Business continuity sul Cloud: l'importanza della Fault Tolerance

10 Novembre 2023 - 10 min. read

Business Continuity | Disaster Recovery (DR) | Fault tolerance | High Availability (HA)

Introduzione

Uno dei principi che cerchiamo ogni giorno di trasmettere ai nostri partner è che il Cloud non è un luogo magico, dove metto le risorse, pago una bolletta e automaticamente ottengo tutti i vantaggi che troviamo sempre elencati negli articoli, white paper e pagine di presentazione dei vari provider.

Il Cloud prevede uno shift di mentalità e, di conseguenza, un nuovo modo di fare investimenti e di vedere la parte "digital" delle aziende. Soprattutto per chi ha servizi rivolti ai consumatori è sempre più fondamentale la continuità, facendo in modo di farci trovare pronti ad esaudire le richieste ogni qual volta che qualcuno ha bisogno di noi.

Non solo va compreso l'impatto finanziario degli imprevisti, ma bisogna mettere in campo tutte le procedure e le pratiche che ci portino a massimizzare il nostro up-time.

Vorrei partire da alcuni dati, alcuni esempi di interruzione di servizio di colossi, che dimostrano che il problema generato non è solo una questione di fiducia dei clienti, ma anche finanziario. Nel 2015, un'interruzione di 12 ore dell'Apple Store è costata all'azienda 25 milioni di dollari. Nel 2016 un'interruzione di corrente di cinque ore di Delta Airlines in un centro operativo ha causato una perdita stimata di 150 milioni di dollari. Nel 2019, un'interruzione di 14 ore è costata a Facebook circa 90 milioni di dollari.

A questi dati si aggiunge anche il dodicesimo Annual Global Data Center Survey (2022) dell'Uptime Institute che fornisce una panoramica critica e un'idea della futura traiettoria del settore. Tra le principali conclusioni del rapporto vorrei soffermarmi su due particolari punti:

- Le **interruzioni di servizio stanno diventando più costose e restano ancora troppo frequenti** - Le interruzioni che costano più di 1 milione di dollari sono salite al 25%, un aumento significativo rispetto al 15% del 2021. Sebbene i dati indichino una tendenza in miglioramento, la frequenza delle interruzioni è ancora troppo alta. Più di due terzi di queste costano agli operatori più di 100.000 dollari, dunque le conseguenze stanno peggiorando.
- **La fiducia degli operatori nei servizi cloud pubblici è in aumento, nonostante i rischi continui di interruzione** - Le organizzazioni sono più propense a fidarsi del cloud. Nel 2022, solo il 63% degli operatori non sta collocando carichi di lavoro critici in un cloud pubblico, un notevole calo rispetto al quasi 75% nel 2019. Tuttavia, più di un terzo dei rispondenti segnala che le interruzioni delle zone di disponibilità del cloud pubblico (che sono relativamente comuni) causerebbero problemi significativi di performance.

Per quanto riguarda le interruzioni di servizio credo che nessuno si sia meravigliato dei risultati, ma sicuramente è interessante notare che l'asticella delle aspettative sia sempre più alta e le conseguenze dei downtime impattino sempre più fortemente sui guadagni e sulla fiducia degli utenti.

Il secondo punto, a mio avviso, ha a che fare con l'utilizzo della tecnologia Cloud senza comprenderne fino a fondo le basi. Come dicevamo, il Cloud non è magico, non basta mettere un workload in Cloud per aumentare l'up-time.

L'articolo vuole essere un tentativo di spiegare come il Cloud ci abilita e cosa dobbiamo fare noi per ottenere **fault-tolerance**, migliorare l'**up-time** e coprire alcuni degli aspetti di un piano strutturato di **business continuity**.

Cos'è la Fault Tolerance?

Se fossimo a scuola bisognerebbe partire dalla definizione:

la tolleranza al guasto, si riferisce alla capacità di un sistema o di un'applicazione di continuare a funzionare in modo affidabile anche quando si verificano guasti o malfunzionamenti in uno o più dei suoi componenti.

Riuscire ad essere tolleranti ai guasti è importante per indirizzare molti degli obiettivi definiti nei piani di business continuity. Una buona progettazione di infrastrutture tolleranti al guasto ci porta a vantaggi come la riduzione del downtime, il miglioramento dell'affidabilità, la riduzione del rischio e il mantenimento della qualità di servizio.

É doveroso fare subito chiarezza sul concetto di tolleranza ai guasti e come si differenzia da concetti come HA (Alta Affidabilità), ridondanza e DR (Disaster Recovery) spesso usati

erroneamente come sinonimi intercambiabili.

L'**alta affidabilità** si riferisce alla capacità di un sistema di rimanere operativo, senza interruzioni apprezzabili, per periodi di tempo continuativi. Questo obiettivo può essere raggiunto attraverso l'uso di monitoraggio continuo, ridondanza e altre misure. Un sistema ad alta affidabilità è progettato per evitare guasti e garantire la disponibilità continua del servizio.

La **ridondanza** serve a migliorare l'affidabilità di un sistema. Progettare infrastrutture che comprendono componenti ridondanti significa che, se un componente dovesse guastarsi, un altro possa prendere il suo posto per evitare interruzioni. La ridondanza può essere applicata sia a livello hardware che software.

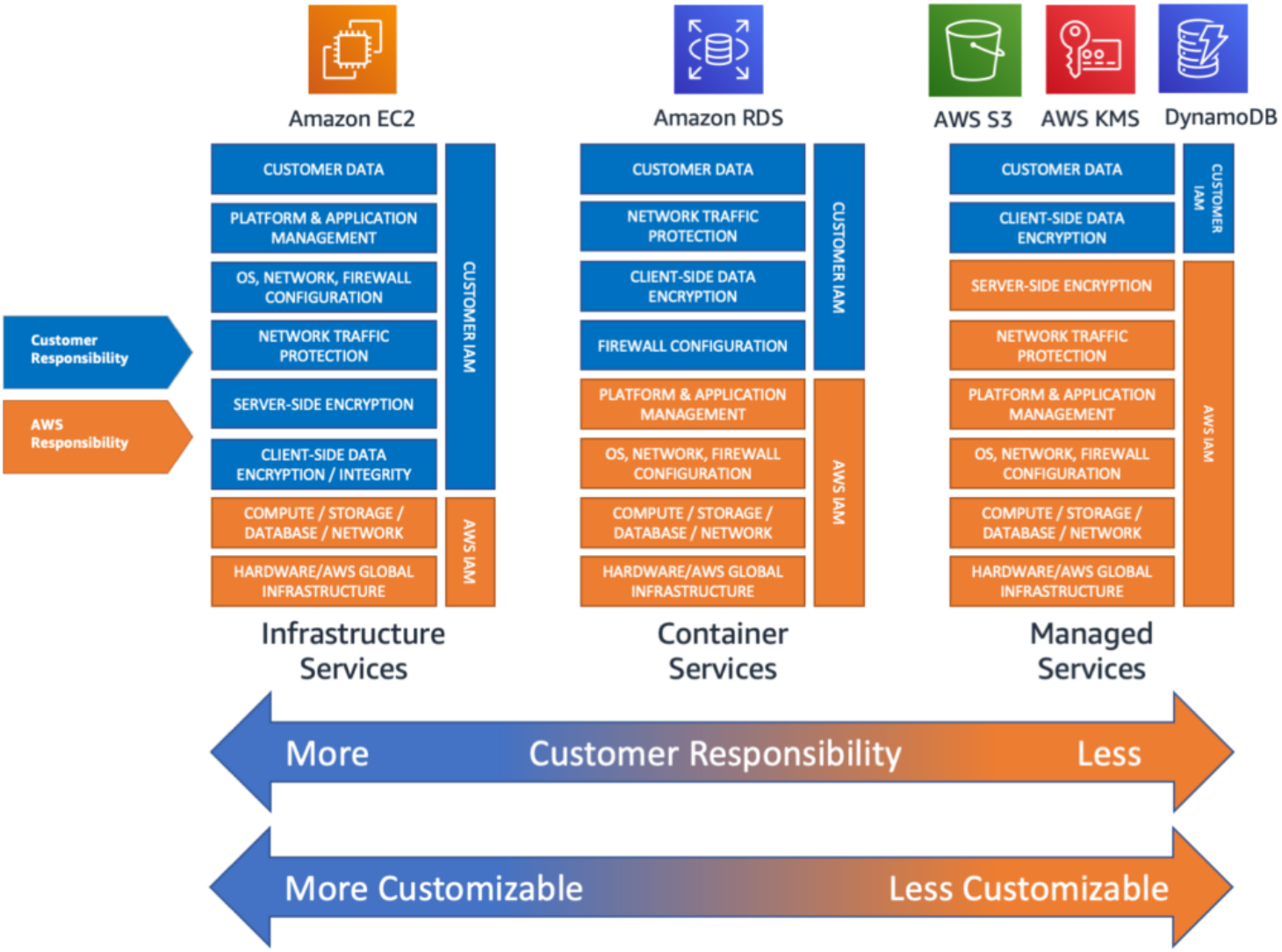
Il **Disaster Recovery** è un piano che mira a ripristinare un sistema o un'applicazione in caso di eventi catastrofici o guasti su larga scala. Questo coinvolge solitamente il backup dei dati, la pianificazione della continuità aziendale e procedure specifiche per ripristinare il sistema in un altro luogo o con risorse alternative.

L'alta affidabilità, la ridondanza e il DR sono tutti componenti importanti per ottenere la tolleranza al guasto, ma da soli potrebbero non essere sufficienti per garantirla. La tolleranza ai guasti richiede una progettazione oculata che prenda in considerazione una vasta gamma di scenari e preveda misure adeguate per gestirli senza interruzioni significative.

La tolleranza al guasto è, quindi, una strategia di progettazione e implementazione che contribuisce in modo significativo a continuare a operare in modo affidabile arrivando così a proteggere la reputazione aziendale e a mantenere la fiducia dei clienti.

Shared Responsibility Model

Chi si approccia al Cloud dovrebbe tenere ben presente che i provider non sono, non vogliono e non possono essere responsabili delle nostre applicazioni in toto. In più a seconda del servizio o del building block che sfruttiamo per le nostre infrastrutture, il provider sposta l’asticella della responsabilità. Prendendo ad esempio AWS, come si evince dall’immagine, servizi più IaaS spostano la responsabilità verso Cliente, mentre servizi gestiti spostano di più la responsabilità verso il provider.



https://aws.amazon.com/blogs/industries/applying-the-aws-shared-responsibility-model-to-your-gxp-solution/

Questo significa che nonostante il provider metta in campo ogni forma di ridondanza hardware e geografica, parte della responsabilità rimane comunque in capo al cliente.

AWS mette in luce queste due facce della medaglia distinguendo il provider, responsabile della resilienza **del** cloud, e il cliente, responsabile della resilienza **nel** cloud. AWS è responsabile della protezione dell'infrastruttura su cui vengono eseguiti tutti i servizi offerti nel Cloud AWS. Questa infrastruttura include l'hardware, il software, le reti e le strutture su cui vengono eseguiti i servizi Cloud AWS. Mentre la responsabilità del cliente è determinata dai servizi Cloud AWS che sceglie, in base alle configurazioni che deve fare e le attività di gestione e progettazione.

Fault Tolerance su AWS

Proviamo a mettere a terra tutti i discorsi fatti sulla Fault Tolerance. Nel caso di AWS dobbiamo partire capendo bene come è stata implementata l'infrastruttura fisica dei datacenter. È necessario ribadire che AWS ci da gli strumenti per creare delle infrastrutture resilienti, ma sta a noi progettare e metterle in opera nella maniera più corretta.

Partiamo dalle Availability Zones (A.Z.) che sono singoli data center o gruppi di essi, raggruppate logicamente in Region.

La **ridondanza** si ottiene sfruttando una singola A.Z., ma duplicando le istanze del servizio in uso all'interno della stessa. Facciamo due esempi pratici. Posso creare due EC2 all'interno delle quali installare MySql, metterle in cluster, active-standby ed essere robusti a fallimenti hardware del provider.

L'**High Availability** è la capacità di mantenere i servizi in esecuzione senza interruzioni. In un contesto SaaS, significa avere sistemi ridondanti in modo che, se uno dei componenti fallisce, un altro prenda immediatamente il suo posto senza causare downtime. L'HA è fondamentale per garantire che i servizi SaaS siano sempre accessibili agli utenti.

L'alta disponibilità la ottengo distribuendo il nostro database su due differenti Availability Zones. Importantissimo è avere progettato, impostato e configurato l'ambiente Cloud nella maniera più corretta. Se non sapete di cosa sto parlando vi devo rimandare alla nostra serie di articoli sulla Landing Zone.

Il **Disaster Recovery** è il piano che un'azienda mette in atto per ripristinare le sue operazioni in seguito a un grave incidente o una catastrofe. In un contesto SaaS, questo significa avere copie di backup dei dati e dei servizi critici in un luogo sicuro e facilmente accessibile. Nel caso di un'emergenza, il DR consente di ripristinare rapidamente i servizi, minimizzando i tempi di inattività.

Il Disaster Recovery si ottiene sfruttando due diverse AWS Region nel mondo. Non esistono molti meccanismi automatici o servizi che ci permettono di ottenere il DR con basso sforzo implementativo. Il DR va ragionato e ci sono una miriade di aspetti da tenere ben presenti e anche su questi aspetti abbiamo creato una serie di articoli.

Fault Tolerance nei rilasci

Finora abbiamo affrontato il discorso solo dal punto di vista infrastrutturale, ma anche dal punto di vista dello sviluppo software questa tematica ha un impatto.

Ad esempio, è buona pratica sfruttare il Cloud che è un ambiente totalmente programmabile, per creare piú ambienti di sviluppo su cui testare le modifiche prima di renderle effettive in produzione.

Lo sfruttamento di piú ambienti in cui sperimentare, validare, e testare i propri workload porta a ridurre il rischio di errori umani creando un processo automatico, controllato e reversibile.

Anche nelle fasi di rilascio delle evoluzioni dei nostri prodotti o dei nostri portali al pubblico è necessario tenere presente la fault-tolerance. Ad esempio, con il **Blue-Green Deployment**, due ambienti identici (blu e verde) vengono mantenuti in parallelo. Durante un aggiornamento, il traffico utente viene reindirizzato dall'ambiente "blu" a quello "verde". Questa tecnica consente di effettuare aggiornamenti senza downtime, e in caso di problemi, è possibile tornare rapidamente all'ambiente "blu".

Il Blue-Green deployment è affiancabile da pratiche come quelle definite nel **Canary Deployment** o quelle del **Rolling deployment.** Con queste tecniche l'aggiornamento viene distribuito solo a un piccolo sottoinsieme di utenti. Questi servono da indicatori per rilevare eventuali problemi e se non vengono riscontrati problemi, l'aggiornamento viene gradualmente esteso a un pubblico più ampio, in caso contrario, viene bloccato l'aggiornamento dirottando i pochi utenti "tester" sul vecchio ambiente.

Tutti questi sono approcci di deployment graduali e controllati che riducono il rischio di incorrere in guasti in produzione e di garantire che le modifiche siano stabili e affidabili prima di essere esposte agli utenti finali.

(Per saperne di più su Blue-Green Deployment, Canary Deployment e Rolling Deployment potete leggere questi articoli: Come creare una Pipeline di Continuous Deployment su AWS per il Deploy Blue/Green su ECS - Strategie di rilascio con Amazon ECS: il blue/green deployment)

Chaos Engineering

Arrivati a questo punto, la domanda che sorge spontanea è: come verifico la mia tolleranza al guasto?

Il **Chaos Engineering** è una pratica progettata per testare la resilienza e la tolleranza ai guasti di un sistema informatico in modo controllato e pianificato. L'obiettivo principale del chaos engineering è identificare le debolezze e le vulnerabilità del sistema prima che possano causare problemi reali in produzione.

Uno degli esempi di strumenti che sono stati creati per verificare la robustezza degli ambienti è la Chaos Monkey sviluppata da Netflix. Andate a vedere il progetto qui. Questo strumento ha questo nome strano perché vuol fare in modo di simulare i comportamenti imprevedibili di una scimmia abbandonata all'interno di un data center. Cosa potrebbe mai accadere? Cavi staccati, bottoni premuti, rack distrutti ecc.

Se l'infrastruttura regge e non si verificano downtime, allora si può finalmente dire di aver raggiunto la fault-tolerance.

In conclusione

Anche la ridondanza, l'alta disponibilità e il disaster recovery non bastano per migliorare la resilienza. Ci sono altri principi che andrebbero seguiti come la scalabilità orizzontale per seguire la curva di carico, il ripristino automatico in caso di guasto (self-healing), l'automazione e la verifica di tutte le procedure di ripristino.

Non vorrei essere banale, ma spiderman docet: "da grandi poteri derivano grandi responsabilità". Il Cloud ci da il potere che dobbiamo imparare a sfruttare nella maniera più corretta, studiando e facendo esperienza.

Spero che l'articolo sia stato utile per fare un po' di chiarezza su cosa si intenda per Fault Tolerance e come ottenerla sul Cloud.

A presto 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!
