Rilevamento e mitigazione dei guasti grigi

Modelli di resilienza Multi-AZ avanzati

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Modelli di resilienza Multi-AZ avanzati: Rilevamento e mitigazione dei guasti grigi

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Riassunto e introduzione i
Introduzione 1
Guasti grigi 3
Osservabilità differenziale 3
Esempio di guasto grigio 6
Risposta ai guasti grigi 7
Osservabilità Multi-AZ 10
Rilevamento dei guasti con allarmi CloudWatch compositi 14
Rileva l'impatto in una singola zona di disponibilità 15
Assicurati che l'impatto non sia regionale 16
Assicurati che l'impatto non sia causato da una singola istanza 17
Mettere tutto insieme 19
Rilevamento dei guasti mediante rilevamento dei valori anomali 20
Rilevamento dei guasti delle risorse zonali a istanza singola 25
Riepilogo 28
Modelli di evacuazione della zona di disponibilità 29
Indipendenza dalla zona di disponibilità 30
Piani di controllo e piani dati 36
Evacuazione controllata dal piano dati 37
Cambio di zona in Route 53 Application Recovery Controller (ARC) 37
Route 53 ARC 39
Utilizzo di un endpoint HTTP autogestito 40
Evacuazione controllata dal piano di controllo 47
Riepilogo 50
Conclusioni 52
Appendice A — Ottenere l'ID della zona di disponibilità 53
Appendice B — Esempio di calcolo al quadrato 55
Fattori determinanti 61
Revisioni del documento 62
Note 63
Glossario AWS 64
..... lxv

Modelli avanzati di resilienza Multi-AZ

Data di pubblicazione:11 luglio 2023(Revisioni del documento)

Molti clienti eseguono i propri carichi di lavoro in configurazioni di Multi-Availability Zone (AZ) ad alta disponibilità. Queste architetture funzionano bene durante gli eventi di errore binario, ma spesso incontrano problemi congrigiofallimenti. Le manifestazioni di questo tipo di guasto possono essere sottili e sfuggire a un rilevamento rapido e definitivo. Questo documento fornisce indicazioni su come strumentare i carichi di lavoro per rilevare l'impatto dei guasti grigi isolati in una singola zona di disponibilità e quindi adottare misure per mitigare tale impatto nella zona di disponibilità.

Introduzione

Lo scopo di questo documento è aiutarti a implementare in modo più efficace architetture Multi-AZ resilienti. Una delle migliori pratiche per la creazione di sistemi resilienti inCloud privato virtuale AmazonLe reti (VPC) sonodistribuisci ogni carico di lavoro su più zone di disponibilità.

UnZona di disponibilitàè uno o più data center discreti con alimentazione, rete e connettività ridondanti. L'utilizzo di più zone di disponibilità consente di gestire carichi di lavoro più altamente disponibili, tolleranti agli errori e scalabili di quanto sarebbe possibile da un singolo data center.

MoltiAWSservizi, comeScalabilità automatica di Amazon Elastic Compute Cloud (EC2)oServizio di database relazionale Amazon(Amazon RDS), fornisce una configurazione Multi-AZ. Questi servizi non richiedono la creazione di strumenti aggiuntivi di osservabilità o failover. Rendono i carichi di lavoro resilienti a modalità di errore binarie facilmente rilevabili all'interno di unRegione AWSche riguardano una singola zona di disponibilità. Potrebbe trattarsi di un completo guasto fisico dell'hardware, di un'interruzione dell'alimentazione o di un bug software latente che interessa la maggior parte delle risorse.

Ma esiste un'altra categoria di fallimenti denominataguasti grigi, le cui manifestazioni sono sottili e sfuggono a un'individuazione rapida e definitiva. Ciò a sua volta comporta tempi più lunghi per mitigare l'impatto causato dal guasto. Questo documento si concentra sugli impatti che i guasti grigi possono avere sulle architetture Multi-AZ, su come rilevarli e, infine, su come mitigarli.

> Le linee guida fornite in questo white paper sono applicabili principalmente a classi specifiche di carichi di lavoro che:

- Usa principalmente zonaleAWSservizi
- Necessità di migliorare la resilienza delle singole regioni
- Sono disposti a fare un investimento significativo per costruire i modelli di osservabilità e resilienza richiesti

In questi carichi di lavoro, potresti non essere disposto a fare alcuni o tutti i compromessi presentati in???o non avere la possibilità di utilizzare più regioni. È probabile che questi tipi di carichi di lavoro rappresentino un piccolo sottoinsieme del portafoglio complessivo e pertanto questa guida dovrebbe essere considerata a livello di carico di lavoro anziché a livello di piattaforma.

Guasti grigi

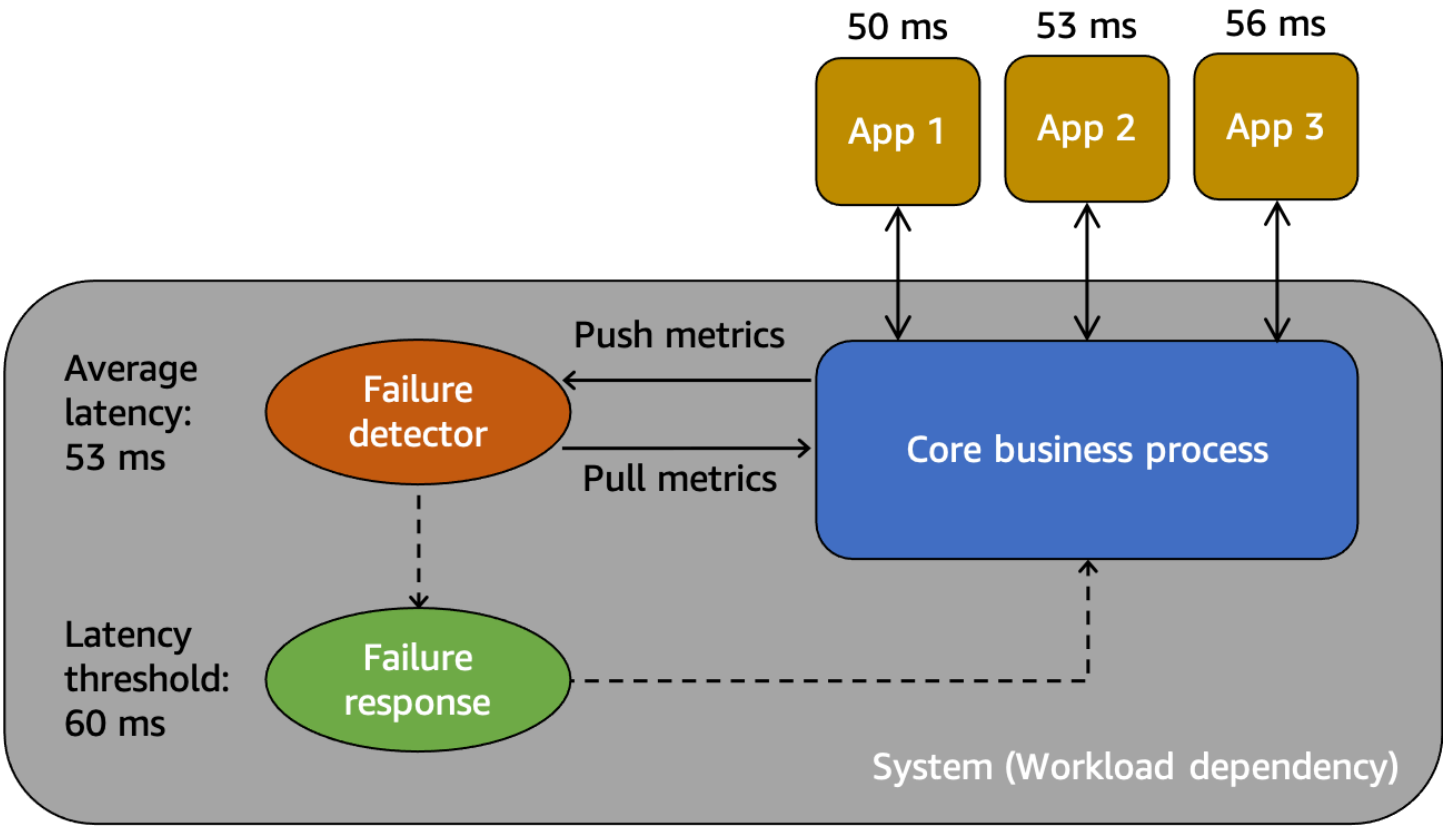
I guasti grigi sono definiti dalla caratteristica diosservabilità differenziale, il che significa che entità diverse osservano il fallimento in modo diverso. Definiamo cosa significa.

Osservabilità differenziale

I carichi di lavoro che gestisci in genere hanno delle dipendenze. Ad esempio, questi possono essereAWSservizi cloud che usi per creare il tuo carico di lavoro o un provider di identità (IdP) di terze parti che usi per la federazione. Queste dipendenze implementano quasi sempre la propria osservabilità, registrando metriche su errori, disponibilità e latenza, tra le altre cose generate dall'utilizzo dei clienti. Quando viene superata una soglia per una di queste metriche, la dipendenza di solito interviene per correggerla.

Queste dipendenze di solito hanno più utenti dei loro servizi. I consumatori implementano anche la propria osservabilità e registrano metriche e registri sulle loro interazioni con le loro dipendenze, registrando elementi come la latenza presente nelle letture su disco, il numero di richieste API non riuscite o il tempo impiegato per una query sul database.

Queste interazioni e misurazioni sono illustrate in un modello astratto nella figura seguente.

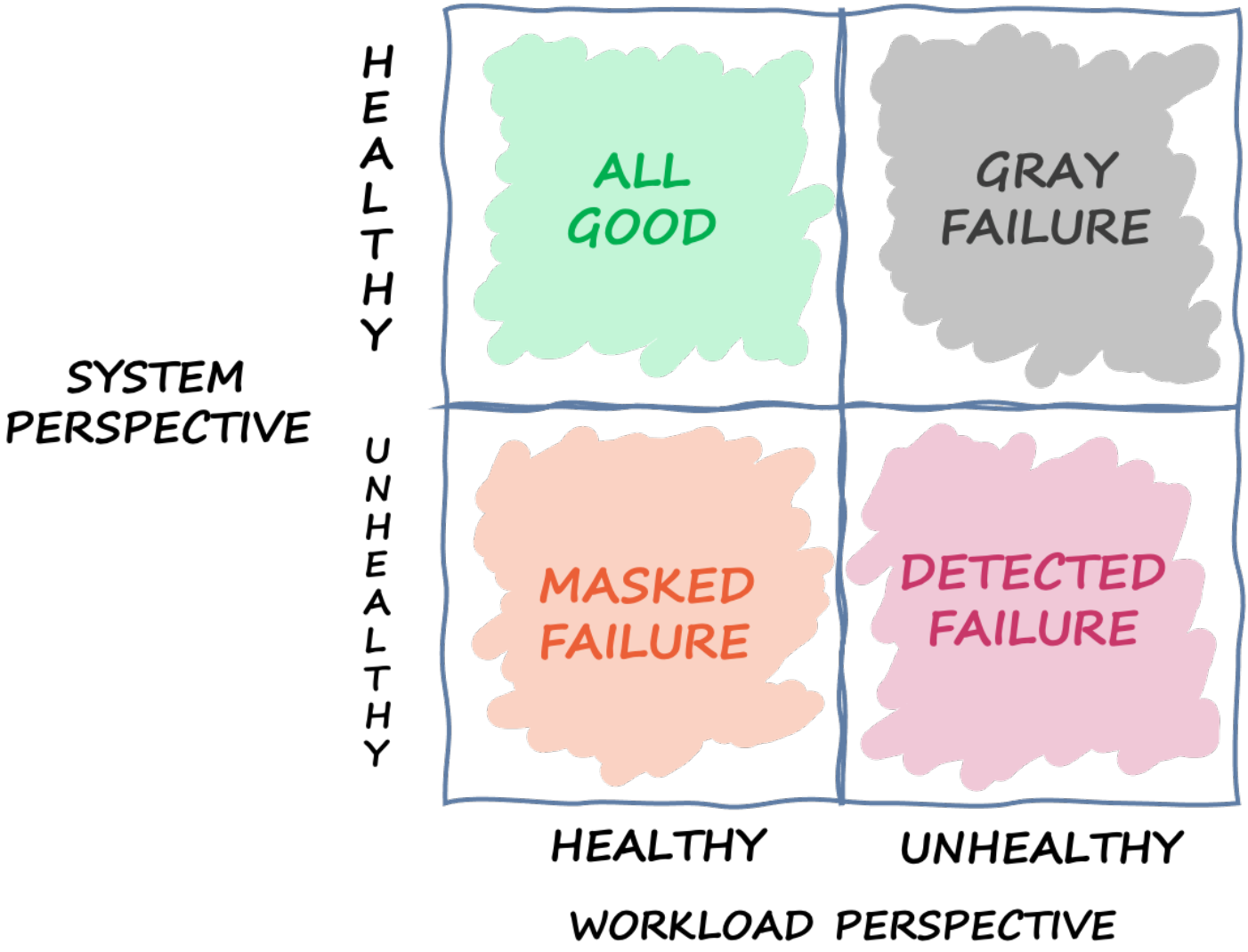


Un modello astratto per comprendere i fallimenti grigi

Innanzitutto, abbiamosistema, che in questo scenario è una dipendenza per i consumatori App 1, App 2 e App 3. Il sistema è dotato di un rilevatore di guasti che esamina le metriche create dal processo aziendale principale. Dispone inoltre di un meccanismo di risposta ai guasti per mitigare o correggere i problemi osservati dal rilevatore di guasti. Il sistema registra una latenza media complessiva di 53 ms e ha impostato una soglia per richiamare il meccanismo di risposta ai guasti quando la latenza media supera i 60 ms. Anche l'App 1, l'App 2 e l'App 3 stanno facendo le proprie osservazioni sulla loro interazione con il sistema, registrando una latenza media rispettivamente di 50 ms, 53 ms e 56 ms.

L'osservabilità differenziale è la situazione in cui uno degli utenti del sistema rileva che il sistema non è sano, ma il monitoraggio del sistema non rileva il problema o l'impatto non supera una soglia di allarme. Immaginiamo che App 1 inizi a registrare una latenza media di 70 ms anziché 50 ms. L'App 2 e l'App 3 non registrano alcuna variazione nelle loro latenze medie. Ciò aumenta la latenza media del sistema sottostante a 59,66 ms, ma non supera la soglia di latenza per attivare il meccanismo di risposta ai guasti. Tuttavia, l'App 1 registra un aumento del 40% della latenza. Ciò potrebbe influire sulla sua disponibilità superando il timeout del client configurato per l'App 1, oppure potrebbe causare impatti a cascata in una catena di interazioni più lunga. Dal punto di vista dell'App 1, il sistema

sottostante da cui dipende non è sano, ma dal punto di vista del sistema stesso, anche dell'App 2 e dell'App 3, il sistema è integro. La figura seguente riassume queste diverse prospettive.



Un quadrante che definisce i diversi stati in cui può trovarsi un sistema in base a diverse prospettive

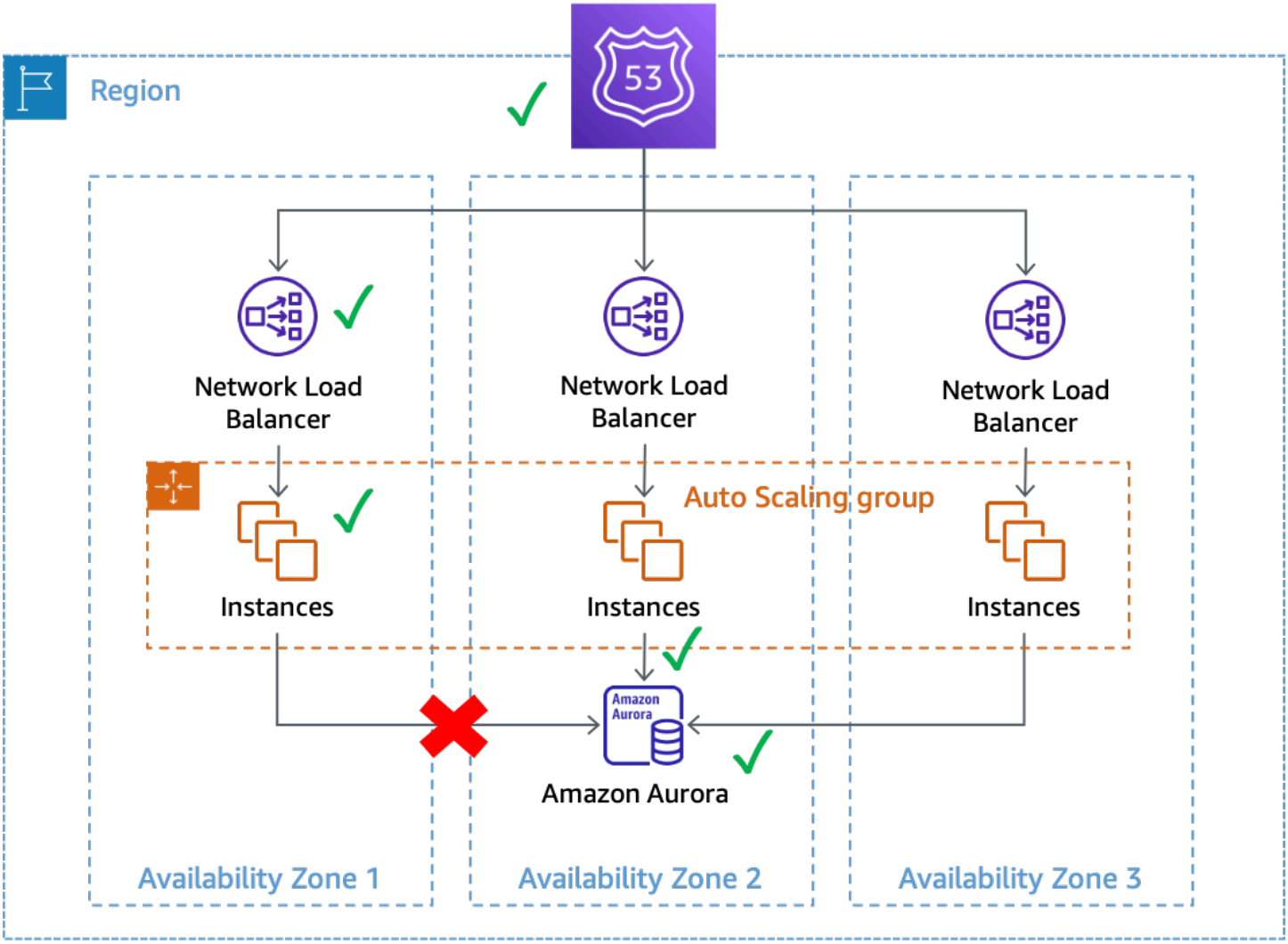
L'errore può anche attraversare questo quadrante. Un evento può iniziare come un errore grigio, quindi diventare un errore rilevato, quindi passare a un errore mascherato e quindi forse tornare a un errore grigio. Non esiste un ciclo definito e c'è quasi sempre la possibilità che il fallimento si ripresenti finché non ne viene risolta la causa principale.

La conclusione che ne traiamo è che i carichi di lavoro non possono sempre fare affidamento sul sistema sottostante per rilevare e mitigare l'errore. Non importa quanto sia sofisticato e resistente il sistema sottostante, ci sarà sempre la possibilità che un guasto possa passare inosservato o rimanere al di sotto della soglia di reazione. Gli utenti di quel sistema, come App 1, devono essere

attrezzati per rilevare e mitigare rapidamente l'impatto causato da un guasto grigio. Ciò richiede la creazione di meccanismi di osservabilità e recupero per queste situazioni.

Esempio di guasto grigio

I guasti grigi possono avere un impatto sui sistemi Multi-AZ inAWS. Prendiamo ad esempio una flotta diAmazon EC2istanze in un gruppo Auto Scaling distribuite in tre zone di disponibilità. Si connettono tutti a un database Amazon Aurora in un'unica zona di disponibilità. Quindi, si verifica un errore grigio che influisce sulla rete tra la zona di disponibilità 1 e la zona di disponibilità 2. Il risultato di questo problema è che una percentuale di connessioni al database nuove ed esistenti provenienti da istanze nella zona di disponibilità 1 fallisce. Questa situazione è illustrata nella figura riportata di seguito.



Un errore grigio che influisce sulle connessioni al database dalle istanze nella zona di disponibilità 1

In questo esempio, Amazon EC2 ritiene che le istanze nella zona di disponibilità 1 siano integre perché continuano a passarecontrolli dello stato del sistema e dell'istanza. Inoltre, Amazon EC2 Auto Scaling non rileva l'impatto diretto su alcuna zona di disponibilità e continua acapacità di avvio nelle zone di disponibilità configurate. Il Network Load Balancer (NLB) ritiene inoltre che le istanze che lo supportano siano integre, così come i controlli di integrità della Route 53 eseguiti sull'endpoint NLB. Analogamente, Amazon Relational Database Service (Amazon RDS) considera il cluster di database integro e non lo faattivare un failover automatico. Abbiamo molti servizi diversi che considerano tutti buoni i propri servizi e le proprie risorse, ma il carico di lavoro rileva un guasto che ne influisce sulla disponibilità. Questo è un grigio fallimento.

Risposta ai guasti grigi

Quando riscontri un errore grigio nel tuoAWSambiente, in genere sono disponibili tre opzioni:

- Non fate nulla e aspettate che la menomazione finisca.
- Se il danno è isolato in un'unica zona di disponibilità, evacuare quella zona di disponibilità.
- Failover su un altroRegione AWSe sfrutta i vantaggi diAWSIsolamento regionale per mitigare l'impatto.

MoltiAWSi clienti sono soddisfatti della prima opzione per la maggior parte dei loro carichi di lavoro. Accettano di avere un'eventuale prorogaObiettivo del tempo di ripristino (RTO)con il compromesso di non aver dovuto creare soluzioni aggiuntive di osservabilità o resilienza. Altri clienti scelgono di implementare la terza opzione,Disaster Recovery multiregionale(DR), come piano di mitigazione per un numero diverso di modalità di guasto. Le architetture multiregionali possono funzionare bene in questi scenari. Tuttavia, ci sono alcuni compromessi quando si utilizza questo approccio (fare riferimento aAWSNozioni di base su più regioniper una discussione completa sulle considerazioni multiregionali).

Innanzitutto, la creazione e la gestione di un'architettura multiregionale può essere un'impresa impegnativa, complessa e potenzialmente costosa. Le architetture multiregionali richiedono un'attenta considerazione di qualiStrategia DRtu selezioni. Potrebbe non essere fiscalmente fattibile implementare una soluzione di ripristino di emergenza attiva in più regioni solo per gestire le disfunzioni zonali, mentre una strategia di backup e ripristino potrebbe non soddisfare i requisiti di resilienza. Inoltre, i failover multiregionali devono essere continuamente utilizzati in fase di produzione, in modo da essere certi che funzioneranno quando necessario. Tutto ciò richiede molto tempo e risorse dedicati per costruire, utilizzare e testare.

In secondo luogo, la replica dei dati suRegioni AWSutilizzandoAWSi servizi oggi vengono eseguiti tutti in modo asincrono. La replica asincrona può causare la perdita di dati. Ciò significa che durante un failover regionale, è possibile che si verifichino perdite e incongruenze di dati. La tua tolleranza alla quantità di perdita di dati è definita comeObiettivo del punto di ripristino (RPO). I clienti, per i quali è richiesta una forte coerenza dei dati, devono creare sistemi di riconciliazione per risolvere questi problemi di coerenza quando la regione principale sarà nuovamente disponibile. In alternativa, devono creare i propri sistemi di replica sincrona o a doppia scrittura, che possono avere un impatto significativo sulla latenza, sui costi e sulla complessità della risposta. Inoltre, rendono la regione secondaria una forte dipendenza per ogni transazione, il che può potenzialmente ridurre la disponibilità dell'intero sistema.

Infine, per molti carichi di lavoro che utilizzano un approccio attivo/standby, è necessario un periodo di tempo diverso da zero per eseguire il failover in un'altra regione. Potrebbe essere necessario ridurre il tuo portafoglio di carichi di lavoro nella regione principale in un ordine specifico, svuotare le connessioni o interrompere processi specifici. Quindi, potrebbe essere necessario ripristinare i servizi in un ordine specifico. Potrebbe anche essere necessario fornire nuove risorse o richiedere tempo per superare i controlli sanitari richiesti prima di essere messe in servizio. Questo processo di failover può essere vissuto come un periodo di completa indisponibilità. Questo è ciò di cui si occupano gli RTO.

All'interno di una regione, moltiAWSi servizi offrono una persistenza dei dati fortemente coerente. Utilizzo delle implementazioni Amazon RDS Multi-AZreplica sincrona. Servizio Amazon Simple Storage(Amazon S3) offerteforteread-after-writeconsistenza. Amazon Elastic Block StorageOfferte (Amazon EBS)istantanee coerenti con crash a più volumi. Amazon DynamoDBlattinaeseguire letture fortemente coerenti. Queste funzionalità possono aiutarti a ottenere un RPO inferiore (nella maggior parte dei casi un RPO pari a zero) in una singola regione rispetto alle architetture multiregionali.

L'evacuazione di una zona di disponibilità può comportare un RTO inferiore rispetto a una strategia multiregionale, poiché l'infrastruttura e le risorse sono già distribuite tra zone di disponibilità. Invece di dover ordinare con attenzione la disattivazione e il backup dei servizi o di esaurire le connessioni, le architetture Multi-AZ possono continuare a funzionare in modo statico quando una zona di disponibilità è compromessa. Invece di un periodo di completa indisponibilità che può verificarsi durante un failover regionale, durante l'evacuazione di una zona di disponibilità, molti sistemi potrebbero subire solo un leggero degrado, poiché il lavoro viene spostato nelle zone di disponibilità rimanenti. Se il sistema è stato progettato perstaticamente stabilea causa di un guasto nella zona di disponibilità (in questo caso, ciò significherebbe predisporre la capacità nelle altre zone di disponibilità per assorbire il carico), i clienti del carico di lavoro potrebbero non vederne affatto l'impatto.

> È possibile che la riduzione di una singola zona di disponibilità influenzi una o piùAWS Servizi regionalioltre al tuo carico di lavoro. Se osservi un impatto regionale, dovresti considerare l'evento come un problema del servizio regionale, sebbene l'origine di tale impatto provenga da un'unica zona di disponibilità. L'evacuazione di una zona di disponibilità non mitigherà questo tipo di problema. Utilizza i piani di risposta che hai a disposizione per rispondere a un'interruzione del servizio regionale quando ciò si verifica.

Il resto di questo documento si concentra sulla seconda opzione, l'evacuazione della zona di disponibilità, come metodo per ottenere RTO e RPO inferiori in caso di guasti grigi Single-AZ. Questi modelli possono contribuire a migliorare il valore e l'efficienza delle architetture Multi-AZ e, per la maggior parte delle classi di carichi di lavoro, possono ridurre la necessità di creare architetture multiregionali per gestire questi tipi di eventi.

Osservabilità Multi-AZ

Per poter evacuare una zona di disponibilità durante un evento isolato in una singola zona di disponibilità, è necessario innanzitutto essere in grado di rilevare che l'errore è, di fatto, isolato in una singola zona di disponibilità. Ciò richiede una visibilità ad alta fedeltà sul comportamento del sistema in ciascuna zona di disponibilità. Molti AWS servizi forniscono out-of-the-box metriche che forniscono informazioni operative sulle risorse. Ad esempio, Amazon EC2 fornisce numerose metriche come l'CPUutilizzo, le letture e le scritture del disco e il traffico di rete in entrata e in uscita.

Tuttavia, quando crei carichi di lavoro che utilizzano questi servizi, hai bisogno di maggiore visibilità rispetto ai soli parametri standard. Desideri avere visibilità sull'esperienza del cliente fornita dal tuo carico di lavoro. Inoltre, è necessario che le metriche siano allineate alle zone di disponibilità in cui vengono prodotte. Queste sono le informazioni necessarie per rilevare i guasti grigi osservabili in modo differenziale. Questo livello di visibilità richiede strumentazione.

La strumentazione richiede la scrittura di codice esplicito. Questo codice dovrebbe eseguire operazioni come registrare la durata delle attività, contare il numero di elementi riusciti o non riusciti, raccogliere metadati sulle richieste e così via. È inoltre necessario definire delle soglie in anticipo per definire cosa è considerato normale e cosa no. È necessario definire obiettivi e diverse soglie di gravità per la latenza, la disponibilità e il conteggio degli errori nel carico di lavoro. L'articolo di Amazon Builders' Library sulla strumentazione dei sistemi distribuiti per la visibilità operativa fornisce una serie di best practice.

Le metriche devono essere generate sia dal lato server che dal lato client. Una best practice per generare metriche lato client e comprendere l'esperienza del cliente consiste nell'utilizzare Canaries, un software che analizza regolarmente il carico di lavoro e registra i parametri.

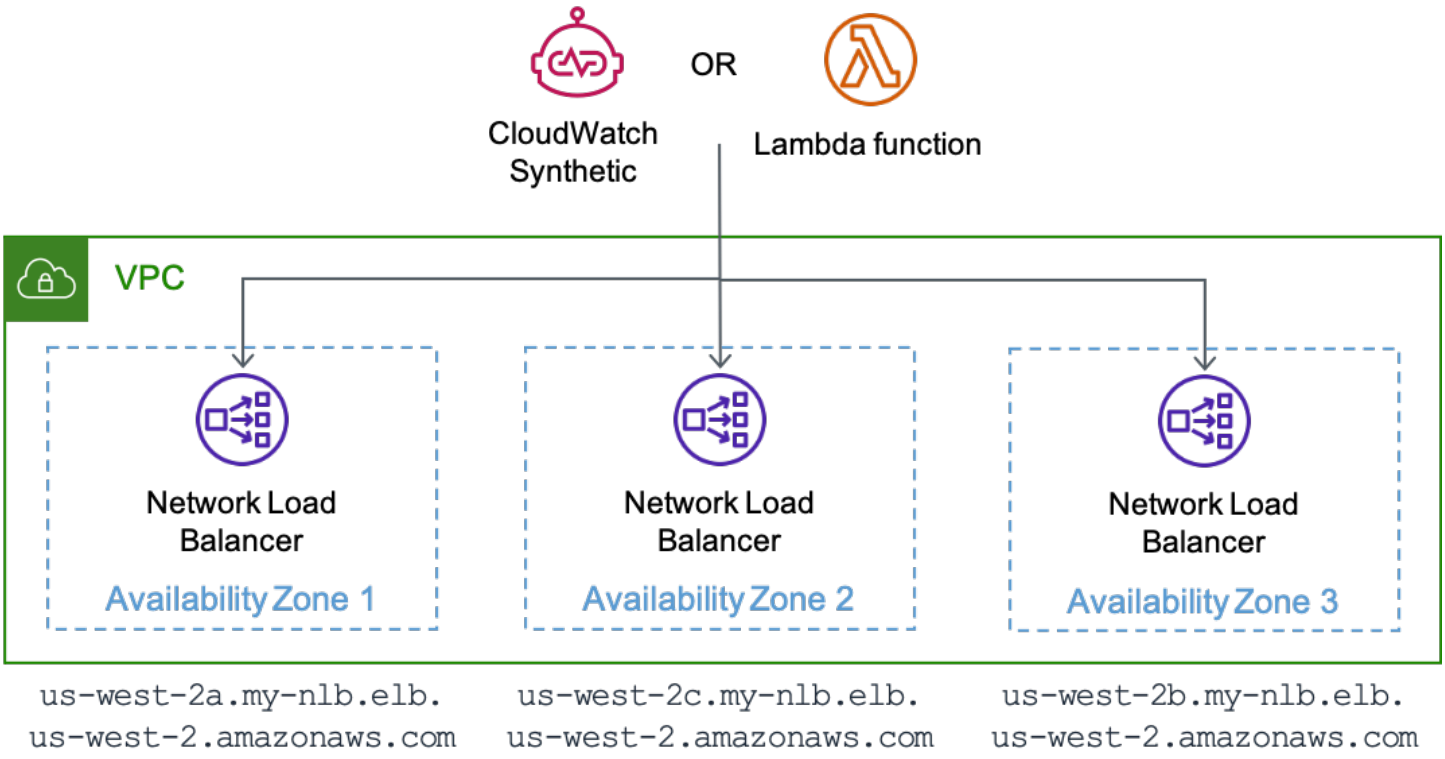
Oltre a produrre queste metriche, devi anche comprenderne il contesto. Un modo per farlo consiste nell'utilizzare le dimensioni. Le dimensioni conferiscono a una metrica un'identità unica e aiutano a spiegare cosa ti dicono le metriche. Per le metriche utilizzate per identificare gli errori nel carico di lavoro (ad esempio, latenza, disponibilità o conteggio degli errori), è necessario utilizzare dimensioni che si allineino ai limiti di isolamento degli errori.

Ad esempio, se si esegue un servizio Web in una regione, in più zone di disponibilità, utilizzando un framework web M odel-view-controller (MVC), è necessario utilizzare`Region`, `Availability Zone ID`, `ControllerAction`, e `InstanceId` come dimensioni per i set di dimensioni (se si utilizzavano microservizi, è possibile utilizzare il nome e il HTTP metodo del servizio anziché i nomi del controller e delle azioni). Questo perché ci si aspetta che diversi tipi di errori vengano isolati entro questi limiti.

Non ti aspetteresti che un bug nel codice del tuo servizio web che influisca sulla sua capacità di elencare i prodotti influisca anche sulla home page. Analogamente, non ci si aspetterebbe che un EBS volume completo su una singola EC2 istanza influisca sulla pubblicazione dei contenuti web da parte di altre EC2 istanze. La dimensione ID della zona di disponibilità è ciò che consente di identificare gli impatti relativi alla zona di disponibilità in modo coerente su tutto il territorio. Account AWS Puoi trovare l'ID della zona di disponibilità nei tuoi carichi di lavoro in diversi modi. Appendice A — Ottenere l'ID della zona di disponibilitàPer alcuni esempi, fare riferimento a.

Sebbene questo documento utilizzi principalmente Amazon EC2 come risorsa di calcolo negli esempi, `InstanceId` potrebbe essere sostituito con un ID contenitore per le risorse di calcolo Amazon Elastic Container Service (AmazonECS) e Amazon Elastic Kubernetes Service EKS (Amazon) come componenti delle tue dimensioni.

I vostri canarini possono anche utilizzare `Controller ActionAZ-ID`, e `Region` come dimensioni nelle loro metriche se disponete di endpoint zonali per il vostro carico di lavoro. In questo caso, allinea i canarini in modo che funzionino nella zona di disponibilità che stanno testando. In questo modo, se un evento isolato relativo alla zona di disponibilità ha un impatto sulla zona di disponibilità in cui si trova il canarino, non registri parametri che facciano apparire inadeguata una zona di disponibilità diversa da quella testata. Ad esempio, il tuo canary può testare ogni endpoint zonale per un servizio basato su Network Load Balancer () o Application Load Balancer NLB () utilizzando i relativi nomi di zonaALB. DNS



Un canarino in esecuzione su CloudWatch Synthetics o AWS Lambda una funzione che verifica ogni endpoint zonale di un NLB

Producendo metriche con queste dimensioni, puoi stabilire CloudWatch allarmi Amazon che ti avvisano quando si verificano cambiamenti di disponibilità o latenza entro tali limiti. Puoi anche analizzare rapidamente tali dati utilizzando i dashboard. Per utilizzare sia le metriche che i log in modo efficiente, Amazon CloudWatch offre il formato metrico incorporato (EMF) che consente di incorporare metriche personalizzate nei dati di registro. CloudWatchestrae automaticamente le metriche personalizzate in modo da poterle visualizzare e generare allarmi in base ad esse. AWS fornisce diverse librerie client per diversi linguaggi di programmazione che semplificano l'avvio. EMF Possono essere utilizzati con AmazonEC2, Amazon ECS EKS AWS Lambda, Amazon e ambienti locali. Con le metriche integrate nei tuoi log, puoi anche utilizzare Amazon CloudWatch Contributor Insights per creare grafici di serie temporali che mostrano i dati dei contributori. In questo scenario, potremmo visualizzare i dati raggruppati per dimensioni come `AZ-ID` `InstanceId`, o insieme a qualsiasi altro campo del registro `Controller` come o. `SuccessLatency` `HttpResponseCode`

```
{
  "_aws": {
    "Timestamp": 1634319245221,
    "CloudWatchMetrics": [
      {
        "Namespace": "workloadname/frontend",
        "Metrics": [
          { "Name": "2xx", "Unit": "Count" },
          { "Name": "3xx", "Unit": "Count" },
          { "Name": "4xx", "Unit": "Count" },
          { "Name": "5xx", "Unit": "Count" },
          { "Name": "SuccessLatency", "Unit": "Milliseconds" }
        ],
        "Dimensions": [
          [ "Controller", "Action", "Region", "AZ-ID", "InstanceId"],
          [ "Controller", "Action", "Region", "AZ-ID"],
          [ "Controller", "Action", "Region"]
        ]
      }
    ],
    "LogGroupName": "/loggroupname"
  },
  "CacheRefresh": false,
  "Host": "use1-az2-name.example.com",
  "SourceIp": "34.230.82.196",
```

```
  "TraceId": "|e3628548-42e164ee4d1379bf.",
  "Path": "/home",
  "OneBox": false,
  "Controller": "Home",
  "Action": "Index",
  "Region": "us-east-1",
  "AZ-ID": "use1-az2",
  "InstanceId": "i-01ab0b7241214d494",
  "LogGroupName": "/loggroupname",
  "HttpResponseCode": 200,
  "2xx": 1,
  "3xx": 0,
  "4xx": 0,
  "5xx": 0,
  "SuccessLatency": 20
}
```

Questo registro ha tre set di dimensioni. Procedono in ordine di granularità, da un'istanza alla zona di disponibilità all'altra (`Controllere Action` sono sempre inclusi in questo esempio). Supportano la creazione di allarmi per tutto il carico di lavoro che indicano quando c'è un impatto su un'azione specifica del controller in una singola istanza, in una singola zona di disponibilità o all'interno di un insieme. Regione AWS Queste dimensioni vengono utilizzate per il conteggio delle metriche di HTTP risposta 2xx, 3xx, 4xx e 5xx, nonché per la latenza per le metriche delle richieste riuscite (se la richiesta fallisce, viene registrata anche una metrica per la latenza delle richieste non riuscite). Il registro registra anche altre informazioni come il HTTP percorso, l'IP di origine del richiedente e se questa richiesta ha richiesto l'aggiornamento della cache locale. Questi punti dati possono quindi essere utilizzati per calcolare la disponibilità e la latenza di ogni elemento fornito API dal carico di lavoro.

> **Una nota sull'utilizzo dei codici di HTTP risposta per le metriche di disponibilità**
>
> In genere, puoi considerare le risposte 2xx e 3xx come riuscite e 5xx come errori. I codici di risposta 4xx si collocano da qualche parte nel mezzo. Di solito vengono prodotti a causa di un errore del client. Forse un parametro non è compreso nell'intervallo e porta a una risposta 400, oppure stanno richiedendo qualcosa che non esiste, con il risultato di una risposta 404. Queste risposte non verranno conteggiate in base alla disponibilità del carico di lavoro. Tuttavia, questo potrebbe anche essere il risultato di un bug nel software.
> Ad esempio, se hai introdotto una convalida degli input più rigorosa che rifiuta una richiesta che prima avrebbe avuto successo, la risposta 400 potrebbe essere considerata un calo della disponibilità. O forse stai limitando il cliente e restituendo una risposta 429. La limitazione di

un cliente protegge il servizio e ne mantiene la disponibilità, ma dal punto di vista del cliente il servizio non è disponibile per elaborare la sua richiesta. Dovrai decidere se i codici di risposta 4xx rientrano o meno nel calcolo della disponibilità.

Sebbene in questa sezione sia stato descritto l'utilizzo CloudWatch come metodo per raccogliere e analizzare le metriche, non è l'unica soluzione che puoi utilizzare. Puoi anche scegliere di inviare i parametri ad Amazon Managed Service for Prometheus e Amazon Managed Grafana, a una tabella Amazon DynamoDB o utilizzare una soluzione di monitoraggio di terze parti. La chiave è che le metriche prodotte dal carico di lavoro devono contenere un contesto relativo ai limiti di isolamento dei guasti del carico di lavoro.

Con carichi di lavoro che producono metriche con dimensioni allineate ai limiti di isolamento dei guasti, è possibile creare un'osservabilità che rilevi i guasti isolati nelle zone di disponibilità. Le sezioni seguenti descrivono tre approcci complementari per rilevare i guasti derivanti dal danneggiamento di una singola zona di disponibilità.

Argomenti

- Rilevamento dei guasti con allarmi compositi CloudWatch
- Rilevamento dei guasti mediante rilevamento dei valori anomali
- Rilevamento dei guasti delle risorse zonali a istanza singola
- Riepilogo

Rilevamento dei guasti con allarmi compositi CloudWatch

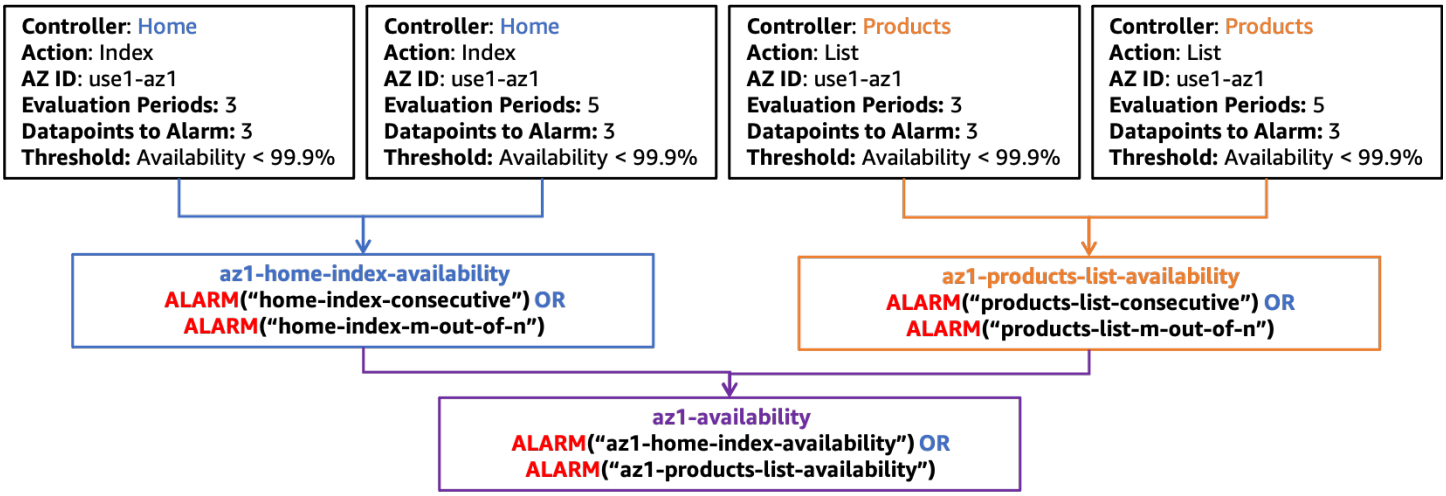
Nelle CloudWatch metriche, ogni set di dimensioni è una metrica unica e puoi creare un allarme su ognuna CloudWatch di esse. Puoi quindi creare allarmi CloudWatch compositi Amazon per aggregare questi parametri.

Per rilevare con precisione l'impatto, gli esempi di questo paper utilizzeranno due diverse strutture di CloudWatch allarme per ogni dimensione su cui viene impostato l'allarme. Ogni allarme utilizzerà un periodo di un minuto, il che significa che la metrica viene valutata una volta al minuto. Il primo approccio prevede l'utilizzo di tre punti dati consecutivi di violazione impostando i periodi di valutazione e i datapoint su Allarme su tre, ovvero un impatto per un totale di tre minuti. Il secondo approccio prevede l'utilizzo di un valore «M su N» in caso di violazione di 3 punti dati qualsiasi in una finestra di cinque minuti, impostando i periodi di valutazione su cinque e i datapoints su Alarm su tre.

Ciò offre la capacità di rilevare un segnale costante, oltre a uno che oscilla per un breve periodo. Le durate temporali e il numero di punti dati qui contenuti sono solo un suggerimento. Utilizzate valori adatti ai vostri carichi di lavoro.

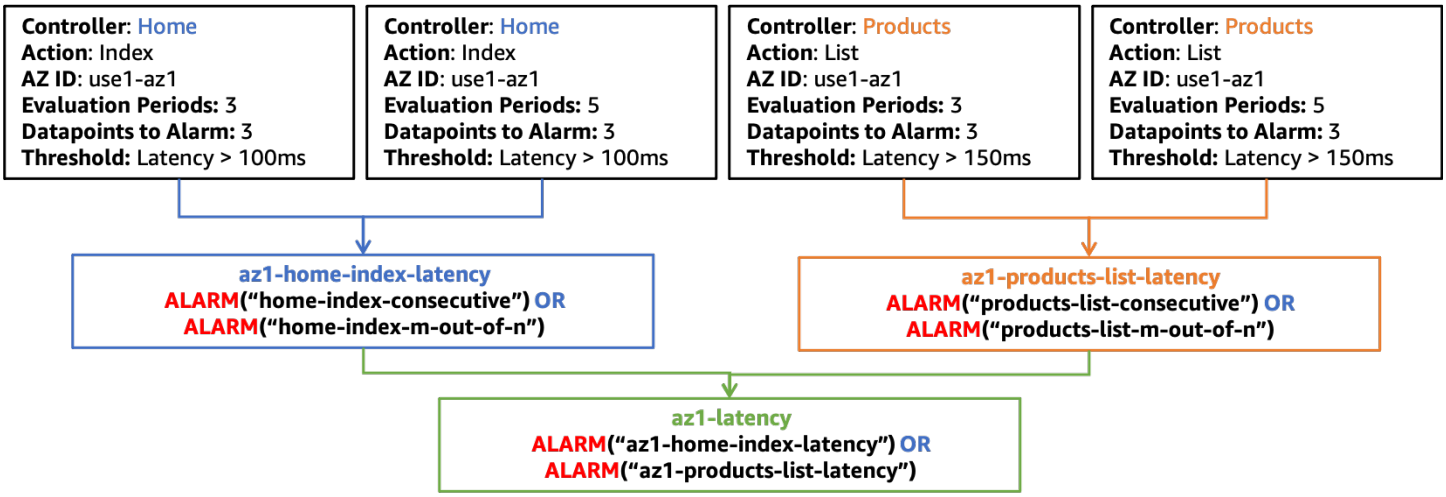
Rileva l'impatto in una singola zona di disponibilità

Utilizzando questo costrutto, considera un carico di lavoro che utilizza`Controller`, `Action` `InstanceIdAZ-ID`, e `Region` come dimensioni. Il carico di lavoro ha due controller, Products e Home, e un'azione per controller, rispettivamente List e Index. Funziona in tre zone di disponibilità nella `us-east-1` regione. È possibile creare due allarmi di disponibilità per ciascuno `Controller` e `Action` una combinazione in ciascuna zona di disponibilità, nonché due allarmi di latenza per ciascuno. Quindi, puoi facoltativamente scegliere di creare un allarme composito per la disponibilità per ciascuna combinazione. `Controller` `Action` Infine, crei un allarme composito che aggrega tutti gli allarmi di disponibilità per la zona di disponibilità. Questo è illustrato nella figura seguente per una singola zona di disponibilità`use1-az1`, utilizzando l'allarme composito opzionale per ogni `Controller` `Action` combinazione (esisterebbero allarmi simili anche per le zone di `use1-az3` disponibilità `use1-az2` e, ma non sono mostrati per semplicità).



Struttura di allarme composita per la disponibilità in *`use1-az1`*

È inoltre necessario creare una struttura di allarme simile anche per quanto riguarda la latenza, illustrata nella figura riportata di seguito.

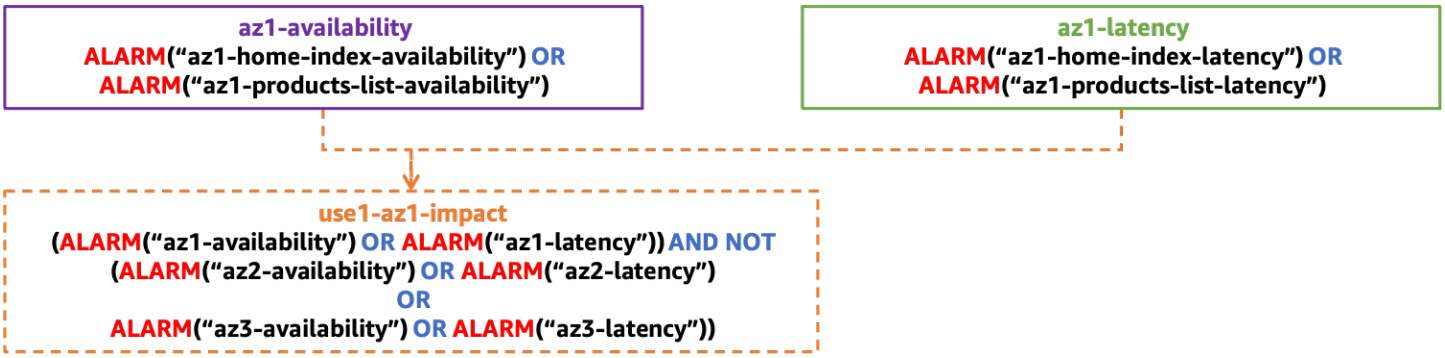


Struttura di allarme composita per la latenza in *use1-az1*

Per il resto delle figure di questa sezione, al `az1-availability` livello superiore verranno mostrati solo gli allarmi `az1-latency` compositi. Questi allarmi compositi, `az1-availability` e`az1-latency`, ti diranno se la disponibilità scende al di sotto o la latenza supera le soglie definite in una particolare zona di disponibilità per qualsiasi parte del tuo carico di lavoro. Potresti anche prendere in considerazione la possibilità di misurare la velocità effettiva per rilevare l'impatto che impedisce al carico di lavoro in una singola zona di disponibilità di ricevere lavoro. Puoi integrare anche gli allarmi prodotti dalle metriche emesse dai canarini in questi allarmi compositi. In questo modo, se il lato server o il lato client riscontrano un impatto sulla disponibilità o sulla latenza, l'allarme creerà un avviso.

Assicurati che l'impatto non sia regionale

È possibile utilizzare un altro set di allarmi compositi per garantire che solo un evento isolato della zona di disponibilità provochi l'attivazione dell'allarme. Ciò viene eseguito assicurando che un allarme composito della zona di disponibilità sia nello `ALARM` stato mentre gli allarmi compositi per le altre zone di disponibilità siano nello `OK` stato. Ciò si tradurrà in un allarme composito per ogni zona di disponibilità utilizzata. Un esempio è illustrato nella figura seguente (ricordate che ci sono allarmi per la latenza e la disponibilità in `use1-az2` and`use1-az3`,`az2-latency`,`az2-availability`, e `az3-latency`az3-availability`, che non sono illustrati per semplicità).



Struttura di allarme composita per rilevare impatti isolati in una singola AZ

Assicurati che l'impatto non sia causato da una singola istanza

Una singola istanza (o una piccola percentuale del parco macchine complessivo) può avere un impatto sproporzionato sulle metriche di disponibilità e latenza, tale da far sembrare che l'intera zona di disponibilità ne risenta, mentre in realtà non lo è. Rimuovere una singola istanza problematica è più veloce ed efficace che evacuare una zona di disponibilità.

Le istanze e i contenitori vengono in genere trattati come risorse effimere, spesso sostituiti da servizi come. AWS Auto Scaling È difficile creare un nuovo CloudWatch allarme ogni volta che viene creata una nuova istanza (ma sicuramente possibile utilizzando gli hook del ciclo di vita di Amazon EventBridge o Amazon EC2 Auto Scaling). Puoi invece utilizzare CloudWatch Contributor Insights per identificare la quantità di persone che contribuiscono alle metriche di disponibilità e latenza.

Ad esempio, per un'applicazione HTTP Web, è possibile creare una regola per identificare i principali contributori per 5xx HTTP risposte in ogni zona di disponibilità. Questo identificherà quali istanze stanno contribuendo a un calo della disponibilità (la nostra metrica di disponibilità definita sopra è determinata dalla presenza di errori 5xx). Utilizzando l'esempio del EMF log, create una regola utilizzando una chiave di. `InstanceId` Quindi, filtra il registro in base al `HttpResponseCode` campo. Questo esempio è una regola per la zona di `use1-az1` disponibilità.

```
{
    "AggregateOn": "Count",
    "Contribution": {
        "Filters": [
            {
                "Match": "$.InstanceId",
                "IsPresent": true
            },
```

```
            {
                "Match": "$.HttpStatusCode",
                "IsPresent": true
            },
            {
                "Match": "$.HttpStatusCode",
                "GreaterThan": 499
            },
            {
                "Match": "$.HttpStatusCode",
                "LessThan": 600
            },
            {
                "Match": "$.AZ-ID",
                "In": ["use1-az1"]
            },
        ],
        "Keys": [
            "$.InstanceId"
        ]
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "/loggroupname"
    ],
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    }
}
```

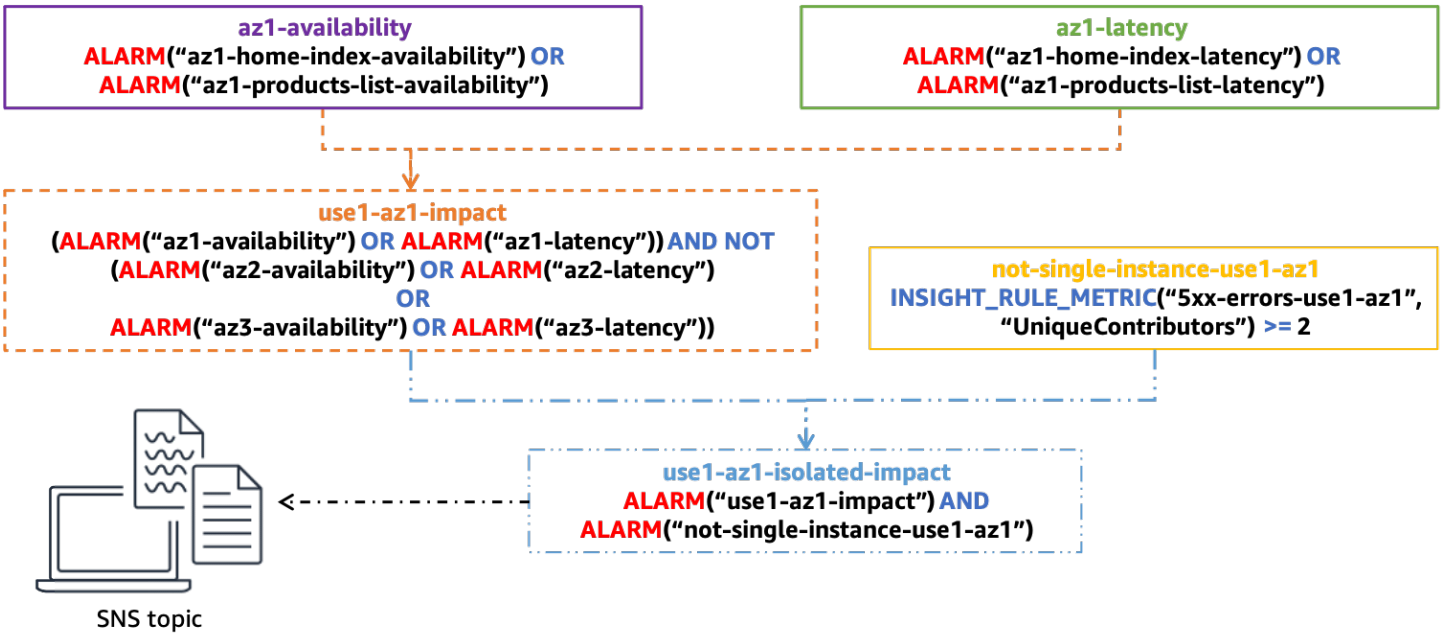
CloudWatch gli allarmi possono essere creati anche in base a queste regole. Puoi creare allarmi in base alle regole di Contributor Insights utilizzando la matematica metrica e la funzione con la `INSIGHT_RULE_METRIC` metrica. `UniqueContributors` Puoi anche creare regole di Contributor Insights aggiuntive con CloudWatch allarmi per metriche come la latenza o il conteggio degli errori oltre a quelli relativi alla disponibilità. Questi allarmi possono essere utilizzati con gli allarmi compositi isolati di Availability Zone Impact per garantire che le singole istanze non attivino l'allarme. La metrica per la regola Insights `use1-az1` potrebbe essere simile alla seguente:

```
INSIGHT_RULE_METRIC("5xx-errors-use1-az1", "UniqueContributors")
```

È possibile definire un allarme quando questa metrica è superiore a una soglia; per questo esempio, due. Viene attivato quando i contributori unici delle risposte 5xx superano tale soglia, a indicare che l'impatto proviene da più di due istanze. Il motivo per cui questo allarme utilizza un confronto maggiore anziché minore di è per assicurarsi che un valore zero per i contributori unici non faccia scattare l'allarme. Ciò indica che l'impatto non proviene da una singola istanza. Modifica questa soglia per il tuo carico di lavoro individuale. Una guida generale prevede che questo numero sia pari o superiore al 5% delle risorse totali nella zona di disponibilità. Oltre il 5% delle risorse interessate mostra una rilevanza statistica, se il campione è sufficientemente ampio.

Mettere tutto insieme

La figura seguente mostra la struttura composita completa degli allarmi per una singola zona di disponibilità:



Struttura di allarme composita completa per determinare l'impatto Single-AZ

L'allarme composito finale,`use1-az1-isolated-impact`, viene attivato quando l'allarme composito che indica l'impatto isolato della zona di disponibilità dalla latenza o dalla disponibilità è attivo e quando anche l'allarme basato sulla regola Contributor Insights per la stessa zona di disponibilità è attivo (vale a `ALARM` dire che l'impatto riguarda più di una singola istanza). `use1-az1-aggregate-alarm` `ALARM` `not-single-instance-use1-az1` Dovresti creare questa pila di allarmi per ogni zona di disponibilità utilizzata dal tuo carico di lavoro.

Puoi allegare un avviso Amazon Simple Notification Service (AmazonSNS) a questo allarme finale. Tutti gli allarmi precedenti sono configurati senza alcuna azione. L'avviso potrebbe avvisare

un operatore via e-mail di avviare un'indagine manuale. Potrebbe anche avviare l'automazione per evacuare la zona di disponibilità. Tuttavia, un avvertimento sull'automazione degli edifici per rispondere a questi avvisi. Dopo l'evacuazione di una zona di disponibilità, il risultato dovrebbe essere che l'aumento dei tassi di errore venga mitigato e l'allarme ritorni a uno stato. 0K Se si verifica un impatto in un'altra zona di disponibilità, è possibile che l'automazione riesca a evacuare una seconda o terza zona di disponibilità, rimuovendo potenzialmente tutta la capacità disponibile del carico di lavoro. L'automazione dovrebbe verificare se è già stata eseguita un'evacuazione prima di intraprendere qualsiasi azione. Potrebbe inoltre essere necessario scalare le risorse in altre zone di disponibilità prima che l'evacuazione abbia successo.

Quando aggiungi nuovi controller o azioni alla tua app MVC web, o un nuovo microservizio o, in generale, qualsiasi funzionalità aggiuntiva che desideri monitorare separatamente, devi solo modificare alcuni allarmi in questa configurazione. Creerai nuovi allarmi di disponibilità e latenza per quella nuova funzionalità e poi li aggiungerai agli allarmi compositi di disponibilità e latenza allineati alla zona di disponibilità e latenza appropriati, come `az1-availability` nell'esempio che abbiamo `az1-latency` usato qui. Gli allarmi compositi rimanenti rimangono statici dopo essere stati configurati. Ciò semplifica l'integrazione di nuove funzionalità con questo approccio.

Rilevamento dei guasti mediante rilevamento dei valori anomali

Una lacuna rispetto all'approccio precedente potrebbe verificarsi quando si riscontrano tassi di errore elevati in più zone di disponibilità che si verificano per un motivo non correlato. Immagina uno scenario in cui EC2 le istanze siano distribuite in tre zone di disponibilità e la soglia di allarme di disponibilità sia del 99%. Quindi, si verifica un danno a una singola zona di disponibilità, che isola molte istanze e fa scendere la disponibilità in quella zona al 55%. Allo stesso tempo, ma in una zona di disponibilità diversa, una singola EC2 istanza esaurisce tutto lo storage del EBS volume e non è più in grado di scrivere file di log. Ciò fa sì che inizi a restituire errori, ma supera comunque i controlli di integrità del load balancer perché questi non attivano la scrittura di un file di registro. Ciò si traduce in una riduzione della disponibilità al 98% in quella zona di disponibilità. In questo caso, l'allarme di impatto di una singola zona di disponibilità non si attiverebbe perché si riscontra un impatto sulla disponibilità in più zone di disponibilità. Tuttavia, è comunque possibile mitigare quasi tutto l'impatto evacuando la zona di disponibilità compromessa.

In alcuni tipi di carichi di lavoro, è possibile che si verifichino errori coerenti in tutte le zone di disponibilità, per cui la precedente metrica di disponibilità potrebbe non essere utile. Prendiamo ad AWS Lambda esempio. AWS consente ai clienti di creare il proprio codice da eseguire nella funzione Lambda. Per utilizzare il servizio, è necessario caricare il codice in un ZIP file, comprese le

dipendenze, e definire il punto di ingresso alla funzione. Ma a volte i clienti sbagliano questa parte, ad esempio potrebbero dimenticare una dipendenza critica nel ZIP file o digitare erroneamente il nome del metodo nella definizione della funzione Lambda. Ciò fa sì che la funzione non venga richiamata e genera un errore. AWS Lambda vede continuamente questo tipo di errori, ma non è indicativo che qualcosa sia necessariamente malsano. Tuttavia, anche qualcosa come una compromissione della zona di disponibilità potrebbe causare la comparsa di questi errori.

Per individuare il segnale in presenza di questo tipo di rumore, è possibile utilizzare il rilevamento dei valori anomali per determinare se esiste una differenza statisticamente significativa nel numero di errori tra le zone di disponibilità. Sebbene riscontriamo errori in più zone di disponibilità, se si verificasse davvero un errore in una di esse, ci aspetteremmo di vedere un tasso di errore molto più elevato in quella zona di disponibilità rispetto alle altre, o potenzialmente molto inferiore. Ma quanto più alto o più basso?

Un modo per eseguire questa analisi consiste nell'utilizzare un test chi-squared (χ^2) per rilevare differenze statisticamente significative nei tassi di errore tra le zone di disponibilità (esistono molti algoritmi diversi per eseguire il rilevamento dei valori anomali). Diamo un'occhiata a come funziona il test chi-squared.

Un test chi-squared valuta la probabilità che si verifichi una certa distribuzione dei risultati. In questo caso, siamo interessati alla distribuzione degli errori in un insieme definito di. AZs Per questo esempio, per semplificare i calcoli, considera quattro zone di disponibilità.

Innanzitutto, stabilite l'ipotesi nulla, che definisce quello che ritenete sia il risultato predefinito. In questo test, l'ipotesi nulla è che ci si aspetti che gli errori vengano distribuiti uniformemente in ogni zona di disponibilità. Quindi, genera l'ipotesi alternativa, ovvero che gli errori non siano distribuiti in modo uniforme, il che indica una compromissione della zona di disponibilità. Ora puoi testare queste ipotesi utilizzando i dati delle tue metriche. A tal fine, campionerai le tue metriche in una finestra di cinque minuti. Supponiamo di ottenere 1000 punti dati pubblicati in quella finestra, in cui vengono visualizzati 100 errori totali. Ti aspetti che con una distribuzione uniforme gli errori si verifichino il 25% delle volte in ciascuna delle quattro zone di disponibilità. Supponiamo che la tabella seguente mostri ciò che ti aspettavi rispetto a ciò che hai effettivamente visto.

Tabella 1: Errori previsti e errori effettivi rilevati

AZ	Expected (Atteso)	Effettivo
`use1-az1`	25	20

AZ	Expected (Atteso)	Effettivo
use1-az2	25	20
use1-az3	25	25
use1-az4	25	35

Quindi, vedete che la distribuzione in realtà non è uniforme. Tuttavia, potresti credere che ciò sia avvenuto a causa di un certo livello di casualità nei punti dati che hai campionato. Esiste un certo livello di probabilità che questo tipo di distribuzione si verifichi nel set di campioni e si presume comunque che l'ipotesi nulla sia vera. Ciò porta alla seguente domanda: qual è la probabilità di ottenere un risultato almeno così estremo? Se tale probabilità è inferiore a una soglia definita, si rifiuta l'ipotesi nulla. Per essere statisticamente significativa, questa probabilità deve essere pari o inferiore al 5%. [1]

[1] Craparo, Robert M. (2007). «Livello di significatività». In Salkind, Neil J. Enciclopedia della misurazione e della statistica 3. Thousand Oaks, CA: Pubblicazioni. pp. 889—891. SAGE ISBN1-412-91611-9.

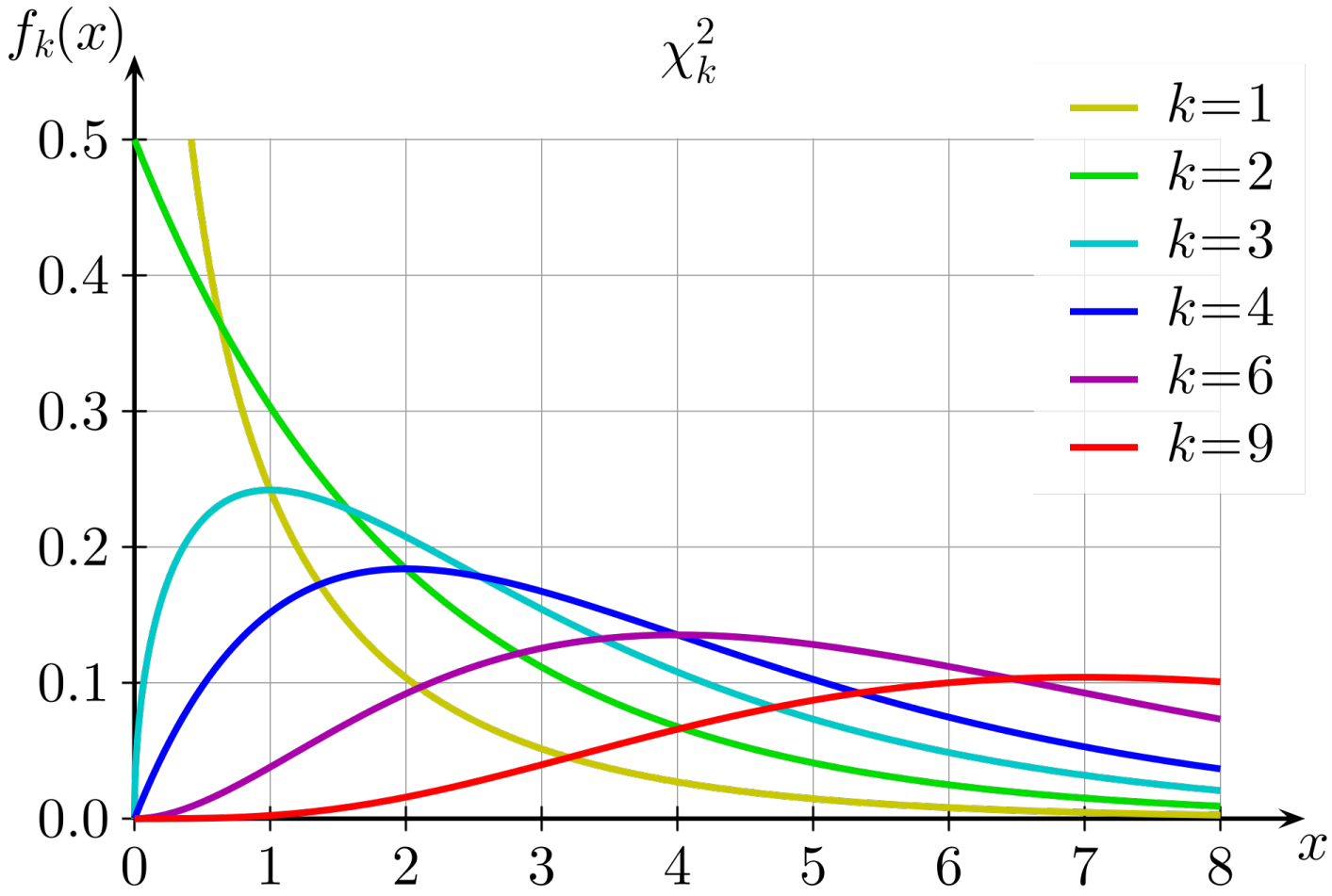
Come si calcola la probabilità di questo risultato? Si utilizza la statistica χ^2 che fornisce distribuzioni molto ben studiate e può essere utilizzata per determinare la probabilità di ottenere un risultato così estremo o più estremo utilizzando questa formula.

$$E_i = expected\ observations\ of\ type\ i$$
$$O_i = actual\ observations\ of\ type\ i \qquad (1)$$
$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Nel nostro esempio, ciò si traduce in:

$$\chi^2 = \frac{(20-25)^2}{25} + \frac{(20-25)^2}{25} + \frac{(25-25)^2}{25} + \frac{(35-25)^2}{25}$$
$$\chi^2 = \frac{-5^2}{25} + \frac{-5^2}{25} + \frac{0^2}{25} + \frac{10^2}{25} \qquad (2)$$
$$\chi^2 = 1 + 1 + 0 + 4$$
$$\chi^2 = 6$$

Quindi, cosa 6 significa in termini di probabilità? È necessario considerare una distribuzione chi-quadrata con il giusto grado di libertà. La figura seguente mostra diverse distribuzioni chi-squared per diversi gradi di libertà.



Distribuzioni chi-squared per diversi gradi di libertà

Il grado di libertà viene calcolato come uno in meno rispetto al numero di scelte del test. In questo caso, poiché vi sono quattro zone di disponibilità, il grado di libertà è tre. Quindi, vuoi conoscere

l'area sotto la curva (l'integrale) per x ≥ 6 sul grafico k = 3. È inoltre possibile utilizzare una tabella precalcolata con valori di uso comune per approssimare tale valore.

Tabella 2: Valori critici al quadrato di Chi

Gradi di libertà	Probabilità inferiore al valore critico				
	0,75	0,90	0,95	0,99	0,999
1	1,323	2,706	3,841	6,635	10,828
2	2,773	4,605	5,991	9,210	13,816
3	4,108	6,251	7,815	11,345	16,266
4	5,385	7,779	9,488	13,277	18,467

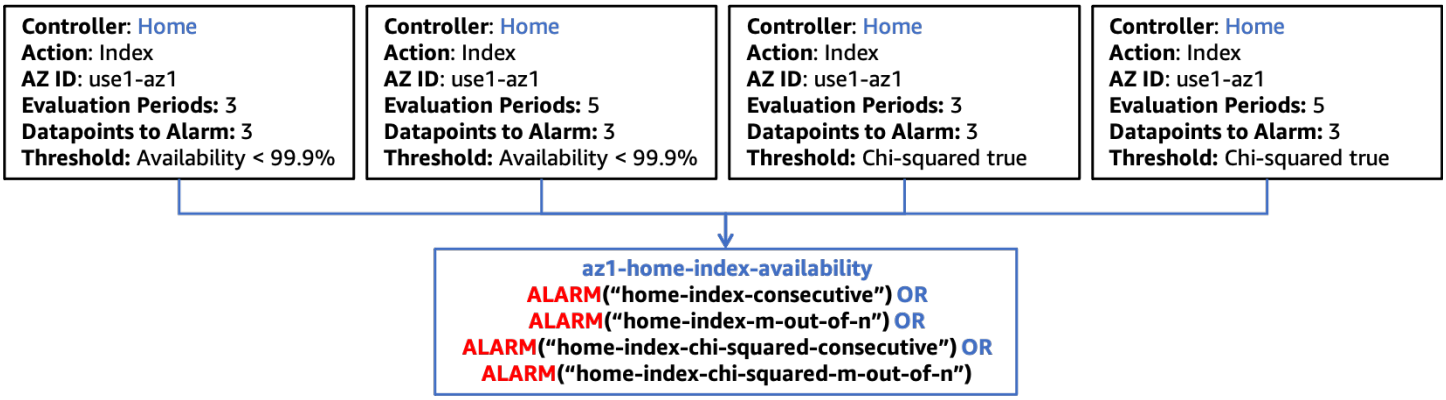
Per tre gradi di libertà, il valore chi-quadrato di sei rientra tra le colonne di probabilità 0,75 e 0,9. Ciò significa che esiste una probabilità superiore al 10% che si verifichi questa distribuzione, che non è inferiore alla soglia del 5%. Pertanto, si accetta l'ipotesi nulla e si stabilisce che non vi è una differenza statisticamente significativa nei tassi di errore tra le zone di disponibilità.

L'esecuzione di un test di statistica chi-squared non è supportata nativamente nella matematica CloudWatch metrica, quindi dovrai raccogliere le metriche di errore applicabili CloudWatch ed eseguire il test in un ambiente di calcolo come Lambda. Puoi decidere di eseguire questo test a livello di MVC Controller/Action o di microservizio individuale oppure a livello di zona di disponibilità. Dovrai valutare se una compromissione della zona di disponibilità influirebbe allo stesso modo su ogni controller/azione o microservizio o se qualcosa come un DNS guasto potrebbe avere un impatto su un servizio a basso throughput e non su un servizio a throughput elevato, che potrebbe mascherare l'impatto se aggregato. In entrambi i casi, selezionate le dimensioni appropriate per creare la query. Il livello di granularità influirà anche sugli CloudWatch allarmi risultanti che creerai.

Raccogli la metrica del conteggio degli errori per ogni AZ e Controller/Action in una finestra temporale specificata. Innanzitutto, calcola il risultato del test chi-squared come vero (c'era un'inclinazione statisticamente significativa) o falso (non c'era, cioè vale l'ipotesi nulla). Se il risultato è falso, pubblica un punto di dati pari a 0 nel flusso di metriche per i risultati chi-squared per ogni zona di disponibilità. Se il risultato è vero, pubblica un punto dati 1 per la zona di disponibilità con gli errori più lontani dal

valore previsto e uno 0 per gli altri (fare riferimento a Appendice B — Esempio di calcolo al quadrato per un codice di esempio che può essere utilizzato in una funzione Lambda). È possibile seguire lo stesso approccio dei precedenti allarmi di disponibilità utilizzando la creazione di un allarme CloudWatch metrico a 3 righe e un allarme CloudWatch metrico a 3 su 5 in base ai punti dati prodotti dalla funzione Lambda. Come negli esempi precedenti, questo approccio può essere modificato per utilizzare più o meno punti dati in una finestra più o meno lunga.

Quindi, aggiungete questi allarmi all'allarme di disponibilità esistente della zona di disponibilità per la combinazione Controller e Azione, mostrata nella figura seguente.



Integrazione del test statistico chi-squared con allarmi compositi

Come accennato in precedenza, quando si incorporano nuove funzionalità nel carico di lavoro, è sufficiente creare gli allarmi CloudWatch metrici appropriati specifici per quella nuova funzionalità e aggiornare il livello successivo nella gerarchia composita degli allarmi per includere tali allarmi. Il resto della struttura degli allarmi rimane statico.

Rilevamento dei guasti delle risorse zonali a istanza singola

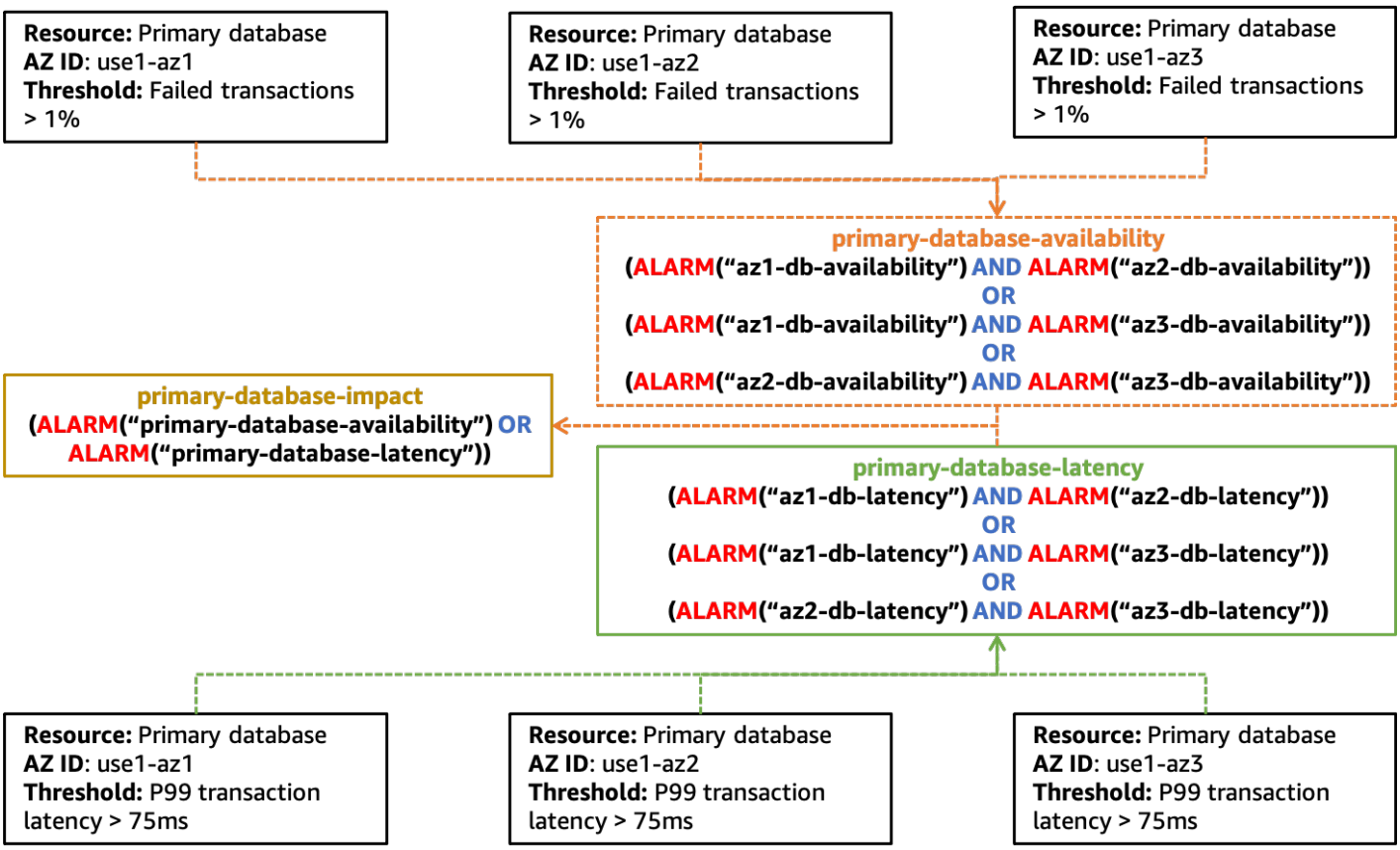
In alcuni casi, potresti avere una singola istanza attiva di una risorsa zonale, in genere sistemi che richiedono un componente a scrittura singola come un database relazionale (come AmazonRDS) o una cache distribuita (come Amazon ElastiCache (Redis OSS)). Se una singola violazione della zona di disponibilità influisce sulla zona di disponibilità in cui si trova la risorsa principale, può avere un impatto su ogni zona di disponibilità che accede alla risorsa. Ciò potrebbe causare il superamento delle soglie di disponibilità in ogni zona di disponibilità, il che significa che il primo approccio non identificherebbe correttamente la singola fonte di impatto della zona di disponibilità. Inoltre, è probabile che si verifichino tassi di errore simili in ogni zona di disponibilità, il che significa che anche l'analisi dei valori anomali non rileverebbe il problema. Ciò significa che è necessario implementare un'osservabilità aggiuntiva per rilevare in modo specifico questo scenario.

È probabile che la risorsa che ti preoccupa produca le proprie metriche sullo stato di salute, ma durante un danneggiamento della zona di disponibilità quella risorsa potrebbe non essere in grado di fornire tali metriche. In questo scenario, è necessario creare o aggiornare gli allarmi per sapere quando si vola alla cieca. Se ci sono metriche importanti che stai già monitorando e che attivano l'allarme, puoi configurare l'allarme in modo da considerare i dati mancanti come violazioni. Questo ti aiuterà a sapere se la risorsa smette di riportare i dati e può essere inclusa nella stessa in una riga e tra gli allarmi utilizzati in precedenza.

È anche possibile che, in alcune metriche che indicano lo stato della risorsa, questa pubblichi un punto dati a valore zero in assenza di attività. Se la compromissione impedisce le interazioni con la risorsa, non è possibile utilizzare l'approccio basato sui dati mancanti per questo tipo di metriche. Inoltre, probabilmente non vorrai allarmarti se il valore è pari a zero, poiché potrebbero esserci scenari legittimi in cui tale valore rientri nelle soglie normali. L'approccio migliore per rilevare questo tipo di problema consiste nell'utilizzare le metriche prodotte dalle risorse che utilizzano questa dipendenza. In questo caso vogliamo rilevare l'impatto in più zone di disponibilità utilizzando allarmi compositi. Questi allarmi dovrebbero utilizzare una manciata di categorie di metriche critiche relative alla risorsa. Di seguito sono elencati alcuni esempi:

- Produttività: la velocità delle unità di lavoro in entrata. Potrebbero trattarsi di transazioni, letture, scritture e così via.
- Disponibilità: misura il numero di unità di lavoro riuscite rispetto a quelle fallite.
- Latenza: misura più percentili di latenza per eseguire con successo il lavoro svolto in operazioni critiche.

  Ancora una volta, puoi creare allarmi metrici consecutivi e m su n per ogni metrica in ogni categoria metrica che desideri misurare. Come in precedenza, questi possono essere combinati in un allarme composito per determinare che questa risorsa condivisa è la fonte dell'impatto sulle zone di disponibilità. Si desidera essere in grado di identificare l'impatto su più di una zona di disponibilità con gli allarmi compositi, ma l'impatto non deve necessariamente riguardare tutte le zone di disponibilità. La struttura di allarme composita di alto livello per questo tipo di approccio è illustrata nella figura seguente.



Un esempio di creazione di allarmi per rilevare l'impatto su più zone di disponibilità causato da una singola risorsa

Noterai che questo diagramma è meno prescrittivo sul tipo di allarmi metrici da utilizzare e sulla gerarchia degli allarmi compositi. Questo perché scoprire questo tipo di problema può essere difficile e richiederà un'attenzione particolare ai segnali giusti per la risorsa condivisa. Potrebbe essere necessario valutare tali segnali anche in modi specifici.

Inoltre, dovresti notare che l'`primary-database-impact`allarme non è associato a una zona di disponibilità specifica. Questo perché l'istanza del database principale può trovarsi in qualsiasi zona di disponibilità per cui è configurata e non esiste una CloudWatch metrica che specifichi dove si trova. Quando vedi questo allarme attivarsi, dovresti usarlo come segnale che potrebbe esserci un problema con la risorsa e avviare un failover verso un'altra zona di disponibilità, se non è stato eseguito automaticamente. Dopo aver spostato la risorsa in un'altra zona di disponibilità, puoi aspettare e vedere se l'allarme isolato della zona di disponibilità è attivato oppure puoi scegliere di richiamare preventivamente il tuo piano di evacuazione della zona di disponibilità.

Riepilogo

Questa sezione descrive tre approcci per aiutare a identificare i problemi legati a una singola zona di disponibilità. Ciascun approccio deve essere utilizzato congiuntamente per fornire una visione olistica dello stato del carico di lavoro.

L'approccio CloudWatch composito agli allarmi consente di individuare problemi in cui la disparità di disponibilità non è statisticamente significativa, ad esempio disponibilità del 98% (la zona di disponibilità compromessa), del 100% e del 99,99%, il che non è causata da un'unica risorsa condivisa.

Il rilevamento dei valori anomali aiuta a rilevare i problemi relativi a una singola zona di disponibilità in cui si verificano errori non correlati in più zone di disponibilità e tutte superano la soglia di allarme.

Infine, l'identificazione del degrado di una risorsa zonale a istanza singola aiuta a scoprire quando una violazione della zona di disponibilità influisce su una risorsa condivisa tra le zone di disponibilità.

Gli allarmi risultanti da ciascuno di questi schemi possono essere combinati in una gerarchia CloudWatch composita di allarmi per scoprire quando si verificano problemi in una singola zona di disponibilità e hanno un impatto sulla disponibilità o sulla latenza del carico di lavoro.

Modelli di evacuazione della zona di disponibilità

Dopo aver rilevato l'impatto in una singola zona di disponibilità, il passaggio successivo consiste nell'evacuare quella zona di disponibilità. Ci sono due risultati che l'evacuazione deve raggiungere.

Innanzitutto, vuoi interrompere l'invio di lavoro alla zona di disponibilità interessata. Ciò potrebbe significare cose diverse in architetture diverse. In un carico di lavoro di richiesta/risposta, ciò significherebbe impedire che cose come le richieste HTTP o gRPC provenienti dai clienti vengano inviate al sistema di bilanciamento del carico o ad altre risorse nella zona di disponibilità. In un sistema di elaborazione in batch o in coda, ciò potrebbe comportare l'interruzione del lavoro di elaborazione delle risorse di elaborazione nella zona di disponibilità interessata. Dovrai inoltre impedire che le risorse nelle zone di disponibilità non interessate interagiscano con le risorse nella zona di disponibilità interessata, ad esempio un'istanza EC2 che invia traffico a uninterfaccia VPC endpointnella zona di disponibilità interessata o connessione all'istanza principale di un database.

Il secondo risultato è impedire che venga fornita nuova capacità nella zona di disponibilità interessata. Questo è importante perché è probabile che le nuove risorse, come le istanze o i container EC2, che vengono fornite nella zona di disponibilità interessata abbiano lo stesso impatto delle risorse esistenti. Inoltre, poiché il primo risultato impedisce loro l'invio di lavoro, non possono assorbire il carico per cui sono stati predisposti a gestire. Ciò comporta un aumento del carico sulle risorse esistenti, che alla fine può portare abrunireo totale indisponibilità del carico di lavoro. Sono disponibili diversi servizi di scalabilità automatica inAWSladdove applicabile:Scalabilità automatica di Amazon EC2,Scalabilità automatica dell'applicazione, eAWS Auto Scaling. Inoltre, servizi come Amazon ECS, Amazon EKS eAWS Batchpossono pianificare il lavoro sugli host in diverse zone di disponibilità in un VPC come parte del loro normale funzionamento.

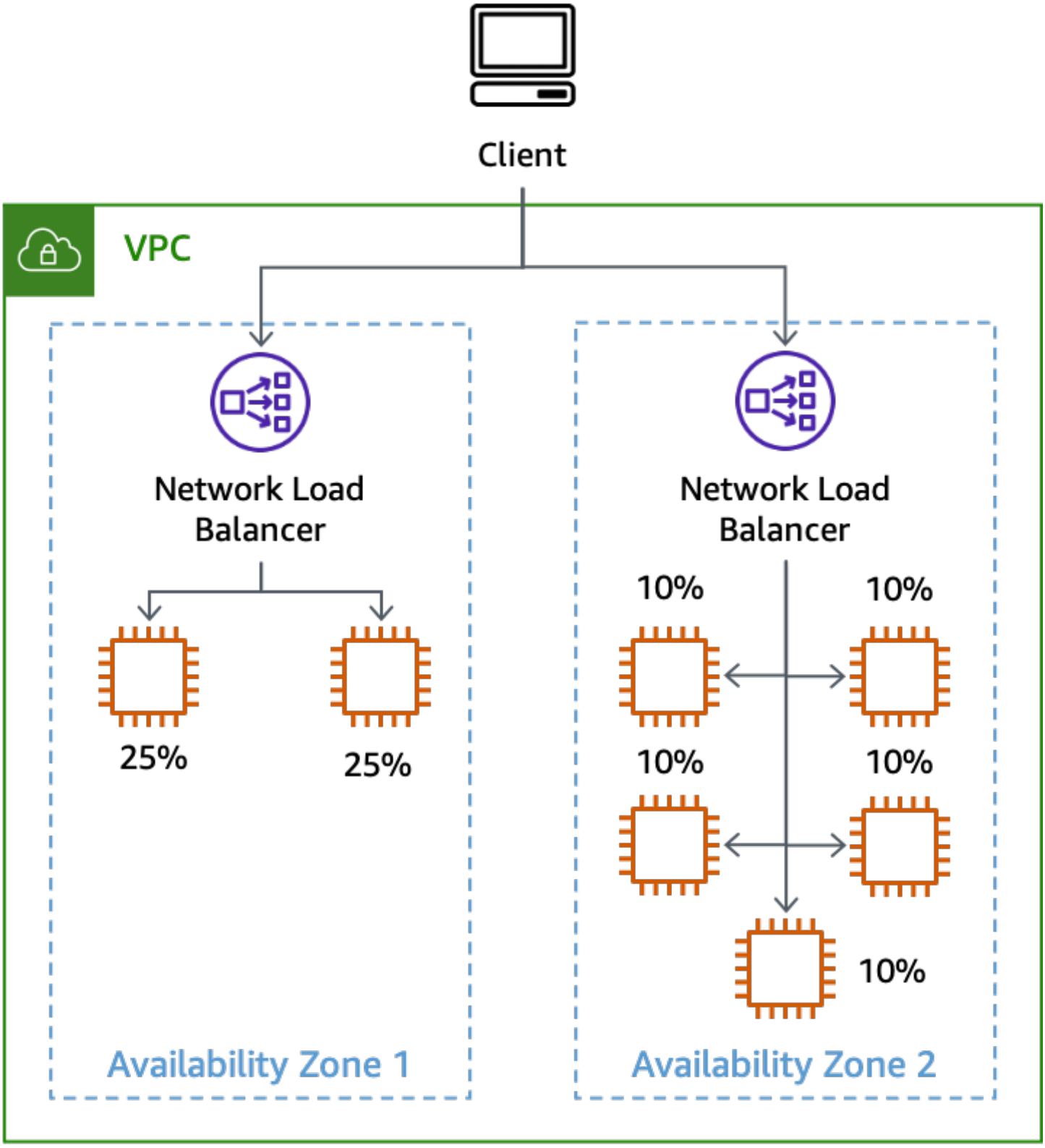
Argomenti

- Indipendenza dalla zona di disponibilità
- Piani di controllo e piani dati
- Evacuazione controllata dal piano dati
- Evacuazione controllata dal piano di controllo
- Riepilogo

Indipendenza dalla zona di disponibilità

Per ottenere il primo risultato, per interrompere l'invio di lavoro nella zona di disponibilità interessata, l'evacuazione richiede l'implementazioneIndipendenza dalla zona di disponibilità(AZI), chiamato anche a volteAffinità della zona di disponibilità. Questo modello architetturale isola le risorse all'interno di una zona di disponibilità e impedisce l'interazione tra risorse in diverse zone di disponibilità, tranne dove assolutamente necessario, come la connessione a un'istanza del database principale in una zona di disponibilità diversa.

In un carico di lavoro di tipo richiesta/risposta, l'implementazione di AZI richiededisabilitare il bilanciamento del carico tra zone per Application Load Balancer(CAMICE),Bilanciatori di carico classici(CLB) eBilanciatori del carico di rete(NLB) (il bilanciamento del carico tra zone è disabilitato per impostazione predefinita per gli NLB). La disattivazione del bilanciamento del carico tra zone presenta alcuni compromessi. Quando disabiliti il bilanciamento del carico tra zone,il traffico è equamente suddiviso tra ciascuna zona di disponibilitàindipendentemente dal numero di casi presenti in ciascuna di esse. Se disponi di risorse o gruppi Auto Scaling non bilanciati, ciò potrebbe comportare un carico aggiuntivo sulle risorse in una zona di disponibilità con meno risorse rispetto alle altre. Ciò è illustrato nella figura seguente, in cui due istanze nella zona di disponibilità 1 ricevono ciascuna il 25% del carico e le cinque istanze nella zona di disponibilità 2 ricevono ciascuna il 10% del carico.



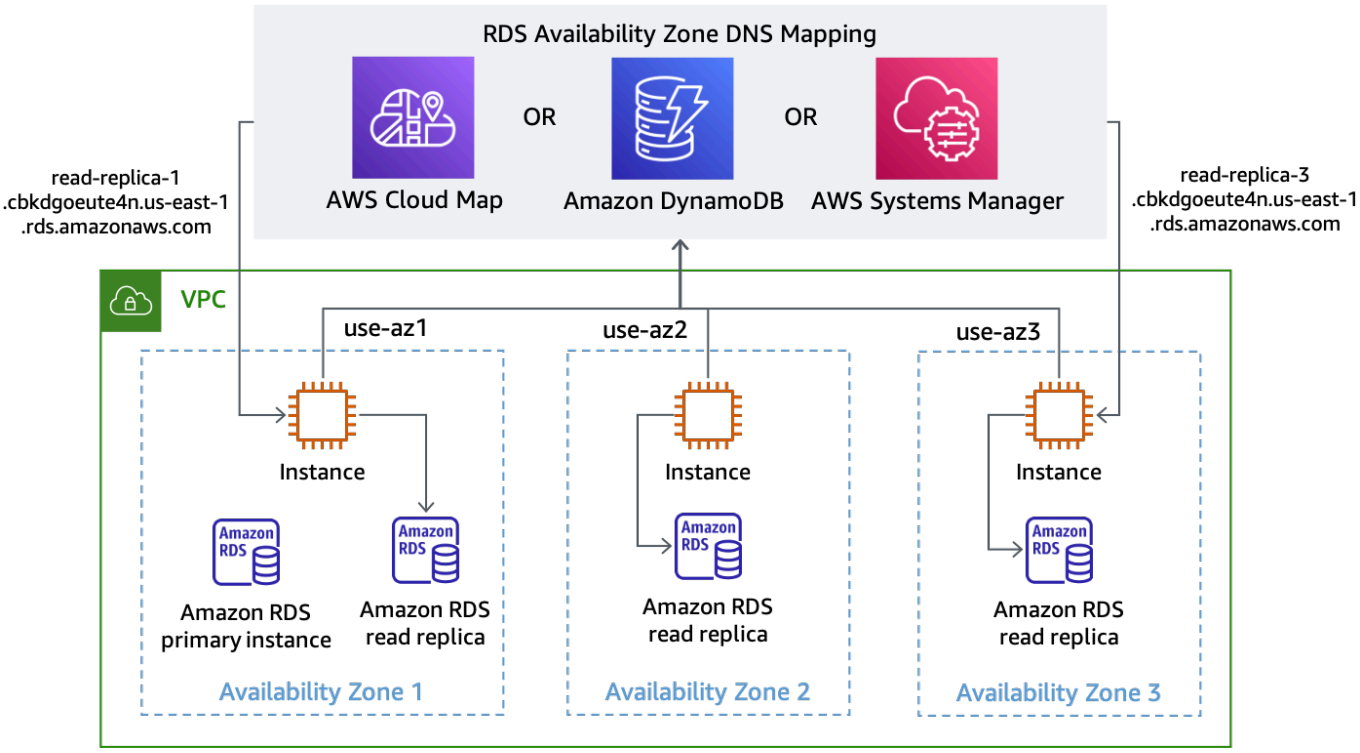
L'effetto della disabilitazione del bilanciamento del carico tra zone con istanze sbilanciate

Anche gli altri servizi zonali utilizzati dovranno essere implementati utilizzando modelli AZI per supportare un'efficace evacuazione della zona di disponibilità. Ad esempio, gli endpoint VPC di interfaccia forniscononomi DNS specifici per ogni zona di disponibilitàl'endpoint dell'interfaccia è reso disponibile in.

Una delle sfide legate all'implementazione di AZI riguarda i database, soprattutto perché la maggior parte dei database relazionali supporta solo un singolo scrittore primario in qualsiasi momento. Quando si comunica con l'istanza principale, potrebbe essere necessario oltrepassare un limite della zona di disponibilità. MoltiAWSi servizi di database supportano una configurazione Multi-AZ definita dall'utente e dispongono di una funzionalità di failover Multi-AZ integrata, ad esempioAmazon RDSoAmazon Aurora. In molti scenari di errore, il servizio è in grado di rilevare l'impatto e eseguire automaticamente il failover del database in una zona di disponibilità diversa quando si verifica un problema. Tuttavia, durante un errore grigio, il servizio potrebbe non rilevare l'impatto che influisce sul carico di lavoro o l'impatto potrebbe non essere affatto correlato al database. In questi casi, una volta rilevato l'impatto in una zona di disponibilità, è possibile richiamare manualmente un failover per spostare il database principale. Ciò consente di reagire in modo efficace a una singola limitazione della zona di disponibilità.

Se utilizzi repliche di lettura con quei database, potresti voler implementare AZI anche per questi perché non puoi eseguire il failover di una replica di lettura in una zona di disponibilità diversa come nel database principale. Se si dispone di una singola replica di lettura nella zona di disponibilità 1 e le istanze di tre zone di disponibilità sono configurate per utilizzarla, un problema che incide sulla zona di disponibilità 1 influirà anche sulle operazioni nelle altre due zone di disponibilità. Questo è l'impatto che vuoi prevenire.

Per le istanze RDS, ricevi un endpoint DNS per accedere alla replica in una zona di disponibilità specifica. Per ottenere l'AZI, è necessaria una replica di lettura per zona di disponibilità e un modo per consentire all'applicazione di sapere quale endpoint di replica utilizzare per la zona di disponibilità in cui si trova. Un approccio che puoi adottare consiste nell'utilizzare l'ID della zona di disponibilità come parte dell'identificatore del database, qualcosa come`use1-az1-read-replica.cbkdgoeute4n.us-east-1.rds.amazonaws.com`. Puoi farlo anche usando Service Discovery (ad esempio conAWS Cloud Map) o cercando una semplice mappa memorizzata inAWSArchivio parametri di Systems Manageroo una tabella DynamoDB. Questo concetto è illustrato nella figura riportata di seguito.

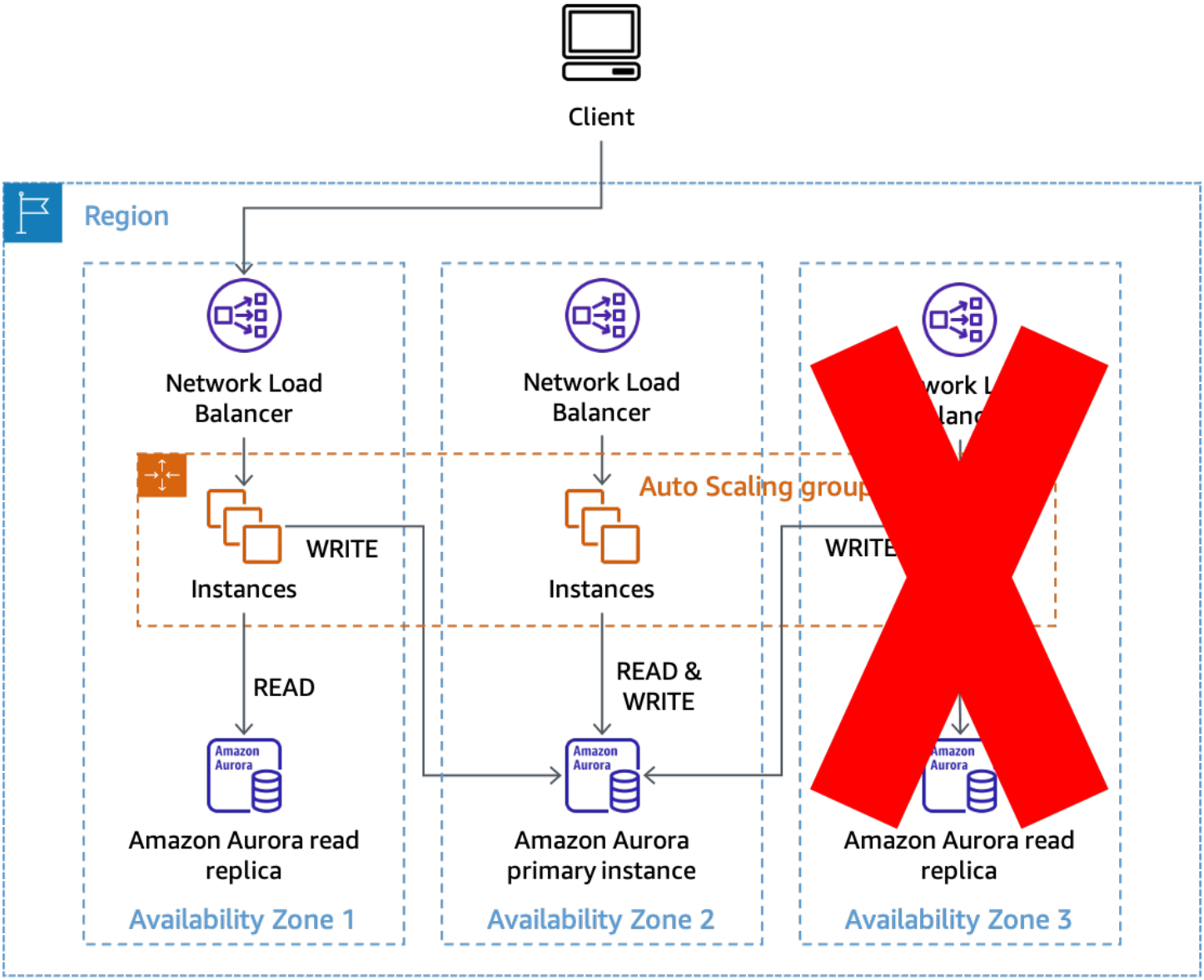


Individuazione dei nomi DNS degli endpoint RDS per ciascuna zona di disponibilità

La configurazione predefinita di Amazon Aurora è quella di fornire unendpoint con lettore singoloche bilancia il carico delle richieste tra le repliche di lettura disponibili. Per implementare AZI utilizzando Aurora, puoi usare unendpoint personalizzatoper ogni replica letta utilizzando ilANYdigitare (in modo da poter promuovere una replica letta, se necessario). Assegna un nome all'endpoint personalizzato in base all'ID della zona di disponibilità in cui viene distribuita la replica. Quindi, è possibile utilizzare il nome DNS fornito dall'endpoint personalizzato per connettersi a una replica di lettura specifica in una zona di disponibilità specifica, illustrata nella figura seguente.

Utilizzo di un endpoint personalizzato per una replica di lettura Aurora

Quando il sistema è progettato in questo modo, l'evacuazione della zona di disponibilità diventa un'operazione molto più semplice. Ad esempio, nella figura seguente, quando si verifica un deterioramento della zona di disponibilità 3, le operazioni di lettura e scrittura nelle zone di disponibilità 1 e 2 non sono interessate.



Utilizzo di AZI per prevenire l'impatto con le repliche di lettura di Amazon Aurora

In alternativa, se la zona di disponibilità 2 fosse interessata, le operazioni di lettura avrebbero comunque esito positivo nelle zone di disponibilità 1 e 3. Quindi, se Amazon Aurora non ha eseguito automaticamente un failover sul database primario, puoi richiamare manualmente un failover in una zona di disponibilità diversa per ripristinare la capacità di elaborazione delle scritture. Questo approccio evita la necessità di apportare modifiche alla configurazione delle connessioni al database quando è necessario evacuare una zona di disponibilità. Ridurre al minimo le modifiche richieste e mantenere il processo il più semplice possibile lo renderà più affidabile.

Piani di controllo e piani dati

Prima di passare agli schemi effettivi che è possibile utilizzare per eseguire l'evacuazione di una zona di disponibilità, dobbiamo discutere i concetti di piani di controllo e piani dati.AWSfa una distinzione tra piani di controllo e piani dati nei nostri servizi. I piani di controllo sono i meccanismi coinvolti nell'apportare modifiche a un sistema, aggiungendo risorse, eliminando risorse, modificando le risorse, e facendo sì che tali modifiche vengano propagate ovunque siano necessarie per avere effetto, ad esempio l'aggiornamento di una configurazione di rete per un ALB o la creazione di unAWS Lambdafunzione.

I piani dati sono la funzione principale di tali risorse, ad esempio l'esecuzione di un'istanza EC2 o la raccolta di elementi da una tabella Amazon DynamoDB o l'inserimento di elementi in una tabella Amazon DynamoDB. Per una discussione più dettagliata dei piani di controllo e dei piani dati, fare riferimento aStabilità statica tramite zone di disponibilitàeAWSLimiti di isolamento dei guasti.

Ai fini di questo documento, si consideri che i piani di controllo tendono ad avere più parti mobili e dipendenze rispetto ai piani dati. Ciò rende statisticamente più probabile che il piano di controllo venga compromesso rispetto al piano dati. Ciò è particolarmente importante per i servizi che forniscono AZI, come Amazon EC2 ed EBS, poiché alcuni di questi servizi dispongono di piani di controllo che sono anche indipendenti dal punto di vista zonale e possono essere influenzati durante un evento Single-AZ.

Sebbene le azioni del piano di controllo possano essere utilizzate per eseguire l'evacuazione AZ, sulla base delle informazioni precedenti, potrebbero avere una probabilità di successo inferiore, specialmente durante un evento di guasto. Per aumentare la probabilità di mitigare con successo l'impatto, è possibile utilizzare due modelli diversi. Il primo modello si basa solo sulle azioni sul piano dati per mitigare inizialmente l'impatto impedendo che il lavoro venga indirizzato o interrompendo il lavoro nella zona di disponibilità interessata. Quindi, è possibile tentare di aggiornare la configurazione delle risorse con azioni sul piano di controllo sia per impedire che la capacità venga fornita nella zona di disponibilità interessata sia per interrompere la comunicazione tra le zone di disponibilità con quella zona di disponibilità.

I modelli di ripristino discussi in questa sezione sonograndi bottoni rossi. Sono i meccanismi che usi per agire su larga scala, rapidamente, come tirare unE un cavo su una catena di montaggio. Presumono che i carichi di lavoro abbiano già provato strategie comeriprova con backoff esponenziale con jitternel loro codice per superare gli errori transitori. Ciò significa che quando viene rilevato un impatto isolato della zona di disponibilità, i suoi effetti sulla disponibilità o sulla latenza

sono sufficientemente gravi da richiedere l'evacuazione della zona di disponibilità per mitigarlo efficacemente.

Evacuazione controllata dal piano dati

Esistono diverse soluzioni che è possibile implementare per eseguire l'evacuazione di una zona di disponibilità utilizzando solo azioni sul piano dati. Questa sezione ne descriverà tre e i casi d'uso in cui potresti preferire l'uno rispetto all'altro.

Quando si utilizza una di queste soluzioni, è necessario assicurarsi di disporre di una capacità sufficiente nelle zone di disponibilità rimanenti per gestire il carico della zona di disponibilità da cui si sta allontanando. Il modo più resiliente per farlo è predisporre la capacità richiesta in ciascuna zona di disponibilità. Se utilizzi tre zone di disponibilità, in ognuna di esse disporrai del 50% della capacità richiesta per gestire il carico di picco, in modo che la perdita di una singola zona di disponibilità lasci comunque il 100% della capacità richiesta senza dover fare affidamento su un piano di controllo per fornirne di più.

Inoltre, se utilizzi EC2 Auto Scaling, assicurati che il tuo gruppo Auto Scaling (ASG) non effettui la scalabilità durante il turno, in modo che, al termine del turno, tu abbia ancora una capacità sufficiente nel gruppo per gestire il traffico dei tuoi clienti. Puoi farlo assicurandoti che la capacità minima desiderata del tuo ASG sia in grado di gestire il carico attuale di clienti. Puoi anche contribuire a garantire che il tuo ASG non venga ridimensionato inavvertitamente utilizzando medie nelle tue metriche anziché metriche percentili anomale come P90 o P99.

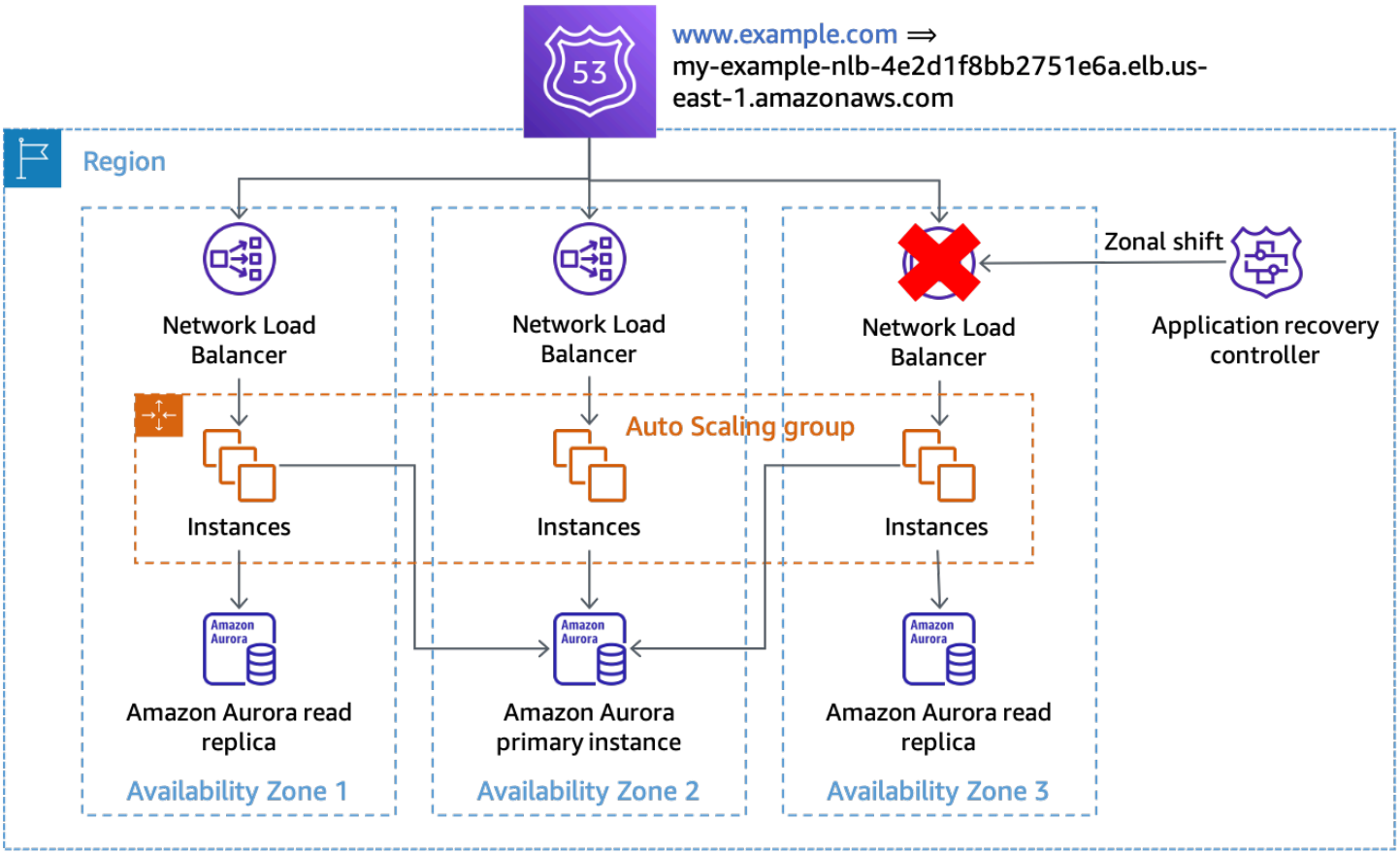
Durante un turno, le risorse che non servono più il traffico dovrebbero avere un utilizzo molto basso, ma le altre risorse aumenteranno il loro utilizzo con il nuovo traffico, mantenendo la media abbastanza costante, il che impedirebbe un'azione di scalabilità. Infine, puoi anche utilizzare le impostazioni sanitarie del gruppo target perCAMICEeNLBper specificare il failover DNS con una percentuale o un numero di host sani. Questo impedisce che il traffico venga indirizzato verso una zona di disponibilità che non dispone di un numero sufficiente di host sani.

Cambio di zona in Route 53 Application Recovery Controller (ARC)

La prima soluzione per gli usi di evacuazione delle zone di disponibilitàspostamento zonale nella Route 53 ARC. Questa soluzione può essere utilizzata per carichi di lavoro di richiesta/risposta che utilizzano un NLB o ALB come punto di ingresso per il traffico dei clienti.

Quando rilevi che una zona di disponibilità è compromessa, puoi avviare un cambio di zona con Route 53 ARC. Una volta completata questa operazione e scadute le risposte DNS memorizzate

nella cache esistenti, tutte le nuove richieste vengono indirizzate solo alle risorse nelle restanti zone di disponibilità. La figura seguente mostra come funziona lo spostamento zonale. Nella figura seguente abbiamo un record di alias Route 53 per`www.example.com`che indica`my-example-nlb-4e2d1f8bb2751e6a.elb.us-east-1.amazonaws.com`. Lo spostamento zonale viene eseguito per la zona di disponibilità 3.



Spostamento zonale

Nell'esempio, se l'istanza del database principale non si trova nella zona di disponibilità 3, l'esecuzione dello spostamento zonale è l'unica azione richiesta per ottenere il primo risultato dell'evacuazione, impedendo l'elaborazione del lavoro nella zona di disponibilità interessata. Se il nodo principale si trovava nella zona di disponibilità 3, puoi eseguire un failover avviato manualmente (che si basa sul piano di controllo di Amazon RDS) in coordinamento con il cambio di zona, se Amazon RDS non ha già eseguito il failover automaticamente. Questo sarà vero per tutte le soluzioni controllate dal piano dati in questa sezione.

È necessario avviare lo spostamento zonale utilizzando i comandi CLI o l'API per ridurre al minimo le dipendenze necessarie per avviare l'evacuazione. Più semplice è il processo di evacuazione, più affidabile sarà. I comandi specifici possono essere memorizzati in un runbook locale a cui i tecnici

di guardia possono accedere facilmente. Lo spostamento zonale è la soluzione più preferita e più semplice per evacuare una zona di disponibilità.

Route 53 ARC

La seconda soluzione utilizza le funzionalità di Route 53 ARC per specificare manualmente lo stato di salute di record DNS specifici. Questa soluzione ha il vantaggio di utilizzare il piano dati del cluster Route 53 ARC ad alta disponibilità, rendendola resiliente alla compromissione di un massimo di due diversiRegioni AWS. Ha il compromesso di costi aggiuntivi e richiede una configurazione aggiuntiva dei record DNS. Per implementare questo modello, è necessario creare record di alias perNomi DNS specifici della zona di disponibilitàfornito dal sistema di bilanciamento del carico (ALB o NLB). Questo è illustrato nella tabella seguente.

Tabella 3: record di alias Route 53 configurati per i nomi DNS zonali del load balancer

Politica di routing: ponderato	Politica di routing:ponderata	Politica di routing:ponderata
Nome: `www.example.com`	Nome: `www.example.com`	Nome: `www.example.com`
Tipo:`A`(alias)	Tipo: `A`(alias)	Tipo: `A`(alias)
Value (Valore): `us-east-1b.load-balancer-name.elb.us-east-1.amazonaws.com`	Valore: `us-east-1a.load-balancer-name.elb.us-east-1.amazonaws.com`	Valore: `us-east-1c.load-balancer-name.elb.us-east-1.amazonaws.com`
Peso:`100`	Peso: `100`	Peso: `100`
Valuta la salute di Target: vero	Valuta la salute di Target: `true`	Valuta la salute di Target: `true`

Per ognuno di questi record DNS, è necessario configurare un controllo dello stato di Route 53 associato a un Route 53 ARCcontrollo del routing. Quando desideri avviare un'evacuazione della zona di disponibilità, imposta lo stato di controllo del routing su`Off`.AWSconsiglia di eseguire questa operazione utilizzando la CLI o l'API per ridurre al minimo le dipendenze necessarie per avviare l'evacuazione della zona di disponibilità. Come unbest practice, è necessario conservare una copia locale degli endpoint del cluster Route 53 ARC in modo da non doverli recuperare dal piano di controllo ARC quando è necessario eseguire un'evacuazione.

Per ridurre al minimo i costi quando si utilizza questo approccio, è possibile creare un singolo cluster Route 53 ARC e controllare lo stato di salute in un'unica soluzioneAccount AWSecondividere i controlli sanitari con altriAccount AWSnella tua organizzazione. Quando si adotta questo approccio, è necessario utilizzareID della zona di disponibilità(AZ-ID) (ad esempio,`use1-az1`) anziché il nome della zona di disponibilità (ad esempio,`us-east-1a`) per i controlli di routing. PerchéAWSassocia la zona di disponibilità fisica in modo casuale ai nomi delle zone di disponibilità per ciascunaAccount AWS, l'utilizzo dell'AZ-ID fornisce un modo coerente per fare riferimento alle stesse posizioni fisiche. Quando avviate l'evacuazione di una zona di disponibilità, dite`use1-az2`, i record della Route 53 in ciascunoAccount AWSdovrebbero assicurarsi di utilizzare la mappatura AZ-ID per configurare il corretto controllo dello stato di salute per ogni record NLB.

Ad esempio, supponiamo di avere un controllo di integrità della Route 53 associato a un controllo di routing Route 53 ARC per`use1-az2`, con un ID di`0385ed2d-d65c-4f63-a19b-2412a31ef431`. Se in un altroAccount AWSche desidera utilizzare questo controllo sanitario,`us-east-1c`è stato mappato su`use1-az2`, dovresti usare il`use1-az2`controllo sanitario per la registrazione`us-east-1c.load-balancer-name.elb.us-east-1.amazonaws.com`. Utilizzeresti l'ID del controllo sanitario`0385ed2d-d65c-4f63-a19b-2412a31ef431`con quel set di record di risorse.

Utilizzo di un endpoint HTTP autogestito

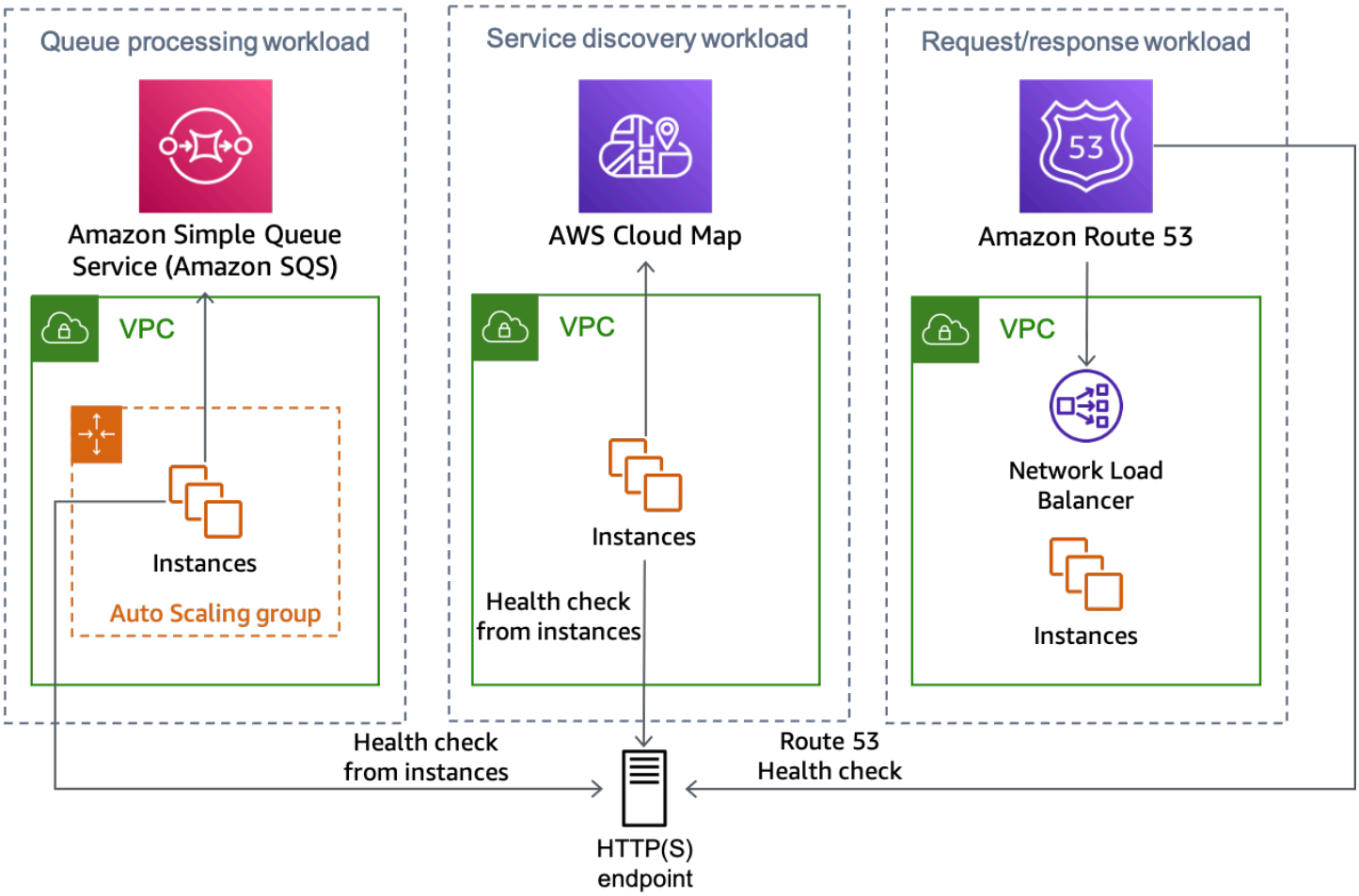
È inoltre possibile implementare questa soluzione gestendo il proprio endpoint HTTP che indica lo stato di una particolare zona di disponibilità. Consente di specificare manualmente quando una zona di disponibilità non è integra in base alla risposta dell'endpoint HTTP. Questa soluzione costa meno rispetto all'utilizzo di Route 53 ARC, ma è più costosa del trasferimento zonale e richiede la gestione di un'infrastruttura aggiuntiva. Ha il vantaggio di essere molto più flessibile per diversi scenari.

Il pattern può essere utilizzato con architetture NLB o ALB e controlli di integrità della Route 53. Può essere utilizzato anche in architetture non bilanciate dal carico, come i sistemi di rilevamento dei servizi o di elaborazione delle code in cui i nodi di lavoro eseguono i propri controlli di integrità. In questi scenari, gli host possono utilizzare un thread in background in cui effettuano periodicamente una richiesta all'endpoint HTTP con il proprio AZ-ID (fare riferimento aAppendice A — Ottenere l'ID della zona di disponibilitàper dettagli su come trovarlo) e ricevere una risposta sullo stato della zona di disponibilità.

Se la zona di disponibilità è stata dichiarata non integra, hanno diverse opzioni su come rispondere. Possono scegliere di non superare un controllo dello stato esterno proveniente da fonti come ELB, Route 53 o controlli di integrità personalizzati nelle architetture di Service Discovery, in modo da apparire non integri a tali servizi. Possono anche rispondere immediatamente con un errore se

ricevono una richiesta, consentendo al client di fare marcia indietro e riprovare. Nelle architetture basate su eventi, i nodi possono intenzionalmente non riuscire a elaborare il lavoro, ad esempio restituendo intenzionalmente un messaggio SQS alla coda. Nelle architetture di router di lavoro in cui un servizio centrale pianifica il lavoro su host specifici, è possibile utilizzare anche questo modello. Il router può verificare lo stato di una zona di disponibilità prima di selezionare un worker, un endpoint o una cella. Architetture di ricerca dei servizi che utilizzanoAWS Cloud Map, puoiscopri gli endpoint fornendo un filtro nella tua richiesta, ad esempio un AZ-ID.

La figura seguente mostra come questo approccio può essere utilizzato per più tipi di carichi di lavoro.



La soluzione endpoint HTTP può essere utilizzata da più tipi di carico di lavoro

Esistono diversi modi per implementare l'approccio degli endpoint HTTP, due dei quali sono descritti di seguito.

Uso di Amazon S3

Questo modello è stato originariamente presentato in questopost sul blogper il disaster recovery multiregionale. È possibile utilizzare lo stesso schema per l'evacuazione della zona di disponibilità.

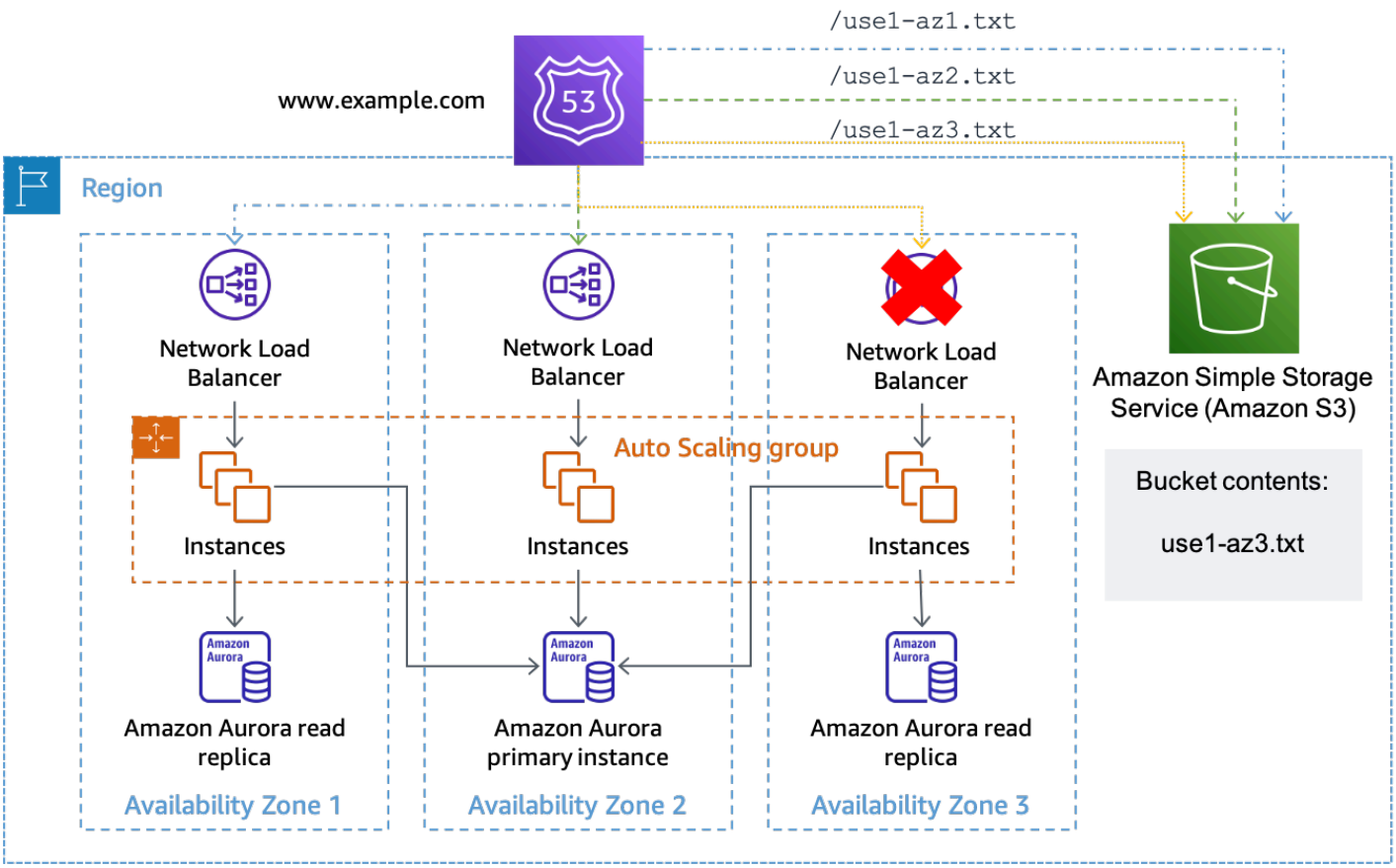
In questo scenario, creeresti set di record di risorse DNS Route 53 per ogni record DNS zonale proprio comePercorso 53 ARCscenario precedente e relativi controlli sanitari. Tuttavia, per questa implementazione, invece di associare i controlli di integrità ai controlli di routing ARC della Route 53, sono configurati per utilizzare unEndpoint HTTPe sono invertiti per evitare che un guasto in Amazon S3 provochi accidentalmente un'evacuazione. Il controllo sanitario è consideratosalutarequando l'oggetto è assente emalsanoquando l'oggetto è presente. Questa configurazione è mostrata nella tabella seguente.

Tabella 4: Configurazione dei record DNS per l'utilizzo dei controlli di integrità della Route 53 per zona di disponibilità

Tipo di controllo sanitario:	Tipo di controllo sanitario:	Tipo di controllo sanitario:		
monitorare un endpoint	monitorare un endpoint	monitorare un endpoint		
Protocol (Protocollo): HTTPS	Protocol (Protocollo): HTTPS	Protocol (Protocollo): HTTPS		
ID:dddd-4444	ID:eeee-5555	ID:ffff-6666	←	Controlli sanitari
URL:https://*bucket name* .s3.us-east-1.amazonaws.com/use1-az1.txt	URL: https://*bucket name* .s3.us-east-1.amazonaws.com/use1-az3.txt	URL:https://*bucket name* .s3.us-east-1.amazonaws.com/use1-az2.txt		
↑	↑	↑		

Politica di routing: ponderato Nome: `www.example.com` Tipo:A(alias) Value (Valore): `us-east-1b.load-balancer-name.elb.us-east-1.amazonaws.com` Peso:`100` Valuta la salute di Target:`true`	Politica di routing:ponderata Nome: `www.example.com` Tipo: A(alias) Value (Valore): `us-east-1a.load-balancer-name.elb.us-east-1.amazonaws.com` Peso: `100` Valuta la salute di Target: `true`	Politica di routing:ponderata Nome: `www.example.com` Tipo: A(alias) Value (Valore): `us-east-1c.load-balancer-name.elb.us-east-1.amazonaws.com` Peso: `100` Valuta la salute di Target: `true`	←	I record di alias A di livello superiore con ponderazione uniforme puntano a endpoint specifici di NLB AZ

Supponiamo che la zona di disponibilità`us-east-1a`è mappato su`use1-az3`nell'account in cui abbiamo un carico di lavoro in cui vogliamo eseguire un'evacuazione della zona di disponibilità. Per il set di record di risorse creato per`us-east-1a.load-balancer-name.elb.us-east-1.amazonaws.com`associerebbe un controllo dello stato di salute che verifica l'URL`https://bucket-name.s3.us-east-1.amazonaws.com/use1-az3.txt`. Quando si desidera avviare un'evacuazione della zona di disponibilità per`use1-az3`, carica un file denominato`use1-az3.txt`al bucket utilizzando la CLI o l'API. Il file non deve contenere alcun contenuto, ma deve essere pubblico in modo che il controllo sanitario di Route 53 possa accedervi. La figura seguente mostra che questa implementazione viene utilizzata per evacuare`use1-az3`.



Utilizzo di Amazon S3 come obiettivo per un controllo dello stato della Route 53

Utilizzo di API Gateway e DynamoDB

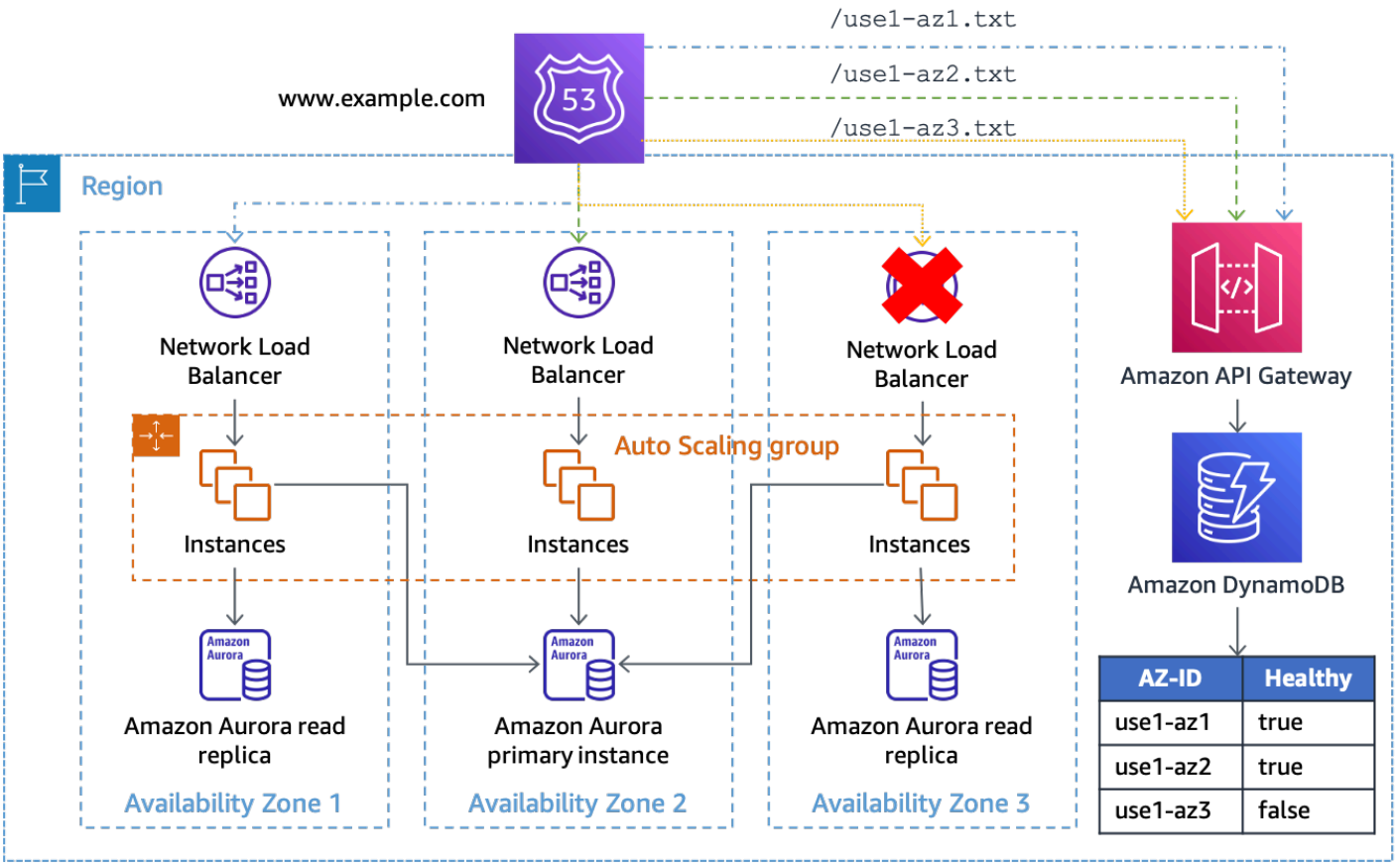
La seconda implementazione di questo modello utilizza unGateway API Amazon RESETTARE L'API. L'API è configurata con unintegrazione dei serviziad Amazon DynamoDB, dove è archiviato lo stato di ogni zona di disponibilità in uso. Questa implementazione è più flessibile dell'approccio Amazon S3, ma richiede la creazione, il funzionamento e il monitoraggio di una maggiore infrastruttura. Può anche essere utilizzato sia con i controlli sanitari di Route 53 che con i controlli sanitari eseguiti dai singoli host.

Se utilizzi questa soluzione con un'architettura NLB o ALB, configura i tuoi record DNS nello stesso modo dell'esempio di Amazon S3 sopra riportato, tranne modificare il percorso di controllo dello stato per utilizzare l'endpoint API Gateway e fornire il`AZ-ID`nel percorso dell'URL. Ad esempio, se l'API Gateway è configurato con un dominio personalizzato di`az-status.example.com`, la richiesta completa di`use1-az1`assomiglierebbe`https://az-status.example.com/status/use1-az1`. Quando desideri avviare un'evacuazione della zona di disponibilità, puoi creare o aggiornare un elemento DynamoDB utilizzando l'interfaccia a riga di comando o l'API. L'articolo utilizza il`AZ-`

`ID`come chiave primaria e quindi ha un attributo booleano chiamato`Healthy`che viene utilizzato indica come risponde API Gateway. Di seguito è riportato un esempio di codice utilizzato nella configurazione di API Gateway per effettuare questa determinazione.

```
#set($inputRoot = $input.path('$'))
#if ($inputRoot.Item.Healthy['BOOL'] == (false))
    #set($context.responseOverride.status = 500)
#end
```

Se l'attributo è`true`(o non è presente), API Gateway risponde al controllo dello stato con un HTTP 200, se è falso, risponde con un HTTP 500. Questa implementazione è illustrata nella figura seguente.



Utilizzo di API Gateway e DynamoDB come obiettivo dei controlli di integrità della Route 53

In questa soluzione è necessario utilizzare API Gateway davanti a DynamoDB in modo da poter rendere l'endpoint accessibile al pubblico e manipolare l'URL della richiesta in un`GetItem`richiesta per DynamoDB. La soluzione offre anche flessibilità se si desidera includere dati aggiuntivi nella richiesta. Ad esempio, se desideri creare stati più granulari, ad esempio per applicazione, puoi

configurare l'URL del controllo dello stato per fornire un ID dell'applicazione nel percorso o nella stringa di query che sia anche confrontato con l'elemento DynamoDB.

L'endpoint di stato della zona di disponibilità può essere distribuito centralmente in modo che più risorse di controllo dello stato di salute siano distribuite su tutto il territorioAccount AWSpossono tutti utilizzare la stessa visione coerente dello stato della zona di disponibilità (assicurando che l'API REST di API Gateway e la tabella DynamoDB siano scalate per gestire il carico) ed elimina la necessità di condividere i controlli di integrità della Route 53.

La soluzione può anche essere scalata su più livelliRegioni AWSutilizzando unTabella globale Amazon DynamoDBe una copia dell'API REST di API Gateway in ogni regione. Ciò impedisce a questa soluzione di dipendere da una singola regione e ne aumenta la disponibilità. È possibile implementare la soluzione in tre o cinque regioni e richiedere a ciascuna di esse lo stato della zona di disponibilità, utilizzando il risultato della maggior parte degli endpoint per garantire il quorum. Ciò consente una replica coerente degli aggiornamenti in tutta la tabella globale e mitiga i problemi che potrebbero impedire a un endpoint di rispondere. Ad esempio, se utilizzi cinque regioni e tre endpoint segnalano una zona di disponibilità come non integra, un endpoint segnala la zona di disponibilità come integra e un endpoint non risponde, sceglierai di trattare la zona di disponibilità come non integra. Puoi anche creare unControllo sanitario calcolato da Route 53utilizzando unm di ncalcoloper eseguire questa logica per determinare lo stato della zona di disponibilità.

Se stavi creando una soluzione per i singoli host da utilizzare come meccanismo per determinare lo stato della loro AZ, in alternativa, invece di fornire un meccanismo di richiamo per i controlli dello stato, puoi utilizzare le notifiche push. Un modo per farlo è utilizzare un argomento SNS a cui i tuoi consumatori sono abbonati. Quando desideri attivare l'interruttore automatico, pubblica un messaggio nell'argomento SNS indicando quale zona di disponibilità è compromessa. Questo approccio fa dei compromessi con il primo. Elimina la necessità di creare e gestire l'infrastruttura API Gateway ed eseguire la gestione della capacità. Può anche fornire una convergenza più rapida dello stato della zona di disponibilità. Tuttavia, elimina la possibilità di eseguire interrogazioni ad hoc e si basa suPolitica dei tentativi di consegna SNSper garantire che ogni endpoint riceva la notifica. Richiede inoltre che ogni carico di lavoro o servizio crei un modo per ricevere la notifica SNS e intervenire in merito.

Ad esempio, ogni nuova istanza o contenitore EC2 che viene lanciato dovrà sottoscrivere l'argomento con un endpoint HTTP durante il bootstrap. Quindi, ogni istanza deve implementare un software che ascolti l'endpoint da cui viene recapitata la notifica. Inoltre, se l'istanza è interessata dall'evento, potrebbe non ricevere la notifica push e continuare a funzionare. Invece, con una notifica pull,

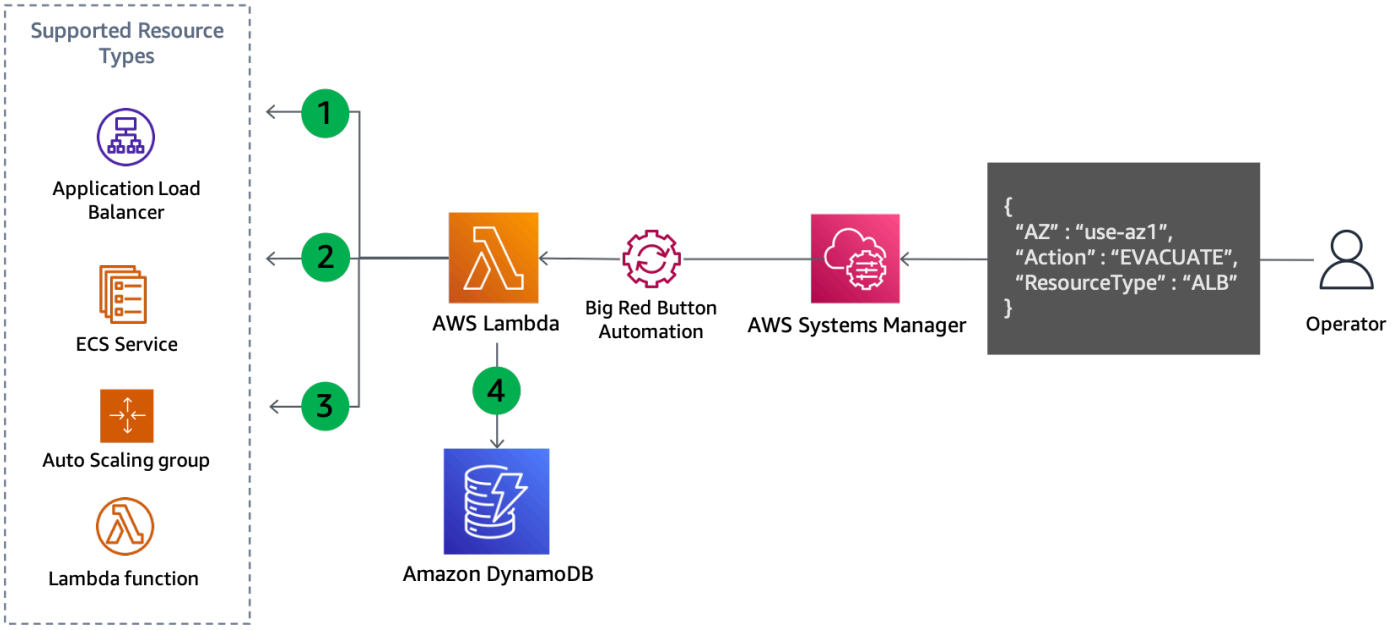
l'istanza saprà se la sua richiesta di pull fallisce e potrà scegliere quale azione intraprendere in risposta.

Un secondo modo per inviare notifiche push è quello di lunga durataWebSocketconnessioni. Amazon API Gateway può essere utilizzato per fornire unWebSocketAPIa cui i consumatori possono connettersi e ricevere un messaggio quandoinviato dal backend. Con unWebSocket, le istanze possono sia eseguire interruzioni periodiche per garantire che la connessione sia funzionante sia ricevere notifiche push a bassa latenza.

Evacuazione controllata dal piano di controllo

Il primo modello utilizza le operazioni sul piano dati per impedire l'esecuzione di operazioni in una zona di disponibilità interessata per mitigare l'impatto di un evento. Tuttavia, è possibile che tu stia utilizzando un'architettura che non utilizza sistemi di bilanciamento del carico o in cui la configurazione di un controllo dello stato per host non è possibile. Oppure, potresti voler impedire che nuove capacità vengano distribuite nella zona di disponibilità interessata tramite Auto Scaling o la normale pianificazione del lavoro.

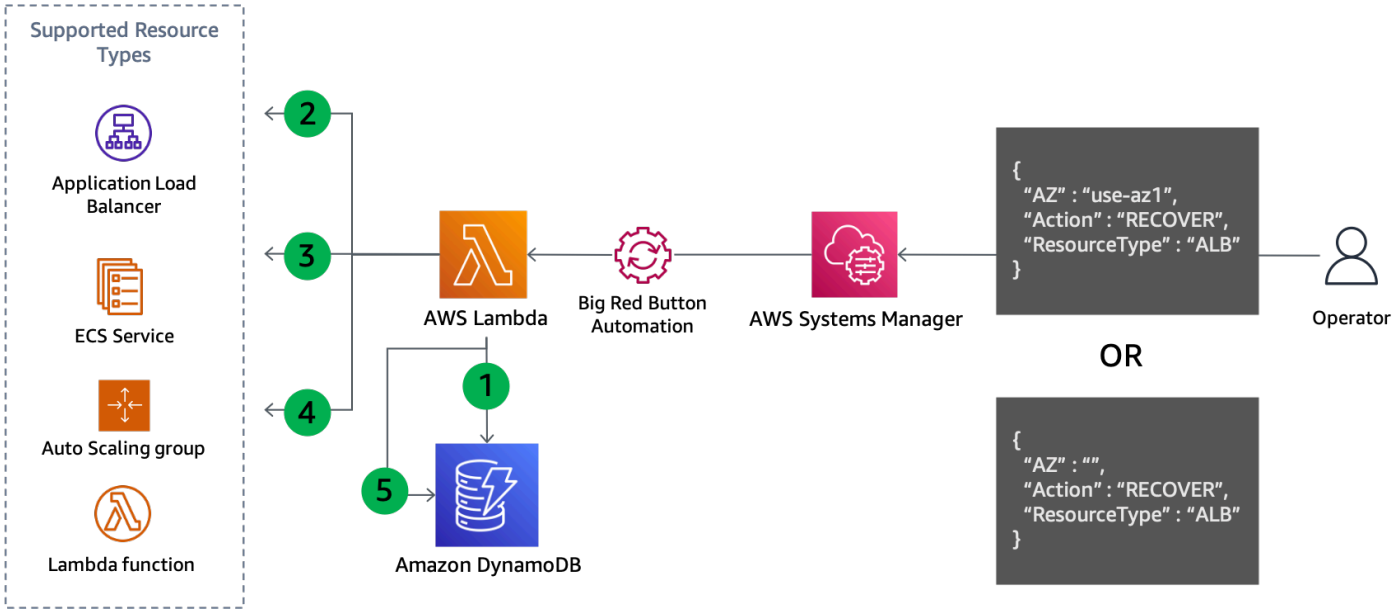
Per risolvere entrambe le situazioni, sono necessarie le azioni del piano di controllo per aggiornare la configurazione della risorsa. Il modello funzionerà per qualsiasi servizio la cui configurazione di rete può essere aggiornata, ad esempio EC2 Auto Scaling, Amazon ECS, Lambda e altri. Richiede la scrittura di codice per ogni servizio, ma la logica aziendale segue uno schema standard. Il codice deve essere eseguito localmente da un operatore che risponde all'evento per ridurre al minimo le dipendenze richieste. Il flusso di base della logica dello script è illustrato nella figura seguente.



Aggiornamento del piano di controllo per evacuare una zona di disponibilità

1. Lo script elenca tutte le risorse del tipo specificato, come il gruppo Auto Scaling, il servizio ECS o la funzione Lambda, e recupera le relative sottoreti dalle informazioni sulle risorse. Le risorse supportate dipendono da ciò che lo script è stato configurato per supportare.
2. Determina quali sottoreti devono essere rimosse confrontando il nome della zona di disponibilità di ciascuna sottorete con l'ID della zona di disponibilità mappato fornito come parametro di input.
3. La configurazione di rete della risorsa viene aggiornata per rimuovere le sottoreti identificate.
4. I dettagli dell'aggiornamento vengono registrati in una tabella DynamoDB. L'ID della zona di disponibilità è memorizzato comechiave di partizionee l'ARN o il nome della risorsa viene memorizzato comechiave di ordinamento. Le sottoreti rimosse vengono archiviate come un array di stringhe. Infine, il tipo di risorsa viene anche memorizzato e utilizzato come chiave hash per unIndice secondario globale(GSI).

Poiché la fase quattro registra gli aggiornamenti effettuati, questo approccio si presta anche a essere facilmente reversibile quando si è pronti per il ripristino, come illustrato nella figura seguente.



Aggiornamento del piano di controllo per il ripristino dopo l'evacuazione della zona di disponibilità

Fasi di ripristino:

1. Interroga il GSI per rimuovere le sottoreti per ogni risorsa del tipo specificato nella zona di disponibilità specificata (o tutte le zone di disponibilità se non ne è specificata una).

2. Descrivi ogni risorsa trovata nella query DynamoDB per ottenere la sua configurazione di rete corrente.
3. Combina le sottoreti della configurazione di rete corrente con quelle recuperate dalla query DynamoDB.
4. Aggiorna la configurazione di rete della risorsa con il nuovo set di sottoreti.
5. Rimuove il record dalla tabella DynamoDB dopo il completamento dell'aggiornamento.

Questo modello generalizzato impedisce sia il routing del lavoro verso la zona di disponibilità interessata sia l'installazione di nuove capacità in tale zona. Di seguito sono riportati alcuni esempi di come questa operazione viene eseguita per diversi servizi.

- Lambda— Aggiorna la funzioneConfigurazione VPCper rimuovere le sottoreti nella zona di disponibilità specificata.
- Gruppo Auto Scaling—Rimuovi le sottoreti dalla configurazione ASGche sostituirà tale capacità nelle restanti zone di disponibilità.
- Amazon ECS—Aggiorna la configurazione VPC del servizio ECSper rimuovere le sottoreti.
- Amazon EKS— Applicacontaminazioniai nodi della zona di disponibilità interessata per eliminare i pod esistenti e impedire che vengano programmati altri pod lì.

Ogni servizio reagirà in modo diverso all'aggiornamento della configurazione. Ad esempio, Amazon ECS seguiràconfigurazione di distribuzione del servizio dopo un aggiornamentoe avvia una distribuzione continua o una distribuzione blu/verde di nuove attività.

Questi aggiornamenti potrebbero spostare il lavoro nelle zone di disponibilità intatte troppo rapidamente per alcuni carichi di lavoro. Pur essendo configurato per essere staticamente stabile al guasto (con una capacità sufficiente preconfigurata nelle zone di disponibilità rimanenti per gestire il lavoro della zona di disponibilità interessata), potresti anche voler eliminare gradualmente la capacità dalla zona di disponibilità interessata.

> Se prevedi di aggiornare la configurazione di rete del tuo gruppo Auto Scaling, questo è un gruppo target per un sistema di bilanciamento del carico con bilanciamento del carico tra zonedisabili, segui questa guida.
>
> Auto Scaling reagisce a questa modifica utilizzando ilLogica di ribilanciamento della zona di disponibilità. Avvierà le istanze nelle altre zone di disponibilità per soddisfare la capacità desiderata e terminerà le istanze nella zona di disponibilità rimossa. Tuttavia, il load balancer

continuerà a suddividere il traffico in modo uniforme in ogni zona di disponibilità, inclusa quella rimossa dall'ASG, mentre le istanze vengono terminate. Ciò potrebbe comportare un esaurimento della capacità residua in quella zona di disponibilità fino a quando tutte le istanze non saranno terminate correttamente. Questo è lo stesso problema descritto inIndipendenza dalla zona di disponibilitàrelativo allo squilibrio della zona di disponibilità quando il bilanciamento del carico tra zone è disabilitato. Per evitare che ciò accada, puoi:

- Esegui sempre prima l'evacuazione della zona di disponibilità, in modo che il traffico venga suddiviso solo tra le zone di disponibilità rimanenti
- Specifica unnumero minimo di obiettivi sani con failover DNSper corrispondere al numero obiettivo minimo richiesto per quella zona di disponibilità.

Ciò contribuirà a garantire che il traffico non venga inviato alla zona di disponibilità che hai rimosso dopo l'inizio della chiusura delle istanze.

Riepilogo

La tabella seguente riassume i pro e i contro dei modelli di evacuazione descritti.

Tabella 5: Pro e contro del modello di evacuazione

Approccio	Professionisti	Svantaggi
Evacuazione controllata dal piano dati	Si basa solo sulle azioni del piano dati Impedisce rapidamente che il lavoro venga svolto nella zona di disponibilità interessata Approccio flessibile a una visione centralizzata dello stato della zona di disponibilità	Non impedisce la distribuzione della capacità in una zona di disponibilità interessata Non tutti i tipi di carico di lavoro possono utilizzare facilmente questo approccio
Evacuazione controllata dal piano di controllo	Impedisce l'implementazione di nuove capacità nella zona di disponibilità interessata	Si basa sul piano di controllo di ogni servizio

Approccio	Professionisti	Svantaggi
	Rimuove la capacità esistente dalla zona di disponibilità interessata	Richiede la scrittura di codice per ogni servizio Deve essere completato servizio per servizio Deve fare attenzione a non sovraccaricare la capacità durante l'aggiornamento

Probabilmente utilizzerai entrambi gli approcci insieme come parte di un piano di evacuazione della zona di disponibilità. Inizia con le azioni di evacuazione controllate dal piano dati che hanno maggiori probabilità di successo per interrompere rapidamente il lavoro di elaborazione nella zona di disponibilità interessata. Quindi, una volta mitigato l'impatto iniziale, prosegui con le azioni di evacuazione controllate dal piano di controllo, se lo ritieni necessario.

Conclusioni

Questo documento ha fornito una panoramica dei guasti grigi, di come si manifestano e ha spiegato perché è necessario creare strumenti di osservabilità ed evacuazione per mitigare questi tipi di eventi quando si verificano. Nella sezione successiva, hai esaminato l'osservabilità Multi-AZ e tre approcci che puoi implementare per rilevare l'impatto di una singola zona di disponibilità. Nell'ultima sezione, questo documento ha presentato due approcci generali per eseguire l'evacuazione delle zone di disponibilità. Il primo approccio utilizza azioni sul piano dati per impedire che il lavoro venga indirizzato alla zona di disponibilità interessata, mentre il secondo approccio utilizza azioni sul piano di controllo per impedire che la capacità venga assegnata nella zona di disponibilità interessata. Insieme, questi due approcci raggiungono i due risultati che l'evacuazione della zona di disponibilità intende.

I modelli di ripristino descritti in questo documento faranno probabilmente parte di una più ampia soluzione di monitoraggio e ripristino dei guasti. Questo approccio alla gestione dei guasti nelle Single Availability Zone Grey richiede interventi di ingegneria per costruire la strumentazione necessaria a rilevarli e gli strumenti per affrontarli. Tuttavia, per molti carichi di lavoro, questo approccio può essere un'alternativa più semplice e meno costosa alla creazione di architetture multiregionali. Inoltre, può contribuire a ottenere RPO e RTO più piccoli (il che aumenta la disponibilità del carico di lavoro) rispetto al DR. multiregionale

Appendice A — Ottenere l'ID della zona di disponibilità

Se si utilizza ilAWS.NET SDK (così come altri similiJavaScript) o eseguendo il sistema su un'istanza EC2 (inclusi Amazon ECS e Amazon EKS), puoi ottenere direttamente l'ID della zona di disponibilità.

- AWS.NET SDK

```
Amazon.Util.EC2InstanceMetadata.GetData(“/placement/availability-zone-id”)
```

- Servizio di metadati delle istanze EC2

```
curl http://169.254.169.254/latest/meta-data/placement/availability-zone-id
```

Su altre piattaforme, come Lambda e Fargate, dovrai recuperare il nome della zona di disponibilità e quindi trovare la mappatura con l'ID della zona di disponibilità. Con il nome della zona di disponibilità puoi trovare l'ID della zona di disponibilità in questo modo:

```
aws ec2 describe-availability-zones --zone-names $AZ --output json
    --query ‘AvailabilityZones[0].ZoneId’
```

I seguenti esempi per trovare il nome della zona di disponibilità da utilizzare nell'esempio precedente sono scritti in bash usando ilAWS CLIe il pacchettojq. Dovranno essere convertiti nel linguaggio di programmazione utilizzato per il carico di lavoro.

- Amazon ECS- Se l'Instance Metadata Service (IMDS) è bloccato dall'host, puoi invece utilizzare il file di metadati del contenitore.

```
AZ=$(cat $ECS_CONTAINER_METADATA_FILE | jq --raw-output
    .AvailabilityZone)
```

- Fargate(versione della piattaforma 1.4 o successiva)

```
AZ=$(curl $ECS_CONTAINER_METADATA_URI_V4/task | jq --raw-output
    .AvailabilityZone)
```

- Lambda— La zona di disponibilità non è esposta direttamente alla funzione. Per trovarlo, devi completare diversi passaggi. Per fare ciò, dovrai creare un endpoint REST API Gateway privato che restituisca l'indirizzo IP del richiedente. Questo identificherà l'IP privato assegnato all'interfaccia di rete elastica utilizzata dalla funzione.
  - Chiama la Lambda`GetFunction`API per trovare l'ID VPC della funzione.
  - Chiama il servizio API Gateway per ottenere l'IP della funzione.
  - Utilizzando l'IP e l'ID VPC, trova l'interfaccia di rete associata ed estrai la zona di disponibilità.

```
VPC_ID=$(aws lambda get-function --function-name $ AWS_LAMBDA_FUNCTION_NAME --
region $AWS_REGION --output json --query ‘Configuration.VpcConfig.VpcId’)

MY_IP=$(curl http://whats-my-private-ip.internal)

AZ=$(aws ec2 describe-network-interfaces --filters Name=private-ip-address,Values=
$MY_IP Name=vpc-id,Values=$VPC_ID --region $AWS_REGION --output json –query
 ‘NetworkInterfaces[0].AvailabilityZone’)
```

Appendice B — Esempio di calcolo al quadrato

Di seguito è riportato un esempio di raccolta di metriche di errore ed esecuzione di un test chi-squared sui dati. Il codice non è pronto per la produzione e non esegue la necessaria gestione degli errori, ma fornisce una dimostrazione del funzionamento della logica. È necessario aggiornare questo esempio per adattarlo alle proprie esigenze.

Innanzitutto, una funzione Lambda viene richiamata ogni minuto da AmazonEventBridgeevento programmato. Il contenuto dell'evento è configurato con i seguenti dati:

```
{
  "timestamp": "2023-03-15T15:26:37.527Z",
  "namespace": "multi-az/frontend",
  "metricName": "5xx",
  "dimensions": [
    { "Name": "Region", "Value": "us-east-1" },
    { "Name": "Controller", "Value": "Home" },
    { "Name": "Action", "Value": "Index" }
  ],
  "period": 60,
  "stat": "Sum",
  "unit": "Count",
  "chiSquareMetricName": "multi-az/chi-squared",
  "azs": [ "use1-az2", "use1-az4", "use1-az6" ]
}
```

I dati vengono utilizzati per specificare i dati comuni necessari per recuperare i dati appropriatiCloudWatchmetriche (come namespace, nome della metrica e dimensioni) e quindi pubblica i risultati al quadrato per ciascuna zona di disponibilità. Il codice nella funzione Lambda è simile al seguente utilizzando Python 3.9. Ad un livello elevato, raccoglie quanto specificatoCloudWatchmetriche per il minuto precedente, esegue il test chi-squared su quei dati e quindi pubblicaCloudWatchmetriche sul risultato del test per ciascuna zona di disponibilità specificata.

```
import os
import boto3
import datetime
import copy
import json
from datetime import timedelta
```

```
from scipy.stats import chisquare
from aws_embedded_metrics import metric_scope

cw_client = boto3.client("cloudwatch", os.environ.get("AWS_REGION", "us-east-1"))

@metric_scope
def handler(event, context, metrics):
    metrics.set_property("Event", json.loads(json.dumps(event, default = str)))
    time = datetime.datetime.strptime(event["timestamp"], "%Y-%m-%dT%H:%M:%S.%fZ")

    # Round down to the previous minute
    end: datetime = roundTime(time)

    # Subtract a minute for the start
    start: datetime = end - timedelta(minutes = 1)

    # Get all the metrics that match the query
    results = get_all_metrics(event, start, end, metrics)
    metrics.set_property("MetricCounts", results)

    # Calculate the chi squared result
    chi_sq_result = chisquare(list(results.values()))
    expected = sum(list(results.values())) / len(results.values())
    metrics.set_property("ChiSquaredResult", chi_sq_result)

    # Put the chi square metrics into CloudWatch
    put_all_metrics(event, results, chi_sq_result[1], expected, start, metrics)

def get_all_metrics(detail: dict, start: datetime, end: datetime, metrics):
    """
    Gets all of the error metrics for each AZ specified
    """
    metric_query = {
        "MetricDataQueries": [
        ],
        "StartTime": start,
        "EndTime": end
    }

    for az in detail["azs"]:

        dim = copy.deepcopy(detail["dimensions"])
        dim.append({"Name": "AZ-ID", "Value": az})
```

```
        query = {
            "Id": az.replace("-", "_"),
            "MetricStat": {
                "Metric": {
                    "Namespace": detail["namespace"],
                    "MetricName": detail["metricName"],
                    "Dimensions": dim
                },
                "Period": int(detail["period"]),
                "Stat": detail["stat"],
                "Unit": detail["unit"]
            },
            "Label": az,
            "ReturnData": True
        }

        metric_query["MetricDataQueries"].append(query)

    metrics.set_property("GetMetricRequest", json.loads(json.dumps(metric_query,
default=str)))
    next_token: str = None
    results = {}

    while True:
        if next_token is not None:
            metric_query["NextToken"] = next_token

        data = cw_client.get_metric_data(**metric_query)

        if next_token is not None:
            metrics.set_property("GetMetricResult::" + next_token,
json.loads(json.dumps(data, default = str)))
        else:
            metrics.set_property("GetMetricResult", json.loads(json.dumps(data, default
= str)))

        for item in data["MetricDataResults"]:
            key = item["Id"].replace("_", "-")
            if key not in results:
              results[key] = 0

            results[key] += sum(item["Values"])

        if "NextToken" in data:
```

```
            next_token = data["NextToken"]

        if next_token is None:
            break

    return results

def put_all_metrics(detail: dict, results: dict, chi_sq_value: float, expected: float,
 timestamp: datetime, metrics):
    """
    Adds the chi squared metric for all AZs to CloudWatch
    """
    farthest_from_expected = None
    if len(results) > 0:
        keys = list(results.keys())
        farthest_from_expected = keys[0]

        for key in keys:
            if abs(results[key] - expected) > abs(results[farthest_from_expected] -
 expected):
               farthest_from_expected = key

    metric_query = {
        "Namespace": detail["namespace"],
        "MetricData": []
    }

    for az in detail["azs"]:
        dim = copy.deepcopy(detail["dimensions"])
        dim.append({"Name": "AZ-ID", "Value": az})

        query = {
            "MetricName": detail["chiSquareMetricName"],
            "Dimensions": dim,
            "Timestamp": timestamp,
        }

        if chi_sq_value <= 0.05 and az == farthest_from_expected:
            query["Value"] = 1
        else:
            query["Value"] = 0

        metric_query["MetricData"].append(query)
```

```
    metrics.set_property("PutMetricRequest", json.loads(json.dumps(metric_query,
 default = str)))

    cw_client.put_metric_data(**metric_query)

def roundTime(dt=None, roundTo=60):
   """Round a datetime object to any time lapse in seconds
   dt : datetime.datetime object, default now.
   roundTo : Closest number of seconds to round to, default 1 minute.
   """
   if dt == None : dt = datetime.datetime.now()
   seconds = (dt.replace(tzinfo=None) - dt.min).seconds
   rounding = (seconds+roundTo/2) // roundTo * roundTo
   return dt + datetime.timedelta(0,rounding-seconds,-dt.microsecond)
```

È quindi possibile creare un allarme per AZ. L'esempio seguente è per`use1-az2`e allarmi per tre punti dati consecutivi della durata di un minuto con un valore massimo pari a 1 (1 è la metrica pubblicata quando il test chi-squared determina una distorsione statisticamente significativa del tasso di errore).

```
{
    "Type": "AWS::CloudWatch::Alarm",
    "Properties": {
        "AlarmName": "use1-az2-chi-squared",
        "ActionsEnabled": true,
        "OKActions": [],
        "AlarmActions": [],
        "InsufficientDataActions": [],
        "MetricName": "multi-az/chi-squared",
        "Namespace": "multi-az/frontend",
        "Statistic": "Maximum",
        "Dimensions": [
            {
                "Name": "AZ-ID",
                "Value": "use1-az2"
            },
            {
                "Name": "Action",
                "Value": "Index"
            },
            {
```

```
                "Name": "Region",
                "Value": "us-east-1"
            },
            {
                "Name": "Controller",
                "Value": "Home"
            }
        ],
        "Period": 60,
        "EvaluationPeriods": 3,
        "DatapointsToAlarm": 3,
        "Threshold": 1,
        "ComparisonOperator": "GreaterThanOrEqualToThreshold",
        "TreatMissingData": "missing"
    }
}
```

Puoi anche creare unm-of-nallarma e combina questi due allarmi con un allarme composito. Dovrai anche creare gli stessi allarmi per ogni combinazione Controller/Azione o microservizio disponibile in ogni zona di disponibilità. Infine, puoi aggiungere l'allarme composito chi-squared all'allarme specifico della zona di disponibilità per ciascuna combinazione controller/azione, come mostrato in[Rilevamento dei guasti mediante rilevamento dei valori anomali](#).

Fattori determinanti

I contributori a questo documento includono:

- Michael Haken, Principal Solutions Architect, Amazon Web Services

Revisioni del documento

Per ricevere notifiche sugli aggiornamenti di questo white paper, iscriviti al feed RSS.

Modifica	Descrizione	Data
Whitepaper aggiornato	Aggiornato con ulteriori indicazioni sull'osservabilità e per utilizzare la nuova funzione di spostamento zonale.	11 luglio 2023
Pubblicazione iniziale	Whitepaper pubblicato per la prima volta.	2 marzo 2022

> **Note**
>
> Per sottoscrivere gli aggiornamenti RSS, devi avere un plug-in RSS abilitato per il browser che stai utilizzando.

Note

I clienti sono responsabili della propria valutazione indipendente delle informazioni contenute in questo documento. Questo documento: (a) è solo a scopo informativo, (b) rappresenta l'attualitàAWSofferte e pratiche di prodotti, che sono soggette a modifiche senza preavviso e (c) non creano alcun impegno o garanzia da parte diAWSe i suoi affiliati, fornitori o licenzianti.AWSi prodotti o i servizi sono forniti «così come sono» senza garanzie, dichiarazioni o condizioni di alcun tipo, espresse o implicite. Le responsabilità e le responsabilità diAWSai propri clienti sono controllati daAWSaccordi e il presente documento non fa parte, né modifica, alcun accordo traAWSe i suoi clienti.

© 2023 Amazon Web Services, Inc. o società affiliate. Tutti i diritti riservati.

Glossario AWS

Per la terminologia AWS più recente, consultare il glossario AWS nella documentazione di riferimento per Glossario AWS.

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.