Whitepaper AWS

Introduzione a DevOps in AWS

Introduzione a DevOps in AWS: Whitepaper AWS

Table of Contents

Introduzione a DevOps in AWS

Data di pubblicazione: 16 ottobre 2020 (Revisioni del documento)

Riassunto

Oggi più che mai, le aziende stanno intraprendendo il loro viaggio di trasformazione digitale al fine di costruire delle connessioni più profonde con i loro clienti e per ottenere un valore aziendale sostenibile e duraturo. Le organizzazioni di tutte le forme e dimensioni stanno superando i loro concorrenti e stanno entrando in nuovi mercati innovando più rapidamente che mai. Per queste organizzazioni, è importante concentrarsi sull'innovazione e sulla trasformazione del software, rendendo fondamentale semplificare la distribuzione del software. Le organizzazioni che riducono il tempo necessario dall'ideazione alla produzione considerano la velocità e l'agilità delle priorità, e questo favorisce l’innovazione.

Sebbene ci siano diversi fattori da considerare per diventare dei grandi innovatori, questo white paper si concentra su DevOps e sui servizi e le funzionalità della piattaforma AWS che contribuiranno ad aumentare la capacità di un'organizzazione di fornire applicazioni e servizi ad alta velocità.

Introduzione

DevOps è la combinazione di pratiche e di modelli culturali, ingegneristici e strumenti che aumentano la capacità di un'organizzazione di fornire applicazioni e servizi in modo rapido e con una migliore qualità. Nel corso del tempo, sono emerse diverse pratiche essenziali relative all'adozione del modello DevOps: integrazione continua, distribuzione continua, Infrastructure as Code, nonché monitoraggio e registrazione.

Questo documento evidenzia le funzionalità di AWS che ti aiutano ad accelerare il tuo percorso DevOps e illustra come i servizi AWS possono aiutare a rimuovere il pesante carico di lavoro indifferenziato associato all'adozione del modello DevOps. Illustriamo anche come creare una capacità di integrazione e distribuzione continua senza gestire server o creare nodi e come sfruttare l'approccio Infrastructure as Code per eseguire il provisioning e gestire le risorse cloud in modo coerente e ripetibile.

- Integrazione continua: è un metodo di sviluppo software in cui gli sviluppatori aggiungono regolarmente modifiche al codice in un repository centralizzato, quindi la creazione di build e i test vengono eseguiti automaticamente.
- Distribuzione continua: è un metodo di sviluppo software in cui le modifiche al codice vengono applicate a una build, testate e preparate per il rilascio in produzione in modo automatico.
- Infrastructure as Code: è una prassi secondo cui provisioning e gestione dell'infrastruttura avvengono tramite metodologie di sviluppo di software e codice quali il controllo di versione e l'integrazione continua.
- Monitoraggio e registrazione: consente alle aziende di tenere sotto controllo i parametri e i log per scoprire in che modo le prestazioni di applicazione e infrastruttura influiscano sull'esperienza dell'utente finale.
- Comunicazione e collaborazione: vengono stabilite pratiche per avvicinare i team, creando flussi di lavoro e distribuendo le responsabilità relative alla strategia DevOps.
- Sicurezza: dovrebbe essere una preoccupazione trasversale. Le pipeline di integrazione continua e distribuzione continua (CI/CD) e i servizi correlati devono essere salvaguardati e devono essere impostate delle autorizzazioni adeguate per il controllo degli accessi.

L'analisi di ciascuno di questi principi rivela una stretta connessione con le offerte di Amazon Web Services (AWS).

Integrazione continua

L'integrazione continua (CI) è una pratica di sviluppo software in cui gli sviluppatori uniscono regolarmente le modifiche al codice in un repository centrale, su cui vengono applicati test automatici e build. L'integrazione continua aiuta a individuare e risolvere i bug con maggiore tempestività, migliorare la qualità del software e ridurre il tempo richiesto per convalidare e pubblicare nuovi aggiornamenti.

AWS offre i seguenti servizi per l'integrazione continua:

Argomenti

- AWS CodeCommit
- AWS CodeBuild
- AWS CodeArtifact

AWS CodeCommit

AWS CodeCommit è un servizio gestito altamente scalabile e sicuro di controllo del codice sorgente che consente l'hosting di repository Git privati. CodeCommit elimina la necessità di utilizzare il proprio sistema di controllo del codice sorgente e non vi è alcun hardware da fornire e scalare o software da installare, configurare e utilizzare. CodeCommit può essere usato per memorizzare molti elementi diversi, dal codice sorgente al codice binario, e supporta le funzionalità standard di Git, così può essere impiegato in modo ottimale insieme agli strumenti basati su Git già in uso. Il tuo team può anche utilizzare gli strumenti online di CodeCommit per sfogliare, modificare e collaborare ai progetti. AWS CodeCommit presenta diversi vantaggi:

Collaborazione- AWS CodeCommit è progettato per lo sviluppo di software collaborativo. Consente di eseguire commit, branch e merge di codice conservando facilmente il controllo dei progetti del team. Inoltre, CodeCommit supporta le richieste pull, che offrono un meccanismo per richiedere verifiche del codice e avviare conversazioni con i collaboratori.

Crittografia- Puoi trasferire i file da e verso AWS CodeCommit tramite HTTPS o SSH, secondo le tue preferenze. I tuoi repository vengono inoltre crittografati automaticamente a riposo mediante AWS Key Management Service (AWS KMS) utilizzando chiavi specifiche del cliente.

Controllo degli accessi- AWS CodeCommit utilizzaAWS Identity and Access Management (IAM) per controllare e monitorare chi, quando e dove può accedere ai dati. CodeCommit ti aiuta anche a monitorare i tuoi repository tramite AWS CloudTrail e Amazon CloudWatch.

Elevata disponibilità e durata- AWS CodeCommit archivia i repository in Amazon Simple Storage Service (Amazon S3) e Amazon DynamoDB. I dati, crittografati, sono memorizzati in modo ridondante su più strutture. Questa architettura aumenta la disponibilità e la durabilità dei dati nei repository.

Notifiche e script personalizzati- Ora puoi ricevere notifiche per eventi che hanno un impatto sui tuoi repository. Le notifiche arriveranno come notifiche di Amazon Simple Notification Service (Amazon SNS). Ogni notifica includerà un messaggio sullo stato e un collegamento alle risorse il cui evento ha generato la notifica in questione. Inoltre, utilizzando i trigger di repository di AWS CodeCommit puoi inviare notifiche e creare webhook HTTP con Amazon SNS o richiamare funzioni AWS Lambda in risposta a eventi di repository specifici.

AWS CodeBuild

AWS CodeBuild è un servizio di integrazione continua completamente gestito che permette di compilare codice sorgente, eseguire test e preparare pacchetti software pronti per essere distribuiti. Non dovi eseguire il provisioning, gestire e scalare i tuoi server di compilazione. CodeBuild può utilizzare GitHub, GitHub Enterprise, BitBucket, AWS CodeCommit oppure Amazon S3 come provider di origine.

CodeBuild è scalabile continuamente e può elaborare più build contemporaneamente. CodeBuild offre diversi ambienti preconfigurati per varie versioni di Microsoft Windows e Linux. I clienti possono anche portare i loro ambienti di compilazione personalizzati come container Docker. CodeBuild si integra inoltre con strumenti open source come Jenkins e Spinnaker.

CodeBuild può anche creare report per test unitari, funzionali o di integrazione. Questi rapporti forniscono una visione generale di quanti casi di test sono stati eseguiti e quanti sono stati superati o falliti. Il processo di creazione può essere eseguito anche all'interno di un Amazon Virtual Private Cloud (Amazon VPC), che può essere utile se i servizi di integrazione o i database vengono distribuiti all'interno di un VPC.

AWS CodeArtifact

AWS CodeArtifact è un servizio di repository di artefatti completamente gestito che permette alle organizzazioni di conservare, pubblicare e condividere in modo sicuro i pacchetti software

utilizzati nel loro processo di sviluppo software. Puoi configurare CodeArtifact per recuperare automaticamente dipendenze e pacchetti software da repository di artefatti pubblici permettendo agli sviluppatori di accedere alle versioni più recenti.

I team di sviluppo software si affidano sempre più a pacchetti open source per eseguire attività comuni nel loro pacchetto di applicazioni. Per i team di sviluppo software è ora diventato fondamentale mantenere il controllo su una particolare versione del software open source affinché questa sia priva di vulnerabilità. Con CodeArtifact, puoi impostare controlli per fare quanto sopra.

CodeArtifact funziona con i gestori di pacchetti più diffusi e strumenti di compilazione, tra cui Maven, Gradle, npm, yarn, twine e pip, e semplifica l'integrazione con flussi di lavoro di sviluppo esistenti.

Distribuzione continua

La distribuzione continua è un metodo di sviluppo software in cui le modifiche al codice vengono preparate automaticamente per un rilascio in produzione. Fondamento dello sviluppo moderno di applicazioni, la distribuzione continua estende l'integrazione continua distribuendo tutte le modifiche al codice all'ambiente di testing e/o di produzione dopo la fase di creazione di build. Se è implementata correttamente, gli sviluppatori avranno sempre a disposizione un artefatto di build pronto per la distribuzione che ha già superato un processo di testing standardizzato.

La distribuzione continua consente agli sviluppatori di automatizzare il testing oltre gli unit test, in modo da verificare l'applicazione degli aggiornamenti su vari livelli prima di renderli disponibili ai clienti. Queste prove possono includere test dell'interfaccia, test di caricamento, test di integrazione, test di affidabilità delle API e così via. Così è più semplice per gli sviluppatori analizzare gli aggiornamenti più approfonditamente e rilevare preventivamente eventuali problemi. Grazie al cloud, automatizzare la creazione e la replica di più di un ambiente a scopo di testing è un'operazione molto semplice e poco costosa, al contrario di quanto avviene in ambienti locali.

AWS offre i seguenti servizi per la distribuzione continua:

- AWS CodeBuild
- AWS CodeDeploy
- AWS CodePipeline

Argomenti

- AWS CodeDeploy
- AWS CodePipeline

AWS CodeDeploy

AWS CodeDeploy è un servizio di distribuzione completamente gestito che automatizza le distribuzioni di software su una vasta gamma di servizi di elaborazione come Amazon Elastic Compute Cloud (Amazon EC2), AWS Fargate, AWS Lambda e i server locali. AWS CodeDeploy semplifica il rilascio rapido di nuove funzionalità, aiuta a evitare i tempi di inattività durante la distribuzione delle applicazioni e semplifica la gestione della complessità dell'aggiornamento delle applicazioni. Puoi usare CodeDeploy per automatizzare le distribuzioni di software, eliminando

l'esigenza di operazioni manuali soggette a errore. Il servizio è scalabile in funzione delle esigenze di distribuzione.

CodeDeploy offre diversi benefici che sono conformi con il principio DevOps della distribuzione continua:

Implementazione automatizzata: CodeDeploy automatizza le implementazioni di software, consentendo operazioni affidabili e rapide.

Controllo centralizzato: CodeDeploy consente di avviare a tenere traccia dello stato delle implementazioni di un'applicazione con la massima semplicità tramite la Console di gestione AWS o l'interfaccia a riga di comando. CodeDeploy fornisce un rapporto dettagliato che consente di vedere quando e dove è stata distribuita ciascuna versione dell'applicazione. Puoi anche configurare notifiche push che inviino aggiornamenti immediati sulle implementazioni.

Tempo di inattività minimo: CodeDeploy permette di ottimizzare la disponibilità dell'applicazione durante il processo di implementazione software. Per ottenere questo risultato, applica le modifiche in modo incrementale e tiene traccia dell'integrità delle applicazioni in base a regole personalizzabili. Le implementazioni software possono essere interrotte con semplicità e, in caso di errori, è possibile ripristinare la versione precedente del software.

Facilità di adozione: CodeDeploy funziona con qualsiasi applicazione e offre la stessa esperienza su piattaforme e linguaggi diversi. Il codice di configurazione esistente può anche essere riutilizzato. CodeDeploy può anche essere integrato con il processo di rilascio di software o con la toolchain di distribuzione continua esistenti (p.es AWS CodePipeline, GitHub, Jenkins).

AWS CodeDeploy supporta diverse opzioni di implementazione. Per ulteriori informazioni, consulta Strategie di implementazione.

AWS CodePipeline

AWS CodePipeline è un servizio di distribuzione continua che consente di modellare, visualizzare e automatizzare le fasi necessarie al rilascio di software. Con AWS CodePipeline, puoi modellare l'intero processo di rilascio per la creazione del codice, la distribuzione in ambienti di pre-produzione, il test dell'applicazione e il rilascio in produzione. AWS CodePipeline quindi crea, testa e distribuisce l'applicazione in base al flusso di lavoro definito ogni volta che si verifica una modifica del codice. Puoi integrare strumenti di partner e strumenti personalizzati in qualunque fase del processo, per dare vita a una soluzione di distribuzione continua completa.

AWS CodePipeline presenta diversi vantaggi che si allineano al principio DevOps della distribuzione continua:

Distribuzione rapida: AWS CodePipeline automatizza il processo di rilascio di software, consentendo di pubblicare più rapidamente nuove caratteristiche. Grazie alla rapidità del servizio, sarà possibile ascoltare il feedback degli utenti e implementare nuove funzioni a tempo record.

Qualità migliorata: AWS CodePipeline, automatizzando i processi di creazione di build, test e rilascio, permette di migliorare frequenza e qualità degli aggiornamenti software apportando nuove modifiche tramite un set uniforme di controlli di qualità.

Facilità di integrazione: AWS CodePipeline può essere esteso facilmente per adattarsi ai bisogni specifici. Puoi utilizzare i plug-in predefiniti o aggiungerne di personalizzati in ogni fase del processo. Ad esempio, puoi estrarre il codice sorgente da GitHub, impiegare un server di build Jenkins locale, eseguire test di carico con servizi di terze parti o trasferire le informazione di distribuzione in un pannello di controllo operativo personalizzato.

Flusso di lavoro configurabile: AWS CodePipeline consente di modellare le diverse fasi del processo di rilascio del software utilizzando l'interfaccia della console, l'interfaccia a riga di comando di AWS, AWS CloudFormation oppure gli SDK AWS. Puoi specificare i test da eseguire e personalizzare le fasi di distribuzione dell'applicazione e le relative dipendenze.

Strategie di distribuzione

Le strategie di distribuzione definiscono le modalità di distribuzione del software. Le organizzazioni seguono diverse strategie in merito alla distribuzione in base al proprio modello di business. Alcune organizzazioni potrebbero scegliere di fornire software completamente testato, mentre altre potrebbero desiderare che i propri utenti forniscano feedback e potrebbero consentire agli utenti di valutare le funzionalità in fase di sviluppo (ad esempio, versioni beta). Nella sezione seguente parleremo di varie strategie di distribuzione.

Argomenti

Distribuzioni locali

In questa strategia, la distribuzione viene eseguita con l'applicazione su ogni istanza nel gruppo di distribuzione. Viene installata la versione più recente della revisione dell'applicazione e viene avviata e convalidata la nuova versione dell'applicazione. Puoi utilizzare un sistema di bilanciamento del carico in modo che la registrazione di ciascuna istanza venga annullata durante la relativa distribuzione e, al completamento della distribuzione, l'istanza venga rimessa in servizio. Le distribuzioni sul posto possono essere eseguite tutte in una volta, presupponendo un'interruzione del servizio o possono essere eseguite come aggiornamento continuo. AWS CodeDeploy e AWS Elastic Beanstalk offrono configurazioni di distribuzione una alla volta, la metà alla volta e tutte in una volta. Queste stesse strategie di distribuzione per le distribuzioni sul posto sono disponibili nelle distribuzioni blu/verdi.

Distribuzioni blu/verde

La distribuzione blu/verde, a volte indicata come distribuzione rosso-nero, è una tecnica per il rilascio di applicazioni spostando il traffico tra due ambienti identici che eseguono versioni diverse dell'applicazione. Le distribuzioni blu/verde consentono di ridurre al minimo i tempi di inattività durante gli aggiornamenti delle applicazioni, mitigando così i rischi correlati ai tempi di inattività Le

distribuzioni blu/verde consentono di avviare una nuova versione (verde) dell'applicazione insieme alla versione precedente (blu) e di monitorare e testare la nuova versione prima di reindirizzare il traffico verso di essa, e consentono il ripristino in caso di problemi.

Distribuzioni Canary

Il traffico viene trasferito in due incrementi. Una distribuzione canary è una strategia blu/verde più avversa al rischio, in cui viene utilizzato un approccio di tipo graduale. Questa distribuzione può essere in due fasi o lineare in cui il nuovo codice dell'applicazione viene distribuito ed esposto per la prova e, dopo l'accettazione, viene distribuito al resto dell'ambiente o in modo lineare.

Distribuzioni lineari

Nelle distribuzioni lineari il traffico viene trasferito in incrementi uguali con lo stesso intervallo di tempo, in minuti, tra ciascun incremento. Puoi scegliere tra opzioni lineari predefinite che specificano la percentuale del traffico trasferito in ogni incremento e l'intervallo di tempo, in minuti, tra ciascun incremento.

Distribuzioni all-at-once

Le distribuzioni all-at-once significano che tutto il traffico viene trasferito dall'ambiente originale all'ambiente di sostituzione tutto in una volta.

Matrice delle strategie di distribuzione

La seguente matrice elenca le strategie di distribuzione supportate per Amazon Elastic Container Service (Amazon ECS), AWS Lambda e Amazon EC2/On-premise.

- Amazon ECS è un servizio di orchestrazione completamente gestito.
- AWS Lambda ti consente di eseguire il codice senza effettuare il provisioning dei server o senza gestirli.
- Amazon EC2 ti consente di avere una capacità di elaborazione sicura e ridimensionabile nel cloud.

	A	B	C	D
1	Matrice delle strategie di distribuzione	Amazon ECS	AWS Lambda	Amazon EC2/ On-Premise
2	Locale	✓	✓	✓
3	Blu/Verde	✓	✓	✓
4	Canary	✓	✓	X
5	Linear (Lineare)	✓	✓	X
6	All-at-once	✓	✓	X

> **Note**
>
> La distribuzione blu/verde con EC2/On-premise funziona solo con le istanze EC2.

AWS Elastic Beanstalk Strategie di distribuzione

AWS Elastic Beanstalk supporta i seguenti tipi di strategie di distribuzione:

- All-at-Once: esegue la distribuzione sul posto su tutte le istanze.

- Graduale: divide le istanze in batch e le distribuisce in un batch alla volta.
- Graduale con batch aggiuntivo: divide le distribuzioni in batch, ma per il primo batch crea nuove istanze EC2 invece di distribuirle sulle istanze EC2 esistenti.
- Immutabile: se è necessario eseguire la distribuzione con una nuova istanza invece di utilizzare un'istanza esistente.
- Suddivisione del traffico: esegue una distribuzione immutabile e quindi inoltra la percentuale di traffico alle nuove istanze per un periodo di tempo predeterminato. Se le istanze rimangono integre, inoltra tutto il traffico alle nuove istanze e chiude le vecchie istanze.

Infrastructure as Code

Uno dei principi fondamentali del modello DevOps consiste nel trattare l'infrastruttura nello stesso modo in cui gli sviluppatori trattano il codice. Il codice dell'applicazione ha un formato e una sintassi definiti. Se il codice non è scritto secondo le regole del linguaggio di programmazione, non è possibile creare delle applicazioni. Il codice viene memorizzato in un sistema di gestione delle versioni o di controllo del codice sorgente che registra una cronologia dello sviluppo del codice, delle modifiche e delle correzioni dei bug. Quando il codice viene compilato o integrato nelle applicazioni, ci aspettiamo che venga creata un'applicazione coerente e che la compilazione sia ripetibile e affidabile.

Adottare l'approccio Infrastructure as code significa applicare lo stesso rigore tipico dello sviluppo del codice dell'applicazione al provisioning dell'infrastruttura. Tutte le configurazioni devono essere definite in modo dichiarativo e memorizzate in un sistema di controllo del codice sorgente come AWS CodeCommit, lo stesso del codice dell'applicazione. Anche il provisioning, l'orchestrazione e la distribuzione dell'infrastruttura dovrebbero supportare l'adozione dell'approccio Infrastructure as code.

L'infrastruttura veniva tradizionalmente fornita utilizzando una combinazione di script e processi manuali. A volte questi script venivano memorizzati in sistemi di controllo delle versioni o documentati passo dopo passo in file di testo o run-book. Spesso la persona che scrive i run-book non è la stessa persona che esegue questi script o che segue i run-book. Se questi script o run-book non vengono aggiornati di frequente, possono diventare un fattore di blocco nelle distribuzioni. Ciò si traduce nella creazione di nuovi ambienti non sempre ripetibili, affidabili o coerenti.

A differenza dell'approccio precedente, AWS offre un modo incentrato su DevOps per creare e mantenere l'infrastruttura. In modo simile al modo in cui gli sviluppatori software scrivono i codici applicativi, AWS fornisce servizi che permettono la creazione, la distribuzione e il mantenimento dell'infrastruttura in modo programmatico, descrittivo e dichiarativo. Tali servizi offrono rigore, chiarezza e affidabilità. I vari servizi AWS menzionati in questo documento sono essenziali per una metodologia DevOps e costituiscono la base di numerosi principi e pratiche AWS DevOps di livello superiore.

AWS offre i seguenti servizi per definire l'approccio Infrastructure as code.

- AWS CloudFormation
- AWS Cloud Development Kit (AWS CDK)
- Kit di sviluppo di AWS Cloud per Kubernetes

AWS CloudFormation

AWS CloudFormation è un servizio che consente agli sviluppatori di creare delle risorse AWS in modo ordinato e prevedibile. Le risorse vengono scritte in file di testo utilizzando il formato JavaScript Object Notation (JSON) o Yet Another Markup Language (YAML). I modelli richiedono una sintassi e una struttura specifiche che dipendono dai tipi di risorse create e gestite. Puoi creare le tue risorse in JSON o YAML con qualsiasi editor di codice come AWS Cloud9, e poi puoi archiviarle in un sistema di controllo delle versioni e successivamente CloudFormation creerà i servizi specificati in modo sicuro e ripetibile.

Un modello di CloudFormation viene distribuito nell'ambiente AWS come stack. Puoi gestire gli stack tramite la Console di gestione AWS, l'interfaccia a riga di comando AWS o le API di AWS CloudFormation. Per apportare modifiche alle risorse in esecuzione in uno stack, puoi aggiornare lo stack. Prima di apportare modifiche alle risorse, puoi generare un set di modifiche, che è un riepilogo delle modifiche proposte. I set di modifiche ti consentono di vedere in che modo le modifiche possono influire sulle risorse in esecuzione, soprattutto nel caso di risorse critiche, prima di applicarle.

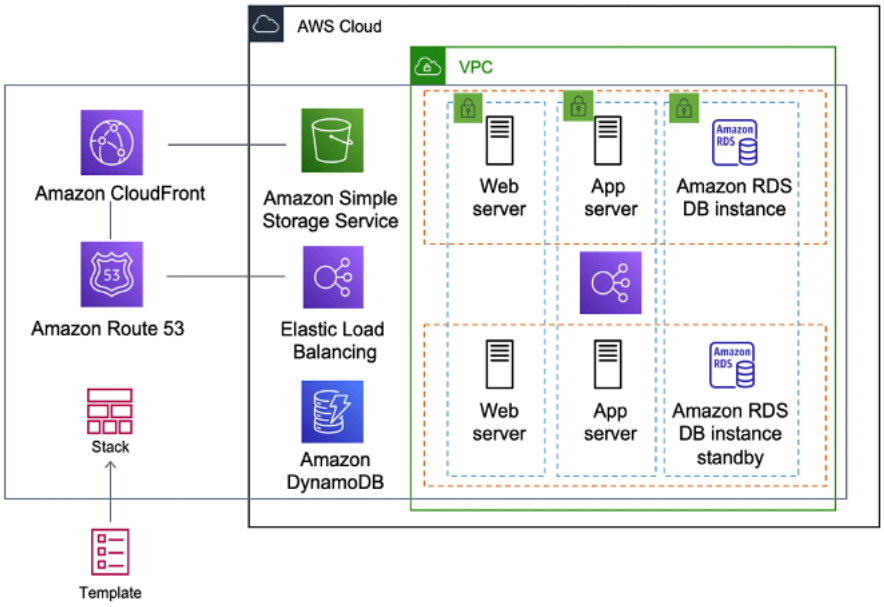

Figura 1 - AWS CloudFormation crea un intero ambiente (stack) partendo da un modello

Puoi utilizzare un singolo modello per creare e aggiornare un intero ambiente o modelli separati per gestire più livelli all'interno di un ambiente. Ciò consente di modulare i modelli e fornisce anche un livello di governance davvero importante per molte organizzazioni.

Quando crei o aggiorni uno stack nella console, vengono visualizzati eventi che mostrano lo stato della configurazione. Se si verifica un errore, per impostazione predefinita lo stack viene riportato

allo stato precedente. Amazon Simple Notification Service (Amazon SNS) fornisce notifiche sugli eventi. Ad esempio, puoi utilizzare Amazon SNS per monitorare l'avanzamento della creazione e dell'eliminazione dello stack tramite e-mail e integrare tali operazioni con altri processi a livello di programmazione.

AWS CloudFormation semplifica la gestione e la distribuzione di raccolte di risorse AWS, consentendo di descrivere dipendenze o passare parametri speciali quando lo stack è stato configurato.

Con i modelli CloudFormation, puoi lavorare con un'ampia gamma di servizi AWS, quali Amazon S3, Auto Scaling, Amazon CloudFront, Amazon DynamoDB, Amazon EC2, Amazon ElastiCache, AWS Elastic Beanstalk, Elastic Load Balancing, AWS OpsWorks, IAM e Amazon VPC. Per l'elenco più recente delle risorse supportate, consulta la pagina di riferimento sui tipi di proprietà e di risorse AWS.

AWS Cloud Development Kit

AWS Cloud Development Kit (AWS CDK) è un framework di sviluppo software open source che consente di modellare ed effettuare il provisioning delle risorse delle applicazioni cloud, utilizzando linguaggi di programmazione familiari. AWS CDK ti permette di modellare l'infrastruttura dell'applicazione utilizzando TypeScript, Python, Java e .NET. Gli sviluppatori possono sfruttare il loro ambiente di sviluppo integrato (IDE) esistente, sfruttando strumenti come il completamento automatico e la documentazione in linea al fine di accelerare lo sviluppo dell'infrastruttura.

AWS CDK utilizza AWS CloudFormation in background per eseguire il provisioning delle risorse in modo sicuro e ripetibile. I costrutti rappresentano gli elementi basilari del codice CDK. Un costrutto rappresenta un componente cloud e racchiude tutto quello di cui AWS CloudFormation ha bisogno per creare il componente. Il CDK di AWS include la libreria AWS Construct contenente costrutti che rappresentano molti servizi AWS. Combinando insieme i costrutti, puoi creare rapidamente e facilmente delle architetture complesse per la distribuzione in AWS.

Kit di sviluppo di AWS Cloud per Kubernetes

AWS Cloud Development Kit for Kubernetes (cdk8s) è un framework di sviluppo software open source per la definizione di applicazioni Kubernetes utilizzando linguaggi di programmazione generici.

Una volta definita l'applicazione in un linguaggio di programmazione (alla data di pubblicazione sono supportati solo i linguaggi Python e TypeScript), cdk8s convertirà la descrizione dell'applicazione

in YML prima di Kubernetes. Questo file YML può quindi essere utilizzato da qualsiasi cluster Kubernetes in esecuzione ovunque. Dal momento che la struttura è definita in un linguaggio di programmazione, puoi utilizzare le ricche funzionalità fornite dal linguaggio di programmazione. Puoi utilizzare la funzione di astrazione del linguaggio di programmazione per creare il codice e riutilizzarlo in tutte le distribuzioni.

Automazione

Una ulteriore filosofia e pratica essenziale di DevOps è l'automazione. L'automazione è incentrata su configurazione, distribuzione e supporto dell'infrastruttura e delle applicazioni che vengono eseguite sulla medesima infrastruttura. Utilizzando l'automazione, puoi configurare gli ambienti più rapidamente in modo standardizzato e ripetibile. La rimozione dei processi manuali è la chiave per una strategia DevOps di successo. Storicamente, la configurazione del server e la distribuzione delle applicazioni sono state prevalentemente un processo manuale. Così gli ambienti diventano non standard e riprodurre un ambiente quando sorgono problemi è davvero difficile.

L'uso dell'automazione è fondamentale per sfruttare appieno i vantaggi del cloud. Internamente, AWS fa molto affidamento sull'automazione al fine di fornire le caratteristiche principali di elasticità e scalabilità. I processi manuali sono soggetti a errori, sono inaffidabili e sono inadeguati a supportare un'azienda agile. Spesso un'organizzazione può impegnare risorse altamente qualificate per fornire la configurazione manuale, mentre quel tempo potrebbe essere speso meglio per supportare altre attività più critiche e di valore più elevato all'interno dell'azienda.

Gli ambienti operativi moderni si basano generalmente sulla completa automazione per eliminare l'intervento manuale o l'accesso agli ambienti di produzione. Questo include tutte le versioni del software, la configurazione della macchina, l'applicazione di patch al sistema operativo, la risoluzione dei problemi o la correzione Molti livelli di pratiche di automazione possono essere utilizzati insieme per fornire un processo automatizzato end-to-end di livello superiore.

L'automazione presenta i seguenti vantaggi chiave:

- Cambiamenti rapidi
- Maggiore produttività
- Configurazioni ripetibili
- Ambienti riproducibili
- Elasticità elevata
- Scalabilità automatica elevata
- Test automatizzati

L'automazione è un fattore essenziale dei servizi AWS ed è supportata internamente in tutti i servizi, le funzionalità e le offerte.

Argomenti

AWS OpsWorks

AWS OpsWorks porta i principi di DevOps anche oltre AWS Elastic Beanstalk. Può essere considerato un servizio di gestione delle applicazioni anziché un semplice container di applicazioni. AWS OpsWorks offre ancora più livelli di automazione con funzionalità aggiuntive come l'integrazione con il software di gestione della configurazione (Chef) e la gestione del ciclo di vita delle applicazioni. Puoi utilizzare la gestione del ciclo di vita delle applicazioni per definire quando le risorse vengono impostate, configurate, distribuite, non distribuite o chiuse.

Per una maggiore flessibilità, in AWS OpsWorks puoi definire la tua applicazione in stack configurabili. Inoltre, puoi selezionare stack di applicazioni predefiniti. Gli stack di applicazioni contengono tutto il provisioning per le risorse AWS richiesto dalla tua applicazione, inclusi server applicazioni, server Web, database e i sistemi di bilanciamento del carico.

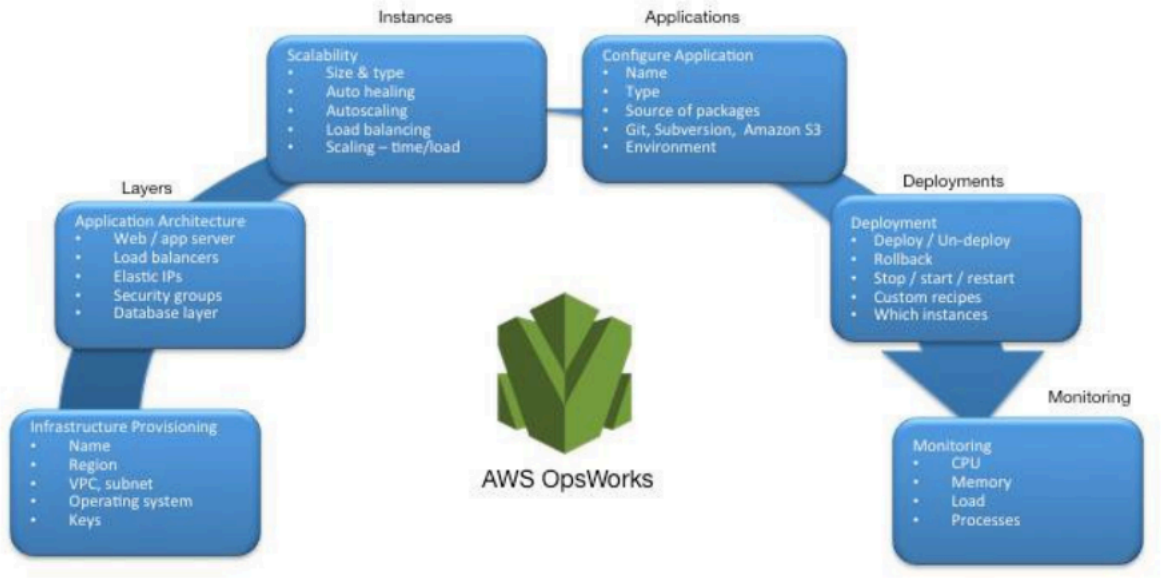

Figura 2 - AWS OpsWorks che mostra le caratteristiche e l'architettura DevOps

Gli stack di applicazioni sono organizzati in livelli dell'architettura in modo che gli stack possano essere mantenuti indipendentemente. I livelli di esempio possono includere il livello Web, il livello dell'applicazione e il livello del database. AWS OpsWorksInoltre, semplifica la configurazione dei

gruppi Auto Scaling e dei sistemi di bilanciamento del carico Elastic Load Balancing, illustrando ulteriormente il principio di automazione DevOps. Allo stesso modo di AWS Elastic Beanstalk, AWS OpsWorks supporta il controllo delle versioni delle applicazioni, la distribuzione continua e la gestione della configurazione dell'infrastruttura.

AWS OpsWorks supporta anche le pratiche DevOps di monitoraggio e registrazione (trattate nella prossima sezione). Il supporto per il monitoraggio è fornito da Amazon CloudWatch. Tutti gli eventi del ciclo di vita sono registrati e un registro Chef separato documenta tutte le ricette Chef eseguite, con eventuali eccezioni.

AWS Elastic Beanstalk

AWS Elastic Beanstalk è un servizio per distribuire e dimensionare applicazioni e servizi Web sviluppati con Java, .NET, PHP, Node.js, Python, Ruby, Go e Docker su server comuni come Apache, NGINX, Passenger e IIS.

Elastic Beanstalk è un'astrazione su Amazon EC2, Auto Scaling, e semplifica la distribuzione fornendo funzionalità aggiuntive quali clonazione, distribuzioni blu/verde, interfaccia a riga di comando di Elastic Beanstalk (eb cli) e integrazione con AWS Toolkit for Visual Studio, Visual Studio Code, Eclipse e IntelliJ per aumentare la produttività degli sviluppatori.

Monitoraggio e registrazione

La comunicazione e la collaborazione sono elementi fondamentali nel contesto di un approccio DevOps. Per poterle facilitare, è necessario ottenere feedback. In AWS, il feedback è fornito da due servizi principali: Amazon CloudWatch e AWS CloudTrail. Insieme, questi servizi forniscono una solida infrastruttura di monitoraggio, avvisi e auditing in modo che gli sviluppatori e i team operativi possano lavorare insieme in modo stretto e trasparente.

AWS fornisce i seguenti servizi per il monitoraggio e la registrazione:

Argomenti

Amazon CloudWatch

I parametri di Amazon CloudWatch raccolgono automaticamente dati dai servizi AWS come le istanze Amazon EC2, i volumi Amazon EBS e le istanze database di Amazon RDS. Questi parametri possono quindi essere organizzati come pannelli di controllo e possono essere creati allarmi o eventi per attivare eventi o eseguire azioni di Auto Scaling.

Allarmi di Amazon CloudWatch

Puoi impostare allarmi in base ai parametri raccolti da Amazon CloudWatch Metrics. L'allarme può quindi inviare una notifica all'argomento del servizio Amazon Simple Notification Service (Amazon SNS) oppure può avviare azioni di Auto Scaling. Un allarme richiede il periodo (il tempo necessario per valutare un parametro), il periodo di valutazione (numero dei punti dati più recenti) e i punti dati per l'allarme (numero di punti dati all'interno del periodo di valutazione).

Amazon CloudWatch Logs

Amazon CloudWatch Logs è un servizio di aggregazione e monitoraggio dei log. AWS CodeBuild, CodeCommit, CodeDeploy e CodePipeline forniscono integrazioni con i log di CloudWatch affinché tutti i log possano essere monitorati centralmente. Inoltre, altri servizi AWS forniscono l'integrazione diretta con CloudWatch.

Con CloudWatch Logs puoi:

- Eseguire query sui dati
- Monitorare i log delle istanze Amazon EC2
- Monitorare gli eventi registrati in AWS CloudTrail
- Definire la policy di conservazione

Amazon CloudWatch Logs Insights

Amazon CloudWatch Logs Insights esegue la scansione dei log e ti consente di eseguire query e visualizzazioni interattive. Comprende vari formati di registro e rileva automaticamente i campi dai registri JSON.

Amazon CloudWatch Events

Amazon CloudWatch Events eroga un flusso quasi in tempo reale di eventi di sistema che descrivono le modifiche alle risorse AWS. Utilizzando semplici regole che puoi impostare rapidamente, puoi abbinare eventi e instradarli verso una o più funzioni o flussi del target. CloudWatch Events si accorge di modifiche operative appena si verificano. Events CloudWatch risponde a questi cambiamenti operativi e prende le necessarie misure correttive inviando messaggi per rispondere all'ambiente, attivando funzioni, facendo modifiche, e catturando informazioni sullo stato.

Puoi configurare regole in CloudWatch Events per ricevere avvisi delle modifiche nei servizi AWS e puoi anche integrare questi eventi con altri sistemi di terze parti utilizzando Amazon EventBridge. Di seguito sono riportati i servizi correlati ad AWS DevOps che sono integrati con CloudWatch Events.

- Eventi Application Auto Scaling
- Eventi CodeBuild
- Eventi CodeCommit

- Eventi CodeDeploy
- Eventi CodePipeline

Amazon EventBridge

Amazon CloudWatch Events ed EventBridge sono lo stesso servizio e la stessa API di base, tuttavia EventBridge fornisce più funzionalità.

Amazon EventBridge è un bus di eventi serverless che consente l'integrazione tra servizi AWS, Software come servizio (SaaS) e le tue applicazioni. Oltre per creare applicazioni basate su eventi, EventBridge può essere utilizzato per notificare gli eventi dai servizi come CodeBuild, CodeDeploy, CodePipeline e CodeCommit.

AWS CloudTrail

Per adottare davvero i principi DevOps di collaborazione, comunicazione e trasparenza, è importante capire chi sta apportando modifiche all'infrastruttura. In AWS questa trasparenza e visibilità è fornita dal servizio AWS CloudTrail. Tutte le interazioni AWS vengono gestite tramite chiamate API AWS monitorate e registrate da AWS CloudTrail. Tutti i file di log generati vengono archiviati in un bucket Amazon S3 definito dall'utente. I file di log sono crittografati mediante la crittografia lato server (SSE) di Amazon S3. Tutte le chiamate API vengono registrate indipendentemente dal fatto che provengano direttamente da un utente o per conto di un utente da un servizio AWS. Numerosi gruppi possono trarre grande vantaggio dai log di CloudTrail, inclusi i team operativi per il supporto, i team di sicurezza per la governance e i team finanziari per la fatturazione.

Comunicazione e collaborazione

Sia che tu stia adottando la cultura DevOps nella tua organizzazione o che tu stia attraversando una trasformazione culturale con l'adozione del modello DevOps, la comunicazione e la collaborazione sono sempre un aspetto essenziale del tuo approccio. In Amazon, ci siamo resi conto che è necessario apportare un cambiamento in relazione alla mentalità dei team e quindi abbiamo adottato il concetto dei Team da due pizze.

Argomenti

I team da due pizze

"Cerchiamo di creare team abbastanza piccoli da essere sfamati con due pizze", ha affermato Bezos. "La chiamiamo la regola del team da due pizze".

Più piccolo è il team, migliore è la collaborazione. Anche la collaborazione è davvero importante dato che i rilasci di software si stanno muovendo più velocemente che mai. E la capacità di un team di fornire il software può essere un fattore di differenziazione per la tua organizzazione rispetto alla concorrenza. Immagina una situazione in cui bisogna rilasciare una nuova funzionalità del prodotto o correggere un bug. In questi casi vogliamo che ciò avvenga il più rapidamente possibile in modo da arrivare subito sul mercato. Questo è importante anche perché non vogliamo che la trasformazione sia un processo lento piuttosto che agile, in cui ondate di cambiamenti iniziano ad avere un impatto.

Anche la comunicazione tra i team è importante: ci muoviamo verso il modello di responsabilità condivisa e abbandoniamo l'approccio di sviluppo a compartimenti stagni. Quindi il concetto stesso di proprietà nel team deve essere visto in modo integrato e complessivo, dall'inizio alla fine. Il tuo team non dovrebbe vedere gli ambienti di produzione come compartimenti stagni che non comunicano tra loro.

Anche la trasformazione culturale è importante in quanto potresti creare un team DevOps comune o potresti avere dei membri del team che si focalizzano sul modello DevOps. Entrambi questi approcci introducono una responsabilità condivisa nell'ambito del team.

Sicurezza

Sia che tu stia attraversando una trasformazione con l'adozione del modello DevOps o implementando i principi DevOps per la prima volta, dovresti pensare alla sicurezza come integrata nei tuoi processi DevOps. Questa dovrebbe essere una preoccupazione trasversale nelle fasi di compilazione e distribuzione dei test.

Prima di parlare di sicurezza in DevOps su AWS, vediamo il modello di responsabilità condivisa AWS.

Argomenti

Modello di responsabilità condivisa AWS

La sicurezza è una responsabilità condivisa tra AWS e il cliente. Le diverse parti del modello di responsabilità condivisa sono spiegate di seguito:

- Responsabilità di AWS "Sicurezza del cloud"- AWS si occupa di proteggere l'infrastruttura globale su cui vengono eseguiti tutti i servizi offerti in AWS Cloud. Questa infrastruttura è composta da hardware, software, reti e strutture che eseguono i servizi AWS Cloud.
- Responsabilità del cliente "Sicurezza nel cloud"- La responsabilità del cliente verrà determinata dai servizi AWS Cloud scelti da un cliente. Ciò determina la quantità di lavoro di configurazione che il cliente deve eseguire nell'ambito delle proprie responsabilità di sicurezza.

Il modello condiviso può contribuire a ridurre l'onere operativo del cliente, dato che AWS agisce, gestisce e controlla tutti i componenti, dal sistema operativo host e il layer di virtualizzazione fino alla sicurezza fisica delle strutture in cui operano i servizi. Questo è fondamentale nei casi in cui il cliente desidera comprendere la sicurezza dei propri ambienti di sviluppo.

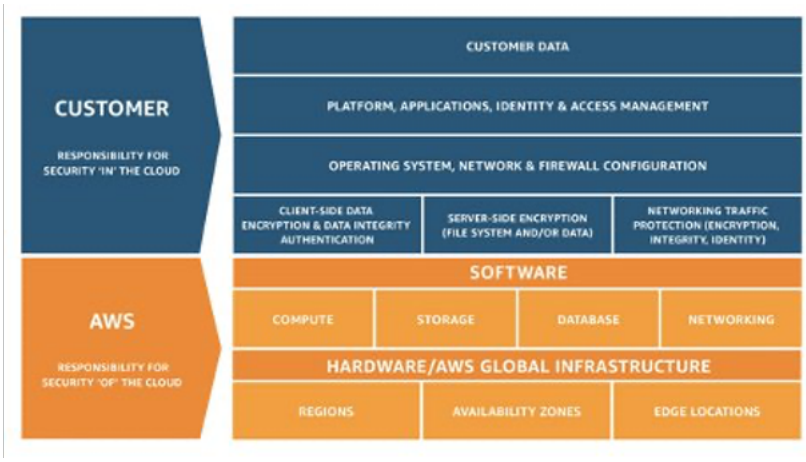

Figura 3 - Modello di responsabilità condivisa AWS

Identity and Access Management

AWS Identity and Access Management (IAM) definisce i controlli e le policy utilizzati per gestire l'accesso alle risorse AWS. Utilizzando il servizio IAM puoi creare utenti e gruppi e definire le autorizzazioni per vari servizi DevOps.

Oltre agli utenti, diversi servizi potrebbero aver bisogno di accedere alle risorse AWS. Ad esempio, il tuo progetto CodeBuild potrebbe aver bisogno di accedere per archiviare immagini Docker in Amazon Elastic Container Registry (Amazon ECR) e potrebbe richiedere autorizzazioni per scrivere in Amazon ECR. Questi tipi di autorizzazioni sono definiti da un ruolo di tipo speciale noto come ruolo di servizio.

IAM è un componente dell'infrastruttura di sicurezza AWS. Con IAM, puoi gestire centralmente gruppi, utenti, ruoli di servizio e credenziali di sicurezza come password, chiavi di accesso e policy di autorizzazione che controllano a quali servizi e risorse AWS possono accedere gli utenti. La policy IAM consente di definire il set di autorizzazioni. Questa policy può quindi essere allegata a un ruolo, a un utente o a un servizio per definire l'autorizzazione. Puoi utilizzare IAM per creare ruoli ampiamente utilizzati all'interno della strategia DevOps desiderata. In alcuni casi, può avere senso eseguire il comando AssumeRole a livello di programmazione invece di ottenere direttamente le autorizzazioni. Quando un servizio o un utente assume dei ruoli, gli vengono fornite credenziali temporanee per accedere a un servizio a cui normalmente non ha accesso.

Conclusione

Per rendere il percorso verso il cloud agevole, efficiente ed efficace, le aziende tecnologiche dovrebbero adottare i principi e le pratiche del modello DevOps. Questi principi sono integrati nella piattaforma AWS. In effetti, costituiscono la base di numerosi servizi AWS, in particolare quelli relativi alla distribuzione e al monitoraggio.

Inizia definendo la tua Infrastructure as Code utilizzando il servizio AWS CloudFormation oppure AWS Cloud Development Kit (AWS CDK). Successivamente, definisci il modo in cui le tue applicazioni utilizzeranno la distribuzione continua grazie all'aiuto di servizi come AWS CodeBuild, AWS CodeDeploy, AWS CodePipeline e AWS CodeCommit. A livello di applicazione, utilizza container come AWS Elastic Beanstalk, Amazon Elastic Container Service (Amazon ECS) oppure Amazon Elastic Kubernetes Service (Amazon EKS) e AWS OpsWorks per semplificare la configurazione di architetture comuni. L'utilizzo di questi servizi semplifica inoltre l'integrazione di altri servizi importanti come Auto Scaling ed Elastic Load Balancing. Infine, utilizza una perfetta strategia di monitoraggio DevOps con Amazon CloudWatch e solide pratiche di sicurezza con AWS IAM.

Con AWS come partner, i principi DevOps forniscono agilità alla tua azienda e all'organizzazione IT e accelerano il tuo percorso verso il cloud.

Revisioni del documento

Per ricevere una notifica sugli aggiornamenti di questo whitepaper, iscriviti al feed RSS.

update-history-change	update-history-description	update-history-date
Sezione Collaboratori mancanti ripristinata	Ripristinata la sezione Collaboratori mancante e piccole modifiche al testo	21 novembre 2020
Sezioni aggiornate per includere nuovi servizi	Sezioni aggiornate per includere nuovi servizi	16 ottobre 2020
Pubblicazione iniziale	Prima pubblicazione del whitepaper	1 dicembre 2014

Collaboratori

I collaboratori di questo documento includono:

- Muhammad Mansoor, Solutions Architect
- Ajit Zadgaonkar, World Wide Tech Leader, Modernization
- Juan Lamadrid - Solutions Architect
- Darren Ball - Solutions Architect
- Rajeswari Malladi - Solutions Architect
- Pallavi Nargund - Solutions Architect
- Bert Zahniser - Solutions Architect
- Abdullahi Olaoye – Cloud Solutions Architect
- Mohamed Kiswani – Software Development Manager
- Tara McCann – Manager Solutions Architect

Avvisi

I clienti sono responsabili della propria valutazione autonoma delle informazioni contenute in questo documento. Questo documento: (a) è solo a scopo informativo, (b) mostra le offerte e le pratiche attuali dei prodotti AWS soggette a modifiche senza preavviso, e (c) non crea alcun impegno o garanzia da parte di AWS e dei suoi affiliati, fornitori o licenziatari. I prodotti o servizi AWS sono forniti "così come sono" senza garanzie, dichiarazioni o condizioni di alcun tipo, sia esplicite che implicite. Le responsabilità e gli obblighi di AWS verso i propri clienti sono disciplinati dagli accordi AWS e il presente documento non fa parte né modifica alcun accordo tra AWS e i suoi clienti.