Best practices per strategie di Disaster Recovery da AWS verso AWS

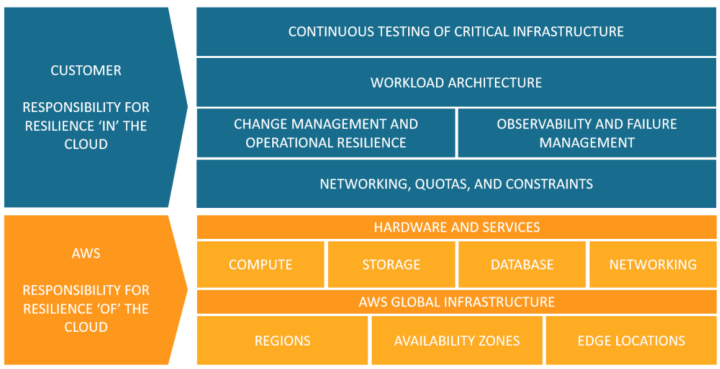
23 Giugno 2023 - 8 min. read

Disaster Recovery (DR) *DR Strategy*

Leggi Parte 1 | Leggi Parte 2

Questo è il terzo e ultimo capitolo della serie sul Disaster Recovery. nalizzeremo come varia il concetto di disaster recovery e quali tecniche adottare se la nostra infrastruttura di produzione si trova già sul Cloud di AWS.

AWS è responsabile della resilienza dell'infrastruttura fisica che sta alla base di tutti i servizi offerti, mentre il cliente ha la responsabilità in base ai servizi che si selezionano. Per esempio, un servizio come Amazon Elastic Compute Cloud (Amazon EC2) richiede di eseguire tutte le necessarie configurazioni e attività di gestione per la resilienza. Infatti, è il cliente che si deve occupare di sfruttare le Availability Zones che AWS ci mette a disposizione per poter ottenere l'alta disponibilità. Per i servizi gestiti, come Amazon S3 e Amazon DynamoDB, AWS gestisce l'infrastruttura, il sistema operativo e le piattaforme, mentre i clienti sono responsabili dei dati, comprese le strategie di backup, versioning e replica.Questo concetto si può riassumere in poche parole dicendo che **AWS** è responsabile della **resilienza DEL Cloud** e il **cliente** è responsabile della **resilienza NEL Cloud**.



Courtesy of AWS

Quindi avere i propri workload nel Cloud non comporta che possiamo dimenticarci di resilienza e business continuity. Anzi, dobbiamo capire come cambia il concetto di disastro.

I tipo di “disastro” sul Cloud

Normalmente si cerca di classificare in diverse categorie le differenti tipologie di disastro.

1. Guasti hardware o software: questo può includere il fallimento di server, dispositivi di archiviazione, switch di rete o componenti critici del sistema. Interruzioni di rete: un'interruzione della connettività può essere causata da problemi di rete, errori di configurazione o interruzioni dei provider di servizi di rete. In questi casi, è necessario ripristinare la connettività per garantire l'accesso alle risorse e alle applicazioni Cloud.
2. Disastri naturali: eventi come terremoti, alluvioni, incendi o tempeste possono causare danni fisici agli edifici o ai data center, compromettendo l'infrastruttura IT. In questi casi, è necessario ripristinare l'ambiente IT in un'area geograficamente diversa.
3. Errori umani: errori umani, come l'eliminazione accidentale di dati critici o la configurazione errata di un'applicazione, possono causare interruzioni e perdite di dati.
4. Attacchi informatici: gli attacchi informatici, come gli attacchi ransomware, possono compromettere la sicurezza dei dati e dei sistemi.

Nel Cloud al verificarsi di eventi catastrofici è AWS a occuparsi di ripristinare la propria infrastruttura fisica, mentre noi dobbiamo occuparci di mantenere i servizi attivi. Per alcune tipologie di eventi, errori umani e attacchi informatici, dobbiamo essere noi a prenderci cura anche del ripristino, oltre che della Business Continuity.

Nel Cloud gli attacchi informatici acquisiscono una nuova sfaccettatura in quanto questo ambiente è programmabile attraverso chiamate ad API pubbliche. La corretta configurazione secondo best practices di AWS IAM (rotazione credenziali, MFA, password sicure, ecc..) è utile al fine di fare prevenzione, ma non basta: a volte sono gli errori umani a stravolgere lo scenario. Si pensi per esempio, al furto delle credenziali da parte di malintenzionati che così, con uno script, possono sospendere il funzionamento dei nostri servizi.

Mettendo insieme tutti questi concetti capiamo quanto fondamentale sia progettare bene anche il Disaster Recovery da Cloud a Cloud.

Disaster Recovery AWS-to-AWS

Implementare strategie di Disaster recovery su AWS significa utilizzare **differenti AWS Region** per tutti gli eventi di interruzione dovuti ad hardware, rete e/o disastri naturali e

significa anche utilizzare, contestualmente, **differenti sottoscrizioni** (AWS le chiama Account) per essere resilienti ad attacchi informatici e/o errori umani.

Concentriamoci innanzitutto sulla selezione della Region secondaria.

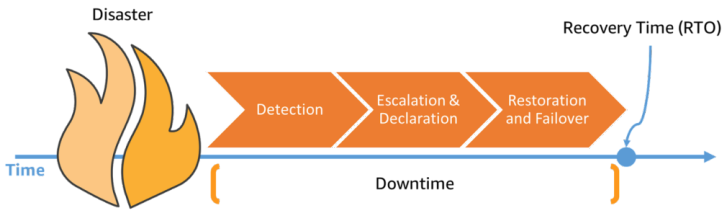
In questo caso dobbiamo tenere presente che **non tutti i servizi AWS sono presenti in tutte le Region**, quindi prima di scegliere dobbiamo assicurarci che siano presenti tutte le componenti che sfruttiamo nel nostro sito primario.

Altro punto da considerare sono **le quote di servizio**. Le quote sono presenti per evitare di effettuare accidentalmente il provisioning di più risorse di quelle necessarie e limitare i tassi di richiesta sulle operazioni API in modo da proteggere i servizi da un uso illecito. Le quote si possono aumentare tramite apertura di ticket di supporto fornendo le motivazioni e aspettando l'intervento del provider. Se abbiamo chiesto degli incrementi di quote nel nostro sito di produzione, dobbiamo ricordarci di richiederli anche per il sito di disaster recovery altrimenti rischieremmo di avere un'infrastruttura funzionante, ma non in grado di reggere il traffico.

Particolare attenzione, come sempre, va prestata ai **costi**. Sia la selezione della Region, che è soggetta a differenti legislazioni e regimi fiscali degli stati, sia la strategia di replica dei dati, influiscono sull'impatto economico.

Le fasi del ripristino

Una buona strategia di disaster recovery prevede, quindi, di valutare ambiente, tempistiche e costi. La strategia va poi declinata in diverse fasi: **rilevamento**, **ripristino** e **failover**. Per adesso il failback lo lasciamo da parte. Queste tre fasi vanno considerate nella definizione di RTO: a valle di un disastro impieghiamo del tempo per accorgerci dell'evento, altro tempo per attivare le nostre risorse e altro tempo per dirottare il traffico.



Courtesy of AWS

Al fine di ridurre il tempo di ripristino, la fase di rilevamento (o detection) andrebbe automatizzata. Non bisogna aspettare che qualcuno si accorga del disastro, dobbiamo anticipare i nostri clienti.

Per esempio, **Amazon Event Bridge** supporta le varie tipologie di eventi presenti nei servizi AWS e anche i trigger per le remediation. Tra gli eventi supportati ci sono quelli dei Cloud Watch Alarms che possiamo configurare sulle metriche che definiamo noi. Otteniamo così degli health check specifici basati sul funzionamento dei nostri workload.

Come detto in precedenza, esistono eventi che impattano direttamente AWS che ci notifica tramite la Personal Health Dashboard. Event Bridge supporta anche questo tipo di eventi.

Per ripristinare l'infrastruttura nella Region di ripristino è fortemente consigliato l'utilizzo del paradigma dell'*Infrastructure as Code (IaC)*. Questo include **AWS CloudFormation** e **AWS Cloud Development Kit (AWS CDK)** per creare in modo coerente l'infrastruttura sia nel sito primario che secondario.

Nella Region (e nell'account) di ripristino dobbiamo quindi trovare tutte le configurazioni di base, dal networking alle immagini delle nostre istanze o container. Per le EC2, sfruttiamo le AMI per incorporare il sistema operativo e i pacchetti necessari e automatizziamo la loro creazione e la distribuzione. Dato che le AMI contengono gli snapshot dei volumi, possiamo temporizzare il processo per avere una copia sempre più recente. Stesso principio vale per i container e sfruttando Elastic Container Registry (ECR) è possibile pubblicare le immagini replicandole in entrambi gli ambienti.

Per poter fare il ripristino anche i dati vanno replicati. Per esempio, se abbiamo file su S3 è possibile configurare la **cross-account e cross-region bucket replication** che ci permette di tenere in sync i file da entrambe le parti.

Piccolo consiglio: i bucket devono avere per forza nomi diversi, ricordatevi di configurare il vostro applicativo per utilizzare puntamenti diversi in caso di failover.

Per quanto riguarda i database possiamo usare la stessa tecnica utilizzata per le EC2 sfruttando gli snapshot copiandoli e portandoli nel sito di disaster recovery. Se l'RPO del nostro workload è molto stringente allora dovremo appoggiarsi a strumenti che tengono in sync il database di produzione con il secondario.

In generale un ottimo strumento per mettere a terra il nostro DR è **AWS Backup** che ci permette di configurare la frequenza con cui facciamo i backup e di distribuirli cross-region e cross-account.

Infine con la fase di failover il traffico viene rediretto sull'infrastruttura di DR. In questo caso viene in nostro soccorso **Amazon Route53** tramite il quale configuriamo il DNS utilizzando una policy di routing di failover con controlli di stato (Health Check). Usare i controlli di

stato è fondamentale per poter automatizzare la redirezione degli utenti verso l'infrastruttura secondaria.

Conclusioni

Siamo giunti alla fine di questa serie di 3 articoli dedicati al Disaster Recovery in cui abbiamo visto come garantire la continuità del business sia in ambienti ibridi, che in ecosistemi fully Cloud.

In particolare abbiamo visto come l'utilizzo dei servizi AWS in questo contesto offra numerosi vantaggi, tra cui la garanzia di resilienza, scalabilità e sicurezza delle infrastrutture, la disponibilità di strumenti avanzati per la protezione dei dati e la gestione semplificata delle risorse e dei costi.

Come emerge da questa nostra panoramica, tuttavia, esiste un aspetto critico che nessun provider, servizio o tecnologia possono gestire e che richiede quindi da parte delle aziende particolare attenzione, ovvero la pianificazione di una strategia di Disaster Recovery realmente disruption-proof. Per pianificare la strategia migliore per la propria organizzazione, infatti, oltre a conoscere i building block utili allo scopo, occorre comprendere appieno i concetti su cui tali strategie si basano, cosa si intende per responsabilità condivise e quali sono all'interno di ciascuna organizzazione e seguire best practices appropriate per affrontare con successo i diversi tipi di disastro. Va inoltre considerato che ciascuna strategia implementata richiede attenzione costante e un'adeguata gestione dei rischi e che il suo mantenimento resti un processo continuativo e in costante evoluzione.

In questa serie abbiamo cercato di porre l'accento proprio sulla criticità di tutto ciò e speriamo di avervi dato un punto di vista più completo su questo aspetto spesso sottovalutato in molte realtà.

Avete avuto esperienze di cui volete discutere? Aspettiamo le vostre testimonianze!

A presto con un nuovo articolo su **Proud2beCloud**!

Leggi Parte 1 | Leggi Parte 2

Related resources:

- AWS Well-Architected Framework (White Paper)
- AWS General Reference
- What is a landing zone?

- Best Practices for Organizational Units with AWS Organizations
- What Is AWS Control Tower?
- Customizations for AWS Control Tower
- Single-sign-on con G Suite sulla console di Amazon Web Services.
- Authenticate AWS Client VPN users with AWS IAM Identity Center
- Disaster Recovery of Workloads on AWS: Recovery in the Cloud

---

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!

Nicola Ferrari

Cloud Infrastructure Line Manager @ beSharp e AWS authorized instructor champion.Vivo la vita “un livello alla volta”. Ottengo i miei superpoteri raccogliendo caffeina nascosta qua e là nella mia mappa quotidiana. Sono un Internet surfer professionale (e ho visto tutto l’Internet per intero... almeno due volte!) e un appassionato di tecnologia, computer e networking. Costruire grandi cose IT - tutte precise e ordinate - contribuisce alla mia missione principale: la ricerca della perfezione!

Simone Merlini

CEO e co-fondatore di beSharp, Cloud Ninja ed early adopter di qualsiasi tipo di soluzione *aaS. Mi divido tra la tastiera del PC e quella a tasti bianchi e neri; sono specializzato nel deploy di cene pantagrueliche e nel test di bottiglie d'annata.
