Struttura del ciclo di vita della resilienza

AWS Guida prescrittiva

AWS Guida prescrittiva: Struttura del ciclo di vita della resilienza

Table of Contents

Framework del ciclo di vita della resilienza: un approccio continuo al miglioramento della resilienza

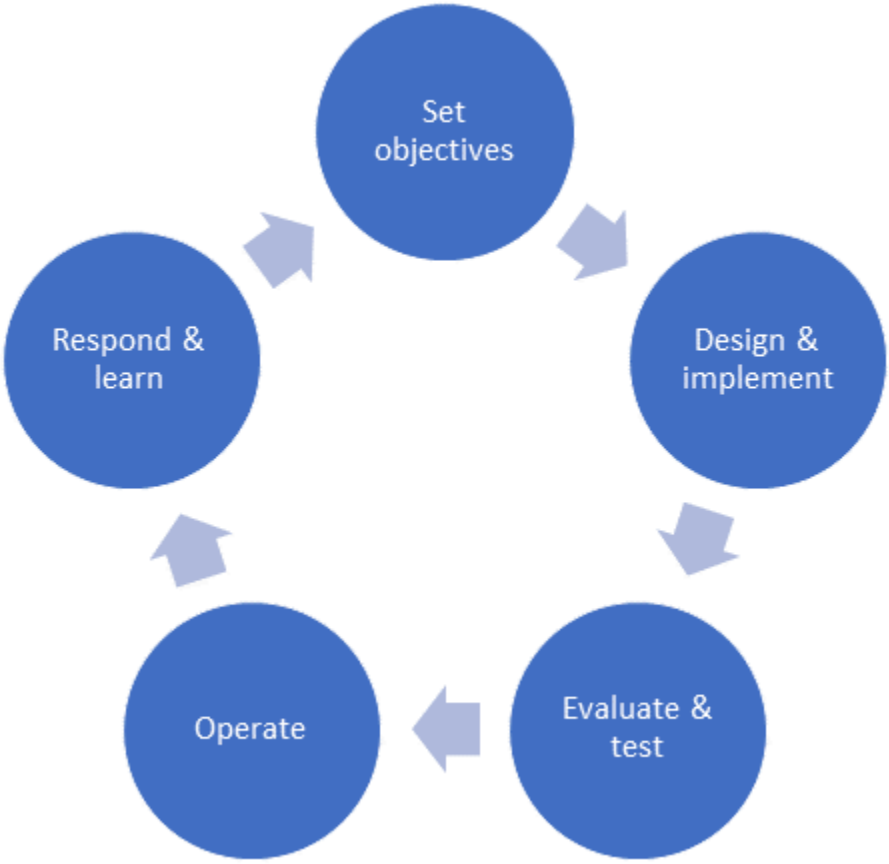
Amazon Web Services (collaboratori)

Ottobre 2023 (cronologia del documento)

Le organizzazioni moderne oggi affrontano un numero sempre crescente di sfide legate alla resilienza, soprattutto perché le aspettative dei clienti si spostano verso una mentalità sempre attiva e sempre disponibile. I team remoti e le applicazioni complesse e distribuite si accompagnano a una crescente necessità di rilasci frequenti. Di conseguenza, un'organizzazione e le sue applicazioni devono essere più resilienti che mai.

AWS definisce la resilienza come la capacità di un'applicazione di resistere o riprendersi da interruzioni, comprese quelle relative all'infrastruttura, ai servizi dipendenti, alle configurazioni errate e ai problemi transitori della rete. (Vedi Resilienza e i componenti dell'affidabilità nella documentazione AWS Well-Architected Framework Reliability Pillar.) Tuttavia, per raggiungere il livello di resilienza desiderato, spesso sono necessari dei compromessi. La complessità operativa, la complessità ingegneristica e i costi dovranno essere valutati e adattati di conseguenza.

Sulla base di anni di collaborazione con clienti e team interni, AWS ha sviluppato un framework per il ciclo di vita della resilienza che raccoglie gli insegnamenti e le migliori pratiche in materia di resilienza. Il framework delinea cinque fasi chiave illustrate nel diagramma seguente. In ogni fase è possibile utilizzare strategie, servizi e meccanismi per migliorare la propria posizione di resilienza.



Queste fasi sono illustrate nelle seguenti sezioni di questa guida:

- Fase 1: fissare gli obiettivi
- Fase 2: progettazione e implementazione
- Fase 3: valutazione e test
- Fase 4: Operare
- Fase 5: Rispondi e impara

Termini e definizioni

I concetti di resilienza di ogni fase vengono applicati a diversi livelli, dai singoli componenti a interi sistemi. L'implementazione di questi concetti richiede una chiara definizione di diversi termini:

- Un componente è un elemento che svolge una funzione e consiste in risorse software e tecnologiche. Esempi di componenti includono la configurazione del codice, l'infrastruttura come la

rete o persino i server, gli archivi dati e le dipendenze esterne come i dispositivi di autenticazione a più fattori (MFA).

- Un'applicazione è una raccolta di componenti che offre valore aziendale, come una vetrina web rivolta ai clienti o il processo di backend che migliora i modelli di machine learning. Un'applicazione può essere costituita da un sottoinsieme di componenti in un unico AWS account oppure può essere una raccolta di più componenti che si estendono su più regioni. Account AWS
- Un sistema è un insieme di applicazioni, persone e processi necessari per gestire una determinata funzione aziendale. Comprende l'applicazione necessaria per eseguire una funzione, i processi operativi come l'integrazione continua e la distribuzione continua (CI/CD), l'osservabilità, la gestione della configurazione, la risposta agli incidenti e il disaster recovery e gli operatori che gestiscono tali attività.
- Un'interruzione è un evento che impedisce all'applicazione di svolgere correttamente le proprie funzioni aziendali.
- La compromissione è l'effetto che un'interruzione ha su un'applicazione se non viene mitigata. Le applicazioni possono essere compromesse se subiscono una serie di interruzioni.

Resilienza continua

Il ciclo di vita della resilienza è un processo continuo. Anche all'interno della stessa organizzazione, i team addetti alle applicazioni potrebbero operare a diversi livelli di completezza in ogni fase, a seconda dei requisiti dell'applicazione. Tuttavia, più ogni fase è completa, maggiore sarà il livello di resilienza dell'applicazione.

È necessario considerare il ciclo di vita della resilienza come un processo standard che l'organizzazione può rendere operativo. AWS ha intenzionalmente modellato il ciclo di vita della resilienza in modo che sia simile al ciclo di vita dello sviluppo del software (SDLC), con l'obiettivo di incorporare pianificazione, test e apprendimento in tutti i processi operativi durante lo sviluppo e il funzionamento delle applicazioni. Come per molti processi di sviluppo agili, il ciclo di vita della resilienza può essere ripetuto ad ogni iterazione del processo di sviluppo. Ti consigliamo di approfondire le pratiche all'interno di ogni fase del ciclo di vita progressivamente nel tempo.

Fase 1: fissare gli obiettivi

Capire quale livello di resilienza è necessario e come lo si misurerà è la base per la fase degli obiettivi prefissati. È difficile migliorare qualcosa se non si ha un obiettivo e non si può misurarlo.

Non tutte le applicazioni richiedono lo stesso livello di resilienza. Quando stabilite degli obiettivi, considerate il livello richiesto per effettuare gli investimenti e i compromessi corretti. Una buona analogia è quella di un'auto: ha quattro pneumatici ma ne trasporta solo una di scorta. La possibilità di avere più pneumatici sgonfi durante una corsa è bassa e disporre di pezzi di ricambio aggiuntivi potrebbe compromettere altre caratteristiche, come lo spazio di carico o il risparmio di carburante, quindi si tratta di un compromesso ragionevole.

Dopo aver definito gli obiettivi, si implementano i controlli di osservabilità nelle fasi successive (Fase 2: progettazione e implementazione e Fase 4: Operazione) per capire se gli obiettivi vengono raggiunti.

Mappatura delle applicazioni critiche

La definizione degli obiettivi di resilienza non dovrebbe essere esclusivamente una conversazione tecnica. Iniziate invece con un approccio orientato al business per capire cosa dovrebbe offrire l'applicazione e le conseguenze di un eventuale deterioramento. Questa comprensione degli obiettivi aziendali si estende quindi a cascata ad aree come l'architettura, l'ingegneria e le operazioni. Qualsiasi obiettivo di resilienza definito potrebbe essere applicato a tutte le applicazioni, ma il modo in cui gli obiettivi vengono misurati spesso varia a seconda della funzione dell'applicazione. È possibile che stiate eseguendo un'applicazione fondamentale per l'azienda e, in caso di danneggiamento di tale applicazione, l'organizzazione potrebbe perdere entrate significative o subire danni alla reputazione. In alternativa, potresti avere un'altra applicazione che non è altrettanto importante e in grado di tollerare alcuni tempi di inattività senza influire negativamente sulla capacità aziendale dell'organizzazione.

Ad esempio, si pensi a un'applicazione di gestione degli ordini per un'azienda di vendita al dettaglio. Se i componenti dell'applicazione di gestione degli ordini sono danneggiati e non funzionano correttamente, le nuove vendite non verranno effettuate. Questa società di vendita al dettaglio ha anche una caffetteria per i suoi dipendenti che si trova in uno dei suoi edifici. La caffetteria dispone di un menu online a cui i dipendenti possono accedere su una pagina Web statica. Se questa pagina Web non è più disponibile, alcuni dipendenti potrebbero lamentarsi, ma ciò non causerà necessariamente danni finanziari all'azienda. Sulla base di questo esempio, l'azienda probabilmente

sceglierebbe di avere obiettivi di resilienza più aggressivi per l'applicazione di gestione degli ordini, ma non effettuerà un investimento significativo per garantire la resilienza dell'applicazione web.

Identificare le applicazioni più critiche, dove dedicare il massimo impegno e dove trovare i compromessi è importante tanto quanto poter misurare la resilienza di un'applicazione in produzione. Per comprendere meglio l'impatto della compromissione, è possibile eseguire un'analisi dell'impatto aziendale (BIA). Una BIA fornisce un approccio strutturato e sistematico per identificare e assegnare priorità alle applicazioni aziendali critiche, valutare potenziali rischi e impatti e identificare le dipendenze di supporto. La BIA aiuta a quantificare il costo dei tempi di inattività per le applicazioni più importanti dell'organizzazione. Questa metrica aiuta a delineare quanto costerà se un'applicazione specifica è danneggiata e non è in grado di completare la sua funzione. Nell'esempio precedente, se l'applicazione di gestione degli ordini è compromessa, l'attività di vendita al dettaglio potrebbe perdere entrate significative.

Mappatura delle storie degli utenti

Durante il processo di BIA, è possibile scoprire che un'applicazione è responsabile di più di una funzione aziendale o che una funzione aziendale richiede più applicazioni. Utilizzando il precedente esempio di una società di vendita al dettaglio, la funzione di gestione degli ordini potrebbe richiedere applicazioni separate per il checkout, la promozione e la determinazione dei prezzi. Se un'applicazione si guasta, l'impatto potrebbe essere avvertito dall'azienda e dagli utenti che interagiscono con l'azienda. Ad esempio, l'azienda potrebbe non essere in grado di aggiungere nuovi ordini, fornire l'accesso a promozioni e sconti o aggiornare il prezzo dei propri prodotti. Queste diverse funzioni richieste dalla funzione di gestione degli ordini potrebbero fare affidamento su più applicazioni. Queste funzioni potrebbero anche avere più dipendenze esterne, il che rende troppo complesso il processo di raggiungimento di una resilienza puramente incentrata sui componenti. Un modo migliore per gestire questo scenario consiste nel concentrarsi sulle storie degli utenti, che descrivono l'esperienza che gli utenti si aspettano quando interagiscono con un'applicazione o un insieme di applicazioni.

Concentrarsi sulle storie degli utenti aiuta a capire quali aspetti dell'esperienza del cliente sono più importanti, in modo da poter creare meccanismi di protezione da minacce specifiche. Nell'esempio precedente, una storia utente potrebbe essere checkout, che coinvolge l'applicazione di checkout e dipende dall'applicazione di determinazione dei prezzi. Un'altra storia utente potrebbe riguardare la visualizzazione di promozioni, che riguarda l'applicazione di promozione. Dopo aver mappato le applicazioni più importanti e le relative storie utente, puoi iniziare a definire le metriche da utilizzare per misurare la resilienza di queste storie utente. Queste metriche possono essere applicate a un intero portfolio o a singole storie utente.

Definizione delle misurazioni

Gli obiettivi dei punti di ripristino (RPO), gli obiettivi dei tempi di ripristino (RTO) e gli obiettivi a livello di servizio (SLO) sono misure standard del settore utilizzate per valutare la resilienza di un determinato sistema. L'RPO si riferisce alla quantità di perdita di dati che l'azienda può tollerare in caso di guasto, mentre l'RTO misura la velocità con cui un'applicazione deve essere nuovamente disponibile dopo un'interruzione. Queste due metriche vengono misurate in unità di tempo: secondi, minuti e ore. È inoltre possibile misurare il periodo di tempo durante il quale l'applicazione funziona correttamente, ovvero esegue le sue funzioni come previsto ed è accessibile ai suoi utenti. Questi SLO descrivono in dettaglio il livello di servizio previsto che i clienti riceveranno e vengono misurati mediante parametri quali la percentuale (%) di richieste soddisfatte senza errori entro un tempo di risposta inferiore a un secondo (ad esempio, il 99,99% delle richieste riceverà una risposta ogni mese). RPO e RTO sono correlati alle strategie di disaster recovery, presupponendo che si verifichino interruzioni nel funzionamento delle applicazioni e nei processi di ripristino che vanno dal ripristino dei backup al reindirizzamento del traffico degli utenti. Gli SLO vengono risolti implementando controlli ad alta disponibilità, che tendono a ridurre i tempi di inattività di un'applicazione.

Le metriche SLO sono comunemente utilizzate nella definizione degli accordi sul livello di servizio (SLA), che sono contratti tra fornitori di servizi e utenti finali. Gli SLA di solito prevedono impegni finanziari e definiscono le sanzioni che devono essere pagate dal fornitore se questi accordi non vengono rispettati. Tuttavia, uno SLA non è una misura del vostro livello di resilienza e un aumento di uno SLA non rende l'applicazione più resiliente.

Puoi iniziare a definire i tuoi obiettivi in base a SLO, RPO e RTO. Dopo aver definito gli obiettivi di resilienza e aver acquisito una chiara comprensione degli obiettivi RPO e RTO, è possibile eseguire una valutazione dell'architettura AWS Resilience Hubper scoprire potenziali punti deboli legati alla resilienza. AWS Resilience Hub valuta un'architettura applicativa rispetto alle best practice di AWS Well-Architected Framework e condivide le linee guida per la correzione nel contesto di ciò che deve essere specificamente migliorato per soddisfare gli obiettivi RTO e RPO definiti.

Creazione di misurazioni aggiuntive

RPO, RTO e SLO sono buoni indicatori di resilienza, ma puoi anche pensare agli obiettivi da una prospettiva aziendale e definire obiettivi relativi alle funzioni dell'applicazione. Ad esempio, il vostro obiettivo potrebbe essere: gli ordini al minuto andati a buon fine rimarranno superiori al 98% se la latenza tra il mio frontend e il backend aumenta del 40%.Oppure: gli stream avviati al secondo

rimarranno entro una deviazione standard dalla media anche in caso di perdita di un componente specifico. È inoltre possibile creare obiettivi per ottenere una riduzione del tempo medio di ripristino (MTTR) per tutti i tipi di errore noti; ad esempio: i tempi di ripristino verranno ridotti del x% se si verifica uno di questi problemi noti. La creazione di obiettivi in linea con le esigenze aziendali consente di anticipare i tipi di guasti che l'applicazione dovrebbe tollerare. Inoltre, consente di identificare gli approcci per ridurre la probabilità di compromissione dell'applicazione.

Se pensate all'obiettivo di continuare a funzionare se perdete il 5% delle istanze che alimentano l'applicazione, potreste decidere che l'applicazione debba essere prescalata o avere la capacità di scalare abbastanza velocemente da supportare il traffico aggiuntivo causato durante quell'evento. In alternativa, potreste decidere di sfruttare diversi modelli architettonici, come descritto nella sezione Fase 2: Progettazione e implementazione.

È inoltre necessario implementare misure di osservabilità per i propri obiettivi aziendali specifici. Ad esempio, puoi monitorare il tasso medio degli ordini, il prezzo medio degli ordini, il numero medio di abbonamenti o altre metriche che possono fornire informazioni sullo stato di salute dell'azienda in base al comportamento dell'applicazione. Implementando funzionalità di osservabilità per la tua applicazione, puoi creare allarmi e intervenire se queste metriche superano i limiti definiti. L'osservabilità è trattata più dettagliatamente nella sezione Stage 4: Operate.

Fase 2: progettazione e implementazione

Nella fase precedente, hai definito i tuoi obiettivi di resilienza. Ora, nella fase di progettazione e implementazione, si cerca di anticipare le modalità di guasto e di identificare le scelte di progettazione, sulla base degli obiettivi fissati nella fase precedente. Inoltre, definisci strategie per la gestione delle modifiche e sviluppi il codice software e la configurazione dell'infrastruttura. Nelle sezioni seguenti vengono illustrate le AWS best practice da prendere in considerazione nel tenere conto di compromessi quali costi, complessità e costi operativi.

AWS Well-Architected Framework

Quando si progetta l'applicazione in base agli obiettivi di resilienza desiderati, è necessario valutare più fattori e trovare compromessi sull'architettura più ottimale. Per creare un'applicazione altamente resiliente è necessario considerare aspetti di progettazione, costruzione e implementazione, sicurezza e operazioni. AWS Well-Architected Framework fornisce una serie di best practice, principi di progettazione e modelli architettonici per aiutarti a progettare applicazioni resilienti su. AWS I sei pilastri del AWS Well-Architected Framework forniscono le migliori pratiche per progettare e gestire sistemi resilienti, sicuri, efficienti, convenienti e sostenibili. Il framework offre un modo per misurare in modo coerente le architetture rispetto alle migliori pratiche e identificare le aree di miglioramento.

Di seguito sono riportati alcuni esempi di come il AWS Well-Architected Framework può aiutarti a progettare e implementare applicazioni che soddisfano i tuoi obiettivi di resilienza:

- Il pilastro dell'affidabilità: il pilastro dell'affidabilità sottolinea l'importanza di creare applicazioni in grado di funzionare correttamente e in modo coerente, anche in caso di guasti o interruzioni. Ad esempio, AWS Well-Architected Framework consiglia di utilizzare un'architettura di microservizi per rendere le applicazioni più piccole e semplici, in modo da poter distinguere tra le esigenze di disponibilità dei diversi componenti all'interno dell'applicazione. È inoltre possibile trovare descrizioni dettagliate delle migliori pratiche per la creazione di applicazioni utilizzando il throttling, il retry with exponential back off, il fail-fast (load shedding), l'idempotenza, il lavoro costante, gli interruttori automatici e la stabilità statica.
- Revisione completa: il AWS Well-Architected Framework incoraggia una revisione completa dell'architettura rispetto alle migliori pratiche e ai principi di progettazione. Fornisce un modo per misurare in modo coerente le architetture e identificare le aree di miglioramento.

- Gestione del rischio: AWS Well-Architected Framework consente di identificare e gestire i rischi che potrebbero influire sull'affidabilità dell'applicazione. Affrontando i potenziali scenari di errore in modo proattivo, è possibile ridurne la probabilità o il conseguente deterioramento.
- Miglioramento continuo: la resilienza è un processo continuo e il AWS Well-Architected Framework enfatizza il miglioramento continuo. Rivedendo e perfezionando regolarmente l'architettura e i processi in base alle linee guida del AWS Well-Architected Framework, puoi assicurarti che i tuoi sistemi rimangano resilienti di fronte alle sfide e ai requisiti in evoluzione.

Comprendere le dipendenze

Comprendere le dipendenze di un sistema è fondamentale per la resilienza. Le dipendenze includono le connessioni tra i componenti all'interno di un'applicazione e le connessioni ai componenti esterni all'applicazione, come API di terze parti e servizi condivisi di proprietà dell'azienda. La comprensione di queste connessioni consente di isolare e gestire le interruzioni, poiché il danneggiamento di un componente può influire su altri componenti. Queste conoscenze aiutano gli ingegneri a valutare l'impatto delle menomazioni e a pianificare di conseguenza e a garantire che le risorse vengano utilizzate in modo efficace. La comprensione delle dipendenze consente di creare strategie alternative e coordinare i processi di ripristino. Inoltre, consente di determinare i casi in cui è possibile sostituire una dipendenza rigida con una dipendenza morbida, in modo che l'applicazione possa continuare a svolgere la propria funzione aziendale in caso di riduzione della dipendenza. Le dipendenze influenzano anche le decisioni sul bilanciamento del carico e sulla scalabilità delle applicazioni. Comprendere le dipendenze è fondamentale quando si apportano modifiche all'applicazione, perché può aiutare a determinare potenziali rischi e impatti. Queste conoscenze vi aiutano a creare applicazioni stabili e resilienti, aiutandovi nella gestione degli errori, nella valutazione dell'impatto, nel ripristino delle interruzioni, nel bilanciamento del carico, nella scalabilità e nella gestione delle modifiche. È possibile tenere traccia delle dipendenze manualmente o utilizzare strumenti e servizi per comprendere le dipendenze delle applicazioni AWS X-Raydistribuite.

Strategie di disaster recovery

Una strategia di disaster recovery (DR) svolge un ruolo fondamentale nella creazione e nella gestione di applicazioni resilienti, principalmente garantendo la continuità aziendale. Garantisce che le operazioni aziendali cruciali possano persistere con il minor danno possibile, anche durante eventi catastrofici, riducendo così al minimo i tempi di inattività e la potenziale perdita di fatturato. Le strategie di ripristino di emergenza sono essenziali per la protezione dei dati perché spesso incorporano backup e repliche regolari dei dati su più sedi, il che aiuta a salvaguardare preziose

informazioni aziendali e aiuta a prevenire la perdita totale in caso di emergenza. Inoltre, molti settori sono regolati da politiche che richiedono alle aziende di adottare una strategia di DR per proteggere i dati sensibili e garantire che i servizi rimangano disponibili in caso di emergenza. Garantendo una riduzione minima del servizio, una strategia di DR rafforza anche la fiducia e la soddisfazione dei clienti. Una strategia di ripristino di emergenza ben implementata e utilizzata frequentemente riduce i tempi di ripristino dopo un disastro e aiuta a garantire che le applicazioni vengano rapidamente ripristinate online. Inoltre, i disastri possono comportare costi considerevoli, non solo a causa delle perdite di entrate dovute ai tempi di inattività, ma anche a causa delle spese di ripristino di applicazioni e dati. Una strategia di ripristino di emergenza ben progettata aiuta a proteggersi da queste perdite finanziarie.

La strategia scelta dipende dalle esigenze specifiche dell'applicazione, dall'RTO e dall'RPO e dal budget a disposizione. AWS Elastic Disaster Recoveryè un servizio di resilienza appositamente progettato che puoi utilizzare per aiutarti a implementare la tua strategia di DR per applicazioni locali e basate sul cloud.

Per ulteriori informazioni, consulta Disaster Recovery of Workloads on e Multi-Region Fundamentals sul sito Web. AWSAWS AWS

Definizione delle strategie CI/CD

Una delle cause più comuni di compromissione delle applicazioni è il codice o altre modifiche che alterano l'applicazione da uno stato di funzionamento precedentemente noto. Se non si affronta con attenzione la gestione delle modifiche, ciò può causare problemi frequenti. La frequenza delle modifiche aumenta le possibilità di impatto. Tuttavia, apportando modifiche meno frequentemente si ottengono set di modifiche più ampi, che hanno molte più probabilità di comprometterli a causa della loro elevata complessità. Le pratiche di integrazione continua e distribuzione continua (CI/CD) sono progettate per mantenere le modifiche piccole e frequenti (con conseguente aumento della produttività), sottoponendo al contempo ogni modifica a un elevato livello di ispezione mediante l'automazione. Alcune delle strategie fondamentali sono:

- Automazione completa: il concetto fondamentale di CI/CD è automatizzare il più possibile i processi di creazione e implementazione. Ciò include la creazione, il test, l'implementazione e persino il monitoraggio. Le pipeline automatizzate aiutano a ridurre la possibilità di errori umani, garantiscono la coerenza e rendono il processo più affidabile ed efficiente.
- Sviluppo basato sui test (TDD): scrivi i test prima di scrivere il codice dell'applicazione. Questa pratica garantisce che a tutto il codice siano associati dei test, il che migliora l'affidabilità del

codice e la qualità dell'ispezione automatizzata. Questi test vengono eseguiti nella pipeline CI per convalidare le modifiche.

- Commit e integrazioni frequenti: incoraggia gli sviluppatori a eseguire spesso il codice e a eseguire spesso le integrazioni. Le modifiche piccole e frequenti sono più facili da testare ed eseguire il debug, il che riduce il rischio di problemi significativi. L'automazione riduce il costo di ogni commit e implementazione, rendendo possibili integrazioni frequenti.
- Infrastruttura immutabile: trattate i server e gli altri componenti dell'infrastruttura come entità statiche e immutabili. Sostituisci l'infrastruttura invece di modificarla il più possibile e crea una nuova infrastruttura utilizzando codice testato e distribuito attraverso la tua pipeline.
- Meccanismo di rollback: disponi sempre di un modo semplice, affidabile e frequentemente testato per ripristinare le modifiche se qualcosa va storto. La possibilità di tornare rapidamente al precedente stato di buono stato noto è essenziale per la sicurezza dell'implementazione. Questo può essere un semplice pulsante per tornare allo stato precedente oppure può essere completamente automatizzato e attivato da allarmi.
- Controllo della versione: conserva tutto il codice dell'applicazione, la configurazione e persino l'infrastruttura come codice in un repository controllato dalla versione. Questa pratica consente di tenere traccia facilmente delle modifiche e di ripristinarle se necessario.
- Implementazioni Canary e implementazioni blu/verdi: la distribuzione di nuove versioni dell'applicazione in un sottoinsieme dell'infrastruttura o la manutenzione di due ambienti (blu/verde) consente di verificare il comportamento di una modifica in produzione e di ripristinarla rapidamente se necessario.

CI/CD non riguarda solo gli strumenti ma anche la cultura. Creare una cultura che valorizzi l'automazione, i test e l'apprendimento dagli errori è tanto importante quanto implementare gli strumenti e i processi giusti. I rollback, se eseguiti molto rapidamente con un impatto minimo, non dovrebbero essere considerati un fallimento ma un'esperienza di apprendimento.

Conduzione degli ORR

Una revisione della prontezza operativa (ORR) aiuta a identificare le lacune operative e procedurali. In Amazon, abbiamo creato gli ORR per sintetizzare gli insegnamenti tratti da decenni di gestione di servizi su larga scala in domande curate con linee guida sulle migliori pratiche. Un ORR raccoglie le lezioni apprese in precedenza e richiede ai nuovi team di assicurarsi di aver tenuto conto di queste lezioni nelle loro applicazioni. Gli ORR possono fornire un elenco di modalità o cause di errore che possono essere incluse nell'attività di modellazione della resilienza descritta nella sezione sulla

modellazione della resilienza riportata di seguito. Per ulteriori informazioni, vedere Operational Readiness Reviews (ORR) sul sito Web Well-Architected AWS Framework.

Comprendere i limiti di isolamento dei guasti AWS

AWS fornisce diversi limiti di isolamento dei guasti per aiutarvi a raggiungere i vostri obiettivi di resilienza. È possibile utilizzare questi limiti per sfruttare l'ambito prevedibile di contenimento dell'impatto che forniscono. È necessario conoscere il modo in cui AWS i servizi sono progettati utilizzando questi limiti, in modo da poter effettuare scelte intenzionali sulle dipendenze selezionate per l'applicazione. Per capire come utilizzare i limiti nell'applicazione, consulta la sezione AWS Fault Isolation Boundaries sul AWS sito Web.

Selezione delle risposte

Un sistema può rispondere in un'ampia gamma di modi a un allarme. Alcuni allarmi potrebbero richiedere una risposta da parte del team operativo, mentre altri potrebbero attivare meccanismi di autoriparazione all'interno dell'applicazione. Potresti decidere di mantenere le risposte che potrebbero essere automatizzate come operazioni manuali per controllare i costi dell'automazione o per gestire i vincoli tecnici. È probabile che il tipo di risposta a un allarme venga scelto in funzione del costo di implementazione della risposta, della frequenza prevista dell'allarme, della precisione dell'allarme e della potenziale perdita aziendale derivante dalla mancata risposta all'allarme.

Ad esempio, quando un processo server si blocca, il processo potrebbe essere riavviato dal sistema operativo, oppure potrebbe essere fornito un nuovo server e quello vecchio terminato, oppure a un operatore potrebbe essere richiesto di connettersi in remoto al server e riavviarlo. Queste risposte hanno lo stesso risultato, ovvero il riavvio del processo del server delle applicazioni, ma hanno diversi livelli di costi di implementazione e manutenzione.

> **Note**
>
> È possibile selezionare più risposte per adottare un approccio di resilienza approfondito. Ad esempio, nello scenario precedente il team dell'applicazione poteva scegliere di implementare tutte e tre le risposte con un intervallo di tempo tra una e l'altra. Se l'indicatore di processo del server non riuscito è ancora in stato di allarme dopo 30 secondi, il team può presumere che il sistema operativo non sia riuscito a riavviare il server delle applicazioni. Pertanto, potrebbero creare un gruppo di auto scaling per creare un nuovo server virtuale e ripristinare il processo del server delle applicazioni. Se l'indicatore è ancora in stato di

allarme dopo 300 secondi, potrebbe essere inviato un avviso al personale operativo per la connessione al server originale e il tentativo di ripristinare il processo.

La risposta scelta dal team applicativo e dall'azienda dovrebbe rispecchiare la propensione dell'azienda a compensare i costi operativi con un investimento iniziale in termini di tempo di progettazione. È necessario scegliere una risposta, un modello di architettura come la stabilità statica, uno schema software come un interruttore automatico o una procedura operativa, considerando attentamente i vincoli e la manutenzione prevista di ciascuna opzione di risposta. Potrebbero esistere alcune risposte standard per guidare i team applicativi, quindi è possibile utilizzare le librerie e i pattern gestiti dalla funzione di architettura centralizzata come input per questa considerazione.

Modellazione della resilienza

La modellazione della resilienza documenta il modo in cui un'applicazione risponderà alle diverse interruzioni previste. Anticipando le interruzioni, il team può implementare l'osservabilità, i controlli automatici e i processi di ripristino per mitigare o prevenire i danni nonostante le interruzioni. AWS ha creato linee guida per lo sviluppo di un modello di resilienza utilizzando il framework di analisi della resilienza. Questo framework può aiutarvi a prevedere le interruzioni e il loro impatto sulla vostra applicazione. Anticipando le interruzioni, è possibile identificare le mitigazioni necessarie per creare un'applicazione resiliente e affidabile. Ti consigliamo di utilizzare il framework di analisi della resilienza per aggiornare il modello di resilienza con ogni iterazione del ciclo di vita dell'applicazione. L'utilizzo di questo framework con ogni iterazione aiuta a ridurre gli incidenti anticipando le interruzioni durante la fase di progettazione e testando l'applicazione prima e dopo l'implementazione in produzione. Lo sviluppo di un modello di resilienza utilizzando questo framework consente di garantire il raggiungimento degli obiettivi di resilienza.

Fallire in modo sicuro

Se non riesci a evitare le interruzioni, fallisci in modo sicuro. Prendi in considerazione la possibilità di creare la tua applicazione con una modalità operativa fail-safe predefinita, in cui non si verifichino perdite aziendali significative. Un esempio di stato a prova di errore per un database potrebbe essere l'impostazione predefinita di operazioni di sola lettura, in cui agli utenti non è consentito creare o modificare alcun dato. A seconda della sensibilità dei dati, potreste anche volere che l'applicazione imposti per impostazione predefinita uno stato di arresto e non esegua nemmeno query di sola

lettura. Considerate quale dovrebbe essere lo stato di sicurezza dell'applicazione e utilizzate come impostazione predefinita questa modalità operativa in condizioni estreme.

Fase 3: valutazione e test

Durante la fase di valutazione e test del ciclo di vita, l'applicazione o le modifiche a un'applicazione esistente sono state progettate ma non sono ancora state rilasciate in produzione. In questa fase, si implementano attività per testare le pratiche eseguite nelle fasi precedenti e valutarne i risultati. L'applicazione potrebbe essere ancora in fase di sviluppo attivo oppure lo sviluppo primario potrebbe essere completo e l'applicazione potrebbe essere sottoposta a test prima di essere rilasciata in produzione. Durante questa fase, ci si concentra sullo sviluppo e sull'esecuzione di test che confermino o smentiscono le aspettative secondo cui l'applicazione soddisferà gli obiettivi di resilienza definiti. Inoltre, sviluppate e testate le procedure operative del sistema. Le procedure di implementazione sviluppate nella fase 2: progettazione e implementazione vengono messe in pratica e i risultati vengono valutati. Sebbene queste attività di test e valutazione inizino durante questa parte del ciclo di vita, non finiscono qui. I test e la valutazione continuano man mano che si passa alla fase 4: Operatività.

La fase di valutazione e test è suddivisa in due fasi: attività di pre-implementazione e attività post-implementazione. Le attività di pre-implementazione consistono in attività che devono essere completate prima di distribuire l'applicazione in qualsiasi ambiente, inclusa la distribuzione di nuove versioni del software e la distribuzione iniziale in un ambiente di test. Le attività successive all'implementazione avvengono dopo che il software è stato distribuito in un ambiente di test o di produzione. Le sezioni seguenti illustrano queste fasi in modo più dettagliato.

Attività di pre-implementazione

Progettazione dell'ambiente

L'ambiente in cui testate e valutate l'applicazione influisce sulla precisione con cui potete testarla e sulla fiducia che avete nel fatto che tali risultati riflettano accuratamente ciò che accadrà in produzione. Potresti essere in grado di eseguire alcuni test di integrazione localmente sui computer degli sviluppatori utilizzando servizi come Amazon DynamoDB (vedi Configurazione di DynamoDB locale nella documentazione di DynamoDB). Tuttavia, a un certo punto è necessario eseguire i test in un ambiente che replichi l'ambiente di produzione per ottenere la massima fiducia nei risultati. Questo ambiente comporterà dei costi, quindi consigliamo di adottare un approccio graduale o basato su pipeline agli ambienti, in cui ambienti simili alla produzione appaiono più avanti nella pipeline.

Test di integrazione

Il test di integrazione è il processo di verifica del corretto funzionamento di un componente ben definito di un'applicazione quando opera con dipendenze esterne. Tali dipendenze esterne potrebbero essere altri componenti sviluppati su misura, AWS servizi utilizzati per l'applicazione, dipendenze di terze parti e dipendenze locali. Questa guida si concentra sui test di integrazione che dimostrano la resilienza dell'applicazione. Si presuppone che esistano già test unitari e di integrazione che dimostrano l'accuratezza funzionale del software.

Ti consigliamo di progettare test di integrazione che testino in modo specifico i modelli di resilienza che hai implementato, come i modelli di interruttori automatici o la riduzione del carico (vedi Fase 2: Progettazione e implementazione). I test di integrazione orientati alla resilienza spesso implicano l'applicazione di un carico specifico o l'introduzione intenzionale di interruzioni nell'ambiente utilizzando funzionalità come ().AWS Fault Injection ServiceAWS FIS Idealmente, dovreste eseguire tutti i test di integrazione come parte della vostra pipeline CI/CD e assicurarvi di eseguirli ogni volta che viene eseguito il commit del codice. Ciò consente di rilevare e reagire rapidamente a eventuali modifiche al codice o alle configurazioni che comportano violazioni degli obiettivi di resilienza. Le applicazioni distribuite su larga scala sono complesse e anche piccole modifiche possono influire in modo significativo sulla resilienza di parti apparentemente non correlate dell'applicazione. Prova a eseguire i test su ogni commit. AWS fornisce un eccellente set di strumenti per il funzionamento della pipeline CI/CD e di altri strumenti. DevOps Per ulteriori informazioni, vedere Introduzione a DevOps on AWS sul sito Web. AWS

Pipeline di distribuzione automatizzate

L'implementazione e il test negli ambienti di preproduzione sono attività ripetitive e complesse che è meglio lasciare all'automazione. L'automazione di questo processo libera risorse umane e riduce la possibilità di errore. Il meccanismo per automatizzare questo processo viene spesso definito pipeline. Quando create la vostra pipeline, vi consigliamo di configurare una serie di ambienti di test che si avvicinino sempre di più alla vostra configurazione di produzione. Utilizzate questa serie di ambienti per testare ripetutamente l'applicazione. Il primo ambiente offre un set di funzionalità più limitato rispetto all'ambiente di produzione, ma comporta un costo notevolmente inferiore. Gli ambienti successivi dovrebbero aggiungere servizi e scalare per rispecchiare meglio l'ambiente di produzione.

Inizia testando nel primo ambiente. Dopo che le distribuzioni hanno superato tutti i test nel primo ambiente di test, lascia che l'applicazione venga eseguita con un certo carico per un certo periodo di tempo per vedere se si verificano problemi nel tempo. Confermate di aver configurato correttamente l'osservabilità (vedete Alarm precision più avanti in questa guida) in modo da poter rilevare

eventuali problemi. Una volta completato con successo questo periodo di osservazione, distribuisci l'applicazione nel tuo ambiente di test successivo e ripeti il processo, aggiungendo test o caricando altri test in base alle esigenze dell'ambiente. Dopo aver sufficientemente testato l'applicazione in questo modo, è possibile utilizzare i metodi di distribuzione precedentemente impostati per distribuire l'applicazione in produzione (vedere Definire le strategie CI/CD più avanti in questa guida). L'articolo Automating safe and hands-off deployments in Amazon Builders' Library è un'ottima risorsa che descrive come Amazon automatizza la distribuzione del codice. Il numero di ambienti che precedono l'implementazione di produzione varia a seconda della complessità dell'applicazione e dei tipi di dipendenze che presenta.

Test di caricamento

In apparenza, i test di carico assomigliano ai test di integrazione. Si testa una funzione discreta dell'applicazione e le relative dipendenze esterne per verificare che funzioni come previsto. I test di carico vanno quindi oltre i test di integrazione per concentrarsi sul funzionamento dell'applicazione in presenza di carichi ben definiti. Il test di carico richiede la verifica della corretta funzionalità, quindi deve essere eseguito dopo un test di integrazione riuscito. È importante capire in che modo l'applicazione risponde ai carichi previsti e come si comporta quando il carico supera le aspettative. Ciò consente di verificare di aver implementato i meccanismi necessari per garantire che l'applicazione rimanga resiliente in caso di carico estremo. Per una guida completa ai test di carico su AWS, vedete Distributed Load Testing on AWS nella AWS Solutions Library.

Attività successive all'implementazione

La resilienza è un processo continuo e la valutazione della resilienza dell'applicazione deve continuare dopo che l'applicazione è stata distribuita. I risultati delle attività post-implementazione, come le valutazioni continue della resilienza, potrebbero richiedere la rivalutazione e l'aggiornamento di alcune delle attività di resilienza eseguite in precedenza nel ciclo di vita della resilienza.

Esecuzione di valutazioni della resilienza

La valutazione della resilienza non si interrompe dopo aver distribuito l'applicazione in produzione. Anche se disponete di pipeline di distribuzione ben definite e automatizzate, a volte le modifiche possono avvenire direttamente in un ambiente di produzione. Inoltre, potrebbero esserci fattori che non avete ancora preso in considerazione nella verifica della resilienza prima dell'implementazione. AWS Resilience Hubfornisce una posizione centrale in cui è possibile valutare se l'architettura implementata soddisfa le esigenze RPO e RTO definite. È possibile utilizzare questo servizio per

eseguire valutazioni su richiesta della resilienza dell'applicazione, automatizzare le valutazioni e persino integrarle negli strumenti CI/CD, come discusso nel post del blog Continually assessment application resilience with and. AWSAWS Resilience HubAWS CodePipeline L'automazione di queste valutazioni è una best practice perché aiuta a garantire una valutazione continua del livello di resilienza in produzione.

test di ripristino di emergenza

Nella Fase 2: Progettazione e implementazione, avete sviluppato strategie di disaster recovery (DR) come parte del sistema. Durante la Fase 4, è necessario testare le procedure di DR per assicurarsi che il team sia completamente preparato per un incidente e che le procedure funzionino come previsto. È necessario testare regolarmente tutte le procedure di DR, inclusi failover e failback, ed esaminare i risultati di ogni esercizio per determinare se e come aggiornare le procedure del sistema per ottenere il miglior risultato possibile. Quando sviluppate inizialmente il test DR, programmate il test con largo anticipo e assicuratevi che l'intero team comprenda cosa aspettarsi, come verranno misurati i risultati e quale meccanismo di feedback verrà utilizzato per aggiornare le procedure in base al risultato. Dopo aver acquisito dimestichezza nell'esecuzione dei test di DR programmati, valuta la possibilità di eseguire test di DR senza preavviso. I veri disastri non si verificano secondo un programma prestabilito, quindi devi essere pronto a mettere in pratica il tuo piano in qualsiasi momento. Tuttavia, non annunciato non significa non pianificato. Le principali parti interessate devono ancora pianificare l'evento per garantire che sia predisposto un monitoraggio adeguato e che i clienti e le applicazioni critiche non subiscano ripercussioni negative.

Rilevamento delle deviazioni

Modifiche impreviste alla configurazione delle applicazioni di produzione possono verificarsi anche in presenza di automazione e procedure ben definite. Per rilevare le modifiche alla configurazione dell'applicazione, è necessario disporre di meccanismi per rilevare la deriva, che si riferisce alle deviazioni da una configurazione di base. Per scoprire come rilevare la deriva negli AWS CloudFormation stack, consulta Rilevamento delle modifiche di configurazione non gestite agli stack e alle risorse nella documentazione. AWS CloudFormation Per rilevare la deriva nell' AWS ambiente dell'applicazione, consulta Rilevare e risolvere la deriva nella documentazione. AWS Control Tower AWS Control Tower

Test sintetici

Il test sintetico è il processo di creazione di software configurabile che viene eseguito in produzione, su base pianificata, per testare le API dell'applicazione in modo da simulare l'esperienza dell'utente

finale. Questi test vengono talvolta chiamati canarini, in riferimento all'uso originario del termine nell'estrazione del carbone. I test sintetici possono spesso fornire avvisi tempestivi quando un'applicazione subisce un'interruzione, anche se la compromissione è parziale o intermittente, come spesso accade con i guasti grigi.

Progettazione del caos

L'ingegneria del caos è un processo sistematico che prevede la deliberata sottomissione di un'applicazione a eventi dirompenti in modo da ridurre il rischio, il monitoraggio attento della sua risposta e l'implementazione dei miglioramenti necessari. Il suo scopo è convalidare o contestare le ipotesi sulla capacità dell'applicazione di gestire tali interruzioni. Invece di lasciare questi eventi al caso, l'ingegneria del caos consente agli ingegneri di orchestrare gli esperimenti in un ambiente controllato, in genere durante i periodi di traffico ridotto e con un supporto tecnico prontamente disponibile per una mitigazione efficace.

L'ingegneria del caos inizia con la comprensione delle normali condizioni operative, note come stato stazionario, dell'applicazione in esame. Da lì, si formula un'ipotesi che descrive in dettaglio il comportamento corretto dell'applicazione in presenza di interruzioni. Si esegue l'esperimento, che prevede l'introduzione deliberata di interruzioni, tra cui, a titolo esemplificativo, latenza di rete, guasti del server, errori del disco rigido e compromissione delle dipendenze esterne. Si analizzano quindi i risultati dell'esperimento e si migliora la resilienza dell'applicazione sulla base delle conoscenze acquisite. L'esperimento rappresenta uno strumento prezioso per migliorare vari aspetti dell'applicazione, comprese le prestazioni, e rivela problemi latenti che altrimenti sarebbero rimasti nascosti. Inoltre, l'ingegneria del caos aiuta a rivelare le carenze nell'osservabilità e negli strumenti allarmanti e aiuta a perfezionarle. Contribuisce inoltre a ridurre i tempi di ripristino e a migliorare le capacità operative. L'ingegneria del caos accelera l'adozione delle migliori pratiche e coltiva una mentalità di miglioramento continuo. In definitiva, consente ai team di sviluppare e affinare le proprie capacità operative attraverso la pratica e la ripetizione regolari.

AWS consiglia di iniziare le attività di ingegneria del caos in un ambiente non di produzione. È possibile utilizzare AWS Fault Injection Service (AWS FIS) per eseguire esperimenti di ingegneria del caos con errori generici e errori specifici. AWS Questo servizio completamente gestito include allarmi di arresto e controlli completi delle autorizzazioni, in modo da poter adottare facilmente l'ingegneria del caos con sicurezza e sicurezza.

Fase 4: Operare

Dopo aver completato la Fase 3: valutazione e test, sei pronto per distribuire l'applicazione in produzione. Nella fase Operative, distribuisci l'applicazione in produzione e gestisci l'esperienza dei tuoi clienti. La progettazione e l'implementazione dell'applicazione determinano molti dei suoi risultati in termini di resilienza, ma questa fase si concentra sulle pratiche operative utilizzate dal sistema per mantenere e migliorare la resilienza. La creazione di una cultura dell'eccellenza operativa aiuta a creare standard e coerenza in queste pratiche.

Osservabilità

La parte più importante della comprensione dell'esperienza del cliente consiste nel monitoraggio e nell'invio di allarmi. È necessario strumentare l'applicazione per comprenderne lo stato e sono necessarie prospettive diverse, il che significa che è necessario effettuare misurazioni sia dal lato server che dal lato client, in genere con Canaries. Le metriche devono includere dati sulle interazioni dell'applicazione con le sue dipendenze e dimensioni in linea con i limiti di isolamento dei guasti. È inoltre necessario produrre registri che forniscano dettagli aggiuntivi su ogni unità di lavoro eseguita dall'applicazione. Potresti prendere in considerazione la combinazione di metriche e log utilizzando una soluzione come il formato metrico CloudWatch incorporato di Amazon. Probabilmente scoprirai che desideri sempre una maggiore osservabilità, quindi considera i compromessi in termini di costi, impegno e complessità necessari per implementare il livello di strumentazione desiderato.

I seguenti collegamenti forniscono le migliori pratiche per la strumentazione dell'applicazione e la creazione di allarmi:

- Monitoraggio dei servizi di produzione su Amazon (presentazione AWS re:Invent 2020)
- Amazon Builders' Library: eccellenza operativa in Amazon (presentazione re:Invent 2021)AWS
- Le migliori pratiche di osservabilità su Amazon (presentazioneAWS re:Invent 2022)
- Strumentazione dei sistemi distribuiti per la visibilità operativa (articolo di Amazon Builders' Library)
- Creazione di dashboard per la visibilità operativa (articolo di Amazon Builders' Library)

Gestione degli eventi

È necessario disporre di un processo di gestione degli eventi che consenta di gestire eventuali problemi quando gli allarmi (o peggio, i clienti) segnalano che qualcosa non va. Questo processo

dovrebbe includere il coinvolgimento di un operatore a chiamata, la segnalazione dei problemi e la creazione di guide per approcci coerenti alla risoluzione dei problemi che aiutino a rimuovere gli errori umani. Tuttavia, i problemi in genere non si verificano isolatamente; una singola applicazione può influire su più altre applicazioni che dipendono da essa. È possibile risolvere rapidamente i problemi comprendendo tutte le applicazioni interessate e riunendo gli operatori di più team in un'unica teleconferenza. Tuttavia, a seconda delle dimensioni e della struttura dell'organizzazione, questo processo potrebbe richiedere un team operativo centralizzato.

Oltre a impostare un processo di gestione degli eventi, è necessario rivedere regolarmente le metriche tramite i dashboard. Le revisioni periodiche ti aiutano a comprendere l'esperienza del cliente e le tendenze a lungo termine delle prestazioni della tua applicazione. Questo vi aiuta a identificare i problemi e le strozzature prima che abbiano un impatto significativo sulla produzione. La revisione delle metriche in modo coerente e standardizzato offre vantaggi significativi, ma richiede un consenso dall'alto verso il basso e un investimento di tempo.

I seguenti collegamenti forniscono le migliori pratiche per la creazione di dashboard e revisioni delle metriche operative:

- Creazione di dashboard per la visibilità operativa (articolo di Amazon Builders' Library)
- L'approccio di Amazon per fallire con successo (presentazione re:Invent 2019)AWS

Resilienza continua

Durante la Fase 2: Progettazione e implementazione e la Fase 3: Valutazione e test, sono state avviate attività di revisione e test prima di implementare l'applicazione in produzione. Durante la fase operativa, è necessario continuare a eseguire iterazioni su tali attività in produzione. È necessario rivedere periodicamente lo stato di resilienza dell'applicazione tramite le revisioni del AWS Well-Architected Framework, le Operational Readiness Review (ORR) e il framework di analisi della resilienza. Questo aiuta a garantire che l'applicazione non si discosti dalle linee di base e dagli standard stabiliti e ti tiene aggiornato con linee guida nuove o aggiornate. Queste attività di resilienza continua ti aiutano a scoprire interruzioni precedentemente impreviste e a trovare nuove mitigazioni.

Potresti anche prendere in considerazione l'idea di organizzare giornate di gioco e esperimenti di ingegneria del caos in produzione dopo averli eseguiti con successo in ambienti di preproduzione. Le giornate di gioco simulano eventi noti che avete creato meccanismi di resilienza per mitigare. Ad esempio, una giornata di gioco potrebbe simulare un'interruzione del servizio AWS regionale e implementare un failover multiregionale. Sebbene l'implementazione di queste attività possa

richiedere un notevole livello di impegno, entrambe le pratiche aiutano a rafforzare la fiducia nella resilienza del sistema alle modalità di errore per cui è stato progettato.

Utilizzando le applicazioni, riscontrando eventi operativi, esaminando le metriche e testando l'applicazione, incontrerete numerose opportunità di risposta e apprendimento.

Fase 5: Rispondi e impara

Il modo in cui l'applicazione risponde agli eventi dirompenti ne influenza l'affidabilità. Imparare dall'esperienza e sapere come l'applicazione ha risposto alle interruzioni del passato è fondamentale anche per migliorarne l'affidabilità.

La fase di risposta e apprendimento si concentra sulle pratiche che è possibile implementare per rispondere meglio agli eventi dirompenti nelle applicazioni. Include anche pratiche per aiutarvi a trarre il massimo apprendimento dalle esperienze dei vostri team operativi e ingegneri.

Creazione di report di analisi degli incidenti

Quando si verifica un incidente, la prima azione consiste nel prevenire ulteriori danni ai clienti e all'azienda il più rapidamente possibile. Dopo il ripristino dell'applicazione, il passaggio successivo consiste nel capire cosa è successo e identificare le misure per evitare che si ripeta. Questa analisi post-incidente viene in genere acquisita come report che documenta la serie di eventi che hanno portato al danneggiamento dell'applicazione e gli effetti dell'interruzione sull'applicazione, sui clienti e sull'azienda. Tali report diventano preziosi artefatti di apprendimento e devono essere ampiamente condivisi in tutta l'azienda.

> **Note**
>
> È fondamentale eseguire l'analisi degli incidenti senza attribuire alcuna colpa. Supponiamo che tutti gli operatori abbiano adottato la linea d'azione migliore e più appropriata alla luce delle informazioni in loro possesso. Non utilizzate i nomi degli operatori o degli ingegneri in un rapporto. Se si adduce l'errore umano come motivo di danneggiamento, si potrebbe far sì che i membri del team vengano messi in guardia per proteggersi, con conseguente acquisizione di informazioni errate o incomplete.

Un buon rapporto di analisi degli incidenti, come quello documentato nel processo Amazon Correction of Error (COE), segue un formato standardizzato e cerca di acquisire, nel modo più dettagliato possibile, le condizioni che hanno portato a un danneggiamento dell'applicazione. Il rapporto descrive in dettaglio una serie di eventi con data e ora e acquisisce dati quantitativi (spesso metriche e schermate da dashboard di monitoraggio) che descrivono lo stato misurabile dell'applicazione nel tempo. Il rapporto dovrebbe raccogliere i processi di pensiero degli operatori e degli ingegneri che hanno agito e le informazioni che li hanno portati alle conclusioni. Il rapporto

dovrebbe inoltre descrivere in dettaglio le prestazioni dei diversi indicatori, ad esempio quali allarmi sono stati generati, se tali allarmi riflettono accuratamente lo stato dell'applicazione, l'intervallo di tempo tra gli eventi e gli allarmi risultanti e il tempo necessario per risolvere l'incidente. La sequenza temporale riporta anche i runbook o le automazioni avviate e il modo in cui hanno aiutato l'applicazione a riguadagnare uno stato utile. Questi elementi della sequenza temporale aiutano il team a comprendere l'efficacia delle risposte automatizzate e degli operatori, compresa la rapidità con cui hanno risolto il problema e l'efficacia nel mitigare l'interruzione.

Questo quadro dettagliato di un evento storico è un potente strumento educativo. I team devono archiviare questi report in un archivio centrale a disposizione di tutta l'azienda, in modo che altri possano esaminare gli eventi e imparare da essi. Ciò può migliorare l'intuizione dei team su ciò che può andare storto nella produzione.

Un archivio di report dettagliati sugli incidenti diventa anche una fonte di materiale di formazione per gli operatori. I team possono utilizzare una segnalazione sugli incidenti per ispirare una giornata di gioco da tavolo o dal vivo, in cui alle squadre vengono fornite informazioni che riproducono la sequenza temporale riportata nel rapporto. Gli operatori possono esaminare lo scenario con informazioni parziali tratte dalla sequenza temporale e descrivere le azioni che intraprenderebbero. Il moderatore della giornata di gioco può quindi fornire indicazioni su come l'applicazione ha risposto in base alle azioni dell'operatore. Ciò sviluppa le capacità di risoluzione dei problemi degli operatori, in modo che possano anticipare e risolvere più facilmente i problemi.

Un team centralizzato responsabile dell'affidabilità delle applicazioni dovrebbe conservare questi report in una libreria centralizzata a cui l'intera organizzazione può accedere. Questo team dovrebbe anche essere responsabile della manutenzione del modello di rapporto e della formazione dei team su come completare il rapporto di analisi degli incidenti. Il team addetto all'affidabilità dovrebbe esaminare periodicamente i report per individuare le tendenze aziendali che possono essere affrontate mediante librerie software, modelli di architettura o modifiche ai processi del team.

Esecuzione di revisioni operative

Come discusso in Stage 4: Operate, le revisioni operative sono un'opportunità per esaminare le recenti versioni di funzionalità, gli incidenti e le metriche operative. La revisione operativa è anche un'opportunità per condividere gli insegnamenti tratti dalle versioni di funzionalità e dagli incidenti con la più ampia comunità di tecnici dell'organizzazione. Durante la revisione operativa, i team esaminano le implementazioni di funzionalità che sono state annullate, gli incidenti che si sono verificati e il modo in cui sono stati gestiti. Ciò offre agli ingegneri di tutta l'organizzazione l'opportunità di imparare dalle esperienze altrui e di porre domande.

Rivolgete le vostre revisioni operative alla comunità ingegneristica della vostra azienda in modo che possa saperne di più sulle applicazioni IT che gestiscono l'azienda e sui tipi di problemi che possono riscontrare. Porteranno con sé queste conoscenze durante la progettazione, l'implementazione e la distribuzione di altre applicazioni aziendali.

Analisi delle prestazioni degli allarmi

Gli allarmi, come illustrato nella fase operativa, possono comportare avvisi sulla dashboard, la creazione di ticket, l'invio di e-mail o la chiamata degli operatori. Un'applicazione avrà numerosi allarmi configurati per monitorare vari aspetti del suo funzionamento. Nel tempo, l'accuratezza e l'efficacia di questi allarmi dovrebbero essere riviste per aumentare la precisione degli allarmi, ridurre i falsi positivi e consolidare gli avvisi duplicati.

Precisione dell'allarme

Gli allarmi devono essere quanto più specifici possibile per ridurre il tempo da dedicare all'interpretazione o alla diagnosi dell'interruzione specifica che ha causato l'allarme. Quando viene generato un allarme in risposta a un problema dell'applicazione, gli operatori che ricevono e rispondono all'allarme devono prima interpretare le informazioni trasmesse dall'allarme. Le informazioni possono essere un semplice codice di errore che corrisponde a una linea di azione, ad esempio una procedura di ripristino, oppure possono includere righe dei registri delle applicazioni che è necessario esaminare per capire il motivo per cui è stato generato l'allarme. Man mano che il team impara a utilizzare un'applicazione in modo più efficace, dovrebbe perfezionare questi allarmi per renderli il più chiari e concisi possibile.

Non è possibile prevedere tutte le possibili interruzioni di un'applicazione, quindi ci saranno sempre allarmi generici che richiedono l'analisi e la diagnosi da parte dell'operatore. Il team dovrebbe lavorare per ridurre il numero di allarmi generali al fine di migliorare i tempi di risposta e ridurre il tempo medio di riparazione (MTTR). Idealmente, dovrebbe esserci una one-to-one relazione tra un allarme e una risposta automatica o eseguita dall'uomo.

Falsi positivi

Gli allarmi che non richiedono alcuna azione da parte degli operatori ma generano avvisi come e-mail, pagine o ticket verranno ignorati dagli operatori nel tempo. Periodicamente, o come parte di un'analisi degli incidenti, esamina gli allarmi per identificare quelli che spesso vengono ignorati o che non richiedono alcun intervento da parte degli operatori (falsi positivi). È necessario adoperarsi per rimuovere l'allarme o migliorarlo in modo che emetta un avviso utilizzabile per gli operatori.

Falsi negativi

Durante un incidente, gli allarmi configurati per avvisare durante l'incidente potrebbero fallire, forse a causa di un evento che influisce sull'applicazione in modo imprevisto. Nell'ambito di un'analisi degli incidenti, è necessario esaminare gli allarmi che avrebbero dovuto essere generati ma non lo sono stati. È necessario lavorare per migliorare questi allarmi in modo che riflettano meglio le condizioni che potrebbero derivare da un evento. In alternativa, potresti dover creare allarmi aggiuntivi che si riferiscano alla stessa interruzione ma che vengano generati da un sintomo diverso dell'interruzione.

Avvisi duplicati

Un'interruzione che danneggia l'applicazione può causare diversi sintomi e generare più allarmi. Periodicamente, o come parte di un'analisi degli incidenti, è necessario esaminare gli allarmi e gli avvisi emessi. Se gli operatori hanno ricevuto avvisi duplicati, create allarmi aggregati per consolidarli in un unico messaggio di avviso.

Esecuzione di revisioni delle metriche

Il team deve raccogliere le metriche operative relative all'applicazione, ad esempio il numero di incidenti per gravità al mese, il tempo impiegato per rilevare l'incidente, il tempo necessario per identificare la causa, il tempo necessario per porvi rimedio e il numero di ticket creati, avvisi inviati e pagine pubblicate. Rivedi queste metriche almeno una volta al mese per comprendere l'onere che grava sul personale operativo, il signal-to-noise rapporto con cui si occupa (ad esempio, avvisi informativi e quelli utilizzabili) e se il team sta migliorando la sua capacità di utilizzare le applicazioni sotto il suo controllo. Utilizza questa recensione per comprendere le tendenze relative agli aspetti misurabili del team operativo. Chiedi al team idee su come migliorare queste metriche.

Fornire formazione e abilitazione

È difficile acquisire una descrizione dettagliata di un'applicazione e del relativo ambiente che ha provocato un incidente o un comportamento imprevisto. Inoltre, modellare la resilienza dell'applicazione per anticipare tali scenari non è sempre semplice. L'organizzazione dovrebbe investire in materiali di formazione e abilitazione per consentire ai team operativi e agli sviluppatori di partecipare ad attività come la modellazione della resilienza, l'analisi degli incidenti, le giornate di gioco e gli esperimenti di ingegneria del caos. Ciò migliorerà la fedeltà dei report prodotti dai team e le conoscenze acquisite. I team saranno inoltre meglio attrezzati per anticipare i guasti senza affidarsi

a un gruppo di ingegneri più ristretto ed esperto, che dovrà fornire le proprie opinioni attraverso revisioni programmate.

Creazione di una base di conoscenze sugli incidenti

Un rapporto di incidente è un output standard di un'analisi degli incidenti. È consigliabile utilizzare lo stesso rapporto o uno simile per documentare gli scenari in cui è stato rilevato un comportamento anomalo dell'applicazione, anche se l'applicazione non è stata danneggiata. Utilizzate la stessa struttura di report standardizzata per registrare i risultati di caos, esperimenti e giornate di gioco. Il report rappresenta un'istantanea dell'applicazione e del relativo ambiente che hanno provocato un incidente o un comportamento altrimenti imprevisto. È necessario archiviare questi report standardizzati in un archivio centrale accessibile a tutti i tecnici dell'azienda.

I team operativi e gli sviluppatori possono quindi consultare questa knowledge base per capire cosa ha causato l'interruzione delle applicazioni in passato, quali tipi di scenari avrebbero potuto causare interruzioni e cosa ha impedito il danneggiamento delle applicazioni. Questa knowledge base diventa un acceleratore per migliorare le competenze dei team operativi e degli sviluppatori e consente loro di condividere le proprie conoscenze ed esperienze. Inoltre, puoi utilizzare i report come materiale di formazione o come scenari per giornate di gioco o esperimenti sul caos per migliorare l'intuizione e la capacità del team operativo di risolvere le interruzioni.

> **Note**
>
> Un formato di report standardizzato fornisce inoltre ai lettori un senso di familiarità e li aiuta a trovare più rapidamente le informazioni che stanno cercando.

Implementazione approfondita della resilienza

Come discusso in precedenza, un'organizzazione avanzata implementerà più risposte a un allarme. Non vi è alcuna garanzia che una risposta sia efficace, quindi stratificando le risposte un'applicazione sarà meglio equipaggiata per fallire senza problemi. Si consiglia di implementare almeno due risposte per ogni indicatore per garantire che una singola risposta non diventi un singolo punto di errore che potrebbe portare a uno scenario di DR. Questi livelli devono essere creati in ordine seriale, in modo che venga eseguita una risposta successiva solo se la risposta precedente era inefficace. Non dovresti eseguire risposte a più livelli a un singolo allarme. Utilizza invece un allarme che indichi se una risposta non ha avuto successo e, in tal caso, avvia la risposta successiva a più livelli.

Conclusioni e risorse

Questa guida presenta un ciclo di vita che consente di migliorare continuamente la resilienza delle applicazioni implementando le migliori pratiche in cinque fasi: definizione degli obiettivi, progettazione e implementazione, valutazione e test, utilizzo, risposta e apprendimento.

Per ulteriori informazioni sui servizi e sui concetti discussi in questa guida, consulta le seguenti risorse.

AWS servizi:

- AWS Backup
- AWS Elastic Disaster Recovery
- AWS Fault Injection Service (AWS FIS)
- AWS Resilience Hub
- Controller di ripristino delle applicazioni Amazon (ARC)
- AWS X-Ray

Post e articoli del blog:

- Disponibilità e oltre: comprensione e miglioramento della resilienza dei sistemi distribuiti su AWS
- AWS Limiti di isolamento dei guasti
- AWS Nozioni di base su più regioni
- Ingegneria del caos nel cloud
- Valutazione continua della resilienza delle applicazioni con e AWS Resilience HubAWS CodePipeline
- Ripristino di emergenza delle applicazioni locali per AWS
- Pilastro dell'affidabilità: AWS Well-Architected Framework
- Quadro di analisi della resilienza

Collaboratori

I collaboratori di questa guida includono:

- Bruno Emer, architetto principale delle soluzioni, AWS
- Clark Richey, architetto principale delle soluzioni, AWS
- Elaine Harvey, direttore generale, servizi di affidabilità, AWS
- Jason Barto, architetto principale delle soluzioni, AWS
- John Formento, architetto principale delle soluzioni, AWS
- Lisi Lewis, responsabile marketing di prodotto senior, AWS
- Michael Haken, architetto principale delle soluzioni, AWS
- Neeraj Kumar, architetto principale delle soluzioni, AWS
- Wangechi Doble, architetto principale delle soluzioni, AWS

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	—	6 ottobre 2023

AWS Glossario delle linee guida prescrittive

I seguenti sono termini comunemente usati 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 di 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 aggregata

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 viene utilizzato AIOps nella strategia di migrazione AWS , 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 for 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 disturbare 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.

distribuzioni blu/verdi

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 AWS Cloud Adoption 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.

CoE

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 del 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 i post di CCoE 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 dimensionare l'adozione del cloud (ad esempio, creazione di una zona di destinazione, definizione di un CCoE, 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 del 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 o AWS CodeCommit. 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, AWS Panorama offre dispositivi che aggiungono CV alle reti di telecamere locali e Amazon SageMaker 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, creazione, test, gestione temporanea e produzione del processo di rilascio del software. Il processo CI/CD è comunemente descritto come una pipeline. CI/CD può aiutare ad automatizzare i processi, migliorare la produttività, migliorare

la qualità del codice e velocizzare le distribuzioni. 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.

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.

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 pipeline CI/CD, l'ambiente di produzione è l'ultimo ambiente di implementazione.
- 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 la 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, vedere 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.

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.

G

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.

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 livello elevato che consente di governare risorse, policy e conformità tra le unità organizzative (OU). 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 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.

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, ad esempio 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

Considera 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. Nel documento Architettura di riferimento per la sicurezza di AWS si consiglia di configurare l'account di rete con VPC in entrata, in uscita e di ispezione 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.

Internet delle cose industriale (IIoT)

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, consulta Creazione di una strategia di trasformazione digitale dell'Internet delle cose industriale (IIoT).

VPC di ispezione

In un'architettura AWS multi-account, un VPC centralizzato che gestisce le ispezioni del traffico di rete tra VPC (uguali o diversi Regioni AWS), Internet e reti locali. Nel documento Architettura di riferimento per la sicurezza di AWS si consiglia di configurare l'account di rete con VPC in entrata, in uscita e di ispezione 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, consulta la sezione Interpretabilità dei modelli 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.

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.

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 piccolo servizio indipendente che comunica tramite API ben definite ed è in genere di proprietà di piccoli team autonomi. 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 tramite un'interfaccia ben definita utilizzando API leggere. 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 includono in genere 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.

O

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. Nel documento Architettura di riferimento per la sicurezza di AWS si consiglia di configurare l'account di rete con VPC in entrata, in uscita e di ispezione 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. Per ulteriori informazioni, consulta la sezione Abilitazione della persistenza dei dati nei microservizi.

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 all'ingegneria dei sistemi che tiene conto della privacy durante l'intero processo di progettazione.

zone ospitate private

Un container che contiene informazioni su come si desidera che Amazon Route 53 risponda alle query DNS per un dominio e i relativi sottodomini all'interno di uno o più VPC. 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.

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.

pubblica/sottoscrivi (pub/sub)

Un pattern 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).

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. Ciò determina quella che viene considerata 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.

Regione

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 o ripristinare le interruzioni. 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.

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 l'obiettivo del punto di ripristino.

RTO

Vedi l'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 AWS Management Console 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 AWS Secrets Manager, 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.

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 policy che fornisce il controllo centralizzato sulle autorizzazioni per tutti gli account di un'organizzazione in AWS Organizations. Le SCP definiscono i guardrail o fissano i limiti alle azioni che un amministratore può delegare a utenti o ruoli. Puoi utilizzare le SCP 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.

T

tags

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 è possibile utilizzare per collegare i VPC e le reti on-premise. 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 VPC che consente di instradare il traffico tramite 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 interrogazioni 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.

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.