Scelta dello GitOps strumento giusto per il tuo cluster Amazon EKS

AWS Guida prescrittiva

AWS Guida prescrittiva: Scelta dello GitOps strumento giusto per il tuo cluster Amazon EKS

Table of Contents

Scelta dello GitOps strumento giusto per il tuo cluster Amazon EKS

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

Aprile 2025 (storia del documento)

Nel panorama in rapida evoluzione delle tecnologie native del cloud, GitOps è emersa come una potente metodologia per la gestione e l'implementazione di applicazioni e infrastrutture. Se utilizzi Amazon Elastic Kubernetes Service (Amazon EKS) GitOps , i principi di implementazione possono migliorare in modo significativo i processi di distribuzione, migliorare l'affidabilità e semplificare le operazioni. Sono disponibili diversi GitOps strumenti e scegliere quello giusto per il cluster EKS è una decisione fondamentale che può influire sull'efficienza del team e sul successo complessivo delle procedure. DevOps

La scelta di uno GitOps strumento appropriato per il tuo ambiente Amazon EKS implica un'attenta considerazione di vari fattori, tra cui requisiti specifici, esperienza del team, esigenze di scalabilità e capacità di integrazione con quelle esistenti Servizi AWS. Ogni strumento è dotato di un proprio set di funzionalità, punti di forza e potenziali limiti, quindi è essenziale allineare la scelta agli obiettivi e al contesto operativo dell'organizzazione.

Questa guida esplora le considerazioni chiave nella selezione GitOps degli strumenti per Amazon EKS, confronta le opzioni utilizzate di frequente e fornisce approfondimenti per aiutarti a prendere una decisione informata. Copre nove strumenti popolari: GitOps

- CD Argo
- Flusso
- Intrecciare GitOps
- Jenkins X
- GitLab CI/CD
- Spinnaker
- Flotta Rancher
- Codice fresco
- Pulumi

Obiettivi aziendali specifici

L'elenco seguente illustra i potenziali obiettivi e risultati quando si sceglie uno strumento per implementare GitOps i principi nei processi di sviluppo e operativi.

Integrazione perfetta con Amazon EKS

Il tuo GitOps strumento dovrebbe integrarsi senza problemi con Amazon EKS e fornire compatibilità con le funzionalità e le ottimizzazioni specifiche di Amazon EKS.

- Supporto nativo per Amazon EKS: cerca strumenti che offrano supporto integrato per Amazon EKS, tra cui una connessione e una gestione semplificate dei cluster.
- Servizio AWS integrazione: assicurati che lo strumento possa interagire con altri strumenti Servizi AWS come AWS Identity and Access Management (IAM), Amazon Elastic Container Registry (Amazon ECR) e Amazon. CloudWatch
- Compatibilità con i componenti aggiuntivi Amazon EKS: verifica che lo strumento supporti i componenti aggiuntivi Amazon EKS e sia in grado di gestirli in modo efficace.

Scalabilità e prestazioni

Il tuo GitOps strumento dovrebbe essere in grado di gestire la scalabilità delle tue operazioni Amazon EKS, da piccoli cluster a grandi ambienti multi-cluster.

- Efficienza delle risorse: valuta il consumo di risorse dello strumento e il suo impatto sulle prestazioni del cluster.
- Operazioni su larga scala: valuta la capacità dello strumento di gestire numerose applicazioni e cluster contemporaneamente.
- Prestazioni sotto carico: considerate le prestazioni dello strumento durante gli aggiornamenti ad alta frequenza e le implementazioni su larga scala.

Conformità e sicurezza

Le funzionalità di sicurezza e le capacità di conformità sono fondamentali, soprattutto nei settori regolamentati o quando si gestiscono dati sensibili.

- Controllo degli accessi: cerca solide funzionalità di controllo degli accessi basato sui ruoli (RBAC) che si integrino con IAM.
- Gestione dei segreti: valuta in che modo lo strumento gestisce le informazioni sensibili e si integra con le nostre altre soluzioni. Gestione dei segreti AWS
- Audit trail: assicurati che lo strumento fornisca funzionalità complete di registrazione e controllo per la conformità e la risoluzione dei problemi.
- Scansione di sicurezza: prendi in considerazione gli strumenti che offrono una scansione di sicurezza integrata per rilevare le vulnerabilità nelle implementazioni.

Facilità d'uso e curva di apprendimento

Lo strumento deve essere facile da usare e in linea con le competenze del team per garantire un'adozione rapida e un utilizzo efficiente.

- Interfaccia utente: valuta l'intuitività delle funzionalità dell'interfaccia a riga di comando (CLI) e dell'interfaccia utente grafica (GUI).
- Qualità della documentazione: cercate documentazione e tutorial completi. up-to-date
- Risorse per l'apprendimento: valuta la disponibilità di materiali di formazione, corsi e risorse per la community.

Supporto comunitario e di rete

Una community e una rete solide possono fornire risorse preziose, plugin e sostenibilità a lungo termine.

- Sviluppo attivo: verifica la frequenza degli aggiornamenti e la reattività dei manutentori.
- Dimensione della comunità: considera le dimensioni e l'attività della comunità di utenti per il supporto e la condivisione delle conoscenze.
- Integrazioni di terze parti: valuta la disponibilità di plugin e integrazioni con altri strumenti del tuo stack.

Funzionalità di gestione multi-cluster

Se disponi di più cluster EKS, la capacità di gestirli in modo efficiente è fondamentale.

- Gestione centralizzata: cerca funzionalità che consentano la gestione di più cluster da un unico piano di controllo.
- Federazione dei cluster: prendi in considerazione gli strumenti che supportano la federazione Kubernetes per applicazioni multi-cluster.
- Parità ambientale: valuta in che modo lo strumento mantiene la coerenza tra diversi ambienti come sviluppo, gestione temporanea e produzione.

Osservabilità e monitoraggio

Lo strumento dovrebbe fornire informazioni chiare sullo stato delle implementazioni e sullo stato del cluster.

- Visibilità dell'implementazione: cerca funzionalità che offrano una visione chiara dello stato e della cronologia dell'implementazione.
- Integrazione con strumenti di monitoraggio: considera quanto bene lo strumento si integra con le soluzioni di monitoraggio più diffuse come Prometheus e Grafana.
- Funzionalità di avviso: valuta la capacità dello strumento di configurare e gestire gli avvisi in caso di problemi di implementazione o deviazioni.

Flessibilità e personalizzazione

La capacità di adattare lo strumento ai flussi di lavoro e ai requisiti specifici è importante per la soddisfazione a lungo termine.

- Estensibilità: cercate architetture di plugin o APIs che consentano di estendere le funzionalità dello strumento.
- Supporto personalizzato per le risorse: verifica che lo strumento sia in grado di gestire in modo efficace le risorse Kubernetes personalizzate.
- Personalizzazione del flusso di lavoro: valuta con quanta facilità puoi adattare i GitOps flussi di lavoro alle esigenze del tuo team.

Fornitura continua e supporto all'implementazione progressiva

Le strategie di implementazione avanzate sono spesso fondamentali per ridurre al minimo i rischi e garantire aggiornamenti senza intoppi.

- Implementazioni Canary: cerca il supporto integrato per le versioni Canary.
- Blue/green deployments: Assess the tool's capabilities for blue/greenstrategie di distribuzione.
- Meccanismi di rollback: assicurate funzionalità affidabili e di easy-to-use rollback per un ripristino rapido in caso di implementazioni fallite.

Convenienza in termini di costi e utilizzo delle risorse

Considerate il costo complessivo di adozione e manutenzione dello strumento, compresi i costi diretti e indiretti.

- Costi di licenza: confronta le opzioni open source con le soluzioni commerciali e prendi in considerazione il supporto e le funzionalità aziendali.
- Costi operativi: valuta i costi operativi aggiuntivi in termini di gestione e manutenzione.
- Consumo di risorse: valuta l'efficienza dello strumento in termini di risorse di elaborazione e archiviazione che sarebbero necessarie.

Considerando attentamente questi risultati e i relativi aspetti, potete prendere una decisione informata sullo GitOps strumento più adatto per il vostro cluster EKS e assicurarvi che lo strumento sia in linea con le esigenze, le capacità e la strategia a lungo termine della vostra organizzazione.

GitOps strumenti per cluster EKS

Esistono diversi GitOps strumenti per Kubernetes attualmente disponibili sul mercato. Ecco un elenco di alcune delle opzioni più utilizzate:

- CD Argo
- Flusso
- Intrecciare GitOps
- Jenkins X
- GitLab CI/CD
- Spinnaker
- Flotta Rancher
- Codice fresco
- Pulumi

Segui i link per visualizzare informazioni dettagliate su come questi strumenti implementano GitOps le pratiche. Ogni strumento presenta punti di forza e casi d'uso. La scelta dipende da fattori quali requisiti specifici, infrastruttura esistente, esperienza del team e funzionalità desiderate. È importante valutare questi strumenti in base alle esigenze dell'organizzazione e alla complessità dell'ambiente Kubernetes.

CD Argo

Argo CD è uno strumento di distribuzione GitOps continua (CD) ampiamente utilizzato per Kubernetes che rispetta diversi principi chiave. GitOps

GitOps supporto

Area	funzionalità dello strumento
Configurazione dichiarativa	Argo CD utilizza configurazioni dichiarative archiviate nei repository Git. Lo stato desiderat o dell'applicazione e dell'infrastruttura è definito nei file YAML. Queste configurazioni descrivon

Area	funzionalità dello strumento
	o cosa deve essere distribuito, non come distribuirlo.
Il sistema di controllo delle versioni come unica fonte di verità	I repository Git fungono da unica fonte di verità per l'intero sistema. Tutte le modifiche all'applicazione e all'infrastruttura vengono apportate tramite Git. Ciò garantisce un audit trail completo e la possibilità di tornare a qualsiasi stato precedente.
Sincronizzazione automatizzata	Argo CD monitora continuamente il repository Git per rilevare eventuali modifiche. Quando vengono rilevate modifiche, sincronizza automaticamente lo stato effettivo del cluster con lo stato desiderato definito in Git. Ciò garantisce che il cluster rifletta sempre lo stato descritto nel repository.
Nativo per Kubernetes	Argo CD è progettato specificamente per gli ambienti Kubernetes. Sfrutta la natura dichiarativa e le risorse personalizzate di Kubernetes per la gestione delle applicazioni.
Riparazione automatica e rilevamento della deriva	Argo CD confronta regolarmente lo stato live del cluster con lo stato desiderato in Git. Se rileva eventuali deviazioni (differenze tra lo stato effettivo e quello desiderato), può correggere automaticamente tali discrepanze.
Supporto multi-cluster e multi-tenancy	Argo CD può gestire più cluster Kubernetes da una singola istanza. Supporta la multi-tenancy, in modo che team diversi possano gestire le proprie applicazioni in modo indipendente.

Area	funzionalità dello strumento
Definizione dell'applicazione	Le applicazioni in Argo CD vengono definite utilizzando l'Application CRD (definizione personalizzata delle risorse). Ciò consente un modo nativo di Kubernetes di definire cosa deve essere distribuito e come.
Separazione tra distribuzione e rilascio	Argo CD separa la distribuzione del codice dal rilascio agli utenti. Ciò si ottiene attravers o varie strategie di implementazione, come le implementazioni di blue/green Or Canary.
Osservabilità e verificabilità	Argo CD fornisce un'interfaccia utente Web e una CLI per osservare lo stato delle applicazi oni e dei cluster. Tutte le azioni vengono registrate per fornire una chiara traccia di controllo delle modifiche e delle implement azioni.
Sicurezza e RBAC	Argo CD si integra con il controllo degli accessi basato sui ruoli (RBAC) di Kubernete s. Supporta l'integrazione Single Sign-On per l'autenticazione e l'autorizzazione.
Architettura collegabile	Argo CD supporta vari sistemi di gestione del controllo del codice sorgente, grafici Helm, Kustomize e altri formati manifest Kubernete s. Questa flessibilità gli consente di adattarsi a diversi ambienti e flussi di lavoro.
Distribuzione continua (CD)	Sebbene Argo CD si concentri sulla distribuz ione continua, può essere integrato con strumenti di integrazione continua (CI) per creare una CI/CD pipeline completa.

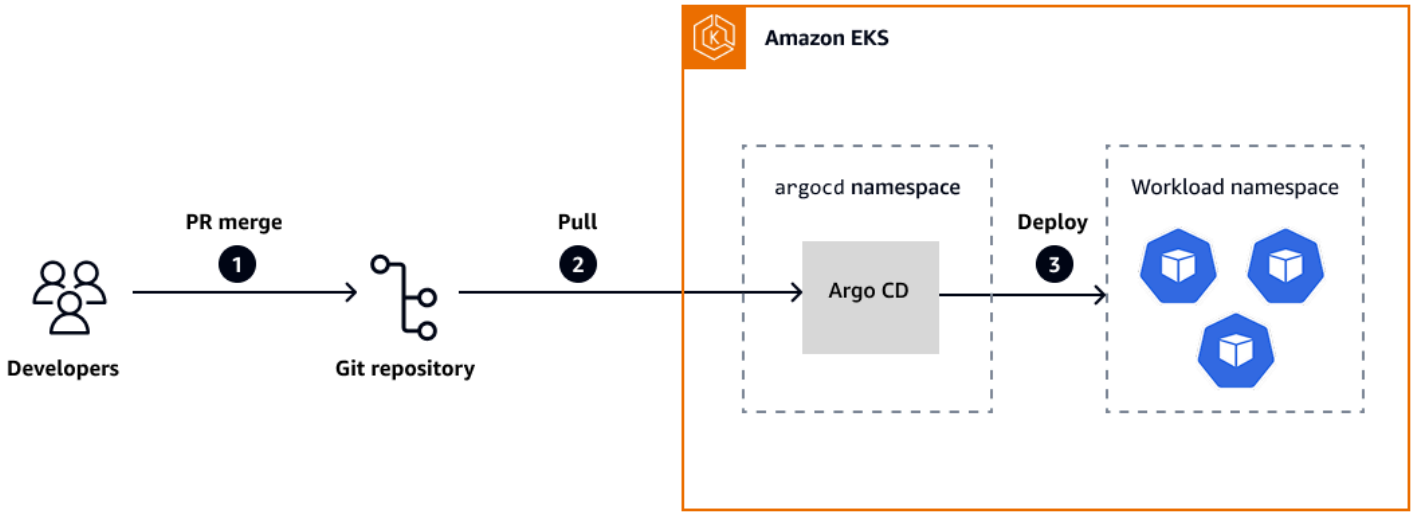
Aderendo a questi GitOps principi, Argo CD offre un modo robusto, scalabile e sicuro per gestire le implementazioni di Kubernetes. Garantisce che lo stato operativo del sistema sia sempre sincronizzato con lo stato desiderato definito nel repository Git e promuove la coerenza, l'affidabilità e la facilità di gestione in ambienti Kubernetes complessi.

Per gli scenari e i requisiti che Argo CD può soddisfare, consulta i casi d'uso di Argo CD più avanti in questa guida. Per un confronto tra Argo CD e Flux, consultate la sezione Confronto delle funzionalità più avanti in questa guida.

Per ulteriori informazioni, consultate la documentazione di Argo CD.

Architettura

Il diagramma seguente illustra un flusso di lavoro GitOps basato su CD che utilizza Argo CD all'interno di un cluster EKS. Per informazioni dettagliate, consultate la documentazione di Argo CD.



dove:

- Fase 1: unione delle Pull Request (PR). Uno sviluppatore esegue modifiche ai manifesti di Kubernetes o ai grafici Helm archiviati in un repository Git. Quando il PR è stato esaminato e unito al ramo principale, lo stato desiderato dell'applicazione viene aggiornato nel controllo del codice sorgente.
- Fase 2: Sincronizzazione del repository. Argo CD viene eseguito all'interno di un namespace dedicato (`argocd`) nel cluster EKS e monitora continuamente il repository Git configurato. Quando rileva le modifiche, estrae gli ultimi aggiornamenti per riconciliare lo stato dichiarato.

- Fase 3: Distribuzione nel namespace di destinazione. Argo CD confronta lo stato desiderato di Git con lo stato live nel cluster. Quindi applica le modifiche necessarie allo spazio dei nomi del carico di lavoro di destinazione in modo che l'applicazione venga distribuita o aggiornata di conseguenza. Ciò include la gestione delle risorse Kubernetes come implementazioni ConfigMaps, servizi e segreti per mantenere la coerenza del cluster con la fonte di verità Git.

Flusso

Flux è un altro strumento per Kubernetes che GitOps implementa i principi in un modo unico.

GitOps supporto

Area	funzionalità dello strumento
Git come unica fonte di verità	Flux utilizza i repository Git come fonte definitiva per definire lo stato desiderato del sistema. Tutte le configurazioni per le applicazioni e l'infrastruttura sono archiviate in Git.
Configurazione dichiarativa	Flux funziona con descrizioni dichiarative dello stato desiderato del cluster. Queste descrizioni sono in genere manifesti di Kubernetes, grafici Helm o sovrapposizioni di Kustomize.
Sincronizzazione automatizzata	Flux monitora continuamente il repository Git per rilevare eventuali modifiche. Quando rileva le modifiche, le applica automaticamente al cluster.
Nativo per Kubernetes	Flux è costruito come un set di controller Kubernetes e risorse personalizzate. Utilizza i meccanismi di estensione di Kubernetes per fornire funzionalità. GitOps
Modello di implementazione basato su Pull	A differenza dei tradizionali CI/CD sistemi basati su push, Flux utilizza un modello basato su pull. Il cluster recupera lo stato desiderato

Area	funzionalità dello strumento
	da Git invece di utilizzare un sistema esterno per inviare le modifiche.
Riconciliazione continua	Flux confronta costantemente lo stato attuale del cluster con lo stato desiderato in Git. Corregge automaticamente qualsiasi deriva rilevata tra questi stati.
Multilocazione	Flux supporta la multi-tenancy attraverso i suoi concetti di Kustomizations e. HelmReleases Diversi team possono gestire le proprie parti della configurazione in modo indipendente.
Consegna progressiva	Flux supporta strategie di implementazione avanzate, come le versioni Canary e i A/B test, tramite il componente Flagger.
Integrazione con Helm	Flux include il supporto nativo per Helm, in modo da poter gestire facilmente le versioni di Helm. GitOps
Automazione dell'aggiornamento delle immagini	Flux può aggiornare automaticamente le immagini dei contenitori in Git quando sono disponibili nuove versioni nel registro dei contenitori.
Personalizza il supporto	Puoi utilizzare il supporto nativo fornito da Flux per Kustomize per personalizzare e correggere i manifesti di Kubernetes.
Sicurezza e RBAC	Flux si integra con Kubernetes RBAC per il controllo degli accessi. Supporta la gestione dei segreti attraverso vari backend.

Area	funzionalità dello strumento
Osservabilità	Flux fornisce informazioni sullo stato e metriche sulla riconciliazione e sulle operazioni. Si integra con gli strumenti di monitoraggio per una migliore osservabilità.
Architettura basata su eventi	Flux utilizza un approccio basato sugli eventi per implementare riconciliazioni e aggiornam enti.
Estensibilità	Lo strumento è progettato per essere estensibi le, in modo da poter aggiungere controller e risorse personalizzati.
Sincronizzazione tra cluster	Flux supporta la gestione di più cluster da un unico set di repository.
Gestione delle dipendenze	Consente di definire le dipendenze tra le diverse parti del sistema e garantisce il corretto ordine delle operazioni.
Ricevitori Webhook	Puoi configurare Flux per ricevere webhook dai provider Git o da altri sistemi per avviare la riconciliazione immediata.

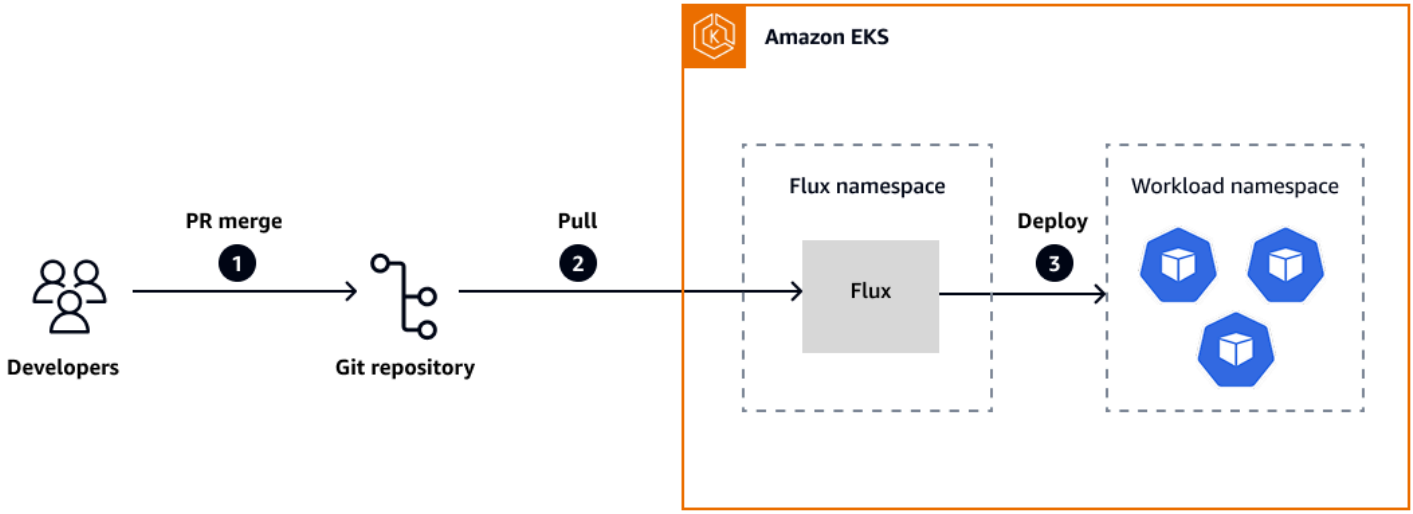
Implementando questi GitOps principi, Flux fornisce un sistema robusto e flessibile per la gestione di cluster e applicazioni Kubernetes. Garantisce che l'infrastruttura e le applicazioni siano sempre sincronizzate con i repository Git e offre coerenza, affidabilità e facilità di gestione in ambienti Kubernetes complessi. L'approccio nativo di Kubernetes dello strumento e l'attenzione all'automazione lo rendono particolarmente adatto per gli ambienti nativi del cloud.

Per gli scenari e i requisiti che Flux può soddisfare, consulta i casi d'uso di Flux più avanti in questa guida. Per un confronto tra Argo CD e Flux, consultate la sezione Confronto delle funzionalità più avanti in questa guida.

Per ulteriori informazioni, consultate la documentazione di Flux.

Architettura

Il diagramma seguente illustra un flusso di lavoro GitOps basato su CD che utilizza Flux all'interno di un cluster EKS. Per informazioni dettagliate, consultate la documentazione di Flux.



dove:

- Fase 1: unione delle pull request (PR). Uno sviluppatore esegue modifiche ai manifesti di Kubernetes o ai grafici Helm archiviati in un repository Git. Quando il PR è stato esaminato e unito al ramo principale, lo stato desiderato dell'applicazione viene aggiornato nel controllo del codice sorgente.
- Fase 2: Sincronizzazione del repository. Flux viene eseguito all'interno di un namespace dedicato nel cluster EKS e monitora continuamente il repository Git configurato. Quando rileva le modifiche, estrae gli ultimi aggiornamenti per riconciliare lo stato dichiarato.
- Fase 3: Distribuzione nel namespace di destinazione. Flux confronta lo stato desiderato di Git con lo stato live nel cluster. Quindi applica le modifiche necessarie allo spazio dei nomi del carico di lavoro di destinazione in modo che l'applicazione venga distribuita o aggiornata di conseguenza.

Intreccia GitOps

Weave GitOps è stato sviluppato da Weaveworks, che è la società che ha introdotto il termine. GitOps Questo strumento fornisce una GitOps soluzione completa che si basa sui principi fondamentali. GitOps

GitOps supporto

Area	funzionalità dello strumento
Git come unica fonte di verità	Weave GitOps utilizza i repository Git come fonte autorevole per definire lo stato desiderat o del sistema. Tutte le configurazioni, inclusi i manifesti delle applicazioni, le definizioni dell'infrastruttura e le politiche, sono archiviate in Git.
Configurazione dichiarativa	Il sistema si basa su descrizioni dichiarative dell'intero stato del sistema. Queste descrizioni sono in genere manifesti di Kubernetes, grafici Helm o altri formati dichiarativi.
Sincronizzazione automatica	Weave monitora GitOps continuamente i repository Git per rilevare eventuali modifiche . Quando rileva le modifiche, le applica automaticamente all'ambiente di destinazione.
Architettura nativa di Kubernetes	Weave GitOps è costruito come un insieme di controller Kubernetes e risorse personali zzate. Utilizza i meccanismi di estensione di Kubernetes per fornire funzionalità. GitOps
Riconciliazione continua	Questo strumento confronta costantemente lo stato effettivo del cluster con lo stato desiderat o definito in Git. Corregge automaticamente qualsiasi deriva rilevata tra questi stati.
Gestione multi-cluster	Weave GitOps supporta la gestione di più cluster Kubernetes da un unico piano di controllo. Consente l'implementazione coerente delle applicazioni in diversi ambienti.
La politica come codice	Weave GitOps incorpora il concetto di politica come codice per far rispettare le regole di

Area	funzionalità dello strumento
	sicurezza e conformità. Le politiche sono controllate dalla versione insieme al codice dell'applicazione e alle definizioni dell'infr astruttura.
Distribuzione progressiva	Questo strumento supporta strategie di implementazione avanzate come le versioni e le blue/green distribuzioni Canary. Si integra con Flagger per una distribuzione automatica e progressiva.
Osservabilità e dashboard	Weave GitOps fornisce dashboard integrate per il monitoraggio dello stato delle applicazioni e dei cluster. Offre approfondimenti sui processi di riconciliazione e sullo stato dei cluster.
Sicuro fin dalla progettazione	Lo strumento implementa le migliori pratiche di sicurezza, tra cui l'integrazione RBAC e la gestione dei segreti. Supporta vari metodi di autenticazione e si integra con i provider di identità aziendali.
Estensibilità e integrazione	Lo strumento è progettato per funzionare con un'ampia gamma di strumenti nativi del cloud. Supporta strumenti popolari come Flux, Helm e Kustomize.
Piattaforme self-service per sviluppatori	Weave GitOps consente la creazione di piattaforme self-service per gli sviluppatori. Fornisce modelli e barriere per la distribuzione delle applicazioni.
Automazione di GitOps	Lo strumento automatizza molti aspetti del GitOps flusso di lavoro, inclusa la generazione di pull request per gli aggiornamenti.

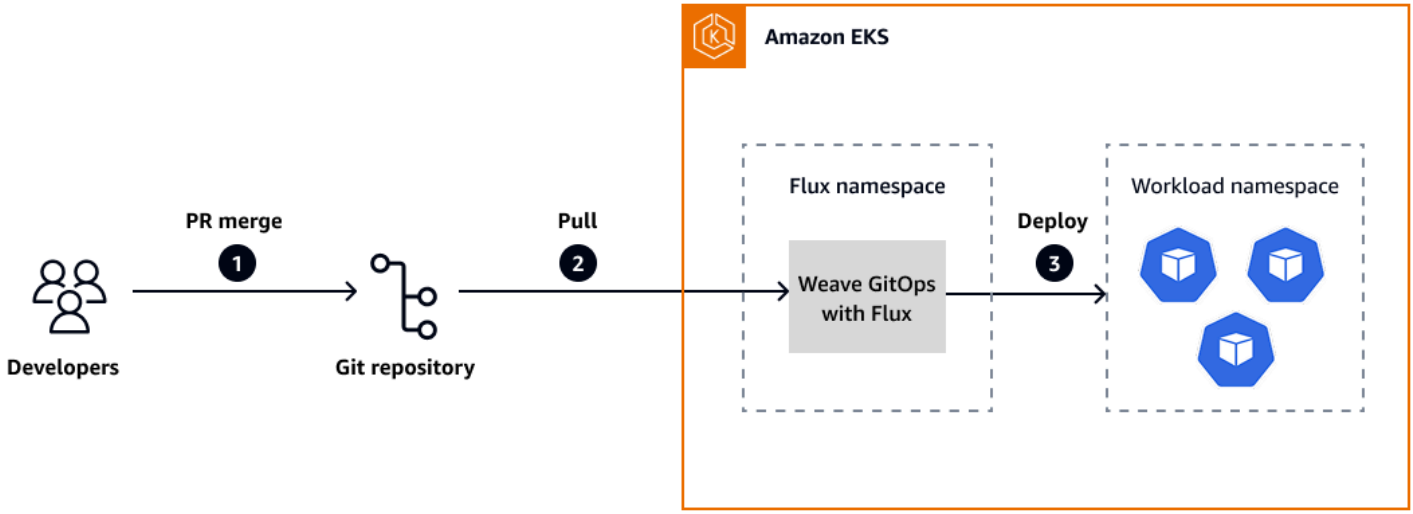
Area	funzionalità dello strumento
Pipeline di distribuzione continua	Si integra con i CI/CD sistemi per creare pipeline di end-to-end distribuzione.
Audit e conformità	Weave DevOps fornisce un audit trail completo di tutte le modifiche e le azioni. Ti aiuta a soddisfare i requisiti di conformità attraverso il controllo delle versioni e i processi automatiz zati.
Scalabilità	Lo strumento è progettato per scalare da piccoli progetti a grandi implementazioni di livello aziendale.
Collaborazione in team	Weave GitOps facilita la collaborazione tra i team di sviluppo e operativi attraverso flussi di lavoro basati su Git.
GitOps come servizio	Questo strumento viene offerto GitOps come servizio gestito, che semplifica l'adozione e la gestione.
Supporto ibrido e multi-cloud	Weave GitOps consente una gestione coerente tra diversi provider di cloud e ambienti locali.
Sicurezza continua	Lo strumento integra la scansione di sicurezza e l'applicazione delle policy durante tutto il processo di implementazione.

Weave GitOps implementa questi principi per fornire una GitOps soluzione completa che va oltre l'automazione di base dell'implementazione. Mira a creare un modello operativo completo per le applicazioni native del cloud incentrato su sicurezza, scalabilità e facilità d'uso. Aderendo a questi GitOps principi, Weave GitOps aiuta le organizzazioni a ottenere una gestione coerente, verificabile ed efficiente dei propri ambienti Kubernetes su più cluster e provider cloud.

Per ulteriori informazioni, consulta la documentazione di Weave. GitOps

Architettura

Il diagramma seguente illustra un flusso di lavoro GitOps basato su CD che utilizza Weave GitOps all'interno di un cluster EKS. Per informazioni dettagliate, consultate il repository Weave. GitOps



dove:

- Fase 1: unione delle pull request (PR). Uno sviluppatore esegue modifiche ai manifesti di Kubernetes o ai grafici Helm archiviati in un repository Git. Quando il PR è stato esaminato e unito al ramo principale, lo stato desiderato dell'applicazione viene aggiornato nel controllo del codice sorgente.
- Fase 2: Sincronizzazione del repository. Weave GitOps viene eseguito all'interno dello spazio dei nomi Flux nel cluster EKS e monitora continuamente il repository Git configurato. Quando rileva le modifiche, estrae gli ultimi aggiornamenti per riconciliare lo stato dichiarato.
- Fase 3: Distribuzione nel namespace di destinazione. Weave GitOps confronta lo stato desiderato di Git con lo stato live nel cluster. Quindi applica le modifiche necessarie allo spazio dei nomi del carico di lavoro di destinazione in modo che l'applicazione venga distribuita o aggiornata di conseguenza.

Jenkins X

Jenkins X è una CI/CD piattaforma open source nativa per il cloud che GitOps implementa i principi per gli ambienti Kubernetes. Sebbene Jenkins X non sia esclusivamente uno GitOps strumento come Argo CD o Flux, incorpora pratiche nei suoi flussi di lavoro. GitOps

GitOps supporto

Area	funzionalità dello strumento
Flusso di lavoro incentrato su Git	Jenkins X utilizza i repository Git come fonte primaria di verità sia per il codice dell'appl icazione che per la configurazione. Tutte le modifiche alle applicazioni e all'infrastruttura vengono apportate tramite Git.
Ambiente come codice (eAC)	Gli ambienti (come lo staging e la produzion e) sono definiti come codice nei repository Git. Ciò consente il controllo della versione e la revisione delle configurazioni dell'ambiente.
Pipeline automatizzate CI/CD	Jenkins X imposta automaticamente le CI/CD pipeline per i progetti. Queste pipeline sono definite come codice (pipeline come codice) e archiviate in Git.
Nativa per Kubernetes	Jenkins X è progettato specificamente per gli ambienti Kubernetes. Utilizza risorse Kubernete s e definizioni di risorse personalizzate (). CRDs
Ambienti di anteprima	Jenkins X crea automaticamente ambienti temporanei per le pull request. Consente una facile revisione e verifica delle modifiche prima delle fusioni.
Promozione tra ambienti	Jenkins X utilizza un GitOps approccio per promuovere le applicazioni tra ambienti (ad esempio, dalla messa in scena alla produzion e). Le promozioni vengono gestite utilizzando le pull request per garantire processi di revisione e approvazione adeguati.

Area	funzionalità dello strumento
Gestione dei grafici di Helm	Jenkins X utilizza i grafici Helm per impacchet tare e distribuire applicazioni. I grafici sono controllati dalla versione nei repository Git.
Controllo automatico delle versioni	Jenkins X gestisce automaticamente il controllo delle versioni di applicazioni e versioni. Utilizza il controllo delle versioni semantiche e genera note di rilascio.
ChatOps integrazione	Jenkins X supporta ChatOps le operazioni più comuni. Ciò è in linea con i GitOps principi di automazione e collaborazione.
Estensibilità	Questo strumento fornisce un sistema di plugin per estendere le funzionalità. Consente l'integra zione con vari strumenti nativi del cloud.
Infrastructure as code (IaC)	Jenkins X supporta Terraform e altri strumenti IAC per definire e gestire l'infrastruttura. CloudFormation AWS Cloud Development Kit (AWS CDK) Le definizioni dell'infrastruttur a sono controllate dalla versione insieme al codice dell'applicazione.
Rollback automatizzati	Jenkins X supporta i rollback automatici se vengono rilevati problemi dopo la distribuzione.
Gestione dei segreti	Lo strumento si integra con soluzioni esterne di gestione dei segreti per gestire le informazioni sensibili in modo sicuro.
Osservabilità	Jenkins X fornisce l'integrazione con strumenti di monitoraggio e registrazione per l'osserva bilità.

Area	funzionalità dello strumento
Supporto multi-cloud	Jenkins X è progettato per funzionare con diversi provider di cloud e ambienti locali.
Collaborazione in team	Questo strumento incoraggia la collabora zione attraverso flussi di lavoro basati su Git e richieste pull.
Feedback continuo	Lo strumento fornisce un feedback rapido sulle modifiche attraverso test automatizzati e ambienti di anteprima.
DevOps migliori pratiche	Jenkins X implementa le DevOps migliori pratiche di default, inclusi i GitOps principi.
Configurazione dichiarativa	Lo strumento utilizza configurazioni dichiarative per definire applicazioni e ambienti.
Aggiornamenti automatizzati	Jenkins X fornisce strumenti per automatizzare gli aggiornamenti della piattaforma Jenkins X stessa.

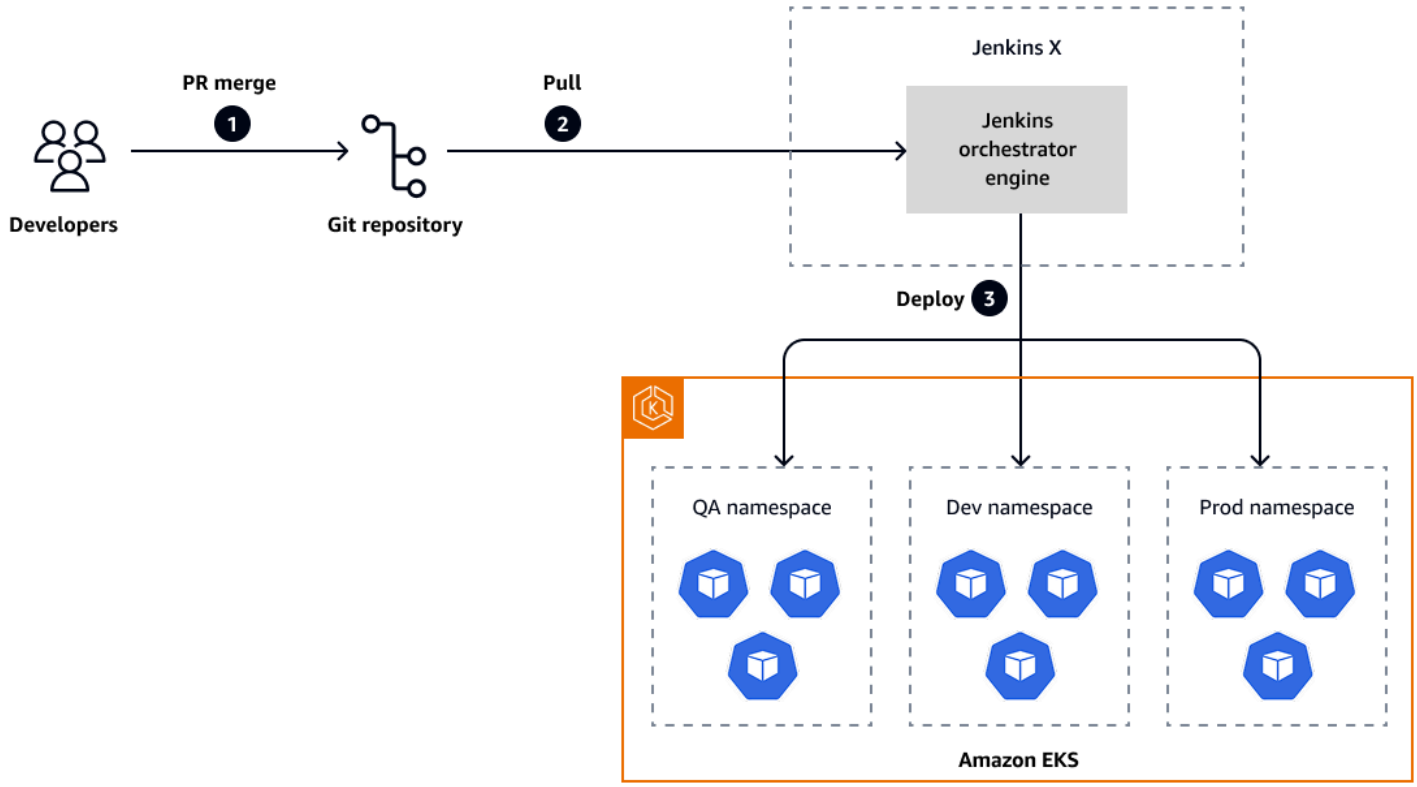
Jenkins X implementa questi GitOps principi per creare una soluzione CI/CD completa per Kubernetes. Mira ad automatizzare e semplificare l'intero processo di distribuzione del software, dall'invio del codice all'implementazione in produzione, rispettando al contempo le pratiche. GitOps In questo modo, aiuta i team a realizzare implementazioni più rapide, affidabili e coerenti in ambienti nativi del cloud.

La differenza fondamentale tra Jenkins X e strumenti come Argo CD o Flux è che Jenkins X offre una CI/CD soluzione più completa, che include l'automazione degli edifici e la gestione delle pipeline, pur incorporando principi per l'implementazione e la gestione dell'ambiente. GitOps Ciò lo rende particolarmente adatto ai team che necessitano di una all-in-one soluzione che copra sia gli aspetti CI che quelli CD all'interno di un unico framework. GitOps

Per ulteriori informazioni, consulta la documentazione di Jenkins X.

Architettura

Il diagramma seguente illustra un flusso di lavoro GitOps basato su CD che utilizza Jenkins X. Per informazioni dettagliate, consulta la documentazione di Jenkins X.



dove:

- Fase 1: unione delle Pull Request (PR). Uno sviluppatore crea una pull request che include modifiche ai manifest di Kubernetes, ai grafici Helm o al codice dell'applicazione archiviato in un repository Git. Dopo la revisione e l'approvazione, il PR viene unito al ramo principale e aggiorna lo stato desiderato nel controllo del codice sorgente.
- Fase 2: Sincronizzazione del repository. Jenkins X attiva automaticamente una CI/CD pipeline quando rileva la modifica. La pipeline crea, testa e promuove l'applicazione in diversi ambienti (ad esempio, staging e produzione) utilizzando principi. GitOps
- Fase 3: Distribuzione nei namespace di destinazione. Jenkins X aggiorna i repository ambientali (staging e production) con le nuove versioni dell'applicazione. Il cluster riconcilia automaticamente le modifiche estraendo i manifesti più recenti da Git e distribuendo l'applicazione nei namespace appropriati.

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/CDnon è esclusivamente uno GitOps strumento, puoi configurarlo per implementare GitOps i principi, specialmente quando lo usi per le implementazioni Kubernetes.

GitOps supporto

Area	funzionalità dello strumento
Git come unica fonte di verità	GitLab CI/CD utilizza i repository Git per archiviare sia il codice dell'applicazione che le configurazioni dell'infrastruttura. Tutte le modifiche al sistema vengono apportate tramite Git, che garantisce una cronologia e un audit trail completi.
Configurazione dichiarativa	GitLab Le pipeline CI/CD sono definite in un file.gitlab-ci.yml, che è una configurazione dichiarativa archiviata nel repository Git. I manifesti di Kubernetes, i grafici Helm o altri file Infrastructure as Code (IaC) possono essere archiviati nello stesso repository per definire lo stato desiderato dell'infrastruttura.
Pipeline automatizzate	GitLab CI/CD attiva automaticamente le pipeline quando le modifiche vengono inviate al repository. Queste pipeline possono includere fasi per la creazione, il test e la distribuzione di applicazioni.
Integrazione con Kubernetes	GitLab CI/CD fornisce l'integrazione nativa di Kubernetes e supporta implementazioni in stile nei cluster Kubernetes. GitOps Può creare e gestire automaticamente le risorse Kubernetes in base alla configurazione in Git.

Area	funzionalità dello strumento
Gestione dell'ambiente	GitLab CI/CD supporta la definizione di più ambienti (come lo staging e la produzione) come codice. Le implementazioni in questi ambienti possono essere automatizzate o richiedere l'approvazione manuale, in conformit à con le pratiche. GitOps
Rivedi le applicazioni	GitLab può creare automaticamente ambienti temporanei per le richieste di unione, simili agli ambienti di anteprima di altri GitOps strumenti . Ciò semplifica la revisione e il test delle modifiche prima delle fusioni.
Distribuzione continua	GitLab CI/CD può essere configurato per implementare automaticamente le modifiche ai cluster Kubernetes quando le modifiche vengono unite a filiali specifiche.
IaC	GitLab CI/CD supporta l'integrazione con strumenti come Terraform e la gestione dell'infr astruttura come codice. CloudFormation Le definizioni dell'infrastruttura possono essere controllate in base alla versione insieme al codice dell'applicazione.
Osservabilità e monitoraggio	GitLab CI/CD offre funzionalità integrate di monitoraggio e osservabilità, inclusa l'integra zione con Prometheus e Grafana.
Scansione di sicurezza	GitLab CI/CD includes built-in security scanning tools that can be integrated into the CI/CDpipe line per applicare la sicurezza come parte del flusso di lavoro. GitOps

Area	funzionalità dello strumento
Registro dei contenitori	GitLab CI/CD include un registro dei contenitori integrato per una perfetta integrazione della gestione delle immagini dei contenitori nel flusso di lavoro. GitOps
Automatico DevOps	La DevOps funzionalità Auto nelle GitLab CI/CD can automatically configure CI/CD pipeline che seguono i GitOps principi delle implementazioni Kubernetes.
Flussi di lavoro di approvazione	GitLab CI/CD supporta i processi di approvazione per le implementazioni, che forniscono promozioni controllate tra gli ambienti.
Gestione dei segreti	GitLab CI/CD provides features to securely manage and use secrets within CI/CDconditure.
Versioni e rilasci	GitLab CI/CD supports automatic versioning and release management as part of the CI/CDprocesso.
Rollback	GitLab CI/CD consente di ripristinare facilmente le versioni precedenti se vengono rilevati problemi dopo la distribuzione.
Audit logs	GitLab CI/CD fornisce registri di controllo completi per tutte le azioni a supporto dell'aspetto della tracciabilità di. GitOps
Pipeline multiprogetto	GitLab CI/CD supporta GitOps flussi di lavoro complessi che si estendono su più progetti o repository.

Area	funzionalità dello strumento
ChatOps	GitLab CI/CD supporta le ChatOps integrazi oni, che forniscono collaborazione e operazioni tramite interfacce di chat.
Gestione dei cluster Kubernetes	GitLab CI/CD offre funzionalità per la gestione dei cluster Kubernetes direttamente dall'inte rfaccia. GitLab

Tuttavia 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, e la gestione di Kubernetes, lo rendono un potente strumento per l'implementazione dei flussi di lavoro. GitOps

La differenza fondamentale tra le funzionalità. 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 Ciò lo rende particolarmente adatto ai team che necessitano di una all-in-one soluzione in grado di implementare GitOps pratiche all'interno di un sistema di sviluppo più ampio.

Per ulteriori informazioni su GitLab CI/CD e la sua architettura, consulta la GitLab documentazione CI/CD.

Spinnaker

Sebbene Spinnaker non sia progettato esclusivamente come GitOps strumento, puoi configurarlo per implementare GitOps i principi, specialmente quando lo utilizzi per implementazioni native del cloud e Kubernetes.

GitOps supporto

Area	funzionalità dello strumento
Configurazione dichiarativa	Spinnaker utilizza definizioni di pipeline dichiarative, che in genere vengono archiviat e come file JSON o YAML. Queste definizio

Area	funzionalità dello strumento
	ni di pipeline possono essere controllate dalla versione nei repository Git, in linea con le pratiche. GitOps
IaC	Spinnaker supporta la definizione dell'infr astruttura e delle configurazioni di implement azione come codice. Queste definizioni possono essere archiviate in archivi Git e possono fungere da unica fonte di verità.
Implementazioni multi-cloud	Spinnaker è progettato per funzionare su più provider cloud e cluster Kubernetes. Consente pratiche coerenti in ambienti diversi. GitOps
Pipeline come codice	Le pipeline Spinnaker possono essere definite come codice e archiviate nei repository Git. Ciò consente il controllo della versione e la revisione dei processi di distribuzione.
Implementazioni automatizzate	È possibile configurare Spinnaker per avviare automaticamente le distribuzioni in base alle modifiche nei repository Git. Lo strumento supporta pratiche di distribuzione continua che sono fondamentali per. GitOps
Infrastruttura immutabile	Spinnaker promuove l'uso di un'infrastruttura immutabile, che è un concetto chiave in. GitOps Incoraggia l'implementazione di nuove istanze invece di modificare le istanze esistenti.
Rollback e controllo delle versioni	Spinnaker offre solide funzionalità di rollback e un rapido ritorno ai precedenti buoni stati noti. Supporta il controllo delle versioni delle implementazioni, in linea con i principi di tracciabilità. GitOps

Area	funzionalità dello strumento
Flussi di lavoro di approvazione	Spinnaker include fasi di valutazione manuali nelle pipeline per supportare promozioni controllate tra ambienti. Ciò supporta GitOps le pratiche di separazione tra distribuzioni e versioni.
Canary e implementazioni blue/green	Spinnaker supporta strategie di implement azione avanzate in linea con le GitOps pratiche per rilasci sicuri e controllati.
Integrazione con i sistemi di controllo delle versioni	Spinnaker può integrarsi con vari provider Git per avviare pipeline basate sugli eventi del repository.
Integrazione con Kubernetes	Spinnaker fornisce supporto nativo per Kubernetes e supporta la gestione in stile delle risorse Kubernetes. GitOps
Gestione degli Artifact	Spinnaker supporta la gestione degli artefatti e il controllo delle versioni, fondamentali per il mantenimento di un flusso di lavoro. GitOps
Osservabilità e monitoraggio	Spinnaker offre l'integrazione con strumenti di monitoraggio per supportare l'aspetto dell'osse rvabilità di. GitOps
Audit trail	Spinnaker fornisce registri e cronologia di implementazione dettagliati, che supportano il principio di verificabilità di. GitOps
Controllo degli accessi basato sui ruoli (RBAC)	Questo strumento implementa RBAC per un controllo dettagliato su chi può eseguire quali azioni, in linea con le pratiche di sicurezza. GitOps

Area	funzionalità dello strumento
Creazione di modelli e parametrizzazione	Spinnaker supporta la creazione di modelli nelle definizioni delle pipeline per consentire implementazioni riutilizzabili e parametrizzate.
Promozione dell'ambiente	Spinnaker facilita la promozione delle applicazioni tra ambienti (ad esempio, dalla messa in scena alla produzione) in modo controllato.
Integrazione con strumenti CI	Spinnaker può integrarsi con vari strumenti di integrazione continua (CI) per fornire una CI/CD pipeline completa che rispetti i principi. GitOps
Fasi ed estensioni personalizzate	Questo strumento supporta fasi ed estensioni personalizzate, in modo che i team possano implementare GitOps flussi di lavoro personalizzati in base alle loro esigenze.
Gestione centralizzata	Spinnaker fornisce una piattaforma centralizzata per la gestione delle implementazioni in più ambienti e provider di cloud.

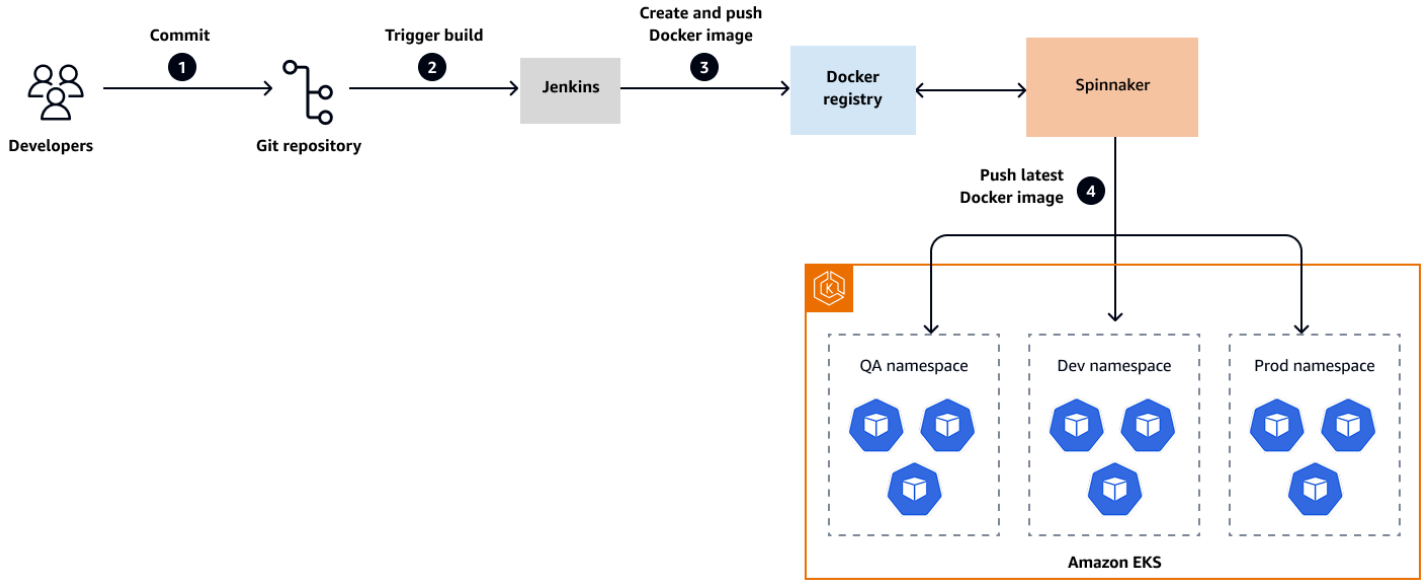
Sebbene Spinnaker non sia commercializzato principalmente come GitOps strumento, la sua flessibilità e il robusto set di funzionalità lo rendono in grado di implementare flussi di GitOps lavoro, specialmente in ambienti complessi e multi-cloud. La differenza fondamentale tra Spinnaker e GitOps strumenti dedicati come Argo CD o Flux è che Spinnaker offre una piattaforma di distribuzione continua più completa con strategie di implementazione avanzate e supporto multi-cloud.

La forza di Spinnaker risiede nella sua capacità di gestire scenari di implementazione complessi tra vari provider di cloud e nel supporto per strategie di implementazione avanzate. Quando Spinnaker è configurato correttamente, può implementare i principi in modo efficace. GitOps Questo lo rende uno strumento potente per le organizzazioni che desiderano adottare GitOps pratiche in ambienti diversi e complessi.

Per ulteriori informazioni, consulta la documentazione di Spinnaker.

Architettura

Il diagramma seguente illustra un flusso di lavoro GitOps basato su CD che utilizza Spinnaker e Jenkins X. Per informazioni dettagliate, consulta la documentazione di Spinnaker.



dove:

- Fase 1: Code commit. Gli sviluppatori trasferiscono le modifiche al codice dell'applicazione in un repository Git. Queste modifiche potrebbero includere aggiornamenti all'applicazione stessa, a Dockerfiles o ai manifest di Kubernetes.
- Fase 2: build e creazione di immagini in Jenkins. Jenkins viene attivato automaticamente dal repository Git tramite un webhook o un polling. Jenkins crea l'applicazione, crea un'immagine Docker e invia l'immagine creata a un registro Docker configurato (come Amazon ECR o Docker Hub).
- Fase 3: monitoraggio delle immagini Spinnaker e attivazione della pipeline. Spinnaker monitora continuamente il registro Docker alla ricerca di nuove immagini. Quando viene rilevata una nuova versione dell'immagine, Spinnaker attiva automaticamente una pipeline per avviare il processo di distribuzione.
- Fase 4: Distribuzione nei namespace di destinazione. Spinnaker distribuisce la nuova immagine Docker su Amazon EKS. In base alle configurazioni della pipeline, l'immagine viene distribuita sui namespace di destinazione nel cluster. Spinnaker garantisce che venga implementata la versione più recente dell'applicazione aderendo a strategie di distribuzione definite come le implementazioni o canary. blue/green

Rancher Fleet

Rancher Fleet è una GitOps-at-scale soluzione progettata specificamente per la gestione di più cluster Kubernetes. Aderisce strettamente ai GitOps principi pur concentrandosi sulla scalabilità e sulla gestione multi-cluster.

GitOps supporto

Area	funzionalità dello strumento
Git come unica fonte di verità	Fleet utilizza gli archivi Git come fonte autorevole per definire lo stato desiderato delle applicazioni e delle risorse su più cluster. Tutte le configurazioni, inclusi i manifesti di Kubernetes, i grafici Helm e le risorse personalizzate, sono archiviate in Git.
Configurazione dichiarativa	Fleet funziona con descrizioni dichiarative dello stato desiderato per applicazioni e risorse. Possono essere Kubernetes YAML non elaborati, grafici Helm, file Kustomize o risorse personalizzate specifiche per Fleet.
Sincronizzazione automatica	Fleet monitora continuamente i repository Git per rilevare eventuali modifiche. Applica automaticamente le modifiche ai cluster di destinazione quando rileva differenze tra lo stato Git e lo stato del cluster.
Gestione di più cluster	Fleet è progettato specificamente per gestire le implementazioni su più cluster Kubernetes. Può gestire migliaia di cluster da un unico piano di controllo.
Architettura nativa di Kubernetes	Fleet è costruito come un insieme di risorse e controller personalizzati Kubernetes. Utilizza i

Area	funzionalità dello strumento
	meccanismi di estensione di Kubernetes per le operazioni. GitOps
Riconciliazione continua	Fleet confronta costantemente lo stato effettivo dei cluster con lo stato desiderato definito in Git. Corregge automaticamente qualsiasi deriva rilevata tra questi stati.
Raggruppamento e targeting dei cluster	Fleet consente di raggruppare i cluster e indirizzare le distribuzioni a gruppi specifici o singoli cluster. Supporta l'implementazione coerente delle applicazioni in diversi ambienti e tipi di cluster.
Configurazioni a più livelli	Fleet supporta configurazioni a più livelli, che forniscono configurazioni di base con sovrapposizioni specifiche dell'ambiente. Ciò è in linea con le pratiche di gestione efficiente di più ambienti. GitOps
Integrazione con Helm	Fleet fornisce supporto nativo per i grafici Helm e consente una facile gestione di applicazioni complesse. Può modificare e gestire le versioni di Helm tramite GitOps flussi di lavoro.
Definizioni di risorse personalizzate () CRDs	Fleet utilizza risorse personalizzate come GitRepo Bundle per definire le distribuzioni. Questi CRDs forniscono un modo nativo di Kubernetes per definire i flussi di lavoro. GitOps
Sicurezza e RBAC	Fleet si integra con Kubernetes RBAC per il controllo degli accessi. Supporta la gestione sicura di informazioni e credenziali sensibili.

Area	funzionalità dello strumento
Osservabilità	Fleet fornisce informazioni sullo stato di sincronizzazione di cluster e applicazioni. Offre informazioni dettagliate sui GitOps processi in tutta la flotta di cluster.
Scalabilità	Fleet è progettato per essere scalabile per gestire migliaia di cluster in modo efficiente. Supporta GitOps operazioni su larga scala in ambienti aziendali.
Gestione delle dipendenze	È possibile definire le dipendenze tra diverse risorse e applicazioni. Fleet garantisce il rispetto dell'ordine corretto delle operazioni nelle implementazioni complesse.
Personalizzazione ed estensibilità	Fleet supporta script personalizzati e ganci per il ciclo di vita per la personalizzazione avanzata delle implementazioni. Consente l'integrazione con gli strumenti e i flussi di lavoro esistenti.
Supporto offline e senza interruzioni	Fleet può operare in ambienti con connettività Internet limitata o assente. Supporta GitOps flussi di lavoro in ambienti ad alta sicurezza o regolamentati.
Implementazioni progressive	Fleet supporta implementazioni graduali tra i cluster, che consentono strategie di implement azione controllate e graduali.
Interfaccia di gestione unificata	Fleet fornisce un'unica interfaccia per la gestione dei GitOps flussi di lavoro in tutti i cluster. Semplifica le operazioni in ambienti complessi e multi-cluster.

Area	funzionalità dello strumento
Integrazione con altri strumenti Rancher	Fleet si integra con altri strumenti Rancher per fornire una soluzione di gestione Kubernetes completa.
Audit trail e conformità	Fleet mantiene una traccia di controllo chiara di tutte le modifiche e le implementazioni. Ti aiuta a soddisfare i requisiti di conformità attravers o operazioni basate su Git con controllo della versione.

Rancher Fleet implementa questi GitOps principi con una forte attenzione alla scalabilità e alla gestione multicluster. Il suo design è particolarmente adatto per le organizzazioni che gestiscono un gran numero di cluster Kubernetes in diversi ambienti, data center o provider di cloud.

Il principale elemento di differenziazione di Fleet è la sua capacità di gestire su larga scala. GitOps Questa funzionalità la rende particolarmente utile per le grandi aziende o i provider di servizi gestiti che gestiscono numerosi cluster. Strumenti come Argo CD o Flux vengono spesso utilizzati per la gestione dei singoli cluster, mentre Fleet è progettato per gestire un ampio parco di GitOps cluster.

Aderendo a questi GitOps principi, Rancher Fleet offre una soluzione per le organizzazioni che desiderano implementare una gestione coerente, scalabile e automatizzata di applicazioni e risorse in un ambiente Kubernetes diversificato e su larga scala.

Per ulteriori informazioni, consulta la documentazione di Fleet.

Architettura

Per informazioni sull'architettura e sul flusso di lavoro, consulta il repository Fleet.

Codefresh

Codefresh è una CI/CD piattaforma moderna che supporta i GitOps principi, in particolare per le implementazioni di Kubernetes. Codefresh offre un set completo di funzionalità e le sue capacità sono notevoli. CI/CD GitOps

GitOps supporto

Area	funzionalità dello strumento
Git come unica fonte di verità	Codefresh utilizza i repository Git come fonte autorevole per il codice dell'applicazione, le definizioni dell'infrastruttura e le configurazioni della pipeline. Tutte le modifiche al sistema vengono apportate tramite Git, che garantisce una cronologia e un audit trail completi.
Configurazione dichiarativa	Codefresh supporta le definizioni dichiarative delle pipeline utilizzando file YAML archiviati in Git. I manifesti di Kubernetes, i grafici Helm, i CloudFormation modelli e altri file IAc possono essere controllati in base alla versione negli stessi repository.
GitOps dashboard	Codefresh fornisce una GitOps dashboard dedicata per la visualizzazione e la gestione dei flussi di lavoro. GitOps Offre una visione chiara dello stato di sincronizzazione tra Git e gli stati del cluster.
Sincronizzazione automatica	Codefresh monitora continuamente i repositor y Git per rilevare eventuali modifiche. Avvia automaticamente le pipeline per applicare le modifiche agli ambienti di destinazione quando rileva differenze.
Integrazione con Kubernetes	Codefresh offre una profonda integrazione con Kubernetes per supportare implement azioni in stile su più cluster. GitOps Supporta varie risorse Kubernetes e definizioni di risorse personalizzate (). CRDs

Area	funzionalità dello strumento
Gestione dell'ambiente	È possibile definire e gestire più ambienti (come sviluppo, staging e produzione) come codice. Codefresh supporta la promozione tra ambienti utilizzando pratiche. GitOps
Integrazione con Argo CD	Codefresh si integra con Argo CD per funzional ità avanzate. GitOps Combina le sue funzional ità CI con i punti di forza CD di Argo CD per fornire una soluzione completa. GitOps
Supporto Helm	Codefresh supporta i grafici Helm e fornisce una facile gestione di applicazioni complesse tramite. GitOps Offre inoltre il controllo delle versioni e la promozione dei grafici Helm.
Consegna progressiva	Codefresh supporta strategie di implement azione avanzate come canary e deployments. blue/green È possibile implementare e gestire queste strategie tramite flussi di lavoro. GitOps
Rollback e controllo delle versioni	Codefresh consente di ripristinare facilmente le versioni precedenti se vengono rilevati problemi dopo la distribuzione. Mantiene il controllo delle versioni di distribuzione per la tracciabilità.
Flussi di lavoro di approvazione	Codefresh supporta processi di approvazione manuali e automatizzati per le distribuzioni. Consente promozioni controllate tra ambienti, in conformità con le pratiche. GitOps
IaC	Codefresh supporta l'integrazione con strumenti IaC come CloudFormation Terraform. Consente il controllo della versione delle definizioni dell'infrastruttura insieme al codice dell'appl icazione.

Area	funzionalità dello strumento
Osservabilità e monitoraggio	Codefresh offre funzionalità integrate di monitoraggio e osservabilità. Offre inoltre integrazioni con strumenti di monitoraggio esterni per una maggiore visibilità.
Scansione di sicurezza	Codefresh include funzionalità di scansione di sicurezza che possono essere integrate nei GitOps flussi di lavoro. I controlli di sicurezza fanno parte del processo di implementazione automatizzato.
Audit trail	Codefresh mantiene registri di controllo completi per tutte le azioni e le modifiche. Supporta gli aspetti di tracciabilità e conformità di. GitOps
RBAC e controllo degli accessi	Codefresh implementa il controllo degli accessi basato sui ruoli (RBAC) per una gestione granulare delle autorizzazioni. Questo aiuta a garantire GitOps operazioni sicure tra team e ambienti.
Automazione di GitOps	Codefresh offre funzionalità per automatizzare vari aspetti dei GitOps flussi di lavoro, tra cui la creazione e la fusione di pull request (PR).
Implementazioni multi-cloud e ibride	Codefresh supporta i GitOps flussi di lavoro tra più provider cloud e ambienti locali.
Modelli e parametrizzazione	Codefresh supporta i modelli nelle configura zioni di pipeline e distribuzione. Ciò consente flussi di lavoro riutilizzabili e parametrizzati. GitOps

Area	funzionalità dello strumento
Gestione integrata delle immagini	Codefresh offre funzionalità integrate di gestione delle immagini dei container. Integra la creazione e l'implementazione di immagini nei flussi di lavoro. GitOps
GitOps per la gestione dei segreti	Codefresh offre modi sicuri per gestire i segreti all'interno GitOps dei flussi di lavoro. Si integra con soluzioni esterne di gestione dei segreti.
Funzionalità di collaborazione	Codefresh offre funzionalità per la collabora zione in team all'interno GitOps dei processi. Queste funzionalità includono commenti, notifiche e dashboard condivise.

L'approccio di Codefresh GitOps si distingue per l'integrazione delle funzionalità CI/CD con le pratiche. GitOps Mira a fornire una piattaforma completa che copra l'intero ciclo di vita della distribuzione del software rispettando i principi. GitOps

Il principale elemento di differenziazione di Codefresh in questo GitOps settore è il suo approccio alla piattaforma unificata, che combina funzionalità CI con CD e funzionalità. GitOps Ciò lo rende particolarmente adatto ai team che desiderano una all-in-one soluzione in grado di gestire CI/CD scenari complessi durante l'implementazione delle pratiche. GitOps

Codefresh offre una piattaforma per le organizzazioni che desiderano adottare GitOps metodologie in un CI/CD contesto più ampio, specialmente quando lavorano con Kubernetes e tecnologie native del cloud.

Per ulteriori informazioni, consulta la documentazione di Codefresh.

Pulumi

Pulumi è una piattaforma iAC che non è progettata esclusivamente per. GitOps Tuttavia, può essere utilizzato efficacemente per implementare GitOps i principi, in particolare per l'infrastruttura cloud e le implementazioni di Kubernetes.

GitOps supporto

Area	funzionalità dello strumento
IaC	Pulumi ti consente di definire la tua infrastru ttura utilizzando linguaggi di programmazione generici come Python e Go. TypeScript Questo approccio basato sul codice si allinea all'enfas i sulle configurazioni dichiarative e versionate GitOps .
Git come unica fonte di verità	Il codice dell'infrastruttura in Pulumi può essere archiviato e controllato dalla versione nei repository Git. Ciò garantisce che Git funga da unica fonte di verità per le definizioni dell'infr astruttura.
Stato dichiarativo desiderato	Sebbene Pulumi utilizzi linguaggi di programma zione, descrive comunque lo stato desiderato dell'infrastruttura in modo dichiarativo. Il codice definisce come dovrebbe essere l'infrastruttura, non il step-by-step processo per crearla.
Sincronizzazione automatica	Pulumi può essere integrato con CI/CD le pipeline per applicare automaticamente le modifiche quando il codice viene aggiornato in Git. Ciò consente l'implementazione continua delle modifiche all'infrastruttura, che è un principio chiave GitOps .
Supporto multi-cloud e Kubernetes	Pulumi supporta un'ampia gamma di provider di cloud e Kubernetes, quindi puoi seguire le pratiche in diversi ambienti. GitOps Lo strumento consente una gestione coerente delle risorse su diverse piattaforme.

Area	funzionalità dello strumento
Gestione dello stato	Pulumi gestisce lo stato dell'infrastruttura, che può essere archiviata in remoto e in modo sicuro. Questa gestione dello stato è fondament ale per GitOps le pratiche e garantisce la coerenza tra lo stato definito e lo stato effettivo dell'infrastruttura.
Rilevamento e riconciliazione delle derive	Pulumi è in grado di rilevare le differenze tra lo stato desiderato (nel codice) e lo stato effettivo dell'infrastruttura. Concilia queste differenze di allineamento con il principio della riconcili azione continua. GitOps
La politica come codice	È possibile definire e applicare le politiche come codice utilizzando Pulumi CrossGuard. Ciò consente una gestione controllata in base alla versione delle GitOps politiche di conformit à e sicurezza.
Gestione dei segreti	Pulumi offre modi sicuri per gestire le informazi oni sensibili all'interno del codice dell'infr astruttura. Supporta l'integrazione con sistemi di gestione dei segreti esterni, fondamentale per le pratiche GitOps di sicurezza.
Componenti modulari e riutilizzabili	Pulumi supporta la creazione di componenti e moduli riutilizzabili. Questa modularità è in linea con GitOps le pratiche per la gestione di implementazioni complesse e multiambiente.
Visualizza in anteprima e pianifica	Pulumi offre la possibilità di visualizzare in anteprima le modifiche prima di applicarle. Ciò supporta il GitOps principio di modifiche sicure e prevedibili all'infrastruttura.

Area	funzionalità dello strumento
Rollback e cronologia	Pulumi mantiene una cronologia delle implementazioni e supporta i rollback agli stati precedenti. Ciò è in linea con i GitOps principi di tracciabilità e reversibilità.
Distribuzione continua per l'infrastruttura	Pulumi può essere integrato nelle CI/CD pipeline per la fornitura continua delle modifiche all'infrastruttura. Supporta il test e la convalida automatizzati del codice dell'infrastruttura.
RBAC e controllo degli accessi	Pulumi fornisce il controllo degli accessi basato sui ruoli per la gestione di chi può apportare modifiche all'infrastruttura. Ciò supporta le pratiche GitOps di sicurezza e governance.
Osservabilità e registrazione	Pulumi offre funzionalità di registrazione e monitoraggio per le modifiche all'infrastruttura . Queste funzionalità supportano l'aspetto dell'osservabilità delle pratiche. GitOps
Integrazione con altri strumenti	Pulumi può integrarsi con vari strumenti nel cloud. Questa flessibilità consente flussi di GitOps lavoro completi.
Gestione dell'ambiente	Pulumi supporta la gestione di più ambienti (sviluppo, staging, produzione) utilizzando la stessa base di codice con configurazioni diverse. Ciò è in linea con le GitOps pratiche per una gestione coerente di più ambienti.
Gestione delle dipendenze	Pulumi gestisce le dipendenze tra le risorse e garantisce il corretto ordine delle operazioni. Questo è fondamentale per GitOps implement azioni complesse che coinvolgono componenti interdipendenti.

Area	funzionalità dello strumento
Fornitori di risorse personalizzate	Pulumi ti consente di creare provider personalizzati per gestire qualsiasi servizio basato su API. Ciò estende GitOps le pratiche a un'ampia gamma di risorse oltre alle offerte cloud standard.
Funzionalità di collaborazione	Pulumi supporta la collaborazione in team attraverso controlli di stato e accesso condivisi . Ciò facilita i GitOps flussi di lavoro negli ambienti di team.

Utilizzando queste funzionalità di Pulumi, le organizzazioni possono implementare GitOps pratiche per la propria infrastruttura, specialmente in scenari in cui necessitano di un controllo granulare o di una logica complessa, o desiderano gestire un set diversificato di risorse cloud e locali all'interno di un unico framework coerente.

L'approccio di Pulumi GitOps è unico perché offre la potenza e la flessibilità dei linguaggi di programmazione generici alla gestione dell'infrastruttura, rispettando al contempo i principi. GitOps Ciò può essere particolarmente vantaggioso per i team che preferiscono lavorare con linguaggi di programmazione familiari e desiderano applicare le migliori pratiche di ingegneria del software alla gestione dell'infrastruttura.

Il principale elemento di differenziazione di Pulumi in GitOps è l'uso di linguaggi di programmazione standard per definire l'infrastruttura. GitOps Gli strumenti tradizionali utilizzano spesso YAML o linguaggi specifici del dominio, mentre Pulumi consente una logica più complessa, un migliore riutilizzo del codice e una più facile integrazione con i flussi di lavoro di sviluppo esistenti.

Per ulteriori informazioni, consulta la documentazione di Pulumi.

GitOps confronto degli strumenti

Ecco un confronto tra i nove GitOps strumenti discussi nelle sezioni precedenti. Quando scegli uno strumento, considera i tuoi requisiti specifici, l'infrastruttura esistente, l'esperienza del team e il livello di controllo e personalizzazione desiderato.

Facilità d'uso

- Argo CD, Flux e Rancher Fleet sono generalmente più facili da configurare.
- Spinnaker e Jenkins X hanno curve di apprendimento più ripide.
- Weave GitOps potrebbe richiedere una maggiore configurazione per le funzionalità avanzate.
- GitLab CI/CD e Codefresh offrono esperienze integrate.

Integrazione con Kubernetes

- Argo CD, Flux e Rancher Fleet sono molto incentrati su Kubernetes.
- Jenkins X e Weave offrono funzionalità più ampie. GitOps DevOps
- Gli altri strumenti supportano Kubernetes senza concentrarsi esclusivamente su di esso.

Funzionalità CI/CD

- Jenkins X, soluzioni. GitLab CI/CD, and Codefresh offer complete CI/CD
- Argo CD, Flux e Weave GitOps si concentrano maggiormente sull'aspetto CD del flusso di lavoro e spesso richiedono l'integrazione con strumenti CI separati.

GitOps purezza

- Argo CD e Flux sono strumenti che si concentrano specificamente su. GitOps
- Gli altri strumenti incorporano GitOps principi a vari livelli.

Supporto multi-cloud

- Spinnaker e Pulumi eccellono in scenari multi-cloud.
- Gli altri strumenti possono funzionare su più cloud ma potrebbero richiedere una configurazione aggiuntiva.

Supporto multi-cluster

- Tutti gli strumenti supportano implementazioni multi-cluster.
- Argo CD e Weave GitOps dispongono di funzionalità di gestione multi-cluster più avanzate.

Integrazione

- Flux gode del forte sostegno della Cloud Native Computing Foundation (CNCF).
- Argo CD ha una comunità ampia e attiva.
- Argo CD e Flux hanno una forte integrazione con Kubernetes.
- Jenkins X utilizza il più ampio sistema Jenkins.
- Weave GitOps è più recente ma sta crescendo con un forte sostegno commerciale.
- GitLab CI/CD si integra perfettamente con. GitLab
- Rancher Fleet funziona bene all'interno del sistema Rancher.

Comunità e supporto

- Flux ha un forte sostegno da parte del CNCF.
- Argo CD e GitLab Spinnaker hanno grandi comunità.
- Il supporto commerciale è disponibile per la maggior parte degli strumenti.

Funzionalità aziendali

- Per impostazione predefinita, Weave GitOps e Jenkins X offrono funzionalità più incentrate sulle aziende.
- Argo CD e Flux offrono offerte aziendali o possono essere estese per uso aziendale.

Flessibilità ed estensibilità

- Flux è altamente modulare ed estensibile.
- Argo CD offre buone opzioni di personalizzazione.
- Jenkins X è molto estensibile ma potrebbe richiedere uno sforzo maggiore.
- Weave GitOps mira a fornire una soluzione completa con meno necessità di estensibilità.

Scalabilità

- Spinnaker e GitLab CI/CD sono noti per la scalabilità aziendale.
- Argo CD e Flux gestiscono bene le implementazioni Kubernetes su larga scala.

Gestione dell'infrastruttura

- Pulumi si concentra sulla gestione dell'infrastruttura.
- Weave GitOps e Flux offrono buone funzionalità IAc.

Modello di programmazione e supporto linguistico

- In Pulumi, puoi definire l'infrastruttura utilizzando linguaggi di programmazione generici come Python, Go TypeScript, C# e Java. L'uso di linguaggi standard da parte di Pulumi consente l'integrazione del codice dell'infrastruttura con flussi di lavoro di sviluppo, pratiche di test e logiche complesse familiari.
- Terraform utilizza il linguaggio di HashiCorp configurazione (HCL).
- CloudFormation utilizza modelli JSON e YAML.
- Argo CD, Flux, Rancher Fleet, Weave, Spinnaker e CI/CD gestiscono GitOps principalmente YAML o file di configurazione dichiarativi GitLab .
- Jenkins X gestisce YAML e pipeline basate su script, ma non offre in modo nativo una programmazione generica per IaC.

Casi d'uso di Argo CD e Flux

Questa sezione si concentra su due strumenti, Argo CD e Flux, che forniscono funzionalità allo stato puro. GitOps In questo contesto, pure GitOps si riferisce a un modello in cui un repository Git funge da unica fonte di verità per lo stato desiderato delle applicazioni e dell'infrastruttura. Tutte le modifiche vengono apportate tramite commit Git e il sistema sincronizza automaticamente l'ambiente live in modo che corrisponda allo stato definito nel repository. Non è richiesto alcun intervento manuale al di fuori delle operazioni Git.

Considerazioni generali

- Potresti preferire utilizzare Argo CD in ambienti in cui la gestione visiva e i flussi di lavoro incentrati sulle applicazioni sono importanti.
- Puoi scegliere Flux se hai bisogno di soluzioni più leggere, una solida multi-tenancy o una profonda integrazione con la più ampia rete Cloud Native Computing Foundations (CNCF).
- Argo CD piace spesso ai team che stanno passando dal CI/CD tradizionale a grazie alla sua interfaccia utente intuitiva. GitOps
- Flux è spesso preferito negli ambienti nativi del cloud in cui i flussi di lavoro basati su CLI e le pratiche IaC sono già consolidati.

In definitiva, la scelta tra Argo CD e Flux dipende spesso dalle esigenze organizzative specifiche, dagli strumenti esistenti e dalle preferenze del team. Entrambi gli strumenti sono in grado di gestire la maggior parte GitOps degli scenari, quindi consigliamo di valutarli in base ai casi d'uso e ai requisiti specifici.

Casi d'uso di Argo CD

Gestione visiva:

- Quando è necessaria un'interfaccia utente intuitiva per gestire le distribuzioni e visualizzare gli stati delle applicazioni.
- Per i team che preferiscono un'interfaccia grafica per il monitoraggio e la risoluzione dei problemi.

Approccio incentrato sulle applicazioni:

- Quando si desidera gestire le implementazioni a livello di applicazione anziché gestire le singole risorse.
- Per le organizzazioni che strutturano le proprie implementazioni in base a concetti applicativi.

Gestione multicluster:

- Quando la gestione delle distribuzioni su più cluster è un requisito primario.
- Per ambienti complessi e distribuiti con molti cluster.

Onde di rollback e sincronizzazione:

- Quando è necessario un controllo dettagliato sul processo di implementazione, comprese le ondate di sincronizzazione e gli interventi manuali.
- Per scenari che richiedono strategie di rollback complesse.

Integrazione con gli strumenti esistenti:

- Quando utilizzi già altri strumenti del progetto Argo come Argo Workflows e Argo Events.

Ambienti aziendali:

- Per le grandi aziende che necessitano di una solida integrazione RBAC e Single Sign-On come impostazione predefinita.

Casi d'uso di Flux

Implementazioni leggere:

- Quando hai bisogno di una soluzione più leggera e meno dispendiosa in termini di risorse GitOps.
- Per scenari di edge computing o IoT in cui le risorse potrebbero essere limitate.

Aggiornamenti automatici delle immagini:

- Quando il rilevamento e l'implementazione automatici di nuove immagini di container sono un requisito fondamentale.

- Per i team che si concentrano sull'implementazione continua con aggiornamenti frequenti delle immagini.

Multi-tenancy:

- Quando è necessario un forte supporto multi-tenancy, specialmente in ambienti cluster condivisi.
- Per fornitori di servizi o grandi organizzazioni che prevedono rigide separazioni tra team o progetti.

IAC:

- Quando si gestiscono sia le applicazioni che l'infrastruttura attraverso lo stesso GitOps flusso di lavoro è importante.
- Per i team che hanno investito molto nel paradigma IaC.

Integrazione Helm:

- Quando l'uso estensivo dei grafici Helm fa parte della strategia di implementazione.
- Per ambienti con implementazioni complesse basate su Helm.

Integrazione del progetto CNCF:

- Quando è importante una stretta integrazione con altri progetti CNCF.
- Per le organizzazioni che si allineano alle tecnologie e ai principi del CNCF.

Architettura modulare:

- Quando è necessaria la flessibilità necessaria per utilizzare solo componenti specifici del GitOps toolkit.
- Per i team che desiderano creare GitOps flussi di lavoro personalizzati utilizzando componenti modulari.

Distribuzione progressiva:

- Quando le strategie di implementazione avanzate come le versioni Canary o A/B i test sono requisiti fondamentali.

Confronto delle funzionalità

Area	CD Argo	Flusso
Support per GitOps i principi fondamentali	 S	 Sì
Architettura	End-to-end applicazione per l'implementazione dei flussi di lavoro Kubernetes GitOps	Fornisce Kubernetes e controller per CRDs GitOps
Configurazione	Semplice	Complesso
Supporto per il timone	 S	 Sì
Personalizza il supporto	 S	 Sì
GUI integrata	CLI e interfaccia utente web completa	CLI e interfaccia web leggera opzionale
Supporto RBAC	Controllo granulare	RBAC nativo per Kubernetes
Supporto multi-tenancy e multi-cluster	Supporto eccellente per più cluster	Eccellente supporto per la multi-tenancy
Autenticazione Single Sign-On	 S	 Sì
Automazione della sincroniz zazione	Possibilità di sincronizzare le finestre	Possibilità di impostare intervalli di riconciliazione

Area	CD Argo	Flusso
Sincronizzazione parziale	 S	 No
Processo di riconciliazione	Supporta sincronizzazioni manuali e automatiche. Sono disponibili diverse strategie.	Supporta sincronizzazioni manuali e automatiche.
Estensibilità	Supporta plugin personalizzati. Opzioni di personalizzazione limitate.	Supporta controller personalizzati. Buona estensibilità e integrazioni di terze parti.
Supporto comunitario	Comunità ampia e attiva.	Comunità più piccola ma in crescita.
Scalabilità	Buona scalabilità, ma limitata dalla velocità di recupero dei dati dell'interfaccia utente Web. L'analisi comunitaria suggerisce il supporto per decine di migliaia di applicazioni.	Guide chiare per la scalabilità orizzontale e verticale, fino a decine di migliaia di applicazioni.

Le migliori pratiche per la scelta di uno GitOps strumento

Questa sezione fornisce considerazioni, suggerimenti e best practice per la scelta di uno GitOps strumento per il cluster EKS. La scelta giusta dipende dal contesto specifico, dai requisiti e dalla strategia a lungo termine. Spesso è utile condurre un proof of concept con le proprie scelte principali prima di prendere una decisione definitiva.

Valuta le esigenze e le capacità della tua organizzazione:

- Considerate l'attuale set di competenze e la disponibilità del vostro team a imparare nuovi strumenti.
- Valuta la complessità del tuo ambiente Amazon EKS. (Ad esempio, stai utilizzando un singolo cluster o più cluster?)
- Determina i tuoi requisiti specifici di conformità, sicurezza e scalabilità.

> **Best practice**
>
> Crea un documento dettagliato sui requisiti che descriva le funzionalità richieste e le funzionalità utili, ma non obbligatorie.

Valuta la maturità e l'adozione degli strumenti:

- Ricerca la maturità dei potenziali GitOps strumenti e i relativi tassi di adozione nel settore.
- Cerca strumenti che abbiano una comprovata esperienza negli ambienti Amazon EKS.

> **Best practice**
>
> Dai priorità agli strumenti che sono stati ampiamente adottati e che hanno una forte presenza nella rete della Cloud Native Computing Foundation (CNCF).

Prendi in considerazione l'integrazione con la tua toolchain esistente:

- Valuta quanto bene lo GitOps strumento si integra con la tua CI/CD pipeline attuale, le soluzioni di monitoraggio e altri strumenti operativi.
- Cerca integrazioni native con Servizi AWS IAM, Amazon ECR e. CloudWatch

> **Best practice**
>
> Crea un proof of concept per testare le funzionalità di integrazione prima di prendere una decisione definitiva.

Valuta le funzionalità di sicurezza:

- Dai priorità agli strumenti che dispongono di solide funzionalità di controllo degli accessi basato sui ruoli (RBAC) e si integrano bene con IAM.
- Cerca funzionalità che supportino la gestione sicura dei segreti e l'applicazione delle policy.

> **Best practice**
>
> Scegli uno strumento che supporti pratiche di sicurezza GitOps basate su criteri di sicurezza, tra cui policy come codice e controlli di conformità automatizzati.

Valuta la scalabilità e le prestazioni:

- Considerate le prestazioni dello strumento con un gran numero di applicazioni e cluster.
- Valuta il suo impatto sulle prestazioni del cluster e sul consumo di risorse.

> **Best practice**
>
> Esegui test delle prestazioni con carichi di lavoro simili al tuo ambiente di produzione per assicurarti che lo strumento sia in grado di gestire la tua scalabilità.

Prendi in considerazione il supporto multi-cluster e multiambiente:

- Se disponi o prevedi di avere più cluster EKS, dai la priorità agli strumenti che dispongono di solide capacità di gestione multicluster.
- Cerca funzionalità che supportino implementazioni coerenti in ambienti diversi (come sviluppo, staging e produzione).

> **Best practice**
>
> Scegli uno strumento che consenta la gestione centralizzata di più cluster mantenendo configurazioni specifiche dell'ambiente.

Valuta l'osservabilità e le capacità di monitoraggio:

- Cerca strumenti che offrano una chiara visibilità sullo stato delle tue implementazioni e sullo stato di salute del cluster.
- Considerate quanto bene lo strumento si integra con le vostre soluzioni di monitoraggio e registrazione esistenti.

> **Best practice**
>
> Dai priorità agli strumenti che offrono dashboard personalizzabili e meccanismi di avviso per il rilevamento proattivo dei problemi.

Valuta la curva di apprendimento e la documentazione:

- Valuta la qualità e la completezza della documentazione dello strumento.
- Valuta la disponibilità di risorse di formazione e il supporto della comunità.

> **Best practice**
>
> Scegli uno strumento con documentazione ben tenuta, forum di community attivi e programmi di formazione o certificazioni ufficiali.

Prendi in considerazione i costi e l'utilizzo delle risorse:

- Valuta sia i costi diretti (come licenze e supporto) che i costi indiretti (come le spese generali operative e i costi di formazione) legati all'adozione dello strumento.
- Valuta l'efficienza dello strumento in termini di consumo di risorse di calcolo e storage.

> **Best practice**
>
> Esegui un'analisi del costo totale di proprietà (TCO) che includa sia i costi a breve che a lungo termine.

Valuta la flessibilità e le opzioni di personalizzazione:

- Cerca strumenti che ti consentano di personalizzare i flussi di lavoro in base alle tue esigenze specifiche.
- Considerate l'estensibilità dello strumento tramite plugin o. APIs

> **Best practice**
>
> Scegliete uno strumento che bilanci le funzionalità predefinite con la possibilità di personalizzazione in base alle vostre esigenze specifiche.

Valuta le funzionalità di distribuzione continua e progressiva:

- Cerca strumenti che supportino strategie di implementazione avanzate come le versioni e le blue/green implementazioni Canary.
- Valuta la facilità di implementazione e gestione di queste strategie.

> **Best practice**
>
> Dai priorità agli strumenti che offrono supporto integrato per modelli di distribuzione progressivi per ridurre al minimo i rischi nelle tue implementazioni.

Prendi in considerazione il legame con il fornitore e la portabilità:

- Valuta le dipendenze dello strumento da specifici provider o tecnologie cloud.
- Considerate la facilità di migrazione a uno strumento diverso in futuro, se necessario.

> **Best practice**
>
> Preferisci strumenti che utilizzano standard aperti e forniscono funzionalità di esportazione per le tue GitOps configurazioni.

Valuta il supporto e le estensioni della community:

- Guarda le dimensioni e l'attività della comunità di utenti.
- Valuta la disponibilità di integrazioni e plugin di terze parti.

> **Best practice**
>
> Partecipa ai forum o ai gruppi di utenti della community per ottenere esperienze di prima mano da altri utenti prima di prendere una decisione.

Prendi in considerazione i requisiti di conformità e audit:

- Valuta in che modo lo strumento supporta le tue esigenze di conformità, compresi gli audit trail e la reportistica.
- Cerca funzionalità che aiutino a mantenere e dimostrare la conformità.

> **Best practice**
>
> Scegli uno strumento che fornisca registri di controllo completi e supporti la generazione di report di conformità.

Valuta le capacità di rollback e disaster recovery:

- Valuta la facilità e l'affidabilità dei meccanismi di rollback.
- Considerate in che modo lo strumento supporta gli scenari di disaster recovery.

Best practice

Testa a fondo i processi di rollback e ripristino come parte della tua valutazione.

Domande frequenti

D: Quali sono gli GitOps strumenti più diffusi per Amazon EKS?

R: Gli GitOps strumenti più diffusi per Amazon EKS includono Argo CD, Flux, Jenkins X e CI/CD. GitLab Ogni strumento ha dei punti di forza, ma Argo CD e Flux sono particolarmente apprezzati per il loro approccio nativo a Kubernetes e il forte supporto della community.

D: In che modo migliora la gestione dei cluster EKS? GitOps

R: GitOps migliora la gestione dei cluster EKS fornendo il controllo delle versioni per l'infrastruttura, implementazioni automatizzate, maggiore sicurezza attraverso configurazioni dichiarative, rollback più semplici e una migliore verificabilità. Inoltre, migliora la collaborazione e riduce l'errore umano nelle implementazioni.

D: Quali caratteristiche principali devo cercare in uno GitOps strumento per Amazon EKS?

R: Le caratteristiche principali da cercare includono: perfetta integrazione con Amazon EKS, robusto RBAC, supporto multi-cluster, funzionalità di osservabilità, supporto per strategie di distribuzione progressive, scalabilità e integrazione con sistemi Servizi AWS come IAM e Amazon ECR.

D: Come posso garantire la sicurezza durante l'implementazione GitOps in Amazon EKS?

R: Per garantire la sicurezza, scegli uno strumento che abbia una forte integrazione RBAC con IAM, una gestione sicura dei segreti, il supporto per i repository Git crittografati e la capacità di implementare politiche di sicurezza come codice. Inoltre, verifica che lo strumento fornisca registri di controllo completi.

D: GitOps Gli strumenti possono gestire ambienti Amazon EKS multi-cluster?

R: Sì, GitOps strumenti come Argo CD e Flux offrono solide funzionalità di gestione multi-cluster. Consentono di gestire più cluster EKS da un unico piano di controllo, garantendo la coerenza tra gli ambienti.

D: In che modo GitOps gli strumenti si integrano con le pipeline CI/CD esistenti?

R: GitOps Gli strumenti in genere si integrano con le pipeline CI/CD esistenti fungendo da fase di implementazione della pipeline. Possono essere attivati dagli strumenti CI quando le modifiche vengono inviate al repository Git e automatizzano il processo di distribuzione nei cluster EKS.

D: Quali sono le sfide dell'implementazione GitOps in Amazon EKS?

R: Le sfide più comuni includono la gestione sicura dei segreti, la garanzia di controlli di accesso adeguati, la gestione delle applicazioni stateful, la gestione della deriva tra Git e lo stato del cluster e l'adattamento dei flussi di lavoro del team al modello. GitOps

D: In che modo GitOps gli strumenti gestiscono i rollback in Amazon EKS?

R: GitOps gli strumenti in genere gestiscono i rollback ripristinando un commit precedente nel repository Git. Ciò attiva automaticamente l'implementazione del precedente stato di buono noto, che si traduce in rollback rapidi e affidabili.

D: GitOps Gli strumenti possono gestire i componenti aggiuntivi di Amazon EKS e altre AWS risorse?

R: Molti GitOps strumenti possono gestire i componenti aggiuntivi di Amazon EKS e alcune AWS risorse, specialmente se combinati con strumenti IaC come Terraform o. CloudFormation Tuttavia, la portata di questa funzionalità può variare; consulta la sezione GitOps strumenti per informazioni specifiche su ogni strumento.

D: In che modo gli GitOps strumenti supportano i requisiti di conformità in Amazon EKS?

R: GitOps Gli strumenti supportano la conformità fornendo una chiara traccia di controllo di tutte le modifiche, applicando i processi di approvazione, implementando le politiche come codice per i controlli automatici di conformità e offrendo funzionalità di registrazione e reporting dettagliate.

D: Qual è la curva di apprendimento per l'implementazione GitOps in Amazon EKS?

R: La curva di apprendimento può variare a seconda dello strumento e delle conoscenze esistenti del team. In genere, i team che conoscono Git, Kubernetes e Amazon EKS si adatteranno più rapidamente di altri. Gli strumenti più diffusi offrono un'ampia documentazione e risorse di formazione per facilitarne l'adozione.

D: In che modo GitOps gli strumenti gestiscono la gestione dei segreti in Amazon EKS?

R: GitOps Gli strumenti in genere si integrano con soluzioni esterne di gestione dei segreti come Gestione dei segreti AWS o HashiCorp Vault. Alcuni strumenti offrono anche la crittografia integrata per i segreti archiviati negli archivi Git.

D: GitOps Gli strumenti possono funzionare con applicazioni stateless e stateful in Amazon EKS?

R: Sì, GitOps gli strumenti possono funzionare sia con applicazioni stateless che stateless. Tuttavia, la gestione delle applicazioni con stato richiede spesso considerazioni aggiuntive, come la gestione dei volumi persistenti e la garanzia della coerenza dei dati durante gli aggiornamenti.

D: In che modo gli GitOps strumenti supportano le blue/green distribuzioni Canary or in Amazon EKS?

R: Molti GitOps strumenti offrono supporto integrato per strategie di distribuzione avanzate. Possono gestire l'implementazione graduale di nuove versioni, monitorare i problemi e ripristinare automaticamente se vengono rilevati problemi. Tutte queste operazioni sono definite come codice nel repository Git.

D: Qual è la differenza tra l'utilizzo di uno GitOps strumento e l'utilizzo **`kubectl apply`** con una CI/CD pipeline?

R: GitOps Gli strumenti offrono vantaggi rispetto ai `kubectl apply` comandi semplici, tra cui il rilevamento e la riconciliazione automatizzati delle deviazioni, una maggiore sicurezza grazie alle implementazioni basate su pull, una migliore verificabilità e strategie di implementazione più sofisticate. Forniscono inoltre un approccio più completo alla gestione dell'intero stato del cluster.

Risorse

Le seguenti risorse forniscono documentazione ufficiale, guide pratiche, case study e analisi approfondite che possono aiutarvi a prendere una decisione informata nella scelta di uno GitOps strumento per il vostro cluster EKS. Esse coprono vari aspetti GitOps, tra cui le strategie di implementazione, le migliori pratiche, i confronti tra diversi strumenti e le esperienze del mondo reale.

AWS risorse:

- Documentazione Amazon EKS
- Automazione di Amazon EKS con GitOps (post AWS sul blog)
- Introduzione a GitOps EKS con Weaveworks (workshop)AWS
- Laboratorio Flux (workshop Amazon EKS)
- Laboratorio CD Argo (laboratorio Amazon EKS)

GitOps e documentazione sugli strumenti:

- GitOps Migliori pratiche per l'implementazione continua e la sicurezza progressiva (webinar DevOps on-demand.com)
- Documentazione Kubernetes
- Documentazione su Argo CD
- Documentazione Flux
- Documentazione Weave GitOps
- Documentazione Jenkins X
- GitLab CI/CD documentazione
- documentazione Spinnaker
- Documentazione Rancher Fleet
- Documentazione Codefresh

Cronologia dei documenti

La tabella seguente descrive le modifiche significative apportate a questa guida. Per ricevere notifiche sugli aggiornamenti futuri, puoi abbonarti a un feed RSS.

Modifica	Descrizione	Data
Pubblicazione iniziale	—	30 aprile 2025

AWS Glossario delle linee guida prescrittive

I seguenti sono termini di uso comune nelle strategie, nelle guide e nei modelli forniti da AWS Prescriptive Guidance. Per suggerire voci, utilizza il link Fornisci feedback alla fine del glossario.

Numeri

7 R

Sette strategie di migrazione comuni per trasferire le applicazioni sul cloud. Queste strategie si basano sulle 5 R identificate da Gartner nel 2011 e sono le seguenti:

- Rifattorizzare/riprogettare: trasferisci un'applicazione e modifica la sua architettura sfruttando appieno le funzionalità native del cloud per migliorare l'agilità, le prestazioni e la scalabilità. Ciò comporta in genere la portabilità del sistema operativo e del database. Esempio: migra il tuo database Oracle locale all'edizione compatibile con Amazon Aurora PostgreSQL.
- Ridefinire la piattaforma (lift and reshape): trasferisci un'applicazione nel cloud e introduci un certo livello di ottimizzazione per sfruttare le funzionalità del cloud. Esempio: migra il tuo database Oracle locale ad Amazon Relational Database Service (Amazon RDS) per Oracle in. Cloud AWS
- Riacquistare (drop and shop): passa a un prodotto diverso, in genere effettuando la transizione da una licenza tradizionale a un modello SaaS. Esempio: migra il tuo sistema di gestione delle relazioni con i clienti (CRM) su Salesforce.com.
- Eseguire il rehosting (lift and shift): trasferisci un'applicazione sul cloud senza apportare modifiche per sfruttare le funzionalità del cloud. Esempio: migra il tuo database Oracle locale a Oracle su un'istanza EC2 in. Cloud AWS
- Trasferire (eseguire il rehosting a livello hypervisor): trasferisci l'infrastruttura sul cloud senza acquistare nuovo hardware, riscrivere le applicazioni o modificare le operazioni esistenti. Esegui la migrazione dei server da una piattaforma locale a un servizio cloud per la stessa piattaforma. Esempio: migra un'applicazione suMicrosoft Hyper-V. AWS
- Riesaminare (mantenere): mantieni le applicazioni nell'ambiente di origine. Queste potrebbero includere applicazioni che richiedono una rifattorizzazione significativa che desideri rimandare a un momento successivo e applicazioni legacy che desideri mantenere, perché non vi è alcuna giustificazione aziendale per effettuarne la migrazione.
- Ritirare: disattiva o rimuovi le applicazioni che non sono più necessarie nell'ambiente di origine.

A

ABAC

Vedi controllo degli accessi basato sugli attributi.

servizi astratti

Vedi servizi gestiti.

ACIDO

Vedi atomicità, consistenza, isolamento, durata.

migrazione attiva-attiva

Un metodo di migrazione del database in cui i database di origine e di destinazione vengono mantenuti sincronizzati (utilizzando uno strumento di replica bidirezionale o operazioni di doppia scrittura) ed entrambi i database gestiscono le transazioni provenienti dalle applicazioni di connessione durante la migrazione. Questo metodo supporta la migrazione in piccoli batch controllati anziché richiedere una conversione una tantum. È più flessibile ma richiede più lavoro rispetto alla migrazione attiva-passiva.

migrazione attiva-passiva

Un metodo di migrazione del database in cui i database di origine e di destinazione vengono mantenuti sincronizzati, ma solo il database di origine gestisce le transazioni provenienti dalle applicazioni di connessione mentre i dati vengono replicati nel database di destinazione. Il database di destinazione non accetta alcuna transazione durante la migrazione.

funzione di aggregazione

Una funzione SQL che opera su un gruppo di righe e calcola un singolo valore restituito per il gruppo. Esempi di funzioni aggregate includono `SUM` e. `MAX`

Intelligenza artificiale

Vedi intelligenza artificiale.

AIOps

Guarda le operazioni di intelligenza artificiale.

anonimizzazione

Il processo di eliminazione permanente delle informazioni personali in un set di dati. L'anonimizzazione può aiutare a proteggere la privacy personale. I dati anonimi non sono più considerati dati personali.

anti-modello

Una soluzione utilizzata frequentemente per un problema ricorrente in cui la soluzione è controproducente, inefficace o meno efficace di un'alternativa.

controllo delle applicazioni

Un approccio alla sicurezza che consente l'uso solo di applicazioni approvate per proteggere un sistema dal malware.

portfolio di applicazioni

Una raccolta di informazioni dettagliate su ogni applicazione utilizzata da un'organizzazione, compresi i costi di creazione e manutenzione dell'applicazione e il relativo valore aziendale. Queste informazioni sono fondamentali per il processo di scoperta e analisi del portfolio e aiutano a identificare e ad assegnare la priorità alle applicazioni da migrare, modernizzare e ottimizzare.

intelligenza artificiale (IA)

Il campo dell'informatica dedicato all'uso delle tecnologie informatiche per svolgere funzioni cognitive tipicamente associate agli esseri umani, come l'apprendimento, la risoluzione di problemi e il riconoscimento di schemi. Per ulteriori informazioni, consulta la sezione Che cos'è l'intelligenza artificiale?

operazioni di intelligenza artificiale (AIOps)

Il processo di utilizzo delle tecniche di machine learning per risolvere problemi operativi, ridurre gli incidenti operativi e l'intervento umano e aumentare la qualità del servizio. Per ulteriori informazioni su come AIOps viene utilizzata nella strategia di AWS migrazione, consulta la guida all'integrazione delle operazioni.

crittografia asimmetrica

Un algoritmo di crittografia che utilizza una coppia di chiavi, una chiave pubblica per la crittografia e una chiave privata per la decrittografia. Puoi condividere la chiave pubblica perché non viene utilizzata per la decrittografia, ma l'accesso alla chiave privata deve essere altamente limitato.

atomicità, consistenza, isolamento, durabilità (ACID)

Un insieme di proprietà del software che garantiscono la validità dei dati e l'affidabilità operativa di un database, anche in caso di errori, interruzioni di corrente o altri problemi.

Controllo degli accessi basato su attributi (ABAC)

La pratica di creare autorizzazioni dettagliate basate su attributi utente, come reparto, ruolo professionale e nome del team. Per ulteriori informazioni, consulta ABAC AWS nella documentazione AWS Identity and Access Management (IAM).

fonte di dati autorevole

Una posizione in cui è archiviata la versione principale dei dati, considerata la fonte di informazioni più affidabile. È possibile copiare i dati dalla fonte di dati autorevole in altre posizioni allo scopo di elaborarli o modificarli, ad esempio anonimizzandoli, oscurandoli o pseudonimizzandoli.

Zona di disponibilità

Una posizione distinta all'interno di un edificio Regione AWS che è isolata dai guasti in altre zone di disponibilità e offre una connettività di rete economica e a bassa latenza verso altre zone di disponibilità nella stessa regione.

AWS Cloud Adoption Framework (CAF)AWS

Un framework di linee guida e best practice AWS per aiutare le organizzazioni a sviluppare un piano efficiente ed efficace per passare con successo al cloud. AWS CAF organizza le linee guida in sei aree di interesse chiamate prospettive: business, persone, governance, piattaforma, sicurezza e operazioni. Le prospettive relative ad azienda, persone e governance si concentrano sulle competenze e sui processi aziendali; le prospettive relative alla piattaforma, alla sicurezza e alle operazioni si concentrano sulle competenze e sui processi tecnici. Ad esempio, la prospettiva relativa alle persone si rivolge alle parti interessate che gestiscono le risorse umane (HR), le funzioni del personale e la gestione del personale. In questa prospettiva, AWS CAF fornisce linee guida per lo sviluppo delle persone, la formazione e le comunicazioni per aiutare a preparare l'organizzazione all'adozione del cloud di successo. Per ulteriori informazioni, consulta il sito web di AWS CAF e il white paper AWS CAF.

AWS Workload Qualification Framework (WQF)AWS

Uno strumento che valuta i carichi di lavoro di migrazione dei database, consiglia strategie di migrazione e fornisce stime del lavoro. AWS WQF è incluso in (). AWS Schema Conversion Tool AWS SCT Analizza gli schemi di database e gli oggetti di codice, il codice dell'applicazione, le dipendenze e le caratteristiche delle prestazioni e fornisce report di valutazione.

B

bot difettoso

Un bot che ha lo scopo di interrompere o causare danni a individui o organizzazioni.

BCP

Vedi la pianificazione della continuità operativa.

grafico comportamentale

Una vista unificata, interattiva dei comportamenti delle risorse e delle interazioni nel tempo. Puoi utilizzare un grafico comportamentale con Amazon Detective per esaminare tentativi di accesso non riusciti, chiamate API sospette e azioni simili. Per ulteriori informazioni, consulta Dati in un grafico comportamentale nella documentazione di Detective.

sistema big-endian

Un sistema che memorizza per primo il byte più importante. Vedi anche endianness.

Classificazione binaria

Un processo che prevede un risultato binario (una delle due classi possibili). Ad esempio, il modello di machine learning potrebbe dover prevedere problemi come "Questa e-mail è spam o non è spam?" o "Questo prodotto è un libro o un'auto?"

filtro Bloom

Una struttura di dati probabilistica ed efficiente in termini di memoria che viene utilizzata per verificare se un elemento fa parte di un set.

implementazione blu/verde

Una strategia di implementazione in cui si creano due ambienti separati ma identici. La versione corrente dell'applicazione viene eseguita in un ambiente (blu) e la nuova versione dell'applicazione nell'altro ambiente (verde). Questa strategia consente di ripristinare rapidamente il sistema con un impatto minimo.

bot

Un'applicazione software che esegue attività automatizzate su Internet e simula l'attività o l'interazione umana. Alcuni bot sono utili o utili, come i web crawler che indicizzano le informazioni su Internet. Alcuni altri bot, noti come bot dannosi, hanno lo scopo di disturbare o causare danni a individui o organizzazioni.

botnet

Reti di bot infettate da malware e controllate da un'unica parte, nota come bot herder o bot operator. Le botnet sono il meccanismo più noto per scalare i bot e il loro impatto.

ramo

Un'area contenuta di un repository di codice. Il primo ramo creato in un repository è il ramo principale. È possibile creare un nuovo ramo a partire da un ramo esistente e quindi sviluppare funzionalità o correggere bug al suo interno. Un ramo creato per sviluppare una funzionalità viene comunemente detto ramo di funzionalità. Quando la funzionalità è pronta per il rilascio, il ramo di funzionalità viene ricongiunto al ramo principale. Per ulteriori informazioni, consulta Informazioni sulle filiali (documentazione). GitHub

accesso break-glass

In circostanze eccezionali e tramite una procedura approvata, un mezzo rapido per consentire a un utente di accedere a un sito a Account AWS cui in genere non dispone delle autorizzazioni necessarie. Per ulteriori informazioni, vedere l'indicatore Implementate break-glass procedures nella guida Well-Architected AWS .

strategia brownfield

L'infrastruttura esistente nell'ambiente. Quando si adotta una strategia brownfield per un'architettura di sistema, si progetta l'architettura in base ai vincoli dei sistemi e dell'infrastruttura attuali. Per l'espansione dell'infrastruttura esistente, è possibile combinare strategie brownfield e greenfield.

cache del buffer

L'area di memoria in cui sono archiviati i dati a cui si accede con maggiore frequenza.

capacità di business

Azioni intraprese da un'azienda per generare valore (ad esempio vendite, assistenza clienti o marketing). Le architetture dei microservizi e le decisioni di sviluppo possono essere guidate dalle capacità aziendali. Per ulteriori informazioni, consulta la sezione Organizzazione in base alle funzionalità aziendali del whitepaper Esecuzione di microservizi containerizzati su AWS.

pianificazione della continuità operativa (BCP)

Un piano che affronta il potenziale impatto di un evento che comporta l'interruzione dell'attività, come una migrazione su larga scala, sulle operazioni e consente a un'azienda di riprendere rapidamente le operazioni.

C

CAF

Vedi Cloud Adoption AWS Framework.

implementazione canaria

Il rilascio lento e incrementale di una versione agli utenti finali. Quando sei sicuro, distribuisci la nuova versione e sostituisci la versione corrente nella sua interezza.

CCoE

Vedi Cloud Center of Excellence.

CDC

Vedi Change Data Capture.

Change Data Capture (CDC)

Il processo di tracciamento delle modifiche a un'origine dati, ad esempio una tabella di database, e di registrazione dei metadati relativi alla modifica. È possibile utilizzare CDC per vari scopi, ad esempio il controllo o la replica delle modifiche in un sistema di destinazione per mantenere la sincronizzazione.

ingegneria del caos

Introduzione intenzionale di guasti o eventi dirompenti per testare la resilienza di un sistema. Puoi usare AWS Fault Injection Service (AWS FIS) per eseguire esperimenti che stressano i tuoi AWS carichi di lavoro e valutarne la risposta.

CI/CD

Vedi integrazione continua e distribuzione continua.

classificazione

Un processo di categorizzazione che aiuta a generare previsioni. I modelli di ML per problemi di classificazione prevedono un valore discreto. I valori discreti sono sempre distinti l'uno dall'altro. Ad esempio, un modello potrebbe dover valutare se in un'immagine è presente o meno un'auto.

crittografia lato client

Crittografia dei dati a livello locale, prima che il destinatario li Servizio AWS riceva.

Centro di eccellenza cloud (CCoE)

Un team multidisciplinare che guida le iniziative di adozione del cloud in tutta l'organizzazione, tra cui lo sviluppo di best practice per il cloud, la mobilitazione delle risorse, la definizione delle tempistiche di migrazione e la guida dell'organizzazione attraverso trasformazioni su larga scala. Per ulteriori informazioni, consulta gli CCoE post sull' Cloud AWS Enterprise Strategy Blog.

cloud computing

La tecnologia cloud generalmente utilizzata per l'archiviazione remota di dati e la gestione dei dispositivi IoT. Il cloud computing è generalmente collegato alla tecnologia di edge computing.

modello operativo cloud

In un'organizzazione IT, il modello operativo utilizzato per creare, maturare e ottimizzare uno o più ambienti cloud. Per ulteriori informazioni, consulta Building your Cloud Operating Model.

fasi di adozione del cloud

Le quattro fasi che le organizzazioni in genere attraversano quando migrano verso Cloud AWS:

- Progetto: esecuzione di alcuni progetti relativi al cloud per scopi di dimostrazione e apprendimento
- Fondamento: effettuare investimenti fondamentali per scalare l'adozione del cloud (ad esempio, creazione di una landing zone, definizione di una CCo E, definizione di un modello operativo)
- Migrazione: migrazione di singole applicazioni
- Reinvenzione: ottimizzazione di prodotti e servizi e innovazione nel cloud

Queste fasi sono state definite da Stephen Orban nel post sul blog The Journey Toward Cloud-First & the Stages of Adoption on the Enterprise Strategy. Cloud AWS Per informazioni su come si relazionano alla strategia di AWS migrazione, consulta la guida alla preparazione alla migrazione.

CMDB

Vedi database di gestione della configurazione.

repository di codice

Una posizione in cui il codice di origine e altri asset, come documentazione, esempi e script, vengono archiviati e aggiornati attraverso processi di controllo delle versioni. Gli archivi cloud più comuni includono GitHub oBitbucket Cloud. Ogni versione del codice è denominata ramo. In una struttura a microservizi, ogni repository è dedicato a una singola funzionalità. Una singola pipeline CI/CD può utilizzare più repository.

cache fredda

Una cache del buffer vuota, non ben popolata o contenente dati obsoleti o irrilevanti. Ciò influisce sulle prestazioni perché l'istanza di database deve leggere dalla memoria o dal disco principale, il che richiede più tempo rispetto alla lettura dalla cache del buffer.

dati freddi

Dati a cui si accede raramente e che in genere sono storici. Quando si eseguono interrogazioni di questo tipo di dati, le interrogazioni lente sono in genere accettabili. Lo spostamento di questi dati su livelli o classi di storage meno costosi e con prestazioni inferiori può ridurre i costi.

visione artificiale (CV)

Un campo dell'intelligenza artificiale che utilizza l'apprendimento automatico per analizzare ed estrarre informazioni da formati visivi come immagini e video digitali. Ad esempio, Amazon SageMaker AI fornisce algoritmi di elaborazione delle immagini per CV.

deriva della configurazione

Per un carico di lavoro, una modifica della configurazione rispetto allo stato previsto. Potrebbe causare la non conformità del carico di lavoro e in genere è graduale e involontaria.

database di gestione della configurazione (CMDB)

Un repository che archivia e gestisce le informazioni su un database e il relativo ambiente IT, inclusi i componenti hardware e software e le relative configurazioni. In genere si utilizzano i dati di un CMDB nella fase di individuazione e analisi del portafoglio della migrazione.

Pacchetto di conformità

Una raccolta di AWS Config regole e azioni correttive che puoi assemblare per personalizzare i controlli di conformità e sicurezza. È possibile distribuire un pacchetto di conformità come singola entità in una regione Account AWS and o all'interno di un'organizzazione utilizzando un modello YAML. Per ulteriori informazioni, consulta i Conformance Pack nella documentazione. AWS Config

integrazione e distribuzione continua (continuous integration and continuous delivery, CI/CD)

Il processo di automazione delle fasi di origine, compilazione, test, gestione temporanea e produzione del processo di rilascio del software. CI/CD viene comunemente descritto come una pipeline. CI/CD può aiutarvi ad automatizzare i processi, migliorare la produttività, migliorare la qualità del codice e velocizzare le consegne. Per ulteriori informazioni, consulta Vantaggi

della distribuzione continua. CD può anche significare continuous deployment (implementazione continua). Per ulteriori informazioni, consulta Distribuzione continua e implementazione continua a confronto.

CV

Vedi visione artificiale.

D

dati a riposo

Dati stazionari nella rete, ad esempio i dati archiviati.

classificazione dei dati

Un processo per identificare e classificare i dati nella rete in base alla loro criticità e sensibilità. È un componente fondamentale di qualsiasi strategia di gestione dei rischi di sicurezza informatica perché consente di determinare i controlli di protezione e conservazione appropriati per i dati. La classificazione dei dati è un componente del pilastro della sicurezza nel AWS Well-Architected Framework. Per ulteriori informazioni, consulta Classificazione dei dati.

deriva dei dati

Una variazione significativa tra i dati di produzione e i dati utilizzati per addestrare un modello di machine learning o una modifica significativa dei dati di input nel tempo. La deriva dei dati può ridurre la qualità, l'accuratezza e l'equità complessive nelle previsioni dei modelli ML.

dati in transito

Dati che si spostano attivamente attraverso la rete, ad esempio tra le risorse di rete.

rete di dati

Un framework architettonico che fornisce la proprietà distribuita e decentralizzata dei dati con gestione e governance centralizzate.

riduzione al minimo dei dati

Il principio della raccolta e del trattamento dei soli dati strettamente necessari. Praticare la riduzione al minimo dei dati in the Cloud AWS può ridurre i rischi per la privacy, i costi e l'impronta di carbonio delle analisi.

perimetro dei dati

Una serie di barriere preventive nell' AWS ambiente che aiutano a garantire che solo le identità attendibili accedano alle risorse attendibili delle reti previste. Per ulteriori informazioni, consulta Building a data perimeter on. AWS

pre-elaborazione dei dati

Trasformare i dati grezzi in un formato che possa essere facilmente analizzato dal modello di ML. La pre-elaborazione dei dati può comportare la rimozione di determinate colonne o righe e l'eliminazione di valori mancanti, incoerenti o duplicati.

provenienza dei dati

Il processo di tracciamento dell'origine e della cronologia dei dati durante il loro ciclo di vita, ad esempio il modo in cui i dati sono stati generati, trasmessi e archiviati.

soggetto dei dati

Un individuo i cui dati vengono raccolti ed elaborati.

data warehouse

Un sistema di gestione dei dati che supporta la business intelligence, come l'analisi. I data warehouse contengono in genere grandi quantità di dati storici e vengono generalmente utilizzati per interrogazioni e analisi.

linguaggio di definizione del database (DDL)

Istruzioni o comandi per creare o modificare la struttura di tabelle e oggetti in un database.

linguaggio di manipolazione del database (DML)

Istruzioni o comandi per modificare (inserire, aggiornare ed eliminare) informazioni in un database.

DDL

Vedi linguaggio di definizione del database.

deep ensemble

Combinare più modelli di deep learning per la previsione. È possibile utilizzare i deep ensemble per ottenere una previsione più accurata o per stimare l'incertezza nelle previsioni.

deep learning

Un sottocampo del ML che utilizza più livelli di reti neurali artificiali per identificare la mappatura tra i dati di input e le variabili target di interesse.

defense-in-depth

Un approccio alla sicurezza delle informazioni in cui una serie di meccanismi e controlli di sicurezza sono accuratamente stratificati su una rete di computer per proteggere la riservatezza, l'integrità e la disponibilità della rete e dei dati al suo interno. Quando si adotta questa strategia AWS, si aggiungono più controlli a diversi livelli della AWS Organizations struttura per proteggere le risorse. Ad esempio, un defense-in-depth approccio potrebbe combinare l'autenticazione a più fattori, la segmentazione della rete e la crittografia.

amministratore delegato

In AWS Organizations, un servizio compatibile può registrare un account AWS membro per amministrare gli account dell'organizzazione e gestire le autorizzazioni per quel servizio. Questo account è denominato amministratore delegato per quel servizio specifico. Per ulteriori informazioni e un elenco di servizi compatibili, consulta Servizi che funzionano con AWS Organizations nella documentazione di AWS Organizations .

implementazione

Il processo di creazione di un'applicazione, di nuove funzionalità o di correzioni di codice disponibili nell'ambiente di destinazione. L'implementazione prevede l'applicazione di modifiche in una base di codice, seguita dalla creazione e dall'esecuzione di tale base di codice negli ambienti applicativi.

Ambiente di sviluppo

Vedi ambiente.

controllo di rilevamento

Un controllo di sicurezza progettato per rilevare, registrare e avvisare dopo che si è verificato un evento. Questi controlli rappresentano una seconda linea di difesa e avvisano l'utente in caso di eventi di sicurezza che aggirano i controlli preventivi in vigore. Per ulteriori informazioni, consulta Controlli di rilevamento in Implementazione dei controlli di sicurezza in AWS.

mappatura del flusso di valore dello sviluppo (DVSM)

Un processo utilizzato per identificare e dare priorità ai vincoli che influiscono negativamente sulla velocità e sulla qualità nel ciclo di vita dello sviluppo del software. DVSM estende il processo di

mappatura del flusso di valore originariamente progettato per pratiche di produzione snella. Si concentra sulle fasi e sui team necessari per creare e trasferire valore attraverso il processo di sviluppo del software.

gemello digitale

Una rappresentazione virtuale di un sistema reale, ad esempio un edificio, una fabbrica, un'attrezzatura industriale o una linea di produzione. I gemelli digitali supportano la manutenzione predittiva, il monitoraggio remoto e l'ottimizzazione della produzione.

tabella delle dimensioni

In uno schema a stella, una tabella più piccola che contiene gli attributi dei dati quantitativi in una tabella dei fatti. Gli attributi della tabella delle dimensioni sono in genere campi di testo o numeri discreti che si comportano come testo. Questi attributi vengono comunemente utilizzati per il vincolo delle query, il filtraggio e l'etichettatura dei set di risultati.

disastro

Un evento che impedisce a un carico di lavoro o a un sistema di raggiungere gli obiettivi aziendali nella sua sede principale di implementazione. Questi eventi possono essere disastri naturali, guasti tecnici o il risultato di azioni umane, come errori di configurazione involontari o attacchi di malware.

disaster recovery (DR)

La strategia e il processo utilizzati per ridurre al minimo i tempi di inattività e la perdita di dati causati da un disastro. Per ulteriori informazioni, consulta Disaster Recovery of Workloads su AWS: Recovery in the Cloud in the AWS Well-Architected Framework.

DML

Vedi linguaggio di manipolazione del database.

progettazione basata sul dominio

Un approccio allo sviluppo di un sistema software complesso collegandone i componenti a domini in evoluzione, o obiettivi aziendali principali, perseguiti da ciascun componente. Questo concetto è stato introdotto da Eric Evans nel suo libro,Domain-Driven Design: Tackling Complexity in the Heart of Software (Boston: Addison-Wesley Professional, 2003). Per informazioni su come utilizzare la progettazione basata sul dominio con il modello del fico strangolatore (Strangler Fig), consulta la sezione Modernizzazione incrementale dei servizi Web Microsoft ASP.NET (ASMX) legacy utilizzando container e il Gateway Amazon API.

DOTT.

Vedi disaster recovery.

rilevamento della deriva

Tracciamento delle deviazioni da una configurazione di base. Ad esempio, è possibile AWS CloudFormation utilizzarlo per rilevare deviazioni nelle risorse di sistema oppure AWS Control Tower per rilevare cambiamenti nella landing zone che potrebbero influire sulla conformità ai requisiti di governance.

DVSM

Vedi la mappatura del flusso di valore dello sviluppo.

E

EDA

Vedi analisi esplorativa dei dati.

MODIFICA

Vedi scambio elettronico di dati.

edge computing

La tecnologia che aumenta la potenza di calcolo per i dispositivi intelligenti all'edge di una rete IoT. Rispetto al cloud computing, l'edge computing può ridurre la latenza di comunicazione e migliorare i tempi di risposta.

scambio elettronico di dati (EDI)

Lo scambio automatizzato di documenti aziendali tra organizzazioni. Per ulteriori informazioni, vedere Cos'è lo scambio elettronico di dati.

crittografia

Un processo di elaborazione che trasforma i dati in chiaro, leggibili dall'uomo, in testo cifrato.

chiave crittografica

Una stringa crittografica di bit randomizzati generata da un algoritmo di crittografia. Le chiavi possono variare di lunghezza e ogni chiave è progettata per essere imprevedibile e univoca.

endianità

L'ordine in cui i byte vengono archiviati nella memoria del computer. I sistemi big-endian memorizzano per primo il byte più importante. I sistemi little-endian memorizzano per primo il byte meno importante.

endpoint

Vedi service endpoint.

servizio endpoint

Un servizio che puoi ospitare in un cloud privato virtuale (VPC) da condividere con altri utenti. Puoi creare un servizio endpoint con AWS PrivateLink e concedere autorizzazioni ad altri Account AWS o a AWS Identity and Access Management (IAM) principali. Questi account o principali possono connettersi al servizio endpoint in privato creando endpoint VPC di interfaccia. Per ulteriori informazioni, consulta Creazione di un servizio endpoint nella documentazione di Amazon Virtual Private Cloud (Amazon VPC).

pianificazione delle risorse aziendali (ERP)

Un sistema che automatizza e gestisce i processi aziendali chiave (come contabilità, MES e gestione dei progetti) per un'azienda.

crittografia envelope

Il processo di crittografia di una chiave di crittografia con un'altra chiave di crittografia. Per ulteriori informazioni, vedete Envelope encryption nella documentazione AWS Key Management Service (AWS KMS).

ambiente

Un'istanza di un'applicazione in esecuzione. Di seguito sono riportati i tipi di ambiente più comuni nel cloud computing:

- ambiente di sviluppo: un'istanza di un'applicazione in esecuzione disponibile solo per il team principale responsabile della manutenzione dell'applicazione. Gli ambienti di sviluppo vengono utilizzati per testare le modifiche prima di promuoverle negli ambienti superiori. Questo tipo di ambiente viene talvolta definito ambiente di test.
- ambienti inferiori: tutti gli ambienti di sviluppo di un'applicazione, ad esempio quelli utilizzati per le build e i test iniziali.

- ambiente di produzione: un'istanza di un'applicazione in esecuzione a cui gli utenti finali possono accedere. In una CI/CD pipeline, l'ambiente di produzione è l'ultimo ambiente di distribuzione.
- ambienti superiori: tutti gli ambienti a cui possono accedere utenti diversi dal team di sviluppo principale. Si può trattare di un ambiente di produzione, ambienti di preproduzione e ambienti per i test di accettazione da parte degli utenti.

epica

Nelle metodologie agili, categorie funzionali che aiutano a organizzare e dare priorità al lavoro. Le epiche forniscono una descrizione di alto livello dei requisiti e delle attività di implementazione. Ad esempio, le epopee della sicurezza AWS CAF includono la gestione delle identità e degli accessi, i controlli investigativi, la sicurezza dell'infrastruttura, la protezione dei dati e la risposta agli incidenti. Per ulteriori informazioni sulle epiche, consulta la strategia di migrazione AWS , consulta la guida all'implementazione del programma.

ERP

Vedi pianificazione delle risorse aziendali.

analisi esplorativa dei dati (EDA)

Il processo di analisi di un set di dati per comprenderne le caratteristiche principali. Si raccolgono o si aggregano dati e quindi si eseguono indagini iniziali per trovare modelli, rilevare anomalie e verificare ipotesi. L'EDA viene eseguita calcolando statistiche di riepilogo e creando visualizzazioni di dati.

F

tabella dei fatti

Il tavolo centrale in uno schema a stella. Memorizza dati quantitativi sulle operazioni aziendali. In genere, una tabella dei fatti contiene due tipi di colonne: quelle che contengono misure e quelle che contengono una chiave esterna per una tabella di dimensioni.

fallire velocemente

Una filosofia che utilizza test frequenti e incrementali per ridurre il ciclo di vita dello sviluppo. È una parte fondamentale di un approccio agile.

limite di isolamento dei guasti

Nel Cloud AWS, un limite come una zona di disponibilità Regione AWS, un piano di controllo o un piano dati che limita l'effetto di un errore e aiuta a migliorare la resilienza dei carichi di lavoro. Per ulteriori informazioni, consulta AWS Fault Isolation Boundaries.

ramo di funzionalità

Vedi filiale.

caratteristiche

I dati di input che usi per fare una previsione. Ad esempio, in un contesto di produzione, le caratteristiche potrebbero essere immagini acquisite periodicamente dalla linea di produzione.

importanza delle caratteristiche

Quanto è importante una caratteristica per le previsioni di un modello. Di solito viene espresso come punteggio numerico che può essere calcolato con varie tecniche, come Shapley Additive Explanations (SHAP) e gradienti integrati. Per ulteriori informazioni, consulta Interpretabilità del modello di machine learning con AWS.

trasformazione delle funzionalità

Per ottimizzare i dati per il processo di machine learning, incluso l'arricchimento dei dati con fonti aggiuntive, il dimensionamento dei valori o l'estrazione di più set di informazioni da un singolo campo di dati. Ciò consente al modello di ML di trarre vantaggio dai dati. Ad esempio, se suddividi la data "2021-05-27 00:15:37" in "2021", "maggio", "giovedì" e "15", puoi aiutare l'algoritmo di apprendimento ad apprendere modelli sfumati associati a diversi componenti dei dati.

prompt con pochi scatti

Fornire a un LLM un numero limitato di esempi che dimostrino l'attività e il risultato desiderato prima di chiedergli di eseguire un'attività simile. Questa tecnica è un'applicazione dell'apprendimento contestuale, in cui i modelli imparano da esempi (immagini) incorporati nei prompt. I prompt con pochi passaggi possono essere efficaci per attività che richiedono una formattazione, un ragionamento o una conoscenza del dominio specifici. Vedi anche zero-shot prompting.

FGAC

Vedi il controllo granulare degli accessi.

controllo granulare degli accessi (FGAC)

L'uso di più condizioni per consentire o rifiutare una richiesta di accesso.

migrazione flash-cut

Un metodo di migrazione del database che utilizza la replica continua dei dati tramite l'acquisizione dei dati delle modifiche per migrare i dati nel più breve tempo possibile, anziché utilizzare un approccio graduale. L'obiettivo è ridurre al minimo i tempi di inattività.

FM

Vedi modello di base.

modello di fondazione (FM)

Una grande rete neurale di deep learning che si è addestrata su enormi set di dati generalizzati e non etichettati. FMs sono in grado di svolgere un'ampia varietà di attività generali, come comprendere il linguaggio, generare testo e immagini e conversare in linguaggio naturale. Per ulteriori informazioni, consulta Cosa sono i modelli Foundation.

G

IA generativa

Un sottoinsieme di modelli di intelligenza artificiale che sono stati addestrati su grandi quantità di dati e che possono utilizzare un semplice messaggio di testo per creare nuovi contenuti e artefatti, come immagini, video, testo e audio. Per ulteriori informazioni, consulta Cos'è l'IA generativa.

blocco geografico

Vedi restrizioni geografiche.

limitazioni geografiche (blocco geografico)

In Amazon CloudFront, un'opzione per impedire agli utenti di determinati paesi di accedere alle distribuzioni di contenuti. Puoi utilizzare un elenco consentito o un elenco di blocco per specificare i paesi approvati e vietati. Per ulteriori informazioni, consulta Limitare la distribuzione geografica dei contenuti nella CloudFront documentazione.

Flusso di lavoro di GitFlow

Un approccio in cui gli ambienti inferiori e superiori utilizzano rami diversi in un repository di codice di origine. Il flusso di lavoro Gitflow è considerato obsoleto e il flusso di lavoro basato su trunk è l'approccio moderno e preferito.

immagine dorata

Un'istantanea di un sistema o di un software utilizzata come modello per distribuire nuove istanze di quel sistema o software. Ad esempio, nella produzione, un'immagine dorata può essere utilizzata per fornire software su più dispositivi e contribuire a migliorare la velocità, la scalabilità e la produttività nelle operazioni di produzione dei dispositivi.

strategia greenfield

L'assenza di infrastrutture esistenti in un nuovo ambiente. Quando si adotta una strategia greenfield per un'architettura di sistema, è possibile selezionare tutte le nuove tecnologie senza il vincolo della compatibilità con l'infrastruttura esistente, nota anche come brownfield. Per l'espansione dell'infrastruttura esistente, è possibile combinare strategie brownfield e greenfield.

guardrail

Una regola di alto livello che aiuta a governare le risorse, le politiche e la conformità tra le unità organizzative (). OUs I guardrail preventivi applicano le policy per garantire l'allineamento agli standard di conformità. Vengono implementati utilizzando le policy di controllo dei servizi e i limiti delle autorizzazioni IAM. I guardrail di rilevamento rilevano le violazioni delle policy e i problemi di conformità e generano avvisi per porvi rimedio. Sono implementati utilizzando Amazon AWS Config AWS Security Hub CSPM GuardDuty AWS Trusted Advisor, Amazon Inspector e controlli personalizzati AWS Lambda .

H

AH

Vedi disponibilità elevata.

migrazione di database eterogenea

Migrazione del database di origine in un database di destinazione che utilizza un motore di database diverso (ad esempio, da Oracle ad Amazon Aurora). La migrazione eterogenea fa in

genere parte di uno sforzo di riprogettazione e la conversione dello schema può essere un'attività complessa. AWS offre AWS SCT che aiuta con le conversioni dello schema.

alta disponibilità (HA)

La capacità di un carico di lavoro di funzionare in modo continuo, senza intervento, in caso di sfide o disastri. I sistemi HA sono progettati per il failover automatico, fornire costantemente prestazioni di alta qualità e gestire carichi e guasti diversi con un impatto minimo sulle prestazioni.

modernizzazione storica

Un approccio utilizzato per modernizzare e aggiornare i sistemi di tecnologia operativa (OT) per soddisfare meglio le esigenze dell'industria manifatturiera. Uno storico è un tipo di database utilizzato per raccogliere e archiviare dati da varie fonti in una fabbrica.

dati di blocco

Una parte di dati storici etichettati che viene trattenuta da un set di dati utilizzata per addestrare un modello di apprendimento automatico. È possibile utilizzare i dati di holdout per valutare le prestazioni del modello confrontando le previsioni del modello con i dati di holdout.

migrazione di database omogenea

Migrazione del database di origine in un database di destinazione che condivide lo stesso motore di database (ad esempio, da Microsoft SQL Server ad Amazon RDS per SQL Server). La migrazione omogenea fa in genere parte di un'operazione di rehosting o ridefinizione della piattaforma. Per migrare lo schema è possibile utilizzare le utilità native del database.

dati caldi

Dati a cui si accede frequentemente, come dati in tempo reale o dati di traduzione recenti. Questi dati richiedono in genere un livello o una classe di storage ad alte prestazioni per fornire risposte rapide alle query.

hotfix

Una soluzione urgente per un problema critico in un ambiente di produzione. A causa della sua urgenza, un hotfix viene in genere creato al di fuori del tipico DevOps flusso di lavoro di rilascio.

periodo di hypercare

Subito dopo la conversione, il periodo di tempo in cui un team di migrazione gestisce e monitora le applicazioni migrate nel cloud per risolvere eventuali problemi. In genere, questo periodo dura

da 1 a 4 giorni. Al termine del periodo di hypercare, il team addetto alla migrazione in genere trasferisce la responsabilità delle applicazioni al team addetto alle operazioni cloud.

I

IaC

Vedi l'infrastruttura come codice.

Policy basata su identità

Una policy associata a uno o più principi IAM che definisce le relative autorizzazioni all'interno dell' Cloud AWS ambiente.

applicazione inattiva

Un'applicazione che prevede un uso di CPU e memoria medio compreso tra il 5% e il 20% in un periodo di 90 giorni. In un progetto di migrazione, è normale ritirare queste applicazioni o mantenerle on-premise.

IIoT

Vedi Industrial Internet of Things.

infrastruttura immutabile

Un modello che implementa una nuova infrastruttura per i carichi di lavoro di produzione anziché aggiornare, applicare patch o modificare l'infrastruttura esistente. Le infrastrutture immutabili sono intrinsecamente più coerenti, affidabili e prevedibili delle infrastrutture mutabili. Per ulteriori informazioni, consulta la best practice Deploy using immutable infrastructure in Well-Architected AWS Framework.

VPC in ingresso (ingress)

In un'architettura AWS multi-account, un VPC che accetta, ispeziona e indirizza le connessioni di rete dall'esterno di un'applicazione. La AWS Security Reference Architecture consiglia di configurare l'account di rete con funzionalità in entrata, in uscita e di ispezione VPCs per proteggere l'interfaccia bidirezionale tra l'applicazione e Internet in generale.

migrazione incrementale

Una strategia di conversione in cui si esegue la migrazione dell'applicazione in piccole parti anziché eseguire una conversione singola e completa. Ad esempio, inizialmente potresti spostare

solo alcuni microservizi o utenti nel nuovo sistema. Dopo aver verificato che tutto funzioni correttamente, puoi spostare in modo incrementale microservizi o utenti aggiuntivi fino alla disattivazione del sistema legacy. Questa strategia riduce i rischi associati alle migrazioni di grandi dimensioni.

Industria 4.0

Un termine introdotto da Klaus Schwab nel 2016 per riferirsi alla modernizzazione dei processi di produzione attraverso progressi in termini di connettività, dati in tempo reale, automazione, analisi e AI/ML.

infrastruttura

Tutte le risorse e gli asset contenuti nell'ambiente di un'applicazione.

infrastruttura come codice (IaC)

Il processo di provisioning e gestione dell'infrastruttura di un'applicazione tramite un insieme di file di configurazione. Il processo IaC è progettato per aiutarti a centralizzare la gestione dell'infrastruttura, a standardizzare le risorse e a dimensionare rapidamente, in modo che i nuovi ambienti siano ripetibili, affidabili e coerenti.

IIoInternet delle cose industriale (T)

L'uso di sensori e dispositivi connessi a Internet nei settori industriali, come quello manifatturiero, energetico, automobilistico, sanitario, delle scienze della vita e dell'agricoltura. Per ulteriori informazioni, vedere Creazione di una strategia di trasformazione digitale per l'Internet of Things (IIoT) industriale.

VPC di ispezione

In un'architettura AWS multi-account, un VPC centralizzato che gestisce le ispezioni del traffico di rete tra VPCs (nello stesso o in modo diverso Regioni AWS), Internet e le reti locali. La AWS Security Reference Architecture consiglia di configurare l'account di rete con informazioni in entrata, in uscita e di ispezione VPCs per proteggere l'interfaccia bidirezionale tra l'applicazione e Internet in generale.

Internet of Things (IoT)

La rete di oggetti fisici connessi con sensori o processori incorporati che comunicano con altri dispositivi e sistemi tramite Internet o una rete di comunicazione locale. Per ulteriori informazioni, consulta Cos'è l'IoT?

interpretabilità

Una caratteristica di un modello di machine learning che descrive il grado in cui un essere umano è in grado di comprendere in che modo le previsioni del modello dipendono dai suoi input. Per ulteriori informazioni, vedere Interpretabilità del modello di machine learning con. AWS

IoT

Vedi Internet of Things.

libreria di informazioni IT (ITIL)

Una serie di best practice per offrire servizi IT e allinearli ai requisiti aziendali. ITIL fornisce le basi per ITSM.

gestione dei servizi IT (ITSM)

Attività associate alla progettazione, implementazione, gestione e supporto dei servizi IT per un'organizzazione. Per informazioni sull'integrazione delle operazioni cloud con gli strumenti ITSM, consulta la guida all'integrazione delle operazioni.

ITIL

Vedi la libreria di informazioni IT.

ITSM

Vedi Gestione dei servizi IT.

L

controllo degli accessi basato su etichette (LBAC)

Un'implementazione del controllo di accesso obbligatorio (MAC) in cui agli utenti e ai dati stessi viene assegnato esplicitamente un valore di etichetta di sicurezza. L'intersezione tra l'etichetta di sicurezza utente e l'etichetta di sicurezza dei dati determina quali righe e colonne possono essere visualizzate dall'utente.

zona di destinazione

Una landing zone è un AWS ambiente multi-account ben progettato, scalabile e sicuro. Questo è un punto di partenza dal quale le organizzazioni possono avviare e distribuire rapidamente carichi di lavoro e applicazioni con fiducia nel loro ambiente di sicurezza e infrastruttura. Per ulteriori

informazioni sulle zone di destinazione, consulta la sezione Configurazione di un ambiente AWS multi-account sicuro e scalabile.

modello linguistico di grandi dimensioni (LLM)

Un modello di intelligenza artificiale di deep learning preaddestrato su una grande quantità di dati. Un LLM può svolgere più attività, come rispondere a domande, riepilogare documenti, tradurre testo in altre lingue e completare frasi. Per ulteriori informazioni, consulta Cosa sono. LLMs

migrazione su larga scala

Una migrazione di 300 o più server.

BIANCO

Vedi controllo degli accessi basato su etichette.

Privilegio minimo

La best practice di sicurezza per la concessione delle autorizzazioni minime richieste per eseguire un'attività. Per ulteriori informazioni, consulta Applicazione delle autorizzazioni del privilegio minimo nella documentazione di IAM.

eseguire il rehosting (lift and shift)

Vedi 7 R.

sistema little-endian

Un sistema che memorizza per primo il byte meno importante. Vedi anche endianità.

LLM

Vedi modello linguistico di grandi dimensioni.

ambienti inferiori

Vedi ambiente.

M

machine learning (ML)

Un tipo di intelligenza artificiale che utilizza algoritmi e tecniche per il riconoscimento e l'apprendimento di schemi. Il machine learning analizza e apprende dai dati registrati, come i dati

dell'Internet delle cose (IoT), per generare un modello statistico basato su modelli. Per ulteriori informazioni, consulta la sezione Machine learning.

ramo principale

Vedi filiale.

malware

Software progettato per compromettere la sicurezza o la privacy del computer. Il malware potrebbe interrompere i sistemi informatici, divulgare informazioni sensibili o ottenere accessi non autorizzati. Esempi di malware includono virus, worm, ransomware, trojan horse, spyware e keylogger.

servizi gestiti

Servizi AWS per cui AWS gestisce il livello di infrastruttura, il sistema operativo e le piattaforme e si accede agli endpoint per archiviare e recuperare i dati. Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) e Amazon DynamoDB sono esempi di servizi gestiti. Questi sono noti anche come servizi astratti.

sistema di esecuzione della produzione (MES)

Un sistema software per tracciare, monitorare, documentare e controllare i processi di produzione che convertono le materie prime in prodotti finiti in officina.

MAP

Vedi Migration Acceleration Program.

meccanismo

Un processo completo in cui si crea uno strumento, si promuove l'adozione dello strumento e quindi si esaminano i risultati per apportare le modifiche. Un meccanismo è un ciclo che si rafforza e si migliora man mano che funziona. Per ulteriori informazioni, consulta Creazione di meccanismi nel AWS Well-Architected Framework.

account membro

Tutti gli account Account AWS diversi dall'account di gestione che fanno parte di un'organizzazione in. AWS Organizations Un account può essere membro di una sola organizzazione alla volta.

MEH

Vedi sistema di esecuzione della produzione.

Message Queuing Telemetry Transport (MQTT)

Un protocollo di comunicazione machine-to-machine (M2M) leggero, basato sul modello di pubblicazione/sottoscrizione, per dispositivi IoT con risorse limitate.

microservizio

Un servizio piccolo e indipendente che comunica tramite canali ben definiti ed è in genere di proprietà di piccoli team autonomi. APIs Ad esempio, un sistema assicurativo potrebbe includere microservizi che si riferiscono a funzionalità aziendali, come vendite o marketing, o sottodomini, come acquisti, reclami o analisi. I vantaggi dei microservizi includono agilità, dimensionamento flessibile, facilità di implementazione, codice riutilizzabile e resilienza. Per ulteriori informazioni, consulta Integrazione dei microservizi utilizzando servizi serverless. AWS

architettura di microservizi

Un approccio alla creazione di un'applicazione con componenti indipendenti che eseguono ogni processo applicativo come microservizio. Questi microservizi comunicano attraverso un'interfaccia ben definita utilizzando sistemi leggeri. APIs Ogni microservizio in questa architettura può essere aggiornato, distribuito e dimensionato per soddisfare la richiesta di funzioni specifiche di un'applicazione. Per ulteriori informazioni, vedere Implementazione dei microservizi su. AWS

Programma di accelerazione della migrazione (MAP)

Un AWS programma che fornisce consulenza, supporto, formazione e servizi per aiutare le organizzazioni a costruire una solida base operativa per il passaggio al cloud e per contribuire a compensare il costo iniziale delle migrazioni. MAP include una metodologia di migrazione per eseguire le migrazioni precedenti in modo metodico e un set di strumenti per automatizzare e accelerare gli scenari di migrazione comuni.

migrazione su larga scala

Il processo di trasferimento della maggior parte del portfolio di applicazioni sul cloud avviene a ondate, con più applicazioni trasferite a una velocità maggiore in ogni ondata. Questa fase utilizza le migliori pratiche e le lezioni apprese nelle fasi precedenti per implementare una fabbrica di migrazione di team, strumenti e processi per semplificare la migrazione dei carichi di lavoro attraverso l'automazione e la distribuzione agile. Questa è la terza fase della strategia di migrazione AWS.

fabbrica di migrazione

Team interfunzionali che semplificano la migrazione dei carichi di lavoro attraverso approcci automatizzati e agili. I team di Migration Factory in genere includono addetti alle operazioni,

analisti e proprietari aziendali, ingegneri addetti alla migrazione, sviluppatori e DevOps professionisti che lavorano nell'ambito degli sprint. Tra il 20% e il 50% di un portfolio di applicazioni aziendali è costituito da schemi ripetuti che possono essere ottimizzati con un approccio di fabbrica. Per ulteriori informazioni, consulta la discussione sulle fabbriche di migrazione e la Guida alla fabbrica di migrazione al cloud in questo set di contenuti.

metadati di migrazione

Le informazioni sull'applicazione e sul server necessarie per completare la migrazione. Ogni modello di migrazione richiede un set diverso di metadati di migrazione. Esempi di metadati di migrazione includono la sottorete, il gruppo di sicurezza e l'account di destinazione. AWS

modello di migrazione

Un'attività di migrazione ripetibile che descrive in dettaglio la strategia di migrazione, la destinazione della migrazione e l'applicazione o il servizio di migrazione utilizzati. Esempio: riorganizza la migrazione su Amazon EC2 AWS con Application Migration Service.

Valutazione del portfolio di migrazione (MPA)

Uno strumento online che fornisce informazioni per la convalida del business case per la migrazione a. Cloud AWS MPA offre una valutazione dettagliata del portfolio (dimensionamento corretto dei server, prezzi, confronto del TCO, analisi dei costi di migrazione) e pianificazione della migrazione (analisi e raccolta dei dati delle applicazioni, raggruppamento delle applicazioni, prioritizzazione delle migrazioni e pianificazione delle ondate). Lo strumento MPA (richiede l'accesso) è disponibile gratuitamente per tutti i AWS consulenti e i consulenti dei partner APN.

valutazione della preparazione alla migrazione (MRA)

Il processo di acquisizione di informazioni sullo stato di preparazione al cloud di un'organizzazione, l'identificazione dei punti di forza e di debolezza e la creazione di un piano d'azione per colmare le lacune identificate, utilizzando il CAF. AWS Per ulteriori informazioni, consulta la guida di preparazione alla migrazione. MRA è la prima fase della strategia di migrazione AWS.

strategia di migrazione

L'approccio utilizzato per migrare un carico di lavoro verso. Cloud AWS Per ulteriori informazioni, consulta la voce 7 R in questo glossario e consulta Mobilita la tua organizzazione per accelerare le migrazioni su larga scala.

ML

Vedi machine learning.

modernizzazione

Trasformazione di un'applicazione obsoleta (legacy o monolitica) e della relativa infrastruttura in un sistema agile, elastico e altamente disponibile nel cloud per ridurre i costi, aumentare l'efficienza e sfruttare le innovazioni. Per ulteriori informazioni, vedere Strategia per la modernizzazione delle applicazioni in. Cloud AWS

valutazione della preparazione alla modernizzazione

Una valutazione che aiuta a determinare la preparazione alla modernizzazione delle applicazioni di un'organizzazione, identifica vantaggi, rischi e dipendenze e determina in che misura l'organizzazione può supportare lo stato futuro di tali applicazioni. Il risultato della valutazione è uno schema dell'architettura di destinazione, una tabella di marcia che descrive in dettaglio le fasi di sviluppo e le tappe fondamentali del processo di modernizzazione e un piano d'azione per colmare le lacune identificate. Per ulteriori informazioni, vedere Valutazione della preparazione alla modernizzazione per le applicazioni in. Cloud AWS

applicazioni monolitiche (monoliti)

Applicazioni eseguite come un unico servizio con processi strettamente collegati. Le applicazioni monolitiche presentano diversi inconvenienti. Se una funzionalità dell'applicazione registra un picco di domanda, l'intera architettura deve essere dimensionata. L'aggiunta o il miglioramento delle funzionalità di un'applicazione monolitica diventa inoltre più complessa man mano che la base di codice cresce. Per risolvere questi problemi, puoi utilizzare un'architettura di microservizi. Per ulteriori informazioni, consulta la sezione Scomposizione dei monoliti in microservizi.

MAPPA

Vedi Migration Portfolio Assessment.

MQTT

Vedi Message Queuing Telemetry Transport.

classificazione multiclasse

Un processo che aiuta a generare previsioni per più classi (prevedendo uno o più di due risultati). Ad esempio, un modello di machine learning potrebbe chiedere "Questo prodotto è un libro, un'auto o un telefono?" oppure "Quale categoria di prodotti è più interessante per questo cliente?"

infrastruttura mutabile

Un modello che aggiorna e modifica l'infrastruttura esistente per i carichi di lavoro di produzione. Per migliorare la coerenza, l'affidabilità e la prevedibilità, il AWS Well-Architected Framework consiglia l'uso di un'infrastruttura immutabile come best practice.

O

OAC

Vedi Origin Access Control.

QUERCIA

Vedi Origin Access Identity.

OCM

Vedi gestione delle modifiche organizzative.

migrazione offline

Un metodo di migrazione in cui il carico di lavoro di origine viene eliminato durante il processo di migrazione. Questo metodo prevede tempi di inattività prolungati e viene in genere utilizzato per carichi di lavoro piccoli e non critici.

OI

Vedi l'integrazione delle operazioni.

OLA

Vedi accordo a livello operativo.

migrazione online

Un metodo di migrazione in cui il carico di lavoro di origine viene copiato sul sistema di destinazione senza essere messo offline. Le applicazioni connesse al carico di lavoro possono continuare a funzionare durante la migrazione. Questo metodo comporta tempi di inattività pari a zero o comunque minimi e viene in genere utilizzato per carichi di lavoro di produzione critici.

OPC-UA

Vedi Open Process Communications - Unified Architecture.

Comunicazioni a processo aperto - Architettura unificata (OPC-UA)

Un protocollo di comunicazione machine-to-machine (M2M) per l'automazione industriale. OPC-UA fornisce uno standard di interoperabilità con schemi di crittografia, autenticazione e autorizzazione dei dati.

accordo a livello operativo (OLA)

Un accordo che chiarisce quali sono gli impegni reciproci tra i gruppi IT funzionali, a supporto di un accordo sul livello di servizio (SLA).

revisione della prontezza operativa (ORR)

Un elenco di domande e best practice associate che aiutano a comprendere, valutare, prevenire o ridurre la portata degli incidenti e dei possibili guasti. Per ulteriori informazioni, vedere Operational Readiness Reviews (ORR) nel Well-Architected AWS Framework.

tecnologia operativa (OT)

Sistemi hardware e software che interagiscono con l'ambiente fisico per controllare le operazioni, le apparecchiature e le infrastrutture industriali. Nella produzione, l'integrazione di sistemi OT e di tecnologia dell'informazione (IT) è un obiettivo chiave per le trasformazioni dell'Industria 4.0.

integrazione delle operazioni (OI)

Il processo di modernizzazione delle operazioni nel cloud, che prevede la pianificazione, l'automazione e l'integrazione della disponibilità. Per ulteriori informazioni, consulta la guida all'integrazione delle operazioni.

trail organizzativo

Un percorso creato da noi AWS CloudTrail che registra tutti gli eventi di un'organizzazione per tutti Account AWS . AWS Organizations Questo percorso viene creato in ogni Account AWS che fa parte dell'organizzazione e tiene traccia dell'attività in ogni account. Per ulteriori informazioni, consulta Creazione di un percorso per un'organizzazione nella CloudTrail documentazione.

gestione del cambiamento organizzativo (OCM)

Un framework per la gestione di trasformazioni aziendali importanti e che comportano l'interruzione delle attività dal punto di vista delle persone, della cultura e della leadership. OCM aiuta le organizzazioni a prepararsi e passare a nuovi sistemi e strategie accelerando l'adozione del cambiamento, affrontando i problemi di transizione e promuovendo cambiamenti culturali e organizzativi. Nella strategia di AWS migrazione, questo framework si chiama accelerazione delle

persone, a causa della velocità di cambiamento richiesta nei progetti di adozione del cloud. Per ulteriori informazioni, consultare la Guida OCM.

controllo dell'accesso all'origine (OAC)

In CloudFront, un'opzione avanzata per limitare l'accesso per proteggere i contenuti di Amazon Simple Storage Service (Amazon S3). OAC supporta tutti i bucket S3 in generale Regioni AWS, la crittografia lato server con AWS KMS (SSE-KMS) e le richieste dinamiche e dirette al bucket S3. `PUT` `DELETE`

identità di accesso origine (OAI)

Nel CloudFront, un'opzione per limitare l'accesso per proteggere i tuoi contenuti Amazon S3. Quando usi OAI, CloudFront crea un principale con cui Amazon S3 può autenticarsi. I principali autenticati possono accedere ai contenuti in un bucket S3 solo tramite una distribuzione specifica. CloudFront Vedi anche OAC, che fornisce un controllo degli accessi più granulare e avanzato.

ORR

Vedi la revisione della prontezza operativa.

NON

Vedi la tecnologia operativa.

VPC in uscita (egress)

In un'architettura AWS multi-account, un VPC che gestisce le connessioni di rete avviate dall'interno di un'applicazione. La AWS Security Reference Architecture consiglia di configurare l'account di rete con funzionalità in entrata, in uscita e di ispezione VPCs per proteggere l'interfaccia bidirezionale tra l'applicazione e Internet in generale.

P

limite delle autorizzazioni

Una policy di gestione IAM collegata ai principali IAM per impostare le autorizzazioni massime che l'utente o il ruolo possono avere. Per ulteriori informazioni, consulta Limiti delle autorizzazioni nella documentazione di IAM.

informazioni di identificazione personale (PII)

Informazioni che, se visualizzate direttamente o abbinate ad altri dati correlati, possono essere utilizzate per dedurre ragionevolmente l'identità di un individuo. Esempi di informazioni personali includono nomi, indirizzi e informazioni di contatto.

Informazioni che consentono l'identificazione personale degli utenti

Visualizza le informazioni di identificazione personale.

playbook

Una serie di passaggi predefiniti che raccolgono il lavoro associato alle migrazioni, come l'erogazione delle funzioni operative principali nel cloud. Un playbook può assumere la forma di script, runbook automatici o un riepilogo dei processi o dei passaggi necessari per gestire un ambiente modernizzato.

PLC

Vedi controllore logico programmabile.

PLM

Vedi la gestione del ciclo di vita del prodotto.

policy

Un oggetto in grado di definire le autorizzazioni (vedi politica basata sull'identità), specificare le condizioni di accesso (vedi politicabasata sulle risorse) o definire le autorizzazioni massime per tutti gli account di un'organizzazione in (vedi politica di controllo dei servizi). AWS Organizations

persistenza poliglotta

Scelta indipendente della tecnologia di archiviazione di dati di un microservizio in base ai modelli di accesso ai dati e ad altri requisiti. Se i microservizi utilizzano la stessa tecnologia di archiviazione di dati, possono incontrare problemi di implementazione o registrare prestazioni scadenti. I microservizi vengono implementati più facilmente e ottengono prestazioni e scalabilità migliori se utilizzano l'archivio dati più adatto alle loro esigenze.

valutazione del portfolio

Un processo di scoperta, analisi e definizione delle priorità del portfolio di applicazioni per pianificare la migrazione. Per ulteriori informazioni, consulta la pagina Valutazione della preparazione alla migrazione.

predicate

Una condizione di interrogazione che restituisce o, in genere, si trova in una clausola`WHERE`. `true` `false`

predicato pushdown

Una tecnica di ottimizzazione delle query del database che filtra i dati della query prima del trasferimento. Ciò riduce la quantità di dati che devono essere recuperati ed elaborati dal database relazionale e migliora le prestazioni delle query.

controllo preventivo

Un controllo di sicurezza progettato per impedire il verificarsi di un evento. Questi controlli sono la prima linea di difesa per impedire accessi non autorizzati o modifiche indesiderate alla rete. Per ulteriori informazioni, consulta Controlli preventivi in Implementazione dei controlli di sicurezza in AWS.

principale

Un'entità in AWS grado di eseguire azioni e accedere alle risorse. Questa entità è in genere un utente root per un Account AWS ruolo IAM o un utente. Per ulteriori informazioni, consulta Principali in Termini e concetti dei ruoli nella documentazione di IAM.

privacy fin dalla progettazione

Un approccio di ingegneria dei sistemi che tiene conto della privacy durante l'intero processo di sviluppo.

zone ospitate private

Un contenitore che contiene informazioni su come desideri che Amazon Route 53 risponda alle query DNS per un dominio e i relativi sottodomini all'interno di uno o più. VPCs Per ulteriori informazioni, consulta Utilizzo delle zone ospitate private nella documentazione di Route 53.

controllo proattivo

Un controllo di sicurezza progettato per impedire l'implementazione di risorse non conformi. Questi controlli analizzano le risorse prima del loro provisioning. Se la risorsa non è conforme al controllo, non viene fornita. Per ulteriori informazioni, consulta la guida di riferimento sui controlli nella AWS Control Tower documentazione e consulta Controlli proattivi in Implementazione dei controlli di sicurezza su. AWS

gestione del ciclo di vita del prodotto (PLM)

La gestione dei dati e dei processi di un prodotto durante l'intero ciclo di vita, dalla progettazione, sviluppo e lancio, attraverso la crescita e la maturità, fino al declino e alla rimozione.

Ambiente di produzione

Vedi ambiente.

controllore logico programmabile (PLC)

Nella produzione, un computer altamente affidabile e adattabile che monitora le macchine e automatizza i processi di produzione.

concatenamento rapido

Utilizzo dell'output di un prompt LLM come input per il prompt successivo per generare risposte migliori. Questa tecnica viene utilizzata per suddividere un'attività complessa in sottoattività o per perfezionare o espandere iterativamente una risposta preliminare. Aiuta a migliorare l'accuratezza e la pertinenza delle risposte di un modello e consente risultati più granulari e personalizzati.

pseudonimizzazione

Il processo di sostituzione degli identificatori personali in un set di dati con valori segnaposto. La pseudonimizzazione può aiutare a proteggere la privacy personale. I dati pseudonimizzati sono ancora considerati dati personali.

publish/subscribe (pub/sub)

Un modello che consente comunicazioni asincrone tra microservizi per migliorare la scalabilità e la reattività. Ad esempio, in un MES basato su microservizi, un microservizio può pubblicare messaggi di eventi su un canale a cui altri microservizi possono abbonarsi. Il sistema può aggiungere nuovi microservizi senza modificare il servizio di pubblicazione.

Q

Piano di query

Una serie di passaggi, come le istruzioni, utilizzati per accedere ai dati in un sistema di database relazionale SQL.

regressione del piano di query

Quando un ottimizzatore del servizio di database sceglie un piano non ottimale rispetto a prima di una determinata modifica all'ambiente di database. Questo può essere causato da modifiche a statistiche, vincoli, impostazioni dell'ambiente, associazioni dei parametri di query e aggiornamenti al motore di database.

R

Matrice RACI

Vedi responsabile, responsabile, consultato, informato (RACI).

RAG

Vedi Retrieval Augmented Generation.

ransomware

Un software dannoso progettato per bloccare l'accesso a un sistema informatico o ai dati fino a quando non viene effettuato un pagamento.

Matrice RASCI

Vedi responsabile, responsabile, consultato, informato (RACI).

RCAC

Vedi controllo dell'accesso a righe e colonne.

replica di lettura

Una copia di un database utilizzata per scopi di sola lettura. È possibile indirizzare le query alla replica di lettura per ridurre il carico sul database principale.

riprogettare

Vedi 7 Rs.

obiettivo del punto di ripristino (RPO)

Il periodo di tempo massimo accettabile dall'ultimo punto di ripristino dei dati. Questo determina ciò che si considera una perdita di dati accettabile tra l'ultimo punto di ripristino e l'interruzione del servizio.

obiettivo del tempo di ripristino (RTO)

Il ritardo massimo accettabile tra l'interruzione del servizio e il ripristino del servizio.

rifattorizzare

Vedi 7 R.

Region

Una raccolta di AWS risorse in un'area geografica. Ciascuna Regione AWS è isolata e indipendente dalle altre per fornire tolleranza agli errori, stabilità e resilienza. Per ulteriori informazioni, consulta Specificare cosa può usare Regioni AWS il tuo account.

regressione

Una tecnica di ML che prevede un valore numerico. Ad esempio, per risolvere il problema "A che prezzo verrà venduta questa casa?" un modello di ML potrebbe utilizzare un modello di regressione lineare per prevedere il prezzo di vendita di una casa sulla base di dati noti sulla casa (ad esempio, la metratura).

riospitare

Vedi 7 R.

rilascio

In un processo di implementazione, l'atto di promuovere modifiche a un ambiente di produzione.

trasferisco

Vedi 7 Rs.

ripiattaforma

Vedi 7 Rs.

riacquisto

Vedi 7 Rs.

resilienza

La capacità di un'applicazione di resistere alle interruzioni o di ripristinarle. L'elevata disponibilità e il disaster recovery sono considerazioni comuni quando si pianifica la resilienza in. Cloud AWSPer ulteriori informazioni, vedere Cloud AWS Resilience.

policy basata su risorse

Una policy associata a una risorsa, ad esempio un bucket Amazon S3, un endpoint o una chiave di crittografia. Questo tipo di policy specifica a quali principali è consentito l'accesso, le azioni supportate e qualsiasi altra condizione che deve essere soddisfatta.

matrice di assegnazione di responsabilità (RACI)

Una matrice che definisce i ruoli e le responsabilità di tutte le parti coinvolte nelle attività di migrazione e nelle operazioni cloud. Il nome della matrice deriva dai tipi di responsabilità definiti nella matrice: responsabile (R), responsabile (A), consultato (C) e informato (I). Il tipo di supporto (S) è facoltativo. Se includi il supporto, la matrice viene chiamata matrice RASCI e, se la escludi, viene chiamata matrice RACI.

controllo reattivo

Un controllo di sicurezza progettato per favorire la correzione di eventi avversi o deviazioni dalla baseline di sicurezza. Per ulteriori informazioni, consulta Controlli reattivi in Implementazione dei controlli di sicurezza in AWS.

retain

Vedi 7 R.

andare in pensione

Vedi 7 Rs.

Retrieval Augmented Generation (RAG)

Una tecnologia di intelligenza artificiale generativa in cui un LLM fa riferimento a una fonte di dati autorevole esterna alle sue fonti di dati di formazione prima di generare una risposta. Ad esempio, un modello RAG potrebbe eseguire una ricerca semantica nella knowledge base o nei dati personalizzati di un'organizzazione. Per ulteriori informazioni, consulta Cos'è il RAG.

rotazione

Processo di aggiornamento periodico di un segreto per rendere più difficile l'accesso alle credenziali da parte di un utente malintenzionato.

controllo dell'accesso a righe e colonne (RCAC)

L'uso di espressioni SQL di base e flessibili con regole di accesso definite. RCAC è costituito da autorizzazioni di riga e maschere di colonna.

RPO

Vedi obiettivo del punto di ripristino.

VERSO

Vedi obiettivo del tempo di ripristino.

runbook

Un insieme di procedure manuali o automatizzate necessarie per eseguire un'attività specifica. In genere sono progettati per semplificare operazioni o procedure ripetitive con tassi di errore elevati.

S

SAML 2.0

Uno standard aperto utilizzato da molti provider di identità (IdPs). Questa funzionalità abilita il single sign-on (SSO) federato, in modo che gli utenti possano accedere Console di gestione AWS o chiamare le operazioni AWS API senza che tu debba creare un utente in IAM per tutti i membri dell'organizzazione. Per ulteriori informazioni sulla federazione basata su SAML 2.0, consulta Informazioni sulla federazione basata su SAML 2.0 nella documentazione di IAM.

SCADA

Vedi controllo di supervisione e acquisizione dati.

SCP

Vedi la politica di controllo del servizio.

Secret

In Gestione dei segreti AWS, informazioni riservate o riservate, come una password o le credenziali utente, archiviate in forma crittografata. È costituito dal valore segreto e dai relativi metadati. Il valore segreto può essere binario, una stringa singola o più stringhe. Per ulteriori informazioni, consulta Cosa c'è in un segreto di Secrets Manager? nella documentazione di Secrets Manager.

sicurezza fin dalla progettazione

Un approccio di ingegneria dei sistemi che tiene conto della sicurezza durante l'intero processo di sviluppo.

controllo di sicurezza

Un guardrail tecnico o amministrativo che impedisce, rileva o riduce la capacità di un autore di minacce di sfruttare una vulnerabilità di sicurezza. Esistono quattro tipi principali di controlli di sicurezza: preventivi, investigativi, reattivi e proattivi.

rafforzamento della sicurezza

Il processo di riduzione della superficie di attacco per renderla più resistente agli attacchi. Può includere azioni come la rimozione di risorse che non sono più necessarie, l'implementazione di best practice di sicurezza che prevedono la concessione del privilegio minimo o la disattivazione di funzionalità non necessarie nei file di configurazione.

sistema di gestione delle informazioni e degli eventi di sicurezza (SIEM)

Strumenti e servizi che combinano sistemi di gestione delle informazioni di sicurezza (SIM) e sistemi di gestione degli eventi di sicurezza (SEM). Un sistema SIEM raccoglie, monitora e analizza i dati da server, reti, dispositivi e altre fonti per rilevare minacce e violazioni della sicurezza e generare avvisi.

automazione della risposta alla sicurezza

Un'azione predefinita e programmata progettata per rispondere o porre rimedio automaticamente a un evento di sicurezza. Queste automazioni fungono da controlli di sicurezza investigativi o reattivi che aiutano a implementare le migliori pratiche di sicurezza. AWS Esempi di azioni di risposta automatizzate includono la modifica di un gruppo di sicurezza VPC, l'applicazione di patch a un'istanza Amazon EC2 o la rotazione delle credenziali.

Crittografia lato server

Crittografia dei dati a destinazione, da parte di chi li riceve. Servizio AWS

Policy di controllo dei servizi (SCP)

Una politica che fornisce il controllo centralizzato sulle autorizzazioni per tutti gli account di un'organizzazione in. AWS Organizations SCPs definire barriere o fissare limiti alle azioni che un amministratore può delegare a utenti o ruoli. È possibile utilizzarli SCPs come elenchi consentiti o elenchi di rifiuto, per specificare quali servizi o azioni sono consentiti o proibiti. Per ulteriori informazioni, consulta le politiche di controllo del servizio nella AWS Organizations documentazione.

endpoint del servizio

L'URL del punto di ingresso per un Servizio AWS. Puoi utilizzare l'endpoint per connetterti a livello di programmazione al servizio di destinazione. Per ulteriori informazioni, consulta Endpoint del Servizio AWS nei Riferimenti generali di AWS.

accordo sul livello di servizio (SLA)

Un accordo che chiarisce ciò che un team IT promette di offrire ai propri clienti, ad esempio l'operatività e le prestazioni del servizio.

indicatore del livello di servizio (SLI)

Misurazione di un aspetto prestazionale di un servizio, ad esempio il tasso di errore, la disponibilità o la velocità effettiva.

obiettivo a livello di servizio (SLO)

Una metrica target che rappresenta lo stato di un servizio, misurato da un indicatore del livello di servizio.

Modello di responsabilità condivisa

Un modello che descrive la responsabilità condivisa AWS per la sicurezza e la conformità del cloud. AWS è responsabile della sicurezza del cloud, mentre tu sei responsabile della sicurezza nel cloud. Per ulteriori informazioni, consulta Modello di responsabilità condivisa.

SIEM

Vedi il sistema di gestione delle informazioni e degli eventi sulla sicurezza.

punto di errore singolo (SPOF)

Un guasto in un singolo componente critico di un'applicazione che può disturbare il sistema.

SLAM

Vedi il contratto sul livello di servizio.

SLI

Vedi l'indicatore del livello di servizio.

LENTA

Vedi obiettivo del livello di servizio.

split-and-seed modello

Un modello per dimensionare e accelerare i progetti di modernizzazione. Man mano che vengono definite nuove funzionalità e versioni dei prodotti, il team principale si divide per creare nuovi team di prodotto. Questo aiuta a dimensionare le capacità e i servizi dell'organizzazione, migliora la produttività degli sviluppatori e supporta una rapida innovazione. Per ulteriori informazioni, vedere Approccio graduale alla modernizzazione delle applicazioni in. Cloud AWS

SPOF

Vedi punto di errore singolo.

schema a stella

Una struttura organizzativa di database che utilizza un'unica tabella dei fatti di grandi dimensioni per archiviare i dati transazionali o misurati e utilizza una o più tabelle dimensionali più piccole per memorizzare gli attributi dei dati. Questa struttura è progettata per l'uso in un data warehouse o per scopi di business intelligence.

modello del fico strangolatore

Un approccio alla modernizzazione dei sistemi monolitici mediante la riscrittura e la sostituzione incrementali delle funzionalità del sistema fino alla disattivazione del sistema legacy. Questo modello utilizza l'analogia di una pianta di fico che cresce fino a diventare un albero robusto e alla fine annienta e sostituisce il suo ospite. Il modello è stato introdotto da Martin Fowler come metodo per gestire il rischio durante la riscrittura di sistemi monolitici. Per un esempio di come applicare questo modello, consulta Modernizzazione incrementale dei servizi Web legacy di Microsoft ASP.NET (ASMX) mediante container e Gateway Amazon API.

sottorete

Un intervallo di indirizzi IP nel VPC. Una sottorete deve risiedere in una singola zona di disponibilità.

controllo di supervisione e acquisizione dati (SCADA)

Nella produzione, un sistema che utilizza hardware e software per monitorare gli asset fisici e le operazioni di produzione.

crittografia simmetrica

Un algoritmo di crittografia che utilizza la stessa chiave per crittografare e decrittografare i dati.

test sintetici

Test di un sistema in modo da simulare le interazioni degli utenti per rilevare potenziali problemi o monitorare le prestazioni. Puoi usare Amazon CloudWatch Synthetics per creare questi test.

prompt di sistema

Una tecnica per fornire contesto, istruzioni o linee guida a un LLM per indirizzarne il comportamento. I prompt di sistema aiutano a impostare il contesto e stabilire regole per le interazioni con gli utenti.

T

tag

Coppie chiave-valore che fungono da metadati per l'organizzazione delle risorse. AWS Con i tag è possibile a gestire, identificare, organizzare, cercare e filtrare le risorse. Per ulteriori informazioni, consulta Tagging delle risorse AWS.

variabile di destinazione

Il valore che stai cercando di prevedere nel machine learning supervisionato. Questo è indicato anche come variabile di risultato. Ad esempio, in un ambiente di produzione la variabile di destinazione potrebbe essere un difetto del prodotto.

elenco di attività

Uno strumento che viene utilizzato per tenere traccia dei progressi tramite un runbook. Un elenco di attività contiene una panoramica del runbook e un elenco di attività generali da completare. Per ogni attività generale, include la quantità stimata di tempo richiesta, il proprietario e lo stato di avanzamento.

ambiente di test

Vedi ambiente.

training

Fornire dati da cui trarre ispirazione dal modello di machine learning. I dati di training devono contenere la risposta corretta. L'algoritmo di apprendimento trova nei dati di addestramento i pattern che mappano gli attributi dei dati di input al target (la risposta che si desidera prevedere). Produce un modello di ML che acquisisce questi modelli. Puoi quindi utilizzare il modello di ML per creare previsioni su nuovi dati di cui non si conosce il target.

Transit Gateway

Un hub di transito di rete che puoi utilizzare per interconnettere le tue reti VPCs e quelle locali. Per ulteriori informazioni, consulta Cos'è un gateway di transito nella AWS Transit Gateway documentazione.

flusso di lavoro basato su trunk

Un approccio in cui gli sviluppatori creano e testano le funzionalità localmente in un ramo di funzionalità e quindi uniscono tali modifiche al ramo principale. Il ramo principale viene quindi integrato negli ambienti di sviluppo, preproduzione e produzione, in sequenza.

Accesso attendibile

Concessione delle autorizzazioni a un servizio specificato dall'utente per eseguire attività all'interno dell'organizzazione AWS Organizations e nei suoi account per conto dell'utente. Il servizio attendibile crea un ruolo collegato al servizio in ogni account, quando tale ruolo è necessario, per eseguire attività di gestione per conto dell'utente. Per ulteriori informazioni, consulta Utilizzo AWS Organizations con altri AWS servizi nella AWS Organizations documentazione.

regolazione

Modificare alcuni aspetti del processo di training per migliorare la precisione del modello di ML. Ad esempio, puoi addestrare il modello di ML generando un set di etichette, aggiungendo etichette e quindi ripetendo questi passaggi più volte con impostazioni diverse per ottimizzare il modello.

team da due pizze

Una piccola DevOps squadra che puoi sfamare con due pizze. Un team composto da due persone garantisce la migliore opportunità possibile di collaborazione nello sviluppo del software.

U

incertezza

Un concetto che si riferisce a informazioni imprecise, incomplete o sconosciute che possono minare l'affidabilità dei modelli di machine learning predittivi. Esistono due tipi di incertezza: l'incertezza epistemica, che è causata da dati limitati e incompleti, mentre l'incertezza aleatoria è causata dal rumore e dalla casualità insiti nei dati. Per ulteriori informazioni, consulta la guida Quantificazione dell'incertezza nei sistemi di deep learning.

compiti indifferenziati

Conosciuto anche come sollevamento di carichi pesanti, è un lavoro necessario per creare e far funzionare un'applicazione, ma che non apporta valore diretto all'utente finale né offre vantaggi competitivi. Esempi di attività indifferenziate includono l'approvvigionamento, la manutenzione e la pianificazione della capacità.

ambienti superiori

Vedi ambiente.

V

vacuum

Un'operazione di manutenzione del database che prevede la pulizia dopo aggiornamenti incrementali per recuperare lo spazio di archiviazione e migliorare le prestazioni.

controllo delle versioni

Processi e strumenti che tengono traccia delle modifiche, ad esempio le modifiche al codice di origine in un repository.

Peering VPC

Una connessione tra due VPCs che consente di indirizzare il traffico utilizzando indirizzi IP privati. Per ulteriori informazioni, consulta Che cos'è il peering VPC? nella documentazione di Amazon VPC.

vulnerabilità

Un difetto software o hardware che compromette la sicurezza del sistema.

W

cache calda

Una cache del buffer che contiene dati correnti e pertinenti a cui si accede frequentemente. L'istanza di database può leggere dalla cache del buffer, il che richiede meno tempo rispetto alla lettura dalla memoria dal disco principale.

dati caldi

Dati a cui si accede raramente. Quando si eseguono interrogazioni di questo tipo di dati, in genere sono accettabili query moderatamente lente.

funzione finestra

Una funzione SQL che esegue un calcolo su un gruppo di righe che si riferiscono in qualche modo al record corrente. Le funzioni della finestra sono utili per l'elaborazione di attività, come il calcolo di una media mobile o l'accesso al valore delle righe in base alla posizione relativa della riga corrente.

Carico di lavoro

Una raccolta di risorse e codice che fornisce valore aziendale, ad esempio un'applicazione rivolta ai clienti o un processo back-end.

flusso di lavoro

Gruppi funzionali in un progetto di migrazione responsabili di una serie specifica di attività. Ogni flusso di lavoro è indipendente ma supporta gli altri flussi di lavoro del progetto. Ad esempio, il flusso di lavoro del portfolio è responsabile della definizione delle priorità delle applicazioni, della pianificazione delle ondate e della raccolta dei metadati di migrazione. Il flusso di lavoro del portfolio fornisce queste risorse al flusso di lavoro di migrazione, che quindi migra i server e le applicazioni.

VERME

Vedi scrivere una volta, leggere molti.

WQF

Vedi AWS Workload Qualification Framework.

scrivi una volta, leggi molte (WORM)

Un modello di storage che scrive i dati una sola volta e ne impedisce l'eliminazione o la modifica. Gli utenti autorizzati possono leggere i dati tutte le volte che è necessario, ma non possono modificarli. Questa infrastruttura di archiviazione dei dati è considerata immutabile.

Z

exploit zero-day

Un attacco, in genere malware, che sfrutta una vulnerabilità zero-day.

vulnerabilità zero-day

Un difetto o una vulnerabilità assoluta in un sistema di produzione. Gli autori delle minacce possono utilizzare questo tipo di vulnerabilità per attaccare il sistema. Gli sviluppatori vengono spesso a conoscenza della vulnerabilità causata dall'attacco.

prompt zero-shot

Fornire a un LLM le istruzioni per eseguire un'attività ma non esempi (immagini) che possano aiutarla. Il LLM deve utilizzare le sue conoscenze pre-addestrate per gestire l'attività. L'efficacia del prompt zero-shot dipende dalla complessità dell'attività e dalla qualità del prompt. Vedi anche few-shot prompting.

applicazione zombie

Un'applicazione che prevede un utilizzo CPU e memoria inferiore al 5%. In un progetto di migrazione, è normale ritirare queste applicazioni.

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.