Le migliori pratiche per l'utilizzo di Terraform Provider AWS

AWS Guida prescrittiva

AWS Guida prescrittiva: Le migliori pratiche per l'utilizzo di Terraform Provider AWS

Table of Contents

Le migliori pratiche per l'utilizzo del provider AWS Terraform

Michael Begin, DevOps consulente senior, Amazon Web Services ()AWS

Maggio 2024 (cronologia dei documenti)

La gestione dell'infrastruttura come codice (IaC) con Terraform on AWS offre importanti vantaggi come una maggiore coerenza, sicurezza e agilità. Tuttavia, man mano che la configurazione di Terraform cresce in termini di dimensioni e complessità, diventa fondamentale seguire le migliori pratiche per evitare insidie.

Questa guida fornisce le migliori pratiche consigliate per l'utilizzo di Terraform AWS Provider di. HashiCorp Ti guida attraverso il corretto controllo delle versioni, i controlli di sicurezza, i backend remoti, la struttura di codebase e i provider di community su cui ottimizzare Terraform. AWS Ogni sezione fornisce maggiori dettagli sulle specifiche dell'applicazione di queste migliori pratiche:

- Sicurezza
- Backend
- Struttura e organizzazione della base del codice
- AWS Gestione delle versioni del provider
- Moduli comunitari

Obiettivi

Questa guida ti aiuta ad acquisire conoscenze operative su Terraform AWS Provider e affronta i seguenti obiettivi aziendali che puoi raggiungere seguendo le migliori pratiche IaC in materia di sicurezza, affidabilità, conformità e produttività degli sviluppatori.

- Migliora la qualità e la coerenza del codice dell'infrastruttura tra i progetti Terraform.
- Accelera l'onboarding degli sviluppatori e la capacità di contribuire al codice dell'infrastruttura.
- Aumenta l'agilità aziendale attraverso modifiche più rapide all'infrastruttura.
- Riduci gli errori e i tempi di inattività legati alle modifiche dell'infrastruttura.
- Ottimizza i costi dell'infrastruttura seguendo le migliori pratiche IaC.
- Rafforza il tuo livello di sicurezza generale attraverso l'implementazione delle migliori pratiche.

Destinatari

Il pubblico di destinazione di questa guida include i responsabili tecnici e i manager che supervisionano i team che utilizzano Terraform for IaC su. AWS Altri potenziali lettori includono ingegneri dell'infrastruttura, DevOps ingegneri, architetti di soluzioni e sviluppatori che utilizzano attivamente Terraform per gestire l'infrastruttura. AWS

Seguire queste best practice farà risparmiare tempo e aiuterà a sbloccare i vantaggi di IaC per questi ruoli.

Panoramica

I provider Terraform sono plugin che consentono a Terraform di interagire con diverse API. Terraform AWS Provider è il plug-in ufficiale per la gestione dell' AWS infrastruttura come codice (IaC) con Terraform. Traduce la sintassi Terraform in chiamate AWS API per creare, leggere, aggiornare ed eliminare risorse. AWS

Il AWS Provider gestisce l'autenticazione, traduce la sintassi Terraform in chiamate AWS API e fornisce risorse in. AWS Si utilizza un blocco di `provider` codice Terraform per configurare il plug-in del provider utilizzato da Terraform per interagire con l'API. AWS Puoi configurare più blocchi AWS Provider per gestire le risorse in diverse Account AWS regioni.

Ecco un esempio di configurazione Terraform che utilizza più blocchi AWS Provider con alias per gestire un database Amazon Relational Database Service (Amazon RDS) con una replica in una regione e un account diversi. I provider primari e secondari assumono ruoli (IAM) diversi AWS Identity and Access Management :

```
# Configure the primary AWS Provider
provider "aws" {
  region = "us-west-1"
  alias  = "primary"
}

# Configure a secondary AWS Provider for the replica Region and account
provider "aws" {
  region      = "us-east-1"
  alias       = "replica"
  assume_role {
    role_arn     = "arn:aws:iam::<replica-account-id>:role/<role-name>"
    session_name = "terraform-session"
  }
}

# Primary Amazon RDS database
resource "aws_db_instance" "primary" {
  provider = aws.primary

  # ... RDS instance configuration
}

# Read replica in a different Region and account
```

```
resource "aws_db_instance" "read_replica" {
  provider = aws.replica

  # ... RDS read replica configuration
  replicate_source_db = aws_db_instance.primary.id
}
```

In questo esempio:

- Il primo `provider` blocco configura il AWS provider principale nella `us-west-1` regione con l'alias`primary`.
- Il secondo `provider` blocco configura un AWS provider secondario nella `us-east-1` regione con l'alias. `replica` Questo provider viene utilizzato per creare una replica di lettura del database primario in una regione e in un account diversi. Il `assume_role` blocco viene utilizzato per assumere un ruolo IAM nell'account di replica. `role_arn`specifica l'Amazon Resource Name (ARN) del ruolo IAM da assumere `session_name` ed è un identificatore univoco per la sessione Terraform.
- La `aws_db_instance.primary` risorsa crea il database Amazon RDS primario utilizzando il `primary` provider della `us-west-1` regione.
- La `aws_db_instance.read_replica` risorsa crea una replica di lettura del database primario nella `us-east-1` regione utilizzando il `replica` provider. L'`replicate_source_db`attributo fa riferimento all'ID del `primary` database.

Best practice di sicurezza

La corretta gestione dell'autenticazione, dei controlli di accesso e della sicurezza è fondamentale per un utilizzo sicuro di Terraform Provider. AWS Questa sezione descrive le migliori pratiche in materia di:

- Ruoli e autorizzazioni IAM per l'accesso con privilegi minimi
- Protezione delle credenziali per impedire l'accesso non autorizzato ad account e risorse AWS
- Crittografia dello stato remoto per proteggere i dati sensibili
- Scansione dell'infrastruttura e del codice sorgente per identificare configurazioni errate
- Controlli di accesso per l'archiviazione remota dello stato
- Applicazione delle policy di Sentinel per implementare le barriere di governance

Seguire queste best practice aiuta a rafforzare il livello di sicurezza quando si utilizza Terraform per gestire l'infrastruttura. AWS

Segui il principio del privilegio minimo

Il privilegio minimo è un principio di sicurezza fondamentale che si riferisce alla concessione solo delle autorizzazioni minime richieste a un utente, processo o sistema per eseguire le funzioni previste. È un concetto fondamentale nel controllo degli accessi e una misura preventiva contro l'accesso non autorizzato e le potenziali violazioni dei dati.

Il principio del privilegio minimo viene enfatizzato più volte in questa sezione perché si riferisce direttamente al modo in cui Terraform autentica ed esegue azioni contro provider di cloud come. AWS

Quando si utilizza Terraform per fornire e gestire AWS le risorse, agisce per conto di un'entità (utente o ruolo) che richiede le autorizzazioni appropriate per effettuare chiamate API. Non seguire il privilegio minimo comporta gravi rischi per la sicurezza:

- Se Terraform dispone di autorizzazioni eccessive oltre a quelle necessarie, un'errata configurazione involontaria potrebbe apportare modifiche o eliminazioni indesiderate.
- Le concessioni di accesso eccessivamente permissive aumentano l'ambito di impatto se i file di stato o le credenziali di Terraform vengono compromessi.

- Il mancato rispetto del privilegio minimo contrasta con le migliori pratiche di sicurezza e i requisiti di conformità normativa per la concessione dell'accesso minimo richiesto.

Uso di ruoli IAM

Utilizza i ruoli IAM anziché gli utenti IAM, ove possibile, per migliorare la sicurezza con AWS Terraform Provider. I ruoli IAM forniscono credenziali di sicurezza temporanee che ruotano automaticamente, eliminando la necessità di gestire le chiavi di accesso a lungo termine. I ruoli offrono anche controlli di accesso precisi tramite le policy IAM.

Concedi l'accesso con il minimo privilegio utilizzando le policy IAM

Costruisci con cura le policy IAM per garantire che i ruoli e gli utenti dispongano solo del set minimo di autorizzazioni richiesto per il loro carico di lavoro. Inizia con una policy vuota e aggiungi in modo iterativo i servizi e le azioni consentiti. Per eseguire questa operazione:

- Abilita IAM Access Analyzer per valutare le policy ed evidenziare le autorizzazioni inutilizzate che possono essere rimosse.
- Esamina manualmente le policy per rimuovere tutte le funzionalità che non sono essenziali per la responsabilità prevista del ruolo.
- Utilizza le variabili e i tag delle policy IAM per semplificare la gestione delle autorizzazioni.

Policy ben strutturate garantiscono l'accesso sufficiente per adempiere alle responsabilità del carico di lavoro e nient'altro. Definisci le azioni a livello operativo e consenti le chiamate solo alle API richieste su risorse specifiche.

Seguire questa best practice riduce la portata dell'impatto e segue i principi di sicurezza fondamentali della separazione dei compiti e dell'accesso con privilegi minimi. Inizia l'accesso aperto e rigoroso gradualmente, se necessario, invece di iniziare con open access e provare a limitare l'accesso in un secondo momento.

Assumi ruoli IAM per l'autenticazione locale

Quando esegui Terraform localmente, evita di configurare chiavi di accesso statiche. Utilizza invece i ruoli IAM per concedere temporaneamente l'accesso privilegiato senza esporre credenziali a lungo termine.

Innanzitutto, crea un ruolo IAM con le autorizzazioni minime necessarie e aggiungi una relazione di fiducia che consenta di assumere il ruolo IAM dal tuo account utente o dall'identità federata. Ciò autorizza l'uso temporaneo del ruolo.

Esempio di politica di relazione fiduciaria:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/terraform-execution"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Quindi, esegui il AWS CLI comando aws sts assume-role per recuperare credenziali di breve durata per il ruolo. Queste credenziali sono generalmente valide per un'ora.

AWS CLI esempio di comando:

```
aws sts assume-role --role-arn arn:aws:iam::111122223333:role/terraform-execution --
role-session-name terraform-session-example
```

L'output del comando contiene una chiave di accesso, una chiave segreta e un token di sessione che è possibile utilizzare per l'autenticazione per AWS:

```
{
    "AssumedRoleUser": {
        "AssumedRoleId": "AROA3XFRBF535PLBIFPI4:terraform-session-example",
        "Arn": "arn:aws:sts::111122223333:assumed-role/terraform-execution/terraform-
session-example"
    },
    "Credentials": {
        "SecretAccessKey": " wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "SessionToken": " AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwqnKwRcOIfrRh3c/LTo6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJOgQs8IZZaIv2BXIa2R4OlgkBN9bkUDNCJiBeb/
AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wnOE49mRl/+OtkIKGO7fAE",
```

```
        "Expiration": "2024-03-15T00:05:07Z",
        "AccessKeyId": "ASIAIOSFODNN7EXAMPLE"
    }
}
```

Il AWS Provider può anche gestire automaticamente l'assunzione del ruolo.

Esempio di configurazione del provider per l'assunzione di un ruolo IAM:

```
provider "aws" {
  assume_role {
    role_arn     = "arn:aws:iam::111122223333:role/terraform-execution"
    session_name = "terraform-session-example"
  }
}
```

Ciò garantisce privilegi elevati esclusivamente per la durata della sessione Terraform. Le chiavi temporanee non possono essere divulgate perché scadono automaticamente dopo la durata massima della sessione.

I vantaggi principali di questa best practice includono una maggiore sicurezza rispetto alle chiavi di accesso di lunga durata, controlli di accesso dettagliati sul ruolo per i privilegi minimi e la possibilità di revocare facilmente l'accesso modificando le autorizzazioni del ruolo. Utilizzando i ruoli IAM, eviti anche di dover archiviare direttamente i segreti localmente negli script o su disco, il che ti aiuta a condividere la configurazione Terraform in modo sicuro tra un team.

Usa i ruoli IAM per l'autenticazione Amazon EC2

Quando esegui Terraform da istanze Amazon Elastic Compute Cloud (Amazon EC2), evita di archiviare le credenziali a lungo termine a livello locale. Utilizza invece i ruoli IAM e i profili di istanza per concedere automaticamente le autorizzazioni con privilegi minimi.

Innanzitutto, crea un ruolo IAM con le autorizzazioni minime e assegna il ruolo al profilo dell'istanza. Il profilo di istanza consente alle istanze EC2 di ereditare le autorizzazioni definite nel ruolo. Quindi, avvia le istanze specificando il profilo dell'istanza. L'istanza verrà autenticata tramite il ruolo allegato.

Prima di eseguire qualsiasi operazione Terraform, verifica che il ruolo sia presente nei metadati dell'istanza per confermare che le credenziali siano state ereditate con successo.

```
TOKEN=$(curl -s -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600")
```

```
curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/meta-data/
iam/security-credentials/
```

Questo approccio evita l'inserimento di AWS chiavi permanenti negli script o nella configurazione Terraform all'interno dell'istanza. Le credenziali temporanee sono rese disponibili a Terraform in modo trasparente tramite il ruolo e il profilo dell'istanza.

I principali vantaggi di questa best practice includono una maggiore sicurezza rispetto alle credenziali a lungo termine, un sovraccarico di gestione delle credenziali ridotto e la coerenza tra ambienti di sviluppo, test e produzione. L'autenticazione dei ruoli IAM semplifica le esecuzioni di Terraform dalle istanze EC2, rafforzando al contempo l'accesso con privilegi minimi.

Utilizza credenziali dinamiche per gli spazi di lavoro HCP Terraform

HCP Terraform è un servizio gestito fornito da HashiCorp che aiuta i team a utilizzare Terraform per fornire e gestire l'infrastruttura su più progetti e ambienti. Quando esegui Terraform in HCP Terraform, utilizza credenziali dinamiche per semplificare e proteggere l'autenticazione. AWS Terraform scambia automaticamente le credenziali temporanee ad ogni esecuzione senza la necessità di assumere ruoli IAM.

I vantaggi includono una rotazione segreta più semplice, una gestione centralizzata delle credenziali tra le aree di lavoro, le autorizzazioni con privilegi minimi e l'eliminazione delle chiavi codificate. Affidarsi a chiavi temporanee con hash migliora la sicurezza rispetto alle chiavi di accesso di lunga durata.

Usa i ruoli IAM in AWS CodeBuild

In AWS CodeBuild, esegui le tue build utilizzando un ruolo IAM assegnato al CodeBuild progetto. Ciò consente a ogni build di ereditare automaticamente le credenziali temporanee dal ruolo invece di utilizzare chiavi a lungo termine.

Esegui GitHub azioni in remoto su HCP Terraform

Configura i flussi di lavoro GitHub Actions per eseguire Terraform in remoto sulle aree di lavoro HCP Terraform. Affidati a credenziali dinamiche e al blocco remoto dello stato anziché alla gestione dei segreti. GitHub

Usa GitHub Actions with OIDC e configura l'azione Credenziali AWS

Utilizza lo standard OpenID Connect (OIDC) per federare GitHub l'identità Actions tramite IAM. Utilizza l'azione Configura AWS credenziali per scambiare il GitHub token con AWS credenziali temporanee senza bisogno di chiavi di accesso a lungo termine.

Da utilizzare GitLab con OIDC e AWS CLI

Utilizza lo standard OIDC per federare le GitLab identità tramite IAM per l'accesso temporaneo. Affidandoti a OIDC, eviti di dover gestire direttamente le chiavi di accesso a lungo termine all'interno. AWS GitLab Le credenziali vengono just-in-time scambiate, il che migliora la sicurezza. Gli utenti ottengono inoltre l'accesso con il privilegio minimo in base alle autorizzazioni nel ruolo IAM.

Utilizza utenti IAM unici con strumenti di automazione legacy

Se disponi di strumenti e script di automazione che non dispongono del supporto nativo per l'utilizzo dei ruoli IAM, puoi creare singoli utenti IAM per concedere l'accesso programmatico. Il principio del privilegio minimo è ancora valido. Riduci al minimo le autorizzazioni relative alle policy e affidati a ruoli separati per ogni pipeline o script. Man mano che passi a strumenti o strumenti più moderni, inizia a supportare i ruoli in modo nativo e passa gradualmente a tali ruoli.

> **Warning**
>
> Gli utenti IAM dispongono di credenziali a lungo termine, il che rappresenta un rischio per la sicurezza. Per contribuire a mitigare questo rischio, ti consigliamo di fornire a questi utenti solo le autorizzazioni necessarie per eseguire l'attività e di rimuoverli quando non sono più necessari.

Usa il plugin Jenkins AWS Credentials

Usa il plug-in AWS Credentials di Jenkins per configurare centralmente e inserire le credenziali nelle build in AWS modo dinamico. In questo modo si evita di inserire i segreti nel controllo del codice sorgente.

Monitora, convalida e ottimizza continuamente il privilegio minimo

Nel tempo, potrebbero essere concesse autorizzazioni aggiuntive che possono superare le politiche minime richieste. Analizza continuamente l'accesso per identificare e rimuovere eventuali autorizzazioni non necessarie.

Monitora continuamente l'utilizzo delle chiavi di accesso

Se non puoi evitare di utilizzare le chiavi di accesso, utilizza i report sulle credenziali IAM per trovare le chiavi di accesso non utilizzate più vecchie di 90 giorni e revoca le chiavi inattive sia negli account utente che nei ruoli macchina. Avvisa gli amministratori affinché confermino manualmente la rimozione delle chiavi per i dipendenti e i sistemi attivi.

Il monitoraggio dell'utilizzo delle chiavi consente di ottimizzare le autorizzazioni poiché è possibile identificare e rimuovere i permessi non utilizzati. Quando si segue questa best practice con la rotazione delle chiavi di accesso, si limita la durata delle credenziali e si applica l'accesso con privilegi minimi.

AWS offre diversi servizi e funzionalità che è possibile utilizzare per configurare avvisi e notifiche per gli amministratori. Ecco alcune opzioni:

- AWS Config: È possibile utilizzare AWS Config le regole per valutare le impostazioni di configurazione delle AWS risorse, incluse le chiavi di accesso IAM. Puoi creare regole personalizzate per verificare condizioni specifiche, ad esempio chiavi di accesso non utilizzate che risalgono a un numero specifico di giorni. Quando una regola viene violata, AWS Config può avviare una valutazione per la correzione o inviare notifiche a un argomento di Amazon Simple Notification Service (Amazon SNS).
- AWS Security Hub: Security Hub offre una visione completa del livello di sicurezza del tuo AWS account e può aiutarti a rilevare e notificare potenziali problemi di sicurezza, incluse le chiavi di accesso IAM inutilizzate o inattive. Security Hub può integrarsi con Amazon EventBridge e Amazon SNS o inviare notifiche AWS Chatbot agli amministratori.
- AWS Lambda: le funzioni Lambda possono essere richiamate da vari eventi, tra cui Amazon CloudWatch Events o AWS Config regole. Puoi scrivere funzioni Lambda personalizzate per valutare l'utilizzo delle chiavi di accesso IAM, eseguire controlli aggiuntivi e inviare notifiche utilizzando servizi come Amazon SNS o. AWS Chatbot

Convalida continuamente le politiche IAM

Utilizza IAM Access Analyzer per valutare le policy associate ai ruoli e identificare eventuali servizi non utilizzati o azioni in eccesso concesse. Implementa revisioni periodiche degli accessi per verificare manualmente che le politiche soddisfino i requisiti attuali.

Confronta la policy esistente con la policy generata da IAM Access Analyzer e rimuovi tutte le autorizzazioni non necessarie. È inoltre necessario fornire report agli utenti e revocare automaticamente le autorizzazioni non utilizzate dopo un periodo di prova. Ciò contribuisce a garantire che le politiche minime rimangano in vigore.

La revoca proattiva e frequente di accessi obsoleti riduce al minimo le credenziali che potrebbero essere a rischio durante una violazione. L'automazione offre un'igiene sostenibile e a lungo termine delle credenziali e un'ottimizzazione delle autorizzazioni. Il rispetto di questa best practice limita l'ambito dell'impatto applicando in modo proattivo il privilegio minimo tra identità e risorse. AWS

Archiviazione sicura dello stato remoto

L'archiviazione dello stato remoto si riferisce all'archiviazione del file di stato Terraform in remoto anziché localmente sulla macchina su cui è in esecuzione Terraform. Il file di stato è fondamentale perché tiene traccia delle risorse fornite da Terraform e dei relativi metadati.

La mancata protezione dello stato remoto può portare a problemi seri come la perdita di dati sullo stato, l'incapacità di gestire l'infrastruttura, l'eliminazione involontaria delle risorse e l'esposizione di informazioni sensibili che potrebbero essere presenti nel file di stato. Per questo motivo, proteggere l'archiviazione dello stato remoto è fondamentale per l'utilizzo di Terraform a livello di produzione.

Abilita la crittografia e i controlli di accesso

Usa la crittografia lato server (SSE) di Amazon Simple Storage Service (Amazon S3) per crittografare lo stato remoto a riposo.

Limita l'accesso diretto ai flussi di lavoro collaborativi

- Strutturate i flussi di lavoro di collaborazione in HCP Terraform o in una pipeline CI/CD all'interno del vostro repository Git per limitare l'accesso diretto allo stato.
- Affidati alle richieste pull, esegui approvazioni, controlli delle politiche e notifiche per coordinare le modifiche.

Il rispetto di queste linee guida aiuta a proteggere gli attributi sensibili delle risorse ed evita conflitti con le modifiche dei membri del team. La crittografia e le rigide protezioni di accesso aiutano a ridurre la superficie di attacco, mentre i flussi di lavoro di collaborazione favoriscono la produttività.

Usa AWS Secrets Manager

Esistono molte risorse e fonti di dati in Terraform che memorizzano valori segreti in testo semplice nel file di stato. Evita di memorizzare i segreti nello stato: usa invece. AWS Secrets Manager

Invece di tentare di crittografare manualmente i valori sensibili, affidati al supporto integrato di Terraform per la gestione degli stati sensibili. Quando esportate valori sensibili in output, assicuratevi che i valori siano contrassegnati come sensibili.

Scansiona continuamente l'infrastruttura e il codice sorgente

Scansiona in modo proattivo e continuo sia l'infrastruttura che il codice sorgente alla ricerca di rischi quali credenziali esposte o configurazioni errate per rafforzare il livello di sicurezza. Risolvi tempestivamente i risultati riconfigurando o applicando patch alle risorse.

Utilizza i servizi per la scansione dinamica AWS

Utilizza strumenti AWS nativi come Amazon Inspector, AWS Security HubAmazon Detective e Amazon GuardDuty per monitorare l'infrastruttura fornita tra account e regioni. Pianifica le scansioni ricorrenti in Security Hub per tenere traccia delle variazioni di implementazione e configurazione. Scansiona istanze EC2, funzioni Lambda, contenitori, bucket S3 e altre risorse.

Esegui analisi statiche

Incorpora analizzatori statici come Checkov direttamente nelle pipeline CI/CD per scansionare il codice di configurazione Terraform (HCL) e identificare i rischi preventivamente prima dell'implementazione. Ciò sposta i controlli di sicurezza a un punto precedente del processo di sviluppo (denominato spostamento a sinistra) e impedisce un'infrastruttura configurata in modo errato.

Garantisci una pronta riparazione

Per tutti i risultati della scansione, assicurati una pronta riparazione aggiornando la configurazione di Terraform, applicando patch o riconfigurando le risorse manualmente, a seconda dei casi. Riduci i livelli di rischio affrontando le cause alla radice.

L'utilizzo sia della scansione dell'infrastruttura che della scansione del codice fornisce informazioni su più livelli sulle configurazioni Terraform, sulle risorse fornite e sul codice dell'applicazione. Ciò massimizza la copertura del rischio e della conformità attraverso controlli preventivi, investigativi e reattivi, integrando al contempo la sicurezza nelle fasi iniziali del ciclo di vita dello sviluppo del software (SDLC).

Applica i controlli delle politiche

Utilizza framework di codice come le politiche HashiCorp Sentinel per fornire barriere di governance e modelli standardizzati per il provisioning dell'infrastruttura con Terraform.

Le politiche Sentinel possono definire requisiti o restrizioni sulla configurazione di Terraform per allinearsi agli standard organizzativi e alle migliori pratiche. Ad esempio, puoi utilizzare le politiche Sentinel per:

- Richiedi tag su tutte le risorse.
- Limita i tipi di istanze a un elenco approvato.
- Applica le variabili obbligatorie.
- Impedire la distruzione delle risorse di produzione.

L'integrazione dei controlli delle policy nei cicli di vita della configurazione Terraform consente l'applicazione proattiva degli standard e delle linee guida dell'architettura. Sentinel fornisce una logica politica condivisa che aiuta ad accelerare lo sviluppo prevenendo al contempo pratiche non approvate.

Le migliori pratiche di backend

L'utilizzo di un backend remoto adeguato per archiviare il file di stato è fondamentale per consentire la collaborazione, garantire l'integrità dei file dello stato tramite il blocco, fornire backup e ripristino affidabili, integrarsi con i flussi di lavoro CI/CD e sfruttare le funzionalità avanzate di sicurezza, governance e gestione offerte da servizi gestiti come HCP Terraform.

Terraform supporta vari tipi di backend come Kubernetes, Consul e HTTP. HashiCorp Tuttavia, questa guida si concentra su Amazon S3, che è una soluzione di backend ottimale per la maggior parte degli utenti. AWS

Come servizio di storage di oggetti completamente gestito che offre durabilità e disponibilità elevate, Amazon S3 offre un backend sicuro, scalabile e a basso costo per la gestione dello stato di Terraform su. AWS L'impronta globale e la resilienza di Amazon S3 superano ciò che la maggior parte dei team può ottenere gestendo autonomamente lo storage di stato. Inoltre, l'integrazione nativa con controlli di AWS accesso, opzioni di crittografia, funzionalità di controllo delle versioni e altri servizi rende Amazon S3 una comoda scelta di backend.

Questa guida non fornisce indicazioni di back-end per altre soluzioni come Kubernetes o Consul perché il pubblico di riferimento principale sono i clienti. AWS Per i team che lavorano a pieno titolo Cloud AWS, Amazon S3 è in genere la scelta ideale rispetto ai cluster Kubernetes o Consul. HashiCorp La semplicità, la resilienza e la stretta AWS integrazione dello storage di stato Amazon S3 forniscono una base ottimale per la maggior parte degli utenti che AWS seguono le best practice. I team possono sfruttare la durabilità, le protezioni di backup e la disponibilità dei AWS servizi per mantenere lo stato remoto di Terraform altamente resiliente.

Seguire i consigli di backend in questa sezione porterà a basi di codice Terraform più collaborative, limitando al contempo l'impatto di errori o modifiche non autorizzate. Implementando un backend remoto ben progettato, i team possono ottimizzare i flussi di lavoro Terraform.

Migliori pratiche:

- Usa Amazon S3 per lo storage remoto
- Facilita la collaborazione in team
- Separa i backend per ogni ambiente
- Monitora attivamente l'attività dello stato remoto

Usa Amazon S3 per lo storage remoto

L'archiviazione dello stato di Terraform in remoto in Amazon S3 e l'implementazione del blocco dello stato e del controllo della coerenza utilizzando Amazon DynamoDB offrono importanti vantaggi rispetto allo storage locale di file. Lo stato remoto consente la collaborazione in team, il monitoraggio delle modifiche, le protezioni di backup e il blocco remoto per una maggiore sicurezza.

L'utilizzo di Amazon S3 con la classe di storage S3 Standard (impostazione predefinita) anziché lo storage locale temporaneo o le soluzioni autogestite offre una durabilità del 99,25% e una disponibilità del 99,99% per prevenire la perdita accidentale dei dati dello stato. AWS i servizi gestiti come Amazon S3 e DynamoDB forniscono accordi sui livelli di servizio (SLA) che superano ciò che la maggior parte delle organizzazioni può ottenere gestendo autonomamente lo storage. Affidati a queste protezioni per mantenere accessibili i backend remoti.

Abilita il blocco remoto dello stato

Il blocco di DynamoDB limita l'accesso allo stato per impedire operazioni di scrittura simultanee. Ciò impedisce modifiche simultanee da parte di più utenti e riduce gli errori.

Esempio di configurazione del backend con blocco dello stato:

```
terraform {
  backend "s3" {
    bucket         = "myorg-terraform-states"
    key            = "myapp/production/tfstate"
    region         = "us-east-1"
    dynamodb_table = "TerraformStateLocking"
  }
}
```

Abilita il controllo delle versioni e i backup automatici

Per una protezione aggiuntiva, abilita il controllo automatico delle versioni e i backup utilizzando i backend di Amazon AWS Backup S3. Il controllo delle versioni conserva tutte le versioni precedenti dello stato ogni volta che vengono apportate modifiche. Consente inoltre di ripristinare istantanee dello stato di funzionamento precedente, se necessario, per ripristinare le modifiche indesiderate o il ripristino in caso di incidenti.

Ripristina le versioni precedenti, se necessario

I bucket di stato Amazon S3 con versione semplificano l'annullamento delle modifiche ripristinando uno snapshot precedente noto in buono stato. Questo aiuta a proteggere da modifiche accidentali e fornisce funzionalità di backup aggiuntive.

Usa HCP Terraform

HCP Terraform fornisce un'alternativa di backend completamente gestita alla configurazione del proprio storage di stato. HCP Terraform gestisce automaticamente l'archiviazione sicura dello stato e della crittografia sbloccando funzionalità aggiuntive.

Quando si utilizza HCP Terraform, lo stato viene archiviato in remoto per impostazione predefinita, il che consente la condivisione e il blocco dello stato all'interno dell'organizzazione. I controlli dettagliati delle politiche aiutano a limitare l'accesso e le modifiche statali.

Le funzionalità aggiuntive includono integrazioni per il controllo delle versioni, barriere politiche, automazione del flusso di lavoro, gestione delle variabili e integrazioni Single Sign-On con SAML. Puoi anche utilizzare la policy di Sentinel come codice per implementare i controlli di governance.

Sebbene HCP Terraform richieda l'utilizzo di una piattaforma SaaS (Software as a Service), per molti team i vantaggi in termini di sicurezza, controllo degli accessi, controlli automatici delle policy e funzionalità di collaborazione lo rendono una scelta ottimale rispetto allo storage di stato a gestione automatica con Amazon S3 o DynamoDB.

La facile integrazione con servizi come GitHub e GitLab con configurazioni minori piace anche agli utenti che adottano appieno gli strumenti cloud e SaaS per flussi di lavoro di squadra migliori.

Facilita la collaborazione in team

Usa i backend remoti per condividere i dati sullo stato tra tutti i membri del tuo team Terraform. Ciò facilita la collaborazione perché offre all'intero team la visibilità sui cambiamenti dell'infrastruttura. I protocolli di backend condivisi combinati con la trasparenza della cronologia dello stato semplificano la gestione interna delle modifiche. Tutte le modifiche all'infrastruttura passano attraverso la pipeline consolidata, che aumenta l'agilità aziendale in tutta l'azienda.

Migliora la responsabilità utilizzando AWS CloudTrail

Esegui l'integrazione AWS CloudTrail con il bucket Amazon S3 per acquisire le chiamate API effettuate allo state bucket. Filtra CloudTrail gli eventi per tracciarli `PutObject` `DeleteObject`, e altre chiamate pertinenti.

CloudTrail i log mostrano l' AWS identità del principale che ha effettuato ogni chiamata API per la modifica dello stato. L'identità dell'utente può essere associata all'account di un computer o ai membri del team che interagiscono con lo storage di backend.

Combina CloudTrail i log con il controllo delle versioni dello stato di Amazon S3 per collegare le modifiche all'infrastruttura al principale che le ha applicate. Analizzando più revisioni, puoi attribuire eventuali aggiornamenti all'account del computer o al membro del team responsabile.

Se si verifica una modifica involontaria o dirompente, il controllo delle versioni dello stato offre funzionalità di rollback. CloudTrail traccia la modifica all'utente in modo da poter discutere dei miglioramenti preventivi.

Ti consigliamo inoltre di applicare le autorizzazioni IAM per limitare l'accesso allo State Bucket. Nel complesso, il controllo delle versioni e il CloudTrail monitoraggio di S3 supportano il controllo in tutte le modifiche all'infrastruttura. I team acquisiscono maggiori responsabilità, trasparenza e capacità di controllo nella storia dello stato di Terraform.

Separa i backend per ogni ambiente

Usa backend Terraform distinti per ogni ambiente applicativo. I backend separati isolano lo stato tra sviluppo, test e produzione.

Riduci la portata dell'impatto

Lo stato di isolamento aiuta a garantire che le modifiche negli ambienti inferiori non influiscano sull'infrastruttura di produzione. Gli incidenti o gli esperimenti negli ambienti di sviluppo e test hanno un impatto limitato.

Limita l'accesso alla produzione

Blocca le autorizzazioni per il backend dello stato di produzione per consentire l'accesso in sola lettura per la maggior parte degli utenti. Limita chi può modificare l'infrastruttura di produzione alla pipeline CI/CD e ai ruoli di Break Glass.

Semplifica i controlli di accesso

La gestione delle autorizzazioni a livello di backend semplifica il controllo degli accessi tra gli ambienti. L'utilizzo di bucket S3 distinti per ogni applicazione e ambiente significa che è possibile concedere ampie autorizzazioni di lettura o scrittura su interi bucket di backend.

Evita gli spazi di lavoro condivisi

Sebbene sia possibile utilizzare gli spazi di lavoro Terraform per separare gli stati tra gli ambienti, backend distinti offrono un isolamento maggiore. Se disponi di spazi di lavoro condivisi, gli incidenti possono comunque avere un impatto su più ambienti.

Mantenere i backend dell'ambiente completamente isolati riduce al minimo l'impatto di ogni singolo guasto o violazione. I backend separati allineano inoltre i controlli di accesso al livello di sensibilità dell'ambiente. Ad esempio, è possibile fornire protezione da scrittura per l'ambiente di produzione e un accesso più ampio per gli ambienti di sviluppo e test.

Monitora attivamente l'attività dello stato remoto

Il monitoraggio continuo dell'attività dello stato remoto è fondamentale per rilevare tempestivamente potenziali problemi. Cerca sblocchi, modifiche o tentativi di accesso anomali.

Ricevi avvisi in caso di sblocchi sospetti

La maggior parte delle modifiche di stato dovrebbe avvenire tramite pipeline CI/CD. Genera avvisi se gli stati sbloccati avvengono direttamente tramite le workstation degli sviluppatori, il che potrebbe segnalare modifiche non autorizzate o non testate.

Monitora i tentativi di accesso

Gli errori di autenticazione nei bucket di stato potrebbero indicare un'attività di ricognizione. Notate se più account stanno tentando di accedere allo stato o se compaiono indirizzi IP insoliti, che segnalano credenziali compromesse.

Procedure consigliate per la struttura e l'organizzazione della base di codice

La struttura e l'organizzazione corrette della base di codice sono fondamentali poiché l'utilizzo di Terraform cresce tra team e aziende di grandi dimensioni. Una base di codice ben architettata consente la collaborazione su larga scala migliorando al contempo la manutenibilità.

Questa sezione fornisce consigli sulla modularità di Terraform, sulle convenzioni di denominazione, sulla documentazione e sugli standard di codifica che supportano qualità e coerenza.

Le linee guida includono la suddivisione della configurazione in moduli riutilizzabili per ambiente e componenti, la definizione di convenzioni di denominazione utilizzando prefissi e suffissi, la documentazione dei moduli e la spiegazione chiara di input e output e l'applicazione di regole di formattazione coerenti utilizzando controlli di stile automatici.

Le migliori pratiche aggiuntive riguardano l'organizzazione logica di moduli e risorse in una gerarchia strutturata, la catalogazione di moduli pubblici e privati nella documentazione e l'astrazione dei dettagli di implementazione non necessari nei moduli per semplificarne l'utilizzo.

Implementando le linee guida sulla struttura del codice base relative a modularità, documentazione, standard e organizzazione logica, puoi supportare un'ampia collaborazione tra i team mantenendo Terraform gestibile man mano che l'utilizzo si diffonde all'interno dell'organizzazione. Applicando convenzioni e standard, è possibile evitare la complessità di una base di codice frammentata.

Le migliori pratiche:

- Implementa una struttura di repository standard
- Struttura per la modularità
- Segui le convenzioni di denominazione
- Usa le risorse relative agli allegati
- Usa tag predefiniti
- Soddisfa i requisiti del registro Terraform
- Usa i sorgenti dei moduli consigliati
- Segui gli standard di codifica

Implementa una struttura di repository standard

Si consiglia di implementare il seguente layout di repository. La standardizzazione su queste pratiche di coerenza tra i moduli migliora la reperibilità, la trasparenza, l'organizzazione e l'affidabilità, consentendo al contempo il riutilizzo in molte configurazioni Terraform.

- Modulo o directory principale: questo dovrebbe essere il punto di ingresso principale sia per i moduli root che per quelli riutilizzabili di Terraform e dovrebbe essere unico. Se disponi di un'architettura più complessa, puoi utilizzare moduli annidati per creare astrazioni leggere. Questo aiuta a descrivere l'infrastruttura in termini di architettura anziché direttamente, in termini di oggetti fisici.
- README: il modulo root e tutti i moduli annidati devono avere file README. Questo file deve avere un nome. `README.md` Dovrebbe contenere una descrizione del modulo e per cosa dovrebbe essere usato. Se vuoi includere un esempio di utilizzo di questo modulo con altre risorse, inseriscilo in una `examples` directory. Prendi in considerazione l'inclusione di un diagramma che illustri le risorse infrastrutturali che il modulo potrebbe creare e le relative relazioni. Usa terraform-docs per generare automaticamente input o output del modulo.
- main.tf: questo è il punto di ingresso principale. Per un modulo semplice, tutte le risorse potrebbero essere create in questo file. Per un modulo complesso, la creazione di risorse potrebbe essere distribuita su più file, ma tutte le chiamate di modulo annidate dovrebbero trovarsi nel `main.tf` file.
- variables.tf e outputs.tf: questi file contengono le dichiarazioni per le variabili e gli output. Tutte le variabili e gli output devono avere descrizioni di una o due frasi che ne spieghino lo scopo. Queste descrizioni vengono utilizzate a scopo di documentazione. Per ulteriori informazioni, consulta la HashiCorp documentazione per la configurazione delle variabili e la configurazione dell'output.
  - Tutte le variabili devono avere un tipo definito.
  - La dichiarazione della variabile può includere anche un argomento predefinito. Se la dichiarazione include un argomento predefinito, la variabile è considerata facoltativa e il valore predefinito viene utilizzato se non si imposta un valore quando si chiama il modulo o si esegue Terraform. L'argomento predefinito richiede un valore letterale e non può fare riferimento ad altri oggetti nella configurazione. Per rendere obbligatoria una variabile, omettete un valore predefinito nella dichiarazione della variabile e valutate se l'impostazione `nullable = false` ha senso.
  - Per le variabili che hanno valori indipendenti dall'ambiente (ad esempio`disk_size`), fornite valori predefiniti.

  - Per le variabili che hanno valori specifici dell'ambiente (come`project_id`), non fornite valori predefiniti. In questo caso, il modulo chiamante deve fornire valori significativi.
  - Utilizzare valori predefiniti vuoti per variabili come stringhe o elenchi vuoti solo quando lasciare la variabile vuota è una preferenza valida che le API sottostanti non rifiutano.
  - Sii prudente nell'uso delle variabili. Parametrizzate i valori solo se devono variare per ogni istanza o ambiente. Quando decidete se esporre una variabile, assicuratevi di avere un caso d'uso concreto per modificare quella variabile. Se c'è solo una piccola possibilità che una variabile possa essere necessaria, non esporla.
    - L'aggiunta di una variabile con un valore predefinito è retrocompatibile.
    - La rimozione di una variabile è incompatibile con le versioni precedenti.
    - Nei casi in cui un valore letterale viene riutilizzato in più posizioni, è necessario utilizzare un valore locale senza esporlo come variabile.
  - Non passate gli output direttamente attraverso le variabili di input, perché così facendo si impedisce che vengano aggiunti correttamente al grafico delle dipendenze. Per garantire che vengano create dipendenze implicite, assicuratevi che le risorse restituiscano gli attributi di riferimento. Invece di fare riferimento direttamente a una variabile di input per un'istanza, passate l'attributo.
- locals.tf: questo file contiene valori locali che assegnano un nome a un'espressione, quindi un nome può essere utilizzato più volte all'interno di un modulo anziché ripetere l'espressione. I valori locali sono come le variabili locali temporanee di una funzione. Le espressioni nei valori locali non si limitano alle costanti letterali; possono anche fare riferimento ad altri valori nel modulo, tra cui variabili, attributi di risorse o altri valori locali, per combinarli.
- providers.tf : questo file contiene il blocco terraform e i blocchi provider. `provider`i blocchi devono essere dichiarati solo nei moduli root dai consumatori di moduli.

  Se utilizzi HCP Terraform, aggiungi anche un blocco cloud vuoto. Il `cloud` blocco deve essere configurato interamente tramite variabili di ambiente e credenziali di variabili di ambiente come parte di una pipeline CI/CD.
- versions.tf: questo file contiene il blocco required_providers. Tutti i moduli Terraform devono dichiarare i provider necessari in modo che Terraform possa installare e utilizzare questi provider.
- data.tf: per una configurazione semplice, metti le fonti di dati accanto alle risorse che vi fanno riferimento. Ad esempio, se state recuperando un'immagine da utilizzare per lanciare un'istanza, posizionatela accanto all'istanza invece di raccogliere risorse di dati nel relativo file. Se il numero di fonti di dati diventa troppo grande, valuta la possibilità di spostarle in un file dedicato`data.tf`.

- .tfvars: per i moduli root, puoi fornire variabili non sensibili utilizzando un file. `.tfvars` Per coerenza, assegnate un nome ai file delle variabili. `terraform.tfvars` Inserisci i valori comuni nella radice del repository e i valori specifici dell'ambiente all'interno della cartella. `envs/`
- Moduli annidati: i moduli annidati devono esistere nella sottodirectory. `modules/` Qualsiasi modulo annidato che dispone di un `README.md` è considerato utilizzabile da un utente esterno. Se un `README.md` non esiste, il modulo viene considerato solo per uso interno. I moduli annidati devono essere utilizzati per suddividere il comportamento complesso in più moduli di piccole dimensioni che gli utenti possano selezionare e scegliere con cura.

  Se il modulo root include chiamate a moduli annidati, queste chiamate dovrebbero utilizzare percorsi relativi, `./modules/sample-module` in modo che Terraform li consideri parte dello stesso repository o pacchetto invece di scaricarli nuovamente separatamente.

  Se un repository o un pacchetto contiene più moduli nidificati, idealmente dovrebbero essere componibili dal chiamante invece di chiamarsi direttamente l'un l'altro e creare un albero di moduli profondamente annidato.
- Esempi: nella `examples/` sottodirectory alla radice del repository dovrebbero esistere esempi di utilizzo di un modulo riutilizzabile. Per ogni esempio, puoi aggiungere un README per spiegare l'obiettivo e l'utilizzo dell'esempio. Gli esempi di sottomoduli devono essere collocati anche nella directory principale`examples/`.

  Poiché gli esempi vengono spesso copiati in altri repository per la personalizzazione, i blocchi di moduli dovrebbero avere la fonte impostata sull'indirizzo che utilizzerebbe un chiamante esterno, non su un percorso relativo.
- File denominati del servizio: gli utenti spesso desiderano separare le risorse Terraform per servizio in più file. Questa pratica dovrebbe essere scoraggiata il più possibile e le risorse dovrebbero invece essere definite in. `main.tf` Tuttavia, se una raccolta di risorse (ad esempio, ruoli e policy IAM) supera le 150 righe, è ragionevole suddividerla in file separati, ad esempio. `iam.tf` Altrimenti, tutto il codice delle risorse dovrebbe essere definito in. `main.tf`
- Script personalizzati: utilizzare gli script solo quando necessario. Terraform non tiene conto né gestisce lo stato delle risorse create tramite script. Usa script personalizzati solo quando le risorse Terraform non supportano il comportamento desiderato. Inserisci gli script personalizzati chiamati da Terraform in una directory. `scripts/`
- Script di supporto: organizza gli script di supporto che non vengono chiamati da Terraform in una directory. `helpers/` Documenta gli script di supporto nel file con spiegazioni ed esempi di

invocazioni. `README.md` Se gli script di supporto accettano argomenti, fornisci il controllo degli argomenti e l'output. `--help`

- File statici: i file statici a cui Terraform fa riferimento ma che non vengono eseguiti (come gli script di avvio caricati su istanze EC2) devono essere organizzati in una directory. `files/` Inserisci documenti lunghi in file esterni, separati dal loro HCL. Fateli riferimento con la funzione file ().
- Modelli: per i file in cui legge la funzione Terraform templatefile, usa l'estensione del file. `.tftpl` I modelli devono essere inseriti in una directory. `templates/`

Struttura del modulo principale

Terraform viene sempre eseguito nel contesto di un singolo modulo root. Una configurazione Terraform completa consiste in un modulo root e nell'albero dei moduli figlio (che include i moduli chiamati dal modulo root, tutti i moduli chiamati da tali moduli e così via).

Esempio di base del layout del modulo root Terraform:

```
.
### data.tf
### envs
#   ### dev
#   #   ### terraform.tfvars
#   ### prod
#   #   ### terraform.tfvars
#   ### test
#       ### terraform.tfvars
### locals.tf
### main.tf
### outputs.tf
### providers.tf
### README.md
### terraform.tfvars
### variables.tf
### versions.tf
```

Struttura del modulo riutilizzabile

I moduli riutilizzabili seguono gli stessi concetti dei moduli root. Per definire un modulo, create una nuova directory e inserite `.tf` i file al suo interno, proprio come fareste con un modulo root. Terraform può caricare i moduli da percorsi relativi locali o da repository remoti. Se prevedi che un

modulo venga riutilizzato da molte configurazioni, inseriscilo nel suo repository di controllo della versione. È importante mantenere l'albero dei moduli relativamente piatto per facilitare il riutilizzo dei moduli in combinazioni diverse.

Esempio di base del layout del modulo riutilizzabile Terraform:

```
.
### data.tf
### examples
#   ### multi-az-new-vpc
#   #   ### data.tf
#   #   ### locals.tf
#   #   ### main.tf
#   #   ### outputs.tf
#   #   ### providers.tf
#   #   ### README.md
#   #   ### terraform.tfvars
#   #   ### variables.tf
#   #   ### versions.tf
#   #   ### vpc.tf
#   ### single-az-existing-vpc
#   #   ### data.tf
#   #   ### locals.tf
#   #   ### main.tf
#   #   ### outputs.tf
#   #   ### providers.tf
#   #   ### README.md
#   #   ### terraform.tfvars
#   #   ### variables.tf
#   #   ### versions.tf
### iam.tf
### locals.tf
### main.tf
### outputs.tf
### README.md
### variables.tf
### versions.tf
```

Struttura per la modularità

In linea di principio, puoi combinare qualsiasi risorsa e altri costrutti in un modulo, ma un uso eccessivo di moduli annidati e riutilizzabili può rendere più difficile la comprensione e la

manutenzione della configurazione complessiva di Terraform, quindi usa questi moduli con moderazione.

Quando ha senso, suddividi la configurazione in moduli riutilizzabili che aumentano il livello di astrazione descrivendo un nuovo concetto nell'architettura costruito con tipi di risorse.

Quando modularizzate l'infrastruttura in definizioni riutilizzabili, puntate a set logici di risorse anziché a singoli componenti o raccolte eccessivamente complesse.

Non racchiudete singole risorse

Non dovreste creare moduli che siano involucri sottili attorno ad altri tipi di risorse singole. Se hai difficoltà a trovare un nome per il tuo modulo che sia diverso dal nome del tipo di risorsa principale al suo interno, probabilmente il modulo non sta creando una nuova astrazione, ma sta aggiungendo complessità non necessarie. Utilizza invece il tipo di risorsa direttamente nel modulo chiamante.

Incapsula le relazioni logiche

Raggruppa set di risorse correlate come basi di rete, livelli di dati, controlli di sicurezza e applicazioni. Un modulo riutilizzabile dovrebbe incapsulare componenti dell'infrastruttura che interagiscono per abilitare una funzionalità.

Mantieni invariata l'ereditarietà

Quando annidi i moduli nelle sottodirectory, evita di approfondire più di uno o due livelli. Strutture ereditarie profondamente annidate complicano le configurazioni e la risoluzione dei problemi. I moduli devono basarsi su altri moduli, non creare tunnel attraverso di essi.

Concentrando i moduli su raggruppamenti di risorse logiche che rappresentano modelli di architettura, i team possono configurare rapidamente basi infrastrutturali affidabili. Bilancia l'astrazione senza ingegnerizzare o semplificare eccessivamente.

Risorse di riferimento negli output

Per ogni risorsa definita in un modulo riutilizzabile, includi almeno un output che faccia riferimento alla risorsa. Le variabili e gli output consentono di dedurre le dipendenze tra moduli e risorse. Senza alcun output, gli utenti non possono ordinare correttamente il modulo in relazione alle loro configurazioni Terraform.

Moduli ben strutturati che forniscono coerenza ambientale, raggruppamenti mirati e riferimenti alle risorse esportati consentono la collaborazione Terraform a livello di organizzazione su larga scala. I team possono assemblare l'infrastruttura partendo da elementi costitutivi riutilizzabili.

Non configurare i provider

Sebbene i moduli condivisi ereditino i provider dai moduli di chiamata, i moduli non dovrebbero configurare autonomamente le impostazioni del provider. Evita di specificare i blocchi di configurazione del provider nei moduli. Questa configurazione deve essere dichiarata solo una volta a livello globale.

Dichiarare i provider richiesti

Sebbene le configurazioni dei provider siano condivise tra i moduli, i moduli condivisi devono inoltre dichiarare i propri requisiti di provider. Questa pratica consente a Terraform di garantire che esista un'unica versione del provider compatibile con tutti i moduli della configurazione e di specificare l'indirizzo di origine che funge da identificatore globale (indipendente dal modulo) per il provider. Tuttavia, i requisiti del provider specifici del modulo non specificano nessuna delle impostazioni di configurazione che determinano a quali endpoint remoti accederà il provider, ad esempio un. Regione AWS

Dichiarando i requisiti di versione ed evitando la configurazione del provider codificata, i moduli forniscono portabilità e riusabilità tra le configurazioni Terraform utilizzando provider condivisi.

Per i moduli condivisi, definisci le versioni minime richieste del provider in un blocco `required_providers` in. `versions.tf`

Per dichiarare che un modulo richiede una particolare versione del AWS provider, usa un blocco all'interno di un `required_providers` blocco: `terraform`

```
terraform {
  required_version = ">= 1.0.0"

  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = ">= 4.0.0"
    }
  }
}
```

Se un modulo condiviso supporta solo una versione specifica del AWS provider, utilizzate l'operatore di vincolo pessimistico (~>), che consente solo al componente della versione più a destra di incrementare:

```
terraform {
  required_version = ">= 1.0.0"

  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 4.0"
    }
  }
}
```

In questo esempio, `~> 4.0` consente l'installazione di and but not. `4.57.1` `4.67.0` `5.0.0` Per ulteriori informazioni, vedete Version Constraint Syntax nella documentazione. HashiCorp

Segui le convenzioni di denominazione

Nomi chiari e descrittivi semplificano la comprensione delle relazioni tra le risorse del modulo e lo scopo dei valori di configurazione. La coerenza con le linee guida di stile migliora la leggibilità sia per gli utenti che per i manutentori del modulo.

Segui le linee guida per la denominazione delle risorse

- Usa snake_case (dove i termini minuscoli sono separati da caratteri di sottolineatura) per far sì che tutti i nomi delle risorse corrispondano agli standard di stile Terraform. Questa pratica garantisce la coerenza con la convenzione di denominazione per i tipi di risorse, i tipi di fonti di dati e altri valori predefiniti. Questa convenzione non si applica agli argomenti relativi ai nomi.
- Per semplificare i riferimenti a una risorsa che è l'unica del suo tipo (ad esempio, un singolo sistema di bilanciamento del carico per un intero modulo), assegnate un nome alla risorsa `main` o `this` per chiarezza.
- Utilizzate nomi significativi che descrivano lo scopo e il contesto della risorsa e che aiutino a distinguere tra risorse simili (ad esempio, per il database principale e `primary` `read_replica` per una replica in lettura del database).
- Usa nomi singolari, non plurali.

- Non ripetete il tipo di risorsa nel nome della risorsa.

Segui le linee guida per la denominazione delle variabili

- Aggiungi unità ai nomi di input, variabili locali e output che rappresentano valori numerici come la dimensione del disco o la dimensione della RAM (ad esempio, `ram_size_gb` per la dimensione della RAM in gigabyte). Questa pratica rende chiara l'unità di input prevista per i gestori della configurazione.
- Utilizza unità binarie come MiB e GiB per le dimensioni di archiviazione e unità decimali come MB o GB per altre metriche.
- Assegna alle variabili booleane nomi positivi come. `enable_external_access`

Usa le risorse relative agli allegati

Alcune risorse contengono delle pseudo-risorse incorporate come attributi. Ove possibile, dovresti evitare di utilizzare questi attributi di risorsa incorporati e utilizzare invece la risorsa unica per allegare quella pseudo-risorsa. Queste relazioni tra le risorse possono causare cause-and-effect problemi unici per ogni risorsa.

Utilizzando un attributo incorporato (evita questo schema):

```
resource "aws_security_group" "allow_tls" {
  ...
  ingress {
    description      = "TLS from VPC"
    from_port        = 443
    to_port          = 443
    protocol         = "tcp"
    cidr_blocks      = [aws_vpc.main.cidr_block]
    ipv6_cidr_blocks = [aws_vpc.main.ipv6_cidr_block]
  }

  egress {
    from_port        = 0
    to_port          = 0
    protocol         = "-1"
    cidr_blocks      = ["0.0.0.0/0"]
    ipv6_cidr_blocks = ["::/0"]
  }
```

```
}
```

Utilizzo delle risorse relative agli allegati (preferito):

```
resource "aws_security_group" "allow_tls" {
  ...
}

resource "aws_security_group_rule" "example" {
  type              = "ingress"
  description      = "TLS from VPC"
  from_port        = 443
  to_port          = 443
  protocol         = "tcp"
  cidr_blocks      = [aws_vpc.main.cidr_block]
  ipv6_cidr_blocks = [aws_vpc.main.ipv6_cidr_block]
  security_group_id = aws_security_group.allow_tls.id
}
```

Usa tag predefiniti

Assegna tag a tutte le risorse che possono accettare tag. Il Terraform AWS Provider ha una fonte di dati aws_default_tags che dovresti usare all'interno del modulo root.

Prendi in considerazione l'aggiunta dei tag necessari a tutte le risorse create da un modulo Terraform. Ecco un elenco di possibili tag da allegare:

- Nome: nome della risorsa leggibile dall'uomo
- AppId: ID dell'applicazione che utilizza la risorsa
- AppRole: La funzione tecnica della risorsa, ad esempio «webserver» o «database»
- AppPurpose: Lo scopo commerciale della risorsa, ad esempio «interfaccia utente frontend» o «processore di pagamento»
- Ambiente: l'ambiente software, ad esempio dev, test o prod
- Progetto: i progetti che utilizzano la risorsa
- CostCenter: A chi fatturare l'utilizzo delle risorse

Soddisfa i requisiti del registro Terraform

Un repository di moduli deve soddisfare tutti i seguenti requisiti per poter essere pubblicato su un registro Terraform.

Dovresti sempre seguire questi requisiti anche se non hai intenzione di pubblicare il modulo in un registro a breve termine. In questo modo, è possibile pubblicare il modulo in un registro in un secondo momento senza dover modificare la configurazione e la struttura del repository.

- Nome dell'archivio: per un archivio di moduli, usa il nome in tre parti`terraform-aws-<NAME>`, che `<NAME>` riflette il tipo di infrastruttura gestita dal modulo. Il `<NAME>` segmento può contenere trattini aggiuntivi (ad esempio,). `terraform-aws-iam-terraform-roles`
- Struttura del modulo standard: il modulo deve aderire alla struttura standard del repository. Ciò consente al registro di ispezionare il modulo e generare documentazione, tenere traccia dell'utilizzo delle risorse e altro ancora.
  - Dopo aver creato il repository Git, copia i file del modulo nella radice del repository. Ti consigliamo di collocare ogni modulo destinato a essere riutilizzabile nella radice del relativo repository, ma puoi anche fare riferimento ai moduli delle sottodirectory.
  - Se utilizzi HCP Terraform, pubblica i moduli che devono essere condivisi nel registro della tua organizzazione. Il registro gestisce i download e controlla l'accesso con i token API HCP Terraform, quindi i consumatori non devono accedere al repository di origine del modulo anche quando eseguono Terraform dalla riga di comando.
- Posizione e autorizzazioni: il repository deve trovarsi in uno dei provider del sistema di controllo della versione (VCS) configurato e l'account utente HCP Terraform VCS deve avere accesso amministratore al repository. Il registro richiede l'accesso da amministratore per creare i webhook per importare nuove versioni del modulo.
- Tag x.y.z per le versioni: è necessario che sia presente almeno un tag release per poter pubblicare un modulo. Il registro utilizza i tag di rilascio per identificare le versioni dei moduli. I nomi dei tag di release devono utilizzare il controllo delle versioni semantico, a cui è possibile aggiungere facoltativamente il prefisso a `v` (ad esempio, e). `v1.1.0` `1.1.0` Il registro ignora i tag che non assomigliano ai numeri di versione. Per ulteriori informazioni sulla pubblicazione dei moduli, consulta la documentazione di Terraform.

Per ulteriori informazioni, consulta Preparazione di un repository di moduli nella documentazione di Terraform.

Usa i sorgenti dei moduli consigliati

Terraform utilizza l'`source`argomento in un blocco di modulo per trovare e scaricare il codice sorgente di un modulo figlio.

Ti consigliamo di utilizzare percorsi locali per moduli strettamente correlati che hanno lo scopo principale di estrarre elementi di codice ripetuti e di utilizzare un registro di moduli Terraform nativo o un provider VCS per moduli destinati a essere condivisi da più configurazioni.

Gli esempi seguenti illustrano i tipi di sorgenti più comuni e consigliati per la condivisione dei moduli. I moduli di registro supportano il controllo delle versioni. È necessario fornire sempre una versione specifica, come illustrato negli esempi seguenti.

Registro

Registro Terraform:

```
module "lambda" {
  source = "github.com/terraform-aws-modules/terraform-aws-lambda.git?
ref=e78cdf1f82944897ca6e30d6489f43cf24539374" #--> v4.18.0

  ...

}
```

Aggiungendo gli hash di commit, è possibile evitare la deviazione dai registri pubblici che sono vulnerabili agli attacchi alla catena di approvvigionamento.

HCP Terraform:

```
module "eks_karpenter" {
  source = "app.terraform.io/my-org/eks/aws"
  version = "1.1.0"

  ...

  enable_karpenter = true
}
```

Terraform Enterprise:

```
module "eks_karpenter" {
  source = "terraform.mydomain.com/my-org/eks/aws"
  version = "1.1.0"

  ...

  enable_karpenter = true
}
```

Fornitori di VCS

I provider VCS supportano l'`ref`argomento a favore della selezione di una revisione specifica, come illustrato negli esempi seguenti.

GitHub (HTTPS):

```
module "eks_karpenter" {
  source = "github.com/my-org/terraform-aws-eks.git?ref=v1.1.0"

  ...

  enable_karpenter = true
}
```

Archivio Git generico (HTTPS):

```
module "eks_karpenter" {
  source = "git::https://example.com/terraform-aws-eks.git?ref=v1.1.0"

  ...

  enable_karpenter = true
}
```

Archivio Git generico (SSH):

> **Warning**
>
> È necessario configurare le credenziali per accedere agli archivi privati.

```
module "eks_karpenter" {
  source = "git::ssh://username@example.com/terraform-aws-eks.git?ref=v1.1.0"

  ...

  enable_karpenter = true
}
```

Segui gli standard di codifica

Applica regole e stili di formattazione Terraform coerenti su tutti i file di configurazione. Applica gli standard utilizzando controlli di stile automatizzati nelle pipeline CI/CD. Quando incorpori le migliori pratiche di codifica nei flussi di lavoro dei team, le configurazioni rimangono leggibili, gestibili e collaborative anche se l'utilizzo si diffonde ampiamente all'interno dell'organizzazione.

Segui le linee guida di stile

- Formatta tutti i file (`.tf`file) Terraform con il comando terraform fmt in modo che HashiCorp soddisfino gli standard di stile.
- Usa il comando terraform validate per verificare la sintassi e la struttura della tua configurazione.
- Analizza staticamente la qualità del codice utilizzando TFlint. Questo linter verifica le migliori pratiche di Terraform oltre alla semplice formattazione e fallisce le build quando riscontra errori.

Configura gli hook di pre-commit

Configura gli hook di pre-commit sul lato client che eseguono e `terraform fmt` eseguono altre scansioni del codice e controlli di stile prima di consentire i commit. `tflint` `checkov` Questa pratica consente di convalidare la conformità agli standard nelle fasi iniziali dei flussi di lavoro degli sviluppatori.

Usa framework di pre-commit come pre-commit per aggiungere il linting, la formattazione e la scansione del codice Terraform come hook sul tuo computer locale. Gli hook vengono eseguiti su ogni commit Git e falliscono il commit se i controlli non vengono superati.

Lo spostamento dei controlli di stile e qualità negli hook locali di pre-commit fornisce un feedback rapido agli sviluppatori prima che vengano introdotte le modifiche. Gli standard entrano a far parte del flusso di lavoro di codifica.

Procedure consigliate per la gestione delle versioni del AWS provider

La gestione attenta delle versioni del AWS Provider e dei moduli Terraform associati è fondamentale per la stabilità. Questa sezione descrive le migliori pratiche relative ai vincoli di versione e agli aggiornamenti.

Le migliori pratiche:

- Aggiungi controlli automatici delle versioni
- Monitora le nuove versioni
- Contribuisci ai fornitori

Aggiungi controlli automatici delle versioni

Aggiungi controlli di versione per i provider Terraform nelle tue pipeline CI/CD per convalidare il blocco delle versioni e fallisci le build se la versione non è definita.

- Aggiungi i controlli TFlint nelle pipeline CI/CD per cercare le versioni del provider che non hanno vincoli di versione principale/secondaria fissati. Utilizza il plug-in TFlint ruleset per Terraform AWS Provider, che fornisce regole per rilevare possibili errori e verifica le migliori pratiche sulle risorse. AWS
- Esecuzioni Fail CI che rilevano le versioni dei provider non bloccate per impedire che gli aggiornamenti impliciti raggiungano la produzione.

Monitora le nuove versioni

- Monitora le note di rilascio e i feed dei log delle modifiche del provider. Ricevi notifiche sulle nuove versioni principali/secondarie.
- Valuta le note di rilascio per individuare eventuali modifiche sostanziali e valuta il loro impatto sull'infrastruttura esistente.
- Aggiorna prima le versioni secondarie in ambienti non di produzione per convalidarle prima di aggiornare l'ambiente di produzione.

Automatizzando i controlli delle versioni nelle pipeline e monitorando le nuove versioni, puoi catturare tempestivamente gli aggiornamenti non supportati e dare ai tuoi team il tempo di valutare l'impatto delle nuove versioni principali/secondarie prima di aggiornare gli ambienti di produzione.

Contribuisci ai fornitori

Contribuisci attivamente al HashiCorp AWS fornitore segnalando difetti o richiedendo funzionalità nei GitHub problemi:

- Apri i problemi ben documentati nell'archivio del AWS Provider per dettagliare eventuali bug riscontrati o funzionalità mancanti. Fornisci passaggi riproducibili.
- Richiedi e vota i miglioramenti per espandere le capacità del AWS fornitore di gestire nuovi servizi.
- Fai riferimento alle pull request emesse quando contribuisci con proposte di correzioni per difetti o miglioramenti del provider. Link a problemi correlati.
- Segui le linee guida per i contributi nell'archivio per le convenzioni di codifica, gli standard di test e la documentazione.

Contribuendo ai provider che utilizzi, puoi fornire un contributo diretto alla loro tabella di marcia e contribuire a migliorarne la qualità e le funzionalità per tutti gli utenti.

Le migliori pratiche per i moduli della community

L'uso efficace dei moduli è fondamentale per gestire configurazioni Terraform complesse e promuovere il riutilizzo. Questa sezione fornisce le migliori pratiche relative ai moduli della comunità, alle dipendenze, alle fonti, all'astrazione e ai contributi.

Le migliori pratiche:

- Scopri i moduli della community
- Comprendi le dipendenze
- Utilizza fonti attendibili
- Contribuisci ai moduli della community

Scopri i moduli della community

Cerca nel registro Terraform e in altre fonti i AWS moduli esistenti che potrebbero risolvere il tuo caso d'uso prima di creare un nuovo modulo. GitHub Cerca le opzioni più comuni che dispongono di aggiornamenti recenti e vengono mantenute attivamente.

Usa le variabili per la personalizzazione

Quando usi i moduli della community, passa gli input attraverso le variabili invece di creare un forking o modificare direttamente il codice sorgente. Sostituisci i valori predefiniti dove richiesto invece di modificare i componenti interni del modulo.

Il forking dovrebbe limitarsi a fornire correzioni o funzionalità al modulo originale a beneficio della comunità più ampia.

Comprendi le dipendenze

Prima di utilizzare il modulo, esamina il codice sorgente e la documentazione per identificare le dipendenze:

- Provider richiesti: prendi nota delle versioni di AWS Kubernetes o di altri provider richiesti dal modulo.
- Moduli annidati: verifica la presenza di altri moduli utilizzati internamente che introducono dipendenze a cascata.

- Fonti di dati esterne: annota le API, i plug-in personalizzati o le dipendenze dell'infrastruttura su cui si basa il modulo.

Mappando l'albero completo delle dipendenze dirette e indirette, è possibile evitare sorprese quando si utilizza il modulo.

Utilizza fonti attendibili

L'approvvigionamento di moduli Terraform da editori non verificati o sconosciuti comporta rischi significativi. Usa i moduli solo da fonti attendibili.

- Preferisci i moduli certificati del Terraform Registry pubblicati da creatori verificati come i nostri partner AWS . HashiCorp
- Per i moduli personalizzati, consulta la cronologia degli editori, i livelli di supporto e la reputazione di utilizzo, anche se il modulo proviene dalla tua organizzazione.

Impedendo l'uso di moduli provenienti da fonti sconosciute o non controllate, è possibile ridurre il rischio di inserire vulnerabilità o problemi di manutenzione nel codice.

Sottoscrizione alle notifiche di

Iscriviti alle notifiche relative alle nuove versioni di moduli inviate da editori affidabili:

- Guarda gli archivi dei GitHub moduli per ricevere avvisi sulle nuove versioni del modulo.
- Monitora i blog e i log delle modifiche degli editori per verificare la presenza di aggiornamenti.
- Ricevi notifiche proattive per le nuove versioni da fonti verificate e altamente apprezzate invece di inserire implicitamente gli aggiornamenti.

Il consumo di moduli solo da fonti attendibili e il monitoraggio delle modifiche garantiscono stabilità e sicurezza. I moduli controllati migliorano la produttività riducendo al minimo il rischio della catena di fornitura.

Contribuisci ai moduli della community

Invia correzioni e miglioramenti per i moduli della community ospitati in: GitHub

- Apri le pull request sui moduli per risolvere i difetti o le limitazioni riscontrati durante l'utilizzo.

- Richiedi nuove configurazioni basate sulle best practice da aggiungere ai moduli OSS esistenti creando problemi.

Contribuire ai moduli della community migliora i modelli riutilizzabili e codificati per tutti i professionisti di Terraform.

Domande frequenti

D: Perché concentrarsi sul fornitore? AWS

R. Il AWS Provider è uno dei provider più utilizzati e complessi per il provisioning dell'infrastruttura in Terraform. Seguire queste migliori pratiche aiuta gli utenti a ottimizzare l'utilizzo del provider per l' AWS ambiente.

D: Sono nuovo su Terraform. Posso usare questa guida?

R. La guida è per le persone che non conoscono Terraform così come per i professionisti più avanzati che vogliono migliorare le proprie competenze. Le pratiche migliorano i flussi di lavoro per gli utenti in qualsiasi fase dell'apprendimento.

D: Quali sono alcune best practice chiave trattate?

R. Le migliori pratiche chiave includono l'utilizzo dei ruoli IAM rispetto alle chiavi di accesso, il pinning delle versioni, l'integrazione di test automatizzati, il blocco remoto dello stato, la rotazione delle credenziali, il contributo ai provider e l'organizzazione logica delle basi di codice.

D: Dove posso trovare ulteriori informazioni su Terraform?

R. La sezione Risorse include collegamenti alla documentazione ufficiale di HashiCorp Terraform e ai forum della community. Utilizza i link per saperne di più sui flussi di lavoro Terraform avanzati.

Passaggi successivi

Ecco alcuni potenziali passaggi successivi dopo aver letto questa guida:

- Se disponi di una base di codice Terraform esistente, rivedi la configurazione e identifica le aree che potrebbero essere migliorate in base ai consigli forniti in questa guida. Ad esempio, esamina le migliori pratiche per l'implementazione di backend remoti, la separazione del codice in moduli, l'utilizzo del pinning delle versioni e così via, e convalidale nella configurazione.
- Se non disponi di una base di codice Terraform esistente, utilizza queste best practice per strutturare la nuova configurazione. Segui i consigli sulla gestione dello stato, l'autenticazione, la struttura del codice e così via fin dall'inizio.
- Prova a utilizzare alcuni dei moduli della HashiCorp community citati in questa guida per vedere se semplificano i tuoi modelli di architettura. I moduli consentono livelli di astrazione più elevati, quindi non è necessario riscrivere risorse comuni.
- Abilita linting, scansioni di sicurezza, controlli delle policy e strumenti di test automatizzati per rafforzare alcune delle migliori pratiche in materia di sicurezza, conformità e qualità del codice. Strumenti come TFlint, tfsec e Checkov possono aiutarti.
- Consulta la documentazione più recente del AWS Provider per vedere se ci sono nuove risorse o funzionalità che potrebbero aiutarti a ottimizzare l'utilizzo di Terraform. Rimani aggiornato sulle nuove versioni del AWS Provider.
- Per ulteriori indicazioni, consulta la documentazione Terraform, la guida alle migliori pratiche e la guida di stile sul HashiCorp sito Web.

Risorse

Riferimenti

I seguenti collegamenti forniscono materiale di lettura aggiuntivo per Terraform AWS Provider e per l'utilizzo di Terraform for IAc su. AWS

- Terraform Provider AWS (documentazione) HashiCorp
- Moduli Terraform per AWS servizi (Terraform Registry)
- The AWS and HashiCorp Partnership (HashiCorp post sul blog)
- Credenziali dinamiche con il AWS provider (documentazione HCP Terraform)
- DynamoDB State Locking (documentazione Terraform)
- Applica la politica con Sentinel (documentazione Terraform)

Strumenti

I seguenti strumenti aiutano a migliorare la qualità del codice e l'automazione delle configurazioni Terraform su AWS, come consigliato in questa guida alle migliori pratiche.

Qualità del codice:

- Checkov: esegue la scansione del codice Terraform per identificare le configurazioni errate prima della distribuzione.
- TFlint: identifica possibili errori, sintassi obsoleta e dichiarazioni non utilizzate. Questo linter può anche applicare le migliori pratiche e le convenzioni di denominazione. AWS
- terraform-docs: genera documentazione dai moduli Terraform in vari formati di output.

Strumenti di automazione:

- HCP Terraform: aiuta i team a modificare, collaborare e creare flussi di lavoro Terraform con controlli delle politiche e porte di approvazione.
- Atlantis: uno strumento di automazione delle pull request Terraform open source per la convalida delle modifiche al codice.

- CDK per Terraform: un framework che consente di utilizzare linguaggi familiari come TypeScript Python, Java, C# e Go anziché HashiCorp Configuration Language (HCL) per definire, fornire e testare l'infrastruttura Terraform come codice.

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	—	28 maggio 2024

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 AWS service 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, puoi utilizzarlo AWS CloudFormation per rilevare la deriva nelle risorse di sistema oppure puoi usarlo AWS Control Tower per rilevare cambiamenti nella tua 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

AWS services 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 il 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 utilizzare Regioni AWS il proprio 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. AWS service

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 AWS service. Puoi utilizzare l'endpoint per connetterti a livello di programmazione al servizio di destinazione. Per ulteriori informazioni, consulta Endpoint del AWS service 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.