Disaster Recovery in Cloud: tecniche efficaci per la Business continuity

26 Maggio 2023 - 11 min. read

Disaster Recovery (DR) *DR Strategy*

Ormai non è più una citazione, ma un mantra:

“Everything fails, all the time” - Werner Vogels

Questo principio ha condizionato la metodologia con cui si progetta infrastrutture in Cloud. Bisogna valutare le diverse tipologie di fallimento prima di progettare. Non tutti i disastri, però, sono uguali. Proviamo a darne una classificazione:

- Eventi su piccola scala: per esempio quando **un server o una VM** vanno offline
- Eventi su grande scala: nel mondo AWS possiamo immaginare un fallimento di **molteplici risorse di una singola Availability Zone** all’interno di una singola Region.
- Eventi colossali: il fallimento comporta il mancato servizio di molti sistemi e affligge tutti gli utenti. Il disastro colpisce **la Region nella sua interezza**.

Per rispondere in maniera efficace a queste tre tipologie di disastro bisogna progettare e pianificare in tre maniere diverse:

- **High Availability**: l'alta disponibilità fornisce ridondanza e tolleranza ai guasti. Un sistema è altamente disponibile quando è in grado di **resistere al fallimento di uno o più componenti individuali** (ad esempio, dischi rigidi, server o connettività di rete). I sistemi di produzione di solito hanno requisiti definiti per l'uptime.
- **Backup**: il backup è fondamentale per proteggere i dati e **garantire la continuità aziendale**. Tuttavia, può essere difficile da implementare. È essenziale tenere i dati critici salvati, in caso di disastro.

- **Disaster Recovery (DR)**: un disastro è qualsiasi **evento che ha un impatto negativo sulla continuità aziendale** o sulle finanze di un'azienda. Tali eventi includono il fallimento di hardware o software, un'interruzione di rete, un'interruzione di alimentazione o danni fisici a un edificio (come un incendio o un'alluvione). La causa può essere un errore umano o qualche altro evento significativo. Il ripristino dopo un disastro è un insieme di politiche e procedure che consentono il recupero o la continuazione delle infrastrutture tecnologiche e dei sistemi vitali dopo qualsiasi disastro.

In questa serie di articoli ci concentreremo in particolare sul Disaster Recovery, dalla sua definizione, fino alle tecniche di ripristino e come l'ecosistema di servizi AWS ci può aiutare a garantire la continuità del business in qualunque situazione.

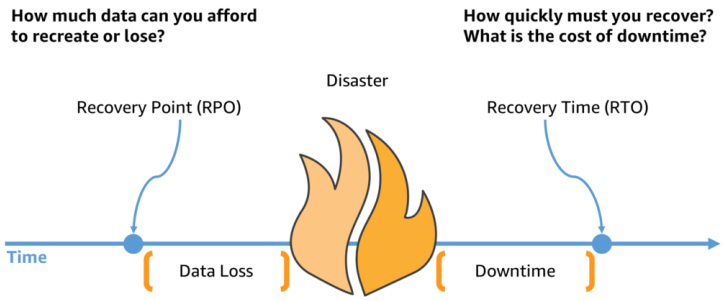
Il Disaster Recovery (DR) è un processo fondamentale per garantire la continuità delle operazioni aziendali in caso di disastri informatici. Esistono diverse tecniche di DR, e tutte mirano a ripristinare le applicazioni, i dati e le infrastrutture IT il più rapidamente possibile dopo un evento avverso.

In ambito Cloud, il DR è ancora più importante, poiché le applicazioni e i dati sono ospitati in remoto su server e infrastrutture che possono essere soggette a interruzioni di servizio a causa di problemi di connettività, guasti hardware, attacchi informatici o disastri naturali.

Ogni volta che si affronta questo argomento e per garantire un DR efficace è necessario partire da due acronimi che sono fondamentali per la selezione della miglior strategia di DR:

- RPO (**Recovery Point Objective**) per definire questo KPI dobbiamo rispondere ad una semplice domanda: quanti dati possiamo permetterci di perdere?
  RPO definisce ogni quanto tempo replicare i dati per assicurarsi il giusto lasso di tempo tra l'ultima replica e l'evento disastroso. Più è frequente è la replica meno sarà l'eventuale perdita massima di dati.

RTO (**Recovery Time Objective**), in questo caso la domanda a cui dobbiamo rispondere è: per quanto tempo possiamo lasciare i nostri sistemi non funzionanti prima di tornare operativi?
Il concetto di RTO si riferisce quindi al tempo che intercorre tra il disastro e il completo ripristino dei sistemi.



Courtesy of AWS (https://aws.amazon.com/it/blogs/architecture/disaster-recovery-dr-architecture-on-aws-part-i-strategies-for-recovery-in-the-cloud/)

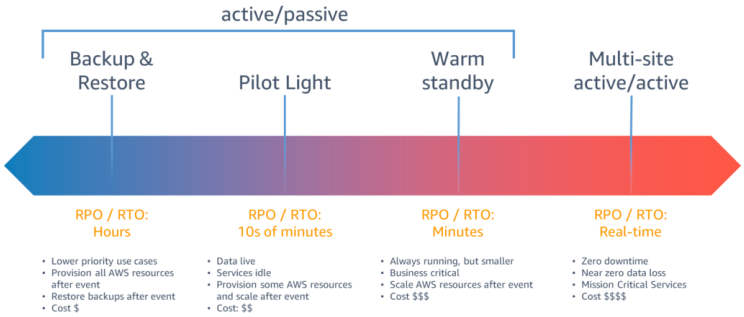
In sintesi, il DR è un processo fondamentale per garantire la continuità delle operazioni aziendali in caso di disastri informatici, e la sua efficacia dipende dall'obiettivo RTO/RPO e dal giusto equilibrio tra consistenza, disponibilità e tolleranza ai guasti.

Esistono diversi approcci per il Disaster Recovery, e la scelta dipende da fattori come il budget, la criticità dei dati e delle applicazioni, il tempo di ripristino accettabile (RTO) e il punto di ripristino accettabile (RPO).

Tra i metodi più comuni per il DR troviamo:

1. Backup and Restore: questa è una delle tecniche più comuni di disaster recovery. Coinvolge il regolare backup dei dati critici e dei sistemi, che vengono successivamente ripristinati in caso di interruzione. Si creano copie dei dati e delle applicazioni su un supporto separato, come dischi rigidi esterni o servizi di archiviazione cloud. Nel caso di un disastro, i dati vengono recuperati dal backup e ripristinati sui sistemi riparati o sostituiti.
2. Pilot Light: la tecnica del "Pilot Light" coinvolge la creazione di un'infrastruttura di base, essenziale per far funzionare le applicazioni critiche. Si tratta di avere un ambiente di backup parzialmente configurato, che può essere rapidamente scalato e attivato in caso di emergenza. Di solito, vengono replicati solo i componenti chiave del sistema, mentre il resto dell'infrastruttura viene creato al momento del ripristino.
3. Warm Standby: la tecnica del "Warm Standby" coinvolge la creazione di una replica dell'ambiente di produzione, in modo che sia pronto per essere attivato in caso di emergenza. L'ambiente di standby viene mantenuto aggiornato con i dati e le configurazioni più recenti, ma i servizi non vengono eseguiti in tempo reale. Quando si verifica un disastro, il sistema di produzione viene interrotto e il sistema di standby viene attivato per riprendere le operazioni.
4. Multi-site: la tecnica del "Multi-site" coinvolge la distribuzione dei sistemi e dei dati su più sedi geograficamente distinte. Le sedi possono essere situate in aree separate,

consentendo di evitare che un singolo evento danneggi tutte le infrastrutture contemporaneamente. In caso di disastro, il traffico e i servizi possono essere reindirizzati verso le sedi alternative per mantenere la continuità delle operazioni.



Courtesy of AWS (https://aws.amazon.com/it/blogs/architecture/disaster-recovery-dr-architecture-on-aws-part-i-strategies-for-recovery-in-the-cloud/)

Per quanto riguarda l'applicazione delle tecniche di DR in realtà di diverse dimensioni, possiamo dire che ogni organizzazione deve scegliere il metodo più adatto alle proprie esigenze e risorse. Le aziende di grandi dimensioni spesso hanno budget più elevati e possono permettersi di investire in infrastrutture costose come repliche attive e in costante sincronizzazione.

Le piccole e medie imprese, invece, possono optare per soluzioni più economiche e flessibili, come la replica dei dati nel cloud.

In sintesi, la scelta del metodo di Disaster Recovery dipende dalle esigenze dell'organizzazione, dal budget e dalla criticità dei dati e delle applicazioni.

Fin qui sembra tutto chiaro e lineare. Durante la replica dei dati per il disaster recovery, però, ci sono diversi fattori tecnici che devono essere presi in considerazione e che possono condizionare RPO/RTO. Tra i principali si annoverano:

- Latenza: la latenza è il ritardo temporale tra la scrittura dei dati nel sistema di produzione e la loro replica nel sito di ripristino. È importante valutare la latenza per garantire che i dati siano replicati in modo tempestivo e che non si verifichino perdite di dati significative in caso di interruzione.
- Consistenza: è essenziale garantire che i dati replicati siano **coerenti** e **integri**. Ciò significa che i dati devono essere replicati in un ordine coerente e che le operazioni di scrittura devono essere applicate in modo consistente su entrambi i siti. La mancanza di coerenza dei dati potrebbe portare a risultati imprevisti o alla perdita di dati.

- Scalabilità: durante la replica dei dati, è importante considerare la capacità di gestire volumi di dati crescenti nel tempo. La soluzione di replica dovrebbe essere in grado di scalare in modo efficiente per gestire l'aumento delle dimensioni dei dati senza compromettere le prestazioni complessive.
- Sicurezza: la sicurezza dei dati è fondamentale durante la replica per il disaster recovery. I dati replicati devono essere protetti da accessi non autorizzati e da minacce alla sicurezza. Ciò può richiedere l'implementazione di meccanismi di crittografia, controlli di accesso adeguati e misure di protezione fisica sia nel sito di produzione che in quello di ripristino.
- Test e verifica: è importante eseguire regolarmente test e verifiche per garantire che il processo di replica dei dati funzioni correttamente e che i dati siano ripristinabili in modo efficace. I test di ripristino dei dati possono aiutare a identificare eventuali problemi o lacune nel processo e consentono di apportare le necessarie correzioni.
- Automazione: l'automazione della replica dei dati può semplificare il processo e ridurre gli errori umani. L'implementazione di strumenti e processi automatizzati può consentire di gestire in modo efficiente la replica dei dati, migliorando la coerenza e l'affidabilità complessiva.
- Capacità di ripristino: durante la replica dei dati, è importante valutare la capacità di ripristinarli nel sito di destinazione. Ciò può richiedere l'implementazione di procedure di ripristino adeguate, la disponibilità di risorse hardware e software consone e la capacità di testare e verificare il processo di ripristino.

Ovviamente la coperta è sempre troppo corta e dobbiamo essere consapevoli che il disaster recovery deve necessariamente essere un compromesso. A sostegno di questo vorrei riportare un teorema fondamentale per i sistemi distribuiti: il CAP theorem.

Il Teaorema CAP

Il teorema **CAP (Consistency, Availability, Partition tolerance)**, che riguarda normalmente i sistemi di database distribuiti, può essere applicato in modo analogo al contesto del Disaster Recovery (DR), sebbene con alcune considerazioni aggiuntive. Noto anche come teorema di Brewer, afferma che è impossibile per un sistema informatico distribuito fornire simultaneamente tutte e tre le seguenti garanzie:

1. Coerenza (Consistency): la coerenza dei dati nel contesto del DR si riferisce alla garanzia che i dati replicati siano coerenti tra il sito di produzione e il sito di ripristino. In altre parole, i dati replicati dovrebbero riflettere le ultime modifiche effettuate nel sistema di produzione. Tuttavia, durante un evento di interruzione e durante il ripristino, potrebbe essere accettato un breve periodo di inconsistenza dei dati tra i

due siti per garantire la continuità operativa. Pertanto, il DR potrebbe sacrificare temporaneamente la coerenza a breve termine per garantire la disponibilità e la tolleranza alla partizione.

2. Disponibilità (Availability): l'obiettivo principale del DR è garantire la disponibilità dei servizi e dei dati critici in caso di interruzione. Ciò significa che il sistema di ripristino dovrebbe essere in grado di fornire servizi essenziali in modo tempestivo, anche se il sito di produzione è compromesso. La disponibilità nel contesto del DR può essere ottenuta sacrificando la coerenza dei dati a breve termine o utilizzando tecniche come il ripristino da backup o l'utilizzo di risorse di ripristino dedicate.
3. Tolleranza alla partizione (Partition tolerance): la tolleranza alla partizione nel contesto del DR è fondamentale. Si riferisce alla capacità del sistema di continuare a funzionare anche quando si verificano interruzioni o partizioni nella comunicazione tra il sito di produzione e il sito di ripristino. La replica dei dati e delle risorse in siti separati geograficamente contribuisce a garantire la tolleranza alla partizione. In caso di interruzione della comunicazione tra i siti, il sistema di ripristino dovrebbe essere in grado di operare autonomamente fino a quando la comunicazione viene ripristinata.

Quando si applica il teorema CAP al DR, spesso si fa una scelta consapevole per bilanciare la coerenza, la disponibilità e la tolleranza alla partizione in base alle esigenze specifiche del business e alle conseguenze delle interruzioni. Ad esempio, in un ambiente di ripristino prioritario, si potrebbe dare priorità alla disponibilità, sacrificando temporaneamente la coerenza dei dati. Al contrario, in un ambiente altamente sensibile alla coerenza, si potrebbe prioritizzare la coerenza sacrificando temporaneamente la disponibilità.

Tutto questo solo per capire come si fa disaster recovery, come si passa da produzione al nostro sito secondario. Ma c'è un punto della questione che è essenziale e che viene trascurato sempre da tutti, ovvero progettare **come si torna in produzione dopo che l'evento disastroso è rientrato**. Insomma, ci si dimentica sempre di definire come si "torna indietro".

Anche in questo caso è importante seguire un processo ben pianificato per garantire una transizione sicura e senza intoppi.

Prima di tutto bisogna fare una valutazione del ripristino. Prima di tornare alla produzione normale, è necessario valutare attentamente lo stato del sistema e dell'ambiente di produzione. **Verificare che il problema o l'evento di interruzione sia stato risolto completamente e che l'ambiente sia pronto per il ripristino.** In questo caso è fondamentale avere dei buoni sistemi di monitoring per rilevare eventuali anomalie o problemi. L'utilizzo di strumenti di monitoraggio e avvisi è utile per identificare e risolvere tempestivamente

qualsiasi problema che possa verificarsi durante la transizione al ritorno in produzione. Repetita iuvant: il **monitoring** è fondamentale anche **post-ripristino** per consentirci di individuare eventuali problemi residui o effetti collaterali che potrebbero emergere e intervenire tempestivamente per risolverli.

Non basta che l'ambiente di produzione sia finalmente disponibile, ma vanno eseguiti test e convalide. Ciò può includere test funzionali, test di carico e altri test appropriati per garantire che tutto funzioni correttamente come previsto. Verificare che i dati siano coerenti e integri.

In alcuni casi, potrebbe essere necessario eseguire un rollback delle modifiche apportate durante il processo di DR. Questo può coinvolgere il ripristino delle configurazioni, delle modifiche applicative o delle versioni precedenti dei dati. È importante pianificare il rollback in modo da minimizzare l'impatto sulle operazioni e garantire la coerenza dei dati.

Da non sottovalutare nelle situazioni emergenziali è la **comunicazione con gli utenti e gli stakeholder coinvolti**. Le persone coinvolte devono essere rese consapevoli delle modifiche apportate e dei passaggi necessari per tornare alla produzione normale.

Per rendere edotte le persone coinvolte è necessario documentare le nostre procedure, ovvero registrare tutti i passaggi, le modifiche apportate e le azioni intraprese durante il processo di DR. Ciò aiuterà a ricostruire l'evento di interruzione e ad analizzare in seguito per migliorare le future strategie di DR.

Conclusioni

Siamo giunti al termine della prima tappa del nostro viaggio nel Disaster Recovery in Cloud. Dopo averne definito i macro concetti e averne compreso la dinamiche, è il momento di entrare nel vivo dell'implementazione delle tecniche di DR in scenari aziendali complessi.

Cominceremo approfondendo nel prossimo articolo le migliori tecniche di DR per i contesti di tipo ibrido on-prem to Cloud per poi parlare di DR per la Business Continuity in ambito Cloud-to-Cloud nel terzo articolo della nostra mini-serie.

Pronti?

Ci vediamo tra 14 giorni 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!

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.
