Incident management in Cloud: strumenti e best practice per identificare, analizzare e risolvere problemi tempestivamente

8 Dicembre 2023 - 7 min. read

AWS Incident Management | Business Continuity

"Io non complico le cose. Sono così, di per sé" - Martin Riggs, Arma Letale

Progettiamo architetture tenendo conto di tutte le best practice, formiamo i nostri collaboratori per far acquisire loro le migliori competenze e documentiamo tutte le nostre infrastrutture. MA... A volte qualcosa va ancora storto, e la parola “incidente” risuona nelle nostre orecchie.

Ma cos’è un incidente?

Un incidente è un evento che riduce la qualità o le prestazioni di un servizio IT, ne causa l'interruzione o pone un rischio per la sicurezza. Può essere il malfunzionamento di un server, un guasto di rete, un’infezione da malware o una violazione dei dati (data breach).

Anche se facciamo tutto il possibile per fare in modo che non si verifichino e per essere sicuri di esporci al minimo rischio, gli incidenti IT sono inevitabili e possono avere un impatto significativo sulle performance, sulla reputazione e sull’esperienza degli utenti. Gestire gli incidenti non è una competenza facile da acquisire, ma è necessario acquisirla nel corso delle nelle nostre carriere lavorative.

In questo articolo vedremo gli elementi chiave e le migliori pratiche per la gestione degli incidenti. Vi presenteremo anche AWS Systems Manager Incident Manager, una funzionalità di AWS Systems Manager che aiuta a prepararsi ed a rispondere agli incidenti applicativi e infrastrutturali.

Incident Management

Abbiamo detto che la gestione degli incidenti è fondamentale per offrire un servizio affidabile e sicuro. Si tratta di un processo che aiuta a identificare, analizzare e risolvere qualsiasi evento o problema imprevisto che influisce sulla qualità, sulla disponibilità o sulle prestazioni percepite dagli utenti.

Gli incident possono avere diversi livelli di gravità e impatto a seconda del tipo di servizio, del numero di utenti coinvolti, della durata dell'interruzione e delle possibili conseguenze.

Un processo di gestione degli incidenti mira a ripristinare la normale operatività del servizio il più rapidamente possibile. Implementando un processo efficace, possiamo migliorare l'esperienza utente, prevenire gli incidenti e ridurre (o eliminare) il downtime, e il tempo di risoluzione.

Un processo di incident management tipicamente contiene queste fasi:

- **Rilevamento**: quando un incidente viene scoperto o segnalato.
- **Classificazione**: viene assegnata una priorità in base alla sua gravità e all'impatto.
- **Escalation**: vengono coinvolte le persone e i team appropriati per gestire l'incidente in base alla sua priorità e categoria.
- **Diagnosi**: analisi della causa principale (root cause) e delle possibili soluzioni.
- **Risoluzione**: viene implementata la migliore soluzione per risolvere l'incidente e verificare che il servizio sia ripristinato alla normalità.
- **Chiusura**: la fase di documentazione e comunicazione dei dettagli, dei risultati, delle lezioni apprese e della chiusura della registrazione dell'incidente.

Un altro punto critico consiste nell'identificare i ruoli responsabili della gestione di ogni fase. Questo dipende dalla dimensione e dalla struttura dell'organizzazione e dalla complessità e natura dell'incidente. Tuttavia, ci sono alcuni ruoli e responsabilità comuni che si trovano nella maggior parte dei team, ad esempio:

- **Incident Manager**: supervisiona e coordina l'intero processo di gestione degli incidenti, dal rilevamento alla chiusura.
- **Incident Owner**: ha l'autorità di prendere decisioni e approvare le azioni relative all'incidente.
- **Incident Team**: il gruppo di persone con le competenze e l'esperienza necessarie coinvolte nella diagnosi, gestione e nella risoluzione dell'incidente

- **Incident Reporter**: rileva o segnala l'incidente e fornisce le informazioni e i dettagli iniziali.

A questo punto dobbiamo trovare una soluzione per semplificarci la vita: Incident Manager, con la sua integrazione con gli altri servizi AWS, può aiutarci nelle varie fasi.

Vediamo prima come può mappare le fasi del processo.

Possiamo definire **contatti** e **contact channels** da utilizzare negli **engagement plans** per coinvolgere le persone giuste mantenendo una comunicazione comune chiara e sincronizzata: è sempre la migliore strategia quando si affrontano i problemi.

Con le **on-call schedules** e gli **escalation plans** si possono ovviamente definire gli orari di reperibilità e le dovute escalation delle notifiche, quando necessario.

Possiamo anche definire degli **Automation Runbooks** che sfruttano AWS Systems Manager Automation per automatizzare i task operativi comuni ed evitare operazioni manuali (che possono essere soggette a errori). Sono utili per automatizzare le operazioni di risposta agli incidenti e fornire informazioni dettagliate agli operatori.

Un **response plan** collega tutti gli elementi di cui abbiamo parlato, definendo cosa deve essere fatto quando si verifica un incidente, come ad esempio chi è tenuto a rispondere, il canale di comunicazione da utilizzare e le azioni automatiche da intraprendere.

Gli **incidents** possono essere creati in modo automatico, ad esempio sfruttando le regole di EventBridge, i risultati di SecurityHub e gli allarmi di CloudWatch.

Quando si verifica un incidente, AWS Incident Mkanager raccoglie automaticamente i dati sulle risorse AWS interessate, mostrandoli nella scheda Elementi correlati. È possibile anche usare un runbook nel response plan per risolvere il problema.

I dati raccolti sulle risorse AWS interessate possono essere passati al runbook, che potrà usarli per cercare di risolvere automaticamente il problema.

Vediamo un semplice caso d'uso, che può essere poi adattato a diverse esigenze.

Monitoreremo una VPN site-to-site per fare in modo che quando un tunnel non risulta operativo vengano eseguite queste azioni:

- creazione di un incidente
- invio di una notifica su un canale Slack

- esecuzione del documento SSM di troubleshooting gestito da AWS che esegue l'analisi iniziale su CloudWatch Logs Insight per fornire informazioni al team.

Questo è un esempio di template; naturalmente **deve essere poi esteso ed adattato per la configurazione reale**:

```
AWSTemplateFormatVersion: 2010-09-09
Description: A template that creates a CloudWatch alarm for a site-to-sit
e VPN connection and an incident response plan for AWS Systems Manager In
cident Manager.

Parameters:
  VpnConnectionId:
    Type: String
    Description: The ID of the VPN connection to monitor.
    AllowedPattern: ^vpn-[0-9a-f]{8,17}$
    ConstraintDescription: Must be a valid VPN connection ID.

Resources:
  VpnAlarm:
    Type: AWS::CloudWatch::Alarm
    Properties:
      AlarmName: !Sub "VPN connection ${VpnConnectionId} status alarm"
      AlarmDescription: An alarm that triggers when the VPN connection st
atus is DOWN.
      Namespace: AWS/VPN
      MetricName: TunnelState
      Dimensions:
        - Name: VpnId
          Value: !Ref VpnConnectionId
      Statistic: Minimum
      Period: 60
      EvaluationPeriods: 1
      Threshold: 1
      ComparisonOperator: LessThanThreshold
      AlarmActions:
        - !Ref IncidentResponsePlan

  IncidentResponsePlan:
    Type: AWS::SSMIncidents::ResponsePlan
```

```
    Properties:
      Name: !Sub "VPN connection ${VpnConnectionId} incident response pla
n"
      DisplayName: !Sub "VPN connection ${VpnConnectionId} incident respo
nse plan"
      ChatChannel:
        ChatbotSns: !Ref NotificationTopic
      IncidentTemplate:
        Title: !Sub "VPN connection ${VpnConnectionId} is DOWN"
        Impact: 3
        Summary: "The VPN connection to the remote site is not working."
        DedupeString: !Sub "VPN connection ${VpnConnectionId} is DOWN"
      Actions:
        - SsmAutomation:
            RoleArn: !GetAtt AutomationRole.Arn
            DocumentName: AWSSupport-TroubleshootVPN
            DocumentVersion: "1"
            Parameters:
              VpnConnectionId: !Ref VpnConnectionId

  NotificationTopic:
    Type: AWS::SNS::Topic
    Properties:
      DisplayName: "VPN connection status notification"
      TopicName: "vpn-connection-status-notification"

  SlackChannelConfiguration:
    Type: AWS::Chatbot::SlackChannelConfiguration
    Properties:
      ConfigurationName: "vpn-connection-status-slack"
      IamRoleArn: !GetAtt ChatbotRole.Arn
      SlackChannelId: "YourChannelID"
      SlackWorkspaceId: "SlackWorkspace"
      SnsTopicArns:
        - !Ref NotificationTopic

  AutomationRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
```

```
      Version: "2012-10-17"
      Statement:
        - Effect: Allow
          Principal:
            Service: ssm.amazonaws.com
          Action: sts:AssumeRole
    Path: "/"
    Policies:
      - PolicyName: "vpn-troubleshoot-policy"
        PolicyDocument:
          Version: "2012-10-17"
          Statement:
            - Effect: Allow
              Action:
                - ec2:DescribeVpnConnections
                - ec2:DescribeVpnGateways
                - ec2:DescribeCustomerGateways
                - ec2:ResetVpnConnection
              Resource: "*"

ChatbotRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: "2012-10-17"
      Statement:
        - Effect: Allow
          Principal:
            Service: chatbot.amazonaws.com
          Action: sts:AssumeRole
    Path: "/"
    Policies:
      - PolicyName: "chatbot-policy"
        PolicyDocument:
          Version: "2012-10-17"
          Statement:
            - Effect: Allow
              Action:
                - cloudwatch:DescribeAlarms
                - cloudwatch:ListMetrics
```

```
            - cloudwatch:GetMetricData
            - cloudwatch:GetMetricStatistics
            - cloudwatch:PutMetricData
            - ec2:DescribeInstances
            - ec2:DescribeRegions
            - ec2:DescribeVpnConnections
            - ec2:DescribeVpnGateways
            - ec2:DescribeCustomerGateways
            - ec2:ResetVpnConnection
            - sns:ListTopics
            - sns:ListSubscriptionsByTopic
            - sns:Publish
          Resource: "*"
```

La documentazione AWS fornisce un altro esempio interessante per monitorare e avvisare in caso vengano svolte attività usando il root account.

Gestire gli incidenti è una sfida, e over-ingegnerizzare il processo può rallentare molto. Il nostro consiglio è di iniziare con una mentalità agile, implementando una soluzione semplice ma che soddisfi le esigenze di base: iniziare ad usare Systems Manager Incident Manager ed adattarlo alle proprie esigenze è molto semplice perché si integra rapidamente con tutti i servizi AWS. Una volta messe le basi ed ottenuti i primi feedback è possibile migliorare il processo e l'implementazione con poco sforzo.

Quando parliamo di processi e metodi, non esiste una "soluzione unica per tutti": ad esempio, abbiamo scoperto che nel nostro caso è utilisssimo organizzare una retrospettiva coinvolgendo i membri del team e tutti gli stakeholder necessari, anche se aggiunge un passaggio aggiuntivo alla fase post-incidente.

Quale pratica trovate efficacie per la preparazione e la gestione degli incidenti sul Cloud?

Fatecelo sapere nei commenti!

About Proud2beCloud

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

regolarmente i nostri migliori spunti con chi come noi, per lavoro o per passione, lavora con il Cloud di AWS. Partecipa alla discussione!

Damiano Giorgi

Ex sistemista on-prem, pigro e incline all'automazione di task noiosi. Alla ricerca costante di novità tecnologiche e quindi passato al cloud per trovare nuovi stimoli.L'unico hardware a cui mi dedico ora è quello del mio basso; se non mi trovate in ufficio o in sala prove provate al pub o in qualche aeroporto!
