de dados. Exemplo: migrar seu banco de dados Oracle on-premises para o Amazon Aurora Edição Compatível com PostgreSQL.
- Redefinir a plataforma (mover e redefinir [mover e redefinir (lift-and-reshape]): mova uma aplicação para a nuvem e introduza algum nível de otimização a fim de aproveitar os recursos da nuvem. Exemplo: migrar seu banco de dados Oracle on-premises para o Amazon Relational Database Service (Amazon RDS) para Oracle na Nuvem AWS.
- Recomprar (drop and shop): mude para um produto diferente, normalmente migrando de uma licença tradicional para um modelo SaaS. Exemplo: migrar seu sistema de gerenciamento de relacionamento com o cliente (CRM) para o Salesforce.com.
- Redefinir a hospedagem (mover sem alterações [lift-and-shift])mover uma aplicação para a nuvem sem fazer nenhuma alteração a fim de aproveitar os recursos da nuvem. Exemplo: migrar seu banco de dados Oracle on-premises para o Oracle em uma instância do EC2 na Nuvem AWS.
- Realocar (mover o hipervisor sem alterações [hypervisor-level lift-and-shift]): mover a infraestrutura para a nuvem sem comprar novo hardware, reescrever aplicações ou modificar suas operações existentes. Você migra servidores de uma plataforma on-premises para um serviço de nuvem para a mesma plataforma. Exemplo: migrar um Microsoft Hyper-V aplicativo para o. AWS
- Reter (revisitar): mantenha as aplicações em seu ambiente de origem. Isso pode incluir aplicações que exigem grande refatoração, e você deseja adiar esse trabalho para um

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

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

A

ABAC

Consulte controle de acesso baseado em atributo.

serviços abstraídos

Veja serviços gerenciados.

ACID

Veja atomicidade, consistência, isolamento, durabilidade.

migração ativa-ativa

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

migração ativa-passiva

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

AGGREGATE FUNCTION

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

AI

Veja inteligência artificial.

AIOps

Veja operações de inteligência artificial.

anonimização

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

antipadrões

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

controle de aplicações

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

portfólio de aplicações

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

inteligência artificial (IA)

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

operações de inteligência artificial (AIOps)

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

criptografia assimétrica

Um algoritmo de criptografia que usa um par de chaves, uma chave pública para criptografia e uma chave privada para 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 malicioso

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

BCP

Veja planejamento de continuidade de negócios

gráfico de comportamento

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

sistema big-endian

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

classificação binária

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

filtro de bloom

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

blue/green deployment (implantação azul/verde)

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

bot

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

botnet

Redes de bots infectadas por malware e sob o controle de uma única parte, conhecidas como bot herder ou operador de bots. Os botnets são o mecanismo mais conhecido para escalar bots e seu impacto.

ramo

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

Acesso de emergência

Em circunstâncias excepcionais e por meio de um processo aprovado, um meio rápido para um usuário obter acesso a um Conta da AWS que ele normalmente não tem permissão para acessar. Para obter mais informações, consulte o indicador Implement break-glass procedures nas orientações do AWS Well-Architected.

estratégia brownfield

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

cache do buffer

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

capacidade de negócios

O que uma empresa faz para gerar valor (por exemplo, vendas, atendimento ao cliente ou marketing). As arquiteturas de microsserviços e as decisões de desenvolvimento podem ser orientadas por recursos de negócios. Para obter mais informações, consulte a seção Organizados de acordo com as capacidades de negócios do whitepaper Executar microsserviços 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

Veja AWS Cloud Adoption Framework.

implantação canário

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

CCoE

Veja Centro de Excelência da Nuvem.

CDC

Veja captura de dados de alteração.

captura de dados de alterações (CDC)

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

engenharia do caos

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

CI/CD

Veja integração e entrega contínuas.

classificação

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

criptografia no lado do cliente

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

Centro de excelência em nuvem (CCoE)

Uma equipe multidisciplinar que impulsiona os esforços de adoção da nuvem em toda a organização, incluindo o desenvolvimento de práticas 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 corporativa.

computação em nuvem

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

modelo operacional em nuvem

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

estágios de adoção da nuvem

As quatro fases pelas quais as organizações normalmente passam ao migrar para a Nuvem AWS:

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

Esses estágios foram definidos por Stephen Orban na postagem do blog The Journey Toward Cloud-First & the Stages of Adoption no blog de estratégia Nuvem AWS empresarial. Para obter informações sobre como eles se relacionam com a estratégia de AWS migração, consulte o guia de preparação para migração.

CMDB

Veja banco de dados de gerenciamento de configuração.

repositório de código

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

cache frio

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

dados frios

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

visão computacional (CV)

Um campo de IA que usa machine learning para analisar e extrair informações de formatos visuais, como vídeos e imagens digitais. Por exemplo, a Amazon SageMaker AI fornece algoritmos de processamento de imagem para CV.

desvio de configuração

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

banco de dados de gerenciamento de configuração (CMDB)

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

pacote de conformidade

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

integração contínua e entrega contínua (CI/CD)

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

CV

Veja visão computacional.

D

dados em repouso

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

classificação de dados

Um processo para identificar e categorizar os dados em sua rede com base em criticalidade e confidencialidade. É um componente crítico de qualquer estratégia de gerenciamento de riscos de

segurança cibernética, pois ajuda a determinar os controles adequados de proteção e retenção para os dados. A classificação de dados é um componente do pilar de segurança no AWS Well-Architected Framework. Para obter mais informações, consulte Classificação de dados.

desvio de dados

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

dados em trânsito

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

data mesh

Um framework de arquitetura que fornece propriedade de dados distribuída e descentralizada com gerenciamento e governança centralizados.

minimização de dados

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

perímetro de dados

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

pré-processamento de dados

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

proveniência dos dados

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

titular dos dados

Um indivíduo cujos dados estão sendo coletados e processados.

data warehouse

Um sistema de gerenciamento de dados compatível com business intelligence, como analytics. Os data warehouses geralmente contêm grandes quantidades de dados históricos e geralmente são usados para consultas e análises.

linguagem de definição de dados (DDL)

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

linguagem de manipulação de dados (DML)

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

DDL

Veja linguagem de definição de banco de dados.

deep ensemble

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

Aprendizado profundo

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

defense-in-depth

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

administrador delegado

Em AWS Organizations, um serviço compatível pode registrar uma conta de AWS membro para administrar as contas da organização e gerenciar as permissões desse serviço. Essa conta

é chamada de administrador delegado para esse serviço Para obter mais informações e uma lista de serviços compatíveis, consulte Serviços que funcionam com o AWS Organizations na documentação do AWS Organizations .

implantação

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

ambiente de desenvolvimento

Veja ambiente.

controle detectivo

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

mapeamento do fluxo de valor de desenvolvimento (DVSM)

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

gêmeo digital

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

tabela de dimensões

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

desastre

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

Recuperação de desastres (RD)

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

DML

Veja linguagem de manipulação de banco de dados.

design orientado por domínio

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

DR

Veja recuperação de desastres.

Detecção da oscilação

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

DVSM

Veja mapeamento do fluxo de valor de desenvolvimento.

E

EDA

Veja análise exploratória de dados.

EDI

Veja intercâmbio eletrônico de dados.

computação de borda

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

intercâmbio eletrônico de dados (EDI)

A troca automatizada de documentos comerciais entre organizações. Para obter mais informações, consulte O que é EDI (Intercâmbio eletrônico de dados)?.

criptografia

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

chave de criptografia

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

endianismo

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

endpoint

Veja endpoint de serviço.

serviço de endpoint

Um serviço que pode ser hospedado em uma nuvem privada virtual (VPC) para ser compartilhado com outros usuários. Você pode criar um serviço de endpoint com AWS PrivateLink e conceder permissões a outros diretores Contas da AWS ou a AWS Identity and Access Management (IAM).

Essas contas ou entidades principais podem se conectar ao serviço de endpoint de maneira privada criando endpoints da VPC de interface. Para obter mais informações, consulte Criar um serviço de endpoint na documentação do Amazon Virtual Private Cloud (Amazon VPC).

planejamento de recursos empresariais (ERP)

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

criptografia envelopada

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

ambiente

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

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

epic

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

ERP

Veja planejamento de recursos empresariais.

análise exploratória de dados (EDA)

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

F

tabela de fatos

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

Antecipar-se à falha

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

delimitação de isolamento contra falhas

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

ramificação de recursos

Veja ramificação.

recursos

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

importância do recurso

O quanto um recurso é importante para as previsões de um modelo. Isso geralmente é expresso como uma pontuação numérica que pode ser calculada por meio de várias técnicas, como

Shapley Additive Explanations (SHAP) e gradientes integrados. Para obter mais informações, consulte Interpretabilidade do modelo de aprendizado de máquina com AWS.

transformação de recursos

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

prompt few shot

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

FGAC

Veja controle de acesso refinado.

Controle de acesso refinado (FGAC)

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

migração flash-cut

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

FM

Veja modelo de base.

modelo de base (FM)

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

G

IA generativa

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

bloqueio geográfico

Veja restrições geográficas.

restrições geográficas (bloqueio geográfico)

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

Fluxo de trabalho do GitFlow

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

golden image

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

estratégia greenfield

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

barreira de proteção

Uma regra de alto nível que ajuda a governar recursos, políticas e conformidade em todas as unidades organizacionais ()OUs. Barreiras de proteção preventivas impõem políticas para

garantir o alinhamento a padrões de conformidade. Elas são implementadas usando políticas de controle de serviço e limites de permissões do IAM. Barreiras de proteção detectivas detectam violações de políticas e problemas de conformidade e geram alertas para remediação. Eles são implementados usando AWS Config, AWS Security Hub CSPM, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector e verificações personalizadas AWS Lambda .

H

HA

Veja alta disponibilidade.

migração heterogênea de bancos de dados

Migrar seu banco de dados de origem para um banco de dados de destino que usa um mecanismo de banco de dados diferente (por exemplo, Oracle para Amazon Aurora). A migração heterogênea geralmente faz parte de um esforço de redefinição da arquitetura, e converter o esquema pode ser uma tarefa complexa. O AWS fornece o AWS SCT para ajudar nas conversões de esquemas.

alta disponibilidade (HA)

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

modernização de historiador

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

dados de hold-out

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

migração homogênea de bancos de dados

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

dados quentes

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

hotfix

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

período de hipercuidados

Imediatamente após a substituição, o período em que uma equipe de migração gerencia e monitora as aplicações migradas na nuvem para resolver quaisquer problemas. Normalmente, a duração desse período é de 1 a 4 dias. No final do período de hipercuidados, a equipe de migração normalmente transfere a responsabilidade pelas aplicações para a equipe de operações de nuvem.

eu

IaC

Veja infraestrutura como código.

Política baseada em identidade

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

aplicação ociosa

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

IIoT

Veja Internet das Coisas Industrial.

infraestrutura imutável

Um modelo que implanta uma nova infraestrutura para workloads de produção em vez de atualizar, aplicar patches ou modificar a infraestrutura existente. Infraestruturas imutáveis são inerentemente mais consistentes, confiáveis e preditivas do que infraestruturas mutáveis. Para obter mais informações, consulte a prática recomendada Implantar usando infraestrutura imutável no AWS Well-Architected Framework.

VPC de entrada (admissão)

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

migração incremental

Uma estratégia de substituição na qual você migra a aplicação em pequenas partes, em vez de realizar uma única substituição completa. Por exemplo, é possível mover inicialmente apenas alguns microsserviços ou usuários para o novo sistema. Depois de verificar se tudo está funcionando corretamente, mova os microsserviços ou usuários adicionais de forma incremental até poder descomissionar seu sistema herdado. Essa estratégia reduz os riscos associados a migrações de grande porte.

Indústria 4.0

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

infraestrutura

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

Infraestrutura como código (IaC)

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

Internet industrial das coisas (IIoT)

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

VPC de inspeção

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

Internet das coisas (IoT)

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

interpretabilidade

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

IoT

Veja Internet das Coisas.

Biblioteca de informações de TI (ITIL)

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

Gerenciamento de serviços de TI (ITSM)

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

ITIL

Veja biblioteca de informações de TI.

ITSM

Veja gerenciamento de serviços de TI.

L

controle de acesso baseado em etiqueta (LBAC)

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

zona de pouso

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

grande modelo de linguagem (LLM)

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

migração de grande porte

Uma migração de 300 servidores ou mais.

LBAC

Veja controle de acesso baseado em rótulo.

privilégio mínimo

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

mover sem alterações (lift-and-shift)

Veja 7 Rs.

sistema little-endian

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

LLM

Veja grande modelo de linguagem.

ambientes inferiores

Veja ambiente.

M

machine learning (ML)

Um tipo de inteligência artificial que usa algoritmos e técnicas para reconhecimento e aprendizado de padrões. O ML analisa e aprende com dados gravados, por exemplo, dados da Internet das Coisas (IoT), para gerar um modelo estatístico baseado em padrões. Para obter mais informações, consulte Machine learning.

ramificação principal

Veja ramificação.

Malware

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

Serviços gerenciados

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

sistema de execução de manufatura (MES)

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

MAP

Veja Programa de Aceleração da Migração.

mecanismo

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

conta de membro

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

MES

Veja sistema de execução de manufatura.

Transporte de Telemetria de Enfileiramento de Mensagens (MQTT)

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

microsserviço

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

arquitetura de microsserviços

Uma abordagem à criação de aplicações com componentes independentes que executam cada processo de aplicação como um microsserviço. Esses microsserviços se comunicam por meio

de uma interface bem definida usando leveza. APIs Cada microsserviço nessa arquitetura pode ser atualizado, implantado e escalado para atender à demanda por funções específicas de uma aplicação. Para obter mais informações, consulte Implementação de microsserviços em. AWS

Programa de Aceleração da Migração (MAP)

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

migração em escala

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

fábrica de migração

Equipes multifuncionais que simplificam a migração de workloads por meio de abordagens automatizadas e ágeis. As equipes da fábrica de migração geralmente incluem operações, analistas e proprietários de negócios, engenheiros de migração, desenvolvedores e DevOps profissionais que trabalham em sprints. Entre 20 e 50% de um portfólio de aplicações corporativas consiste em padrões repetidos que podem ser otimizados por meio de uma abordagem de fábrica. Para obter mais informações, consulte discussão sobre fábricas de migração e o guia do Cloud Migration Factory neste conjunto de conteúdo.

metadados de migração

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

padrão de migração

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

Avaliação de Portfólio para Migração (MPA)

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

Avaliação de Preparação para Migração (MRA)

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

estratégia de migração

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

ML

Veja machine learning.

modernização

Transformar uma aplicação desatualizada (herdada ou monolítica) e sua infraestrutura em um sistema ágil, elástico e altamente disponível na nuvem para reduzir custos, ganhar eficiência e aproveitar as inovações. Para obter mais informações, consulte Strategy for modernizing applications in the Nuvem AWS.

avaliação de preparação para modernização

Uma avaliação que ajuda a determinar a preparação para modernização das aplicações de uma organização. Ela identifica benefícios, riscos e dependências e determina o quão bem a organização pode acomodar o estado futuro dessas aplicações. O resultado da avaliação é um esquema da arquitetura de destino, um roteiro que detalha as fases de desenvolvimento e os marcos do processo de modernização e um plano de ação para abordar as lacunas identificadas. Para obter mais informações, consulte Evaluating modernization readiness for applications in the Nuvem AWS.

aplicações monolíticas (monólitos)

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

MPA

Veja Avaliação do Portfólio para Migração.

MQTT

Veja Transporte de Telemetria de Enfileiramento de Mensagens.

classificação multiclasse

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

infraestrutura mutável

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

O

OAC

Veja controle de acesso de origem.

OAI

Veja identidade de acesso de origem.

OCM

Veja gerenciamento de alterações organizacionais.

migração offline

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

OI

Veja integração de operações.

Ola

Veja acordo de nível operacional.

migração online

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

OPC-UA

Veja Open Process Communications - Unified Architecture.

Open Process Communications - Unified Architecture (OPC-UA)

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

acordo de nível operacional (OLA)

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

análise de prontidão operacional (ORR)

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

tecnologia operacional (TO)

Sistemas de hardware e software que trabalham com o ambiente físico para controlar operações, equipamentos e infraestrutura industriais. Na manufatura, a integração dos sistemas de

tecnologia da informação (TI) e tecnologia operacional (TO) é o foco principal das transformações da Indústria 4.0.

integração de operações (OI)

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

trilha organizacional

Uma trilha criada por ela AWS CloudTrail registra todos os eventos de 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 análise de prontidão operacional.

OT

Veja tecnologia operacional.

VPC de saída (egresso)

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

P

limite de permissões

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

Informações de identificação pessoal (PII)

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

PII

Veja informações de identificação pessoal.

manual

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

PLC

Veja controlador lógico programável.

PLM

Veja gerenciamento do ciclo de vida do produto.

política

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

persistência poliglota

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

avaliação do portfólio

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

predicado

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

pushdown de predicados

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

controle preventivo

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

principal (entidade principal)

Uma entidade AWS que pode realizar ações e acessar recursos. Essa entidade geralmente é um usuário raiz para um Conta da AWS, uma função do IAM ou um usuário. Para obter mais

informações, consulte Entidade principal em Termos e conceitos de perfis na documentação do IAM.

Privacidade por design

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

zonas hospedadas privadas

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

controle proativo

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

gerenciamento do ciclo de vida do produto (PLM)

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

ambiente de produção

Veja ambiente.

controlador lógico programável (PLC)

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

encadeamento de prompts

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

pseudonimização

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

publish/subscribe (pub/sub)

Um padrão que permite comunicações assíncronas entre microsserviços para melhorar a escalabilidade e a capacidade de resposta. Por exemplo, em um MES baseado em microsserviços, um microsserviço pode publicar mensagens de eventos em um canal em que outros microsserviços possam assinar. O sistema pode adicionar novos microsserviços sem alterar o serviço de publicação.

Q

plano de consulta

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

regressão de planos de consultas

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

R

Matriz RACI

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

RAG

Veja geração aumentada via recuperação.

ransomware

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

Matriz RASCI

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

RCAC

Veja controle de acesso por linha e coluna.

réplica de leitura

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

Redefinir arquitetura

Veja 7 Rs.

objetivo de ponto de recuperação (RPO).

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

objetivo de tempo de recuperação (RTO)

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

refatorar

Veja 7 Rs.

Região

Uma coleção de AWS recursos em uma área geográfica. Cada um Região da AWS é isolado e independente dos outros para fornecer tolerância a falhas, estabilidade e resiliência. Para obter informações, consulte Specify which Regiões da AWS your account can use.

regressão

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

redefinir a hospedagem

Veja 7 Rs.

versão

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

realocar

Veja 7 Rs.

redefinir a plataforma

Veja 7 Rs.

recomprar

Veja 7 Rs.

resiliência

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

política baseada em recurso

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

matriz responsável, accountable, consultada, informada (RACI)

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

controle responsivo

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

reter

Veja 7 Rs.

Retirada

Veja 7 Rs.

Geração Aumentada de Recuperação (RAG)

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

alternância

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

controle de acesso por linha e coluna (RCAC)

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

RPO

Veja objetivo de ponto de recuperação.

RTO

Veja objetivo de tempo de recuperação.

runbook

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

S

SAML 2.0

Um padrão aberto que muitos provedores de identidade (IdPs) usam. Esse recurso permite o login único federado (SSO), para que os usuários possam fazer login Console de gerenciamento da AWS ou chamar as operações da AWS API sem que você precise criar um usuário no IAM

para todos em sua organização. Para obter mais informações sobre a federação baseada em SAML 2.0, consulte Sobre a federação baseada em SAML 2.0 na documentação do IAM.

SCADA

Veja controle de supervisão e aquisição de dados.

SCP

Veja política de controle de serviço.

secret

Em AWS Secrets Manager, informações confidenciais ou restritas, como uma senha ou credenciais de usuário, que você armazena de forma criptografada. Consiste no valor secreto e em seus metadados. O valor secreto pode ser binário, uma única string ou várias strings. Para obter mais informações, consulte What's in a Secrets Manager secret? na documentação do Secrets Manager.

segurança desde a concepção

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

controle de segurança

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

hardening da segurança

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

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

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

automação de resposta de segurança

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

Criptografia do lado do servidor

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

política de controle de serviços (SCP)

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

service endpoint (endpoint de serviço)

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

acordo de serviço (SLA)

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

indicador de nível de serviço (SLI)

Uma avaliação de um aspecto de performance de um serviço, como taxa de erro, disponibilidade ou throughput.

objetivo de nível de serviço (SLO)

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

modelo de responsabilidade compartilhada

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

SIEM

Veja sistema de gerenciamento de eventos e informações de segurança.

ponto único de falha (SPOF)

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

SLA

Veja cacordo de serviço.

SLI

Veja indicador de nível de serviço.

SLO

Veja objetivo de nível de serviço.

split-and-seed modelo

Um padrão para escalar e acelerar projetos de modernização. À medida que novos recursos e lançamentos de produtos são definidos, a equipe principal se divide para criar novas equipes de produtos. Isso ajuda a escalar os recursos e os serviços da sua organização, melhora a produtividade do desenvolvedor e possibilita inovações rápidas. Para obter mais informações, consulte Phased approach to modernizing applications in the Nuvem AWS.

SPOF

Veja ponto único de falha.

esquema em estrela

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

padrão strangler fig

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

sub-rede

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

controle supervisório e aquisição de dados (SCADA)

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

symmetric encryption (criptografia simétrica)

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

testes sintéticos

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

prompt do sistema

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

T

tags

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

variável-alvo

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

lista de tarefas

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

ambiente de teste

Veja ambiente.

treinamento

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

gateway de trânsito

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

fluxo de trabalho baseado em troncos

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

Acesso confiável

Conceder permissões a um serviço que você especifica para realizar tarefas em sua organização AWS Organizations e em suas contas em seu nome. O serviço confiável cria um perfil vinculado ao serviço em cada conta, quando esse perfil é necessário, para realizar tarefas de

gerenciamento para você. Para obter mais informações, consulte Usando AWS Organizations com outros AWS serviços na AWS Organizations documentação.

tuning (ajustar)

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

equipe de duas pizzas

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

U

incerteza

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

tarefas indiferenciadas

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

ambientes superiores

Veja ambiente.

V

aspiração

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

controle de versões

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

emparelhamento da VPC

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

Vulnerabilidade

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

W

cache quente

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

dados mornos

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

função de janela

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

workload

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

workstreams

Grupos funcionais em um projeto de migração que são responsáveis por um conjunto específico de tarefas. Cada workstream é independente, mas oferece suporte aos outros workstreams do

projeto. Por exemplo, o workstream de portfólio é responsável por priorizar aplicações, planejar ondas e coletar metadados de migração. O workstream de portfólio entrega esses ativos ao workstream de migração, que então migra os servidores e as aplicações.

WORM

Veja gravação única e várias leituras.

WQF

Veja AWS Workload Qualification Framework.

gravação única e várias leituras (WORM)

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

Z

exploração de dia zero

Um ataque, normalmente malware, que tira proveito de uma vulnerabilidade zero-day.

vulnerabilidade de dia zero

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

prompt zero shot

Fornecer a um LLM instruções para realizar uma tarefa, mas sem exemplos (shots) que possam ajudar a orientá-lo. O LLM deve usar seu conhecimento pré-treinado para lidar com a tarefa. A eficácia dos prompts zero-shot depende da complexidade da tarefa e da qualidade do prompt. Veja também prompts few-shot.

aplicação zumbi

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

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