Chiudere uno o più Account AWS: come terminare in un click tutte le risorse attive ed evitare addebiti inaspettati

18 Agosto 2023 - 8 min. read

AWS CDK | AWS Organizations | Landing Zone

Sempre più aziende stanno scegliendo l'approccio Landing Zone per la gestione dei propri account AWS. Di conseguenza, il nostro lavoro richiede talvolta di operare su decine o, più raramente, centinaia di account AWS contemporaneamente, anche per un singolo progetto.

Dubbi su cosa sia una Landing Zone? Abbiamo pubblicato una serie sull'argomento!

Di recente, per esempio, è stata necessaria la creazione all'interno dei nostri account di un ambiente di test nel quale eseguire implementazioni di prova prima del rilascio all'interno dell'organization AWS del cliente. Questa necessità era dovuta alle dimensioni dell'organizzazione: quasi 400 account AWS! Di conseguenza, anche un piccolo errore avrebbe avuto un impatto notevole. Per replicare il loro ambiente, all'interno dei nostri account è stata creata una AWS Organization ad hoc con una decina di account e risorse accuratamente selezionate.

Al termine del lavoro, è stato poi necessario chiudere i suddetti account, ma la chiusura di più account AWS pieni di risorse può richiedere molto tempo.

Naturalmente, è proprio quando un compito richiede molto tempo che bisogna automatizzare il più possibile!

Chiusura di un account AWS

Per la chiusura di un account AWS può sembrare sufficiente premere il pulsante corretto, ma questa azione presenta una serie di accortezze da prendere. Alcune di esse dipendono dalla tipologia dell'account (indipendente o account membro di un'organizzazione AWS); altre sono comuni a entrambi i tipi di account.

Di seguito un piccolo riassunto dei punti più importanti a cui prestare attenzione:

- L'indirizzo email del root user non può essere riutilizzato dopo la chiusura dell'account.
- I domini registrati con Amazon Route 53 non vengono cancellati automaticamente; devono essere trasferiti a un altro registrar o account AWS, oppure si può disattivare il rinnovo automatico e lasciarli scadere.
- Se un account AWS viene riaperto durante il periodo di "post-closure", potrebbero essere addebitati i costi di tutti i servizi AWS che non sono stati interrotti o delle risorse che non sono state eliminate prima della chiusura.
- Dopo la chiusura di un account AWS, qualsiasi richiesta di accesso ai servizi AWS dell'account chiuso da parte di altri account AWS fallisce.
- AWS non elimina le peering connections di Amazon Virtual Private Cloud (Amazon VPC) quando un account che partecipa alla connessione viene chiuso.

Se l'account è membro di un'organizzazione AWS, ci sono ancora più punti che richiedono attenzione:

- L'account chiuso viene rimosso dall'organizzazione solo dopo il periodo di "post-closure".
- Solo il 10% degli account membri può essere chiuso in un periodo di 30 giorni.
- Nel caso in cui si utilizzi AWS Control Tower, è necessario che sia annullata la gestione degli account prima di tentare di chiuderli.

Pertanto, automatizzare il processo di chiusura di un account AWS non è un compito semplice. Se gli account sono autonomi, probabilmente significa che sono pochi, quindi la chiusura manuale è la scelta migliore, non vale nemmeno la pena di provare ad automatizzarla. Se invece fanno parte di un'organizzazione la storia cambia, ma in questo caso non è possibile chiuderne più del 10% al mese, quindi non è comunque molto efficiente.

La soluzione più efficiente a questo problema è probabilmente quella di riutilizzare gli account per un altro scopo invece di chiuderli, magari cambiando il loro alias. Questi account possono essere ripuliti, inseriti in un'unità organizzativa (OU) specifica e utilizzati per lo sviluppo o i test.

Eliminazione automatica delle risorse

Un'altra problematica comune a entrambi i tipi di account è:

"When you close your AWS account, you must terminate all your resources, or you might continue to incur charges”

ovvero, “quando chiudi il tuo account AWS, devi terminare tutte le tue risorse, altrimenti potresti continuare a subire addebiti". Questa non è certo una bella situazione!

Spieghiamo meglio questa parte: quando un account AWS viene chiuso, non è *veramente* chiuso; si trova nel periodo di “post-closure”, una finestra di 90 giorni in cui un utente può ancora accedere all'account, visualizzare le fatture passate, pagare le fatture AWS e... incorrere in spese.

Quando un account AWS viene chiuso, la fatturazione on-demand delle risorse si interrompe, ma alcune altre fonti di costo non vengono interrotte (ad esempio, gli abbonamenti ad AWS Marketplace). È quindi consigliabile eliminare tutte le risorse prima di chiudere l'account.

Cosa raccomanda AWS

AWS consiglia due modi per controllare le risorse attive:

- Attraverso la **Billing dashboard**: controllare nella sezione **Bills**.
- Utilizzare il **Tag Editor** all'interno della pagina **Resource Groups** nella console, selezionando "*Tutte le regioni*" e "*Tutti i tipi di risorse supportate*".

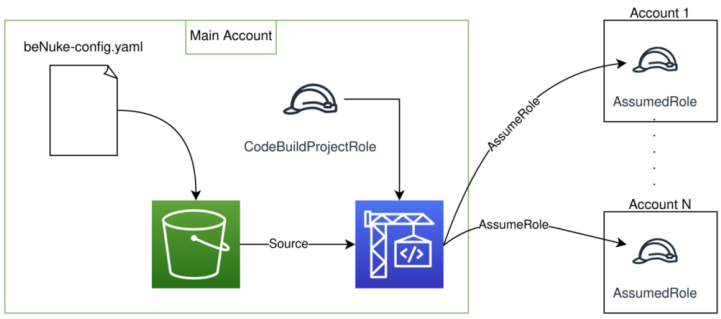
Queste sono entrambe soluzioni valide per trovare le risorse attive, ma non viene offerta alcuna soluzione per terminarle automaticamente.

beSharp-nuke

Per risolvere questo problema, abbiamo deciso di implementare una soluzione il più possibile minimale per cancellare tutte le risorse di uno o più account AWS.

Naturalmente, la maggior parte del lavoro è già stata svolta da aws-nuke, sviluppato da rebuy; si tratta di un fantastico strumento open-source in grado di eseguire il *nuke* (ovvero l'eliminazione di tutte le risorse) di un account AWS. Utilizzando un file di configurazione YAML è possibile filtrare alcune risorse da mantenere e creare una blocklist di account di produzione che non devono mai essere toccati. Purtroppo, però, questo software è stato progettato per essere eseguito su un singolo account AWS, ma a noi serviva una soluzione per più account!

La soluzione finale, riassunta nel diagramma sottostante, è piuttosto minimale: un modello CDK contenente solo un Amazon S3 Bucket per memorizzare la configurazione e AWS CodeBuild per l'esecuzione. Durante la fase di "build", viene analizzata la configurazione alla ricerca degli account di destinazione e viene eseguito aws-nuke, in sequenza, su ogni account.



Ora, perché scegliere AWS CodeBuild per la parte di "compute"? Non è stata una scelta semplice, ma è stato preferito per le sue esigenze minime in termini di risorse e per la sua natura *gestita*.

Ecco un riepilogo delle opzioni scartate:

- AWS Lambda: il timeout di 15 minuti non è adatto alla cancellazione di grandi account.
- Amazon EC2: richiede tempo per l'avvio/risveglio, inoltre ha bisogno di un VPC; la soluzione deve funzionare con requisiti minimi e non deve dipendere da risorse già esistenti.
- Amazon ECS: richiede anch'esso un VPC.
- AWS Step Function: non risolve il timeout di 15 minuti se usato insieme a Lambda in quanto l'esecuzione di aws-nuke non può essere suddivisa in più funzioni. Se usato

solo come orchestratore è probabilmente eccessivo.

- AWS Batch: sembra la scelta migliore per attività lunghe come il nuking di un'intera organizzazione, ma richiede una VPC.

Utilizzo

Tutto il codice è disponibile su Github

Prima di tutto, alcune avvertenze:

- È **MOLTO IMPORTANTE** filtrare il ruolo/utente AWS utilizzato per accedere all'account AWS. Un account indipendente può ancora essere accessibile utilizzando l'account root, ma gli account che fanno parte di un'organizzazione AWS potrebbero non averlo, quindi potrebbero risultare inaccessibili. Un filtro di questo tipo è già presente nella configurazione di esempio, ma è fondamentale controllare e capire sempre prima di utilizzare questa soluzione.
- aws-nuke può bypassare la *Termination Protection*. La configurazione di esempio è progettata per fare ciò, quindi è consigliabile cercare il flag "disable-deletion-protection" e modificarlo in base alle vostre esigenze.

Ora che sono stati esposti i principali rischi legati a questa soluzione, il primo passo è decidere quali account distruggere. In ognuno di essi, è richiesto il rilascio del template ”assumed-role.yaml”; trattandosi di un template CloudFormation, quindi la distribuzione tramite StackSet è molto comoda.

Una volta soddisfatto questo prerequisito, è sufficiente modificare i file di configurazione e distribuire il modello CDK all'interno di un account a scelta.

Ci sono due file di configurazione:

- `bucket_content/beNuke-config.yaml`contenente la configurazione di aws-nuke.
- `parameters.ts` contenente l'ID dell'account e la regione in cui viene rilasciata la soluzione e il nome delle risorse.

Una volta che tutto è stato rilasciato, è sufficiente eseguire il CodeBuild manualmente, tramite CLI, pianificandolo con Amazon EventBridge o in qualsiasi altro modo... come con un AWS IoT Button!

Bonus: AWS IoT Button come trigger

Immaginate la sensazione di onnipotenza nel poter chiudere un account AWS con la sola pressione di un pulsante. È possibile lanciare il job di CodeBuild utilizzando una funzione Lambda attivata da un IoT Button. Per rilasciare anche la Lambda è sufficiente impostare "buttonEnabled = true" all'interno dei parametri del template. Purtroppo, è necessario un passaggio manuale nella console AWS per impostare l'IoT button come trigger per la funzione.

Istruzioni più dettagliate sono disponibili nel file README.md del codice sorgente.

Sviluppi futuri

Un'aggiunta interessante a questa soluzione potrebbe essere un report sulle risorse cancellate contenente un elenco di account di destinazione, risorse fallite e altre informazioni. Purtroppo però non è semplice da realizzare: l'output di aws-nuke non è adatto a essere inserito direttamente in un report, è stracarico di informazioni, la maggior parte delle quali non è utile all'utente finale. L'output dovrebbe essere analizzato per estrarre solo le informazioni rilevanti e un breve sunto potrebbe essere inviato ad un indirizzo email specificato utilizzando Amazon SNS.

In ogni caso, l'output di aws-nuke potrebbe essere migliorato, molti errori sono stampati nell'output standard. Attualmente stiamo lavorando ad alcuni miglioramenti che invieremo come pull request al loro repository.

Conclusioni

Chiudere un account AWS non è facile come sembra, ma prestando attenzione a un paio di raccomandazioni contenute nella documentazione non è un compito poi tanto gravoso.

Purtroppo, la cancellazione delle risorse è una questione differente, procedere manualmente, soprattutto nei casi in cui le risorse si estendono su più account, è proibitivo. Ma grazie allo strumento giusto e a un po' di codice CDK, può essere fatto in molto meno tempo e diventare un compito divertente, anche se molto stressante.

Che ve ne pare? :) Fateci sapere nei commenti se la vostra vita è cambiata tanto quanto la nostra.

A presto con un nuovo articolo su Proud2beCloud!

About Proud2beCloud

Proud2beCloud è il blog di beSharp, APN Premier Consulting Partner italiano esperto nella progettazione, implementazione e gestione di infrastrutture Cloud complesse e servizi AWS avanzati. Prima di essere scrittori, siamo Solutions Architect che, dal 2007, lavorano quotidianamente con i servizi AWS. Siamo innovatori alla costante ricerca della soluzione più all'avanguardia per noi e per i nostri clienti. Su Proud2beCloud condividiamo regolarmente i nostri migliori spunti con chi come noi, per lavoro o per passione, lavora con il Cloud di AWS. Partecipa alla discussione!

Andrea Pusineri

DevOps Engineer @ beSharp. Mi diverto a risolvere problemi e sono cintura nera nel trovarli. Linux enthusiast e security guy wannabe, mi piacciono le CTF e nel tempo libero sono un avido lettore di fumetti/manga/libri. btw I use Arch

---