Amazon CloudWatch Logs

Benutzerhandbuch

Amazon CloudWatch Logs: Benutzerhandbuch

Table of Contents

Was ist Amazon CloudWatch Logs? ... 1
- Funktionen ... 1
- Zugehörige AWS-Sevices ... 2
- Preise ... 2
- Konzepte ... 2

Einrichten ... 4
- Registrierung bei Amazon Web Services (AWS) ... 4
- Melden Sie sich bei der Amazon CloudWatch-Konsole an. ... 4
- Einrichten der Befehlszeilenschnittstelle ... 4

Erste Schritte ... 5
- Verwenden des vereinheitlichten CloudWatch-Agenten für die ersten Schritte mit CloudWatch Logs ... 5
- Verwenden des vorherigen CloudWatch Logs-Agenten für die ersten Schritte mit CloudWatch Logs ... 6
  - Voraussetzungen für den CloudWatch Logs-Agenten ... 6
  - Schnellstart Installieren des Agenten bei einer EC2-Linux-Instanz ... 7
  - Schnellstart Installieren des Agenten bei einer EC2 Linux Instanz bei Start ... 12
  - Schnellstart Verwenden CloudWatch Logs mit Windows Server 2016 Instanzen ... 14
  - Schnellstart Verwenden von CloudWatch Logs mit Windows Server 2012- und Windows Server 2008-Instances ... 22
  - Schnellstart Installieren des Agenten mit AWS OpsWorks ... 29
  - Den Status des CloudWatch Logs-Agenten melden ... 33
  - Den CloudWatch Logs-Agenten starten ... 34
  - Beenden des CloudWatch Logs-Agenten ... 34
- Schnellstart Verwenden AWS CloudFormation damit beginnen mit CloudWatch Logs ... 34

Analysieren von Protokolldaten mit CloudWatch Logs Insights ... 36
- Unterstützte Protokolle und erkannte Felder ... 36
  - Felder in JSON-Protokollen ... 38
- Praktische Anleitung Ausführen und Ändern einer Beispielabfrage ... 39
  - Ausführen einer Beispielabfrage ... 39
  - Ändern der Beispielabfrage ... 39
  - Hinzufügen eines Filterbefehls zur Beispielabfrage ... 40
- Praktische Anleitung Ausführen einer Abfrage mit einer Aggregationsfunktion ... 41
- Praktische Anleitung Ausführen einer Abfrage, die eine Visualisierung erzeugt, die nach Protokollfeldern gruppiert ist ... 41
- Praktische Anleitung Ausführen einer Abfrage, die eine Zeitreihenvisualisierung erzeugt ... 42
- Abfragesyntax ... 42
  - Unterstützte Abfragebefehle ... 43
  - Übereinstimmungen und reguläre Ausdrücke im Filterbefehl ... 47
  - Verwenden von Aliasen in Abfragen ... 48
  - Verwenden von Kommentaren in Abfragen ... 48
  - Unterstützte Operationen und Funktionen ... 48
- Visualisieren von Protokolldaten in Diagrammen ... 53
  - Visualisierung von Zeitreihendaten ... 54
  - Visualisieren von nach Feldern gruppierten Protokolldaten ... 54
- Speichern und erneutes Ausführen von Abfragen ... 55
- Beispielabfragen ... 56
- Abfrage zum Dashboard hinzufügen oder Abfrageergebnisse exportieren ... 59
- Anzeigen von laufenden Abfragen oder Abfrageverlauf ... 60

Arbeiten mit Log-Gruppen und Log-Streams ... 61
- Eine Protokollgruppe erstellen ... 61
- Protokolle an eine Protokollgruppe senden ... 61
- Protokolldaten anzeigen lassen ... 61
- Suchen von Protokolldaten mithilfe von Filtermustern ... 62
  - Suchen von Protokolleinträge mithilfe der Konsole ... 62
  - Suchen von Protokolleinträgen mithilfe der AWS CLI ... 63

Wechseln von Metriken zu Protokollen 63
Troubleshooting 64
Aufbewahrung von Protokolldaten ändern 64
Protokollgruppen kennzeichnen 64
Grundlagen zu Tags 65
Kosten mithilfe von Tags verfolgen 65
Einschränkungen für Tags 65
Protokollgruppen mithilfe der AWS CLI mit Tags kennzeichnen 66
Protokollgruppen mithilfe der CloudWatch Logs-API mit Tags kennzeichnen 66
Verschlüsseln von Protokolldaten mit AWS KMS 67
Limits 67
Schritt 1 Erstellen AWS KMS CMK 68
Schritt 2 Berechtigungen auf dem CMK einstellen 68
Schritt 3 Eine Protokollgruppe mit einem CMK verknüpfen 70
Schritt 4. Eine Protokollgruppe von einem CMK trennen 70
KMS-Schlüssel und Verschlüsselungskontext 70
Aktivieren der Protokollierung von bestimmten AWS-Services aus 73
Erstellen von Metriken aus Protokollereignissen mithilfe von Filtern 74
Concepts 74
Filter- und Mustersyntax 75
Auffinden von Begriffen in Protokollereignissen 75
Einrichten, wie sich der Metrikwert bei gefundenen Übereinstimmungen ändert 82
Veröffentlichen der in Protokolleinträgen gefundenen numerischen Werte 82
Erstellen von Metrikfiltern 83
Beispiel. Zählprotokoll-Ereignisse 83
Beispiel. Anzahl der Wiederholungen einer Laufzeit 84
Beispiel. HTTP 404 Codes zählen 85
Beispiel. HTTP 4xx Codes zählen 87
Beispiel. Felder aus einem Apache-Protokoll extrahieren 88
Auflisten von Metrikfiltern 89
Löschen eines Metrikfilters 89
Protokolldaten-Verarbeitung in Echtzeit mit Abonnements 91
Concepts 91
Verwenden von -Abonnementfiltern 92
Beispiel 1 Abonnementfilter mit Kinesis 92
Beispiel 2 Abonnementfilter mit AWS Lambda 96
Beispiel 3 Abonnementfilter mit Amazon Kinesis Data Firehose 98
Freigabe von Protokolldaten mit Abonnements für mehrere Konten 103
Erstellen eines Ziels 104
Erstellen eines Abonnementfilters 107
Validierung des Flusses von Protokollereignissen 107
Ändern der Mitgliedschaft im Ziel zur Laufzeit 109
Senden von Protokollen direkt an Amazon S3 oder Kinesis Data Firehose 111
Exportieren von Protokolldaten in Amazon S3 112
Concepts 112
Exportieren von Protokolldaten in Amazon S3 mit der Konsole 113
Schritt 1 Erstellen eines Amazon S3-Buckets 113
Schritt 2 Erstellen eines IAM Benutzer mit Vollzugriff auf Amazon S3 und CloudWatch Logs 113
Schritt 3 Festlegen von Berechtigungen für eine Amazon S3 Behälter 114
Schritt 4. Erstellen einer Exportaufgabe 115
Exportieren von Protokolldaten mit der AWS CLI in Amazon S3 116
Schritt 1 Erstellen eines Amazon S3-Buckets 116
Schritt 2 Erstellen eines IAM Benutzer mit Vollzugriff auf Amazon S3 und CloudWatch Logs 116
Schritt 3 Festlegen von Berechtigungen für eine Amazon S3 Behälter 117
Schritt 4. Erstellen einer Exportaufgabe 119
Schritt 5. Exportaufgaben beschreiben 119
Schritt 6. Eine Exportaufgabe abbrechen 120

Daten in Amazon ES streamen 122
Voraussetzungen 122
Abonnieren einer Protokollgruppe für Amazon ES 122
AWS-Services, die Protokolle veröffentlichen 124
Sicherheit 126
Datenschutz 126
Verschlüsselung im Ruhezustand 127
Verschlüsselung während der Übertragung 127
Identitäts- und Zugriffsverwaltung 127
Authentication 127
Zugriffskontrolle 129
Übersicht über die Verwaltung des Zugriffs 129
Verwenden identitätsbasierter Richtlinien (IAM-Richtlinien) 133
Referenz für CloudWatch Logs-Berechtigungen 138
Verwenden von serviceverknüpften Rollen 142
Compliance-Validierung 144
Ausfallsicherheit 144
Sicherheit der Infrastruktur 145
Schnittstellen-VPC-Endpunkte 145
Verfügbarkeit 145
Erstellen eines VPC-Endpunkts für CloudWatch Logs 146
Testen der Verbindung zwischen Ihrer VPC und CloudWatch Logs 146
Steuern des Zugriffs auf Ihren CloudWatch Logs-VPC-Endpunkt 147
Support für VPC-Kontextschlüssel 148
Protokollieren von API-Aufrufen 149
CloudWatch Logs-Informationen in CloudTrail 149
Grundlagen zu Protokolldateieinträgen 150
Referenztabelle für den Agenten 152
Agent-Konfigurationsdatei 152
Verwenden des CloudWatch Logs-Agent mit HTTP-Proxys 156
Aufgliedern der Konfigurationsdateien für den CloudWatch Logs-Agent 157
CloudWatch Logs-Agent – Häufig gestellte Fragen 157
Überwachung der Nutzung mit CloudWatch-Metriken 161
CloudWatch Logs-Metriken 161
Dimensionen für CloudWatch Logs-Metriken 162
Servicekontingente 163
Dokumentverlauf 165
AWS-Glossar 167
..... clxviii

Was ist Amazon CloudWatch Logs?

Sie können Amazon CloudWatch Logs verwenden, um Ihre Protokolldateien aus Amazon Elastic Compute Cloud (Amazon EC2)-Instances, AWS CloudTrail, Route 53 und aus anderen Quellen zu überwachen, zu speichern und darauf zuzugreifen.

CloudWatch Logs ermöglicht es Ihnen, die Protokolle von allen Ihren Systemen, Anwendungen und AWS-Services, die Sie verwenden, in einem einzigen, hochgradig skalierbaren Service zu zentralisieren. Sie können sie dann einfach anzeigen, sie auf bestimmte Fehlercodes oder Muster durchsuchen, sie basierend auf bestimmten Feldern filtern oder sie für die spätere Analyse sicher archivieren. CloudWatch Logs ermöglicht Ihnen, alle Ihr Protokolle, unabhängig von ihrer Quelle, als einzelnen und konsistenten zeitlich angeordneten Fluss von Ereignissen anzuzeigen. Sie können sie abfragen und basierend auf anderen Dimensionen sortieren, sie nach bestimmten Feldern gruppieren, benutzerdefinierte Berechnungen mit einer leistungsstarken Abfragesprache erstellen und Protokolldaten in Dashboards visualisieren.

Funktionen

- Abfragen Ihrer Protokolldaten – Mit CloudWatch Logs Insights können Sie interaktiv Ihre Protokolldaten durchsuchen und analysieren. Sie können Abfragen durchführen, die Ihnen helfen, schnell und effektiv auf betriebliche Probleme zu reagieren. CloudWatch Logs Insights enthält eine speziell entwickelte Abfragesprache mit einigen einfachen, aber leistungsstarken Befehlen. Wir stellen Beispielabfragen, Befehlsbeschreibungen, automatische Abfragevervollständigung und Protokollfelderkennung zur Verfügung, um Ihnen den Einstieg zu erleichtern. Beispielabfragen sind für verschiedene Arten von AWS-Serviceprotokollen enthalten. Lesen Sie zum Einstieg Analysieren von Protokolldaten mit CloudWatch Logs Insights (p. 36).
- Protokolle aus Amazon EC2-Instances überwachen – Sie können CloudWatch Logs verwenden, um Anwendungen und Systeme, die Protokolldaten verwenden, zu überwachen. Beispielsweise kann CloudWatch Logs die Fehler zählen, die in Ihren Anwendungsprotokollen aufgeführt werden, und Ihnen eine Benachrichtigung senden, wenn die Fehlerrate einen von Ihnen vorgegebenen Schwellenwert überschreitet. CloudWatch Logs verwendet Ihre Protokolldaten für die Überwachung, es sind also keine Codeänderungen erforderlich. Sie können beispielsweise Anwendungsprotokolle auf bestimmte Begriffe (wie z. B. „NullReferenceException“) hin überwachen oder zählen, wie oft ein Begriff in den Protokolldaten an einer bestimmten Position auftritt (z. B. „404“-Statuscodes in einem Apache-Zugriffsprotokoll). Wird der von Ihnen gesuchte Begriff gefunden, meldet CloudWatch Logs die Daten an eine von Ihnen angegebene CloudWatch-Metrik. Die Protokolldaten werden während der Übermittlung und Speicherung verschlüsselt. Lesen Sie zum Einstieg Erste Schritte mit CloudWatch Logs (p. 5).
- Protokollierte Ereignisse für AWS CloudTrail überwachen – Sie können Alarme in CloudWatch erstellen und Benachrichtigungen von bestimmten von CloudTrail erfassten API-Aktivitäten erhalten und anhand der Benachrichtigungen eine Fehlerbehebung durchführen. Informationen zu den ersten Schritten finden Sie unter Senden von Ereignissen an CloudTrailCloudWatch Logs im AWS CloudTrail User Guide.
- Aufbewahrung von Protokollen – Standardmäßig werden Protokolle unbegrenzt aufbewahrt und laufen nicht ab. Sie können die Aufbewahrungsrichtlinie für jede Protokollgruppe anpassen und Protokolle entweder unbegrenzt speichern oder einen Aufbewahrungszeitraum zwischen 10 Jahren und einem Tag auswählen.
- Protokolldaten archivieren – Sie können CloudWatch Logs verwenden, um Ihre Protokolldaten in einem Speicher von hoher Haltbarkeit abzulegen. Mit dem CloudWatch Logs-Agenten ist es ganz einfach, sowohl rotierte als auch nicht rotierte Protokolldaten von einem Host in den Protokoll-Service zu senden. Sie können dann bei Bedarf auf die unformatierten Protokolldaten zugreifen.
- Protokollierung von Route 53-DNS-Abfragen – Mit CloudWatch Logs können Sie Informationen über die DNS-Abfragen protokollieren, die Route 53 erhält. Weitere Informationen finden Sie unter Protokollierung von DNS-Abfragen im Entwicklerhandbuch für Amazon Route 53.

Zugehörige AWS-Sevices

Die folgende Services werden in Verbindung mit CloudWatch Logs verwendet:

- AWS CloudTrail ist ein Webservice, mit dem Sie die Aufrufe an die CloudWatch Logs API für Ihr Konto, einschließlich der Aufrufe, die von der AWS Management Console, der AWS Command Line Interface (AWS CLI) und sonstigen Services ausgingen, überwachen können. Wenn die CloudTrail-Protokollierung aktiviert ist, erfasst CloudTrail die API-Aufrufe in Ihrem Konto und stellt die Protokolldateien in dem von Ihnen angegebenen Amazon S3-Bucket bereit. Jede Protokolldatei kann eine oder mehrere Datensätze enthalten, je nachdem, wie viele Aktionen zur Erfüllung einer Anfrage durchgeführt werden müssen. Weitere Informationen über AWS CloudTrail finden Sie unter Was ist AWS CloudTrail? im AWS CloudTrail User Guide. Ein Beispiel für eine solche Art von Daten, die CloudWatch in CloudTrail-Protokolldateien schreibt, finden Sie unter Protokollieren von Amazon CloudWatch Logs-API-Aufrufen in AWS CloudTrail (p. 149).
- AWS Identity and Access Management (IAM) ist ein Webservice, mit dem Sie auf sichere Weise den Zugriff auf AWS-Ressourcen für Ihre Benutzer steuern können. Kontrollieren Sie mit IAM, wer Ihre AWS-Ressourcen verwenden kann (Authentifizierung) und welche Ressourcen auf welche Weise verwendet werden können (Autorisierung). Weitere Informationen finden Sie unter Was ist IAM? im IAM-Benutzerhandbuch.
- Amazon Kinesis Data Streams ist ein Web-Service, den Sie für die schnelle und kontinuierliche Aufnahme und Aggregation von Daten verwenden können. Der verwendete Datentyp umfasst Protokolldaten zur IT-Infrastruktur, Anwendungsprotokolle, Data-Feeds von sozialen Medien, Marktdaten-Feeds sowie Web-Clickstream-Daten. Da die Reaktionszeit für die Aufnahme und Verarbeitung der Daten in Echtzeit erfolgt, ist die Verarbeitung in der Regel ein leichtgewichtiger Prozess. Weitere Informationen finden Sie unter Was ist Amazon Kinesis Data Streams? im Amazon Kinesis Data Streams-Entwicklerhandbuch.
- AWS Lambda ist ein Web-Service, den Sie verwenden können, um Anwendungen zu erstellen, die schnell auf neue Informationen reagieren. Laden Sie Ihren Anwendungscode hoch, sobald Lambda funktioniert und Lambda führt Ihren Code auf einer hochverfügbaren Datenverarbeitungsinfrastruktur aus und erledigt die gesamte Administration der Datenverarbeitungsressourcen, einschließlich der Server- und Betriebssystemwartung, Kapazitätsbereitstellung, automatischen Skalierung, Bereitstellung des Code- und Sicherheitspatches sowie der Code-Überwachung und -Protokollierung. Sie müssen lediglich Ihren Code in einer der von Lambda unterstützten Sprachen angeben. Weitere Informationen finden Sie unter Was ist AWS Lambda? im AWS Lambda Developer Guide.

Preise

Wenn Sie sich bei AWS anmelden, können Sie kostenlos mit der Verwendung von CloudWatch Logs beginnen, indem Sie das AWS kostenlose Kontingent von AWS nutzen.

Die Standard-Preise gelten für Protokolle, die von anderen Services unter Verwendung von CloudWatch Logs gespeichert werden (z. B. Amazon VPC Flow- und Lambda-Protokolle).

Weitere Informationen dazu finden Sie unter Amazon CloudWatch – Preise.

Amazon CloudWatch Logs-Konzepte

Die Terminologie und Konzepte, die für Ihr Verständnis und den Umgang mit CloudWatch Logs von wesentlicher Bedeutung sind, werden nachstehend beschrieben.

Protokollereignisse

Ein Protokollereignis ist ein Datensatz von einigen Aktivitäten, der von der überwachten Anwendung oder Ressource aufgezeichnet wird. Der Protokollereignis-Datensatz, den CloudWatch Logs versteht,

enthält zwei Eigenschaften: den Zeitstempel mit dem Zeitpunkt, zu dem das Ereignis auftrat, und die unformatierte Ereignismeldung. Ereignismeldungen müssen UTF-8-kodiert sein.

Protokollstreams

Ein Protokollstream ist eine Abfolge von Protokollereignissen, die dieselbe Quelle nutzen. Genauer gesagt ist ein Protokollstream allgemein dafür gedacht, die Abfolge der aus der überwachten Anwendungs-Instance oder Ressource stammenden Ereignisse darzustellen. Ein Protokollstream kann beispielsweise mit einem Apache-Zugriffsprotokoll auf einem bestimmten Host verknüpft sein. Wenn Sie einen Protokollstream nicht mehr benötigen, können Sie ihn mithilfe des Befehls aws logs delete-log-stream löschen. Darüber hinaus kann AWS leere Protokollstreams löschen, die älter als 2 Monate sind.

Protokollgruppen

Protokollgruppen definieren Gruppen von Protokollstreams, die dieselben Einstellungen für die Aufbewahrung, Überwachung und Zugriffskontrolle besitzen. Jeder Protokollstream muss zu einer Protokollgruppe gehören. Wenn Sie beispielsweise über einen separaten Protokoll-Stream für die Apache-Zugriffsprotokolle von jedem Host verfügen, können Sie diese in einer einzelnen Protokollgruppe mit dem Namen `MyWebsite.com/Apache/access_log` gruppieren.

Es gibt keine Begrenzung dazu, wie viele Protokoll-Streams zu einer Protokollgruppe gehören können.

Metrikfilter

Sie können Metrikfilter verwenden, um Metrik-Beobachtungen von übernommenen Ereignissen zu extrahieren und in Datenpunkte in einer CloudWatch-Metrik umzuwandeln. Metrikfilter sind Protokollgruppen zugewiesen, und alle einer Protokollgruppe zugewiesenen Filter werden auf ihre Protokollstreams angewendet.

Einstellungen für die Aufbewahrung

Die Einstellungen für die Aufbewahrung können verwendet werden, um festzulegen, wie lange Protokollereignisse in CloudWatch Logs gespeichert werden sollen. Abgelaufene Protokollereignisse werden automatisch gelöscht. Wie Metrikfilter werden auch die Einstellungen für die Aufbewahrung den Protokollgruppen zugewiesen, und die einer Protokollgruppe zugewiesene Aufbewahrung wird auf ihre Protokollstreams angewendet.

Einrichten

Um Amazon CloudWatch Logs verwenden zu können, benötigen Sie ein AWS-Konto. Ihr AWS-Konto gibt Ihnen die Möglichkeit, Services (z. B. Amazon EC2) zum Generieren von Protokollen zu verwenden, die Sie in der CloudWatch-Konsole, eine webbasierten, grafischen Benutzeroberfläche, anzeigen können. Darüber hinaus können Sie die AWS Command Line Interface (AWS CLI) installieren und konfigurieren.

Registrierung bei Amazon Web Services (AWS)

Wenn Sie ein AWS-Konto erstellen, wird Ihr Konto automatisch für alle AWS-Services registriert. Berechnet werden Ihnen nur die Services, die Sie nutzen.

Wenn Sie bereits ein AWS-Konto haben, wechseln Sie zum nächsten Schritt. Wenn Sie kein AWS-Konto haben, befolgen Sie diese Schritte zum Erstellen eines Kontos.

Registrieren Sie sich für ein AWS-Konto wie folgt:

1. Öffnen Sie https://portal.aws.amazon.com/billing/signup.
2. Folgen Sie den Onlineanweisungen.

   Der Anmeldeprozess beinhaltet auch einen Telefonanruf und die Eingabe eines Verifizierungscodes über die Telefontastatur.

Melden Sie sich bei der Amazon CloudWatch-Konsole an.

Anmelden bei der Amazon CloudWatch-Konsole

1. Melden Sie sich bei der AWS Management Console an. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Ändern Sie, falls erforderlich, die Region. Wählen Sie auf der Navigationsleiste die Region, in der sich Ihre AWS-Ressourcen befinden.
3. Wählen Sie im Navigationsbereich Logs aus.

Einrichten der Befehlszeilenschnittstelle

Für die Durchführung der AWS CLI-Operationen können Sie die CloudWatch Logs verwenden.

Informationen zur Installation und Konfiguration der AWS CLI finden Sie unter Einrichtung der AWS-Befehlszeilenschnittstelle im Benutzerhandbuch für AWS Command Line Interface.

Erste Schritte mit CloudWatch Logs

In AWS haben Sie zwei Möglichkeiten zum Erfassen von Protokollen für Ihre Amazon EC2-Instances und Ihre lokalen Server in CloudWatch Logs:

- Empfohlen: Vereinheitlichter CloudWatch-Agent. Dieser Agent ermöglicht das Erfassen von Protokollen und erweiterte Metriken mit nur einem Agenten. Er unterstützt mehrere Betriebssysteme, einschließlich Servern mit Windows Server. Dieser Agenten ist außerdem schneller.

  Wenn Sie den vereinheitlichten Agenten verwenden, um CloudWatch-Metriken zu erfassen, können Sie zusätzliche Systemmetriken für die Gast-Sichtbarkeit erfassen. Er unterstützt auch das Erfassen von benutzerdefinierten Metriken mithilfe von `StatsD` oder `collectd`.

  Weitere Informationen finden Sie unter Installieren des CloudWatch-Agenten im Amazon CloudWatch-Benutzerhandbuch.
- Unterstützt, aber bald veraltet: Der ältere CloudWatch Logs-Agent unterstützt nur die Erfassung von Protokollen von Servern, auf denen Linux ausgeführt wird. Wenn Sie diesen Agenten bereits verwenden, können Sie dies weiterhin tun. Der ältere Agent benötigt jedoch Python 2.7, 3.0 und 3.3. Da aktuelle EC2-Instances diese Python-Versionen nicht verwenden und diese Versionen veraltet sind und nicht mehr gepatcht werden, wird dringend empfohlen, dass Sie zum Unified CloudWatch Agent migrieren.

  Wenn Sie vom CloudWatch Logs-Agenten zum vereinheitlichten CloudWatch-Agenten migrieren, können Sie mit dem Assistenten des vereinheitlichten Agenten Ihre aktuelle CloudWatch Logs-Agentenkonfigurationsdatei einlesen und den neuen Agenten so einrichten, dass dieselben Protokolle erfasst werden. Weitere Informationen über den Assistenten finden Sie unter Erstellen der CloudWatch-Agent-Konfigurationsdatei mit dem Assistenten im Amazon CloudWatch-Benutzerhandbuch.

Inhalt:

- Verwenden des vereinheitlichten CloudWatch-Agenten für die ersten Schritte mit CloudWatch Logs (p. 5)
- Verwenden des vorherigen CloudWatch Logs-Agenten für die ersten Schritte mit CloudWatch Logs (p. 6)
- Schnellstart Verwenden AWS CloudFormation damit beginnen mit CloudWatch Logs (p. 34)

Verwenden des vereinheitlichten CloudWatch-Agenten für die ersten Schritte mit CloudWatch Logs

Weitere Informationen über die Verwendung des vereinheitlichten CloudWatch-Agenten für die ersten Schritte mit CloudWatch Logs finden Sie unter Erfassen von Metriken und Protokollen von Amazon EC2-Instances und lokalen Servern mit dem CloudWatch-Agenten im Amazon CloudWatch-Benutzerhandbuch. Führen Sie die in diesem Abschnitt aufgeführten Schritte aus, um den Agenten zu installieren, zu konfigurieren und zu starten. Wenn Sie den Agenten nicht auch für die Erfassung von CloudWatch-Metriken verwenden, können Sie alle Abschnitte ignorieren, die auf Metriken verweisen.

Wenn Sie aktuell den älteren CloudWatch Logs-Agenten verwenden und auf den neuen, vereinheitlichten Agenten migrieren wollen, empfehlen wir, dass Sie den Assistenten verwenden, der in dem Paket

mit dem neuen Agenten enthalten ist. Dieser Assistent kann Ihre aktuelle CloudWatch Logs-Agentenkonfigurationsdatei lesen und den CloudWatch-Agenten einrichten, um dieselben Protokolle zu erfassen. Weitere Informationen über den Assistenten finden Sie unter Erstellen der CloudWatch-Agent-Konfigurationsdatei mit dem Assistenten im Amazon CloudWatch-Benutzerhandbuch.

Verwenden des vorherigen CloudWatch Logs-Agenten für die ersten Schritte mit CloudWatch Logs

Mit dem CloudWatch Logs-Agenten können Sie Protokolldaten aus Amazon EC2-Instances unter Linux oder Windows Server sowie protokollierte Ereignisse aus AWS CloudTrail veröffentlichen. Wir empfehlen, stattdessen den vereinheitlichten CloudWatch-Agenten für die Veröffentlichung Ihrer Protokolldaten zu verwenden. Weitere Informationen über den neuen Agenten finden Sie unter Erfassen von Metriken und Protokollen von Amazon EC2-Instances und lokalen Servern mit dem CloudWatch-Agenten im Amazon CloudWatch-Benutzerhandbuch. Alternativ können Sie den vorherigen CloudWatch Logs-Agenten weiterhin verwenden.

Inhalt:

- Voraussetzungen für den CloudWatch Logs-Agenten (p. 6)
- Schnellstart Installieren und Konfigurieren CloudWatch Logs Agent auf einer laufenden EC2-Linux-Instanz (p. 7)
- Schnellstart Installieren und Konfigurieren CloudWatch Logs Agent auf einer EC2-Linux-Instanz bei Start (p. 12)
- Schnellstart Aktivieren Sie Ihre Amazon EC2 Instanzen, die Windows Server 2016 ausführen, um Protokolle zu senden an CloudWatch Logs Verwendung der CloudWatch Logs Agent (p. 14)
- Schnellstart Ermöglichen des Sendens von Protokollen an CloudWatch Logs durch Amazon EC2-Instances mit Windows Server 2012 und Windows Server 2008 (p. 22)
- Schnellstart Installieren Sie die CloudWatch Logs Agent Using AWS OpsWorks und Chefkoch (p. 29)
- Den Status des CloudWatch Logs-Agenten melden (p. 33)
- Den CloudWatch Logs-Agenten starten (p. 34)
- Beenden des CloudWatch Logs-Agenten (p. 34)

Voraussetzungen für den CloudWatch Logs-Agenten

Für den CloudWatch Logs-Agenten wird Python-Version 2.7, 3.0 oder 3.3 und eine der folgenden Linux-Versionen benötigt:

- Amazon Linux-Version 2014.03.02 oder höher Amazon Linux 2 wird nicht unterstützt
- Ubuntu-Serverversion 12.04, 14.04 oder 16.04
- CentOS-Version 6, 6.3, 6.4, 6.5 oder 7.0
- Version Red Hat Enterprise Linux (RHEL) 6.5 oder 7.0
- Debian 8.0

Schnellstart Installieren und Konfigurieren CloudWatch Logs Agent auf einer laufenden EC2-Linux-Instanz

Tip

CloudWatch umfasst einen neuen vereinheitlichten Agenten, der sowohl Protokolle als auch Metriken von EC2-Instances und lokalen Servern sammeln kann. Wenn Sie nicht bereits mit den älteren CloudWatch Logs-Agent verwenden, empfehlen wir die Verwendung des neueren vereinheitlichten CloudWatch-Agents. Weitere Informationen finden Sie im Erste Schritte mit CloudWatch Logs (p. 5).
Der Rest dieses Abschnitts beschreibt die Verwendung des älteren CloudWatch Logs-Agents.

Konfigurieren des älteren CloudWatch Logs-Agents auf einer ausgeführten EC2-Linux-Instance

Sie können das Installationsprogramm für den CloudWatch Logs-Agenten auf einer vorhandenen EC2-Instance verwenden, um den CloudWatch Logs-Agenten zu installieren und zu konfigurieren. Nachdem die Installation abgeschlossen ist, werden die Protokolle automatisch von der Instanz zum Protokoll-Stream geleitet, den Sie erstellen, während der Agent installiert wird. Sie erhalten vom Agenten eine Bestätigung, dass er gestartet wurde, und er wird weiterhin ausgeführt, bis Sie ihn deaktivieren.

Außer über den Agenten, können Sie auch über AWS CLI, CloudWatch Logs SDK oder die CloudWatch Logs-API Protokolldaten veröffentlichen. Die AWS CLI eignet sich am besten für die Veröffentlichung von Daten in der Befehlszeile oder mithilfe von Skripts. Das CloudWatch Logs SDK eignet sich am besten für die Veröffentlichung von Protokolldaten direkt aus Anwendungen heraus oder für die Erstellung einer eigenen Anwendung zum Veröffentlichen von Protokollen.

Schritt 1 Konfigurieren Sie Ihre IAM Rolle oder Benutzer für CloudWatch Logs

Der CloudWatch Logs-Agent unterstützt IAM-Rollen und -Benutzer. Verfügt Ihre Instance bereits über eine mit ihm verknüpfte IAM-Rolle, stellen Sie sicher, dass Sie die nachstehende IAM-Richtlinie einbinden. Wenn Ihrer Instance noch keine IAM-Rolle zugewiesen ist, können Sie Ihre IAM-Anmeldeinformationen für die nächsten Schritte verwenden oder dieser Instance eine IAM-Rolle zuweisen. Weitere Informationen finden Sie unter Verbinden einer IAM-Rolle mit einer Instance.

So konfigurieren Sie die IAM-Rolle oder den -Benutzer für CloudWatch Logs

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Roles (Rollen) aus.
3. Wählen Sie die Rolle, indem Sie den Rollennamen auswählen (aktivieren Sie nicht das Kontrollkästchen neben dem Namen).
4. Wählen Sie Attach Policies (Richtlinien anfügen), Create Policy (Richtlinie erstellen).

   Es wird eine neue Registerkarte im Browser oder ein neues Browser-Fenster geöffnet.
5. Wählen Sie die Registerkarte JSON aus und geben Sie den Text aus dem folgenden JSON-Richtliniendokument ein.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
      "Resource": [
        "*"
    ]
  }
 ]
}
```

6. Wählen Sie, wenn Sie fertig sind, Review policy (Richtlinie überprüfen) aus. Die Richtlinienvalidierung meldet mögliche Syntaxfehler.
7. Geben Sie auf der Seite Review Policy (Richtlinie überprüfen) unter Name einen Namen und unter Description (Beschreibung) (optional) eine Beschreibung für die Richtlinie ein, die Sie erstellen. Überprüfen Sie unter Summary (Zusammenfassung) die Richtlinienzusammenfassung, um die Berechtigungen einzusehen, die von Ihrer Richtlinie gewährt werden. Wählen Sie dann Create policy (Richtlinie erstellen) aus, um Ihre Eingaben zu speichern.
8. Schließen Sie die Registerkarte im Browser oder das Browser-Fenster, und gehen Sie zurück auf die Seite Add permissions (Berechtigungen hinzufügen) für Ihre Rolle. Klicken Sie erst auf Refresh (Aktualisieren), und wählen Sie dann die neue Richtlinie, um sie ihrer Rolle anzufügen.
9. Wählen Sie Attach Policy (Richtlinie anfügen) aus.

Schritt 2 Installieren und konfigurieren CloudWatch Logs auf einem vorhandenen Amazon EC2 Instanz

Die Vorgehensweise zum Installieren des CloudWatch Logs-Agenten ist eine andere, je nachdem ob auf Ihrer Amazon EC2-Instance Amazon Linux, Ubuntu, CentOS oder Red Hat ausgeführt wird. Führen Sie die für die Version von Linux entsprechenden Schritte auf Ihrer Instance aus.

So installieren und konfigurieren Sie CloudWatch Logs auf einer vorhandenen Amazon Linux-Instance

Ab Version Amazon Linux AMI 2014.09 steht der CloudWatch Logs-Agent als RPM-Installation mit dem awslogs-Paket zur Verfügung. Bei früheren Versionen von Amazon Linux kann der Zugriff auf das awslogs-Paket über die Aktualisierung Ihrer Instance mit dem Befehl `sudo yum update -y` erfolgen. Dadurch dass Sie das awslogs-Paket als RPM installieren, statt das Installationsprogramm von CloudWatch Logs zu verwenden, erhält Ihre Instance regelmäßige Paket-Updates und -Patches von AWS, ohne den CloudWatch Logs-Agenten manuell neu installieren zu müssen.

> **Warning**
>
> Aktualisieren Sie den CloudWatch Logs-Agenten nicht, indem Sie das Installationsverfahren von RPM anwenden, wenn Sie zuvor das Python-Skript verwendet haben, um den Agenten zu installieren. Dies kann zu Konfigurationsproblemen führen, die den CloudWatch Logs-Agenten daran hindern, Ihre Protokolle an CloudWatch zu versenden.

1. Stellen Sie eine Verbindung zu Ihrer Amazon Linux-Instance her. Weitere Informationen finden Sie unter Herstellen einer Verbindung mit Ihrer Instance im Amazon EC2-Benutzerhandbuch für Linux-Instances.

   Weitere Informationen über Verbindungsprobleme finden Sie unter Beheben von Verbindungsproblemen mit Ihrer Instance im Amazon EC2-Benutzerhandbuch für Linux-Instances.
2. Aktualisieren Sie Ihre Amazon Linux-Instance, um über die neuesten Änderungen in den Paket-Repositorys zu verfügen.

```
sudo yum update -y
```

3. Installieren Sie das Paket `awslogs`: Dies ist die empfohlene Methode zum Installieren von awslogs auf Amazon Linux-Instances.

   ```
   sudo yum install -y awslogs
   ```

4. Bearbeiten Sie die Datei `/etc/awslogs/awslogs.conf`, um die nachzuverfolgenden Protokolle zu konfigurieren. Weitere Informationen zum Bearbeiten dieser Datei finden Sie unter Referenz für den CloudWatch Logs-Agenten (p. 152).
5. Standardmäßig verweist `/etc/awslogs/awscli.conf` auf die Region us-east-1. Um Ihre Protokolle in eine andere Region zu verschieben, bearbeiten Sie die Datei `awscli.conf` und geben diese Region an.
6. Starten Sie den Service `awslogs`.

   ```
   sudo service awslogs start
   ```

   Wenn Sie Amazon Linux 2 verwenden, starten Sie den `awslogs`-Service mit dem folgenden Befehl.

   ```
   sudo systemctl start awslogsd
   ```

7. (Optional) Überprüfen Sie die Datei `/var/log/awslogs.log` auf Fehler, die beim Start des Services protokolliert wurden.
8. (Optional) Führen Sie den folgenden Befehl aus, um den Service `awslogs` bei jedem Systemstart zu starten.

   ```
   sudo chkconfig awslogs on
   ```

   Wenn Sie Amazon Linux 2 verwenden, starten Sie den Service bei jedem Systemstart mit dem folgenden Befehl.

   ```
   sudo systemctl enable awslogsd.service
   ```

9. Ihnen sollten in der CloudWatch-Konsole einen kurzen Moment nach dem Ausführen des Agenten die neu erstellte Protokollgruppe und der Protokoll-Stream angezeigt werden.

   Weitere Informationen finden Sie im An CloudWatch Logs gesendete Protokolldaten anzeigen lassen (p. 61).

So installieren und konfigurieren Sie CloudWatch Logs auf einer vorhandenen Ubuntu-Server-, CentOS- oder Red Hat-Instance

Wenn Sie ein AMI mit Ubuntu Server, CentOS oder Red Hat verwenden, verwenden Sie die folgende Vorgehensweise, um den CloudWatch Logs-Agent auf Ihrer Instance manuell zu installieren.

1. Stellen Sie eine Verbindung mit Ihrer EC2-Instance her: Weitere Informationen finden Sie unter Herstellen einer Verbindung mit Ihrer Instance im Amazon EC2-Benutzerhandbuch für Linux-Instances.

   Weitere Informationen über Verbindungsprobleme finden Sie unter Beheben von Verbindungsproblemen mit Ihrer Instance im Amazon EC2-Benutzerhandbuch für Linux-Instances.
2. Führen Sie das Installationsprogramm für den CloudWatch Logs-Agent mit ein oder zwei Optionen aus. Sie können es direkt aus dem Internet ausführen oder die Dateien herunterladen und es eigenständig ausführen.

Note

Wenn Sie CentOS 6.x, Red Hat 6.x oder Ubuntu 12.04 verwenden, verwenden Sie die Schritte zum Herunterladen und Ausführen des Installationsprogramms. Die Installation des CloudWatch Logs-Agenten direkt aus dem Internet wird auf diesen Systemen nicht unterstützt.

Note

Führen Sie auf Ubuntu `apt-get update` aus, bevor Sie die nachstehenden Befehle ausführen.

Um es direkt aus dem Internet auszuführen, verwenden Sie die folgenden Befehle und befolgen die Anweisungen:

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
```

```
sudo python ./awslogs-agent-setup.py --region us-east-1
```

Wenn der vorherige Befehl nicht funktioniert, versuchen Sie Folgendes:

```
sudo python3 ./awslogs-agent-setup.py --region us-east-1
```

Um es herunterzuladen und eigenständig auszuführen, verwenden Sie die folgenden Befehle und die Anweisungen:

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
```

```
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/AgentDependencies.tar.gz
 -O
```

```
tar xvf AgentDependencies.tar.gz -C /tmp/
```

```
sudo python ./awslogs-agent-setup.py --region us-east-1 --dependency-path /tmp/
AgentDependencies
```

Sie können den CloudWatch Logs-Agenten installieren, indem Sie die Regionen us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-1, eu-central-1, eu-west-1, or sa-east-1 angeben.

Note

Weitere Informationen zur aktuellen Version und zum Versionsverlauf von `awslogs-agent-setup` finden Sie in der Datei CHANGELOG.txt.

Das Installationsprogramm des CloudWatch Logs-Agenten verlangt bei der Einrichtung bestimmte Informationen. Bevor Sie beginnen, müssen Sie wissen, welche Protokolldatei zu überwachen ist und müssen dessen Zeitstempelformat kennen. Darüber hinaus sollten Sie auch die folgenden Informationen bereit halten.

Item	Description (Beschreibung)
AWS-Zugriffsschlüssel-ID	Drücken Sie die Eingabetaste, wenn die eine IAM-Rolle verwenden. Geben Sie ansonsten Ihre AWS-Zugriffsschlüssel-ID ein.
Geheimer AWS-Zugriffsschlüssel	Drücken Sie die Eingabetaste, wenn die eine IAM-Rolle verwenden. Geben Sie ansonsten Ihren geheimen AWS-Zugriffsschlüssel ein.
Standardmäßiger Regionsname	Drücken Sie die Eingabetaste. Der Standardwert ist us-east-2. Sie können diesen auf us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-1, eu-central-1, eu-west-1, or sa-east-1 setzen.
Standard-Ausgabeformat	Lassen Sie das Feld leer, und drücken Sie die Eingabetaste.
Pfad der hochzuladenden Protokolldatei	Der Speicherort der Datei, der die Protokolldaten enthält, die Sie senden möchten. Das Installationsprogramm schlägt Ihnen einen Pfad vor.
Ziel-Protokollgruppenname	Der Name für Ihre Protokollgruppe. Das Installationsprogramm schlägt Ihnen einen Protokollgruppenamen vor.
Ziel-Protokoll-Streamname	Standardmäßig ist dies der Name des Hosts. Das Installationsprogramm schlägt Ihnen einen Hostnamen vor.
Zeitstempelformat	Geben Sie das Format des Zeitstempels innerhalb der angegebenen Protokolldatei an. Wählen Sie „benutzerdefiniert”, um ihr eigenes Format anzugeben.
Ausgangsposition	Wie die Daten hochgeladen werden. Legen Sie diesen Wert auf start_of_file fest, um alles in die Datendatei hochzuladen. Legen Sie den Wert auf end_of_file fest, um nur neu hinzugefügte Daten hochzuladen.

Nachdem Sie diese Schritte ausgeführt haben, fragt das Installationsprogramm Sie, ob Sie eine weitere Protokolldatei konfigurieren möchten. Sie können den Vorgang für jede Protokolldatei beliebig oft durchführen. Wenn keine weiteren Protokolldateien überwacht werden sollen und ein weiteres Protokoll eingerichtet werden soll, wählen Sie N, wenn Sie vom Installationsprogramm dazu aufgefordert werden. Weitere Informationen zu den Einstellungen in der Konfigurationsdatei des Agenten finden Sie unter Referenz für den CloudWatch Logs-Agenten (p. 152).

Note

Die Konfiguration von mehreren Protokollquellen, damit Daten an einen einzelnen Protokoll-Stream gesendet werden, wird nicht unterstützt.

3. Ihnen sollten in der CloudWatch-Konsole einen kurzen Moment nach dem Ausführen des Agenten die neu erstellte Protokollgruppe und der Protokoll-Stream angezeigt werden.

   Weitere Informationen finden Sie im An CloudWatch Logs gesendete Protokolldaten anzeigen lassen (p. 61).

Schnellstart Installieren und Konfigurieren CloudWatch Logs Agent auf einer EC2-Linux-Instanz bei Start

Tip

Der ältere CloudWatch Logs-Agent, der in diesem Abschnitt beschrieben wird, wird demnächst veraltet sein. Es wird dringend empfohlen, stattdessen den neuen einheitlichen CloudWatch-Agenten zu verwenden, der Protokolle und Metriken erfassen kann. Darüber hinaus erfordert der ältere CloudWatch Logs-Agent Python 3.3 oder früher und diese Versionen werden standardmäßig nicht auf neuen EC2-Instances installiert. Weitere Informationen zum einheitlichen CloudWatch-Agenten finden Sie unter Installieren des CloudWatch- Agenten.
Der Rest dieses Abschnitts beschreibt die Verwendung des älteren CloudWatch Logs-Agents.

Installieren des älteren CloudWatch Logs-Agents auf einer EC2-Linux-Instance beim Start

Sie können die Amazon EC2-Benutzerdaten, eine Funktion von Amazon EC2 verwenden, die es ermöglicht, dass Parameterinformationen beim Start an die Instance übermittelt werden, um den CloudWatch Logs-Agenten auf dieser Instance zu installieren und zu konfigurieren. Um die Installations- und Konfigurationsinformationen des CloudWatch Logs-Agenten an Amazon EC2 zu übermitteln, können Sie die Konfigurationsdatei an einem Netzwerkstandort wie z. B. einem Amazon S3-Bucket bereitstellen.

Die Konfiguration von mehreren Protokollquellen, damit Daten an einen einzelnen Protokoll-Stream gesendet werden, wird nicht unterstützt.

Prerequisite

Erstellen Sie eine Konfigurationsdatei für den Agenten, in der alle Ihre Protokollgruppen und Protokollstreams beschrieben werden. Dies ist eine Textdatei, in der die zu überwachenden Protokolldateien sowie die Protokollgruppen und die Protokollstreams, in die diese hochgeladen werden sollen, beschrieben werden. Der Agent nutzt diese Konfigurationsdatei und startet das Überwachen und Hochladen aller darin beschriebenen Protokolldateien. Weitere Informationen zu den Einstellungen in der Konfigurationsdatei des Agenten finden Sie unter Referenz für den CloudWatch Logs-Agenten (p. 152).

Im Folgenden finden Sie ein Beispiel-Agenten-Konfigurationsdatei für Amazon Linux

```
[general]
state_file = /var/awslogs/state/agent-state

[/var/log/messages]
file = /var/log/messages
log_group_name = /var/log/messages
log_stream_name = {instance_id}
datetime_format = %b %d %H:%M:%S
```

Im Folgenden finden Sie ein Beispiel-Agenten-Konfigurationsdatei für Ubuntu

```
[general]
state_file = /var/awslogs/state/agent-state

[/var/log/syslog]
file = /var/log/syslog
log_group_name = /var/log/syslog
log_stream_name = {instance_id}
datetime_format = %b %d %H:%M:%S
```

So konfigurieren Sie Ihre IAM-Rolle

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Policies und Create Policy aus.
3. Wählen Sie auf der Seite Create Policy für Create Your Own Policy die Option Select aus. Weitere Informationen zum Erstellen von benutzerdefinierten Richtlinien finden Sie unter IAM-Richtlinien für Amazon EC2 im Amazon EC2-Benutzerhandbuch für Linux-Instances.
4. Geben Sie auf der Seite Review Policy im Feld Policy Name einen Namen für die Richtlinie ein.
5. Fügen Sie die folgende Richtlinie unter Policy Document ein:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams"
            ],
            "Resource": [
                "arn:aws:logs:*:*:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::myawsbucket/*"
            ]
        }
    ]
}
```

6. Wählen Sie Create Policy (Richtlinie erstellen) aus.
7. Wählen Sie im Navigationsbereich Roles und Create New Role aus.
8. Geben Sie auf der Seite Set Role Name einen Namen für die Rolle ein und wählen Sie dann Next Step aus.
9. Wählen Sie auf der Seite Select Role Type neben Amazon EC2 die Option Select aus.
10. Wählen Sie auf der Seite Attach Policy im Tabellenkopf die Option Policy Type und Customer Managed aus.
11. Wählen Sie die von Ihnen erstellte IAM-Richtlinie aus und anschließend Next Step (Nächster Schritt).
12. Wählen Sie Create Role aus.

    Weitere Informationen über IAM-Benutzer und -Richtlinien finden Sie unter IAM-Benutzer und -Gruppen und Verwalten von IAM-Richtlinien im IAM-Benutzerhandbuch.

So starten Sie eine neue Instance und aktivieren CloudWatch Logs

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie Launch Instance (Instance starten) aus.

   Weitere Informationen finden Sie unter Starten einer Instance in Amazon EC2-Benutzerhandbuch für Linux-Instances.

3. Am Schritt 1: Wählen Sie ein Amazon Machine Image (AMI) , wählen Sie den Linux-Instanztyp aus, und klicken Sie dann auf Schritt 2: Instanz-Typ auswählen Seite, wählen Sie Als Nächstes: Konfigurieren von Instance-Details

   Stellen Sie sicher, dass Cloud-Init ist in Ihrem Amazon Machine Image (AMI) enthalten. Amazon Linux Amis und Amis für Ubuntu und RHEL enthalten bereits Cloud-Init, Centos und andere Amide im AWS Marketplace darf nicht.
4. Am Schritt 3: Instanzdetails konfigurieren Seite, für IAM-Rolle, wählen Sie IAM Rolle, die Sie erstellt haben.
5. Fügen Sie unter Advanced Details für User data das folgende Skript in das Feld ein. Aktualisieren Sie dann das Skript, indem Sie den Wert der Option -c in den Speicherort Ihrer Agenten-Konfigurationsdatei ändern:

```
#!/bin/bash
curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
chmod +x ./awslogs-agent-setup.py
./awslogs-agent-setup.py -n -r us-east-1 -c s3://DOC-EXAMPLE-BUCKET1/my-config-file
```

6. Nehmen Sie weitere Änderungen an der Instance vor, prüfen Sie die Starteinstellungen und wählen Sie dann Launch aus.
7. Ihnen sollten in der CloudWatch-Konsole einen kurzen Moment nach dem Ausführen des Agenten die neu erstellte Protokollgruppe und der Protokoll-Stream angezeigt werden.

   Weitere Informationen finden Sie im An CloudWatch Logs gesendete Protokolldaten anzeigen lassen (p. 61).

Schnellstart Aktivieren Sie Ihre Amazon EC2 Instanzen, die Windows Server 2016 ausführen, um Protokolle zu senden an CloudWatch Logs Verwendung der CloudWatch Logs Agent

Tip

CloudWatch umfasst einen neuen vereinheitlichten Agenten, der sowohl Protokolle als auch Metriken von EC2-Instances und lokalen Servern sammeln kann. Wir empfehlen die Verwendung des neueren vereinheitlichten CloudWatch-Agents. Weitere Informationen finden Sie im Erste Schritte mit CloudWatch Logs (p. 5).
Der Rest dieses Abschnitts beschreibt die Verwendung des älteren CloudWatch Logs-Agents.

Ermöglichen des Sendens von Protokollen an CloudWatch Logs mit dem älteren CloudWatch Logs-Agent für Amazon EC2-Instances mit Windows Server 2016

Es gibt eine Vielzahl von Methoden, die Sie verwenden können, um Instances, die auf Windows Server 2016 laufen, in die Lage zu versetzen, Protokolle an CloudWatch Logs zu senden. Die Schritte in diesem Abschnitt verwenden Systems Manager Run Command. Weitere Informationen zu anderen möglichen Methoden finden Sie unter Senden von Protokollen, Ereignissen und Leistungsindikatoren an Amazon CloudWatch.

Schritte

- Herunterladen der Beispielkonfigurationsdatei (p. 15)
- Konfigurieren der JSON-Datei für CloudWatch (p. 15)

- Erstellen eines IAM-Benutzers und einer -Rolle für Systems Manager (p. 21)
- Überprüfen der Systems Manager-Voraussetzungen (p. 21)
- Überprüfen des Internetzugangs (p. 21)
- Aktivieren von CloudWatch Logs mithilfe von Systems Manager Run Command (p. 21)

Herunterladen der Beispielkonfigurationsdatei

Laden Sie die folgende Beispieldatei auf Ihren Computer herunter: AWS.EC2.Windows.cloudwatch.json.

Konfigurieren der JSON-Datei für CloudWatch

Sie bestimmen, welche Protokolle an CloudWatch gesendet werden, indem Sie Ihre Auswahl in einer Konfigurationsdatei angeben. Das Erstellen dieser Datei und die Angabe Ihrer Auswahl können 30 Minuten oder mehr beanspruchen. Nachdem Sie diese Aufgabe abgeschlossen haben, können Sie die Konfigurationsdatei für alle Instances wiederverwenden.

Schritte

- Schritt 1 Cloudwatch-Protokolle aktivieren (p. 15)
- Schritt 2 Einstellungen konfigurieren für CloudWatch (p. 15)
- Schritt 3 Die zu sendenden Daten konfigurieren (p. 16)
- Schritt 4. Flow Control konfigurieren (p. 20)
- Schritt 5. JSON-Inhalt speichern (p. 21)

Schritt 1 Cloudwatch-Protokolle aktivieren

Ändern Sie oben in der JSON-Datei "false" in "true" für `IsEnabled`:

```
"IsEnabled": true,
```

Schritt 2 Einstellungen konfigurieren für CloudWatch

Geben Sie die Anmeldeinformationen, die Region, eine Protokollgruppennamen und einen Protokoll-Stream-Namespace ein. Auf diese Weise kann die Instance Protokolldaten an CloudWatch Logs senden. Wenn Sie dieselben Protokolldaten an verschiedene Standorte schicken möchten, können Sie weitere Abschnitte mit eindeutigen IDs (z. B. „CloudWatchLogs2“ und „CloudWatchLogs3“) sowie einer unterschiedlichen Region für jede ID hinzufügen.

So konfigurieren Sie Einstellungen zum Senden von Protokolldaten an CloudWatch Logs

1. Suchen Sie in der JSON-Datei nach dem Abschnitt `CloudWatchLogs`.

```
{
    "Id": "CloudWatchLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "AccessKey": "",
        "SecretKey": "",
        "Region": "us-east-1",
        "LogGroup": "Default-Log-Group",
        "LogStream": "{instance_id}"
    }
},
```

2. Lassen Sie die Felder `AccessKey` und `SecretKey` leer. Sie können mithilfe einer IAM-Rolle die Anmeldeinformationen konfigurieren.

3. Geben Sie für `Region` die Region ein, an die Sie die Protokolldaten senden möchten (z. B. `us-east-2`).
4. Geben Sie in das Feld `LogGroup` den Namen Ihrer Protokollgruppe ein. Dieser Name wird auf dem Bildschirm Log Groups in der CloudWatch-Konsole angezeigt.
5. Geben Sie für `LogStream` den Ziel-Protokoll-Stream ein. Dieser Name wird auf dem Bildschirm Log Groups > Streams (Protokollgruppen > Streams) in der CloudWatch-Konsole angezeigt.

   Wenn Sie `{instance_id}` verwenden, ist der Standard-Protokoll-Stream die Instance-ID dieser Instance.

   Wenn Sie einen Protokoll-Stream-Namen angeben, der noch nicht vorhanden ist, erstellt CloudWatch Logs ihn automatisch. Sie können den Protokoll-Stream-Namen mithilfe einer Literalzeichenfolge oder den vordefinierten Variablen `{instance_id}`, `{hostname}` und `{ip_address}` oder einer Kombination aus beiden definieren.

Schritt 3 Die zu sendenden Daten konfigurieren

Sie können Ereignisprotokolldaten, Daten zu Ablaufverfolgung von Ereignissen für Windows (ETW) und andere Protokolldaten an CloudWatch Logs senden.

So senden Sie Windows-Anwendungsereignisprotokolldaten an CloudWatch Logs

1. Suchen Sie in der JSON-Datei nach dem Abschnitt `ApplicationEventLog`.

```
{
    "Id": "ApplicationEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Application",
        "Levels": "1"
    }
},
```

2. Geben Sie für `Levels` den hochzuladenden Meldungstyp an. Sie können einen der folgenden Werte angeben:

   - **`1`** - Nur Fehlermeldungen hochladen.
   - **`2`** - Nur Warnmeldungen hochladen.
   - **`4`** - Nur Informationsmeldungen hochladen.

   Sie können Werte kombinieren, um mehr als einen Meldungstyp einzuschließen. Beispielsweise lädt ein Wert von **`3`** Fehlermeldungen (**`1`**) und Warnmeldungen (**`2`**) hoch. Ein Wert von **`7`** lädt Fehlermeldungen (**`1`**), Warnmeldungen (**`2`**) und Informationsmeldungen (**`4`**) hoch.

So senden Sie Sicherheitsprotokolldaten an CloudWatch Logs

1. Suchen Sie in der JSON-Datei nach dem Abschnitt `SecurityEventLog`.

```
{
    "Id": "SecurityEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Security",
        "Levels": "7"
    }
```

```
},
```

2. Geben Sie für `Levels` **`7`** ein, um alle Meldungen hochzuladen.

So senden Sie Systemereignisprotokolldaten an CloudWatch Logs

1. Suchen Sie in der JSON-Datei nach dem Abschnitt `SystemEventLog`.

```
{
    "Id": "SystemEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
        "Levels": "7"
    }
},
```

2. Geben Sie für `Levels` den hochzuladenden Meldungstyp an. Sie können einen der folgenden Werte angeben:

   - **`1`** - Nur Fehlermeldungen hochladen.
   - **`2`** - Nur Warnmeldungen hochladen.
   - **`4`** - Nur Informationsmeldungen hochladen.

   Sie können Werte kombinieren, um mehr als einen Meldungstyp einzuschließen. Beispielsweise lädt ein Wert von **`3`** Fehlermeldungen (**`1`**) und Warnmeldungen (**`2`**) hoch. Ein Wert von **`7`** lädt Fehlermeldungen (**`1`**), Warnmeldungen (**`2`**) und Informationsmeldungen (**`4`**) hoch.

So senden Sie andere Arten von Ereignisprotokolldaten an CloudWatch Logs

1. Fügen Sie der JSON-Datei einen neuen Abschnitt hinzu. Jeder Abschnitt muss eine eindeutige `Id` haben.

```
{
    "Id": "Id-name",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Log-name",
        "Levels": "7"
    }
},
```

2. Geben Sie für `Id` einen Namen für das hochzuladende Protokoll ein (z. B. **`WindowsBackup`**).
3. Geben Sie für `LogName` den Namen des hochzuladenden Protokolls ein. Sie können den Namen des Protokolls wie folgt suchen.

   a. Öffnen Sie die Ereignisanzeige.

   b. Wählen Sie im Navigationsbereich Applications and Services Logs aus.

   c. Navigieren Sie zum Protokoll, und wählen Sie dann Actions und Properties aus.
4. Geben Sie für `Levels` den hochzuladenden Meldungstyp an. Sie können einen der folgenden Werte angeben:

   - **`1`** - Nur Fehlermeldungen hochladen.
   - **`2`** - Nur Warnmeldungen hochladen.

- **4** - Nur Informationsmeldungen hochladen.

Sie können Werte kombinieren, um mehr als einen Meldungstyp einzuschließen. Beispielsweise lädt ein Wert von **3** Fehlermeldungen (**1**) und Warnmeldungen (**2**) hoch. Ein Wert von **7** lädt Fehlermeldungen (**1**), Warnmeldungen (**2**) und Informationsmeldungen (**4**) hoch.

So senden Sie Event Tracing for Windows-Daten an CloudWatch Logs

ETW (Event Tracing for Windows) bietet einen effizienten und detaillierten Protokollierungsmechanismus, auf den Anwendungen Protokolle schreiben können. Jeder ETW wird über einen Session Manager gesteuert, der die Protokollierungssitzung starten und beenden kann. Jede Sitzung hat einen Anbieter und einen oder mehrere Verbraucher.

1. Suchen Sie in der JSON-Datei nach dem Abschnitt `ETW`.

```
{
    "Id": "ETW",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Microsoft-Windows-WinINet/Analytic",
        "Levels": "7"
    }
},
```

2. Geben Sie für `LogName` den Namen des hochzuladenden Protokolls ein.
3. Geben Sie für `Levels` den hochzuladenden Meldungstyp an. Sie können einen der folgenden Werte angeben:

   - **1** - Nur Fehlermeldungen hochladen.
   - **2** - Nur Warnmeldungen hochladen.
   - **4** - Nur Informationsmeldungen hochladen.

   Sie können Werte kombinieren, um mehr als einen Meldungstyp einzuschließen. Beispielsweise lädt ein Wert von **3** Fehlermeldungen (**1**) und Warnmeldungen (**2**) hoch. Ein Wert von **7** lädt Fehlermeldungen (**1**), Warnmeldungen (**2**) und Informationsmeldungen (**4**) hoch.

So senden Sie benutzerdefinierte Protokolle (jede beliebige textbasierte Protokolldatei) an CloudWatch Logs

1. Suchen Sie in der JSON-Datei nach dem Abschnitt `CustomLogs`.

```
{
    "Id": "CustomLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\CustomLogs\\",
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "Local",
        "LineCount": "5"
    }
},
```

2. Geben Sie für `LogDirectoryPath` den Pfad ein, auf dem die Protokolle auf Ihrer Instance gespeichert werden sollen.
3. Geben Sie unter `TimestampFormat` das Zeitstempelformat ein, das Sie verwenden möchten. Weitere Informationen über unterstützte Werte finden Sie unter dem Thema Benutzerdefinierte Datums- und Zeitformat-Zeichenfolgen auf MSDN.

   Important

   Die Quell-Protokolldatei muss zu Beginn jeder Protokollzeile einen Zeitstempel haben, und nach dem Zeitstempel muss eine Leerstelle folgen.
4. Geben Sie für `Encoding` die zu verwendende Datei-Kodierung ein (z. B: UTF-8). Eine Liste der unterstützten Werte finden Sie unter dem Thema Encoding Class auf MSDN.

   Note

   Verwenden Sie den Kodierungsnamen, nicht den Anzeigenamen.
5. (Optional) Geben Sie für `Filter` das Präfix des Protokollnamens ein. Lassen Sie diesen Parameter leer, um alle Dateien zu überwachen. Weitere Informationen über unterstützte Werte finden Sie unter dem Thema FileSystemWatcherFilter-Eigenschaft auf MSDN.
6. (Optional) Geben Sie für `CultureName` das Gebietsschema ein, unter dem Zeitstempel protokolliert wird. Wenn `CultureName` leer ist, wird standardmäßig dasselbe Gebietsschema verwendet, das von der Windows-Instance verwendet wird. Weitere Informationen finden Sie in der Spalte `Language tag` in der Tabelle im Thema Produktverhalten in MSDN.

   Note

   Die `div`, `div-MV`, `hu`, und `hu-HU` Werte werden nicht unterstützt.
7. (Optional) für `TimeZoneKind`, Typ `Local` oder `UTC`. Sie können dies einstellen, um Zeitzoneninformationen anzugeben, wenn keine Zeitzoneninformationen in den Zeitstempel Ihres Protokolls enthalten sind. Wenn dieser Parameter leer bleibt und Ihr Zeitstempel keine Zeitzoneninformationen enthält, verwendet CloudWatch Logs standardmäßig die lokale Zeitzone. Dieser Parameter wird ignoriert, wenn der Zeitstempel bereits Zeitzoneninformationen enthält.
8. (Optional) Geben Sie für `LineCount` die Anzahl der Zeilen im Header ein, um die Protokolldatei zu identifizieren. Beispielsweise haben IIS-Protokolldateien praktisch identische Header. Sie können **5** eingeben, dann würden die ersten drei Zeilen des Headers der Protokolldatei gelesen, um diese zu identifizieren. In den IIS-Protokolldateien ist die dritte Zeile das Datum und der Zeitstempel, aber es ist nicht immer sichergestellt, dass der Zeitstempel von zwei verschiedenen Protokolldateien unterschiedlich ist. Aus diesem Grund empfehlen wir, mindestens eine Zeile der tatsächlichen Protokolldaten hinzuzufügen, so dass die Protokolldatei eindeutig identifizierbar ist.

So senden Sie IIS-Protokolldaten an CloudWatch Logs

1. Suchen Sie in der JSON-Datei nach dem Abschnitt `IISLog`.

```
{
    "Id": "IISLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "UTC",
        "LineCount": "5"
    }
},
```

2. Geben Sie für `LogDirectoryPath` das Verzeichnis an, in dem die IIS-Protokolldateien für einen individuellen Standort gespeichert sind (z. B. `C:\inetpub\logs\LogFiles\W3SVC`*n*).

   Note

   Es wird nur das W3C-Protokollformat unterstützt. IIS, NCSA und benutzerdefinierte Formate werden nicht unterstützt.

3. Geben Sie unter `TimestampFormat` das Zeitstempelformat ein, das Sie verwenden möchten. Weitere Informationen über unterstützte Werte finden Sie unter dem Thema Benutzerdefinierte Datums- und Zeitformat-Zeichenfolgen auf MSDN.
4. Geben Sie für `Encoding` die zu verwendende Datei-Kodierung ein (z. B: UTF-8). Weitere Informationen über unterstützte Werte finden Sie unter dem Thema Encoding Class auf MSDN.

   Note

   Verwenden Sie den Kodierungsnamen, nicht den Anzeigenamen.

5. (Optional) Geben Sie für `Filter` das Präfix des Protokollnamens ein. Lassen Sie diesen Parameter leer, um alle Dateien zu überwachen. Weitere Informationen über unterstützte Werte finden Sie unter dem Thema FileSystemWatcherFilter-Eigenschaft auf MSDN.
6. (Optional) Geben Sie für `CultureName` das Gebietsschema ein, unter dem Zeitstempel protokolliert wird. Wenn `CultureName` leer ist, wird standardmäßig dasselbe Gebietsschema verwendet, das von der Windows-Instance verwendet wird. Weitere Informationen über unterstützte Werte finden Sie in der Spalte `Language tag` in der Tabelle im Thema Produktverhalten in MSDN.

   Note

   Die `div`, `div-MV`, `hu`, und `hu-HU` Werte werden nicht unterstützt.

7. (Optional) für `TimeZoneKind`, geben Sie `Local` oder `UTC`. Sie können dies einstellen, um Zeitzoneninformationen anzugeben, wenn keine Zeitzoneninformationen in den Zeitstempel Ihres Protokolls enthalten sind. Wenn dieser Parameter leer bleibt und Ihr Zeitstempel keine Zeitzoneninformationen enthält, verwendet CloudWatch Logs standardmäßig die lokale Zeitzone. Dieser Parameter wird ignoriert, wenn der Zeitstempel bereits Zeitzoneninformationen enthält.
8. (Optional) Geben Sie für `LineCount` die Anzahl der Zeilen im Header ein, um die Protokolldatei zu identifizieren. Beispielsweise haben IIS-Protokolldateien praktisch identische Header. Sie können **`5`** eingeben, dann würden die ersten fünf Zeilen des Headers der Protokolldatei gelesen, um diese zu identifizieren. In den IIS-Protokolldateien ist die dritte Zeile das Datum und der Zeitstempel, aber es ist nicht immer sichergestellt, dass der Zeitstempel von zwei verschiedenen Protokolldateien unterschiedlich ist. Aus diesem Grund empfehlen wir, mindestens eine Zeile der tatsächlichen Protokolldaten hinzuzufügen, sodass die Protokolldatei eindeutig identifizierbar ist.

Schritt 4. Flow Control konfigurieren

Jeder Datentyp muss ein entsprechendes Ziel im Bereich `Flows` haben. Um zum Beispiel ein kundendefiniertes Protokoll, ein ETW-Protokoll und ein Systemprotokoll an CloudWatch Logs zu senden, fügen Sie dem Bereich `Flows` Folgendes hinzu: `(CustomLogs,ETW,SystemEventLog),CloudWatchLogs`.

Warning

Wenn Sie einen ungültigen Schritt hinzufügen, ist der Fluss blockiert. Wenn Sie beispielsweise einen Metrikschritt für eine Festplatte hinzufügen, Ihre Instance aber keine Festplatte hat, sind alle Schritte im Fluss blockiert.

Sie können dieselbe Protokolldatei an mehrere Ziele senden. Wenn Sie beispielsweise das Anwendungsprotokoll an zwei unterschiedliche Ziele senden möchten, die Sie im Abschnitt `CloudWatchLogs` definiert haben, fügen Sie dem Abschnitt `Flows` Folgendes hinzu: `ApplicationEventLog,(CloudWatchLogs,CloudWatchLogs2)`.

So konfigurieren Sie die Flussteuerung:

1. Suchen Sie in der Datei `AWS.EC2.Windows.CloudWatch.json` den Abschnitt `Flows`.

```
"Flows": {
    "Flows": [
      "PerformanceCounter,CloudWatch",
      "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
      "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
      "CustomLogs, CloudWatchLogs2",
      "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
    ]
}
```

2. Geben Sie für `Flows` jeden Datentyp ein, der hochgeladen werden soll (z. B. `ApplicationEventLog`) sowie sein Ziel (z. B. `CloudWatchLogs`).

Schritt 5. JSON-Inhalt speichern

Sie haben jetzt die Bearbeitung der JSON-Datei abgeschlossen. Speichern Sie sie und fügen Sie den Dateiinhalt in einem anderen Fenster in einen Texteditor ein. Sie benötigen den Dateiinhalt in einem späteren Schritt dieses Vorgangs.

Erstellen eines IAM-Benutzers und einer -Rolle für Systems Manager

Sie benötigen eine IAM-Rolle für Instance-Anmeldeinformationen, wenn Sie Systems Manager Run Command verwenden. Diese Rolle ermöglicht Systems Manager, Aktionen auf der Instance auszuführen. Sie können optional ein eindeutiges IAM-Benutzerkonto erstellen, um Systems Manager zu konfigurieren und auszuführen. Weitere Informationen finden Sie unter Konfigurieren von Sicherheitsrollen für Systems Manager im AWS Systems Manager-Benutzerhandbuch. Weitere Informationen dazu, wie Sie ein IAM-Instance-Profil mit einer vorhandenen Instance verbinden, finden Sie unter Verbinden einer IAM-Rolle mit einer Instance im Amazon EC2-Benutzerhandbuch für Windows-Instances.

Überprüfen der Systems Manager-Voraussetzungen

Bevor Sie Systems Manager Run Command verwenden, um die Integration mit CloudWatch Logs zu konfigurieren, überprüfen Sie, ob Ihre Instances die Mindestanforderungen erfüllen. Weitere Informationen finden Sie unter Systems Manager-Voraussetzungen im AWS Systems Manager-Benutzerhandbuch.

Überprüfen des Internetzugangs

Ihre Amazon EC2-Windows-Server-Instances und verwaltete Instances müssen über ausgehenden Internetzugang verfügen, um Protokoll- und Ereignisdaten an CloudWatch zu senden. Weitere Informationen dazu, wie Sie den Internetzugang konfigurieren, finden Sie unter Internet-Gateways im Amazon VPC Benutzerhandbuch.

Aktivieren von CloudWatch Logs mithilfe von Systems Manager Run Command

Run Command ermöglicht Ihnen die bedarfsgerechte Verwaltung der Konfiguration Ihrer Instances. Sie geben ein Systems Manager-Dokument und Parameter an und führen den Befehl auf mindestens einer Instance aus. Der SSM-Agent auf der Instance verarbeitet den Befehl und konfiguriert die Instance wie angegeben.

So konfigurieren Sie die Integration mit CloudWatch Logs mithilfe von Run Command

1. Öffnen Sie die Amazon EC2-Konsole unter https://console.aws.amazon.com/ec2/.
2. Öffnen Sie die SSM-Konsole unter https://console.aws.amazon.com/systems-manager/.

3. Wählen Sie im Navigationsbereich die Option Run Command (Befehl ausführen).
4. Wählen Sie die Option Run a command.
5. Wählen Sie für Command document die Option AWS-ConfigureCloudWatch aus.
6. Für Target instances wählen Sie die Instances, die in CloudWatch Logs integriert werden sollen. Wenn keine Instanz in der Liste angezeigt wird, ist möglicherweise für Run Command keine konfiguriert. Weitere Informationen finden Sie unter Systems Manager-Voraussetzungen im Amazon EC2-Benutzerhandbuch für Windows-Instances.
7. Wählen Sie für Status die Option Enabled.
8. Für Properties kopieren Sie die JSON-Inhalte, die Sie in den vorherigen Schritten erstellt haben, und fügen Sie diese ein.
9. Füllen Sie die verbleibenden optionalen Felder aus, und wählen Sie Run.

Befolgen Sie die folgende Prozedur, um die Ergebnisse der Befehlsausführung in der Amazon EC2-Konsole anzuzeigen.

So zeigen Sie die Befehlsausgabe in der Konsole an:

1. Wählen Sie einen Befehl.
2. Wählen Sie die Registerkarte Output aus.
3. Wählen Sie View Output aus. Der Befehlsausgabeseite zeigt die Ergebnisse der Befehlsausführung an.

Schnellstart Ermöglichen des Sendens von Protokollen an CloudWatch Logs durch Amazon EC2-Instances mit Windows Server 2012 und Windows Server 2008

> Tip
>
> CloudWatch umfasst einen neuen vereinheitlichten Agenten, der sowohl Protokolle als auch Metriken von EC2-Instances und lokalen Servern sammeln kann. Wir empfehlen die Verwendung des neueren vereinheitlichten CloudWatch-Agents. Weitere Informationen finden Sie im Erste Schritte mit CloudWatch Logs (p. 5).
> Der Rest dieses Abschnitts beschreibt die Verwendung des älteren CloudWatch Logs-Agents.

Ermöglichen des Sendens von Protokollen an CloudWatch Logs durch Amazon EC2-Instances mit Windows Server 2012 und Windows Server 2008

Führen Sie die folgenden Schritte aus, um Instances, die mit Windows Server 2012 und Windows Server 2008 laufen, zu ermöglichen, Protokolle an CloudWatch Logs zu senden.

Herunterladen der Beispielkonfigurationsdatei

Laden Sie die folgende Probe-JSON-Datei auf Ihren Computer herunter: AWS.EC2.Windows.cloudwatch.json. Sie werden sie in den folgenden Schritten bearbeiten.

Konfigurieren der JSON-Datei für CloudWatch

Sie bestimmen, welche Protokolle an CloudWatch gesendet werden, indem Sie Ihre Auswahl in der JSON-Konfigurationsdatei angeben. Das Erstellen dieser Datei und die Angabe Ihrer Auswahl können

30 Minuten oder mehr beanspruchen. Nachdem Sie diese Aufgabe abgeschlossen haben, können Sie die Konfigurationsdatei für alle Instances wiederverwenden.

Schritte

- Schritt 1 Cloudwatch-Protokolle aktivieren (p. 23)
- Schritt 2 Einstellungen konfigurieren für CloudWatch (p. 23)
- Schritt 3 Die zu sendenden Daten konfigurieren (p. 24)
- Schritt 4. Flow Control konfigurieren (p. 28)

Schritt 1 Cloudwatch-Protokolle aktivieren

Ändern Sie oben in der JSON-Datei "false" in "true" für `IsEnabled`:

```
"IsEnabled": true,
```

Schritt 2 Einstellungen konfigurieren für CloudWatch

Geben Sie die Anmeldeinformationen, die Region, eine Protokollgruppennamen und einen Protokoll-Stream-Namespace ein. Auf diese Weise kann die Instance Protokolldaten an CloudWatch Logs senden. Wenn Sie dieselben Protokolldaten an verschiedene Standorte schicken möchten, können Sie weitere Abschnitte mit eindeutigen IDs (z. B. „CloudWatchLogs2“ und „CloudWatchLogs3“) sowie einer unterschiedlichen Region für jede ID hinzufügen.

So konfigurieren Sie Einstellungen zum Senden von Protokolldaten an CloudWatch Logs

1. Suchen Sie in der JSON-Datei nach dem Abschnitt `CloudWatchLogs`.

   ```
   {
       "Id": "CloudWatchLogs",
       "FullName":
    "AWS.EC2.Windows.CloudWatch.CloudWatchLogsOutput,AWS.EC2.Windows.CloudWatch",
       "Parameters": {
           "AccessKey": "",
           "SecretKey": "",
           "Region": "us-east-1",
           "LogGroup": "Default-Log-Group",
           "LogStream": "{instance_id}"
       }
   },
   ```

2. Lassen Sie die Felder `AccessKey` und `SecretKey` leer. Sie können mithilfe einer IAM-Rolle die Anmeldeinformationen konfigurieren.
3. Geben Sie für `Region` die Region ein, an die Sie die Protokolldaten senden möchten (z. B. `us-east-2`).
4. Geben Sie in das Feld `LogGroup` den Namen Ihrer Protokollgruppe ein. Dieser Name wird auf dem Bildschirm Log Groups in der CloudWatch-Konsole angezeigt.
5. Geben Sie für `LogStream` den Ziel-Protokoll-Stream ein. Dieser Name wird auf dem Bildschirm Log Groups > Streams (Protokollgruppen > Streams) in der CloudWatch-Konsole angezeigt.

   Wenn Sie `{instance_id}` verwenden, ist der Standard-Protokoll-Stream die Instance-ID dieser Instance.

   Wenn Sie einen Protokoll-Stream-Namen angeben, der noch nicht vorhanden ist, erstellt CloudWatch Logs ihn automatisch. Sie können den Protokoll-Stream-Namen mithilfe einer Literalzeichenfolge oder den vordefinierten Variablen `{instance_id}`, `{hostname}` und `{ip_address}` oder einer Kombination aus beiden definieren.

Schritt 3 Die zu sendenden Daten konfigurieren

Sie können Ereignisprotokolldaten, Daten zu Ablaufverfolgung von Ereignissen für Windows (ETW) und andere Protokolldaten an CloudWatch Logs senden.

So senden Sie Windows-Anwendungsereignisprotokolldaten an CloudWatch Logs

1. Suchen Sie in der JSON-Datei nach dem Abschnitt `ApplicationEventLog`.

```
{
    "Id": "ApplicationEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Application",
        "Levels": "1"
    }
},
```

2. Geben Sie für `Levels` den hochzuladenden Meldungstyp an. Sie können einen der folgenden Werte angeben:

   - **`1`** - Nur Fehlermeldungen hochladen.
   - **`2`** - Nur Warnmeldungen hochladen.
   - **`4`** - Nur Informationsmeldungen hochladen.

   Sie können Werte kombinieren, um mehr als einen Meldungstyp einzuschließen. Beispielsweise lädt ein Wert von **`3`** Fehlermeldungen (**`1`**) und Warnmeldungen (**`2`**) hoch. Ein Wert von **`7`** lädt Fehlermeldungen (**`1`**), Warnmeldungen (**`2`**) und Informationsmeldungen (**`4`**) hoch.

So senden Sie Sicherheitsprotokolldaten an CloudWatch Logs

1. Suchen Sie in der JSON-Datei nach dem Abschnitt `SecurityEventLog`.

```
{
    "Id": "SecurityEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Security",
        "Levels": "7"
    }
},
```

2. Geben Sie für `Levels` **`7`** ein, um alle Meldungen hochzuladen.

So senden Sie Systemereignisprotokolldaten an CloudWatch Logs

1. Suchen Sie in der JSON-Datei nach dem Abschnitt `SystemEventLog`.

```
{
    "Id": "SystemEventLog",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "System",
        "Levels": "7"
    }
```

```
},
```

2. Geben Sie für `Levels` den hochzuladenden Meldungstyp an. Sie können einen der folgenden Werte angeben:

   - **`1`** - Nur Fehlermeldungen hochladen.
   - **`2`** - Nur Warnmeldungen hochladen.
   - **`4`** - Nur Informationsmeldungen hochladen.

   Sie können Werte kombinieren, um mehr als einen Meldungstyp einzuschließen. Beispielsweise lädt ein Wert von **`3`** Fehlermeldungen (**`1`**) und Warnmeldungen (**`2`**) hoch. Ein Wert von **`7`** lädt Fehlermeldungen (**`1`**), Warnmeldungen (**`2`**) und Informationsmeldungen (**`4`**) hoch.

So senden Sie andere Arten von Ereignisprotokolldaten an CloudWatch Logs

1. Fügen Sie der JSON-Datei einen neuen Abschnitt hinzu. Jeder Abschnitt muss eine eindeutige `Id` haben.

```
{
    "Id": "Id-name",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Log-name",
        "Levels": "7"
    }
},
```

2. Geben Sie für `Id` einen Namen für das hochzuladende Protokoll ein (z. B. **`WindowsBackup`**).
3. Geben Sie für `LogName` den Namen des hochzuladenden Protokolls ein. Sie können den Namen des Protokolls wie folgt suchen.

   a. Öffnen Sie die Ereignisanzeige.

   b. Wählen Sie im Navigationsbereich Applications and Services Logs aus.

   c. Navigieren Sie zum Protokoll, und wählen Sie dann Actions und Properties aus.
4. Geben Sie für `Levels` den hochzuladenden Meldungstyp an. Sie können einen der folgenden Werte angeben:

   - **`1`** - Nur Fehlermeldungen hochladen.
   - **`2`** - Nur Warnmeldungen hochladen.
   - **`4`** - Nur Informationsmeldungen hochladen.

   Sie können Werte kombinieren, um mehr als einen Meldungstyp einzuschließen. Beispielsweise lädt ein Wert von **`3`** Fehlermeldungen (**`1`**) und Warnmeldungen (**`2`**) hoch. Ein Wert von **`7`** lädt Fehlermeldungen (**`1`**), Warnmeldungen (**`2`**) und Informationsmeldungen (**`4`**) hoch.

So senden Sie Event Tracing for Windows-Daten an CloudWatch Logs

ETW (Event Tracing for Windows) bietet einen effizienten und detaillierten Protokollierungsmechanismus, auf den Anwendungen Protokolle schreiben können. Jeder ETW wird über einen Session Manager gesteuert, der die Protokollierungssitzung starten und beenden kann. Jede Sitzung hat einen Anbieter und einen oder mehrere Verbraucher.

1. Suchen Sie in der JSON-Datei nach dem Abschnitt `ETW`.

```
{
    "Id": "ETW",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.EventLog.EventLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogName": "Microsoft-Windows-WinINet/Analytic",
        "Levels": "7"
    }
},
```

2. Geben Sie für `LogName` den Namen des hochzuladenden Protokolls ein.
3. Geben Sie für `Levels` den hochzuladenden Meldungstyp an. Sie können einen der folgenden Werte angeben:

   - **`1`** - Nur Fehlermeldungen hochladen.
   - **`2`** - Nur Warnmeldungen hochladen.
   - **`4`** - Nur Informationsmeldungen hochladen.

   Sie können Werte kombinieren, um mehr als einen Meldungstyp einzuschließen. Beispielsweise lädt ein Wert von **`3`** Fehlermeldungen (**`1`**) und Warnmeldungen (**`2`**) hoch. Ein Wert von **`7`** lädt Fehlermeldungen (**`1`**), Warnmeldungen (**`2`**) und Informationsmeldungen (**`4`**) hoch.

So senden Sie benutzerdefinierte Protokolle (jede beliebige textbasierte Protokolldatei) an CloudWatch Logs

1. Suchen Sie in der JSON-Datei nach dem Abschnitt `CustomLogs`.

```
{
    "Id": "CustomLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\CustomLogs\\",
        "TimestampFormat": "MM/dd/yyyy HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "Local",
        "LineCount": "5"
    }
},
```

2. Geben Sie für `LogDirectoryPath` den Pfad ein, auf dem die Protokolle auf Ihrer Instance gespeichert werden sollen.
3. Geben Sie unter `TimestampFormat` das Zeitstempelformat ein, das Sie verwenden möchten. Weitere Informationen über unterstützte Werte finden Sie unter dem Thema Benutzerdefinierte Datums- und Zeitformat-Zeichenfolgen auf MSDN.

   Important

   Die Quell-Protokolldatei muss zu Beginn jeder Protokollzeile einen Zeitstempel haben, und nach dem Zeitstempel muss eine Leerstelle folgen.
4. Geben Sie für `Encoding` die zu verwendende Datei-Kodierung ein (z. B: UTF-8). Weitere Informationen über unterstützte Werte finden Sie unter dem Thema Encoding Class auf MSDN.

Note

Verwenden Sie den Kodierungsnamen, nicht den Anzeigenamen.

5. (Optional) Geben Sie für `Filter` das Präfix des Protokollnamens ein. Lassen Sie diesen Parameter leer, um alle Dateien zu überwachen. Weitere Informationen über unterstützte Werte finden Sie unter dem Thema FileSystemWatcherFilter-Eigenschaft auf MSDN.
6. (Optional) Geben Sie für `CultureName` das Gebietsschema ein, unter dem Zeitstempel protokolliert wird. Wenn `CultureName` leer ist, wird standardmäßig dasselbe Gebietsschema verwendet, das von der Windows-Instance verwendet wird. Weitere Informationen über unterstützte Werte finden Sie in der Spalte `Language tag` in der Tabelle im Thema Produktverhalten in MSDN.

   Note

   Die `div`, `div-MV`, `hu`, und `hu-HU` Werte werden nicht unterstützt.
7. (Optional) für `TimeZoneKind`, Typ `Local` oder `UTC`. Sie können dies einstellen, um Zeitzoneninformationen anzugeben, wenn keine Zeitzoneninformationen in den Zeitstempel Ihres Protokolls enthalten sind. Wenn dieser Parameter leer bleibt und Ihr Zeitstempel keine Zeitzoneninformationen enthält, verwendet CloudWatch Logs standardmäßig die lokale Zeitzone. Dieser Parameter wird ignoriert, wenn der Zeitstempel bereits Zeitzoneninformationen enthält.
8. (Optional) Geben Sie für `LineCount` die Anzahl der Zeilen im Header ein, um die Protokolldatei zu identifizieren. Beispielsweise haben IIS-Protokolldateien praktisch identische Header. Sie können **`5`** eingeben, dann würden die ersten drei Zeilen des Headers der Protokolldatei gelesen, um diese zu identifizieren. In den IIS-Protokolldateien ist die dritte Zeile das Datum und der Zeitstempel, aber es ist nicht immer sichergestellt, dass der Zeitstempel von zwei verschiedenen Protokolldateien unterschiedlich ist. Aus diesem Grund empfehlen wir, mindestens eine Zeile der tatsächlichen Protokolldaten hinzuzufügen, so dass die Protokolldatei eindeutig identifizierbar ist.

So senden Sie IIS-Protokolldaten an CloudWatch Logs

1. Suchen Sie in der JSON-Datei nach dem Abschnitt `IISLog`.

```
{
    "Id": "IISLogs",
    "FullName":
 "AWS.EC2.Windows.CloudWatch.CustomLog.CustomLogInputComponent,AWS.EC2.Windows.CloudWatch",
    "Parameters": {
        "LogDirectoryPath": "C:\\inetpub\\logs\\LogFiles\\W3SVC1",
        "TimestampFormat": "yyyy-MM-dd HH:mm:ss",
        "Encoding": "UTF-8",
        "Filter": "",
        "CultureName": "en-US",
        "TimeZoneKind": "UTC",
        "LineCount": "5"
    }
},
```

2. Geben Sie für `LogDirectoryPath` das Verzeichnis an, in dem die IIS-Protokolldateien für einen individuellen Standort gespeichert sind (z. B. `C:\inetpub\logs\LogFiles\W3SVC`*`n`*).

   Note

   Es wird nur das W3C-Protokollformat unterstützt. IIS, NCSA und benutzerdefinierte Formate werden nicht unterstützt.
3. Geben Sie unter `TimestampFormat` das Zeitstempelformat ein, das Sie verwenden möchten. Weitere Informationen über unterstützte Werte finden Sie unter dem Thema Benutzerdefinierte Datums- und Zeitformat-Zeichenfolgen auf MSDN.

4. Geben Sie für `Encoding` die zu verwendende Datei-Kodierung ein (z. B: UTF-8). Weitere Informationen über unterstützte Werte finden Sie unter dem Thema Encoding Class auf MSDN.

   Note

   Verwenden Sie den Kodierungsnamen, nicht den Anzeigenamen.
5. (Optional) Geben Sie für `Filter` das Präfix des Protokollnamens ein. Lassen Sie diesen Parameter leer, um alle Dateien zu überwachen. Weitere Informationen über unterstützte Werte finden Sie unter dem Thema FileSystemWatcherFilter-Eigenschaft auf MSDN.
6. (Optional) Geben Sie für `CultureName` das Gebietsschema ein, unter dem Zeitstempel protokolliert wird. Wenn `CultureName` leer ist, wird standardmäßig dasselbe Gebietsschema verwendet, das von der Windows-Instance verwendet wird. Weitere Informationen über unterstützte Werte finden Sie in der Spalte `Language tag` in der Tabelle im Thema Produktverhalten in MSDN.

   Note

   Die `div`, `div-MV`, `hu`, und `hu-HU` Werte werden nicht unterstützt.
7. (Optional) für `TimeZoneKind`, geben Sie `Local` oder `UTC`. Sie können dies einstellen, um Zeitzoneninformationen anzugeben, wenn keine Zeitzoneninformationen in den Zeitstempel Ihres Protokolls enthalten sind. Wenn dieser Parameter leer bleibt und Ihr Zeitstempel keine Zeitzoneninformationen enthält, verwendet CloudWatch Logs standardmäßig die lokale Zeitzone. Dieser Parameter wird ignoriert, wenn der Zeitstempel bereits Zeitzoneninformationen enthält.
8. (Optional) Geben Sie für `LineCount` die Anzahl der Zeilen im Header ein, um die Protokolldatei zu identifizieren. Beispielsweise haben IIS-Protokolldateien praktisch identische Header. Sie können **5** eingeben, dann würden die ersten fünf Zeilen des Headers der Protokolldatei gelesen, um diese zu identifizieren. In den IIS-Protokolldateien ist die dritte Zeile das Datum und der Zeitstempel, aber es ist nicht immer sichergestellt, dass der Zeitstempel von zwei verschiedenen Protokolldateien unterschiedlich ist. Aus diesem Grund empfehlen wir, mindestens eine Zeile der tatsächlichen Protokolldaten hinzuzufügen, sodass die Protokolldatei eindeutig identifizierbar ist.

Schritt 4. Flow Control konfigurieren

Jeder Datentyp muss ein entsprechendes Ziel im Bereich `Flows` haben. Um zum Beispiel ein kundendefiniertes Protokoll, ein ETW-Protokoll und ein Systemprotokoll an CloudWatch Logs zu senden, fügen Sie dem Bereich `Flows` Folgendes hinzu: `(CustomLogs,ETW,SystemEventLog),CloudWatchLogs`.

Warning

Wenn Sie einen ungültigen Schritt hinzufügen, ist der Fluss blockiert. Wenn Sie beispielsweise einen Metrikschritt für eine Festplatte hinzufügen, Ihre Instance aber keine Festplatte hat, sind alle Schritte im Fluss blockiert.

Sie können dieselbe Protokolldatei an mehrere Ziele senden. Wenn Sie beispielsweise das Anwendungsprotokoll an zwei unterschiedliche Ziele senden möchten, die Sie im Abschnitt `CloudWatchLogs` definiert haben, fügen Sie dem Abschnitt `Flows` Folgendes hinzu: `ApplicationEventLog,(CloudWatchLogs,CloudWatchLogs2)`.

So konfigurieren Sie die Flusssteuerung:

1. Suchen Sie in der Datei `AWS.EC2.Windows.CloudWatch.json` den Abschnitt `Flows`.

```
"Flows": {
    "Flows": [
      "PerformanceCounter,CloudWatch",
      "(PerformanceCounter,PerformanceCounter2), CloudWatch2",
      "(CustomLogs, ETW, SystemEventLog),CloudWatchLogs",
      "CustomLogs, CloudWatchLogs2",
      "ApplicationEventLog,(CloudWatchLogs, CloudWatchLogs2)"
```

```
    ]
}
```

2. Geben Sie für `Flows` jeden Datentyp ein, der hochgeladen werden soll (z. B. `ApplicationEventLog`) sowie sein Ziel (z. B. `CloudWatchLogs`).

Sie haben jetzt die Bearbeitung der JSON-Datei abgeschlossen. Sie werden sie in einem späteren Schritt verwenden.

Den Agenten starten

Um ein Amazon EC2 Instanz Ausführen von Windows Server 2012 oder Windows Server 2008 zum Senden von Protokollen an CloudWatch Logs, verwenden Sie den EC2Config-Service (`EC2Config.exe`). Ihre Instanz sollte EC2Config 4.0 oder höher haben und Sie können dieses Verfahren verwenden. Weitere Informationen über die Verwendung einer früheren Version von EC2Config finden Sie unter Verwendung von EC2Config 3.x oder früher zur Konfiguration von CloudWatch im Amazon EC2-Benutzerhandbuch für Windows-Instances

So konfigurieren Sie CloudWatch mithilfe von EC2Config 4.x

1. Prüfen Sie die Kodierung der Datei AWS.EC2.Windows.CloudWatch.json, die Sie zu einem früheren Zeitpunkt in dieser Anleitung bearbeitet haben. Es wird nur UTF-8 ohne BOM-Kodierung unterstützt. Speichern Sie die Datei in folgendem Verzeichnis auf Windows Server 2008 – 2012 R2-Instance: . `C:\Program Files\Amazon\SSM\Plugins\awsCloudWatch\`.
2. Start Sie den SSM-Agenten, oder starten Sie ihn neu (`AmazonSSMAgent.exe`) mithilfe der Windows Services-Systemsteuerung oder mit dem folgenden PowerShell-Befehl:

```
PS C:\> Restart-Service AmazonSSMAgent
```

Nachdem der SSM-Agent neu startet, erkennt er die Konfigurationsdatei und konfiguriert die Instance für die CloudWatch-Integration. Wenn Sie die Parameter und Einstellungen in der lokalen Konfigurationsdatei ändern, müssen Sie den SSM-Agenten neu starten, damit die Änderungen übernommen werden. Um die CloudWatch-Integration auf der Instance zu deaktivieren, ändern Sie `IsEnabled` zu `false`, und speichern Sie die Änderungen in der Konfigurationsdatei.

Schnellstart Installieren Sie die CloudWatch Logs Agent Using AWS OpsWorks und Chefkoch

Sie können den CloudWatch Logs-Agenten installieren und mithilfe von AWS OpsWorks und Chef – einem Automatisierungstool für Drittanbietersysteme und Cloud-Infrastrukturen – Protokollstreams erstellen. Chef verwendet sog. „Rezepte", die Sie schreiben, um Software auf Ihrem Computer zu installieren und zu konfigurieren, und sog. „Rezeptbücher", also Rezeptsammlungen, mit denen die Konfigurations- und Richtlinienverteilungsaufgaben des Tools ausgeführt werden. Weitere Informationen finden Sie unter Chef.

Die nachstehenden Beispiele für Chef-Rezepte zeigen, wie Sie eine Protokolldatei auf jeder EC2-Instance überwachen. Für die Rezepte wird der Stacknamen als Protokollgruppe und der Hostnamen der Instance als Protokollstreamnamen verwendet. Um mehrere Protokolldateien zu überwachen, müssen Sie die Rezepte erweitern, um mehrere Protokollgruppen und Protokollstreams zu erstellen.

Schritt 1 Benutzerdefinierte Rezepte erstellen

Erstellen Sie ein Repository, um Ihre Rezepte zu speichern. AWS OpsWorks unterstützt Git und Subversion, Sie können ein Archiv aber auch in Amazon S3 speichern. Die Struktur des Rezeptbuch-Repository ist unter Rezeptbuch-Repositorys im AWS OpsWorks User Guide beschrieben. Die nachfolgenden Beispiele gehen davon aus, dass das Kochbuch benannt wird `logs`. Das Rezept install.rb

installiert das CloudWatch Logs Agent. Sie können das Beispiel-Rezeptbuch auch herunterladen (CloudWatchLogs-Cookbooks.zip).

Erstellen Sie eine Datei mit dem Namen metadata.rb, die folgenden Code enthält:

```
#metadata.rb

name             'logs'
version          '0.0.1'
```

Erstellen Sie die CloudWatch Logs-Konfigurationsdatei.

```
#config.rb

template "/tmp/cwlogs.cfg" do
  cookbook "logs"
  source "cwlogs.cfg.erb"
  owner "root"
  group "root"
  mode 0644
end
```

Herunterladen und Installieren des CloudWatch Logs-Agenten:

```
# install.rb

directory "/opt/aws/cloudwatch" do
  recursive true
end

remote_file "/opt/aws/cloudwatch/awslogs-agent-setup.py" do
  source "https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py"
  mode "0755"
end

  execute "Install CloudWatch Logs agent" do
  command "/opt/aws/cloudwatch/awslogs-agent-setup.py -n -r region -c /tmp/cwlogs.cfg"
  not_if { system "pgrep -f aws-logs-agent-setup" }
end
```

> **Note**
>
> Im obigen Beispiel ersetzen Sie *region* mit einer der folgenden Punkte: us-east-1, us-west-1, us-west-2, ap-south-1, ap-northeast-2, ap-southeast-1, ap-southeast-2, ap-northeast-1, eu-central-1, eu-west-1, or sa-east-1.

Wenn die Installation des Agenten fehlschlägt, prüfen Sie, ob das Paket `python-dev` installiert ist. Wenn dies nicht der Fall ist, verwenden Sie den folgenden Befehl und wiederholen Sie die Agenteninstallation:

```
sudo apt-get -y install python-dev
```

Dieses Rezept verwendet eine Vorlagendatei namens cwlogs.cfg.erb, die Sie ändern können, um verschiedene Attribute festzulegen, wie z. B. welche Dateien protokolliert werden sollen. Weitere Informationen zu diesen Attributen finden Sie unter Referenz für den CloudWatch Logs-Agenten (p. 152).

```
[general]
# Path to the AWSLogs agent's state file. Agent uses this file to maintain
# client side state across its executions.
state_file = /var/awslogs/state/agent-state
```

```
## Each log file is defined in its own section. The section name doesn't
## matter as long as its unique within this file.
#
#[kern.log]
#
## Path of log file for the agent to monitor and upload.
#
#file = /var/log/kern.log
#
## Name of the destination log group.
#
#log_group_name = kern.log
#
## Name of the destination log stream.
#
#log_stream_name = {instance_id}
#
## Format specifier for timestamp parsing.
#
#datetime_format = %b %d %H:%M:%S
#
#

[<%= node[:opsworks][:stack][:name] %>]
datetime_format = [%Y-%m-%d %H:%M:%S]
log_group_name = <%= node[:opsworks][:stack][:name].gsub(' ','_') %>
file = <%= node[:cwlogs][:logfile] %>
log_stream_name = <%= node[:opsworks][:instance][:hostname] %>
```

Die Vorlage erhält den Stack-Namen und den Host-Namen, indem auf die entsprechenden Attribute in der Stack-Konfiguration und Bereitstellungs-JSON verwiesen wird. Das Attribut, das die Datei zu protokollierende Datei festlegt, ist in der Attributdatei des cwlogs-Rezeptbuchs „default.rb” definiert (logs/attributes/default.rb).

```
default[:cwlogs][:logfile] = '/var/log/aws/opsworks/opsworks-agent.statistics.log'
```

Schritt 2 Erstellen AWS OpsWorks Stapel

1. Öffnen Sie die AWS OpsWorks-Konsole unter https://console.aws.amazon.com/opsworks.
2. Wählen Sie auf dem OpsWorks Dashboard die Option Add stack (Stack hinzufügen), um einen AWS OpsWorks-Stack zu erstellen.
3. Wählen Sie auf dem Bildschirm Add stack die Option Chef 11 stack aus.
4. Geben Sie für Stack name einen Namen ein.
5. Wählen Sie bei Use custom Chef Cookbooks die Option Yes.
6. Wählen Sie für Repository type den Repository-Typ, den Sie verwenden. Wenn Sie das oben genannte Beispiel verwenden, wählen Sie Http Archive.
7. Geben Sie für Repository URL das Repository ein, in dem Sie das Rezeptbuch gespeichert haben, das Sie im vorherigen Schritt erstellt haben. Wenn Sie das oben genannte Beispiel verwenden, geben Sie **`https://s3.amazonaws.com/aws-cloudwatch/downloads/CloudWatchLogs-Cookbooks.zip`** ein.
8. Wählen Sie Add Stack aus, um den Stack zu erstellen.

Schritt 3 Verlängern Sie Ihre IAM Rolle

Um CloudWatch Logs mit Ihren AWS OpsWorks-Instances zu verwenden, müssen Sie die von Ihren Instances verwendeten IAM-Rollen erweitern.

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich Policies und Create Policy aus.
3. Wählen Sie auf der Seite Create Policy unter Create Your Own Policy die Option Select aus. Weitere Informationen zum Erstellen von benutzerdefinierten Richtlinien finden Sie unter IAM-Richtlinien für Amazon EC2 im Amazon EC2-Benutzerhandbuch für Linux-Instances.
4. Geben Sie auf der Seite Review Policy im Feld Policy Name einen Namen für die Richtlinie ein.
5. Fügen Sie die folgende Richtlinie unter Policy Document ein:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
    "Resource": [
      "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

6. Wählen Sie Create Policy (Richtlinie erstellen) aus.
7. Wählen Sie im Navigationsbereich Roles (Rollen) und im Inhaltsbereich für Role Name (Rollenname) den Namen der von Ihrem AWS OpsWorks-Stack verwendeten Instance-Rolle. Sie finden den von Ihrem Stack verwendeten Namen in den Stack-Einstellungen (der Standardnamen lautet `aws-opsworks-ec2-role`).

   Note

   Wählen Sie den Rollennamen, aktivieren Sie nicht das Kontrollkästchen.
8. Wählen Sie auf der Registerkarte Permissions unter Managed Policies die Option Attach Policy aus.
9. Wählen Sie auf der Seite Attach Policy im Tabellen-Header (neben Filter und Search) die Option Policy Type und Customer Managed Policies aus.
10. Wählen Sie für Customer Managed Policies (Vom Kunden verwaltete Richtlinien) die IAM-Richtlinie, die Sie zuvor erstellt haben, und wählen Sie Attach Policy (Richtlinie anfügen) aus.

    Weitere Informationen über IAM-Benutzer und -Richtlinien finden Sie unter IAM-Benutzer und -Gruppen und Verwalten von IAM-Richtlinien im IAM-Benutzerhandbuch.

Schritt 4. Layer hinzufügen

1. Öffnen Sie die AWS OpsWorks-Konsole unter https://console.aws.amazon.com/opsworks.
2. Wählen Sie im Navigationsbereich Layers aus.
3. Wählen Sie im Inhaltsbereich einen Layer aus und dann Add layer.
4. Wählen Sie auf der Registerkarte OpsWorks für Layer type die Option Custom aus.
5. Geben Sie für die Felder Name und Short name den Lang- und Kurznamen des Layers ein und wählen Sie dann Add layer aus.
6. Auf der Registerkarte Recipes (Rezepte) gibt es unter Custom Chef Recipes (Benutzerdefinierte Chef-Rezepte) mehrere Überschriften – Setup (Einrichtung), Configure (Konfigurieren), Deploy (Bereitstellen), Undeploy (Bereitstellung aufheben) und Shutdown (Herunterfahren). Dies entspricht

AWS OpsWorks-Lebenszyklusereignissen. AWS OpsWorks löst diese Ereignisse an diesen Schlüsselpunkten im Lebenszyklus einer Instance aus, die die zugeordneten Rezepte ausführt.

Note

Wenn die oben genannten Überschriften nicht sichtbar sind, wählen Sie unter Custom Chef Recipes die Option edit.

7. Geben Sie neben Setup den Text logs::config, logs::install ein und wählen Sie +, um ihn der Liste hinzuzufügen, wählen Sie anschließend Save.

   AWS OpsWorks führt dieses Rezept direkt nach dem Start der Instanz auf jeder der neuen Instances in diesem Layer aus.

Schritt 5. Instanz hinzufügen

Mit dem Layer wird nur gesteuert, wie Instances konfiguriert werden. Sie müssen nun dem Layer einige Instances hinzufügen und diese starten.

1. Öffnen Sie die AWS OpsWorks-Konsole unter https://console.aws.amazon.com/opsworks.
2. Wählen Sie im Navigationsbereich Instances aus und dann unter Ihrem Layer die Option + Instance.
3. Akzeptieren Sie die Standardeinstellungen und wählen Sie Add Instance, um dem Layer die Instance hinzuzufügen.
4. Klicken Sie in der Spalte Actions auf start, um die Instance zu starten.

   AWS OpsWorks startet eine neue EC2-Instance und konfiguriert CloudWatch Logs. Der Status der Instance ändert sich zu „online”, sobald sie bereit ist.

Schritt 6. Ihre Protokolle anzeigen

Ihnen sollten in der CloudWatch-Konsole einen kurzen Moment nach dem Ausführen des Agenten die neu erstellte Protokollgruppe und der Protokoll-Stream angezeigt werden.

Weitere Informationen finden Sie im An CloudWatch Logs gesendete Protokolldaten anzeigen lassen (p. 61).

Den Status des CloudWatch Logs-Agenten melden

Führen Sie die folgenden Schritte aus, um den Status des CloudWatch Logs-Agenten auf Ihrer EC2-Instance zu melden.

So melden Sie den Status des Agenten

1. Stellen Sie eine Verbindung mit Ihrer EC2-Instance her: Weitere Informationen finden Sie unter Herstellen einer Verbindung mit Ihrer Instance im Amazon EC2-Benutzerhandbuch für Linux-Instances.

   Weitere Informationen über Verbindungsprobleme finden Sie unter Beheben von Verbindungsproblemen mit Ihrer Instance im Amazon EC2-Benutzerhandbuch für Linux-Instances
2. Geben Sie als Eingabeaufforderung den folgenden Befehl ein:

```
sudo service awslogs status
```

   Wenn Sie Amazon Linux 2 ausführen, geben Sie den folgenden Befehl ein:

```
sudo service awslogsd status
```

3. Prüfen Sie die Datei /var/log/awslogs.log auf Fehler, Warnungen oder Probleme hinsichtlich des CloudWatch Logs-Agenten.

Den CloudWatch Logs-Agenten starten

Wenn der CloudWatch Logs-Agent auf Ihrer EC2-Instance nach der Installation nicht automatisch gestartet ist oder Sie den Agenten beendet haben, können Sie das folgende Verfahren anwenden, um den Agenten zu starten.

So starten Sie den -Agenten:

1. Stellen Sie eine Verbindung mit Ihrer EC2-Instance her: Weitere Informationen finden Sie unter Herstellen einer Verbindung mit Ihrer Instance im Amazon EC2-Benutzerhandbuch für Linux-Instances.

   Weitere Informationen über Verbindungsprobleme finden Sie unter Beheben von Verbindungsproblemen mit Ihrer Instance im Amazon EC2-Benutzerhandbuch für Linux-Instances.
2. Geben Sie als Eingabeaufforderung den folgenden Befehl ein:

   ```
   sudo service awslogs start
   ```

   Wenn Sie Amazon Linux 2 ausführen, geben Sie den folgenden Befehl ein:

   ```
   sudo service awslogsd start
   ```

Beenden des CloudWatch Logs-Agenten

Führen Sie die folgenden Schritte aus, um den CloudWatch Logs-Agenten auf Ihrer EC2-Instance zu beenden.

So beenden Sie den Agenten

1. Stellen Sie eine Verbindung mit Ihrer EC2-Instance her: Weitere Informationen finden Sie unter Herstellen einer Verbindung mit Ihrer Instance im Amazon EC2-Benutzerhandbuch für Linux-Instances.

   Weitere Informationen über Verbindungsprobleme finden Sie unter Beheben von Verbindungsproblemen mit Ihrer Instance im Amazon EC2-Benutzerhandbuch für Linux-Instances.
2. Geben Sie als Eingabeaufforderung den folgenden Befehl ein:

   ```
   sudo service awslogs stop
   ```

   Wenn Sie Amazon Linux 2 ausführen, geben Sie den folgenden Befehl ein:

   ```
   sudo service awslogsd stop
   ```

Schnellstart Verwenden AWS CloudFormation damit beginnen mit CloudWatch Logs

Mit AWS CloudFormation können Sie Ihre AWS-Ressourcen im JSON-Format beschreiben und bereitstellen. Zu den Vorteilen dieser Methode gehört die Möglichkeit, eine Sammlung von AWS-

Ressourcen als eine Einheit zu verwalten und Ihre AWS-Ressourcen in verschiedenen Regionen bequem replizieren zu können.

Wenn Sie AWS mithilfe von AWS CloudFormation bereitstellen, erstellen Sie Vorlagen, die die zu verwendenden AWS-Ressourcen beschreiben. Das folgende Beispiel ist ein Vorlagenausschnitt, mit dem eine Protokollgruppe und ein Metrikfilter erstellt werden, der 404 Vorkommnisse zählt und diese Zählung an die Protokollgruppe sendet.

```
"WebServerLogGroup": {
    "Type": "AWS::Logs::LogGroup",
    "Properties": {
        "RetentionInDays": 7
    }
},

"404MetricFilter": {
    "Type": "AWS::Logs::MetricFilter",
    "Properties": {
        "LogGroupName": {
            "Ref": "WebServerLogGroup"
        },
        "FilterPattern": "[ip, identity, user_id, timestamp, request, status_code = 404,
 size, ...]",
        "MetricTransformations": [
            {
                "MetricValue": "1",
                "MetricNamespace": "test/404s",
                "MetricName": "test404Count"
            }
        ]
    }
}
```

Dies ist ein einfaches Beispiel. Sie können mithilfe von CloudWatch Logs viel umfassendere AWS CloudFormation-Bereitstellungen einrichten. Weitere Informationen über Vorlagenbeispiele finden Sie unter Amazon CloudWatch Logs-Vorlagenausschnitte im AWS CloudFormation Benutzerhandbuch. Weitere Informationen finden Sie unter Erste Schritte mit AWS CloudFormation im AWS CloudFormation Benutzerhandbuch.

Analysieren von Protokolldaten mit CloudWatch Logs Insights

Mit CloudWatch Logs Insights können Sie Ihre Protokolldaten interaktiv durchsuchen und in Amazon CloudWatch Logs analysieren. Sie können Abfragen durchführen, die Ihnen helfen, schnell und effektiv auf betriebliche Probleme zu reagieren. Wenn ein Problem auftritt, können Sie mit CloudWatch Logs Insights potenzielle Ursachen identifizieren und bereitgestellte Lösungen validieren.

CloudWatch Logs Insights enthält eine speziell entwickelte Abfragesprache mit ein paar einfachen, aber mächtigen Befehlen. CloudWatch Logs Insights bietet Beispielabfragen, Befehlsbeschreibungen, automatische Vervollständigung von Abfragen und Erkennung von Protokollfeldern, damit Sie schnell loslegen können. Beispielabfragen sind für verschiedene Arten von AWS-Serviceprotokollen enthalten.

CloudWatch Logs Insights erkennt automatisch Felder in Protokollen von AWS-Services wie Amazon Route 53, AWS Lambda, AWS CloudTrailund Amazon VPC sowie in jeder Anwendung und jedem benutzerdefinierten Protokoll, das Protokollereignisse als JSON aussendet.

Sie können mit CloudWatch Logs Insights nach Protokolldaten suchen, die am 5. November 2018 oder später an CloudWatch Logs gesendet wurden.

Eine einzelne Anforderung kann bis zu 20 Protokollgruppen abfragen. Abfragen werden nach 15 Minuten beendet, wenn sie nicht abgeschlossen sind. Abfrageergebnisse sind 7 Tage lang verfügbar.

Sie können Abfragen speichern, die Sie erstellt haben. Dies kann Ihnen helfen, bei Bedarf komplexe Abfragen auszuführen, ohne sie jedes Mal neu erstellen zu müssen, wenn Sie diese ausführen möchten.

> **Important**
>
> Wenn Ihr Netzwerksicherheitsteam die Verwendung von Web-Sockets nicht zulässt, können Sie zurzeit nicht auf den CloudWatch Logs Insights-Abschnitt der CloudWatch-Konsole zugreifen. Sie können die CloudWatch Logs Insights-Abfragemöglichkeiten über APIs nutzen. Weitere Informationen finden Sie unter StartQuery im Amazon CloudWatch Logs API Reference.

Inhalt:

- Unterstützte Protokolle und erkannte Felder (p. 36)
- Praktische Anleitung Ausführen und Ändern einer Beispielabfrage (p. 39)
- Praktische Anleitung Ausführen einer Abfrage mit einer Aggregationsfunktion (p. 41)
- Praktische Anleitung Ausführen einer Abfrage, die eine Visualisierung erzeugt, die nach Protokollfeldern gruppiert ist (p. 41)
- Praktische Anleitung Ausführen einer Abfrage, die eine Zeitreihenvisualisierung erzeugt (p. 42)
- CloudWatch Logs Insights-Abfragesyntax (p. 42)
- Visualisieren von Protokolldaten in Diagrammen (p. 53)
- Speichern und erneutes Ausführen von CloudWatch Logs Insights-Abfragen (p. 55)
- Beispielabfragen (p. 56)
- Abfrage zum Dashboard hinzufügen oder Abfrageergebnisse exportieren (p. 59)
- Anzeigen von laufenden Abfragen oder Abfrageverlauf (p. 60)

Unterstützte Protokolle und erkannte Felder

CloudWatch Logs Insights unterstützt alle Arten von Protokollen. Für jedes Protokoll, das an CloudWatch Logs gesendet wird, werden fünf Systemfelder automatisch generiert:

- `@message` enthält das rohe, unverarbeitete Protokollereignis. Dies entspricht dem `message`-Feld in InputLogEvent.
- `@timestamp` enthält den Ereignis-Zeitstempel, der im `timestamp`-Feld des Protokollereignisses enthalten ist. Dies entspricht dem `timestamp`-Feld in InputLogEvent.
- `@ingestionTime` enthält den Zeitpunkt, zu dem das Protokollereignis von CloudWatch Logs empfangen wurde.
- `@logStream` enthält den Namen des Protokollstreams, zu dem das Protokollereignis hinzugefügt wurde. Protokolldatenströme werden verwendet, um Protokolle nach demselben Prozess zu gruppieren, der sie generiert hat.
- `@log` ist eine Protokollgruppenkennung in Form von *`account-id`*`:`*`log-group-name`*. (z. B.. Dies kann in Abfragen mehrerer Protokollgruppen nützlich sein, um zu identifizieren, zu welcher Protokollgruppe ein bestimmtes Ereignis gehört.

CloudWatch Logs Insights fügt das @-Zeichen am Anfang der Felder, die davon generiert werden, ein.

Für viele Protokolltypen erkennt CloudWatch Logs auch automatisch die in den Protokollen enthaltenen Protokollfelder. Diese automatischen Suchfelder sind in der folgenden Tabelle dargestellt.

Für andere Arten von Protokollen mit Feldern, die CloudWatch Logs Insights nicht automatisch erkennt, können Sie den Befehl **`parse`** verwenden, um ephemere Felder für die Verwendung in dieser Abfrage zu extrahieren und zu erstellen. Weitere Informationen finden Sie unter CloudWatch Logs Insights-Abfragesyntax (p. 42).

Wenn der Name eines gefundenen Protokollfeldes mit dem Zeichen `@` beginnt, zeigt CloudWatch Logs Insights es mit einem zusätzlichen `@` an den Anfang angehängt an. Wenn z. B. ein Protokollfeldname `@example.com`lautet, wird dieser Feldname als `@@example.com` angezeigt.

Protokolltyp	Erkannte Protokollfelder
Amazon VPC-Flussprotokolle	`@timestamp`, `@logStream`, `@message`, `accountId`, `endTime`, `interfaceId`, `logStatus`, `startTime`, `version`, `action`, `bytes`, `dstAddr`, `dstPort`, `packets`, `protocol`, `srcAddr`, `srcPort`
Route 53-Protokolle	`@timestamp`, `@logStream`, `@message`, `edgeLocation`, `hostZoneId`, `protocol`, `queryName`, `queryTimestamp`, `queryType`, `resolverIp`, `responseCode`, `version`
Lambda-Protokolle	`@timestamp`, `@logStream`, `@message`, `@requestId`, `@duration`, `@billedDuration`, `@type`, `@maxMemoryUsed`, `@memorySize` Wenn ein Lambda Protokollzeile enthält eine X-Ray Ablaufverfolgungs-ID enthalten, enthält sie auch die folgenden Felder: `@xrayTraceId` und `@xraySegmentId`. CloudWatch Logs Insights erkennt automatisch Protokollfelder in Lambda-Protokollen, aber nur für das erste eingebettete JSON-Fragment in jedem Protokollereignis. Wenn ein Lambda-Protokollereignis mehrere JSON-Fragmente enthält, können Sie die Protokollfelder mit dem **`parse`**-Befehl analysieren und extrahieren. Weitere Informationen finden Sie im Felder in JSON-Protokollen (p. 38).
CloudTrail-Protokolle Protokolle im JSON-Format	Weitere Informationen finden Sie im Felder in JSON-Protokollen (p. 38).

Protokolltyp	Erkannte Protokollfelder
Andere Protokolltypen	`@timestamp`, `@ingestionTime`, `@logStream`, `@message`, `@log`.

Felder in JSON-Protokollen

CloudWatch Logs Insights repräsentiert verschachtelte JSON-Felder mit der Punktnotation. Im folgenden Beispiel für ein JSON-Ereignis wird das Feld `type` im JSON-Objekt `userIdentity` als `userIdentity.type` dargestellt.

JSON-Arrays werden zu einer Liste von Feldnamen und Werten zusammengefasst. Um beispielsweise den Wert von `instanceId` für das erste Element im `requestParameters.instancesSet` anzugeben, verwenden Sie `requestParameters.instancesSet.items.0.instanceId`.

CloudWatch Logs Insights kann maximal 100 Protokollereignisfelder aus einem JSON-Protokoll extrahieren. Zusätzliche Felder, die nicht extrahiert werden, können Sie mit dem Befehl parse aus dem nicht analysierten Protokollereignis im Nachrichtenfeld analysieren.

```
{ "eventVersion": "1.0",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn: aws: iam: : 123456789012: user/Alice",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "accountId": "123456789012",
        "userName": "Alice"
    },
    "eventTime": "2014-03-06T21: 22: 54Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "StartInstances",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "205.251.233.176",
    "userAgent": "ec2-api-tools1.6.12.2",
    "requestParameters": {
        "instancesSet": {
            "items": [
                {
                    "instanceId": "i-abcde123"
                }
            ]
        }
    },
    "responseElements": {
        "instancesSet": {
            "items": [
                {
                    "instanceId": "i-abcde123",
                    "currentState": {
                        "code": 0,
                        "name": "pending"
                    },
                    "previousState": {
                        "code": 80,
                        "name": "stopped"
                    }
                }
            ]
        }
    }
```

Praktische Anleitung Ausführen und Ändern einer Beispielabfrage

Das folgende Tutorial hilft Ihnen bei den ersten Schritten mit CloudWatch Logs Insights. Sie führen eine Beispielabfrage aus und sehen dann, wie Sie sie ändern und erneut ausführen können.

Um eine Abfrage auszuführen, müssen Sie bereits über Protokolle verfügen, die in CloudWatch Logs gespeichert sind. Wenn Sie bereits CloudWatch Logs verwenden und Protokollgruppen und Protokollstreams eingerichtet haben, können Sie loslegen. Sie können auch bereits Protokolle haben, wenn Sie Services wie z. B. AWS CloudTrail, Amazon Route 53oder Amazon VPC nutzen und die Übermittlung von Protokollen von diesen Services nach CloudWatch Logs eingerichtet haben. Weitere Informationen zum Senden von Protokollen an CloudWatch Logs finden Sie unter Erste Schritte mit CloudWatch Logs (p. 5).

Abfragen in CloudWatch Logs Insights geben entweder ein Set an Feldern aus Protokollereignissen zurück oder das Ergebnis einer mathematischen Aggregation oder einer anderen Operation, die bei Protokollereignissen durchgeführt wurde. Dieses Tutorial demonstriert eine Abfrage, die eine Liste von Protokollereignissen zurückgibt.

Ausführen einer Beispielabfrage

Beginnen Sie mit der Ausführung einer Beispielabfrage.

So führen Sie eine CloudWatch Logs Insights-Beispielabfrage durch

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Insights aus.

   Oben auf der Seite befindet sich der Abfrage-Editor. Wenn Sie CloudWatch Logs Insights zum ersten Mal öffnen, enthält dieses Feld eine Standardabfrage, die die 20 letzten Protokollereignisse zurückgibt.
3. Wählen Sie eine oder mehrere abzufragende Protokollgruppen oberhalb des Abfrage-Editors aus. Um Ihre Protokollgruppen zu finden, können Sie Text in die Suchleiste eingeben. Anschließend zeigt CloudWatch Logs auf der Suchleiste übereinstimmende Protokollgruppen an.

   Wenn Sie eine Protokollgruppe auswählen, erkennt CloudWatch Logs Insights automatisch Felder in den Daten in der Protokollgruppe. Um diese erkannten Felder anzuzeigen, wählen Sie rechts auf der Seite Fields (Felder) aus.
4. (Optional) Verwenden Sie die Zeitauswahl oben rechts, um den Zeitraum auszuwählen, den Sie abfragen möchten.
5. Wählen Sie Run aus.

   Die Ergebnisse der Abfrage werden angezeigt. In diesem Beispiel sind die Ergebnisse die letzten 20 Protokollereignisse aller Art.

   CloudWatch Logs zeigt außerdem ein Balkendiagramm der Protokollereignisse in dieser Protokollgruppe im Zeitverlauf an. Dieses Balkendiagramm zeigt die Verteilung der Ereignisse in der Protokollgruppe, die Ihrer Abfrage und Ihrem Zeitraum entspricht, nicht nur die in der Tabelle angezeigten Ereignisse.
6. Um alle Felder eines der zurückgegebenen Protokollereignisse anzuzeigen, wählen Sie das Symbol links neben diesem Protokollereignis aus.

Ändern der Beispielabfrage

In diesem Tutorial ändern Sie die Beispielabfrage, um die 50 neuesten Protokollereignisse anzuzeigen.

Wenn Sie das vorherige Tutorial noch nicht ausgeführt haben, tun Sie das jetzt. Dieses Tutorial beginnt dort, wo das vorherige Tutorial endet.

> Note
>
> Einige Beispielabfragen, die mit bereitgestellt werden CloudWatch Logs Insights verwenden `head` oder `tail` Befehle anstelle von `limit`. (z. B.. Diese Befehle sind veraltet und wurden ersetzt durch `limit`. (z. B.. Verwenden `limit` anstatt der `head` oder `tail` in allen Abfragen, die Sie schreiben.

So ändern Sie die CloudWatch Logs Insights-Beispielabfrage

1. Ändern Sie im Abfrage-Editor 20 in 50. Wählen Sie anschließend Run (Ausführen) aus.

   Die Ergebnisse der neuen Abfrage werden angezeigt. Unter der Annahme, dass in der Protokollgruppe im Standardzeitraum genügend Daten vorhanden sind, werden nun 50 Protokollereignisse aufgelistet.
2. (Optional) Sie können Abfragen speichern, die Sie erstellt haben. Um diese Abfrage zu speichern, wählen Sie Save (Speichern) aus. Weitere Informationen finden Sie im Speichern und erneutes Ausführen von CloudWatch Logs Insights-Abfragen (p. 55).

Hinzufügen eines Filterbefehls zur Beispielabfrage

Dieses Tutorial zeigt, wie Sie eine umfangreichere Änderung der Abfrage im Abfrageeditor vornehmen können. In diesem Tutorial filtern Sie die Ergebnisse der vorherigen Abfrage basierend auf einem Feld in den abgerufenen Protokollereignissen.

Wenn Sie die vorherigen Tutorials noch nicht ausgeführt haben, tun Sie das jetzt. Dieses Tutorial beginnt dort, wo das vorherige Tutorial endet.

So fügen Sie der vorherigen Abfrage einen Filterbefehl hinzu

1. Entscheiden Sie sich für ein zu filterndes Feld. Zur Anzeige der häufigsten Felder, die CloudWatch Logs in den Protokollereignissen in den ausgewählten Protokollgruppen in den letzten 15 Minuten erkannt hat, und des Prozentsatzes der Protokollereignisse, in denen die einzelnen Feld angezeigt werden, wählen Sie rechts auf der Seite Fields (Felder) aus.

   Um die in einem bestimmten Protokollereignis enthaltenen Felder anzuzeigen, wählen Sie das Symbol links neben dieser Zeile aus.

   Möglicherweise wird das Feld awsRegion im Protokollereignis angezeigt. Dies ist von den Ereignissen abhängig, die in Ihren Protokollen enthalten sind. Im Rest dieses Tutorials verwenden Sie awsRegion als Filterfeld. Sie können jedoch auch ein anderes Feld verwenden, wenn dieses Feld nicht verfügbar ist.
2. Positionieren Sie den Cursor im Feld des Abfrageeditors hinter 50, und drücken Sie die Eingabetaste.
3. Geben Sie in der neuen Zeile zunächst | (das Pipe-Zeichen) und ein Leerzeichen ein. Befehle in einer CloudWatch Logs Insights-Abfrage müssen durch das Pipe-Zeichen getrennt werden.
4. Geben Sie **`filter awsRegion="us-east-1"`** ein.
5. Wählen Sie Run aus.

   Die Abfrage läuft erneut und zeigt nun die 50 neuesten Ergebnisse an, die dem neuen Filter entsprechen.

   Wenn Sie nach einem anderen Feld gefiltert haben und ein Fehlerergebnis erhalten haben, müssen Sie möglicherweise den Feldnamen maskieren. Wenn der Feldname auch andere als alphanumerische Zeichen enthält, müssen Sie vor und nach dem Feldnamen Backtick-Zeichen (`) verwenden (z. B. **`` `error-code`="102" ``**).

Sie müssen die Backtick-Zeichen für Feldnamen verwenden, die auch andere als alphanumerische Zeichen enthalten, nicht jedoch für Werte. Werte sind stets von Anführungszeichen (") umschlossen.

CloudWatch Logs Insights enthält leistungsstarke Abfragefunktionen, einschließlich mehrerer Befehle und Unterstützung für reguläre Ausdrücke, mathematische und statistische Operationen. Weitere Informationen finden Sie im CloudWatch Logs Insights-Abfragesyntax (p. 42).

Praktische Anleitung Ausführen einer Abfrage mit einer Aggregationsfunktion

In diesem Tutorial führen Sie eine Abfrage aus, die die Ergebnisse der Ausführung von Aggregationsfunktionen auf Protokolldatensätzen zurückgibt.

So führen Sie eine Aggregationsabfrage aus

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Insights aus.
3. Wählen Sie eine oder mehrere Protokollgruppen oberhalb des Abfrage-Editors aus. Um Ihre Protokollgruppen zu finden, geben Sie Text in die Suchleiste ein. Anschließend zeigt CloudWatch Logs übereinstimmende Protokollgruppen auf der Suchleiste an.
4. Löschen Sie im Abfrage-Editor die aktuell angezeigte Abfrage und geben Sie Folgendes ein. Wählen Sie anschließend Run (Ausführen) aus. Ersetzen Sie fieldname durch den Namen eines Felds, das im Bereich Fields (Felder) rechts auf der Seite angezeigt wird.

```
stats count(*) by fieldname
```

   Die Ergebnisse zeigen die Anzahl der Protokollereignisse in der Protokollgruppe an, die von CloudWatch Logs empfangen wurden und die jeden anderen Wert für den von Ihnen gewählten Feldnamen enthalten.

Praktische Anleitung Ausführen einer Abfrage, die eine Visualisierung erzeugt, die nach Protokollfeldern gruppiert ist

Wenn Sie eine Abfrage ausführen, die `stats` um die zurückgegebenen Ergebnisse nach den Werten eines oder mehrerer Felder in den Protokolleinträgen zu gruppieren, können Sie die Ergebnisse als Balkendiagramm, Kreisdiagramm, Liniendiagramm oder gestapeltes Flächendiagramm anzeigen. Auf diese Weise können Sie Trends in Ihren Protokollen effizienter visualisieren.

So führen Sie eine Abfrage zur Visualisierung durch

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Insights aus.
3. Wählen Sie eine oder mehrere abzufragende Protokollgruppen aus.
4. Löschen Sie im Abfrageeditor den aktuellen Inhalt, geben Sie dann die folgende `stats`-Funktion ein und klicken Sie auf Abfrage ausführen.

```
stats count(*) by @logStream
    | limit 100
```

Die Ergebnisse zeigen für jeden Protokolldatenstrom die Anzahl der Protokollereignisse in der Protokollgruppe. Die Ergebnisse sind auf nur 100 Zeilen begrenzt.

5. Wählen Sie die Registerkarte Visualization (Visualisierung) aus.
6. Wählen Sie den Pfeil neben Line (Zeile) aus. Wählen Sie anschließend Bar (Balken) aus.

   Das Balkendiagramm wird angezeigt, wobei jeder Protokolldatenstrom in der Protokollgruppe von einem Balken dargestellt wird.

Praktische Anleitung Ausführen einer Abfrage, die eine Zeitreihenvisualisierung erzeugt

Wenn Sie eine Abfrage ausführen, die `bin()` um die zurückgegebenen Ergebnisse nach einem Zeitraum zu gruppieren, können Sie die Ergebnisse als Liniendiagramm, gestapeltes Flächendiagramm, Kreisdiagramm oder Balkendiagramm anzeigen. Dies hilft Ihnen, Trends in Protokollereignissen im Laufe der Zeit effizienter zu visualisieren.

So führen Sie eine Abfrage zur Visualisierung durch

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Insights aus.
3. Wählen Sie eine oder mehrere abzufragende Protokollgruppen aus.
4. Löschen Sie im Abfrageeditor den aktuellen Inhalt, geben Sie dann die folgende `stats`-Funktion ein und klicken Sie auf Abfrage ausführen.

```
stats count(*) by bin(30s)
```

Die Ergebnisse zeigen die Anzahl der Protokollereignisse in der Protokollgruppe, die von CloudWatch Logs empfangen wurden, für jeden 30-Sekunden-Zeitraum an.

5. Wählen Sie die Registerkarte Visualization (Visualisierung) aus.

   Die Ergebnisse werden als Liniendiagramm dargestellt. Um zu einem Balkendiagramm, Kreisdiagramm oder gestapelten Flächendiagramm zu wechseln, wählen Sie den Pfeil neben Leitung oben links im Diagramm.

CloudWatch Logs Insights-Abfragesyntax

CloudWatch Logs Insights unterstützt eine Abfragesprache, mit der Sie Abfragen für Ihre Protokollgruppen durchführen können. Jede Abfrage kann einen oder mehrere Abfragebefehle beinhalten, die durch Pipe-Zeichen im Unix-Stil getrennt sind (`|`).

Es werden 6 Abfragebefehle sowie viele unterstützende Funktionen und Operationen unterstützt, darunter reguläre Ausdrücke, arithmetische Operationen, Vergleichsoperationen, numerische Funktionen, Datumszeitfunktionen, Zeichenkettenfunktionen und allgemeine Funktionen.

Kommentare werden ebenfalls unterstützt. Zeilen in einer Abfrage, die mit dem Zeichen # beginnen, werden ignoriert.

Felder, die mit dem @ Symbol werden generiert von CloudWatch Logs Erkenntnisse. Weitere Informationen zu den Feldern, die CloudWatch Logs erkennt automatisch und generiert, siehe Unterstützte Protokolle und erkannte Felder (p. 36).

CloudWatch Logs Insights-Abfragebefehle

Die folgende Tabelle listet die 6 unterstützten Abfragebefehle zusammen mit grundlegenden Beispielen auf. Umfangreichere Beispielabfragen finden Sie unter Beispielabfragen (p. 56).

Befehl	Description (Beschreibung)	Beispiele
`display`	Gibt an, welche Felder in den Abfrageergebnissen angezeigt werden sollen. Wenn Sie diesen Befehl mehrmals in der Abfrage angeben, werden nur die Felder verwendet, die Sie im letzten Vorkommen angeben.	Im folgenden Beispiel wird das Feld verwendet `@message` und erstellt die flüchtigen Felder `loggingType` und `loggingMessage` zur Verwendung in der Abfrage. Es filtert die Ereignisse nur nach denen mit `ERROR` als Wert von `loggingType`, aber zeigt dann nur die `loggingMessage` dieser Ereignisse in den Ergebnissen. `fields @message	parse @message "[*] *" as loggingType, loggingMessage	filter loggingType = "ERROR"	display loggingMessage`

`fields`	Ruft die angegebenen Felder aus Protokollereignissen für die Anzeige ab.
	
	Sie können Funktionen und Operationen innerhalb eines Feldbefehls verwenden, um Feldwerte für die Anzeige zu ändern und neue Felder für die Verwendung im Rest der Abfrage zu erstellen.
		Das folgende Beispiel zeigt die Felder `foo-bar`, , und Sie haben die Möglichkeit `action`und den absoluten Wert der Differenz zwischen `f3` und `f4` für alle Protokollereignisse in der Protokollgruppe.
		`fields `foo-bar`, action, abs(f3-f4)`
		
		Im folgenden Beispiel wird ein flüchtiges Feld erstellt und angezeigt `opStatus`. (z. B.. Der Wert von `opStatus` für jeden Protokolleintrag ist die Verkettung der Werte der `Operation` und `StatusCode` Felder, mit einem Bindestrich zwischen diesen Werten.
		`fields concat(Operation, '-', StatusCode) as opStatus`

`filter`	Filtert die Ergebnisse einer Abfrage basierend auf einer oder mehreren Bedingungen. Sie können eine Vielzahl von Operatoren und Ausdrücken im **`filter`** Befehl. Weitere Informationen finden Sie im the section called “Übereinstimmungen und reguläre Ausdrücke im Filterbefehl” (p. 47).	Im folgenden Beispiel werden die Felder abgerufen `f1`, , und Sie haben die Möglichkeit `f2`, und `f3` für alle Protokollereignisse mit einem Wert über 2000 im `duration` Feld.

Befehl	Description (Beschreibung)	Beispiele

		`fields f1, f2, f3 | filter (duration>2000)`
		
		Das folgende Beispiel enthält auch eine gültige Abfrage, aber die Ergebnisse zeigen keine separaten Felder an. Stattdessen zeigen die Ergebnisse die `@timestamp` und alle Protokolldaten im `@message` Feld für alle Protokollereignisse, bei denen die Dauer mehr als 2000 beträgt.
		
		`filter (duration>2000)`
		
		Im folgenden Beispiel werden die Felder abgerufen `f1` und `f2` für alle Protokollereignisse, bei denen `f1` ist 10 oder `f3` ist mehr als 25.
		
		`fields f1, f2 | filter (f1=10 or f3>25)`
		
		Das nächste Beispiel gibt Protokollereignisse zurück, bei denen das Feld `statusCode` hat einen Wert zwischen 200 und 299.
		
		`fields f1 | filter statusCode like /2\d\d/`
		
		Das nächste Beispiel gibt Protokollereignisse zurück, die einen enthalten `statusCode` von "300", "400" oder "500".
		
		`fields @timestamp, @message | filter statusCode in [300,400,500]`
		
		Dieses letzte Beispiel gibt Protokollereignisse zurück, die nicht haben `Type` Felder mit Werten von "foo", "bar" oder "1".
		
		`fields @timestamp, @message | filter Type not in ["foo","bar",1]`

Befehl	Description (Beschreibung)	Beispiele
`stats`	Berechnet aggregierte Statistiken basierend auf den Werten von Protokollfeldern. Wenn Sie `stats` verwenden, können Sie auch `by` verwenden, um ein oder mehrere Kriterien anzugeben, nach denen Daten bei der Berechnung der Statistik gruppiert werden sollen. Mehrere statistische Operatoren werden unterstützt, darunter `sum()`, `avg()`, `count()`, `min()`und `max()`.	Das folgende Beispiel berechnet den Durchschnittswert von `f1` für jeden eindeutigen Wert von `f2`. `stats avg (f1) by f2`
`sort`	Sortiert die abgerufenen Protokollereignisse. Sowohl aufsteigend (`asc`) als auch absteigend (`desc`) werden unterstützt.	Das folgende Beispiel sortiert die zurückgegebenen Ereignisse in absteigender Reihenfolge basierend auf dem Wert von `f1`und zeigt die Felder `f1`, , und Sie haben die Möglichkeit `f2`, und `f3`. `fields f1, f2, f3 | sort f1 desc`
`limit`	Legt die Anzahl der von der Abfrage zurückgegebenen Protokollereignisse fest. Sie können die Ergebnisse damit auf eine kleine Zahl beschränken, um nur wenige, relevante Ergebnisse aufzurufen. Sie können auch `limit` mit einer Zahl zwischen 1000 und 10.000 verwenden, um die in der Konsole angezeigten Abfrageergebniszeilen auf eine Zahl zu erhöhen, die größer als der Standardwert von 1000 Zeilen ist. Wenn Sie kein Limit angeben, zeigt die Abfrage standardmäßig maximal 1000 Zeilen an.	Das folgende Beispiel sortiert die Ereignisse in absteigender Reihenfolge basierend auf dem Wert von `@timestamp`und zeigt die Felder `f1` und `f2` für die ersten 25 Ereignisse nach Sortierreihenfolge. In diesem Fall erfolgt die Sortierung nach Zeitstempel beginnend mit den aktuellen Ereignissen, sodass die letzten 25 Ereignisse zurückgegeben werden. `sort @timestamp desc | limit 25  | display f1, f2`

Befehl	Description (Beschreibung)	Beispiele

`parse`	Extrahiert Daten aus einem Protokollfeld und erstellt ein oder mehrere flüchtige Felder, die Sie in der Abfrage weiter verarbeiten können. `parse` akzeptiert sowohl Glob-Ausdrücke als auch reguläre Ausdrücke.
	
	Geben Sie für glob-Ausdrücke den Befehl `parse` mit einer konstanten Zeichenfolge an (Zeichen werden von einfachen oder doppelten Anführungszeichen umschlossen), wobei alle variablen Teile des Textes durch ein Sternchen (*) ersetzt werden. Diese werden in der Reihenfolge der Position in kurzlebige Felder extrahiert und erhalten nach dem Schlüsselwort `as` einen Alias.
	
	Setzen Sie reguläre Ausdrücke in Schrägstriche (/). Innerhalb des Ausdrucks wird jeder Teil der Zeichenfolgenentsprechung, der extrahiert werden soll, in eine benannte erfassende Gruppe gestellt. Ein Beispiel einer benannten erfassenden Gruppe ist `(?<name>.*)`, wobei `name` der Name und `.*` das Muster ist.
		Mit dieser einzelnen Protokollzeile als Beispiel:
		
		`25 May 2019 10:24:39,474 [ERROR] {foo=2, bar=data} The error was: DataIntegrityException`
		
		Die folgenden zwei `parse` Ausdrücke führen jeweils Folgendes aus: die flüchtigen Felder `level`, , und Sie haben die Möglichkeit `config`, und `exception` erstellt werden. `level` hat einen Wert von `ERROR`, , und Sie haben die Möglichkeit `config` hat einen Wert von `{foo=2, bar=data}`, und `exception` hat einen Wert von `DataIntegrityException`. (z. B..) Das erste Beispiel verwendet einen glob-Ausdruck und das zweite einen regulären Ausdruck.
		
		`parse @message "[*] * The error was: *" as level, config, exception`
		
		`parse @message /\[(?<level>\S+)\]\s+(?<config>\{.*\})\s+The error was: (?<exception>\S+)/`
		
		Im folgenden Beispielen wird ein regulärer Ausdruck zum Extrahieren der kurzlebigen Felder `@user2`, `@method2` und `@latency2` aus dem Protokollfeld `@message` und zur Rückgabe der durchschnittlichen Latenz für jede eindeutige Kombination aus `@method2` und `@user2` verwendet.
		
		`parse @message /user=(?<user2>.*?), method:(?<method2>.*?), latency := (?<latency2>.*?)/ | stats avg(@latency2) by @method2, @user2`

Hinweise zu Abfragebefehlen in der vorherigen Tabelle

Die folgenden Regeln, Richtlinien und Tipps gelten für die Abfragebefehle in der vorherigen Tabelle.

- Jedes in einer Abfrage genannte Protokollfeld, das andere Zeichen als @, Punkt (.) und alphanumerische Zeichen enthält, muss von Backtick-Zeichen (`) umschlossen werden. Beispielsweise muss der Feldname `foo-bar` von Backtick-Zeichen umschlossen werden, da er ein nicht alphanumerisches Zeichen enthält.
- Sowohl als auch **`fields`** und **`display`** werden verwendet, um die Felder anzugeben, die in den Abfrageergebnissen angezeigt werden sollen. Die Unterschiede zwischen den beiden lauten wie folgt:

- Sie verwenden die **`display`** , um anzugeben, welche Felder in den Ergebnissen angezeigt werden sollen. Sie können die **`fields`** Befehl mit der und zwar als Schlüsselwort zum Erstellen neuer flüchtiger Felder mithilfe von -Funktionen und den Feldern, die sich im Protokollereignis befinden. Zum Beispiel: `fields ispresent(resolverArn) as isRes` erstellt ein flüchtiges Feld mit dem Namen `isRes` die im Rest der Abfrage verwendet werden können. Der Wert von `isRes` ist entweder 0 oder 1, je nachdem, ob `resolverArn` ist ein erkanntes Feld im Protokollereignis.
- Wenn Sie mehrere **`fields`** und enthalten keine **`display`** die Felder, die in allen **`fields`** `commands are displayed.`
- Wenn Sie mehrere **`display`** nur die Felder, die im letzten **`display`** `command are displayed.`

Übereinstimmungen und reguläre Ausdrücke im Filterbefehl

Sie können Vergleichsoperatoren (**=**, **!=**, **<**, **<=**, **>**, **>=**), Boolesche Operatoren (`and`, , und Sie haben die Möglichkeit `or`, und `not`) und reguläre Ausdrücke im **`filter`** Befehl.

Sie können `in` zum Testen auf festgelegte Mitgliedschaft verwenden. Stellen Sie ein Array mit den Elementen ein, nach denen sofort gesucht werden soll `in`. (z. B.. Sie können `not` mit dem `in`. (z. B.. Zeichenfolgenübereinstimmungen mit `in` muss vollständige Zeichenfolgenübereinstimmungen sein.

Um nach Teilzeichenfolgen zu filtern, können Sie `like` oder =~ (Gleichheitszeichen gefolgt von einer Tilde) im `filter` Befehl. Für eine Teilzeichenfolge-Übereinstimmung mit `like` oder =~, schließen Sie Ihre Teilzeichenfolge ein, um mit doppelten oder einfachen Anführungszeichen zu übereinstimmen. Um einen regulären Ausdrucksabgleich durchzuführen, schließen Sie den Ausdruck ein, der mit den Schrägstrichen übereinstimmt. Die Abfrage gibt nur Protokollereignisse zurück, die den von Ihnen festgelegten Kriterien entsprechen.

Beispiele

Die folgenden drei Beispiele geben alle Ereignisse zurück, in denen `f1` enthält das Wort `Exception`. (z. B.. Die ersten beiden Beispiele verwenden reguläre Ausdrücke. Das dritte Beispiel verwendet eine Teilzeichenfolgenübereinstimmung. Bei allen drei Beispielen muss die Groß-/Kleinschreibung beachtet werden.

```
fields f1, f2, f3 | filter f1 like /Exception/
```

```
fields f1, f2, f3 | filter f1 =~ /Exception/
```

```
fields f1, f2, f3 | filter f1 like "Exception"
```

Im folgenden Beispiel wird die Suche nach „Ausnahme“ in nicht Groß-/Kleinschreibung beachtet.

```
fields f1, f2, f3 | filter f1 like /(?i)Exception/
```

Das folgende Beispiel verwendet einen regulären Ausdruck. Es gibt alle Ereignisse zurück, in denen `f1` ist genau das Wort `Exception`. (z. B.. Bei der Abfrage wird nicht zwischen Groß- und Kleinschreibung unterschieden.

```
fields f1, f2, f3 | filter f1 =~ /^(?i)Exception$/
```

Verwenden von Aliasen in Abfragen

Sie können mit `as` einen oder mehrere Aliase in einer Abfrage erstellen. Aliase werden in den Befehlen `fields`, `stats` und `sort` unterstützt.

Sie können Aliase für Protokollfelder sowie für die Ergebnisse von Operationen und Funktionen anlegen.

Beispiele

Die folgenden Beispiele zeigen die Verwendung von Aliasen in Abfragebefehlen.

```
fields abs(myField) as AbsoluteValuemyField, myField2
```

Gibt den absoluten Wert von `myField` als `AbsoluteValuemyField` und das Feld `myField2` zurück.

```
stats avg(f1) as myAvgF1 | sort myAvgF1 desc
```

Berechnet den Mittelwert der Werte von `f1` als `myAvgF1` und gibt sie in absteigender Reihenfolge dieses Werts zurück.

Verwenden von Kommentaren in Abfragen

Sie können Zeilen in einer Query mithilfe des Zeichens # auskommentieren. Zeilen, die mit dem Zeichen # beginnen, werden ignoriert. Dies kann nützlich sein, um Ihre Abfrage zu dokumentieren oder einen Teil einer komplexen Abfrage für einen Aufruf vorübergehend zu ignorieren, ohne diese Zeile zu löschen.

Im folgenden Beispiel wird die zweite Zeile der Abfrage ignoriert.

```
fields @timestamp, @message
    # | filter @message like /delay/
    | limit 20
```

Unterstützte Operationen und Funktionen

Die Abfragesprache unterstützt viele Arten von Operationen und Funktionen, wie in den folgenden Tabellen dargestellt.

Vergleichsoperationen

Sie können Vergleichsoperationen im Befehl `filter` sowie als Argumente für andere Funktionen verwenden. Vergleichsoperationen akzeptieren alle Datentypen als Argumente und liefern ein boolesches Ergebnis.

```
= != < <= > >=
```

Boolesche Operatoren

Sie können die booleschen Operatoren verwenden, **`and`**, , und Sie haben die Möglichkeit **`or`**, und **`not`**. (z. B.. Sie können diese booleschen Operatoren nur in Funktionen verwenden, die einen booleschen Wert zurückgeben.

Arithmetische Operationen

Sie können arithmetische Operationen in den Befehlen `filter` und `fields` sowie als Argumente für andere Funktionen verwenden. Arithmetische Operationen akzeptieren numerische Datentypen als Argumente und liefern numerische Ergebnisse.

Operation	Description (Beschreibung)
`a + b`	Addition
`a - b`	Subtraktion
`a * b`	Multiplikation
`a / b`	Division
`a ^ b`	Erklärung. `2 ^ 3` Rückgaben `8`
`a % b`	Rest oder Modul. `10 % 3` Rückgaben `1`

Numerische Operationen

Sie können numerische Operationen in den Befehlen `filter` und `fields` sowie als Argumente für andere Funktionen verwenden. Numerische Operationen akzeptieren numerische Datentypen als Argumente und liefern numerische Ergebnisse.

Operation	Ergebnistyp	Description (Beschreibung)
`abs(a: number)`	Zahl	Absoluter Wert.
`ceil(a: number)`	Zahl	Aufrunden (die kleinste ganze Zahl, die größer ist als der Wert von `a`).
`floor(a: number)`	Zahl	Abrunden (die größte ganze Zahl, die kleiner ist als der Wert von `a`).
`greatest(a: number, ...numbers: number[])`	Zahl	Liefert den größten Wert.
`least(a: number, ...numbers: number[])`	Zahl	Liefert den kleinsten Wert.
`log(a: number)`	Zahl	Natürlicher Logarithmus.
`sqrt(a: number)`	Zahl	Quadratwurzel.

Allgemeine Funktionen

Sie können allgemeine Funktionen in den Befehlen `filter` und `fields` sowie als Argumente für andere Funktionen verwenden.

Funktion	Ergebnistyp	Description (Beschreibung)
`ispresent(fieldName: LogField)`	Boolean	Gibt `true` zurück, wenn das Feld existiert.
`coalesce(fieldName: LogField, ...fieldNames: LogField[])`	Protokollfeld	Liefert den ersten Nicht-Null-Wert aus der Liste.

Funktionen für Zeichenfolgen

Sie können Zeichenfolgenfunktionen in den Befehlen `filter` und `fields` sowie als Argumente für andere Funktionen verwenden.

Funktion	Ergebnistyp	Description (Beschreibung)
`isempty(fieldName: string)`	Boolean	Gibt `true` zurück, wenn das Feld fehlt oder eine leere Zeichenkette ist.
`isblank(fieldName: string)`	Boolean	Gibt `true` zurück, wenn das Feld fehlt, eine leere Zeichenkette ist oder nur Leerzeichen enthält.
`concat(str: string, ...strings: string[])`	: Zeichenfolge	Verkettet die Zeichenketten.

`ltrim(str: string)`
 	
`ltrim(str: string, subStr: string)`
	: Zeichenfolge	Entfernt Leerzeichen auf der linken Seite der Zeichenfolge. Wenn die Funktion über ein zweites Zeichenfolgenargument verfügt, werden die Zeichen von entfernt `subStr` von der linken Seite `str`. (z. B.. Zum Beispiel: `ltrim("xyZfooxyZ","xyZ")` Rückgaben `"fooxyZ"`.

`rtrim(str: string)`
 	
`rtrim(str: string, subStr: string)`
	: Zeichenfolge	Entfernt Leerzeichen auf der rechten Seite der Zeichenfolge. Wenn die Funktion über ein zweites Zeichenfolgenargument verfügt, werden die Zeichen von entfernt `subStr` von der rechten Seite `str`. (z. B.. Zum Beispiel: `rtrim("xyZfooxyZ","xyZ")` Rückgaben `"xyZfoo"`.

`trim(str: string)`
 	
`trim(str: string, subStr: string)`
	: Zeichenfolge	Entfernt Leerzeichen an beiden Enden der Zeichenfolge. Wenn die Funktion über ein zweites Zeichenfolgenargument verfügt, werden die Zeichen von entfernt `subStr` von beiden Seiten der `str`. (z. B.. Zum Beispiel: `trim("xyZfooxyZ","xyZ")` Rückgaben `"foo"`.

`strlen(str: string)`	Zahl	Liefert die Länge der Zeichenkette in Unicode-Codepunkten.
`toupper(str: string)`	: Zeichenfolge	Konvertiert die Zeichenkette in Großbuchstaben.
`tolower(str: string)`	: Zeichenfolge	Konvertiert die Zeichenkette in Kleinbuchstaben.

Funktion	Ergebnistyp	Description (Beschreibung)

`substr(str: string, startIndex: number)`
 	
`substr(str: string, startIndex: number, length: number)`
	: Zeichenfolge	Gibt eine Teilzeichenkette aus dem durch das Zahlenargument angegebenen Index bis zum Ende der Zeichenkette zurück. Wenn die Funktion ein zweites Zahlenargument hat, enthält sie die Länge der abzurufenden Teilzeichenkette. Beispielsweise gibt `substr("xyZfooxyZ",3, 3) "foo"` zurück.

`replace(str: string, searchValue: string, replaceValue: string)`	: Zeichenfolge	Ersetzt alle Instanzen von `searchValue` in (in) `str` mit dem `replaceValue`. (z. B.. Zum Beispiel: `replace("foo","o","0")` Rückgaben `"f00"`.
`strcontains(str: string, searchValue: string)`	Zahl	Gibt 1 zurück, wenn `str` `searchValue` enthält; ansonsten 0.

DateTime-Funktionen

Sie können die Datums-/Uhrzeit-Funktionen in den Befehlen `filter` und `fields` sowie als Argumente für andere Funktionen verwenden. Mit diesen Funktionen können Sie Zeiträume für Abfragen mit Aggregationsfunktionen anlegen.

Im Rahmen der Datumsfunktionen können Sie Zeiträume verwenden, die aus einer Zahl und dann entweder `m` für Minuten oder `h` für Stunden bestehen. Zum Beispiel steht `10m` für 10 Minuten und `1h` für 1 Stunde.

Funktion	Ergebnistyp	Description (Beschreibung)
`bin(period: Period)`	Zeitstempel	Rundet den Wert von `@timestamp` auf den angegebenen Zeitraum und kürzt ihn dann.
`datefloor(timestamp: Timestamp, period: Period)`	Zeitstempel	Kürzt den Zeitstempel auf den angegebenen Zeitraum. Zum Beispiel kürzt `datefloor(@timestamp, 1h)` alle Werte von `@timestamp` auf die letzte volle Stunde.
`dateceil(timestamp: Timestamp, period: Period)`	Zeitstempel	Rundet den Zeitstempel auf den angegebenen Zeitraum auf und kürzt ihn dann. Zum Beispiel kürzt `dateceil(@timestamp, 1h)` alle Werte von `@timestamp` auf die nächste volle Stunde.
`fromMillis(fieldName: number)`	Zeitstempel	Interpretiert das Eingabefeld als die Anzahl der Millisekunden seit der Unix-Epoche und konvertiert es in einen Zeitstempel.
`toMillis(fieldName: Timestamp)`	Zahl	Konvertiert den im benannten Feld gefundenen Zeitstempel in eine Zahl, die die Millisekunden seit der Unix-Epoche darstellt.

IP-Adress-Funktionen

Sie können IP-Adressen-Zeichenfolgenfunktionen in den Befehlen `filter` und `fields` sowie als Argumente für andere Funktionen verwenden.

Funktion	Ergebnistyp	Description (Beschreibung)
`isValidIp(fieldName: string)`	Boolean	Gibt `true` zurück, wenn das Feld eine gültige IPv4- oder IPv6-Adresse ist.
`isValidIpV4(fieldName: string)`	Boolean	Gibt `true` zurück, wenn das Feld eine gültige IPv4-Adresse ist.
`isValidIpV6(fieldName: string)`	Boolean	Gibt `true` zurück, wenn das Feld eine gültige IPv6-Adresse ist.
`isIpInSubnet(fieldName: string, subnet: string)`	Boolean	Gibt `true` zurück, wenn das Feld eine gültige IPv4- oder IPv6-Adresse innerhalb des angegebenen IPv4- oder IPv6-Subnetzes ist. Wenn Sie das Subnetz angeben, verwenden Sie die CIDR-Notation wie `192.0.2.0/24` oder `2001:db8::/32`.
`isIpv4InSubnet(fieldName: string, subnet: string)`	Boolean	Gibt `true` zurück, wenn das Feld eine gültige IPv4-Adresse innerhalb des angegebenen v4-Subnetzes ist. Wenn Sie das Subnetz angeben, verwenden Sie die CIDR-Notation wie `192.0.2.0/24`.
`isIpv6InSubnet(fieldName: string, subnet: string)`	Boolean	Gibt `true` zurück, wenn das Feld eine gültige IPv6-Adresse innerhalb des angegebenen v6-Subnetzes ist. Wenn Sie das Subnetz angeben, verwenden Sie die CIDR-Notation wie `2001:db8::/32`.

Funktionen für die Aggregation von Statistiken

Sie können Aggregationsfunktionen im Befehl `stats` sowie als Argumente für andere Funktionen verwenden.

Funktion	Ergebnistyp	Description (Beschreibung)
`avg(fieldName: NumericLogField)`	Zahl	Der Mittelwert der Werte im angegebenen Feld.

`count()`
 	
`count(fieldName: LogField)`
	Zahl	Zählt die Protokollereignisse. `count()` (oder `count(*)`) zählt alle von der Abfrage zurückgegebenen Ereignisse, während `count(fieldName)` zählt alle Datensätze, die den angegebenen Feldnamen enthalten.

`count_distinct(fieldName: LogField)`	Zahl	Liefert die Anzahl der eindeutigen Werte für das Feld. Wenn das Feld eine sehr hohe Kardinalität hat (zahlreiche eindeutige Werte enthält), ist der von `count_distinct` zurückgegebene Wert lediglich eine Annäherung.
`max(fieldName: LogField)`	Protokollfeldwert	Das Maximum der Werte für dieses Protokollfeld in den abgefragten Protokollen.

Funktion	Ergebnistyp	Description (Beschreibung)
`min(fieldName: LogField)`	Protokollfeldwert	Das Minimum der Werte für dieses Protokollfeld in den abgefragten Protokollen.
`pct(fieldName: LogFieldValue, percent: number)`	Protokollfeldwert	Ein Perzentil gibt die relative Stelle eines Wertes in einer Datenmenge an. Zum Beispiel gibt `pct(@duration, 95)` den `@duration`-Wert zurück, bei dem 95 Prozent der Werte von `@duration` niedriger als dieser Wert und 5 Prozent höher als dieser Wert sind.
`stddev(fieldName: NumericLogField)`	Zahl	Die Standardabweichung der Werte im angegebenen Feld.
`sum(fieldName: NumericLogField)`	Zahl	Die Summe der Werte im angegebenen Feld.

Nicht-Aggregationsfunktionen für Statistiken

Sie können Nicht-Aggregationsfunktionen im Befehl `stats` sowie als Argumente für andere Funktionen verwenden.

Funktion	Ergebnistyp	Description (Beschreibung)
`earliest(fieldName: LogField)`	Protokollfeld	Gibt den Wert von `fieldName` aus dem Protokollereignis mit dem frühesten Zeitstempel in den abgefragten Protokollen zurück.
`latest(fieldName: LogField)`	Protokollfeld	Gibt den Wert von `fieldName` aus dem Protokollereignis mit dem neuesten Zeitstempel in den abgefragten Protokollen zurück.
`sortsFirst(fieldName: LogField)`	Protokollfeld	Gibt den Wert von `fieldName` zurück, der in der Sortierung der abgefragten Protokolle an erster Stelle steht.
`sortsLast(fieldName: LogField)`	Protokollfeld	Gibt den Wert von `fieldName` zurück, der in der Sortierung der abgefragten Protokolle an letzter Stelle steht.

Visualisieren von Protokolldaten in Diagrammen

Sie können Visualisierungen wie Balkendiagramme, Liniendiagramme und gestapelte Flächendiagramme verwenden, um Muster in Ihren Protokolldaten effizienter zu identifizieren. CloudWatch Logs Insights erzeugt Visualisierungen für Abfragen, die die `stats`-Funktion und eine oder mehrere Aggregationsfunktionen verwenden. Weitere Informationen finden Sie im Aggregation Functions in the Stats Command (p. 52).

Mit all diesen Abfragen können Balkendiagramme erzeugt werden. Wenn Ihre Abfrage die Funktion `bin()` verwendet, um Daten nach einem einzelnen Feld im Zeitverlauf zu gruppieren, können Sie auch Liniendiagramme und gestapelte Flächendiagramme anzeigen.

Themen

- Visualisierung von Zeitreihendaten (p. 54)

- Visualisieren von nach Feldern gruppierten Protokolldaten (p. 54)

Visualisierung von Zeitreihendaten

Visualisierungen von Zeitreihen funktionieren für Abfragen mit folgenden Merkmalen:

- Die Abfrage enthält eine oder mehrere Aggregationsfunktionen. Weitere Informationen finden Sie im Aggregation Functions in the Stats Command (p. 52).
- Die Abfrage verwendet die `bin()`-Funktion. Damit können Sie die Daten nach einem Feld gruppieren.

Diese Abfragen können Liniendiagramme, gestapelte Flächendiagramme, Balkendiagramme und Kreisdiagramme erzeugen.

Beispiele

Ein vollständiges Tutorial finden Sie unter the section called “Praktische Anleitung Ausführen einer Abfrage, die eine Zeitreihenvisualisierung erzeugt” (p. 42).

Im Folgenden finden Sie weitere Beispielabfragen, die für die Zeitreihenvisualisierung funktionieren.

Die folgende Abfrage erzeugt eine Visualisierung der Durchschnittswerte des `myfield1`-Felds an, mit einem Datenpunkt, der alle fünf Minuten erstellt wird. Jeder Datenpunkt ist die Aggregation der Durchschnitte der `myfield1`-Werte aus den Protokollen der letzten fünf Minuten.

```
stats avg(myfield1) by bin(5m)
```

Die folgende Abfrage erzeugt eine Visualisierung von drei Werten basierend auf verschiedenen Feldern, wobei alle fünf Minuten ein Datenpunkt erstellt wird. Die Visualisierung wird erzeugt, weil die Abfrage Aggregationsfunktionen enthält und `bin()` als Gruppierungsfeld verwendet.

```
stats avg(myfield1), min(myfield2), max(myfield3) by bin(5m)
```

Einschränkungen von Liniendiagrammen und gestapelten Flächendiagrammen

Abfragen, die Protokolleintragsinformationen aggregieren, die Funktion `bin()` jedoch nicht verwenden, können Balkendiagramme generieren. Die Abfragen können jedoch keine Liniendiagramme oder gestapelte Flächendiagramme generieren. Weitere Informationen zu diesen Abfragetypen finden Sie unter the section called “Visualisieren von nach Feldern gruppierten Protokolldaten” (p. 54).

Visualisieren von nach Feldern gruppierten Protokolldaten

Sie können Balkendiagramme für Abfragen erstellen, die die `stats`-Funktion und eine oder mehrere Aggregationsfunktionen verwenden. Weitere Informationen finden Sie im Aggregation Functions in the Stats Command (p. 52).

Führen Sie die Abfrage aus, um die Visualisierung aufzurufen. Wählen Sie dann die Registerkarte Visualisierung aus, klicken Sie auf den Pfeil neben Linie und auf Balken. Visualisierungen sind mit maximal 100 Balken im Balkendiagramm beschränkt.

Beispiele

Ein vollständiges Tutorial finden Sie unter the section called “Praktische Anleitung Ausführen einer Abfrage, die eine Visualisierung erzeugt, die nach Protokollfeldern gruppiert ist” (p. 41). Die folgenden Absätze enthalten weitere Beispielabfragen für die Visualisierung nach Feldern.

Die folgende VPC-Flow-Protokollabfrage ermittelt die durchschnittliche Anzahl von Bytes, die pro Sitzung für die einzelnen Zieladressen übertragen werden.

```
stats avg(bytes) by dstAddr
```

Sie können auch ein Diagramm erstellen, das mehr als einen Balken für jeden resultierenden Wert enthält. Die folgende VPC-Flow-Protokollabfrage ermittelt beispielsweise die durchschnittliche und maximale Anzahl von Bytes, die pro Sitzung an die einzelnen Zieladressen übertragen werden.

```
stats avg(bytes), max(bytes) by dstAddr
```

Die folgende Abfrage ermittelt die Anzahl der Amazon Route 53-Abfrageprotokolle für jeden Abfragetyp.

```
stats count(*) by queryType
```

Speichern und erneutes Ausführen von CloudWatch Logs Insights-Abfragen

Nachdem Sie eine Abfrage erstellt haben, können Sie sie speichern, um sie später erneut auszuführen. Ihre gespeicherten Abfragen werden in einer Ordnerstruktur gespeichert, damit Sie sie organisieren können. Sie können bis zu 1000 CloudWatch Logs Insights-Abfragen pro Region und Konto speichern.

Um eine Abfrage zu speichern, müssen Sie bei einer Rolle mit der Berechtigung `logs:PutQueryDefinition` angemeldet sein. Um die Liste der gespeicherten Abfragen anzuzeigen, müssen Sie bei einer Rolle mit der Berechtigung `logs:DescribeQueryDefinitions` angemeldet sein.

So speichern Sie eine Abfrage

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Insights aus.
3. Erstellen Sie im Abfrage-Editor eine Abfrage.
4. Wählen Sie Save aus.

   Wenn Ihnen die Schaltfläche Save (Speichern) nicht angezeigt wird, müssen Sie zum neuen Design für die CloudWatch Logs-Konsole wechseln. Hierzu gehen Sie wie folgt vor:

   a. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus.
   b. Wählen Sie Try the new design (Neues Design testen) aus.
   c. Wählen Sie im Navigationsbereich Insights (Einsichten) aus und kehren Sie zu Schritt 3 dieses Verfahrens zurück.
5. Geben Sie einen Namen für die Abfrage ein.
6. (Optional) Wählen Sie den Ordner aus, in dem Sie die Abfrage speichern möchten. Wählen Sie Create new (Neu erstellen) aus, um einen Ordner zu erstellen. Wenn Sie einen neuen Ordner erstellen, können Sie Schrägstriche (/) im Ordnernamen verwenden, um eine Ordnerstruktur zu definieren. Wenn Sie beispielsweise einen neuen Ordner mit **`folder-level-1/folder-level-2`** benennen, wird der Ordner **`folder-level-1`** auf der obersten Ebene erstellt. In diesem Ordner befindet sich ein weiterer Ordner namens **`folder-level-2`**. Die Abfrage wird in **`folder-level-2`** gespeichert.
7. (Optional) Ändern Sie die Protokollgruppen oder den Abfragetext der Abfrage.
8. Wählen Sie Save aus.

So führen Sie eine gespeicherte Abfrage aus

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Insights aus.
3. Wählen Sie auf der rechten Seite Queries (Abfragen) aus.
4. Wählen Sie Ihre Abfrage aus der Liste Saved queries (Gespeicherte Abfragen) aus. Anschließend wird sie im Abfrage-Editor angezeigt.
5. Wählen Sie Run aus.

So speichern Sie eine neue Version einer gespeicherten Abfrage

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Insights aus.
3. Wählen Sie auf der rechten Seite Queries (Abfragen) aus.
4. Wählen Sie Ihre Abfrage aus der Liste Saved queries (Gespeicherte Abfragen) aus. Anschließend wird sie im Abfrage-Editor angezeigt.
5. Ändern Sie die Abfrage. Wenn Sie die Abfrage ausführen müssen, um Ihre Arbeit zu überprüfen, wählen Sie Run query (Abfrage ausführen) aus.
6. Wenn Sie bereit sind, die neue Version zu speichern, wählen Sie Actions (Aktionen) und Save as (Speichern unter) aus.
7. Geben Sie einen Namen für die Abfrage ein.
8. (Optional) Wählen Sie den Ordner aus, in dem Sie die Abfrage speichern möchten. Wählen Sie Create new (Neu erstellen) aus, um einen Ordner zu erstellen. Wenn Sie einen neuen Ordner erstellen, können Sie Schrägstriche (/) im Ordnernamen verwenden, um eine Ordnerstruktur zu definieren. Wenn Sie beispielsweise einen neuen Ordner mit **`folder-level-1/folder-level-2`** benennen, wird der Ordner **`folder-level-1`** auf der obersten Ebene erstellt. In diesem Ordner befindet sich ein weiterer Ordner namens **`folder-level-2`**. Die Abfrage wird in **`folder-level-2`** gespeichert.
9. (Optional) Ändern Sie die Protokollgruppen oder den Abfragetext der Abfrage.
10. Wählen Sie Save aus.

Um eine Abfrage zu löschen, müssen Sie bei einer Rolle mit der Berechtigung `logs:DeleteQueryDefinition` angemeldet sein.

So bearbeiten oder löschen Sie eine gespeicherte Abfrage

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Insights aus.
3. Wählen Sie auf der rechten Seite Queries (Abfragen) aus.
4. Wählen Sie Ihre Abfrage aus der Liste Saved queries (Gespeicherte Abfragen) aus. Anschließend wird sie im Abfrage-Editor angezeigt.
5. Wählen Sie Actions (Aktionen), Edit (Bearbeiten) oder Actions (Aktionen), Delete (Löschen) aus.

Beispielabfragen

Dieser Abschnitt enthält Beispielabfragen, die die Leistungsfähigkeit von CloudWatch Logs Insights demonstrieren.

Allgemeine Abfragen

Findet die 25 zuletzt hinzugefügten Protokollereignisse.

```
fields @timestamp, @message | sort @timestamp desc | limit 25
```

Liefert eine Liste der Anzahl der Ausnahmen pro Stunde.

```
filter @message like /Exception/
    | stats count(*) as exceptionCount by bin(1h)
    | sort exceptionCount desc
```

Ermittelt eine Liste von Protokollereignissen, die keine Ausnahmen sind.

```
fields @message | filter @message not like /Exception/
```

Abfragen für Lambda Protokolle

Bestimmt die Menge des zu viel bereitgestellten Speichers.

```
filter @type = "REPORT"
    | stats max(@memorySize / 1024 / 1024) as provisonedMemoryMB,
        min(@maxMemoryUsed / 1024 / 1024) as smallestMemoryRequestMB,
        avg(@maxMemoryUsed / 1024 / 1024) as avgMemoryUsedMB,
        max(@maxMemoryUsed / 1024 / 1024) as maxMemoryUsedMB,
        provisonedMemoryMB - maxMemoryUsedMB as overProvisionedMB
```

Erstellt einen Latenzbericht.

```
filter @type = "REPORT" |
    stats avg(@duration), max(@duration), min(@duration) by bin(5m)
```

Abfragen für Amazon VPC Ablaufprotokolle

Findet die 15 wichtigsten Paketübertragungen zwischen den Hosts:

```
stats sum(packets) as packetsTransferred by srcAddr, dstAddr
    | sort packetsTransferred  desc
    | limit 15
```

Findet die Top 15-Byte-Übertragungen für Hosts in einem bestimmten Subnetz.

```
filter isIpv4InSubnet(srcAddr, "192.0.2.0/24")
    | stats sum(bytes) as bytesTransferred by dstAddr
    | sort bytesTransferred desc
    | limit 15
```

Findet die IP-Adressen, die UDP als Datenübertragungsprotokoll verwenden.

```
filter protocol=17 | stats count(*) by srcAddr
```

Findet die IP-Adressen, bei denen während des Erfassungsfensters Flussaufzeichnungen übersprungen wurden.

```
filter logStatus="SKIPDATA"
    | stats count(*) by bin(1h) as t
    | sort t
```

Abfragen für Route 53 Protokolle

Findet die Verteilung der Datensätze pro Stunde nach Abfragetyp.

```
stats count(*) by queryType, bin(1h)
```

Findet die 10 DNS-Resolver mit der höchsten Anzahl von Anforderungen.

```
stats count(*) as numRequests by resolverIp
    | sort numRequests desc
    | limit 10
```

Ermittelt die Anzahl der Datensätze nach Domäne und Subdomäne, bei denen der Server die DNS-Anforderung nicht abgeschlossen hat.

```
filter responseCode="SERVFAIL" | stats count(*) by queryName
```

Abfragen für CloudTrail Protokolle

Ermittelt die Anzahl der Protokolleinträge pro Service, Ereignistyp und AWS-Region.

```
stats count(*) by eventSource, eventName, awsRegion
```

Findet die Amazon EC2-Hosts, die in einer bestimmten AWS-Region gestartet oder angehalten wurden.

```
filter (eventName="StartInstances" or eventName="StopInstances") and region="us-east-2"
```

Findet die AWS-Regionen, Benutzernamen und ARNs neu erstellter IAM-Benutzer.

```
filter eventName="CreateUser"
    | fields awsRegion, requestParameters.userName, responseElements.user.arn
```

Ermittelt die Anzahl der Datensätze, bei denen beim Aufruf des API `UpdateTrail` eine Ausnahme aufgetreten ist.

```
filter eventName="UpdateTrail" and ispresent(errorCode)
    | stats count(*) by errorCode, errorMessage
```

Beispiele des parse-Befehls

Verwenden Sie einen "glob"-Ausdruck zum Extrahieren der flüchtigen Felder `@user`, `@method` und `@latency` aus dem Protokollfeld `@message` und zur Rückgabe der durchschnittlichen Latenz für jede eindeutige Kombination aus `@method` und `@user`.

```
parse @message "user=*, method:*, latency := *" as @user,
```

```
    @method, @latency | stats avg(@latency) by @method,
    @user
```

Verwenden Sie einen regulären Ausdruck zum Extrahieren der kurzlebigen Felder `@user2`, `@method2` und `@latency2` aus dem Protokollfeld `@message` und zur Rückgabe der durchschnittlichen Latenz für jede eindeutige Kombination aus `@method2` und `@user2`.

```
parse @message /user=(?<user2>.*?), method:(?<method2>.*?),
    latency := (?<latency2>.*?)/ | stats avg(@latency2) by @method2,
    @user2
```

Extrahiert die flüchtigen Felder `loggingTime`, `loggingType` und `loggingMessage`, filtert nach Protokollereignissen, die die Zeichenfolgen `ERROR` oder `INFO` enthalten, und zeigt dann nur die Felder `loggingMessage` und `loggingType` für Ereignisse an, die die Zeichenfolge `ERROR` enthalten.

```
FIELDS @message
    | PARSE @message "* [*] *" as loggingTime, loggingType, loggingMessage
    | FILTER loggingType IN ["ERROR", "INFO"]
    | DISPLAY loggingMessage, loggingType = "ERROR" as isError
```

Abfrage zum Dashboard hinzufügen oder Abfrageergebnisse exportieren

Nachdem Sie eine Abfrage ausgeführt haben, können Sie die Abfrage zu einem CloudWatch-Dashboard hinzufügen oder die Ergebnisse in die Zwischenablage kopieren.

Zu Dashboards hinzugefügte Abfragen werden bei jedem Laden des Dashboards und bei jeder Aktualisierung des Dashboards ausgeführt. Diese Abfragen zählen zu Ihrem Limit von vier gleichzeitigen CloudWatch Logs Insights-Abfragen.

So fügen Sie Abfrageergebnisse zu einem Dashboard hinzu

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Insights aus.
3. Wählen Sie eine oder mehrere Protokollgruppen aus und führen Sie eine Abfrage aus.
4. Wählen Sie Add to dashboard (Zu Dashboard hinzufügen) aus.
5. Wählen Sie das Dashboard aus. Sie können auch Create new (Neu erstellen) auswählen, um ein Dashboard für die Abfrageergebnisse zu erstellen.
6. Wählen Sie den Widget-Typ aus, der für die Abfrageergebnisse verwendet werden soll.
7. Geben Sie einen Namen für das Widget ein.
8. Wählen Sie Add to dashboard (Zu Dashboard hinzufügen) aus.

So kopieren Sie Abfrageergebnisse in die Zwischenablage oder laden die Abfrageergebnisse herunter

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Insights aus.
3. Wählen Sie eine oder mehrere Protokollgruppen aus und führen Sie eine Abfrage aus.
4. Wählen Sie Export results (Ergebnisse exportieren) und dann die gewünschte Option aus.

Anzeigen von laufenden Abfragen oder Abfrageverlauf

Sie können die aktuell laufenden Abfragen sowie Ihren aktuellen Abfrageverlauf einsehen.

Abfragen, die derzeit ausgeführt werden, beinhalten Abfragen, die Sie einem Dashboard hinzugefügt haben. Sie sind auf vier gleichzeitige CloudWatch Logs Insights-Abfragen pro Konto beschränkt, einschließlich Abfragen, die zu Dashboards hinzugefügt wurden.

So zeigen Sie Ihren aktuellen Abfrageverlauf an

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Insights aus.
3. Wählen Sie History (Verlauf) aus, wenn Sie das neue Design für die CloudWatch Logs-Konsole verwenden. Wenn Sie das alte Design verwenden, wählen Sie Actions (Aktionen) und dann View query history for this account (Abfrageverlauf für dieses Konto anzeigen) aus.

   Eine Liste Ihrer letzten Abfragen wird angezeigt. Sie können sie erneut ausführen, indem Sie die Abfrage und dann Run (Ausführen) auswählen.

   CloudWatch Logs zeigt in Status (Status) für alle aktuell ausgeführten Abfragen In progress (In Bearbeitung) an.

Arbeiten mit Log-Gruppen und Log-Streams

Ein Protokollstream ist eine Abfolge von Protokollereignissen, die dieselbe Quelle nutzen. Jede separate Quelle für Protokolle in CloudWatch Logs bildet einen separaten Protokollstream.

Eine Protokollgruppe ist eine Gruppe von Protokollstreams, die dieselben Einstellungen für die Aufbewahrung, Überwachung und Zugriffskontrolle besitzen. Sie können Protokollgruppen definieren und angeben, welche Streams in welche Gruppe geschickt werden sollen. Es gibt keine Begrenzung dazu, wie viele Protokoll-Streams zu einer Protokollgruppe gehören können.

Verwenden Sie die Verfahren in diesem Abschnitt für die Arbeit mit Log-Gruppen und Log-Streams.

In CloudWatch Logs eine Protokollgruppe erstellen

Wenn Sie den CloudWatch Logs-Agenten auf einer Amazon EC2-Instance mithilfe der in den vorherigen Abschnitten genannten Schritte des Amazon CloudWatch Logs User Guide installieren, wird die Protokollgruppe im Rahmen dieses Vorgangs erstellt. Darüber hinaus können Sie eine Protokollgruppe auch direkt in der CloudWatch-Konsole erstellen.

So erstellen Sie eine Protokollgruppe

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus.
3. Wählen Sie Actions (Aktionen) und anschließend Create log group (Protokollgruppe erstellen) aus.
4. Geben Sie einen Namen für die Protokollgruppe ein. Wählen Sie anschließend Create log group (Protokollgruppe erstellen) aus.

Protokolle an eine Protokollgruppe senden

CloudWatch Logs empfängt automatisch Protokollereignisse von mehreren AWS Services. Sie können auch andere Protokollereignisse an CloudWatch Logs unter Verwendung eines der folgenden Methoden:

- CloudWatch Agent— Die vereinheitlichte CloudWatch Agenten kann sowohl Metriken als auch Protokolle an CloudWatch Logs. Informationen zur Installation und Verwendung der CloudWatch Agent, siehe Erfassen von Metriken und Protokollen von Amazon EC2 Instanzen und Vor-Ort-Server mit CloudWatch Agent im Amazon CloudWatch-Benutzerhandbuch.
- AWS CLI—Die Put-Log-Ereignisse lädt Stapel von Protokollereignissen in CloudWatch Logs.
- Programmatisch— Die Putlogevents API ermöglicht Ihnen programmatisch, Chargen von Protokollereignissen auf CloudWatch Logs.

An CloudWatch Logs gesendete Protokolldaten anzeigen lassen

Sie können sich die Protokolldaten Stream für Stream ansehen und durchscrollen, so wie sie vom CloudWatch Logs-Agenten an CloudWatch Logs gesendet wurden. Sie können den Zeitraum für die Anzeige der Protokolldaten festlegen.

So lassen Sie sich Protokolldaten anzeigen

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus.
3. Wählen Sie für Log Groups die Protokollgruppe, um die Streams anzuzeigen.
4. Wählen Sie in der Liste der Protokollgruppen den Namen der Protokollgruppe aus, die Sie anzeigen möchten.
5. Wählen Sie aus der Liste der Protokoll-Streams den Namen des Protokoll-Streams aus, den Sie anzeigen möchten.
6. Um die Ansicht der Protokolldaten zu ändern, führen Sie einen der folgenden Schritte aus:

   - Um ein einzelnes Protokollereignis zu erweitern, wählen Sie den Pfeil neben diesem Protokollereignis.
   - Um alle Protokollereignisse zu expandieren und diese als Nur-Text anzuzeigen, wählen Sie über der obigen Liste der Protokollereignisse die Option Text.
   - Um die Protokollereignisse zu filtern, geben Sie den gewünschten Suchfilter in das Suchfeld ein. Weitere Informationen finden Sie im Erstellen von Metriken aus Protokollereignissen mithilfe von Filtern (p. 74).
   - Um Protokolldaten für einen bestimmten Datums- und Zeitbereich anzuzeigen, wählen Sie neben dem Suchfilter den Pfeil neben Datum und Uhrzeit aus. Um einen Datums- und Zeitbereich festzulegen, wählen Sie Absolute (Absolut) aus. Um eine vordefinierte Anzahl von Minuten, Stunden, Tagen oder Wochen auszuwählen, wählen Sie Relative (Relativ) aus. Sie können auch zwischen UTC und lokaler Zeitzone wechseln.

Suchen von Protokolldaten mithilfe von Filtermustern

Sie können Ihre Protokolldaten mithilfe der Filter- und Mustersyntax (p. 75) suchen. Sie können alle Log-Streams in einer Protokollgruppe suchen, oder mit der AWS CLI auch spezifische Log-Streams suchen. Wenn die Suche ausgeführt wird, kehrt sie zur ersten Daten der gefunden Daten und einem Token zurück, um die nächste Datenseite aufzurufen oder die Suche fortzusetzen. Wenn keine Ergebnisse zurückgegeben werden, können Sie die Suche fortsetzen.

Sie können den Zeitbereich einschränken, den Sie abfragen möchten, und damit den Umfang der Suche reduzieren. Sie können mit einem größeren Bereich beginnen, um zu erkennen, ob die Protokollzeilen, die Sie interessieren, dazu gehören. Dann können Sie den Zeitraum auf diesen Umfang verkürzen, um die Protokolle für den Zeitbereich anzuzeigen, der für Sie von Interesse ist.

Sie können auch direkt von den aus den Protokollen extrahierten Metriken zu den entsprechenden Protokolle wechseln.

Suchen von Protokolleinträge mithilfe der Konsole

Sie können mithilfe der Konsole nach den Protokolleinträgen suchen, die ein bestimmtes Kriterien erfüllen.

So suchen Sie Ihre Protokolle mithilfe der Konsole

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus.
3. Wählen Sie für Log Groups den Namen der Protokollgruppe mit dem Protokoll-Stream aus, nach dem gesucht werden soll.

4. Wählen Sie für Log Streams den Namen des zu suchenden Protokoll-Streams.
5. Geben Sie unter Protokollereignisse die zu verwendende Filtersyntax ein.

So suchen Sie alle Protokolleinträge für einen Zeitbereich mithilfe der Konsole

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus.
3. Wählen Sie für Log Groups den Namen der Protokollgruppe mit dem Protokoll-Stream aus, nach dem gesucht werden soll.
4. Wählen Sie Protokollgruppe suchen aus.
5. Wählen Sie für Protokollereignisse den Datums- und Uhrzeitbereich aus und geben Sie die Filtersyntax ein.

Suchen von Protokolleinträgen mithilfe der AWS CLI

Sie können mithilfe der AWS CLI nach den Protokolleinträgen suchen, die ein bestimmtes Kriterium erfüllen.

So suchen Sie mithilfe der AWS CLI nach Protokolleinträgen

Führen Sie an der Eingabeaufforderung den folgenden filter-log-events-Befehl aus. Mit `--filter-pattern` begrenzen Sie die Ergebnisse auf das angegebene Filtermuster, und mit `--log-stream-names` begrenzen Sie die Ergebnisse auf die angegebene Protokollgruppe.

```
aws logs filter-log-events --log-group-name my-group [--log-stream-
names LIST_OF_STREAMS_TO_SEARCH] --filter-pattern VALID_METRIC_FILTER_PATTERN]
```

So suchen Sie Protokolleinträge in einem bestimmten Zeitraum mithilfe der AWS CLI

Führen Sie an der Eingabeaufforderung den folgenden filter-log-events-Befehl aus.

```
aws logs filter-log-events --log-group-name my-group [--log-stream-
names LIST_OF_STREAMS_TO_SEARCH] [--start-time 1482197400000] [--end-time 1482217558365]
 [--filter-pattern VALID_METRIC_FILTER_PATTERN]
```

Wechseln von Metriken zu Protokollen

Sie können aus anderen Teilen der Konsole zu bestimmten Protokolleinträge wechseln.

So rufen Sie Protokolle von Dashboard-Widgets aus auf

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Dashboards aus.
3. Wählen Sie ein Dashboard.
4. Wählen Sie im Widget das Symbol zum Anzeigen der Protokolle und dann die Option View logs in this time range. Wenn mehrere Metrikfilter vorhanden sind, wählen Sie einen Filter aus der Liste aus. Wenn mehr Metrikfilter vorhanden sind, als in der Liste angezeigt werden, wählen Sie More metric filters aus und wählen Sie einen Metrikfilter aus oder suchen Sie danach.

So rufen Sie Protokolle aus Metriken ab

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.

2. Wählen Sie im Navigationsbereich Metrics aus.
3. Geben Sie in das Suchfeld auf der Registerkarte All metrics den Namen der Metrik ein und drücken Sie die Eingabetaste.
4. Wählen Sie eine oder mehrere Metriken aus den Suchergebnissen aus.
5. Wählen Sie Actions und View logs aus. Wenn mehrere Metrikfilter vorhanden sind, wählen Sie einen Filter aus der Liste aus. Wenn mehr Metrikfilter vorhanden sind, als in der Liste angezeigt werden, wählen Sie More metric filters aus und wählen Sie einen Metrikfilter aus oder suchen Sie danach.

Troubleshooting

Suche dauert zu lange

Wenn viele Protokolldaten vorhanden sind, kann die Suche sehr lange dauern. Gehen Sie wie folgt vor, um die Suche zu beschleunigen:

- Wenn Sie die AWS CLI verwenden, können Sie die Suche auf ganz bestimmte Protokoll-Streams einschränken. Wenn in der Protokollgruppe beispielsweise 1.000 Protokoll-Streams vorhanden sind, Sie aber Sie nur drei Protokoll-Streams anzeigen möchten, von denen Sie wissen, dass sie wichtig sind, können Sie unter Verwendung der AWS CLI die Suche auf nur diese drei Protokoll-Streams in der Protokollgruppe begrenzen.
- Verwenden Sie einen kürzeren, individuelleren Zeitraum, wodurch die Menge der durchsuchten Daten reduziert und die Abfrage beschleunigt werden.

Ändern der Aufbewahrung von Protokolldaten in CloudWatch Logs

Standardmäßig werden Daten in CloudWatch Logs für eine unbegrenzte Dauer gespeichert. Sie können jedoch konfigurieren, wie lange Protokolldaten in einer Protokollgruppe gespeichert werden sollen. Alle Daten, die älter als die aktuelle Aufbewahrungseinstellung sind, werden automatisch gelöscht. Sie können die Protokoll-Aufbewahrung für jede Protokollgruppe jederzeit ändern.

So ändern Sie die Einstellungen für die Aufbewahrung der Protokolldateien

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus.
3. Gehen Sie zur Protokollgruppe, die aktualisiert werden soll.
4. Wählen Sie in der Spalte Expire Events After für diese Protokollgruppe die aktuelle Einstellung für die Aufbewahrung, wie z. B. Never Expire.
5. Wählen Sie im Dialogfeld Edit Retention (Aufbewahrung bearbeiten) in Retention (Aufbewahrung) einen Wert für die Protokollaufbewahrung aus. Wählen Sie anschließend OK aus.

Protokollgruppen in Amazon CloudWatch Logs kennzeichnen

Sie können den Protokollgruppen, die Sie in Amazon CloudWatch Logs erstellen, eigene Metadaten in Form von Tags zuweisen. Ein Tag ist ein Schlüssel-Wert-Paar, das Sie für eine Protokollgruppe definieren.

Die Verwendung von Tags ist ein einfacher, aber effizienter Weg, um AWS-Ressourcen zu verwalten und Daten, einschließlich Fakturierungsdaten, zu organisieren.

Inhalt:

- Grundlagen zu Tags (p. 65)
- Kosten mithilfe von Tags verfolgen (p. 65)
- Einschränkungen für Tags (p. 65)
- Protokollgruppen mithilfe der AWS CLI mit Tags kennzeichnen (p. 66)
- Protokollgruppen mithilfe der CloudWatch Logs-API mit Tags kennzeichnen (p. 66)

Grundlagen zu Tags

Sie können die AWS CLI- oder CloudWatch Logs-API verwenden, um die folgenden Aufgaben auszuführen:

- Hinzufügen von Tags zu einer Protokollgruppe während der Erstellung
- Hinzufügen von Tags zu einer vorhandenen Protokollgruppe
- Auflistung der Tags für eine Protokollgruppe
- Entfernung der Tags von einer Protokollgruppe

Sie können Tags verwenden, um Ihre Protokollgruppen zu kategorisieren. Sie können sie beispielsweise nach Zweck, Inhaber oder Umgebung kategorisieren. Da Sie für jeden Tag den Schlüssel und Wert definieren, können Sie eine auf benutzerdefinierte Reihe von Kategorien anlegen, die Ihren jeweiligen Anforderungen gerecht wird. Sie könnten zum Beispiel eine Reihe von Tags definieren, mit der Sie Protokollgruppen nach Inhaber und zugehöriger Anwendung nachverfolgen können. Im Folgenden finden Sie einige Beispiele für Tags:

- Projekt Project name
- Eigentümer Name
- Zweck Lasttest
- Anwendung Anwendungsname.
- Umgebung Produktion

Kosten mithilfe von Tags verfolgen

Sie können Tags verwenden, um Ihre AWS-Kosten zu kategorisieren und zu verfolgen. Wenn Sie Tags auf AWS-Ressourcen, einschließlich Protokollgruppen, anwenden, enthält der AWS-Kostenzuordnungsbericht mit Tags aggregierte Nutzungs- und Kostendaten. Sie können Tags anwenden, die geschäftliche Kategorien (wie Kostenstellen, Anwendungsnamen oder Eigentümer) darstellen, um die Kosten für mehrere Services zu organisieren. Weitere Informationen finden Sie unter Verwenden von Kostenzuordnungs-Tags für benutzerdefinierte Fakturierungsberichte im Benutzerhandbuch für AWS Billing and Cost Management.

Einschränkungen für Tags

Für Tags gelten die folgenden Einschränkungen:

Grundlegende Einschränkungen

- Die maximale Anzahl von Tags pro Protokollgruppe beträgt 50.

- Bei Tag-Schlüsseln und -Werten muss die Groß-/Kleinschreibung beachtet werden.
- Sie können Tags für eine gelöschte Protokollgruppe nicht ändern oder bearbeiten.

Einschränkungen für Tag-Schlüssel

- Jeder Tag-Schlüssel muss einmalig sein. Wenn Sie einen Tag mit einem Schlüssel hinzufügen, der bereits verwendet wird, wird das vorhandene Schlüssel-Wert-Paar durch den neuen Tag überschrieben.
- Sie können einen Tag-Schlüssel nicht mit `aws:` starten, da dieses Präfix für die Verwendung durch AWS reserviert ist. AWS erstellt zwar in Ihrem Namen Tags, die mit diesem Präfix beginnen, Sie können diese jedoch nicht bearbeiten oder löschen.
- Tag-Schlüssel müssen zwischen 1 und 128 Unicode-Zeichen lang sein.
- Tag-Schlüssel müssen aus den folgenden Zeichen bestehen: Unicode-Buchstaben, Ziffern, Leerzeichen und die folgenden Sonderzeichen: `_ . / = + - @`.

Einschränkungen für den Tag-Wert:

- Tag-Werte müssen zwischen 0 und 255 Unicode-Zeichen lang sein.
- Tag-Werte können leer sein. Andernfalls müssen sie aus den folgenden Zeichen bestehen: Unicode-Buchstaben, Ziffern, Leerzeichen und jede der folgenden Sonderzeichen: `_ . / = + - @`.

Protokollgruppen mithilfe der AWS CLI mit Tags kennzeichnen

Sie können Tags mithilfe der AWS CLI hinzufügen, auflisten und entfernen. Beispiele finden Sie in der folgenden Dokumentation:

create-log-group

Erstellt eine Protokollgruppe. Sie können beim Erstellen einer Protokollgruppe optional Tags hinzufügen.

tag-log-group

Fügt Tags für die angegebene Protokollgruppe hinzu oder aktualisiert diese.

list-tags-log-group

Führ die Tags für die angegebene Protokollgruppe auf

untag-log-group

Entfernt Tags von der angegebenen Protokollgruppe.

Protokollgruppen mithilfe der CloudWatch Logs-API mit Tags kennzeichnen

Sie können Tags mithilfe der CloudWatch Logs-API hinzufügen, auflisten und entfernen. Beispiele finden Sie in der folgenden Dokumentation:

CreateLogGroup

Erstellt eine Protokollgruppe. Sie können beim Erstellen einer Protokollgruppe optional Tags hinzufügen.

TagLogGroup

Fügt Tags für die angegebene Protokollgruppe hinzu oder aktualisiert diese.

ListTagsLogGroup

Führ die Tags für die angegebene Protokollgruppe auf

UntagLogGroup

Entfernt Tags von der angegebenen Protokollgruppe.

Protokolldaten in CloudWatch Logs mithilfe von AWS KMS verschlüsseln

Protokollgruppendaten werden in CloudWatch Logs immer verschlüsselt. Optional können Sie AWS AWS Key Management Service für diese Verschlüsselung verwenden. Wenn Sie dies tun, erfolgt die Verschlüsselung mit einem AWS KMS (AWS KMS) Customer Master Key (CMK). Die Verschlüsselung mit AWS KMS wird auf der Ebene der Protokollgruppe aktiviert, indem ein CMK entweder beim Erstellen der Protokollgruppe mit dieser verknüpft wird oder nachdem diese vorhanden ist.

Important

CloudWatch Logs unterstützt jetzt den Verschlüsselungskontext, wobei `kms:EncryptionContext:aws:logs:arn` als Schlüssel und der ARN der Protokollgruppe als Wert für den Schlüssel verwendet werden. Wenn es Protokollgruppen gibt, die Sie bereits mit einem CMK verschlüsselt haben, und Sie die Verwendung des CMK auf ein einzelnes Konto und eine einzelne Protokollgruppe einschränken möchten, sollten Sie einen neuen CMK zuweisen, der in der IAM-Richtlinie eine Bedingung enthält. Weitere Informationen finden Sie im KMS-Schlüssel und Verschlüsselungskontext (p. 70).

Nachdem Sie ein CMK mit einer Protokollgruppe verknüpft haben, werden alle für die Protokollgruppe neu übernommenen Daten mithilfe des CMK verschlüsselt. Diese Daten werden während des gesamten Aufbewahrungszeitraums im verschlüsselten Format gespeichert. CloudWatch Logs entschlüsselt diese Daten, sobald sie angefordert werden. CloudWatch Logs muss über Berechtigungen für den CMK verfügen, sobald verschlüsselte Daten angefordert werden.

Nachdem Sie die Verknüpfung eines CMK mit einer Protokollgruppe aufgehoben haben, beendet CloudWatch Logs die Verschlüsselung von neu aufgenommenen Daten für die Protokollgruppe. Alle zuvor übernommenen Daten bleiben verschlüsselt.

Important

CloudWatch Logs unterstützt nur symmetrische CMKs. Verwenden Sie kein asymmetrisches CMK, um die Daten in Ihren Protokollgruppen zu verschlüsseln. Weitere Informationen finden Sie unter Verwendung symmetrischer und asymmetrischer Schlüssel.

Limits

- Um die folgenden Schritte auszuführen, müssen Sie die folgenden Berechtigungen haben: `kms:CreateKey`, `kms:GetKeyPolicy`, und `kms:PutKeyPolicy`.
- Nachdem Sie die Verknüpfung eines CMK mit einer Protokollgruppe hergestellt oder aufgehoben haben, kann es bis zu fünf Minuten dauern, bis der Vorgang wirksam wird.
- Wenn Sie den CloudWatch Logs-Zugriff auf einen verknüpften CMK aufheben oder einen verknüpften CMK löschen, können Ihre verschlüsselten Daten in CloudWatch Logs nicht mehr abgerufen werden.
- Sie können einen CMK nicht mithilfe der CloudWatch-Konsole mit einer Protokollgruppe verknüpfen.

Schritt 1 Erstellen AWS KMS CMK

Um einen AWS KMS-CMK zu erstellen, verwenden Sie den folgenden Befehl create-key:

```
aws kms create-key
```

Die Ausgabe enthält die Schlüssel-ID und den Amazon Resource Name (ARN) des CMK. Im Folgenden finden Sie eine Beispielausgabe.

```
{
    "KeyMetadata": {
        "KeyId": "6f815f63-e628-448c-8251-e40cb0d29f59",
        "Description": "",
        "Enabled": true,
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1478910250.94,
        "Arn": "arn:aws:kms:us-west-2:123456789012:key/6f815f63-e628-448c-8251-
e40cb0d29f59",
        "AWSAccountId": "123456789012"
    }
}
```

Schritt 2 Berechtigungen auf dem CMK einstellen

Standardmäßig sind alle AWS KMS-CMKs privat. Nur der Ressourcenbesitzer kann mit ihnen Daten verschlüsseln und entschlüsseln. Der Ressourceninhaber kann jedoch anderen Benutzern und Ressourcen Zugriffsberechtigungen für den CMK erteilen. Mit diesem Schritt erteilen Sie dem CloudWatch-Service-Prinzipal Berechtigung für die Nutzung des CMK. Dieser Service-Prinzipal muss sich in der AWS-Region befinden, in der der CMK gespeichert ist.

Als bewährte Methode sollten Sie die Verwendung des Schlüssels auf die von Ihnen angegebenen AWS-Konten oder -Protokollgruppen einschränken.

Speichern Sie zunächst die Standardrichtlinie für Ihren CMK als `policy.json` ab, indem Sie den folgenden Befehl get-key-policy verwenden:

```
aws kms get-key-policy --key-id key-id --policy-name default --output text > ./policy.json
```

Öffnen Sie die Datei `policy.json` in einem Texteditor und fügen Sie den in Fettschrift angezeigten Abschnitt aus einer der folgenden Anweisungen hinzu. Sie trennen die vorhandene Anweisung von der neuen Anweisung durch ein Komma. Diese Anweisungen verwenden `Condition`-Abschnitte, um die Sicherheit des KMS-Schlüssels zu optimieren. Weitere Informationen finden Sie im KMS-Schlüssel und Verschlüsselungskontext (p. 70).

Der `Condition`-Abschnitt in diesem Beispiel schränkt den Schlüssel auf einen einzelnen Protokollgruppen-ARN ein.

```
{
 "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Your_account_ID:root"
            },
```

```
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:region:account-
id:log-group:log-group-name"
                }
            }
        }
    ]
}
```

Der `Condition`-Abschnitt in diesem Beispiel beschränkt die Verwendung des KMS auf das angegebene Konto, kann jedoch für jede Protokollgruppe verwendet werden.

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::Your_account_ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:region:account-
id:*"
                }
            }
        }
    ]
}
```

Fügen Sie abschließend die aktualisierte Richtlinie hinzu, indem Sie folgenden Befehl put-key-policy verwenden:

```
aws kms put-key-policy --key-id key-id --policy-name default --policy file://policy.json
```

Schritt 3 Eine Protokollgruppe mit einem CMK verknüpfen

Sie können einen CMK während oder nach der Erstellung einer Protokollgruppe mit dieser verknüpfen.

Mit dem folgenden describe-log-groups-Befehl können Sie ermitteln, ob einer Protokollgruppe bereits ein CMK zugeordnet ist:

```
aws logs describe-log-groups --log-group-name-prefix "log-group-name-prefix"
```

Wenn die Ausgabe ein `kmsKeyId`-Feld enthält, wird die Protokollgruppe dem Schlüssel zugeordnet, der für den Wert dieses Feldes angezeigt wird.

So verknüpfen Sie einen CMK mit einer Protokollgruppe, wenn Sie sie erstellen

Verwenden Sie den Befehl create-log-group wie folgt:

```
aws logs create-log-group --log-group-name my-log-group --kms-key-id "key-arn"
```

So verknüpfen Sie den CMK mit einer vorhandenen Protokollgruppe

Verwenden Sie den Befehl associate-kms-key wie folgt:

```
aws logs associate-kms-key --log-group-name my-log-group --kms-key-id "key-arn"
```

Schritt 4. Eine Protokollgruppe von einem CMK trennen

Um die Verknüpfung eines CMK mit einer Protokollgruppe aufzuheben, verwenden Sie den folgenden Befehl disassociate-kms-key:

```
aws logs disassociate-kms-key --log-group-name my-log-group
```

KMS-Schlüssel und Verschlüsselungskontext

Um die Sicherheit Ihrer KMS-Schlüssel und Ihrer verschlüsselten Protokollgruppen zu erhöhen, verwendet CloudWatch Logs jetzt Protokollgruppen-ARNs als Teil des Verschlüsselungskontexts, der für die Verschlüsselung Ihrer Protokolldaten verwendet wird. Der Verschlüsselungskontext ist ein Satz von Schlüssel-Wert-Paaren, die als zusätzliche authentifizierte Daten verwendet werden. Im Verschlüsselungskontext können Sie IAM-Richtlinienbedingungen verwenden, um den Zugriff auf Ihren KMS-Schlüssel nach AWS-Konto und Protokollgruppe zu begrenzen. Weitere Informationen finden Sie unter Verschlüsselungskontext und IAM JSON-Richtlinienelemente: Bedingung

Sie sollten für jede verschlüsselte Protokollgruppe einen anderen CMK-Schlüssel verwenden.

Wenn Sie zuvor eine Protokollgruppe verschlüsselt haben und jetzt für diese Protokollgruppe einen neuen CMK verwenden möchten, der nur für diese Protokollgruppe funktioniert, gehen Sie folgendermaßen vor.

So konvertieren Sie eine verschlüsselte Protokollgruppe für die Verwendung eines CMK mit einer Richtlinie, die diesen auf diese Protokollgruppe einschränkt

1. Geben Sie den folgenden Befehl ein, um den ARN des aktuellen CMK der Protokollgruppe zu finden:

```
aws logs describe-log-groups
```

   Die Ausgabe enthält die folgende Zeile. Notieren Sie den ARN. Sie benötigen ihn in Schritt 7.

```
...
"kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/01234567-89ab-
cdef-0123-456789abcdef"
...
```

2. Geben Sie den folgenden Befehl ein, um einen neuen CMK zu erstellen.

```
aws kms create-key
```

3. Geben Sie den folgenden Befehl ein, um die Richtlinie des neuen Schlüssels in einer `policy.json`-Datei zu speichern:

```
aws kms get-key-policy --key-id new-key-id --policy-name default --output text > ./
policy.json
```

4. Verwenden Sie einen Texteditor zum Öffnen von `policy.json` und fügen Sie der Richtlinie einen `Condition`-Ausdruck hinzu:

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::ACCOUNT-ID:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:Decrypt*",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:Describe*"
            ],
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "kms:EncryptionContext:aws:logs:arn": "arn:aws:logs:REGION:ACCOUNT-
ID:log-
group:LOG-GROUP-NAME"
                }
            }
        }
```

```
    ]
}
```

5. Geben Sie den folgenden Befehl ein, um dem neuen CMK die aktualisierte Richtlinie hinzuzufügen:

```
aws kms put-key-policy --key-id new-key-ARN --policy-name default --policy file://
policy.json
```

6. Geben Sie den folgenden Befehl ein, um die Richtlinie mit Ihrer Protokollgruppe zu verknüpfen:

```
aws logs associate-kms-key --log-group-name my-log-group --kms-key-id new-key-ARN
```

CloudWatch Logs verschlüsselt jetzt alle neuen Daten mit dem neuen CMK.

7. Als Nächstes widerrufen Sie alle Berechtigungen außer `Decrypt` aus dem alten CMK. Geben Sie zunächst den folgenden Befehl ein, um die alte Richtlinie abzurufen:

```
aws kms get-key-policy --key-id old-key-ARN --policy-name default --output text > ./
policy.json
```

8. Verwenden Sie einen Texteditor, um `Action` zu öffnen und alle Werte aus der Liste `policy.json` außer `kms:Decrypt*` zu entfernen.

```
{
    "Version": "2012-10-17",
    "Id": "key-default-1",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::REGION:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "logs.region.amazonaws.com"
            },
            "Action": [
                "kms:Decrypt*"
            ],
            "Resource": "*"
        }
    ]
}
```

9. Geben Sie den folgenden Befehl ein, um dem alten CMK die aktualisierte Richtlinie hinzuzufügen:

```
aws kms put-key-policy --key-id old-key-ARN --policy-name default --policy file://
policy.json
```

Aktivieren der Protokollierung von bestimmten AWS-Services aus

Wenn Sie Protokolle von einigen AWS-Services aus an CloudWatch Logs senden möchten, müssen Sie eine CloudWatch Logs-Ressourcenrichtlinie verwenden oder erstellen, die dem Service die Berechtigung zum Senden ihrer Protokolle an CloudWatch Logs erteilt. Dieses Problem kann sich auf Amazon API Gateway AWS Step Functions und Amazon Managed Streaming for Apache Kafka auswirken.

Diese Services müssen jede Protokollgruppe, die sie senden, in der Ressourcenrichtlinie auflisten und CloudWatch Logs-Ressourcenrichtlinien sind auf 5120 Zeichen beschränkt. Daher kann ein Service, der Protokolle an eine große Anzahl von Protokollgruppen sendet, durchaus diesen Grenzwert erreichen.

Um dies zu minimieren, überwacht CloudWatch Logs die Größe der Ressourcenrichtlinien, die von anderen AWS-Services verwendet werden, und wenn festgestellt wird, dass eine Richtlinie sich der Größenbeschränkung von 5120 Zeichen nähert, aktiviert CloudWatch Logs automatisch `/aws/vendedlogs/*` in der Ressourcenrichtlinie für diesen Service. Sie können dann mit dem Erstellen von Protokollabonnements beginnen, indem Sie Protokollgruppen für diese Services verwenden, deren Namen mit `/aws/vendedlogs/` beginnen.

Erstellen von Metriken aus Protokollereignissen mithilfe von Filtern

Wenn der CloudWatch Logs-Agent mit der Veröffentlichung von Protokolldaten in Amazon CloudWatch beginnt, können Sie mit der Suche und dem Filtern von Protokolldaten beginnen, indem Sie einen oder mehrere Metrikfilter erstellen. Metrikfilter definieren die Bedingungen und Muster, nach denen in den Protokolldateien gesucht wird, wenn sie an CloudWatch Logs gesendet werden. CloudWatch Logs verwendet diese Metrikfilter, um Protokolldateien in numerische CloudWatch-Metriken umzuwandeln, die Sie als Diagramm darstellen, oder für die Sie einen Alarm festlegen können. Bei der Ansicht dieser Metriken oder beim Festlegen von Alarmen können Sie jede Art von CloudWatch-Statistik, einschließlich Perzentil-Statistiken, verwenden

> **Note**
>
> Perzentil-Statistiken werden für eine Metrik nur dann unterstützt, wenn keiner der Metrikwerte negativ ist. Wenn Sie Ihren Metrikfilter so einrichten, dass er negative Zahlen melden kann, stehen für diese Metrik keine Perzentilstatistiken zur Verfügung, wenn negative Zahlen als Werte angezeigt werden. Weitere Informationen finden Sie unter Perzentile.

Filter können nicht rückwirkend auf Daten angewendet werden. Filter veröffentlichen nur die Metrikdatenpunkte für Ereignisse, die aufgetreten sind, nachdem der Filter erstellt wurde. Gefilterte Ergebnisse geben die ersten 50 Zeilen zurück, die nicht angezeigt werden, wenn der Zeitstempel für die gefilterten Ergebnisse vor der Metrikerstellung liegt.

Inhalt:

- Concepts (p. 74)
- Filter- und Mustersyntax (p. 75)
- Erstellen von Metrikfiltern (p. 83)
- Auflisten von Metrikfiltern (p. 89)
- Löschen eines Metrikfilters (p. 89)

Concepts

Jeder Metrikfilter besteht aus den folgenden Schlüsselelementen:

Filtermuster

> Eine symbolische Beschreibung dazu, wie CloudWatch Logs die Daten in den einzelnen Protokollereignissen interpretiert werden soll. Ein Protokolleintrag enthält z. B. möglicherweise Zeitstempel, IP-Adressen, Zeichenfolgen und so weiter. Mit dem Muster können Sie angeben, wonach in der Protokolldatei gesucht werden soll.

Metrikname

> Der Name der CloudWatch-Metrik, an die die überwachten Protokollinformationen veröffentlicht werden. Sie können beispielsweise eine Metrik mit dem Namen "ErrorCount" veröffentlichen.

Namespace der Metrik

Der Ziel-Namespace der neuen CloudWatch-Metrik.

Metrikwert

Der numerische Wert, der jedes Mal für die Metrik veröffentlicht werden soll, wenn ein übereinstimmendes Protokoll gefunden wird. Wenn Sie beispielsweise die Häufigkeit des Vorkommens eines bestimmten Begriffs wie "Fehler" zählen, lautet der Wert für jedes Vorkommen "1". Wenn Sie die übertragenen Bytes zählen, können Sie die Zahl um die tatsächliche Anzahl der Bytes, die im Protokollereignis gefunden wurden, erhöhen.

Standardwert

Der Wert der während eines Zeitraums an das Metrik-Filter gemeldet wird, wenn keine passenden Protokolle gefunden wurden. Wenn Sie diesen auf 0 setzen, stellen Sie sicher, dass die Daten in jedem Zeitraum gemeldet werden und verhindern „lückenhafte“ Metriken mit Zeiträumen, die keine Daten enthalten.

Filter- und Mustersyntax

Mit Metrikfiltern können Sie in den Protokollereignissen nach übereinstimmenden Begriffen, Ausdrücken oder Werten suchen. Wenn ein Metrikfilter einen der Begriffe, Ausdrücke oder Werte in den Protokollereignissen findet, können Sie den Wert der CloudWatch-Metrik erhöhen. So können Sie z. B. einen Metrikfilter zum Suchen und Zählen des Vorkommens des Wortes FEHLER in den Protokollereignissen erstellen.

Metrikfilter können auch numerische Werte aus Protokollereignissen extrahieren, deren Platz begrenzt ist (z. B. die Latenz von Webanfragen). In diesen Beispielen können Sie den metrischen Wert um den tatsächlichen numerischen Wert erhöhen, der aus dem Protokoll extrahiert wurde.

Darüber hinaus können Sie Bedingungsoperatoren und Platzhalter verwenden, um exakte Übereinstimmungen zu erzeugen. Bevor Sie einen Metrikfilter erstellen, können Sie die Suchmuster in der CloudWatch-Konsole testen. In den folgenden Abschnitten wird die Syntax für Metrikfilter detailliert erläutert.

Auffinden von Begriffen in Protokollereignissen

Wenn Sie in den Protokollereignissen nach einem Begriff suchen möchten, verwenden Sie den Begriff als Metrikfiltermuster. Sie können mehrere Begriffe in einem Metrikfiltermuster angeben, aber alle Begriffe müssen in einem Protokollereignis vorhanden sein, damit eine Übereinstimmung gefunden wird. Bei Metrikfiltern sind Groß- und Kleinschreibung zu beachten.

Metrikfilterbegriff, die andere als alphanumerische Zeichen oder Unterstriche enthalten, müssen in Anführungszeichen ("") gesetzt werden.

Zum Ausschließen eines Begriffs verwenden Sie ein Minuszeichen (-) vor dem Begriff.

Beispiel 1 Alles Passende

Das Filtermuster "" nimmt eine Übereinstimmung mit allen Protokollereignissen vor.

Beispiel 2 Einzelbegriff

Der Filtermuster "FEHLER" findet Protokollereignisnachrichten, die diesen Begriff enthalten, wie z. B. die folgenden:

- [FEHLER] Eine schwerwiegende Ausnahme ist aufgetreten
- Beenden mit ERRORCODE: 1.

Beispiel 3 Einen Begriff einschließen und einen Begriff ausschließen

Wenn Sie im vorherigen Beispiel das Filtermuster in "FEHLER" - "Beenden" geändert haben, wird die Protokollereignisnachricht "Beenden mit FEHLERCODE: -1" ausgeschlossen.

Beispiel 4 Mehrere Begriffe

Das Filtermuster "FEHLER Ausnahme" findet Protokollereignisnachrichten, die beide Begriffe enthalten, wie z. B. die folgenden:

- [FEHLER] Ungültig Argumentausnahme gefunden
- [FEHLER] Unbehandelte Ausnahme

Das Filtermuster "Fehler beim Verarbeiten der Anforderung" findet Protokollereignisnachrichten, die alle Begriffe enthalten, z. B. die folgenden:

- [WARNUNG] Fehler beim Verarbeiten der Anforderung
- [FEHLER] Fortsetzung: Fehler beim Verarbeiten der Anfrage

ODER-Mustervergleich

Sie können Begriffe in textbasierten Filtern mithilfe des ODER-Mustervergleichs verbinden. Verwenden Sie ein Fragezeichen für OR, wie z. B. `?term`.

Schauen Sie sich die drei Beispiele für die Ereignisereignisveranstaltung an. `ERROR` entspricht Beispiele 1 und 2. `?ERROR ?WARN` entspricht den Beispielen 1, 2 und 3, da sie entweder das Wort ERROR (FEHLER) oder das Wort WARNING (Warnung) beinhalten. `ERROR WARN` entspricht nur Beispiel 1, da es sich dabei um die einzigen Wörter handelt. ERROR -WARN entspricht Beispiel 2, da es mit einer Zeichenfolge übereinstimmt, die ERROR, nicht jedoch WARN enthält.

1. `ERROR WARN message`
2. `ERROR message`
3. `WARN message`

Sie können Begriffe mithilfe des ODER-Mustervergleichs in durch Leerzeichen getrennten Filtern abgleichen. Mit durch Leerzeichen getrennten Filtern steht `w1` für das erste Wort im Protokollereignis, `w2` für das zweite Wort und so weiter. Für die folgenden Beispielmuster stimmt [w1=ERROR, w2] mit Muster 2 überein, da ERROR das erste Wort ist, und [w1=ERROR || w1=WARN, w2] stimmt mit Muster 2 und 3 überein. [w1!=ERROR&&w1!=WARN, w2] findet Zeilen, die sowohl ERROR als auch WARN enthalten (Muster 1).

1. FEHLER WARNUNG-Mitteilung
2. FEHLER-Mitteilung
3. WARNUNG-Mitteilung

Sie können Begriffe mithilfe des ODER-Mustervergleichs in JSON-Filtern abgleichen. Für das Beispielmuster unten: {$.foo = bar} findet Muster 1, {$.foo = baz } findet Muster 2 und {$.foo = bar || $.foo = baz } findet die Muster 1 und 2.

1. {"foo": "bar"}

2. {"foo": "baz"}

Auffinden von Begriffen in JSON-Protokollereignissen

Sie können Werte aus JSON-Protokollereignissen extrahieren. Wenn Sie Werte aus JSON-Protokollereignissen extrahieren möchten, müssen Sie einen Metrikfilter für Zeichenfolgen erstellen. Zeichenfolgen, die wissenschaftliche Notationen enthalten, werden nicht unterstützt. Die Elemente in den JSON-Protokollereignisdaten müssen genau mit dem Metrikfilter übereinstimmen. Sie sollten Metrikfilter in JSON-Protokollereignissen erstellen, um Folgendes anzuzeigen:

- Ein bestimmtes Ereignis tritt ein. Beispielsweise ist der Ereignisname "UpdateTrail".
- Die IP ist außerhalb von einem bekannten Subnetz. Beispielsweise befindet sich sourceIPAddress nicht in einem bekannten Subnetz.
- Eine Kombination aus zwei oder mehr anderen Bedingungen ist erfüllt. Beispielsweise ist der Ereignisname "UpdateTrail" und die recipientAccountId ist 123456789012.

Verwenden von Metrikfiltern zum Extrahieren von Werten aus JSON-Protokollereignissen

Mit Metrikfiltern können Sie Werte aus JSON-Protokollereignissen extrahieren. Ein Metrikfilter überprüft eingehende Protokolle und ändert den numerische Wert, wenn der Filter eine Übereinstimmung in der Protokolldaten findet. Wenn Sie ein Metrikfilter erstellen, können Sie einfach jedes Mal die Zählung erhöhen, wenn in einem Protokoll ein übereinstimmender Text gefunden wurde, oder Sie können die numerischen Werte aus dem Protokoll extrahieren und diese Zunahme zur Erhöhung des Metrikwerts verwenden.

Die Zuordnung von JSON-Begriffen mithilfe von Metrikfiltern

Die Syntax für Metrikfilter für JSON-Protokollereignisse verwendet das folgende Format:

```
{ SELECTOR EQUALITY_OPERATOR STRING }
```

Der Metrikfilter muss in geschweiften Klammern {} stehen, damit deutlich wird, dass es sich um einen JSON-Ausdruck handelt. Der Metrikfilter enthält die folgenden Teile:

SELECTOR

Gibt an, welche JSON-Eigenschaft zu prüfen ist. Eigenschaftenselektoren beginnen immer mit dem Dollarzeichen ($), das den Stamm der JSON angibt. Eigenschaftenselektoren sind alphanumerische Zeichenfolgen, die auch die Zeichen "-" und "_" unterstützen. Array-Elemente werden mit [NUMBER]-Syntax bezeichnet und müssen auf eine Eigenschaft folgen. Beispiele: $.eventId, $.users[0], $.users[0].id, $.requestParameters.instanceId.

EQUALITY_OPERATOR

Kann oder sein.

STRING

Eine Zeichenfolge mit oder ohne Anführungszeichen. Sie können das Sternchen "*" als Platzhalterzeichen für einen Text vor oder nach einem Suchbegriff verwenden. Beispiel: *Event entspricht PutEvent und GetEvent. Event* stimmt überein mit EventId und EventName. Ev*ent stimmt nur mit der tatsächlichen Zeichenfolge Ev*ent überein. Zeichenfolgen, die vollständig aus alphanumerischen Zeichen bestehen, müssen nicht in Anführungszeichen gesetzt werden. Zeichenfolgen mit Unicode- oder anderen Zeichen, wie beispielsweise '@', '$', '\', usw. müssen in doppelte Anführungszeichen eingeschlossen werden, damit sie gültig sind.

Beispiele für JSON-Metrikfilter

Im Folgenden sehen Sie ein JSON-Beispiel:

```
{
  "eventType": "UpdateTrail",
  "sourceIPAddress": "111.111.111.111",
  "arrayKey": [
        "value",
        "another value"
  ],
  "objectList": [
       {
         "name": "a",
         "id": 1
       },
       {
         "name": "b",
         "id": 2
       }
  ],
  "SomeObject": null,
  "ThisFlag": true
}
```

Die folgenden Filter würden Folgendem entsprechen:

```
{ $.eventType = "UpdateTrail" }
```

Filter für den Ereignistyp UpdateTrail.

```
{ $.sourceIPAddress != 123.123.* }
```

Filter für die IP-Adresse außerhalb des Subnetzpräfixes 123.123.

```
{ $.arrayKey[0] = "value" }
```

Filter für den ersten Eintrag im Array-Schlüssel "Wert". Wenn der Array-Schlüssel kein Array ist, ist die Aussage falsch.

```
{ $.objectList[1].id = 2 }
```

Filter für den zweiten Eintrag in "objectList" mit der Eigenschaft "id = 2". Wenn die Objektliste kein Array ist, ist die Aussage falsch. Wenn die Elemente in Objektliste keine Objekte sind oder keine ID-Eigenschaft haben, ist die Aussage falsch.

```
{ $.SomeObject IS NULL }
```

Filter für "SomeObject" mit Null. Dies ist nur wahr, wenn das angegebene Objekt auf Null festgelegt ist.

```
{ $.SomeOtherObject NOT EXISTS }
```

Filter für "SomeOtherObject", das nicht vorhanden ist. Wird nur wahr, wenn das angegebene Objekt nicht in den Protokolldaten vorhanden ist.

```
{ $.ThisFlag IS TRUE }
```

Filter für "ThisFlag" mit TRUE. Funktioniert auch für boolesche Filter, die auf den Wert FALSE prüfen.

Zusammengesetzte JSON-Bedingungen

Sie können mehrere Bedingungen in einem zusammengesetzten Ausdruck mit OR (| |) und AND (& &) kombinieren. Klammer sind zulässig, und die Syntax folgt der Standardreihenfolge für Operationen () > & & > ||.

```
{
    "user": {
        "id": 1,
        "email": "John.Stiles@example.com"
    },
    "users": [
        {
         "id": 2,
         "email": "John.Doe@example.com"
        },
        {
         "id": 3,
         "email": "Jane.Doe@example.com"
        }
    ],
    "actions": [
        "GET",
        "PUT",
        "DELETE"
    ],
    "coordinates": [
        [0, 1, 2],
        [4, 5, 6],
        [7, 8, 9]
    ]
}
```

Examples

```
{ ($.user.id = 1) && ($.users[0].email = "John.Doe@example.com") }
```

Entspricht der oben genannte JSON.

```
{ ($.user.id = 2 && $.users[0].email = "nonmatch") || $.actions[2] = "GET" }
```

Keine Übereinstimmung mit der oben genannten JSON.

```
{ $.user.email = "John.Stiles@example.com" || $.coordinates[0][1] = nonmatch &&
 $.actions[2] = nomatch }
```

Entspricht der oben genannte JSON.

```
{ ($.user.email = "John.Stiles@example.com" || $.coordinates[0][1] = nonmatch) &&
 $.actions[2] = nomatch }
```

Keine Übereinstimmung mit der oben genannten JSON.

Spezielle JSON-Anforderungen

Die SELECTOR muss auf einen Wertknoten (Zeichenfolge oder Nummer) in der JSON verweisen. Wenn er auf ein Array oder Objekt verweist, ist der Filter nicht gültig, weil das Protokollformat nicht dem

Filter entspricht. Beispielsweise stimmen sowohl {$.users = 1} als auch {$.users != 1} nicht mit einem Protokollereignis überein, wenn users das folgenden Array ist:

```
{
  "users": [1, 2, 3]
}
```

Numerische Vergleiche

Die Metrikfiltersyntax unterstützt exakte Übereinstimmung mit numerischen Vergleichen. Die folgenden numerischen Vergleiche werden unterstützt: <, >, >=, <=, =, !=

Numerische Filter haben folgende Syntax

```
{ SELECTOR NUMERIC_OPERATOR NUMBER }
```

Der Metrikfilter muss in geschweiften Klammern {} stehen, damit deutlich wird, dass es sich um einen JSON-Ausdruck handelt. Der Metrikfilter enthält die folgenden Teile:

SELECTOR

Gibt an, welche JSON-Eigenschaft zu prüfen ist. Eigenschaftenselektoren beginnen immer mit dem Dollarzeichen ($), das den Stamm der JSON angibt. Eigenschaftenselektoren sind alphanumerische Zeichenfolgen, die auch die Zeichen "-" und "_" unterstützen. Array-Elemente werden mit [NUMBER]-Syntax bezeichnet und müssen auf eine Eigenschaft folgen. Beispiele: $.latency, $.numbers[0], $.errorCode, $.processes[4].averageRuntime.

NUMERIC_OPERATOR

Dabei kann es sich um einen der folgenden handeln: =, !=, <, >, <= oder >=.

NUMBER

Eine Ganzzahl mit einem optionalen +- oder —Zeichen, eine Dezimalzahl mit einem optionalen +- oder —Zeichen oder eine Zahl in Exponentialschreibweise, bei der es sich um eine ganze Zahl oder eine Dezimalzahl mit einem optionalen +- oder —Zeichen gefolgt von "e", gefolgt von einer Ganzzahl mit einem optionalen +- oder —Zeichen handelt.

Beispiele

```
{ $.latency >= 500 }
{ $.numbers[0] < 10e3 }
{ $.numbers[0] < 10e-3 }
{ $.processes[4].averageRuntime <= 55.5 }
{ $.errorCode = 400 }
{ $.errorCode != 500 }
{ $.latency > +1000 }
```

Verwenden von Metrikfiltern zum Extrahieren von Werten aus durch Leerstellen getrennten Protokollereignissen

Mit Metrikfiltern können Sie Werte aus durch Leerzeichen getrennten Protokollereignisse extrahieren. Die Zeichen zwischen zwei eckige Klammern [] oder zwei Anführungszeichen ("") werden als ein einzelnes Feld behandelt. Beispiel, .

```
127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] "GET /apache_pb.gif HTTP/1.0" 200 1534
127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] "GET /apache_pb.gif HTTP/1.0" 500 5324
```

```
127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] "GET /apache_pb.gif HTTP/1.0" 200 4355
```

Wenn Sie ein Metrikfiltermuster angeben, das durch Leerzeichen getrennte Ereignisse analysiert, müssen in dem Metrikfiltermuster auch die Felder mit einem Namen, getrennt durch Kommata angegeben werden, wobei das gesamte Muster in eckigen Klammern stehen muss. Beispiel: [ip, user, username, timestamp, request, status_code, bytes].

Wenn Sie die Anzahl der Felder nicht kennen, können Sie eine Kurzbenachrichtigung mit Auslassungspunkten (...) verwenden. Beispiel, .

```
[..., status_code, bytes]
[ip, user, ..., status_code, bytes]
[ip, user, ...]
```

Sie können den Feldern auch Bedingungen hinzufügen, sodass nur Protokollereignisse, die alle Bedingungen erfüllen, den Filtern entsprechen. Beispiel, .

```
[ip, user, username, timestamp, request, status_code, bytes > 1000]
[ip, user, username, timestamp, request, status_code = 200, bytes]
[ip, user, username, timestamp, request, status_code = 4*, bytes]
[ip, user, username, timestamp, request = *html*, status_code = 4*, bytes]
```

Sie können `&&` als logischen UND-Operator und `||` als logischen ODER-Operator verwenden, wie in den folgenden Beispielen:

```
[ip, user, username, timestamp, request, status_code = 4* && bytes > 1000]
[ip, user, username, timestamp, request, status_code = 403 || status_code = 404, bytes]
```

CloudWatch Logs unterstützt sowohl Zeichenfolge- als auch numerische Bedingungsfelder. Für Zeichenfolgenfelder können Sie die Operatoren = oder != mit einem Stern (*) verwenden.

Für numerische Felder können Sie die Operatoren **>**, **<**, **>=**, **<=**, **=** und **!=** verwenden.

Wenn Sie einen durch Leerzeichen getrennten Filter verwenden, entsprechen die extrahierten Felder den Namen der durch Leerzeichen getrennten Feldern (wie im Filter angegeben) mit den Werten der einzelnen Felder. Wenn Sie keinen durch Leerzeichen getrennten Filter verwenden, bleibt dieses Feld leer.

Beispiel für Filter:

```
[..., request=*.html*, status_code=4*,]
```

Beispiel für ein Protokollereignis für den Filter:

```
127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /index.html HTTP/1.0\" 404 1534
```

Extrahierte Felder für das Protokollereignis und Filtermuster:

```
{
   "$status_code": "404",
   "$request": "GET /products/index.html HTTP/1.0",
   "$7": "1534",
   "$4": "10/Oct/2000:13:25:15 -0700",
   "$3": "frank",
   "$2": "-",
   "$1": "127.0.0.1"
}
```

Einrichten, wie sich der Metrikwert bei gefundenen Übereinstimmungen ändert

Wenn ein Metrikfilter einen der übereinstimmenden Begriffe, Ausdrücke oder Werte in Ihren Protokollereignissen findet, erhöht er die Zählung in der CloudWatch-Metrik um den Betrag, den Sie für den metrischen Wert festlegen. Der Metrikwert wird aggregiert und jede Minute gemeldet.

Wenn Protokolle während eines einminütigen Zeitraums übernommen, aber keine Übereinstimmungen gefunden werden, wird der als Standardwert angegebene Wert gemeldet (sofern vorhanden). Wenn während eines einminütigen Zeitraums allerdings keine Protokollereignisse übernommen werden, wird kein Wert gemeldet.

Die Angabe eines Standardwerts (selbst wenn dieser Wert 0 ist) stellt sicher, dass Daten häufiger gemeldet werden. Dadurch werden lückenhafte Metriken aufgrund nicht gefundener Übereinstimmungen verhindert.

Beispiel: Angenommen eine Protokollgruppe veröffentlicht jede Minute zwei Datensätze, der Metrikwert ist 1 und der Standardwert ist 0. Wenn in beiden Protokollsätzen Übereinstimmungen gefunden werden, lautet der metrische Wert für diese Minute 2. Wenn es keine Übereinstimmungen in den Protokolldatensätzen gibt, die in der zweiten Minute veröffentlicht werden, wird der Standardwert 0 für beide Protokolldatensätze verwendet und der Metrikwert für diese Minute ist 0.

Wenn Sie keine Standardwert angeben, werden für alle Zeiträume, in denen keine übereinstimmenden Muster gefunden werden, keine Daten gemeldet.

Veröffentlichen der in Protokolleinträgen gefundenen numerischen Werte

Anstatt nur die Anzahl der übereinstimmenden Elemente in Protokollen zu zählen, können Sie mit dem Metrikfilter auch auf numerischen Werten basierende Werte in den Protokollen veröffentlichen. Mit dem folgenden Vorgang wird eine Metrik mit der Latenz aus der JSON-Anfrage `metricFilter: { $.latency = * } metricValue: $.latency` veröffentlicht.

So veröffentlichen Sie eine Metrik mit der Latenz in einer JSON-Anforderung

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus.
3. Wählen Sie `Actions`, Metrikfilter erstellen aus.
4. Geben Sie unter Filtermuster den Wert **`{ $.latency = * }`** ein und wählen Sie dann Weiter.
5. Geben Sie für Metric Name myMetric ein.
6. Geben Sie für Metrikwert **`$.latency`** ein.
7. Geben Sie für Standardwert „0“ ein und wählen Sie dann Weiter. Durch die Angabe eines Standardwerts wird sichergestellt, dass Daten auch in Zeiträumen gemeldet werden, in denen keine Protokollereignisse mit dem Filter übereinstimmen. Dadurch werden lückenhafte oder fehlende Metriken verhindert, wenn Protokolle aufgenommen werden, aber nicht mit dem Filter übereinstimmen.
8. Wählen Sie Metrikfilter erstellen aus.

Das folgende Protokollereignis würde nach der Filtererstellung einen Wert von 50 in der Metrik myMetric veröffentlichen.

```
{
"latency": 50,
"requestType": "GET"
```

```
}
```

Erstellen von Metrikfiltern

Die folgenden Beispiele zeigen, wie Sie Metrikfilter erstellen.

Beispiele

- Beispiel. Zählprotokoll-Ereignisse (p. 83)
- Beispiel. Anzahl der Wiederholungen einer Laufzeit (p. 84)
- Beispiel. HTTP 404 Codes zählen (p. 85)
- Beispiel. HTTP 4xx Codes zählen (p. 87)
- Beispiel. Felder aus einem Apache-Protokoll extrahieren (p. 88)

Beispiel. Zählprotokoll-Ereignisse

Die einfachste Art der Überwachung von Protokollereignissen ist die Zählung der Anzahl von Protokollereignissen. Dies ist sinnvoll, um alle Ereignisse zu zählen, um eine "Heartbeat"-Überwachung zu erstellen oder um lediglich die Erstellung von Metrikfiltern zu üben.

Im folgenden Beispiel einer Befehlszeilenschnittstelle wird ein Metrikfilter mit dem Namen "MyAppAccessCount" auf die Protokollgruppe "MyApp/access.log" angewendet, sodass die Metrik "EventCount" im CloudWatch-Namespace "MyNamespace" erzeugt wird. Der Filter ist so konfiguriert, dass er dem Inhalt eines beliebigen Protokollereignisses entspricht und die Metrik um "1" erhöht wird.

So erstellen Sie einen Metrikfilter mithilfe der CloudWatch-Konsole

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus.
3. Wählen Sie den Namen einer Protokollgruppe aus.
4. Wählen Sie `Actions`, Metrikfilter erstellen aus.
5. Lassen Sie die Felder Filtermuster und Zu testende Protokolldaten auswählen leer.
6. Wählen Sie Weiter und geben Sie dann für Filtername **`EventCount`** ein.
7. Geben Sie unter Metric Details (Metrikdetails) für Metric Namespace (Metrik-Namesspace) **`MyNameSpace`** ein.
8. Geben Sie für Metric Name (Metrikname) den Wert **`MyAppEventCount`** ein.
9. Vergewissern Sie sich, dass Metrikwert „1“ lautet. Dadurch ist festgelegt, dass die Zählung für jedes Protokollereignis um 1 erhöht wird.
10. Geben Sie für Standardwert „0“ ein und wählen Sie dann Weiter. Die Angabe eines Standardwerts stellt sicher, dass auch in Zeiträumen, in denen keine Protokollereignisse auftreten, Daten gemeldet werden, und verhindert so lückenhafte Metriken, in denen manchmal gar keine Daten vorkommen.
11. Wählen Sie Metrikfilter erstellen aus.

So erstellen Sie einen Metrikfilter mithilfe derAWS CLI

Führen Sie an einer Eingabeaufforderung den folgenden Befehl aus.

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name EventCount \
```

```
  --filter-pattern "" \
  --metric-transformations \
  metricName=MyAppEventCount,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

Sie können diese neue Richtlinie durch Veröffentlichen beliebiger Ereignisdaten testen. Sie sollten Datenpunkte sehen, die in der Metrik MyAppAccessEventCount veröffentlicht sind.

So veröffentlichen Sie Ereignisdaten mithilfe der AWS CLI

Führen Sie an einer Eingabeaufforderung den folgenden Befehl aus.

```
aws logs put-log-events \
  --log-group-name MyApp/access.log --log-stream-name TestStream1 \
  --log-events \
    timestamp=1394793518000,message="Test event 1" \
    timestamp=1394793518000,message="Test event 2" \
    timestamp=1394793528000,message="This message also contains an Error"
```

Beispiel. Anzahl der Wiederholungen einer Laufzeit

Protokollereignisse enthalten häufig wichtige Nachrichten (z. B. über den Erfolg oder Misserfolg von Operationen), die Sie zählen möchten. Beispielsweise kann ein Fehler auftreten und in einer Protokolldatei aufgezeichnet werden, wenn eine bestimmte Operation fehlschlägt. Sie sollten diese Einträge überwachen, um sich einen Überblick über die Entwicklung Ihrer Fehler zu verschaffen.

In dem unten stehenden Beispiel wird ein Metrikfilter erstellt, mit dem der Begriff "Fehler" überwacht wird. Die Richtlinie wurde erstellt und der Protokollgruppe MyApp/message.log hinzugefügt. CloudWatch Logs veröffentlicht einen Datenpunkt in der benutzerdefinierten CloudWatch-Metrik ErrorCount im Namespace MyApp/message.log mit dem Wert „1“ für jedes Ereignis, das Error enthält. Wenn kein Ereignis das Wort "Error" enthält, werden der Wert 0 veröffentlicht. Bei der grafischen Darstellung dieser Daten in der CloudWatch-Konsole müssen Sie die Summenstatistik verwenden.

Nachdem Sie einen Metrikfilter erstellt haben, können Sie die Metrik in der CloudWatch-Konsole anzeigen. Wenn Sie die anzuzeigende Metrik auswählen, wählen Sie den Metrik-Namespace aus, der mit dem Namen der Protokollgruppe übereinstimmt. Weitere Informationen finden Sie unter Anzeigen der verfügbaren Metriken.

So erstellen Sie einen Metrikfilter mithilfe der CloudWatch-Konsole

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus.
3. Wählen Sie den Namen der Protokollgruppe aus.
4. Wählen Sie Aktionen, Metrikfilter erstellen aus.
5. Geben Sie für Filtermuster **`Error`** ein.

   Note

   Alle Einträge in Filter Pattern berücksichtigen Groß- und Kleinschreibung.
6. Um das Filtermuster zu testen, wählen Sie Testmuster.
7. Wählen Sie Weiter und geben Sie dann auf der Seite Metrik zuweisen für Filtername **`MyAppErrorCount`** ein.
8. Geben Sie unter Metric Details für Metric Namespace MyNameSpace ein.
9. Geben Sie für Metric Name ErrorCount ein.
10. Vergewissern Sie sich, dass Metrikwert „1“ lautet. Dadurch ist festgelegt, dass die Zählung für jedes Protokollereignis, das „Error“ enthält, um 1 erhöht wird.

11. Geben Sie für Standardwert „0“ ein und wählen Sie dann Weiter.
12. Wählen Sie Metrikfilter erstellen aus.

So erstellen Sie einen Metrikfilter mithilfe derAWS CLI

Führen Sie an einer Eingabeaufforderung den folgenden Befehl aus.

```
aws logs put-metric-filter \
  --log-group-name MyApp/message.log \
  --filter-name MyAppErrorCount \
  --filter-pattern 'Error' \
  --metric-transformations \
      metricName=ErrorCount,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

Sie können diese neue Richtlinie testen, indem Sie Ereignisse mit dem Begriff "Fehler" in der Nachricht veröffentlichen.

So veröffentlichen Sie Ereignisse mithilfe der AWS CLI

Führen Sie an einer Eingabeaufforderung den folgenden Befehl aus. Beachten Sie, dass in Mustern Groß- und Kleinschreibung berücksichtigt wird.

```
aws logs put-log-events \
  --log-group-name MyApp/access.log --log-stream-name TestStream1 \
  --log-events \
    timestamp=1394793518000,message="This message contains an Error" \
    timestamp=1394793528000,message="This message also contains an Error"
```

Beispiel. HTTP 404 Codes zählen

Mit CloudWatch Logs können Sie überwachen, wie oft der Apache-Server eine HTTP-Antwort mit dem Antwort-Code für "404 Seite nicht gefunden" zurückgibt. Sie sollten diese Zahl überwachen, um zu verstehen, wie oft die Besucher Ihrer Website die gesuchte Ressource nicht finden. Gehen Sie davon aus, dass die Protokolldatensätze so strukturiert sind, dass sie folgende Informationen für jede Protokollereigniss (Besuch der Website) enthalten:

- IP-Adresse des Anforderers
- RFC 1413-Identität
- Benutzername
- Zeitstempel
- Anforderungsmethode mit angeforderter Ressource und Protokoll
- Anzufordernder HTTP-Antwortcode
- Bei der Anforderung übertragene Bytes.

Ein Beispiel könnte folgendermaßen aussehen:

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 404 2326
```

Sie können eine Regel festlegen, die versucht, Ereignisse in der Struktur der HTTP 404-Fehler zu finden. Dies ist in folgendem Beispiel dargestellt:

So erstellen Sie einen Metrikfilter mithilfe der CloudWatch-Konsole

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.

2. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus.
3. Wählen Sie `Actions`, Metrikfilter erstellen aus.
4. Geben Sie für Filtermuster **`[IP, UserInfo, User, Timestamp, RequestInfo, StatusCode=404, Bytes]`** ein.
5. Um das Filtermuster zu testen, wählen Sie Testmuster.
6. Wählen Sie Weiter und geben Sie dann für Filtername HTTP404Errors ein.
7. Geben Sie unter Metrikdetails für Namesspace der Metrik **`MyNameSpace`** ein.
8. Geben Sie für Metrikname **`ApacheNotFoundErrorCount`** ein.
9. Vergewissern Sie sich, dass Metrikwert „1“ lautet. Dadurch ist festgelegt, dass die Zählung für jedes Protokollereignis „404 Error“ um 1 erhöht wird.
10. Geben Sie für Standardwert „0“ ein und wählen Sie dann Weiter.
11. Wählen Sie Metrikfilter erstellen aus.

So erstellen Sie einen Metrikfilter mithilfe derAWS CLI

Führen Sie an einer Eingabeaufforderung den folgenden Befehl aus.

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name HTTP404Errors \
  --filter-pattern '[ip, id, user, timestamp, request, status_code=404, size]' \
  --metric-transformations \
      metricName=ApacheNotFoundErrorCount,metricNamespace=MyNamespace,metricValue=1
```

In diesem Beispiel werden Literalzeichen wie die linke und rechte eckige Klammern, Anführungszeichen und die Zeichenfolge 404 verwendet. Das Muster muss mit der gesamten Protokollereignisnachricht für das zu überwachende Protokollereignis übereinstimmen.

Sie können überprüfen, ob der Metrikfilter erstellt wurde, indem Sie den Befehl describe-metric-filters ausführen. Die Ausgabe sollte in etwa wie folgt aussehen:

```
aws logs describe-metric-filters --log-group-name MyApp/access.log

{
    "metricFilters": [
        {
            "filterName": "HTTP404Errors",
            "metricTransformations": [
                {
                    "metricValue": "1",
                    "metricNamespace": "MyNamespace",
                    "metricName": "ApacheNotFoundErrorCount"
                }
            ],
            "creationTime": 1399277571078,
            "filterPattern": "[ip, id, user, timestamp, request, status_code=404, size]"
        }
    ]
}
```

Jetzt können Sie einige Ereignisse manuell veröffentlichen:

```
aws logs put-log-events \
--log-group-name MyApp/access.log --log-stream-name hostname \
--log-events \
timestamp=1394793518000,message="127.0.0.1 - bob [10/Oct/2000:13:55:36 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 404 2326" \
```

```
timestamp=1394793528000,message="127.0.0.1 - bob [10/Oct/2000:13:55:36 -0700] \"GET /
apache_pb2.gif HTTP/1.0\" 200 2326"
```

Kurz nachdem diese Beispiel-Protokollereignisse erfasst wurden, können Sie in der CloudWatch-Konsole die Metrik mit dem Namen "ApacheNotFoundErrorCount" abrufen.

Beispiel. HTTP 4xx Codes zählen

Wie im vorherigen Beispiel sollten Sie die Zugriffsprotokolle für Webservices sowie die Ebenen des HTTP-Antwortcodes überwachen. Sie sollten beispielsweise alle Fehler auf HTTP-400-Ebene überwachen. Sie sollten jedoch einen neuen Metrikfilter für jeden Rückgabecode angeben.

Im folgenden Beispiel wird gezeigt, wie Sie eine Metrik erstellen, die alle Antworten auf der Ebene des HTTP 400-Codes aus einem Zugriffsprotokoll enthält, und dazu das Format des Apache-Zugriffsprotokolls aus dem Beispiel Beispiel. HTTP 404 Codes zählen (p. 85) verwenden.

So erstellen Sie einen Metrikfilter mithilfe der CloudWatch-Konsole

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus.
3. Wählen Sie den Namen der Protokollgruppe für den Apache-Server aus.
4. Wählen Sie `Actions`, Metrikfilter erstellen aus.
5. für Filtername, geben Sie **`HTTP4xxErrors`**.
6. für Filtermuster, geben Sie **`[ip, id, user, timestamp, request, status_code=4*, size]`**.
7. Um das Filtermuster zu testen, wählen Sie Testmuster.
8. Wählen Nächstes, und dann für Filtername, Typ **`HTTP4xxErrors`**.
9. Unter Metrikdetails, für Metriknamespace, geben Sie **`MyNameSpace`**.
10. für Metrikname, geben Sie HTTP4xxFehler.
11. für Metrikwert, geben Sie 1 ein. Dadurch ist festgelegt, dass die Zählung für jedes Protokollereignis, das einen „4xx“-Fehler enthält, um 1 erhöht wird.
12. für Standardwert geben Sie 0 ein und wählen Sie dann Nächstes.
13. Wählen Sie Metrikfilter erstellen aus.

So erstellen Sie einen Metrikfilter mithilfe derAWS CLI

Führen Sie an einer Eingabeaufforderung den folgenden Befehl aus.

```
aws logs put-metric-filter \
  --log-group-name MyApp/access.log \
  --filter-name HTTP4xxErrors \
  --filter-pattern '[ip, id, user, timestamp, request, status_code=4*, size]' \
  --metric-transformations \
  metricName=HTTP4xxErrors,metricNamespace=MyNamespace,metricValue=1,defaultValue=0
```

Sie können die folgenden Daten in put-Ereignisaufrufen verwenden, um diese Regel zu testen. Wenn Sie die Überwachungsregel im vorherigen Beispiel nicht entfernt haben, werden zwei verschiedene Metriken generiert.

```
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /~test/ HTTP/1.1" 200 3
```

```
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:51:34 -0700] "GET /~test/index.html HTTP/1.1" 200 3
```

Beispiel. Felder aus einem Apache-Protokoll extrahieren

Manchmal ist es hilfreich, Werte aus einzelnen Protokollereignissen für Metrikwerte zu verwenden statt sie zu zählen. In diesem Beispiel wird gezeigt, wie Sie eine Extraktionsregel zum Erstellen einer Metrik erstellen können, mit der die von einem Apache-Webserver übertragenen Bytes gemessen werden.

Diese Extraktionsregel entspricht den sieben Feldern des Protokollereignisses. Der Metrikwert ist der Wert des siebten übereinstimmenden Token. Sie können den Verweis auf das Token als "$7" in dem Feld `metricValue` der Extraktionsregel erkennen.

So erstellen Sie einen Metrikfilter mithilfe der CloudWatch-Konsole

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus.
3. Wählen Sie den Namen der Protokollgruppe für den Apache-Server aus.
4. Wählen Sie `Actions`, Metrikfilter erstellen aus.
5. für Filtermuster, geben Sie **`[ip, id, user, timestamp, request, status_code, size]`**.
6. Um das Filtermuster zu testen, wählen Sie Testmuster.
7. Wählen Nächstes, und dann für Filtername, Typ **`size`**.
8. Unter Metrikdetails, für Metriknamespace, geben Sie **`MyNameSpace`**. Da dies ein neuer Namespace ist, stellen Sie sicher, dass Neue erstellen ist ausgewählt.
9. für Metrikname, geben Sie **`BytesTransferred`**
10. für Metrikwert, geben Sie **`$size`**.
11. für Standardwert geben Sie 0 ein und wählen Sie dann Nächstes.
12. Wählen Sie Metrikfilter erstellen aus.

So erstellen Sie einen Metrikfilter mithilfe derAWS CLI

Führen Sie an einer Eingabeaufforderung den folgenden Befehl aus.

```
aws logs put-metric-filter \
--log-group-name MyApp/access.log \
--filter-name BytesTransferred \
--filter-pattern '[ip, id, user, timestamp, request, status_code, size]' \
--metric-transformations \
metricName=BytesTransferred,metricNamespace=MyNamespace,metricValue=$size,defaultValue=0
```

Sie können die folgenden Daten in put-log-Ereignisaufrufen verwenden, um diese Regel zu testen. Damit werden zwei unterschiedliche Metriken erstellt, wenn Sie die Überwachungsregel im vorherigen Beispiel nicht entfernt haben.

```
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:49:52 -0700] "GET /index.html HTTP/1.1" 404 287
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /~test/ HTTP/1.1" 200 3
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:50:51 -0700] "GET /favicon.ico HTTP/1.1" 404 308
127.0.0.1 - - [24/Sep/2013:11:51:34 -0700] "GET /~test/index.html HTTP/1.1" 200 3
```

Auflisten von Metrikfiltern

Sie können alle Metrikfilter in einer Protokollgruppe auflisten.

So listen Sie Metrikfilter mit der CloudWatch-Konsole auf

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus.
3. Wählen Sie im Inhaltsbereich in der Liste der Protokollgruppen in der Spalte Metric Filters die Anzahl der Filter aus.

   Auf dem Bildschirm Log Groups > Filters for werden alle Metrikfilter für die Protokollgruppe aufgelistet.

So listen Sie Metrikfilter mithilfe der AWS CLI auf

Führen Sie an einer Eingabeaufforderung den folgenden Befehl aus.

```
aws logs describe-metric-filters --log-group-name MyApp/access.log
```

Im Folgenden finden Sie eine Beispielausgabe.

```
{
    "metricFilters": [
        {
            "filterName": "HTTP404Errors",
            "metricTransformations": [
                {
                    "metricValue": "1",
                    "metricNamespace": "MyNamespace",
                    "metricName": "ApacheNotFoundErrorCount"
                }
            ],
            "creationTime": 1399277571078,
            "filterPattern": "[ip, id, user, timestamp, request, status_code=404, size]"
        }
    ]
}
```

Löschen eines Metrikfilters

Eine Richtlinie wird anhand des Namens und der entsprechenden Protokollgruppe identifiziert.

So löschen Sie einen Metrikfilter mithilfe der CloudWatch-Konsole

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus.
3. Wählen Sie im Inhaltsbereich in der Spalte Metric Filter den Metrikfilter aus.
4. Wählen Sie auf dem Bildschirm Logs Metric Filters im Metrikfilter die Option Delete Filter aus.
5. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Yes, Delete (Ja, löschen).

So löschen Sie einen Metrikfilter mithilfe der AWS CLI

Führen Sie an einer Eingabeaufforderung den folgenden Befehl aus.

```
aws logs delete-metric-filter --log-group-name MyApp/access.log \
 --filter-name MyFilterName
```

Protokolldaten-Verarbeitung in Echtzeit mit Abonnements

Sie können Abonnements verwenden, um Zugang zu einem Echtzeit-Feed von Protokollereignissen von CloudWatch Logs und sie an andere Dienstleistungen wie z. B. Amazon Kinesis Stream, ein Amazon Kinesis Data Firehose Stream oder AWS Lambda zur benutzerdefinierten Verarbeitung, Analyse oder zum Laden auf andere Systeme. Wenn Protokollereignisse an den empfangenden Dienst gesendet werden, sind sie mit dem gzip-Format verbunden und komprimiert.

Um das Abonnieren von Ereignissen zu beginnen, erstellen Sie die empfangende Ressource, z. B. Kinesis Stream, wo die Ereignisse zugestellt werden. Ein Abonnementfilter definiert die Filtermuster, die zum Filtern der für die AWS-Ressource bereitgestellten Protokollereignisse verwendet werden und als Informationen zum Speicherort für die passenden Ereignisse dienen.

Jeder Protokollgruppe kann nur ein Abonnementfilter zugeordnet sein.

> Note
>
> Wenn der Zielservice einen wiederholbaren Fehler zurückgibt, z. B. eine Ablehnungsausnahme oder eine wiederholbare Serviceausnahme (z. B. HTTP 5xx), versucht CloudWatch Logs die Zustellung bis zu 24 Stunden. CloudWatch Logs versucht die Zustellung nicht weiter, wenn es sich beim Fehler um einen nicht wiederholbaren Fehler handelt, wie AccessDeniedException oder ResourceNotFoundException.

CloudWatch Logs produziert darüber hinaus CloudWatch-Metriken über die Weiterleitung von Protokollereignissen an Abonnements. Weitere Informationen finden Sie unter Amazon CloudWatch Logs-Metriken und -Dimensionen.

Inhalt:

- Concepts (p. 91)
- Verwenden von CloudWatch Logs-Abonnementfiltern (p. 92)
- Freigabe von Protokolldaten mit Abonnements für mehrere Konten (p. 103)

Concepts

Jeder Abonnementfilter besteht aus den folgenden Schlüsselelementen:

Name der Protokollgruppe

Die Protokollgruppe, die mit dem Abonnementfilter verknüpft ist. Alle Protokollereignisse, die in diese Protokollgruppe hochgeladen wurden, würden dem Abonnementfilter unterliegen, und diejenigen, die dem Filter entsprechen, werden an den Zieldienst geliefert, der die passenden Protokollereignisse erhält.

Filtermuster

Eine symbolische Beschreibung dazu, wie CloudWatch Logs die Daten in jedem Protokollereignis interpretiert, zusammen mit Filterausdrücken, die festlegen, was an die AWS-Zielressourcen gesendet wird. Weitere Informationen über die Syntax von Filtermustern finden Sie unter Filter- und Mustersyntax (p. 75).

Ziel-ARN

Der Amazon-Ressourcenname (ARN) des Kinesis-Streams, des Kinesis Data Firehose-Streams oder der Lambda-Funktion, den oder die Sie als Ziel des Abonnement-Feeds verwenden möchten.

--role-arn

Ein IAM Rolle, die CloudWatch Logs die erforderlichen Berechtigungen zum Ablegen von Daten in das gewählte Ziel. Diese Rolle ist für Lambda-Ziele nicht erforderlich, da CloudWatch Logs die erforderlichen Berechtigungen aus Zugriffssteuerungseinstellungen für die Lambda-Funktion selbst abrufen kann.

Verteilung

Die Methode zur Verteilung von Protokolldaten an das Ziel, wenn das Ziel eine Amazon Kinesis Stream ist. Standardmäßig werden Daten nach Protokoll-Stream gruppiert. Für eine gleichmäßige Verteilung können Sie Protokolldaten zufällig gruppieren.

Verwenden von CloudWatch Logs-Abonnementfiltern

Sie können einen Abonnementfilter mit Kinesis, Lambda oder Kinesis Data Firehose verwenden. Protokolle, die an einen Empfangsdienst über einen Abonnementfilter gesendet werden, sind Base64 kodiert und mit dem gzip-Format komprimiert.

Beispiele

- Beispiel 1 Abonnementfilter mit Kinesis (p. 92)
- Beispiel 2 Abonnementfilter mit AWS Lambda (p. 96)
- Beispiel 3 Abonnementfilter mit Amazon Kinesis Data Firehose (p. 98)

Beispiel 1 Abonnementfilter mit Kinesis

Im folgenden Beispiel wird ein Abonnementfilter mit einer Protokollgruppe verknüpft, die alle AWS CloudTrail-Ereignisse enthält. Somit wird jede protokollierte Aktivität, die mit „Stamm“-AWS-Anmeldeinformationen durchgeführt wird, an einen Kinesis-Stream mit dem Namen „RootAccess“ gesendet. Weitere Informationen zum Senden von AWS CloudTrail-Ereignissen an CloudWatch Logs finden Sie unter Senden von CloudTrail-Ereignisreferenz an CloudWatch Logs im AWS CloudTrail User Guide.

Note

Bevor Sie den Kinesis-Stream erstellen, berechnen Sie das Volumen der generierten Protokolldaten. Vergewissern Sie sich, dass Sie einen Kinesis-Stream mit einer ausreichenden Anzahl von Shards zur Bearbeitung dieses Volumens erstellen. Wenn der Stream nicht über genügend Shards verfügt, wird der Protokoll-Stream gedrosselt. Weitere Informationen zu den Volumenlimits bei Kinesis-Streams finden Sie unter Limits für Amazon Kinesis Data Streams.

So erstellen Sie einen Abonnementfilter für Kinesis

1. Erstellen Sie einen Ziel-Kinesis-Stream mit dem folgenden Befehl:

```
$ C:\>  aws kinesis create-stream --stream-name "RootAccess" --shard-count 1
```

2. Warten Sie, bis der Kinesis-Stream "Aktiv" wird (dies kann einige Minuten dauern). Verwenden Sie den folgenden Kinesis-Befehl describe-stream, um die Eigenschaft StreamDescription.StreamStatus zu überprüfen. Notieren Sie sich außerdem den Wert StreamDescription.StreamARN, da Sie ihn in einem späteren Schritt benötigen:

```
aws kinesis describe-stream --stream-name "RootAccess"
```

Im Folgenden finden Sie eine Beispielausgabe.

```
{
    "StreamDescription": {
        "StreamStatus": "ACTIVE",
        "StreamName": "RootAccess",
        "StreamARN": "arn:aws:kinesis:us-east-1:123456789012:stream/RootAccess",
        "Shards": [
            {
                "ShardId": "shardId-000000000000",
                "HashKeyRange": {
                    "EndingHashKey": "340282366920938463463374607431768211455",
                    "StartingHashKey": "0"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
                    "49551135218688818456679503831981458784591352702181572610"
                }
            }
        ]
    }
}
```

3. Erstellen Sie die IAM-Rolle, die CloudWatch Logs die Berechtigung erteilt, Daten in den Kinesis-Stream zu übertragen. Zunächst müssen Sie eine Vertrauensrichtlinie in einer Datei erstellen (z. B. `~/TrustPolicyForCWL.json`). Verwenden Sie einen Text-Editor, um diese Richtlinie zu erstellen. Verwenden Sie zum Erstellen nicht die IAM-Konsole.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.region.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

4. Verwenden Sie den Befehl create-role, um die IAM-Rolle zu erstellen und die Vertrauensrichtlinie anzugeben. Notieren Sie den zurückgegebenen Wert Role.Arn, da Sie ihn in einem späteren Schritt benötigen:

```
aws iam create-role --role-name CWLtoKinesisRole --assume-role-policy-document file://
~/TrustPolicyForCWL.json
```

Es folgt ein Beispiel für die Ausgabe.

```
{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.region.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
```

```
    }
}
```

5. Erstellen Sie eine Berechtigungsrichtlinie, um zu definieren, welche Aktionen CloudWatch Logs an Ihrem Konto durchführen darf. Erstellen Sie zunächst eine Berechtigungsrichtlinie in einer Datei (z. B. `~/PermissionsForCWL.json`). Verwenden Sie einen Text-Editor, um diese Richtlinie zu erstellen. Verwenden Sie zum Erstellen nicht die IAM-Konsole.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:123456789012:stream/RootAccess"
    }
  ]
}
```

6. Verknüpfen Sie die Berechtigungsrichtlinie mit der Rolle, und verwenden Sie dazu den Befehl put-role-policy:

```
aws iam put-role-policy  --role-name CWLtoKinesisRole  --policy-name Permissions-
Policy-For-CWL  --policy-document file://~/PermissionsForCWL.json
```

7. Wenn sich der Kinesis-Stream im Status Active (Aktiv) befindet und Sie die IAM-Rolle erstellt haben, können Sie den CloudWatch Logs-Abonnementfilter erstellen. Der Abonnementfilter startet sofort den Fluss der Echtzeit-Protokolldaten aus der gewählten Protokollgruppe an den Kinesis-Stream:

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail" \
    --filter-name "RootAccess" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:kinesis:region:123456789012:stream/RootAccess" \
    --role-arn "arn:aws:iam::123456789012:role/CWLtoKinesisRole"
```

8. Wenn Sie den Abonnementfilter eingerichtet haben, leitet CloudWatch Logs alle eingehenden Protokollereignisse, die dem Filtermuster entsprechen, an den Kinesis-Stream weiter. Sie können überprüfen, ob dies auch geschieht, indem Sie einen Kinesis-Shard-Iterator aufrufen und mit dem Kinesis-Befehl „get-records“ einige Kinesis-Datensätze abrufen:

```
aws kinesis get-shard-iterator --stream-name RootAccess --shard-id shardId-000000000000
 --shard-iterator-type TRIM_HORIZON
```

```
{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiK2OSh0uP"
}
```

```
aws kinesis get-records --limit 10 --shard-iterator "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiK2OSh0uP"
```

Beachten Sie, dass Sie diesen Befehl möglicherweise mehrmals aufrufen müssen, bevor Kinesis Daten zurückgibt.

Sie sollten davon ausgehen, dass die Antwort in einem Datensatz-Array angezeigt wird. Das Attribut Data in einem Kinesis-Datensatz ist Base64-kodiert und im GZIP-Format komprimiert. Sie können die Rohdaten über die Befehlszeile mit den folgenden Unix-Befehlen überprüfen:

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Die dekodierten und dekomprimierten Base64-Daten sind als JSON mit folgender Struktur formatiert:

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail",
    "logStream": "111111111111_CloudTrail_us-east-1",
    "subscriptionFilters": [
        "Destination"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        }
    ]
}
```

Die Schlüsselelemente in der oben genannten Datenstruktur sind folgende:

owner

Die ID des AWS-Kontos der ursprünglichen Protokolldaten.

logGroup

Der Name der Protokollgruppe der ursprünglichen Protokolldaten.

logStream

Der Name des Protokoll-Stream der ursprünglichen Protokolldaten.

subscriptionFilters

Die Namenliste der Abonnementfilter, die mit den ursprünglichen Protokolldaten übereingestimmt haben.

messageType

Datennachrichten verwenden den Typ "DATA_MESSAGE". In einigen Fällen kann CloudWatch Logs Kinesis-Datensätze mit dem Typ "CONTROL_MESSAGE" ausgeben. Sie dienen hauptsächlich zur Prüfung, ob das Ziel erreichbar ist.

logEvents

Die tatsächlichen Protokolldaten, die als Array von Protokollereignis-Datensätzen dargestellt werden. Die "id"-Eigenschaft ist eine eindeutige Kennung für jedes Protokollereignis.

Beispiel 2 Abonnementfilter mit AWS Lambda

In diesem Beispiel erstellen Sie einen CloudWatch Logs-Abonnementfilter, der Protokolldaten an die AWS Lambda-Funktion sendet.

Note

Bevor Sie die Lambda-Funktion erstellen, berechnen Sie das Volumen der generierten Protokolldaten. Vergewissern Sie sich, dass Sie eine Funktion erstellen, die dieses Volumen bearbeiten kann. Wenn die Funktion nicht über genügend Volumen verfügt, wird der Protokollstream gedrosselt. Weitere Informationen zu den Limits für Lambda finden Sie unter AWS Lambda-Limits.

So erstellen Sie einen Abonnementfilter für Lambda

1. Erstellen der AWS Lambda-Funktion

   Stellen Sie sicher, dass Sie die Lambda-Ausführungsrolle eingerichtet haben. Weitere Informationen finden Sie unter Schritt 2.2: Erstellen einer IAM-Rolle (Ausführungsrolle) im AWS Lambda Developer Guide.
2. Öffnen Sie einen Text-Editor, und erstellen Sie eine Datei mit dem Namen `helloWorld.js` mit folgendem Inhalt:

```
var zlib = require('zlib');
exports.handler = function(input, context) {
    var payload = Buffer.from(input.awslogs.data, 'base64');
    zlib.gunzip(payload, function(e, result) {
        if (e) {
            context.fail(e);
        } else {
            result = JSON.parse(result.toString('ascii'));
            console.log("Event Data:", JSON.stringify(result, null, 2));
            context.succeed();
        }
    });
};
```

3. Komprimieren Sie die Datei "helloWorld.js", und speichern Sie sie unter dem Namen `helloWorld.zip`.
4. Verwenden Sie den folgenden Befehl, wobei die Rolle die Lambda-Ausführungsrolle ist, die Sie im ersten Schritt eingerichtet haben:

```
aws lambda create-function \
    --function-name helloworld \
    --zip-file fileb://file-path/helloWorld.zip \
    --role lambda-execution-role-arn \
    --handler helloWorld.handler \
    --runtime nodejs12.x
```

5. Erteilen Sie CloudWatch Logs die Berechtigung zum Ausführen der Funktion. Verwenden Sie den folgenden Befehl, und ersetzen Sie den Platzhalter für das Konto mit Ihrem eigenen Konto und den Platzhalter für die Protokollgruppe mit der zu verarbeitenden Protokollgruppe:

```
aws lambda add-permission \
    --function-name "helloworld" \
    --statement-id "helloworld" \
    --principal "logs.region.amazonaws.com" \
    --action "lambda:InvokeFunction" \
    --source-arn "arn:aws:logs:region:123456789123:log-group:TestLambda:*" \
    --source-account "123456789012"
```

6. Erstellen Sie einen Abonnementfilter mit den folgenden Befehl, und ersetzen Sie den Platzhalter für das Konto mit Ihrem eigenen Konto und den Platzhalter für die Protokollgruppe mit der zu verarbeitenden Protokollgruppe:

```
aws logs put-subscription-filter \
    --log-group-name myLogGroup \
    --filter-name demo \
    --filter-pattern "" \
    --destination-arn arn:aws:lambda:region:123456789123:function:helloworld
```

7. (Optional) Führen Sie einen Test mit einem Beispiel-Protokollereignis durch. Geben Sie an der Eingabeaufforderung den folgenden Befehl ein. Damit wird eine einfache Protokollnachricht in den registrierten Stream gestellt.

   Wenn Sie die Ausgabe der Lambda-Funktion anzeigen möchten, navigieren Sie zu der Lambda-Funktion. Dort wird die Ausgabe unter /aws/lambda/helloworld angezeigt:

```
aws logs put-log-events --log-group-name myLogGroup --log-stream-name stream1 --log-
events "[{\"timestamp\":<CURRENT TIMESTAMP MILLIS> , \"message\": \"Simple Lambda
 Test\"}]"
```

   Sie sollten davon ausgehen, dass die Antwort mit in einem Lambda-Array angezeigt wird. Das Attribut Data im Lambda-Datensatz ist Base64-kodiert und im GZIP-Format komprimiert. Die tatsächliche Nutzlast, die Lambda erhält, hat folgendes Format: `{ "awslogs": {"data": "BASE64ENCODED_GZIP_COMPRESSED_DATA"} }`. Sie können die Rohdaten über die Befehlszeile mit den folgenden Unix-Befehlen überprüfen:

```
echo -n "<BASE64ENCODED_GZIP_COMPRESSED_DATA>" | base64 -d | zcat
```

   Die dekodierten und dekomprimierten Base64-Daten sind als JSON mit folgender Struktur formatiert:

```
{
    "owner": "123456789012",
    "logGroup": "CloudTrail",
    "logStream": "123456789012_CloudTrail_us-east-1",
    "subscriptionFilters": [
        "Destination"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424298221568",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221569",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
```

```
        },
        {
            "id": "31953106606966983378809025079804211143289615424298221570",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root
\"}"
        }
    ]
}
```

Die Schlüsselelemente in der oben genannten Datenstruktur sind folgende:

owner

Die ID des AWS-Kontos der ursprünglichen Protokolldaten.

logGroup

Der Name der Protokollgruppe der ursprünglichen Protokolldaten.

logStream

Der Name des Protokoll-Stream der ursprünglichen Protokolldaten.

subscriptionFilters

Die Namenliste der Abonnementfilter, die mit den ursprünglichen Protokolldaten übereingestimmt haben.

messageType

Datennachrichten verwenden den Typ "DATA_MESSAGE". In einigen Fällen kann CloudWatch Logs Lambda-Datensätze mit dem Typ "CONTROL_MESSAGE" ausgeben. Sie dienen hauptsächlich zur Prüfung, ob das Ziel erreichbar ist.

logEvents

Die tatsächlichen Protokolldaten, die als Array von Protokollereignis-Datensätzen dargestellt werden. Die "id"-Eigenschaft ist eine eindeutige Kennung für jedes Protokollereignis.

Beispiel 3 Abonnementfilter mit Amazon Kinesis Data Firehose

In diesem Beispiel erstellen Sie ein CloudWatch Logs-Abonnement, das alle eingehenden Ereignisse, die mit dem definierten Filter übereinstimmen, an den Amazon Kinesis Data Firehose-Bereitstellungsstream sendet. Die Daten, die von CloudWatch Logs an Amazon Kinesis Data Firehose gesendet wurden, sind bereits mit GZIP-Level 6 komprimiert. Es ist also nicht erforderlich, die Komprimierung innerhalb des Kinesis Data Firehose-Bereitstellungs-Stream zu verwenden.

Note

Bevor Sie den Kinesis Data Firehose-Stream erstellen, berechnen Sie das Volumen der generierten Protokolldaten. Vergewissern Sie sich, dass Sie einen Kinesis Data Firehose-Stream erstellen, der dieses Volumen bearbeiten kann. Wenn der Stream das Volumen nicht bearbeiten kann, wird die Protokoll-Stream gedrosselt. Weitere Informationen zu den Volumenlimits bei Kinesis Data Firehose-Streams finden Sie unter Amazon Kinesis Data Firehose Data Limits.

So erstellen Sie einen Abonnementfilter für Kinesis Data Firehose

1. Erstellen eines Amazon Simple Storage Service (Amazon S3)-Buckets Wir empfehlen, dass Sie einen Bucket verwenden, der speziell für CloudWatch Logs erstellt wurde. Wenn Sie jedoch einen vorhandenen Bucket verwenden möchten, gehen Sie direkt zu Schritt 2.

Führen Sie den folgenden Befehl aus, und ersetzen Sie den Platzhalter für die Region mit der Region, die Sie verwenden möchten:

```
aws s3api create-bucket --bucket my-bucket --create-bucket-configuration
 LocationConstraint=region
```

Im Folgenden finden Sie eine Beispielausgabe.

```
{
    "Location": "/my-bucket"
}
```

2. Erstellen Sie die IAM-Rolle, die Amazon Kinesis Data Firehose die Berechtigung erteilt, Daten in den Amazon S3-Bucket zu übertragen.

   Weitere Informationen finden Sie unter Zugriffskontrolle mit Amazon Kinesis Data Firehose im Amazon Kinesis Data Firehose-Entwicklerhandbuch.

   Zuerst verwenden Sie einen Text-Editor zum Erstellen einer Vertrauensrichtlinie in einer Datei `~/TrustPolicyForFirehose.json` wie nachfolgend aufgeführt. Ersetzen Sie dabei account-id durch die ID Ihres AWS-Kontos:

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "firehose.amazonaws.com" },
    "Action": "sts:AssumeRole",
    "Condition": { "StringEquals": { "sts:ExternalId":"account-id" } }
  }
}
```

3. Verwenden Sie den Befehl create-role, um die IAM-Rolle zu erstellen und die Vertrauensrichtlinie anzugeben. Notieren Sie den zurückgegebenen Wert Role.Arn, da Sie ihn in einem späteren Schritt benötigen:

```
aws iam create-role \
      --role-name FirehosetoS3Role \
      --assume-role-policy-document file://~/TrustPolicyForFirehose.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "firehose.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "FirehosetoS3Role",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/FirehosetoS3Role"
    }
}
```

4. Erstellen Sie eine Berechtigungsrichtlinie, um zu definieren, welche Aktionen Kinesis Data Firehose an Ihrem Konto durchführen darf. Verwenden Sie zunächst einen Text-Editor zum Erstellen einer Berechtigungsrichtlinie in einer Datei `~/PermissionsForFirehose.json`:

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "s3:AbortMultipartUpload",
          "s3:GetBucketLocation",
          "s3:GetObject",
          "s3:ListBucket",
          "s3:ListBucketMultipartUploads",
          "s3:PutObject" ],
      "Resource": [
          "arn:aws:s3:::my-bucket",
          "arn:aws:s3:::my-bucket/*" ]
    }
  ]
}
```

5. Verknüpfen Sie die Berechtigungsrichtlinie mit der Rolle, und verwenden Sie dazu den Befehl put-role-policy:

```
aws iam put-role-policy --role-name FirehosetoS3Role --policy-name Permissions-Policy-
For-Firehose --policy-document file://~/PermissionsForFirehose.json
```

6. Erstellen Sie einen Ziel-Bereitstellungs-Stream in Kinesis Data Firehose wie nachfolgend aufgeführt, und ersetzen Sie dabei die Werte für RoleARN und BucketARN mit der Rolle und den Bucket-ARNs, die Sie erstellt haben:

```
aws firehose create-delivery-stream \
   --delivery-stream-name 'my-delivery-stream' \
   --s3-destination-configuration \
  '{"RoleARN": "arn:aws:iam::123456789012:role/FirehosetoS3Role", "BucketARN":
 "arn:aws:s3:::my-bucket"}'
```

   Beachten Sie, dass Kinesis Data Firehose für die bereitgestellten Amazon S3-Objekte automatisch ein Präfix im Zeitformat JJJJ/MM/TT/HH (UTC) verwendet. Sie können ein zusätzliches Präfix vor dem Zeitformat-Präfix hinzufügen. Wenn das Präfix mit einem Schrägstrich (/) endet, wird es als Ordner im Amazon S3-Bucket angezeigt.

7. Warten Sie, bis der Stream aktiv wird (dies kann einige Minuten dauern). Verwenden Sie den Kinesis Data Firehose-Befehl describe-delivery-stream, um die Eigenschaft DeliveryStreamDescription.DeliveryStreamStatus zu überprüfen. Notieren Sie sich außerdem den Wert DeliveryStreamDescription.DeliveryStreamARN, da Sie ihn in einem späteren Schritt benötigen:

```
aws firehose describe-delivery-stream --delivery-stream-name "my-delivery-stream"
{
    "DeliveryStreamDescription": {
        "HasMoreDestinations": false,
        "VersionId": "1",
        "CreateTimestamp": 1446075815.822,
        "DeliveryStreamARN": "arn:aws:firehose:us-east-1:123456789012:deliverystream/
my-delivery-stream",
        "DeliveryStreamStatus": "ACTIVE",
        "DeliveryStreamName": "my-delivery-stream",
        "Destinations": [
            {
```

```
                "DestinationId": "destinationId-000000000001",
                "S3DestinationDescription": {
                    "CompressionFormat": "UNCOMPRESSED",
                    "EncryptionConfiguration": {
                        "NoEncryptionConfig": "NoEncryption"
                    },
                    "RoleARN": "delivery-stream-role",
                    "BucketARN": "arn:aws:s3:::my-bucket",
                    "BufferingHints": {
                        "IntervalInSeconds": 300,
                        "SizeInMBs": 5
                    }
                }
            }
        ]
    }
}
```

8. Erstellen Sie die IAM-Rolle, die CloudWatch Logs die Berechtigung erteilt, Daten in den Kinesis Data Firehose-Bereitstellungs-Stream zu stellen. Verwenden Sie zunächst einen Text-Editor zum Erstellen einer Vertrauensrichtlinie in einer Datei `~/TrustPolicyForCWL.json`:

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.region.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

9. Verwenden Sie den Befehl create-role, um die IAM-Rolle zu erstellen und die Vertrauensrichtlinie anzugeben. Notieren Sie den zurückgegebenen Wert Role.Arn, da Sie ihn in einem späteren Schritt benötigen:

```
aws iam create-role \
      --role-name CWLtoKinesisFirehoseRole \
      --assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.region.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisFirehoseRole",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
    }
}
```

10. Erstellen Sie eine Berechtigungsrichtlinie, um zu definieren, welche Aktionen CloudWatch Logs an Ihrem Konto durchführen darf. Verwenden Sie zunächst einen Text-Editor zum Erstellen einer Berechtigungsrichtliniendatei (beispielsweise `~/PermissionsForCWL.json`):

```
{
    "Statement":[
```

```
      {
        "Effect":"Allow",
        "Action":["firehose:*"],
        "Resource":["arn:aws:firehose:region:123456789012:*"]
      }
    ]
}
```

11. Verknüpfen Sie die Berechtigungsrichtlinie mit der Rolle, und verwenden Sie dazu den Befehl put-role-policy:

```
aws iam put-role-policy --role-name CWLtoKinesisFirehoseRole --policy-name Permissions-Policy-For-CWL --policy-document file://~/PermissionsForCWL.json
```

12. Wenn sich der Amazon Kinesis Data Firehose-Bereitstellungsstream im Status Active (Aktiv) befindet und Sie die IAM-Rolle erstellt haben, können Sie den CloudWatch Logs-Abonnementfilter erstellen. Der Abonnementfilter startet sofort den Fluss der Echtzeit-Protokolldaten aus der gewählten Protokollgruppe an den Amazon Kinesis Data Firehose-Bereitstellungsstream:

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail" \
    --filter-name "Destination" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:firehose:region:123456789012:deliverystream/my-delivery-stream" \
    --role-arn "arn:aws:iam::123456789012:role/CWLtoKinesisFirehoseRole"
```

13. Wenn Sie den Abonnementfilter eingerichtet haben, leitet CloudWatch Logs alle eingehenden Protokollereignisse, die dem Filtermuster entsprechen, an den Amazon Kinesis Data Firehose-Bereitstellungs-Stream weiter. Je nach Zeitpufferintervall, das für den Amazon S3-Bereitstellungs-Stream festgelegt ist, werden die Daten dann in Amazon Kinesis Data Firehose angezeigt. Sobald genügend Zeit abgelaufen ist, können Sie die Daten im Amazon S3-Bucket überprüfen.

```
aws s3api list-objects --bucket 'my-bucket' --prefix 'firehose/'
{
    "Contents": [
        {
            "LastModified": "2015-10-29T00:01:25.000Z",
            "ETag": "\"a14589f8897f4089d3264d9e2d1f1610\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-stream-2015-10-29-00-01-21-a188030a-62d2-49e6-b7c2-b11f1a7ba250",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b5"
            },
            "Size": 593
        },
        {
            "LastModified": "2015-10-29T00:35:41.000Z",
            "ETag": "\"a7035b65872bb2161388ffb63dd1aec5\"",
            "StorageClass": "STANDARD",
            "Key": "firehose/2015/10/29/00/my-delivery-stream-2015-10-29-00-35-40-7cc92023-7e66-49bc-9fd4-fc9819cc8ed3",
            "Owner": {
                "DisplayName": "cloudwatch-logs",
                "ID": "1ec9cf700ef6be062b19584e0b7d84ecc19237f87b6"
            },
            "Size": 5752
        }
    ]
}
```

```
aws s3api get-object --bucket 'my-bucket' --key 'firehose/2015/10/29/00/my-delivery-
stream-2015-10-29-00-01-21-a188030a-62d2-49e6-b7c2-b11f1a7ba250' testfile.gz

{
    "AcceptRanges": "bytes",
    "ContentType": "application/octet-stream",
    "LastModified": "Thu, 29 Oct 2015 00:07:06 GMT",
    "ContentLength": 593,
    "Metadata": {}
}
```

Die Daten im Amazon S3-Objekt sind im GZIP-Format komprimiert. Sie können die Rohdaten über die Befehlszeile mit dem folgenden Unix-Befehl überprüfen:

```
zcat testfile.gz
```

Freigabe von Protokolldaten mit Abonnements für mehrere Konten

Sie können mit einem Eigentümer eines anderen AWS-Kontos zusammenarbeiten und seine Protokollereignisse, wie z. B. einen Amazon Kinesis-Stream, in Ihren AWS-Ressourcen erhalten (dies wird auch kontoübergreifende Freigabe von Daten genannt). Diese Protokollereignisdaten können dann von einem zentralisierten Amazon Kinesis-Stream gelesen werden, und es können benutzerdefinierte Verarbeitungsvorgänge und Analysen durchgeführt werden. Benutzerdefinierte Verarbeitung ist besonders nützlich, wenn Sie über viele Konten zusammenarbeiten und Daten analysieren. Wenn beispielsweise die IT-Sicherheitsgruppe eines Unternehmens Daten zur Echtzeit-Erkennung von Eindringversuchen oder auf Unregelmäßigkeiten analysieren möchten, kann sie eine Prüfung aller Konten in allen Abteilungen des Unternehmens durchführen und dabei alle Produktionsprotokolle für eine zentrale Verarbeitung erfassen. Dabei kann ein Echtzeit-Stream von Ereignisdaten aller dieser Konten zusammengestellt und an die IT-Sicherheitsgruppe gesendet werden, die die Daten wiederum mithilfe von Kinesis mit den bestehenden Sicherheits-Analyse-Systemen verknüpfen kann.

Kinesis-Streams sind derzeit die einzige Ressource, die als Ziel für kontoübergreifende Abonnements unterstützt wird.

Wenn Sie Protokolldaten für mehrere Konten freigeben möchten, müssen Sie Sender und Receiver der Protokolldaten festlegen:

- Der Log data sender — erhält die Zielinformationen vom Empfänger und teilt CloudWatch Logs mit, dass er bereit ist, Protokollereignisse an das angegebene Ziel zu senden. In den Verfahren im übrigen Abschnitt werden die Protokolldatensender mit der fiktionalen AWS-Kontonummer 111111111111 angezeigt.
- Log data recipient (Protokolldaten-Empfänger) – Richtet ein Ziel ein, das einen Kinesis-Stream enthält, und teilt CloudWatch Logs mit, dass der Empfänger Protokolldaten empfangen möchte. Der Empfänger gibt dann die Informationen über seine Ziel für den Absender frei. In den Verfahren im übrigen Abschnitt werden die Protokolldatenempfänger mit der fiktionalen AWS-Kontonummer 999999999999 angezeigt.

Zum Starten des Empfangs von Protokollereignissen von kontoübergreifenden Benutzern, erstellt der Empfänger der Protokolldaten zunächst ein CloudWatch Logs-Ziel. Jedes Ziel umfasst die folgenden Schlüsselelemente:

Name des Ziels

Der Name der Zielregion, die Sie erstellen möchten.

Ziel-ARN

Der Amazon-Ressourcenname (ARN) der AWS-Ressourcen, die Sie als Ziel für den Abonnement-Feed verwenden möchten.

ARN der Rolle: .

Eine AWS Identity and Access Management (IAM)-Rolle, die CloudWatch Logs die erforderlichen Berechtigungen gewährt, um Daten in die den ausgewählten Kinesis-Stream zu stellen.

Zugriffsrichtlinie

Ein IAM-Richtliniendokument (im JSON-Format, das in der IAM-Richtliniengrammatik geschrieben ist), das kontrolliert, welche Benutzer Schreibberechtigung für das Ziel haben.

Die Log-Gruppe und das Ziel müssen sich in derselben AWS-Region befinden. Die AWS-Ressource, auf die das Ziel verweist, kann sich in einer anderen Region befinden.

Themen

- Erstellen eines Ziels (p. 104)
- Erstellen eines Abonnementfilters (p. 107)
- Validierung des Flusses von Protokollereignissen (p. 107)
- Ändern der Mitgliedschaft im Ziel zur Laufzeit (p. 109)

Erstellen eines Ziels

Important

Alle Schritte in diesem Verfahren müssen im Konto des Protokolldatenempfängers ausgeführt werden.

In diesem Beispiel hat der Empfänger Protokolldaten die AWS-Konto-ID 999999999999, während der Sender der Protokolldaten die AWS-Konto-ID 111111111111 hat.

In diesem Beispiel werden ein Ziel mithilfe eines Kinesis-Stream namens „RecipientStream“ sowie eine Rolle erstellt, mit der CloudWatch Logs Schreibberechtigung für das Ziel erhält.

So erstellen Sie ein Ziel

1. Erstellen Sie einen Ziel-Stream in Kinesis. Geben Sie an der Eingabeaufforderung Folgendes ein:

```
aws kinesis create-stream --stream-name "RecipientStream" --shard-count 1
```

2. Warten Sie, bis der Kinesis-Stream aktiv wird. Verwenden Sie den Befehl aws kinesis describe-stream verwenden, um die Eigenschaft StreamDescription.StreamStatus zu überprüfen. Notieren Sie außerdem den Wert StreamDescription.StreamARN, da er später an CloudWatch Logs übergeben wird:

```
aws kinesis describe-stream --stream-name "RecipientStream"
{
  "StreamDescription": {
    "StreamStatus": "ACTIVE",
```

```
    "StreamName": "RecipientStream",
    "StreamARN": "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream",
    "Shards": [
      {
        "ShardId": "shardId-000000000000",
        "HashKeyRange": {
          "EndingHashKey": "340282366920938463463374607431768EXAMPLE",
          "StartingHashKey": "0"
        },
        "SequenceNumberRange": {
          "StartingSequenceNumber":
 "49551135218688818456679503831981458784591352702 18EXAMPLE"
        }
      }
    ]
  }
}
```

Es kann einige Minuten dauern, bis der Stream im aktiven Status angezeigt wird.

3. Erstellen Sie die IAM-Rolle, die CloudWatch Logs die Berechtigung erteilt, Daten in den Kinesis-Stream zu übertragen. Sie müssen zunächst eine Vertrauensrichtlinie in einer Datei ~/TrustPolicyForCWL.json erstellen. Erstellen Sie in einem Text-Editor die Richtliniendatei, verwenden Sie nicht die IAM-Konsole.

```
{
  "Statement": {
    "Effect": "Allow",
    "Principal": { "Service": "logs.region.amazonaws.com" },
    "Action": "sts:AssumeRole"
  }
}
```

4. Verwenden Sie den Befehl aws iam create-role, um die IAM-Rolle zu erstellen und die Vertrauensrichtlinie anzugeben. Notieren Sie den ausgegebenen Wert "Role.Arn", da er später an CloudWatch Logs übergeben wird:

```
aws iam create-role \
    --role-name CWLtoKinesisRole \
    --assume-role-policy-document file://~/TrustPolicyForCWL.json

{
    "Role": {
        "AssumeRolePolicyDocument": {
            "Statement": {
                "Action": "sts:AssumeRole",
                "Effect": "Allow",
                "Principal": {
                    "Service": "logs.region.amazonaws.com"
                }
            }
        },
        "RoleId": "AAOIIAH450GAB4HC5F431",
        "CreateDate": "2015-05-29T13:46:29.431Z",
        "RoleName": "CWLtoKinesisRole",
        "Path": "/",
        "Arn": "arn:aws:iam::999999999999:role/CWLtoKinesisRole"
    }
}
```

5. Erstellen Sie eine Berechtigungsrichtlinie, um zu definieren, welche Aktionen CloudWatch Logs an Ihrem Konto durchführen darf. Erstellen Sie zunächst in einem Text-Editor eine Berechtigungsrichtlinie in einer Datei ~/PermissionsForCWL.json:

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:region:999999999999:stream/RecipientStream"
    }
  ]
}
```

6. Verknüpfen Sie die Berechtigungsrichtlinie mit der Rolle, und verwenden Sie dazu den Befehl aws iam put-role-policy:

```
aws iam put-role-policy \
    --role-name CWLtoKinesisRole \
    --policy-name Permissions-Policy-For-CWL \
    --policy-document file://~/PermissionsForCWL.json
```

7. Wenn sich der Kinesis-Stream im Status "Aktiv" befindet und Sie die IAM-Rolle erstellt haben, können Sie das CloudWatch Logs-Ziel erstellen.

   a. In diesem Schritt wird keine Zugriffsrichtlinie mit Ihrem Ziel verknüpft. Es ist zudem erst der erste von zwei Schritten zum Erstellen eines Ziels. Notieren Sie sich den in der Nutzlast ausgegebenen Wert DestinationArn:

```
aws logs put-destination \
    --destination-name "testDestination" \
    --target-arn "arn:aws:kinesis:region:999999999999:stream/RecipientStream" \
    --role-arn "arn:aws:iam::999999999999:role/CWLtoKinesisRole"

{
  "DestinationName" : "testDestination",
  "RoleArn" : "arn:aws:iam::999999999999:role/CWLtoKinesisRole",
  "DestinationArn" : "arn:aws:logs:us-
east-1:999999999999:destination:testDestination",
  "TargetArn" : "arn:aws:kinesis:us-east-1:999999999999:stream/RecipientStream"
}
```

   b. Verknüpfen Sie, nachdem Schritt 7a abgeschlossen ist, im Empfängerkonto der Protokolldaten eine Zugriffsrichtlinie mit dem Ziel. Diese Richtlinie ermöglicht dem Senderkonto der Protokolldaten (111111111111) auf das Ziel im Empfängerkonto der Protokolldaten (999999999999)zuzugreifen. Sie können diese Richtlinie mit einem Text-Editor in der Datei ~/AccessPolicy.json speichern:

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111111111111"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:region:999999999999:destination:testDestination"
    }
  ]
}
```

Note

Wenn mehrere Konten Protokolle an dieses Ziel senden, muss jedes Absenderkonto separat in der Richtlinie aufgeführt werden. Diese Richtlinie unterstützt nicht die Angabe `*` als `Principal` oder die Verwendung des globalen Schlüssels `aws:PrincipalOrgId`.

c. Damit wird eine Richtlinie erstellt, die bestimmt, wer Schreibzugriff auf das Ziel hat. In dieser Richtlinie muss die Aktion logs:PutSubscriptionFilter für den Zugriff auf das Ziel angegeben sein. Kontoübergreifende Benutzer verwenden die Aktion PutSubscriptionFilter, um Protokollereignisse an das Ziel zu senden:

```
aws logs put-destination-policy \
    --destination-name "testDestination" \
    --access-policy file://~/AccessPolicy.json
```

Diese Zugriffsrichtlinie ermöglicht die Verwendung von Benutzern im AWS-Konto mit ID 111111111111 PutsubscriptionFilter gegen das Ziel mit ARN:aws:logs:*region*: 999999999999:destination:testdestination. Die Versuche aller anderen Benutzer, PutSubscriptionFilter für dieses Ziel aufzurufen, werden zurückgewiesen.

Informationen dazu, wie Sie die Berechtigungen eines Benutzers in einer Zugriffsrichtlinie prüfen, finden Sie unter Verwenden der Richtlinienvalidierung im IAM-Benutzerhandbuch.

Erstellen eines Abonnementfilters

Wenn Sie ein Ziel erstellt haben, kann das Empfängerkonto der Protokolldaten den Ziel-ARN (arn:aws:logs:us-east-1:999999999999:destination:testDestination) für andere AWS-Konten freigeben, damit diese Protokollereignisse an dasselbe Ziel senden können. Die anderen sendenden Kontobenutzer erstellen einen Abonnementfilter für die jeweiligen Protokollgruppen für dieses Ziel. Der Abonnementfilter startet sofort den Fluss der Echtzeit-Protokolldaten aus der gewählten Protokollgruppe an das angegebene Ziel.

Im folgenden Beispiel wird ein Abonnementfilter in einem sendenden Konto erstellt. Der Filter ist einer Protokollgruppe zugeordnet, die AWS CloudTrail-Ereignisse enthält, sodass alle protokollierten Aktivitäten von "Root"-AWS-Anmeldeinformationen an das zuvor erstelle Ziel übermittelt werden. Dieses Ziel kapselt einen Kinesis-Stream mit dem Namen "RecipientStream". Weitere Informationen zum Senden von AWS CloudTrail-Ereignissen an CloudWatch Logs finden Sie unter Senden von CloudTrail-Ereignisreferenz an CloudWatch Logs im AWS CloudTrail User Guide.

```
aws logs put-subscription-filter \
    --log-group-name "CloudTrail" \
    --filter-name "RecipientStream" \
    --filter-pattern "{$.userIdentity.type = Root}" \
    --destination-arn "arn:aws:logs:region:999999999999:destination:testDestination"
```

Die Log-Gruppe und das Ziel müssen sich in derselben AWS-Region befinden. Die Zieladresse kann jedoch auf eine AWS-Ressource wie ein Kinesis-Stream verweisen, der sich in einer anderen Region befindet.

Validierung des Flusses von Protokollereignissen

Wenn Sie einen Abonnementfilter erstellt haben, leitet CloudWatch Logs alle eingehenden Protokollereignisse, die dem Filtermuster entsprechen, an den Kinesis-Stream weiter, der im Ziel-Stream mit dem Namen "RecipientStream" enthalten ist. Der Eigentümer des Ziels kann überprüfen, dass dies

geschieht, indem er mit dem Befehl aws kinesis get-shard-iterator eine Kinesis-Shard abruft und mit dem Befehl aws kinesis get-records einige Kinesis-Datensätze abruft:

```
aws kinesis get-shard-iterator \
      --stream-name RecipientStream \
      --shard-id shardId-000000000000 \
      --shard-iterator-type TRIM_HORIZON

{
    "ShardIterator":
    "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
}

aws kinesis get-records \
      --limit 10 \
      --shard-iterator
      "AAAAAAAAAAFGU/
kLvNggvndHq2UIFOw5PZc6F01s3e3afsSscRM70JSbjIefg2ub07nk1y6CDxYR1UoGHJNP4m4NFUetzfL+wev
+e2P4djJg4L9wmXKvQYoE+rMUiFq+p4Cn3IgvqOb5dRA0yybNdRcdzvnC35KQANoHzzahKdRGb9v4scv+3vaq+f
+OIK8zM5My8ID+g6rMo7UKWeI4+IWiKEXAMPLE"
```

> **Note**
>
> Kurz bevor Kinesis mit der Rückgabe von Daten beginnt, müssen Sie den Befehl "get-records" möglicherweise erneut ausführen.

Sie sollten eine Antwort mit einem Array von Kinesis-Datensätze sehen. Das Datenattribut im Kinesis-Datensatz ist im GZIP-Format komprimiert und dann Base64-kodiert. Sie können die Rohdaten über die Befehlszeile mit dem folgenden Unix-Befehl überprüfen:

```
echo -n "<Content of Data>" | base64 -d | zcat
```

Die dekodierten und dekomprimierten Base64-Daten sind als JSON mit folgender Struktur formatiert:

```
{
    "owner": "111111111111",
    "logGroup": "CloudTrail",
    "logStream": "111111111111_CloudTrail_us-east-1",
    "subscriptionFilters": [
        "RecipientStream"
    ],
    "messageType": "DATA_MESSAGE",
    "logEvents": [
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root\"}"
        },
        {
            "id": "31953106606966983378809025079804211143289615424 29EXAMPLE",
            "timestamp": 1432826855000,
            "message": "{\"eventVersion\":\"1.03\",\"userIdentity\":{\"type\":\"Root\"}"
        }
    ]
```

```
}
```

Die wichtigsten Elemente in dieser Datenstruktur lauten wie folgt:

owner

Die ID des AWS-Kontos der ursprünglichen Protokolldaten.

logGroup

Der Name der Protokollgruppe der ursprünglichen Protokolldaten.

logStream

Der Name des Protokoll-Stream der ursprünglichen Protokolldaten.

subscriptionFilters

Die Namenliste der Abonnementfilter, die mit den ursprünglichen Protokolldaten übereingestimmt haben.

messageType

Datennachrichten verwenden den Typ "DATA_MESSAGE". In einigen Fällen kann CloudWatch Logs Kinesis-Datensätze mit dem Typ "CONTROL_MESSAGE" ausgeben. Sie dienen hauptsächlich zur Prüfung, ob das Ziel erreichbar ist.

logEvents

Die tatsächlichen Protokolldaten, die als Array von Protokollereignis-Datensätzen dargestellt werden. Die ID-Eigenschaft ist eine eindeutige Kennung für jedes Protokollereignis.

Ändern der Mitgliedschaft im Ziel zur Laufzeit

In manchen Situationen müssen Sie die Mitgliedschaft einiger Benutzer in einem Ziel, das Ihr Eigentum ist, hinzufügen oder entfernen. Sie können die Aktion PutDestinationPolicy für Ihr Ziel mit einer neuen Zugriffsrichtlinie verwenden. Im folgenden Beispiel wird festgelegt, dass das zuvor hinzugefügte Konto 111111111111 keine Protokolldaten mehr sendet und das Konto 222222222222 aktiviert wird.

1. Rufen Sie die Richtlinie ab, die derzeit mit dem Ziel testDestination verknüpft ist, und notieren Sie die AccessPolicy:

```
aws logs describe-destinations \
    --destination-name-prefix "testDestination"

{
 "Destinations": [
   {
     "DestinationName": "testDestination",
     "RoleArn": "arn:aws:iam::222222222222:role/CWLtoKinesisRole",
     "DestinationArn": "arn:aws:logs:region:222222222222:destination:testDestination",
     "TargetArn": "arn:aws:kinesis:region:222222222222:stream/RecipientStream",
     "AccessPolicy": "{\"Version\": \"2012-10-17\", \"Statement\":
 [{\"Sid\": \"\", \"Effect\": \"Allow\", \"Principal\": {\"AWS\":
 \"111111111111\"}, \"Action\": \"logs:PutSubscriptionFilter\", \"Resource\":
 \"arn:aws:logs:region:123456789012:destination:testDestination\"}] }"
   }
 ]
}
```

2. Aktualisieren Sie die Richtlinie, sodass angezeigt wird, dass das Konto 111111111111 angehalten wurde und das Konto 222222222222 aktiviert wurde. Speichern Sie diese Richtlinie in der Datei ~/NewAccessPolicy.json:

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "222222222222"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:region:222222222222:destination:testDestination"
    }
  ]
}
```

3. Rufen Sie PutDestinationPolicy auf, um die in der Datei NewAccessPolicy.json definierte Richtlinie mit dem folgenden Ziel zu verknüpfen:

```
aws logs put-destination-policy \
--destination-name "testDestination" \
--access-policy file://~/NewAccessPolicy.json
```

Damit werden schließlich die Protokollereignisse von der Konto-ID 111111111111 deaktiviert. Protokollereignisse der Konto-ID 222222222222 werden an das Ziel übertragen, sobald der Eigentümer des Kontos 222222222222 mit PutSubscriptionFilter einen Abonnementfilter erstellt.

Senden von Protokollen direkt an Amazon S3 oder Kinesis Data Firehose

Einige AWS -Services können Protokolle direkt in veröffentlichen Amazon S3 oder Kinesis Data Firehose. Auf diese Weise können Sie den -Service, der die Protokolle direkt an sendet, einfach an senden lassen, wenn Ihre Hauptanforderung für Protokolle die Speicherung oder Verarbeitung in einem dieser Services ist Amazon S3 oder Kinesis Data Firehose ohne zusätzliche Infrastruktur einzurichten.

Protokolle, die in Amazon S3 veröffentlicht werden, werden in einen Bucket veröffentlicht, den Sie zuvor angegeben haben. Alle fünf Minuten werden ein oder mehrere Protokolldateien in dem angegebenen Bucket erstellt.

Auch wenn Protokolle direkt in veröffentlicht werden Amazon S3 oder Kinesis Data Firehose, , und Sie haben die Möglichkeit CloudWatch Logs Gebühren fallen an. Weitere Informationen finden Sie unter Gekaufte Protokolle auf dem Protokolle Registerkarte Amazon CloudWatch Preisgestaltung.

Die folgenden Protokolle können direkt in Amazon S3 veröffentlicht werden:

- VPC Flow Logs Weitere Informationen finden Sie unter Veröffentlichung von Flow Logs in Amazon S3 im Amazon VPC Benutzerhandbuch.
- AWS Global Accelerator Flow-Protokolle. Weitere Informationen finden Sie unter Publizieren von Flow-Protokollen in Amazon S3 im AWS Global Accelerator Developer Guide.

Die folgenden Protokolle können direkt in Kinesis Data Firehose veröffentlicht werden:

- Amazon Managed Streaming for Apache Kafka-Protokolle Weitere Informationen finden Sie unter Protokollierung in der Amazon Managed Streaming for Apache Kafka Entwicklerhandbuch.

Exportieren von Protokolldaten in Amazon S3

Sie können Protokolldaten aus den Protokollgruppen in einen Amazon S3-Bucket exportieren und diese Daten für benutzerdefinierte Prozesse und Analysen verwenden oder in andere Systeme laden.

Exportieren von Protokolldaten nach Amazon S3 -Buckets, die von verschlüsselt sind AWS KMS wird nicht unterstützt.

Um den Exportprozess zu beginnen, müssen Sie einen S3-Bucket zum Speichern der exportierten Protokolldaten erstellen. Sie können die exportierten Dateien im Amazon S3-Bucket speichern und Amazon S3-Lebenszyklusregeln definieren, mit denen die exportierten Dateien automatisch archiviert oder gelöscht werden.

Der Export in S3-Buckets, die mit AES-256 verschlüsselt sind, wird unterstützt. Der Export in S3-Buckets, die mit SSE-KMS verschlüsselt sind, wird nicht unterstützt. Weitere Informationen finden Sie unter Wie aktiviere ich die Standardverschlüsselung für einen S3-Bucket?

Sie können Protokolle aus mehreren Protokollgruppen oder mehreren Zeitbereichen in den gleichen S3-Bucket exportieren. Um die Protokolldaten für jeden Exportvorgang voneinander zu trennen, können Sie ein Präfix angeben, das als Amazon S3-Schlüsselpräfix für alle exportierten Objekten verwendet wird.

Protokolldaten können bis zu 12 Stunden für den Export verfügbar sein. Informationen zu Quasi-Echtzeit-Analysen von Protokolldaten finden Sie hingegen unter Analysieren von Protokolldaten mit CloudWatch Logs Insights (p. 36) oder Protokolldaten-Verarbeitung in Echtzeit mit Abonnements (p. 91).

> **Note**
>
> Ab dem 15. Februar 2019 muss beim Aufrufen der Funktion für den Export nach Amazon S3 der `s3:PutObject`-Zugriff auf den Ziel-Bucket vorliegen.

Inhalt:

- Concepts (p. 112)
- Exportieren von Protokolldaten in Amazon S3 mit der Konsole (p. 113)
- Exportieren von Protokolldaten mit der AWS CLI in Amazon S3 (p. 116)

Concepts

Bevor Sie beginnen, sollten Sie sich mit folgenden Exportkonzepten vertraut machen:

Name der Protokollgruppe

: Der Name der Protokollgruppe für einen Exportvorgang. Die Protokolldaten in dieser Protokollgruppe werden in den angegebenen Amazon S3-Bucket exportiert.

von (Zeitstempel)

: Ein erforderlicher Zeitstempel in Millisekunden seit dem 1. Januar 1970 00:00:00 UTC. Alle Protokollereignisse in der Protokollgruppe, die nach diesem Zeitpunkt aufgenommen wurden, werden exportiert.

bis (Zeitstempel)

: Ein erforderlicher Zeitstempel in Millisekunden seit dem 1. Januar 1970 00:00:00 UTC. Alle Protokollereignisse in der Protokollgruppe, die vor diesem Zeitpunkt aufgenommen wurden, werden exportiert.

Ziel-Bucket

Der Name des Amazon S3-Bucket für einen Exportvorgang. Dieses Bucket dient zum Exportieren von Protokolldaten aus der angegebenen Protokollgruppe.

Ziel-Präfix

Ein optionales Attribut, das als S3-Schlüsselpräfix für alle exportierten Objekte verwendet wird. Dadurch können Sie eine Art Ordnerstruktur in Ihrem Bucket erstellen.

Exportieren von Protokolldaten in Amazon S3 mit der Konsole

Im folgenden Beispiel verwenden Sie die Amazon CloudWatch-Konsole für den Export aller Daten aus einer Amazon CloudWatch Logs-Protokollgruppe mit dem Namen `my-log-group` in einen Amazon S3-Bucket mit dem Namen `my-exported-logs`.

Exportieren von Protokolldaten nach Amazon S3 -Buckets, die von verschlüsselt sind AWS KMS wird nicht unterstützt.

Schritt 1 Erstellen eines Amazon S3-Buckets

Wir empfehlen, dass Sie einen Bucket verwenden, der speziell für CloudWatch Logs erstellt wurde. Wenn Sie jedoch einen vorhandenen Bucket verwenden möchten, gehen Sie direkt zu Schritt 2.

Note

Der Amazon S3-Bucket muss sich in derselben Region wie die Protokolldaten befinden, die exportiert werden sollen. CloudWatch Logs unterstützt kein Exportieren von Daten in Amazon S3-Buckets in einer anderen Region.

So erstellen Sie einen Amazon S3-Bucket

1. Öffnen Sie die Amazon S3-Konsole unter der Adresse https://console.aws.amazon.com/s3/.
2. Ändern Sie, falls erforderlich, die Region. Wählen Sie auf der Navigationsleiste die Region aus, in der sich CloudWatch Logs befindet.
3. Klicken Sie auf Create Bucket (Bucket erstellen).
4. Geben Sie unter Bucket-Name einen Namen für den Bucket ein.
5. Wählen Sie für Region die Region aus, in der sich Ihre CloudWatch Logs-Daten befinden.
6. Wählen Sie Create (Erstellen) aus.

Schritt 2 Erstellen eines IAM Benutzer mit Vollzugriff auf Amazon S3 und CloudWatch Logs

In den folgenden Schritten erstellen Sie den IAM-Benutzer mit den erforderlichen Berechtigungen.

So erstellen Sie den erforderlichen IAM-Benutzer

1. Öffnen Sie die IAM-Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie Benutzer, Benutzer hinzufügen.
3. Geben Sie einen Benutzernamen ein, z. B. *`CWLExportUser`*.
4. Wählen Sie die beiden Optionen Programmatic access (Programmgesteuerter Zugriff) und AWS Management Console access (Zugriff auf die AWS-Managementkonsole) aus.

5. Wählen Sie entweder Autogenerated password (Automatisch generiertes Passwort) oder Custom password (Benutzerdefiniertes Passwort) aus.
6. Wählen Sie Next (weiter). Permissions (Berechtigungen)
7. Wählen Sie Attach existing policies directly (Vorhandene Richtlinien direkt zuordnen) aus und fügen Sie dem Benutzer die Richtlinien `AmazonS3FullAccess` und `CloudWatchLogsFullAccess` hinzu. Sie können im Suchfeld nach Richtlinien suchen.
8. Wählen Sie Next (weiter). Kennzeichnungen, , und Sie haben die Möglichkeit Als Nächstes: Überprüfungund dann Benutzer erstellen.

Schritt 3 Festlegen von Berechtigungen für eine Amazon S3 Behälter

Standardmäßig werden alle Amazon S3-Buckets und Objekte als privat eingestuft. Nur der Ressourceneigentümer, das AWS-Konto, in dem der Bucket erstellt wurde, kann auf den Bucket und alle darin enthaltenen Objekte zugreifen. Der Ressourcenbesitzer kann jedoch anderen Ressourcen und Benutzern Zugriffsberechtigungen gewähren, indem er eine Zugriffsrichtlinie schreibt.

Wenn Sie die Richtlinie festlegen, empfehlen wir Ihnen, eine zufällig generierte Zeichenfolge als Präfix für den Bucket einzufügen, so dass nur die beabsichtigten Protokoll-Streams in den Bucket exportiert werden.

So legen Sie Berechtigungen für einen Amazon S3-Bucket fest

1. Wählen Sie in der Amazon S3-Konsole den Bucket aus, den Sie in Schritt 1 erstellt haben.
2. Klicken Sie auf Permissions (Berechtigungen), Bucket policy (Bucket-Richtlinie).
3. Fügen Sie eine der folgenden Richtlinien unter Bucket Policy Editor (Bucket-Richtlinien-Editor) hinzu. Ändern Sie `my-exported-logs` in den Namen Ihres S3-Buckets und `random-string` in eine Zufallszeichenfolge. Stellen Sie sicher, dass für Principal (Prinzipal) der richtige Endpunkt für die Region angegeben ist.

   - Wenn sich der Bucket in Ihrem Konto befindet, fügen Sie die folgende Richtlinie hinzu.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
          "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
          "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
      }
    ]
}
```

   - Wenn sich der Bucket in einem anderen Konto befindet, verwenden Sie stattdessen die folgende Richtlinie. Sie enthält eine zusätzliche Anweisung unter Verwendung des im vorherigen Schritts erstellten IAM-Benutzers.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
          "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
          "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
          "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
          "Principal": { "AWS": "arn:aws:iam::SendingAccountID:user/CWLExportUser" }
      }
    ]
}
```

4. Wählen Sie Save aus, um die Richtlinie, die Sie gerade hinzugefügt haben, als Zugriffsrichtlinie für den Bucket festzulegen. Aufgrund dieser Richtlinie kann CloudWatch Logs Protokolldaten in den Amazon S3-Bucket exportieren. Der Bucket-Eigentümer hat vollen Zugriff auf alle exportierten Objekte.

   Warning

   Sind dem vorhandenen Bucket bereits ein oder mehrere Richtlinien angefügt, fügen Sie für den CloudWatch Logs-Zugriff auf diese Richtlinie oder auf Richtlinien die Anweisungen hinzu. Sie sollten eine Beurteilung der daraus resultierenden Berechtigungen vornehmen, um sicherzustellen, dass sie für die Benutzer, die auf den Bucket zugreifen werden, geeignet sind.

Schritt 4. Erstellen einer Exportaufgabe

In diesem Schritt erstellen Sie die Exportaufgabe zum Exportieren von Protokollen aus einer Protokollgruppe.

So exportieren Sie Daten mithilfe der CloudWatch-Konsole in Amazon S3

1. Melden Sie sich als an IAM Benutzer, den Sie in erstellt haben Schritt 2: Erstellen eines IAM Benutzer mit Vollzugriff auf Amazon S3 und CloudWatch Logs.
2. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
3. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus.
4. Wählen Sie im Bildschirm Protokollgruppen den Namen der Protokollgruppe aus.
5. Wählen Sie Aktionen, Daten in Amazon S3 exportieren aus.
6. Legen Sie im Bildschirm Daten in Amazon S3 exportieren unter Datenexport definieren den Zeitbereich für die zu exportierenden Daten mit Von und Bis fest.
7. Wenn Ihre Protokollgruppe über mehrere Protokoll-Streams verfügt, können Sie ein Protokoll-Stream-Präfix angeben, um die Protokollgruppedaten an einen bestimmten Stream zu beschränken. Wählen Sie Advanced (Erweitert) aus und geben Sie für Stream prefix (Stream-Präfix) das Protokoll-Stream-Präfix ein.

8. Wählen Sie unter Choose S3 bucket (S3-Bucket auswählen) das Konto für den Amazon S3-Bucket aus.
9. Wählen Sie für S3 bucket name (S3-Bucket-Name) einen Amazon S3-Bucket aus.
10. Geben Sie für S3-Bucket-Präfix die zufällig generierte Zeichenfolge ein, die Sie in der Bucket-Richtlinie angegeben haben.
11. Wählen Sie Exportieren zum Exportieren von Protokolldaten zu Amazon S3 aus.
12. Um den Status der Protokolldaten anzuzeigen, die Sie in Amazon S3 exportiert haben, wählen Sie Actions (Aktionen) und dann View all exports to Amazon S3 (Alle Exporte in S3 anzeigen) aus.

Exportieren von Protokolldaten mit der AWS CLI in Amazon S3

Im folgenden Beispiel verwenden Sie eine Exportaufgabe, um alle Daten aus einem CloudWatch Logs Protokollgruppe mit dem Namen `my-log-group` zu einem Amazon S3 Bucket mit dem Namen `my-exported-logs`. (z. B.. In diesem Beispiel wird davon ausgegangen, dass Sie bereits eine Protokollgruppe mit dem Namen `my-log-group`.

Exportieren von Protokolldaten nach Amazon S3 -Buckets, die von verschlüsselt sind AWS KMS wird nicht unterstützt.

Schritt 1 Erstellen eines Amazon S3-Buckets

Wir empfehlen, dass Sie einen Bucket verwenden, der speziell für CloudWatch Logs erstellt wurde. Wenn Sie jedoch einen vorhandenen Bucket verwenden möchten, gehen Sie direkt zu Schritt 2.

> Note
>
> Der Amazon S3-Bucket muss sich in derselben Region wie die Protokolldaten befinden, die exportiert werden sollen. CloudWatch Logs unterstützt kein Exportieren von Daten in Amazon S3-Buckets in einer anderen Region.

So erstellen Sie einen Amazon S3-Bucket mit der AWS CLI

Führen Sie an der Eingabeaufforderung den Befehl create-bucket aus, wobei `LocationConstraint` die Region ist, in der Sie Protokolldaten exportieren.

```
aws s3api create-bucket --bucket my-exported-logs --create-bucket-configuration
 LocationConstraint=us-east-2
```

Im Folgenden finden Sie eine Beispielausgabe.

```
{
    "Location": "/my-exported-logs"
}
```

Schritt 2 Erstellen eines IAM Benutzer mit Vollzugriff auf Amazon S3 und CloudWatch Logs

In den folgenden Schritten erstellen Sie den IAM-Benutzer mit den erforderlichen Berechtigungen.

So erstellen Sie den Benutzer und weisen ihm Berechtigungen zu

1. Erstellen Sie den IAM-Benutzer mit dem folgenden Befehl.

```
aws iam create-user --user-name CWLExportUser
```

2. Verknüpfen Sie die verwalteten IAM-Richtlinien mit dem soeben erstellten IAM-Benutzer.

```
export S3POLICYARN=$(aws iam list-policies --query 'Policies[?
PolicyName==`AmazonS3FullAccess`].{ARN:Arn}' --output text)
```

```
export CWLPOLICYARN=$( aws iam list-policies --query 'Policies[?
PolicyName==`CloudWatchLogsFullAccess`].{ARN:Arn}' --output text)
```

```
aws iam attach-user-policy --user-name CWLExportUser --policy-arn $S3POLICYARN
```

```
aws iam attach-user-policy --user-name CWLExportUser --policy-arn $CWLPOLICYARN
```

3. Vergewissern Sie sich, dass die zwei verwalteten Richtlinien verknüpft sind.

```
aws iam list-attached-user-policies --user-name CWLExportUser
```

4. Konfigurieren Sie Ihre AWS CLI um die IAM Anmeldeinformationen der *CWLExportUser* IAM Benutzer Weitere Informationen finden Sie unter Konfigurieren der AWS CLI.

Schritt 3 Festlegen von Berechtigungen für eine Amazon S3 Behälter

Standardmäßig werden alle Amazon S3-Buckets und Objekte als privat eingestuft. Nur der Ressourceneigentümer, das Konto, in dem der Bucket erstellt wurde, kann auf den Bucket und alle darin enthaltenen Objekte zugreifen. Der Ressourcenbesitzer kann jedoch anderen Ressourcen und Benutzern Zugriffsberechtigungen gewähren, indem er eine Zugriffsrichtlinie schreibt.

So legen Sie Berechtigungen für einen Amazon S3-Bucket fest

1. Erstellen Sie eine Datei mit dem Namen `policy.json`, und fügen Sie folgende Zugriffsrichtlinie hinzu. Ändern Sie dabei `Resource` in den Namen Ihres S3-Buckets und `Principal` in den Endpunkt der Region, in der Sie Protokolldaten exportieren. Verwenden Sie einen Text-Editor, um diese Richtliniendatei zu erstellen. Verwenden Sie nicht die IAM-Konsole.

   - Wenn sich der Bucket in Ihrem Konto befindet, benutzen Sie die folgende Richtlinie.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "s3:GetBucketAcl",
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs",
            "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
        },
        {
            "Action": "s3:PutObject" ,
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::my-exported-logs/*",
            "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
            "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
```

```
        }
    ]
}
```

- Wenn sich der Bucket in einem anderen Konto befindet, verwenden Sie stattdessen die folgende Richtlinie. Sie enthält eine zusätzliche Anweisung unter Verwendung des im vorherigen Schritts erstellten IAM-Benutzers.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
          "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
          "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
          "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
          "Principal": { "AWS": "arn:aws:iam::SendingAccountID:user/CWLExportUser" }
      }
    ]
}
```

- Wenn sich der Bucket in einem anderen Konto befindet und Sie eine IAM-Rolle anstelle eines IAM-Benutzers verwenden, nutzen Sie stattdessen die folgende Richtlinie.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
          "Action": "s3:GetBucketAcl",
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs",
          "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
          "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
          "Principal": { "Service": "logs.us-west-2.amazonaws.com" }
      },
      {
          "Action": "s3:PutObject" ,
          "Effect": "Allow",
          "Resource": "arn:aws:s3:::my-exported-logs/random-string/*",
          "Condition": { "StringEquals": { "s3:x-amz-acl": "bucket-owner-full-
control" } },
          "Principal": { "AWS": "arn:aws:iam::SendingAccountID:role/CWLExportUser" }
      }
```

```
    ]
}
```

2. Legen Sie mithilfe des Befehls put-bucket-policy die soeben hinzugefügte Richtlinie als Zugriffsrichtlinie für Ihren Bucket fest. Aufgrund dieser Richtlinie kann CloudWatch Logs Protokolldaten in den Amazon S3-Bucket exportieren. Der Bucket-Eigentümer hat vollen Zugriff auf alle exportierten Objekte.

```
aws s3api put-bucket-policy --bucket my-exported-logs --policy file://policy.json
```

   Warning

   Sind dem vorhandenen Bucket bereits ein oder mehrere Richtlinien angefügt, fügen Sie für den CloudWatch Logs-Zugriff auf diese Richtlinie oder auf Richtlinien die Anweisungen hinzu. Sie sollten eine Beurteilung der daraus resultierenden Berechtigungen vornehmen, um sicherzustellen, dass sie für die Benutzer, die auf den Bucket zugreifen werden, geeignet sind.

Schritt 4. Erstellen einer Exportaufgabe

Nach dem Erstellen der Exportaufgabe für den Export von Protokollen aus einer Protokollgruppe, kann der Exportvorgang von einigen Sekunden bis zu einigen Stunden dauern, abhängig von der Größe der zu exportierenden Daten.

So erstellen Sie einen Exportvorgang mithilfe der AWS CLI

Geben Sie an einer Eingabeaufforderung den folgenden Befehl create-export-task zum Erstellen der Exportaufgabe ein.

```
aws logs create-export-task --profile CWLExportUser --task-name "my-log-group-09-10-2015"
 --log-group-name "my-log-group" --from 1441490400000 --to 1441494000000 --destination "my-
exported-logs" --destination-prefix "export-task-output"
```

Im Folgenden finden Sie eine Beispielausgabe.

```
{
    "taskId": "cda45419-90ea-4db5-9833-aade86253e66"
}
```

Schritt 5. Exportaufgaben beschreiben

Nachdem Sie eine Exportaufgabe erstellt haben, können Sie den aktuellen Status der Aufgabe abrufen.

So beschreiben Sie Exportaufgaben mithilfe der AWS CLI

Geben Sie an der Eingabeaufforderung den folgenden Befehl describe-export-tasks ein.

```
aws logs --profile CWLExportUser describe-export-tasks --task-id "cda45419-90ea-4db5-9833-
aade86253e66"
```

Im Folgenden finden Sie eine Beispielausgabe.

```
{
   "exportTasks": [
   {
```

```
      "destination": "my-exported-logs",
      "destinationPrefix": "export-task-output",
      "executionInfo": {
         "creationTime": 1441495400000
      },
      "from": 1441490400000,
      "logGroupName": "my-log-group",
      "status": {
         "code": "RUNNING",
         "message": "Started Successfully"
      },
      "taskId": "cda45419-90ea-4db5-9833-aade86253e66",
      "taskName": "my-log-group-09-10-2015",
      "tTo": 1441494000000
   }]
}
```

Sie können den `describe-export-tasks`-Befehl auf drei verschiedene Arten verwenden:

- Ohne Filter: Listet alle Ihre Exportaufgaben in umgekehrter Reihenfolge der Erstellung auf.
- Nach Aufgaben-ID filtern: Listet die Exportaufgabe, falls vorhanden, mit der angegebenen ID auf.
- Nach Aufgabenstatus filtern: Listet die Exportaufgaben mit dem angegebenen Status auf.

Verwenden Sie z. B. den folgenden Befehl, um einen Filter für den Status `FAILED` zu nutzen.

```
aws logs --profile CWLExportUser describe-export-tasks --status-code "FAILED"
```

Im Folgenden finden Sie eine Beispielausgabe.

```
{
   "exportTasks": [
   {
      "destination": "my-exported-logs",
      "destinationPrefix": "export-task-output",
      "executionInfo": {
         "completionTime": 1441498600000
         "creationTime": 1441495400000
      },
      "from": 1441490400000,
      "logGroupName": "my-log-group",
      "status": {
         "code": "FAILED",
         "message": "FAILED"
      },
      "taskId": "cda45419-90ea-4db5-9833-aade86253e66",
      "taskName": "my-log-group-09-10-2015",
      "to": 1441494000000
   }]
}
```

Schritt 6. Eine Exportaufgabe abbrechen

Sie können eine Exportaufgabe stornieren, wenn sie entweder den Status `PENDING` oder `RUNNING` aufweist.

So stornieren Sie eine Exportaufgabe mithilfe der AWS CLI

Geben Sie an der Eingabeaufforderung den folgenden cancel-export-task-Befehl ein:

```
aws logs --profile CWLExportUser cancel-export-task --task-id "cda45419-90ea-4db5-9833-aade86253e66"
```

Sie können den Befehl describe-export-tasks verwenden, um zu überprüfen, ob die Aufgabe erfolgreich abgebrochen wurde.

CloudWatch Logs-Daten in Amazon Elasticsearch Service streamen

Sie können eine CloudWatch Logs-Protokollgruppe zum Streamen der empfangenen Daten im Amazon Elasticsearch Service (Amazon ES)-Cluster nahezu in Echtzeit über ein CloudWatch Logs-Abonnement konfigurieren. Weitere Informationen finden Sie unter Protokolldaten-Verarbeitung in Echtzeit mit Abonnements (p. 91).

Abhängig von der Menge der gestreamten Protokolldaten möchten Sie möglicherweise eine Begrenzung der gleichzeitigen Ausführung der Funktion auf Funktionsebene festlegen. Weitere Informationen finden Sie unter Begrenzung der gleichzeitigen Ausführung der Funktion auf Funktionsebene.

> Note
>
> Das Streamen großer Mengen an CloudWatch Logs-Daten in Amazon ES kann zu hohen nutzungsabhängigen Gebühren führen. Wir empfehlen, dass Sie ein Budget in der Fakturierungs- und Kostenverwaltungskonsole erstellen. Weitere Informationen finden Sie unter Verwalten der Kosten mit Budgets.

Voraussetzungen

Bevor Sie beginnen, erstellen Sie eine Amazon ES-Domäne. Die Amazon ES-Domäne kann entweder öffentlichen Zugriff oder VPC-Zugriff haben, aber Sie können den Zugriffstyp dann nicht ändern, nachdem die Domäne erstellt wurde. Sie sollten die Amazon ES-Domäneneinstellungen später überprüfen und die Cluster-Konfiguration basierend auf der verarbeiteten Datenmenge ändern.

Weitere Informationen zu Amazon ESfinden Sie im Amazon Elasticsearch Service-Entwicklerhandbuch.

So erstellen Sie eine Amazon ES-Domäne

Geben Sie als Eingabeaufforderung den folgenden Befehl create-elasticsearch-domain ein:

```
aws es create-elasticsearch-domain --domain-name my-domain
```

Abonnieren einer Protokollgruppe für Amazon ES

Sie können die CloudWatch-Konsole verwenden, um eine Protokollgruppe für Amazon ES zu abonnieren.

Abonnieren einer Protokollgruppe für Amazon ES

1. Öffnen Sie die CloudWatch-Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im Navigationsbereich Log groups (Protokollgruppen) aus.
3. Wählen Sie den Namen der Protokollgruppe aus.
4. Wählen Sie Actions (Aktionen), Create Elastisearch subscription filter (Elastisearch-Abonnementfilter erstellen).
5. Wählen Sie aus, ob Sie zu einem Cluster in diesem Konto oder einem anderen Konto streamen möchten.

6. Wählen Sie für Amazon ES Cluster den Cluster aus, den Sie im vorherigen Schritt erstellt haben.
7. Wählen Sie unter Lambda Function (LAM-Funktion) für Lambda IAM Execution Role (LAM-Ausführungsrolle) die IAM-Rolle aus, die Sie in Lambda verwenden sollten, wenn Sie Aufrufe an Amazon ES ausführen, und wählen Sie dann Next (Weiter) aus.

   Die IAM-Rolle, die Sie auswählen, muss diese Anforderungen erfüllen:

   - Es muss `lambda.amazonaws.com` im Verhältnis stehen.
   - Die Richtlinie muss Folgendes umfassen:

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Action": [
                   "es:*"
               ],
               "Effect": "Allow",
               "Resource": "arn:aws:es:region:account-id:domain/target-domain-name/*"
           }
       ]
   }
   ```

   - Wenn die Ziel-Amazon ES-Domäne den VPC-Zugriff verwendet, muss der Rolle die Richtlinie AWSLambdaVPCAccessExecutionRole beigefügt sein. Diese von Amazon verwaltete Richtlinie gewährt Lambda-Zugriff auf die VPC des Kunden und ermöglicht Lambda den Amazon ES-Endpunkt in der VPC zu schreiben.
8. Wählen Sie unter Log Format (Protokollformat) ein Protokollformat aus.
9. Geben Sie unter Subscription Filter Pattern (Abonnementfiltermuster) die Begriffe oder Muster ein, die in Ihren Protokollereignissen zu finden sind. Auf diese Weise wird sichergestellt, dass Sie nur die Daten an das Amazon ES-Cluster senden, die für Sie von Interesse sind. Weitere Informationen finden Sie unter Erstellen von Metriken aus Protokollereignissen mithilfe von Filtern (p. 74).
10. (Optional) Wählen Sie unter Select Log Data to Test (Zu testende Protokolldaten auswählen) einen Protokoll-Stream und anschließend Test Pattern (Muster testen) aus, um zu überprüfen, ob Ihre Suchfilter die erwarteten Ergebnisse zurückgeben.
11. Wählen Sie dann Start Streaming (Streamen starten).

AWS-Services, die Protokolle in CloudWatch Logs veröffentlichen

Die folgenden AWS-Services veröffentlichen Metriken in CloudWatch Logs. Informationen zu den Protokollen, die diese Services senden, finden Sie in der verknüpften Dokumentation.

Service	Dokumentation:
Amazon API Gateway	Einrichten CloudWatch API-Anmeldung API Gateway
Amazon Aurora MySQL	Veröffentlichung Amazon Aurora MySQL Protokolliert in Amazon CloudWatch Logs
AWS CloudHSM	Überwachung AWS CloudHSM Prüfprotokolle in Amazon CloudWatch Logs
AWS CloudTrail	Überwachung CloudTrail Protokolldateien mit Amazon CloudWatch Logs
Amazon Cognito	Erstellen der CloudWatch Logs IAM-Rolle
Amazon Connect	Protokollierung und Überwachung Amazon Connect
AWS DataSync	Zulassend DataSync um Protokolle auf Amazon hochzuladen CloudWatch Protokollgruppen
AWS Elastic Beanstalk	Verwenden von Elastic Beanstalk mit dem Amazon CloudWatch Logs
Amazon Elastic Container Service	CloudWatch Logs mit Container-Instances verwenden
Amazon Elastic Kubernetes Service	Amazonas (Amazon Amazon Elastic Kubernetes Service Protokollierung der Steuerungsebene
AWS Fargate	Verwenden des awslogs-Protokolltreibers
AWS Glue	Kontinuierliche Protokollierung für AWS Glue-Aufträge
AWS IoT	Überwachung mit CloudWatch Logs
AWS Lambda	Zugriff auf Amazon CloudWatch Logs für AWS Lambda
Amazon MQ	Konfigurieren Amazon MQ zum Veröffentlichen von allgemeinen und Prüfprotokollen in Amazon CloudWatch Logs
AWS OpsWorks	Verwenden von Amazon CloudWatch Logs mit AWS OpsWorks Stacks

Service	Dokumentation:
Amazon Relational Database Service	Veröffentlichen von PostgreSQL-Protokollen in CloudWatch Logs
AWS RoboMaker	AWS RoboMaker CloudWatch ROS-Knoten mit Offline-Unterstützung
Amazon Route 53	Protokollierung und Überwachung in Amazon Route 53
Amazon SageMaker	Protokollieren Amazon SageMaker Ereignisse mit Amazon CloudWatch
Amazon Simple Notification Service	Anzeigen CloudWatch Logs
Amazon VPC	VPC Flow Logs

Sicherheit in Amazon CloudWatch Logs

Cloud-Sicherheit hat bei AWS höchste Priorität. Als AWS-Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die zur Erfüllung der Anforderungen von Organisationen entwickelt wurden, für die Sichherheit eine kritische Bedeutung hat.

Sicherheit ist eine übergreifende Verantwortlichkeit zwischen AWS und Ihnen. Das Modell der übergreifenden Verantwortlichkeit beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud selbst – AWS ist dafür verantwortlich, die Infrastruktur zu schützen, mit der AWS-Services in der AWS- Cloud ausgeführt werden. AWS stellt Ihnen außerdem Services bereit, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der AWS-Compliance-Programme regelmäßig. Weitere Informationen zu den Compliance-Programmen, die für Amazon WorkSpaces gelten, finden Sie unter AWS-Services in Scope nach Compliance-Programm.
- Sicherheit in der Cloud in – Ihr Verantwortungsumfang wird durch den AWS-Service bestimmt, den Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation zeigt Ihnen, wie das Modell der gemeinsamen Verantwortung bei der Verwendung von Amazon CloudWatch Logs angewendet wird. Sie veranschaulicht, wie Sie Amazon CloudWatch Logs konfigurieren, damit Ihre Sicherheits- und Compliance-Ziele erreicht werden. Sie erfahren außerdem, wie Sie andere AWS-Services einsetzen, um Ihre CloudWatch Logs-Ressourcen zu überwachen und zu schützen.

Inhalt

- Datenschutz in Amazon CloudWatch Logs (p. 126)
- Identity and Access Management für Amazon CloudWatch Logs (p. 127)
- Compliance-Validierung für Amazon CloudWatch Logs (p. 144)
- Ausfallsicherheit in Amazon CloudWatch Logs (p. 144)
- Sicherheit der Infrastruktur in Amazon CloudWatch Logs (p. 145)
- Verwenden von CloudWatch Logs mit Schnittstellen-VPC-Endpunkten (p. 145)

Datenschutz in Amazon CloudWatch Logs

Amazon CloudWatch Logs entspricht dem AWS Modell der geteilten Verantwortung, das Vorschriften und Richtlinien zum Datenschutz umfasst. AWS ist für den Schutz der globalen Infrastruktur verantwortlich, die alle AWS-Services ausführt. AWS behält die Kontrolle über die in dieser Infrastruktur gehosteten Daten, einschließlich der Sicherheitskonfigurationskontrollen für die Handhabung von Kundeninhalten und personenbezogenen Daten. AWS-Kunden und APN-Partner, die entweder als Datenverantwortliche oder Datenverarbeiter fungieren, sind für alle personenbezogenen Daten verantwortlich, die sie in die AWS Cloud einspeisen.

Zum Zweck des Datenschutzes empfehlen wir, die Anmeldeinformationen für das AWS-Konto zu schützen und individuelle Benutzerkonten mit AWS Identity and Access Management (IAM) einzurichten, damit jeder Benutzer nur die Berechtigungen besitzt, die er für seine beruflichen Aufgaben benötigt. Außerdem sollten Sie die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Factor Authentication (MFA).
- Verwenden Sie TLS für die Kommunikation mit AWS-Ressourcen.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit AWS CloudTrail ein.
- Verwenden Sie AWS-Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen in AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheits-Services wie Amazon Macie, um Unterstützung bei der Erkennung und beim Schutz von persönlichen Daten zu erhalten, die in Amazon S3 gespeichert sind.

Es wird nachdrücklich empfohlen, sensible personenbezogene Informationen wie Kontonummern von Kunden in Freitextfelder oder in Metadaten wie Funktionsnamen und Tags einzugeben. Alle Daten, die Sie in Metadaten eingeben, werden möglicherweise in Diagnoseprotokolle aufgenommen. Wenn Sie eine URL für einen externen Server bereitstellen, schließen Sie keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL ein.

Weitere Informationen zum Datenschutz enthält der Blog-Beitrag AWS Shared Responsibility Model and GDPR im AWS-Sicherheitsblog.

Verschlüsselung im Ruhezustand

CloudWatch Logs schützt Daten im Ruhezustand mithilfe von Verschlüsselung. Alle Protokollgruppen sind verschlüsselt. Der CloudWatch Logs-Service verwaltet standardmäßig die serverseitigen Verschlüsselungsschlüssel.

Verwenden Sie die Kundenmasterschlüssel (CMK) von AWS Key Management Service, um die Schlüssel zu verwalten, die zum Verschlüsseln und Entschlüsseln Ihrer Protokolle verwendet werden. Weitere Informationen finden Sie unter Protokolldaten in CloudWatch Logs mithilfe von AWS KMS verschlüsseln (p. 67).

Verschlüsselung während der Übertragung

CloudWatch Logs verwendet End-to-End-Verschlüsselung von Daten während der Übertragung. Der CloudWatch Logs-Service verwaltet die serverseitigen Verschlüsselungsschlüssel.

Identity and Access Management für Amazon CloudWatch Logs

Für den Zugriff auf Amazon CloudWatch Logs werden Anmeldeinformationen benötigt, die AWS zur Authentifizierung Ihrer Anforderungen verwenden kann. Diese Anmeldeinformationen müssen über Berechtigungen für den Zugriff auf AWS-Ressourcen wie z. B. für den Abruf von CloudWatch Logs-Daten zu Ihrer Cloud-Ressourcen verfügen. In den folgenden Abschnitten erfahren Sie, wie Sie Ihre Ressourcen mithilfe von AWS Identity and Access Management (IAM) und CloudWatch Logs sichern können, indem Sie den Zugriff darauf kontrollieren.

- Authentication (p. 127)
- Zugriffskontrolle (p. 129)

Authentication

Sie können mit einer der folgenden Identitäten auf AWS zugreifen:

- Root-Benutzer des AWS-Kontos – Wenn Sie sich bei AWS anmelden, geben Sie eine mit Ihrem AWS-Konto verknüpfte E-Mail-Adresse und das Passwort an. Dies sind Ihre Root-Anmeldeinformationen. Sie bieten vollständigen Zugriff auf alle Ihre AWS-Ressourcen.

  Important

  Aus Sicherheitsgründen empfehlen wir, die Root-Anmeldeinformationen nur zum Erstellen eines Administrator-Benutzers zu verwenden. Hierbei handelt es sich um einen IAM-Benutzer mit vollständigen Berechtigungen für Ihr AWS-Konto. Anschließend können Sie mit diesem Administratorbenutzer andere IAM-Benutzer und -Rollen mit eingeschränkten Berechtigungen erstellen. Weitere Informationen finden Sie unter Bewährte Methoden für IAM und Erstellen eines Admin-Benutzers und einer Gruppe im IAM-Benutzerhandbuch.

- IAM-Benutzer – Ein IAM-Benutzer ist eine Identität in Ihrem AWS-Konto mit bestimmten benutzerdefinierten Berechtigungen (z. B. Berechtigungen zum Anzeigen von Metriken in CloudWatch Logs). Sie können einen IAM-Benutzernamen und ein Passwort für die Anmeldung bei sicheren AWS-Webseiten verwenden. Dazu zählen beispielsweise die AWS Management Console, AWS-Diskussionsforen oder das AWS Support Center.

  Zusätzlich zu einem Benutzernamen und Passwort können Sie Zugriffsschlüssel für jeden Benutzer erstellen. Verwenden Sie diese Schlüssel, wenn Sie über eines der verschiedenen SDKs oder über die AWS Command Line Interface (AWS CLI) programmgesteuert auf AWS-Services zugreifen. Das SDK und die CLI-Tools verwenden die Zugriffsschlüssel, um Ihre Anfrage verschlüsselt zu signieren. Wenn Sie die AWS-Tools nicht verwenden, müssen Sie die Anforderung selbst signieren. CloudWatch Logs supports Unterschrift Version 4, ein Protokoll für die Authentifizierung eingehender API-Anforderungen. Weitere Informationen zur Authentifizierung von Anfragen finden Sie unter Signature Version 4-Signaturprozess im AWS General Reference.

- IAM-Rolle – Eine IAM-Rolle ist eine andere IAM-Identität, die Sie in Ihrem Konto mit bestimmten Berechtigungen erstellen können. Sie ähnelt einem IAM-Benutzer, ist aber nicht mit einer bestimmten Person verknüpft. Eine IAM-Rolle ermöglicht Ihnen, temporäre Zugriffsschlüssel zu erhalten, mit denen Sie auf die AWS-Services und -Ressourcen zugreifen können. IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

  - Verbundener Benutzerzugriff – Statt einen IAM-Benutzer zu erstellen, können Sie bereits vorhandene Benutzeridentitäten von AWS Directory Service, dem Benutzerverzeichnis Ihres Unternehmens, oder von einem Web-Identitätsanbieter verwenden. Diese werden als verbundene Benutzer bezeichnet. AWS weist einem verbundenen Benutzer eine Rolle zu, wenn der Zugriff über einen Identitätsanbieter angefordert wird. Weitere Informationen zu verbundenen Benutzern finden Sie unter Verbundene Benutzer und Rollen im IAM-Benutzerhandbuch.

  - Kontenübergreifender Zugriff – Sie können eine IAM-Rolle in Ihrem Konto verwenden, um einem anderen AWS-Konto Berechtigungen für den Zugriff auf die Ressourcen Ihres Kontos zu erteilen. Ein Beispiel finden Sie unter. Lernprogramm: Delegieren des Zugriffs über AWS-Konten mithilfe von IAM-Rollen in der IAM-Benutzerhandbuch.

  - Zugriff auf AWS-Services – Sie können eine IAM-Rolle in Ihrem Konto verwenden, um einem AWS-Service Berechtigungen für den Zugriff auf die Ressourcen Ihres Konto zu erteilen. Sie können beispielsweise eine Rolle erstellen, mit der Amazon Redshift in Ihrem Namen auf einen Amazon S3-Bucket zugreifen und die im Bucket gespeicherten Daten in einen Amazon Redshift-Cluster laden kann. Weitere Informationen finden Sie unter Erstellen von einer Rolle, um die Berechtigungen an einen AWS-Service zu delegieren im IAM-Benutzerhandbuch.

- Anwendungen in Amazon EC2 – Anstatt Zugriffsschlüssel in der EC2-Instance zu speichern, die von den dort ausgeführten Anwendungen zum Senden von AWS API-Anforderungen verwendet werden, können Sie eine IAM-Rolle nutzen, um temporäre Anmeldeinformationen für diese Anwendungen zu verwalten. Erstellen Sie ein Instance-Profil, das an die Instance angefügt ist, um eine AWS-Rolle zu einer EC2-Instance zuzuweisen und die Rolle für sämtliche Anwendungen der Instance bereitzustellen. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2 Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter Verwenden von Rollen für Anwendungen in Amazon EC2 im IAM-Benutzerhandbuch.

Zugriffskontrolle

Auch wenn Sie über gültige Anmeldeinformationen zur Authentifizierung Ihrer Anforderungen verfügen, können die CloudWatch Logs-Ressourcen nur mit entsprechenden Berechtigungen erstellen oder darauf zugreifen. Sie müssen beispielsweise über Berechtigungen verfügen, um Protokollstreams, Protokollgruppen und so weiter zu erstellen.

In den folgenden Abschnitten wird die Verwaltung von Berechtigungen für CloudWatch Logs beschrieben: Wir empfehlen Ihnen, zunächst die Übersicht zu lesen.

- Übersicht über die Verwaltung von Zugriffsberechtigungen für Ihre CloudWatch Logs-Ressourcen (p. 129)
- Verwenden identitätsbasierter Richtlinien (IAM-Richtlinien) für CloudWatch Logs (p. 133)
- Referenz für CloudWatch Logs-Berechtigungen (p. 138)

Übersicht über die Verwaltung von Zugriffsberechtigungen für Ihre CloudWatch Logs-Ressourcen

Jede AWS-Ressource ist Eigentum eines AWS-Kontos und die Berechtigungen für die Erstellung einer Ressource oder den Zugriff darauf werden durch Berechtigungsrichtlinien geregelt. Ein Kontoadministrator kann IAM-Identitäten (d. h. Benutzer, Gruppen und Rollen) Berechtigungsrichtlinien zuweisen. Manche Services (z. B. AWS Lambda) unterstützen auch die Zuweisung von Berechtigungsrichtlinien zu Ressourcen.

> Note
>
> Ein Kontoadministrator (oder IAM-Administratorbenutzer) ist ein Benutzer mit Administratorrechten. Weitere Informationen finden Sie unter Bewährte Methoden für IAM im IAM-Benutzerhandbuch.

Beim Erteilen von Berechtigungen entscheiden Sie, wer die Berechtigungen erhält, für welche Ressourcen die Berechtigungen gelten und welche Aktionen an diesen Ressourcen gestattet werden sollen.

Themen

- CloudWatch Logs-Ressourcen und -Vorgänge (p. 130)
- Grundlegendes zum Eigentum an Ressourcen (p. 130)
- Verwalten des Zugriffs auf Ressourcen (p. 130)
- Festlegen von Richtlinienelementen: Aktionen, Auswirkungen und Prinzipale (p. 133)

- Angeben von Bedingungen in einer Richtlinie (p. 133)

CloudWatch Logs-Ressourcen und -Vorgänge

In CloudWatch Logs sind primäre Ressourcen Protokollgruppen, Protokollstreams und Ziele. CloudWatch Logs unterstützt keine Subressourcen (andere Ressourcen für die Verwendung mit der primären Ressource).

Diesen Ressourcen und Unterressourcen sind eindeutige Amazon-Ressourcennamen (ARN) zugeordnet, wie in der folgenden Tabelle zu sehen ist.

Ressourcentyp	ARN-Format
Protokollgruppe	Arn:ws:Protokolle:*region*:*account-id*:Protokollgruppe:*log_group_name*
Protokollstream	Arn:ws:Protokolle:*region*:*account-id*:Protokollgruppe:*log_group_name*:log-stream (Protokoll-Stream):*log-stream-name*
Zieladresse	Arn:ws:Protokolle:*region*:*account-id*Ziel*destination_name*

Weitere Informationen zum Erstellen von ARNs finden Sie unter ARNs im IAM-Benutzerhandbuch. Weitere Informationen über CloudWatch Logs ARNs finden Sie unter Amazon-Ressourcennamen (ARNs) and AWS-Service-Namespaces im Allgemeine Amazon Web Services-Referenz. Ein Beispiel für eine Richtlinie, die CloudWatch Logs abdeckt, finden Sie unter Verwenden identitätsbasierter Richtlinien (IAM-Richtlinien) für CloudWatch Logs (p. 133).

CloudWatch Logs bietet eine Reihe von Operationen, um mit den CloudWatch Logs-Ressourcen zu arbeiten. Eine Liste der verfügbaren Operationen finden Sie unter Referenz für CloudWatch Logs-Berechtigungen (p. 138).

Grundlegendes zum Eigentum an Ressourcen

Das AWS-Konto ist Eigentümer aller Ressourcen, die innerhalb des Kontos erstellt werden, unabhängig davon, wer sie erstellt. Genauer gesagt ist das AWS-Konto der Prinzipal-Entität (d. h. das Root-Konto, ein IAM-Benutzer oder eine IAM-Rolle), welche die Ressourcenerstellungsanforderung authentifiziert, der Ressourceneigentümer. Die Funktionsweise wird anhand der folgenden Beispiele deutlich:

- Wenn Sie die Root-Konto-Anmeldeinformationen für Ihr AWS-Konto verwenden, um eine Protokollgruppe zu erstellen, ist Ihr AWS-Konto der Eigentümer der CloudWatch Logs-Ressource.
- Wenn Sie einen IAMBenutzer in Ihrem AWS-Konto erstellen und diesem Berechtigungen zum Erstellen von CloudWatch Logs-Ressourcen erteilen, kann dieser Benutzer CloudWatch Logs-Ressourcen erstellen. Eigentümer der CloudWatch Logs-Ressourcen ist jedoch das AWS-Konto, zu dem der Benutzer gehört.
- Wenn Sie in Ihrem AWS-Konto eine IAM-Rolle mit Berechtigungen zum Erstellen von CloudWatch Logs-Ressourcen einrichten, kann jeder Benutzer, der die Rolle übernimmt, CloudWatch Logs-Ressourcen erstellen. Ihr AWS-Konto, dem die Rolle angehört, ist der Eigentümer der CloudWatch Logs-Ressourcen.

Verwalten des Zugriffs auf Ressourcen

Eine Berechtigungsrichtlinie beschreibt, wer Zugriff auf welche Objekte hat. Im folgenden Abschnitt werden die verfügbaren Optionen zum Erstellen von Berechtigungsrichtlinien erläutert.

> **Note**
>
> Dieser Abschnitt behandelt die Verwendung von IAM im Zusammenhang mit CloudWatch Logs. Er enthält keine detaillierten Informationen über den IAM-Service. Eine umfassende IAM-Dokumentation finden Sie im Thema Was ist IAM? im IAM-Benutzerhandbuch. Informationen zur IAM-Richtliniensyntax und Beschreibungen finden Sie in der IAM-Richtlinienreferenz im IAM-Benutzerhandbuch.

Richtlinien, die einer IAM-Identität zugeordnet sind, werden als identitätsbasierte Richtlinien (IAM-Richtlinien) bezeichnet, während Richtlinien, die einer Ressource zugeordnet sind, als ressourcenbasierte Richtlinien bezeichnet werden. CloudWatch Logs unterstützt Richtlinien auf Identititätsbasis und Richtlinien auf Ressourcenbasis für Ziele, die verwendet werden, um kontoübergreifende Abonnements zu ermöglichen. Weitere Informationen finden Sie im Freigabe von Protokolldaten mit Abonnements für mehrere Konten (p. 103).

Themen

- Protokollgruppenberechtigungen und Beitragerkenntnisse (p. 131)
- Identitätsbasierte Richtlinien (IAM-Richtlinien) (p. 131)
- Ressourcenbasierte Richtlinien (p. 132)

Protokollgruppenberechtigungen und Beitragerkenntnisse

Contributor Insights ist eine Funktion von CloudWatch mit der Sie Daten aus Protokollgruppen analysieren und Zeitreihen erstellen können, die Contributor-Daten anzeigen. Sie können Metriken über die Top-N-Contributors, die Gesamtzahl der eindeutigen Contributors und deren Nutzung anzeigen. Weitere Informationen finden Sie unter Verwenden von Contributor Insights zum Analysieren von Daten mit hoher Kardinalität.

Wenn Sie einem Benutzer die `cloudwatch:PutInsightRule` und `cloudwatch:GetInsightRuleReport` -Berechtigungen kann dieser Benutzer eine Regel erstellen, die jede Protokollgruppe in CloudWatch Logs und sehen Sie dann die Ergebnisse. Die Ergebnisse können Contributor-Daten für diese Protokollgruppen enthalten. Stellen Sie sicher, dass Sie diese Berechtigungen nur Benutzern gewähren, die diese Daten anzeigen können sollten.

Identitätsbasierte Richtlinien (IAM-Richtlinien)

Richtlinien können IAM-Identitäten zugewiesen werden. Sie können z. B. Folgendes tun:

- Einem Benutzer oder einer Gruppe in Ihrem Konto eine Berechtigungsrichtlinie zuweisen – Wenn Sie einem Benutzer Berechtigungen zum Anzeigen von Protokollen in der CloudWatch Logs-Konsole erteilen möchten, können Sie dem Benutzer oder der Gruppe, zu der er gehört, eine Berechtigungsrichtlinie zuweisen.
- Anfügen einer Berechtigungsrichtlinie an eine Rolle (Erteilen kontoübergreifender Berechtigungen) – Sie können einer IAM-Rolle eine identitätsbasierte Berechtigungsrichtlinie anfügen, um kontoübergreifende Berechtigungen zu erteilen. Beispielsweise kann der Administrator in Konto A eine Rolle erstellen, um einem anderen AWS-Konto (z. B. Konto B) oder einem AWS-Service kontoübergreifende Berechtigungen zu erteilen. Dazu geht er folgendermaßen vor:
  1. Der Administrator von Konto A erstellt eine IAM-Rolle und weist ihr eine Berechtigungsrichtlinie zu, die Berechtigungen für Ressourcen in Konto A erteilt.
  2. Der Administrator von Konto A fügt der Rolle eine Vertrauensrichtlinie an, die Konto B als den Prinzipal identifiziert, der die Rolle übernehmen kann.
  3. Der Administrator von Konto B kann nun Berechtigungen zur Übernahme der Rolle an alle Benutzer in Konto B delegieren. Daraufhin können die Benutzer in Konto B auf Ressourcen von Konto A zugreifen oder auch Ressourcen erstellen. Der Prinzipal in der Vertrauensrichtlinie kann auch ein AWS-Service-

Prinzipal sein. Somit können Sie auch einem AWS-Service die Berechtigungen zur Übernahme der Rolle erteilen.

Weitere Informationen zum Delegieren von Berechtigungen mithilfe von IAM finden Sie unter Zugriffsverwaltung im IAM-Benutzerhandbuch.

Es folgt ein Beispiel für eine Richtlinie, die Berechtigungen für die Aktionen `logs:PutLogEvents`, `logs:CreateLogGroup` und `logs:CreateLogStream` für alle Ressourcen in us-east-1 erteilt. Bei einigen der API-Aktionen unterstützt CloudWatch Logs für Protokollgruppen mit der Verwendung von Ressourcen-ARNs (auch als „Berechtigungen auf Ressourcenebene" bezeichnet) die Identifizierung bestimmter Ressourcen. Wenn Sie alle Protokollgruppen aufnehmen möchten, müssen Sie den Platzhalter (*) angeben.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"",
         "Effect":"Allow",
         "Action":[
            "logs:PutLogEvents",
            "logs:CreateLogGroup",
            "logs:CreateLogStream"
         ],
         "Resource":"arn:aws:logs:us-east-1:*:*"
      }
   ]
}
```

Weitere Informationen zur Verwendung von identitätsbasierten Richtlinien mit CloudWatch Logs finden Sie unter Verwenden identitätsbasierter Richtlinien (IAM-Richtlinien) für CloudWatch Logs (p. 133). Weitere Informationen zu Benutzern, Gruppen, Rollen und Berechtigungen finden Sie im Thema Identitäten (Benutzer, Gruppen und Rollen im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

CloudWatch Logs unterstützt ressourcenbasierte Richtlinien für Ziele, die Sie verwenden können, um kontoübergreifenden Abonnements zu aktivieren. Weitere Informationen finden Sie im Erstellen eines Ziels (p. 104). Ziele können mithilfe der PutDestination-API erstellt werden, und Sie können dem Ziel mithilfe der PutDestination-API eine Ressourcen-Richtlinie hinzufügen. Beim folgenden Beispiel wird es einem anderen AWS-Konto mit der Konto-ID 111122223333 ermöglicht, dass seine Protokollgruppen das Ziel `arn:aws:logs:us-east-1:123456789012:destination:testDestination` abonnieren.

```
{
  "Version" : "2012-10-17",
  "Statement" : [
    {
      "Sid" : "",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "111122223333"
      },
      "Action" : "logs:PutSubscriptionFilter",
      "Resource" : "arn:aws:logs:us-east-1:123456789012:destination:testDestination"
    }
  ]
}
```

Festlegen von Richtlinienelementen: Aktionen, Auswirkungen und Prinzipale

Für jede CloudWatch Logs-Ressource definiert der Dienst eine Reihe von API-Operationen. Zur Erteilung von Berechtigungen für diese API-Operationen definiert CloudWatch Logs Aktionen, die Sie in einer Richtlinie angeben können. Einige API-Operationen erfordern möglicherweise Berechtigungen für mehr als eine Aktion, um die API-Operation auszuführen. Weitere Informationen zu Ressourcen und API-Operationen finden Sie unter CloudWatch Logs-Ressourcen und -Vorgänge (p. 130) und Referenz für CloudWatch Logs-Berechtigungen (p. 138).

Grundlegende Richtlinienelemente:

- Ressource – Sie verwenden einen Amazon-Ressourcennamen (ARN), um die Ressource, für welche die Richtlinie gilt, zu identifizieren. Weitere Informationen finden Sie im CloudWatch Logs-Ressourcen und -Vorgänge (p. 130).
- Aktion – Mit Aktionsschlüsselwörtern geben Sie die Ressourcenoperationen an, die Sie zulassen oder verweigern möchten. Die `logs.DescribeLogGroups`-Berechtigung erteilt dem Benutzer zum Beispiel Berechtigungen zum Ausführen der `DescribeLogGroups`-Operation.
- Effekt – Die von Ihnen festgelegte Auswirkung (entweder Zugriffserlaubnis oder Zugriffsverweigerung), wenn ein Benutzer die jeweilige Aktion anfordert. Wenn Sie den Zugriff auf eine Ressource nicht ausdrücklich gestatten („Allow“), wird er automatisch verweigert. Sie können den Zugriff auf eine Ressource auch explizit verweigern. So können Sie sicherstellen, dass Benutzer nicht darauf zugreifen können, auch wenn der Zugriff durch eine andere Richtlinie gestattet wird.
- Prinzipal – – In identitätsbasierten Richtlinien (IAM-Richtlinien) ist der Benutzer, dem die Richtlinie zugewiesen ist, automatisch der Prinzipal. In ressourcenbasierten Richtlinien müssen Sie den Benutzer, das Konto, den Service oder die sonstige Entität angeben, die die Berechtigungen erhalten soll (gilt nur für ressourcenbasierte Richtlinien). CloudWatch Logs bietet keine Unterstützung für ressourcenbasierte Richtlinien für Ziele.

Weitere Informationen zur IAM-Richtliniensyntax und entsprechende Beschreibungen enthält die AWS IAM-Richtlinienreferenz im IAM-Benutzerhandbuch.

Eine Tabelle mit allen CloudWatch Logs-API-Aktionen und den Ressourcen, für die diese gelten, finden Sie unter Referenz für CloudWatch Logs-Berechtigungen (p. 138).

Angeben von Bedingungen in einer Richtlinie

Beim Erteilen von Berechtigungen können Sie mithilfe der Sprache der Zugriffsrichtlinie die Bedingungen angeben, wann die Richtlinie wirksam werden soll. Beispielsweise kann festgelegt werden, dass eine Richtlinie erst ab einem bestimmten Datum gilt. Weitere Informationen zum Angeben von Bedingungen in einer Richtliniensyntax finden Sie unter dem Thema Bedingung im IAM-Benutzerhandbuch.

Bedingungen werden mithilfe vordefinierter Bedingungsschlüssel formuliert. Eine Liste aller Kontextschlüssel, die von den einzelnen AWS-Services unterstützt werden, und eine Liste von AWS-weiten Richtlinienschlüsseln finden Sie unter AWS-Serviceaktionen und Bedingungskontextschlüssel und Globale und IAM-Bedingungskontextschlüssel im IAM-Benutzerhandbuch.

Verwenden identitätsbasierter Richtlinien (IAM-Richtlinien) für CloudWatch Logs

In diesem Thema finden Sie Beispiele für identitätsbasierte Richtlinien, in denen ein Kontoadministrator den IAM-Identitäten (Benutzer, Gruppen und Rollen) Berechtigungsrichtlinien anfügen kann.

> **Important**
>
> Wir empfehlen Ihnen, zunächst die einführenden Themen zu lesen, in denen die Grundkonzepte und verfügbaren Optionen zum Verwalten des Zugriffs auf Ihre CloudWatch Logs-Ressourcen erläutert werden. Weitere Informationen finden Sie im Übersicht über die Verwaltung von Zugriffsberechtigungen für Ihre CloudWatch Logs-Ressourcen (p. 129).

In diesem Thema wird Folgendes behandelt:

- Erforderliche Berechtigungen für die Verwendung der CloudWatch-Konsole (p. 134)
- Von AWS verwaltete (vordefinierte) Richtlinien für CloudWatch Logs (p. 136)
- Beispiele für vom Kunden verwaltete Richtlinien (p. 136)

Nachstehend finden Sie ein Beispiel für eine Berechtigungsrichtlinie:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
      "Resource": [
        "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

In dieser Richtlinie gibt es eine Anweisung, die Berechtigungen erteilt, um Protokollgruppen und Protokollstreams zu erstellen, Protokollereignisse hochzuladen und Details zu Protokollstreams aufzuführen.

Das Platzhalterzeichen (*) am Ende des `Resource`-Werts bedeutet, dass die Anweisung Berechtigungen für die `logs:CreateLogGroup`-, `logs:CreateLogStream`-, `logs:PutLogEvents`- und `logs:DescribeLogStreams`-Aktionen einer Protokollgruppe zulässt. Um diese Berechtigungen auf eine bestimmte Protokollgruppe zu beschränken, ersetzen Sie das Platzhalterzeichen (*) im ARN der Ressource durch den spezifischen ARN der Protokollgruppe. Weitere Informationen über die Abschnitte in einer IAM-Richtlinienanweisung finden Sie unterReferenz für IAM-Richtlinienelemente im IAM-Benutzerhandbuch. Eine Liste mit allen CloudWatch Logs-Aktionen finden Sie unter Referenz für CloudWatch Logs-Berechtigungen (p. 138).

Erforderliche Berechtigungen für die Verwendung der CloudWatch-Konsole

Damit Benutzer mit CloudWatch Logs in der CloudWatch-Konsole arbeiten können, müssen diese über eine Mindestmenge an Berechtigungen verfügen, die den Benutzern erlauben, andere AWS-Ressourcen für ihr AWS-Konto zu beschreiben. Um CloudWatch Logs in der CloudWatch-Konsole zu verwenden, benötigen Sie Berechtigungen der folgenden Services:

- CloudWatch
- CloudWatch Logs
- Amazon ES

- IAM
- Kinesis
- Lambda
- Amazon S3

Wenn Sie eine IAM-Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Benutzer mit dieser IAM-Richtlinie. Um sicherzustellen, dass diese Benutzer die CloudWatch-Konsole weiterhin verwenden können, fügen Sie dem Benutzer auch die verwaltete Richtlinie `CloudWatchReadOnlyAccess` an. Einzelheiten dazu finden Sie unter Von AWS verwaltete (vordefinierte) Richtlinien für CloudWatch Logs (p. 136).

Für Benutzer, die nur Aufrufe an die AWS CLI oder CloudWatch Logs-API durchführen, müssen Sie keine Mindestberechtigungen in der Konsole erteilen.

Die vollständige Satz von Berechtigungen, die ein Benutzer, der mit der CloudWatch-Konsole arbeitet, sie jedoch nicht zum Verwalten von Protokoll-Abonnements verwendet, benötigt:

- cloudwatch:getMetricData
- cloudwatch:listMetrics
- logs:cancelExportTask
- logs:createExportTask
- logs:createLogGroup
- logs:createLogStream
- logs:deleteLogGroup
- logs:deleteLogStream
- logs:deleteMetricFilter
- logs:deleteQueryDefinition
- logs:deleteRetentionPolicy
- logs:deleteSubscriptionFilter
- logs:describeExportTasks
- logs:describeLogGroups
- logs:describeLogStreams
- logs:describeMetricFilters
- logs:describeQueryDefinitions
- logs:describeSubscriptionFilters
- logs:filterLogEvents
- logs:getLogEvents
- logs:putMetricFilter
- logs:putQueryDefinition
- logs:putRetentionPolicy
- logs:putSubscriptionFilter
- logs:testMetricFilter

Ein Benutzer, der die Konsole auch zum Verwalten von Protokoll-Abonnements verwendet, benötigt die folgenden Berechtigungen:

- es:describeElasticsearchDomain
- es:listDomainNames

- iam:attachRolePolicy
- iam:createRole
- iam:getPolicy
- iam:getPolicyVersion
- iam:getRole
- iam:listAttachedRolePolicies
- iam:listRoles
- kinesis:describeStreams
- kinesis:listStreams
- lambda:addPermission
- lambda:createFunction
- lambda:getFunctionConfiguration
- lambda:listAliases
- lambda:listFunctions
- lambda:listVersionsByFunction
- lambda:removePermission
- s3:listBuckets

Von AWS verwaltete (vordefinierte) Richtlinien für CloudWatch Logs

Durch die Bereitstellung von eigenständigen IAM-Richtlinien, die von AWS erstellt und administriert werden, deckt AWS viele häufige Anwendungsfälle ab. Die verwalteten Richtlinien erteilen die erforderlichen Berechtigungen für viele häufige Anwendungsfälle, sodass Sie nicht mühsam ermitteln müssen, welche Berechtigungen erforderlich sind. Weitere Informationen finden Sie unter AWS-verwaltete Richtlinien im IAM-Benutzerhandbuch.

Die folgenden AWS-verwalteten Richtlinien, die Sie an Benutzer in Ihrem Konto anfügen können, sind CloudWatch Logs-spezifisch:

- CloudWatchLogsFullAccess – gewährt vollständigen Zugriff auf CloudWatch Logs.
- CloudWatchLogsReadOnlyAccess – gewährt nur Lesezugriff auf CloudWatch Logs.

> **Note**
>
> Sie können diese Berechtigungsrichtlinien prüfen, indem Sie sich bei der IAM-Konsole anmelden und dort nach bestimmten Richtlinien suchen.

(Sie können auch Ihre eigenen benutzerdefinierten IAM-Richtlinien erstellen, um Berechtigungen für CloudWatch Logs-Aktionen und -Ressourcen zu gewähren). Die benutzerdefinierten Richtlinien können Sie dann den IAM-Benutzern oder -Gruppen zuweisen, die diese Berechtigungen benötigen.

Beispiele für vom Kunden verwaltete Richtlinien

In diesem Abschnitt finden Sie Beispiele für Benutzerrichtlinien, die Berechtigungen für verschiedene CloudWatch Logs-Aktionen gewähren. Diese Richtlinien sind nur wirksam, wenn Sie die CloudWatch Logs-API, AWS-SDKs oder die AWS CLI verwenden.

Beispiele

- Beispiel 1 Vollen Zugriff erlauben auf CloudWatch Logs (p. 137)

- Beispiel 2 Schreibgeschützten Zugriff erlauben auf CloudWatch Logs (p. 137)
- Beispiel 3 Zugriff auf eine Protokollgruppe zulassen (p. 137)

Beispiel 1 Vollen Zugriff erlauben auf CloudWatch Logs

Die folgende Richtlinie erlaubt einem Benutzer den Zugriff auf alle CloudWatch Logs-Aktionen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Beispiel 2 Schreibgeschützten Zugriff erlauben auf CloudWatch Logs

AWS stellt die Richtlinie CloudWatchLogsReadOnlyAccess bereit, die den Lesezugriff auf CloudWatch Logs-Daten ermöglicht. Diese Richtlinie umfasst die folgenden Berechtigungen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:Describe*",
                "logs:Get*",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Beispiel 3 Zugriff auf eine Protokollgruppe zulassen

Mit der folgenden Richtlinie kann ein Benutzer Protokollereignisse in einer bestimmten Protokollgruppe lesen und schreiben.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
      "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents",
        "logs:GetLogEvents"
      ],
      "Effect": "Allow",
```

```
      "Resource": "arn:aws:logs:us-west-2:123456789012:log-group:SampleLogGroupName:*"
      }
   ]
}
```

Verwenden von Tagging und IAM-Richtlinien für die Steuerung auf der Protokollgruppenebene

Sie können Benutzern Zugriff auf bestimmte Protokollgruppen gewähren und sie gleichzeitig daran hindern, auf andere Protokollgruppen zuzugreifen. Hierzu markieren Sie Ihre Protokollgruppen und verwenden IAM-Richtlinien, die auf diese Tags verweisen.

Weitere Informationen zum Taggen von Protokollgruppen finden Sie unter Protokollgruppen in Amazon CloudWatch Logs kennzeichnen (p. 64).

Wenn Sie Protokollgruppen markieren, können Sie einem Benutzer eine IAM-Richtlinie erteilen, um nur Zugriff auf die Protokollgruppen mit einem bestimmten Tag zu gewähren. Beispiel: Die folgende Richtlinienanweisung gewährt nur den Zugriff auf Regeln mit dem Wert `Green` für den Tag-Schlüssel `Team`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:*"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "logs:ResourceTag/Team": "Green"
                }
            }
        }
    ]
}
```

Weitere Informationen zum Verwenden von IAM-Richtlinienanweisungen finden Sie unter Zugriffssteuerung mit Richtlinien im IAM-Benutzerhandbuch.

Referenz für CloudWatch Logs-Berechtigungen

Wenn Sie die Zugriffskontrolle (p. 129) einrichten und Berechtigungsrichtlinien für eine IAM-Identität (identitätsbasierte Richtlinie) verfassen, können Sie die folgende Tabelle als Referenz verwenden. In der Tabelle werden alle CloudWatch Logs-API-Operationen und die zugehörigen Aktionen aufgeführt, für deren Ausführung Sie Berechtigungen erteilen können. Sie geben die Aktionen im Feld `Action` der Richtlinie an. Für das `Resource`-Feld können Sie den ARN einer Protokollgruppe oder eines Protokollstreams angeben oder `*` angeben, um alle CloudWatch Logs-Ressourcen darzustellen.

Zur Formulierung von Bedingungen in Ihren CloudWatch Logs-Richtlinien können Sie die globalen AWS-Bedingungsschlüssel verwenden. Eine vollständige Liste der AWS-weiten Schlüssel finden Sie unter AWSGlobale und IAM-Bedingungskontextschlüssel im IAM-Benutzerhandbuch.

> Note
>
> Um eine Aktion anzugeben, verwenden Sie das Präfix `logs:` gefolgt vom Namen der API-Operation. Beispiel, . `logs:CreateLogGroup`, , und Sie haben die Möglichkeit `logs:CreateLogStream`, oder `logs:*` (für alle CloudWatch Logs Aktionen).

CloudWatch Logs API-Operationen und erforderliche Berechtigungen für Aktionen

CloudWatch Logs-API-Operationen	Erforderliche Berechtigungen (API-Aktionen):
CancelExportTask	`logs:CancelExportTask` Erforderlich zum Abbrechen einer ausstehenden oder laufenden Exportaufgabe.
CreateExportTask	`logs:CreateExportTask` Erforderlich zum Exportieren von Daten aus einer Protokollgruppe in einen Amazon S3-Bucket.
CreateLogGroup	`logs:CreateLogGroup` Erforderlich zum Erstellen einer neuen Protokollgruppe.
CreateLogStream	`logs:CreateLogStream` Erforderlich zum Erstellen eines neuen Protokoll-Stream in eine Protokollgruppe.
DeleteDestination	`logs:DeleteDestination` Erforderlich zum Löschen eines Protokollziels und Aktivieren von Abonnementfiltern für das Ziel.
DeleteLogGroup	`logs:DeleteLogGroup` Erforderlich zum Löschen einer Protokollgruppe und der jeweiligen archivierten Protokollereignisse.
DeleteLogStream	`logs:DeleteLogStream` Erforderlich zum Löschen eines Protokoll-Stream und der jeweiligen archivierten Protokollereignisse.
DeleteMetricFilter	`logs:DeleteMetricFilter` Erforderlich zum Löschen eines Metrikfilters für eine Protokollgruppe.
DeleteQueryDefinition	`logs:DeleteQueryDefinition` Erforderlich, um eine gespeicherte Abfragedefinition in CloudWatch Logs Insights zu löschen.
DeleteResourcePolicy	`logs:DeleteResourcePolicy` Erforderlich, um eine CloudWatch Logs-Ressourcenrichtlinie zu löschen.
DeleteRetentionPolicy	`logs:DeleteRetentionPolicy` Erforderlich zum Löschen der Aufbewahrungsrichtlinie einer Protokollgruppe.
DeleteSubscriptionFilter	`logs:DeleteSubscriptionFilter`

CloudWatch Logs-API-Operationen	Erforderliche Berechtigungen (API-Aktionen):
	Erforderlich zum Löschen des Abonnementfilters für eine Protokollgruppe.
DescribeDestinations	`logs:DescribeDestinations` Erforderlich zum Anzeigen aller Ziele für ein Konto.
DescribeExportTasks	`logs:DescribeExportTasks` Erforderlich zum Anzeigen aller Exportaufgaben für das Konto.
DescribeLogGroups	`logs:DescribeLogGroups` Erforderlich zum Anzeigen aller Protokollgruppen für ein Konto.
DescribeLogStreams	`logs:DescribeLogStreams` Erforderlich zum Anzeigen aller Protokoll-Streams für eine Protokollgruppe.
DescribeMetricFilters	`logs:DescribeMetricFilters` Erforderlich zum Anzeigen aller Metriken für eine Protokollgruppe.
DescribeQueryDefinitions	`logs:DescribeQueryDefinitions` Erforderlich, um die Liste der gespeicherten Abfragedefinitionen in CloudWatch Logs Insights anzuzeigen.
DescribeQueries	`logs:DescribeQueries` Erforderlich, um die Liste der CloudWatch Logs Insights-Abfragen anzuzeigen, die geplant sind, ausgeführt werden oder kürzlich ausgeführt wurden.
DescribeResourcePolicies	`logs:DescribeResourcePolicies` Erforderlich, um eine Liste der CloudWatch Logs-Ressourcenrichtlinien anzuzeigen.
DescribeSubscriptionFilters	`logs:DescribeSubscriptionFilters` Erforderlich zum Anzeigen aller Abonnementfilter für eine Protokollgruppe.
FilterLogEvents	`logs:FilterLogEvents` Erforderlich zum Sortieren von Protokollereignissen nach Gruppenfiltermuster.
GetLogEvents	`logs:GetLogEvents` Erforderlich zum Abrufen von Protokollereignissen aus einem Protokoll-Stream.

CloudWatch Logs-API-Operationen	Erforderliche Berechtigungen (API-Aktionen):
GetLogGroupFields	`logs:GetLogGroupFields` Erforderlich, um die Liste der Felder abzurufen, die in den Protokollereignissen in einer Protokollgruppe enthalten sind.
GetLogRecord	`logs:GetLogRecord` Erforderlich, um die Details aus einem einzelnen Protokollereignis abzurufen.
GetQueryResults	`logs:GetQueryResults` Erforderlich, um die Ergebnisse von CloudWatch Logs Insights-Abfragen abzurufen.
ListTagsLogGroup	`logs:ListTagsLogGroup` Erforderlich zum Auflisten der mit einer Protokollgruppe verbundenen Tags.
PutDestination	`logs:PutDestination` Erforderlich zum Erstellen oder Aktualisieren eines Ziel-Protokoll-Streams (z. B. ein Kinesis-Stream).
PutDestinationPolicy	`logs:PutDestinationPolicy` Erforderlich zum Erstellen oder Aktualisieren einer Zugriffsrichtlinie im Zusammenhang mit einem vorhandenen Protokollziel.
PutLogEvents	`logs:PutLogEvents` Erforderlich zum Hochladen eines Stapels von Protokollereignissen in einen Protokoll-Stream.
PutMetricFilter	`logs:PutMetricFilter` Erforderlich zum Erstellen oder Aktualisieren eines Metrikfilters, der einer Protokollgruppe zugeordnet wird.
PutQueryDefinition	`logs:PutQueryDefinition` Erforderlich, um eine Abfrage in CloudWatch Logs Insights zu speichern.
PutResourcePolicy	`logs:PutResourcePolicy` Erforderlich, um eine CloudWatch Logs-Ressourcenrichtlinie zu erstellen.
PutRetentionPolicy	`logs:PutRetentionPolicy` Erforderlich zum Festlegen der Anzahl der Tage, die Protokollereignisse (Aufbewahrung) in einer Protokollgruppe aufbewahrt werden sollen.

CloudWatch Logs-API-Operationen	Erforderliche Berechtigungen (API-Aktionen):
PutSubscriptionFilter	`logs:PutSubscriptionFilter` Erforderlich zum Erstellen oder Aktualisieren eines Abonnementfilters, der einer Protokollgruppe zugeordnet wird.
StartQuery	`logs:StartQuery` Erforderlich, um CloudWatch Logs Insights-Abfragen zu starten.
StopQuery	`logs:StopQuery` Erforderlich, um eine CloudWatch Logs Insights-Abfrage zu beenden, die gerade ausgeführt wird.
TagLogGroup	`logs:TagLogGroup` Erforderlich zum Hinzufügen oder Aktualisieren von Protokollgruppen-Tags.
TestMetricFilter	`logs:TestMetricFilter` Erforderlich zum Testen eines Filtermusters anhand einer Stichprobe von Protokollereignis-Nachrichten.

Verwenden von serviceverknüpften Rollen für CloudWatch Logs

Amazon CloudWatch Logs verwendet serviceverknüpfte Rollen von AWS Identity and Access Management (IAM). Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit CloudWatch Logs verknüpft ist. Serviceverknüpfte Rollen werden von CloudWatch Logs vordefiniert und schließen alle Berechtigungen ein, die der Service zum Aufrufen anderer AWS-Services in Ihrem Namen erfordert.

Eine serviceverknüpfte Rolle macht das Einrichten von CloudWatch Logs effizienter, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. CloudWatch Logs definiert die Berechtigungen seiner serviceverknüpften Rollen und, sofern nicht anders definiert, nur CloudWatch Logs kann diese Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann nicht an andere IAM Entität.

Weitere Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter AWS-Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Yes (Ja) in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Berechtigungen von serviceverknüpften Rollen für CloudWatch Logs

CloudWatch Logs verwendet die serviceverknüpfte Rolle mit dem Namen AWSServiceRoleForLogBereitstellung. (z. B.. CloudWatch Logs verwendet diese serviceverknüpfte Rolle, um Protokolle direkt in zu schreiben Kinesis Data Firehose. Weitere Informationen finden Sie im Senden von Protokollen direkt an Amazon S3 oder Kinesis Data Firehose (p. 111).

Die Schaltfläche AWSServiceRoleForLogBereitstellung Die serviceverknüpfte Rolle vertraut den folgenden Services, um die Rolle zu übernehmen:

- `CloudWatch Logs`

Die Rollenberechtigungsrichtlinie erlaubt CloudWatch Logs die Durchführung der folgenden Aktionen für die angegebenen Ressourcen:

- Aktion: , `firehose:PutRecord` und `firehose:PutRecordBatch` auf alle Kinesis Data Firehose -Streams, die ein Tag mit einem `LogDeliveryEnabled` Schlüssel mit dem Wert `True`. (z. B.. Dieses Tag wird automatisch an eine Kinesis Data Firehose -Stream, wenn Sie ein Abonnement erstellen, um die Protokolle an zu übermitteln Kinesis Data Firehose.

Sie müssen Berechtigungen konfigurieren, um eine IAM -Entität, um eine serviceverknüpfte Rolle zu erstellen, zu bearbeiten oder zu löschen. Diese Entität kann ein Benutzer, eine Gruppe oder Rolle sein. Weitere Informationen finden Sie unter Berechtigungen von serviceverknüpften Rollen im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für CloudWatch Logs

Sie müssen keine serviceverknüpfte Rolle manuell erstellen. Wenn Sie Protokolle einrichten, die direkt an eine Kinesis Data Firehose Stream in der AWS Management Console, die AWS CLI, oder die AWS API zu verwenden, CloudWatch Logs erstellt die serviceverknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie Protokolle erneut so einrichten, dass sie direkt an eine Kinesis Data Firehose Datenstrom, CloudWatch Logs erstellt die serviceverknüpfte Rolle erneut für Sie.

Bearbeiten einer serviceverknüpften Rolle für CloudWatch Logs

CloudWatch Logs erlaubt Ihnen nicht, AWSServiceRoleForLogBereitstellungoder eine andere serviceverknüpfte Rolle, nachdem Sie erstellt haben. Sie können den Namen der Rolle nicht ändern, da verschiedene Entitäten möglicherweise auf die Rolle verweisen. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für CloudWatch Logs

Wenn Sie eine Funktion oder einen Service, die bzw. der eine serviceverknüpfte Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte Entität, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

> **Note**
>
> Wenn der CloudWatch Logs-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn das passiert, warten Sie einige Minuten und versuchen Sie es erneut.

So löschen Sie CloudWatch Logs -Ressourcen, die vom AWSServiceRoleForLogDelivery serviceverknüpfte Rolle

- Senden von Protokollen direkt an beenden Kinesis Data Firehose strömt.

So löschen Sie die serviceverknüpfte Rolle manuell mit IAM

Verwenden Sie die IAM Konsole, die AWS CLI, oder die AWS API zum Löschen der AWSServiceRoleForLogBereitstellung serviceverknüpfte Rolle. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle

Unterstützte Regionen für serviceverknüpfte CloudWatch Logs-Rollen

CloudWatch Logs unterstützt die Verwendung von servicegebundenen Rollen in allen AWS-Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter Regionen und Endpunkte von CloudWatch Logs.

Compliance-Validierung für Amazon CloudWatch Logs

Externe Auditoren bewerten im Rahmen verschiedener AWS-Compliance-Programme die Sicherheit und Compliance von Amazon CloudWatch Logs. Hierzu zählen unter anderem SOC, PCI, FedRAMP und HIPAA.

Eine Liste der AWS-Services, die in bestimmten Compliance-Programmen enthalten sind, finden Sie unter AWS-Services in Scope nach Compliance-Programm. Allgemeine Informationen finden Sie unter AWS-Compliance-Programme.

Die Auditberichte von Drittanbietern lassen sich mit AWS Artifact herunterladen. Weitere Informationen finden Sie unter Herunterladen von Berichten in AWS Artifact.

Ihre Verantwortung in Bezug auf die Compliance bei der Verwendung von Amazon CloudWatch Logs wird durch die Vertraulichkeit Ihrer Daten, die Compliance-Ziele Ihres Unternehmens und die geltenden Gesetze und Vorschriften bestimmt. AWS stellt die folgenden Ressourcen zur Unterstützung bei Compliance-Fragen bereit:

- Kurzanleitungen für Sicherheit und Compliance – In diesen Bereitstellungsleitfäden finden Sie wichtige Überlegungen zur Architektur sowie die einzelnen Schritte zur Bereitstellung von sicherheits- und Compliance-orientierten Basisumgebungen in AWS.
- Whitepaper zur Erstellung einer Architektur mit HIPAA-konformer Sicherheit und Compliance – Dieses Whitepaper beschreibt, wie Unternehmen mithilfe von AWS HIPAA-konforme Anwendungen erstellen.
- AWS-Compliance-Ressourcen – Diese Arbeitsbücher und Leitfäden könnten für Ihre Branche und Ihren Standort relevant sein.
- Auswertung von Ressourcen mit Regeln im AWS Config Developer Guide – Der AWS Config- Service bewertet, wie gut Ihre Ressourcenkonfigurationen mit internen Praktiken, Branchenrichtlinien und Vorschriften übereinstimmen.
- AWS Security Hub: Dieser AWS-Service liefert einen umfassenden Überblick über den Sicherheitsstatus in AWS. So können Sie die Compliance mit den Sicherheitsstandards in der Branche und den bewährten Methoden abgleichen.

Ausfallsicherheit in Amazon CloudWatch Logs

Im Zentrum der globalen AWS-Infrastruktur stehen die AWS-Regionen und -Availability Zones (Verfügbarkeitszonen, AZs). Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die über hoch redundante Netzwerke mit niedriger Latenz und hohen Durchsätzen verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen über zu AWS-Regionen und Availability Zones finden Sie unter Weltweite AWS-Infrastruktur.

Sicherheit der Infrastruktur in Amazon CloudWatch Logs

Als verwalteter Service ist Amazon CloudWatch Logs durch die globalen Verfahren zur Gewährleistung der Netzwerksicherheit von AWS geschützt, die im Whitepaper Amazon Web Services: Übersicht über die Sicherheitsprozesse beschrieben sind.

Sie verwenden von AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Amazon CloudWatch Logs zuzugreifen. Clients müssen Transport Layer Security (TLS) 1.0 oder höher unterstützen. Wir empfehlen TLS 1.2 oder höher. Clients müssen außerdem Cipher Suites mit PFS (Perfect Forward Secrecy) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) unterstützen. Die meisten modernen Systemen wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der mit einem IAM-Prinzipal verknüpft ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Verwenden von CloudWatch Logs mit Schnittstellen-VPC-Endpunkten

Wenn Sie Amazon Virtual Private Cloud (Amazon VPC) zum Hosten von AWS-Ressourcen verwenden, können Sie eine private Verbindung zwischen der VPC und CloudWatch Logs einrichten. Sie können diese Verbindung verwenden, um Protokolle an CloudWatch Logs zu senden, ohne sie über das Internet senden.

Amazon VPC ist ein AWS-Service, den Sie verwenden können, um AWS-Ressourcen in einem von Ihnen definierten virtuellen Netzwerk auszuführen. Mit einer VPC haben Sie die Kontrolle über Ihre Netzwerkeinstellungen, wie IP-Adressbereich, Subnetze, Routing-Tabellen und Netzwerk-Gateways. Zum Herstellen einer Verbindung der VPC mit CloudWatch Logs definieren Sie einen Schnittstellen-VPC-Endpunkt für CloudWatch Logs. Mit dieser Art Endpunkt können Sie Ihre VPC mit AWS-Services verbinden. Der Endpunkt bietet zuverlässige, skalierbare Konnektivität zu CloudWatch Logs, ohne dass ein Internet-Gateway, eine NAT-Instance (Network Address Translation) oder eine VPN-Verbindung erforderlich ist. Weitere Informationen finden Sie unter Was ist Amazon VPC im Amazon VPC Benutzerhandbuch.

VPC-Schnittstellenendpunkten werden über AWS PrivateLink bereitgestellt, eine AWS-Technologie, die eine private Kommunikation zwischen AWS-Services unter Verwendung einer Elastic Network-Schnittstelle mit privaten IP-Adressen ermöglicht. Weitere Informationen finden Sie im Blogeintrag zum Thema AWS PrivateLink für AWS-Services.

Die folgenden Schritte sind für Benutzer von Amazon VPC vorgesehen. Weitere Informationen finden Sie unter Erste Schritte im Amazon VPC Benutzerhandbuch.

Verfügbarkeit

Zurzeit unterstützt CloudWatch Logs VPC-Endpunkte in den folgenden Regionen:

- USA Ost (Ohio)

- USA Ost (Nord-Virginia)
- USA West (Nordkalifornien)
- USA West (Oregon)
- Asien-Pazifik (Hongkong)
- Asien-Pazifik (Mumbai)
- Asien-Pazifik (Seoul)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Tokio)
- Kanada (Zentral)
- Europa (Frankfurt)
- Europa (Irland)
- Europa (London)
- Europa (Paris)
- Südamerika (São Paulo)
- AWS GovCloud (USA Ost)
- AWS GovCloud (US-West)

Erstellen eines VPC-Endpunkts für CloudWatch Logs

Um CloudWatch Logs mit der VPC zu nutzen, erstellen Sie zunächst einen Schnittstellen-VPC-Endpunkt für CloudWatch Logs. Der auszuwählende Service ist com.amazonaws.***Region***.logs. Weitere Informationen finden Sie unter Erstellen eines Schnittstellenendpunkts im Amazon VPC Benutzerhandbuch.

Sie müssen die Einstellungen für CloudWatch Logs nicht ändern. CloudWatch Logs ruft andere AWS-Dienste unter Verwendung öffentlicher Endpunkte oder privater Schnittstellen-VPC-Endpunkte auf, je nachdem, was verwendet wird. Wenn Sie beispielsweise einen Schnittstellen-VPC-Endpunkt für CloudWatch Logs erstellen und bereits einen CloudWatch Logs-Abonnementfilter für Kinesis Data Streams und einen Schnittstellen-VPC-Endpunkt für Kinesis Data Streams haben, werden Aufrufe zwischen CloudWatch Logs und Kinesis Data Streams durch den Schnittstellen-VPC-Endpunkt geleitet.

Testen der Verbindung zwischen Ihrer VPC und CloudWatch Logs

Nachdem Sie den Endpunkt erstellt haben, können Sie die Verbindung testen.

Testen der Verbindung zwischen Ihrer VPC und Ihrem CloudWatch Logs-Endpunkt

1. Stellen Sie eine Verbindung mit einer Amazon EC2-Instance in Ihrer VPC her. Weitere Informationen zum Herstellen einer Verbindung finden Sie unter Herstellen einer Verbindung mit Ihrer Linux-Instance or Herstellen einer Verbindung mit Ihrer Windows-Instance in der Amazon EC2-Dokumentation.
2. Verwenden Sie von der Instance aus die AWS CLI, um einen Protokolleintrag in einer Ihrer vorhandenen Protokollgruppen anzulegen.

   Zuerst erstellen Sie eine JSON-Datei mit einem Protokollereignis. Der Zeitstempel muss als Millisekunden seit dem 1. Januar 1970 00:00:00 UTC angegeben werden.

```
[
  {
    "timestamp": 1533854071310,
    "message": "VPC Connection Test"
```

```
  }
]
```

Verwenden Sie anschließend den `put-log-events`-Befehl zum Erstellen des Protokolleintrags:

```
aws logs put-log-events --log-group-name LogGroupName --log-stream-name LogStreamName
 --log-events file://JSONFileName
```

Wenn die Antwort auf den Befehl `nextSequenceToken` enthält, war der Befehl erfolgreich und Ihr VPC-Endpunkt funktioniert.

Steuern des Zugriffs auf Ihren CloudWatch Logs-VPC-Endpunkt

Eine VPC-Endpunktrichtlinie ist eine IAM-Ressourcenrichtlinie, die Sie einem Endpunkt beim Erstellen oder Ändern des Endpunkts zuordnen. Wenn Sie einem Endpunkt beim Erstellen keine Richtlinie zuordnen, wird ihm eine Standardrichtlinie mit Vollzugriff auf den Service zugeordnet. IAM-Benutzerrichtlinien oder servicespezifische Richtlinien werden durch Endpunktrichtlinien nicht überschrieben oder ersetzt. Endpunktrichtlinien steuern unabhängig vom Endpunkt den Zugriff auf den angegebenen Service.

Endpunktrichtlinien müssen im JSON-Format erstellt werden.

Weitere Informationen finden Sie unter Zugriffskontrolle auf Services mit VPC-Endpunkten im Amazon VPC Benutzerhandbuch.

Im Folgenden finden Sie ein Beispiel für eine Endpunktrichtlinie für CloudWatch Logs. Diese Richtlinie ermöglicht es Benutzern, eine Verbindung zu CloudWatch Logs über die VPC zum Erstellen von Protokollstreams und Senden von Protokollen an CloudWatch Logs herzustellen und hindert sie daran, andere CloudWatch Logs-Aktionen auszuführen.

```
{
  "Statement": [
    {
      "Sid": "PutOnly",
      "Principal": "*",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

So ändern Sie die VPC-Endpunktrichtlinie für CloudWatch Logs

1. Öffnen Sie die Amazon VPC-Konsole unter der Adresse https://console.aws.amazon.com/vpc/.
2. Wählen Sie im Navigationsbereich Endpoints (Endpunkte) aus.
3. Sofern Sie nicht bereits den Endpunkt für CloudWatch Logs erstellt haben, wählen Sie Create Endpunkt (Endpoint erstellen) aus. Wählen Sie anschließend com.amazonaws.***Region***.logs und danach Create endpoint (Endpunkt erstellen) aus.
4. Wählen Sie den Endpunkt com.amazonaws.***Region***.logs und danach die Registerkarte Policy (Richtlinie) in der unteren Hälfte des Bildschirms aus.
5. Wählen Sie Edit Policy (Richtlinie bearbeiten) und nehmen Sie die Änderungen an der Richtlinie vor.

Support für VPC-Kontextschlüssel

CloudWatch Logs unterstützt die Kontextschlüssel `aws:SourceVpc` und `aws:SourceVpce`, mit denen der Zugriff auf bestimmte VPCs oder bestimmte VPC-Endpunkte beschränkt werden kann. Diese Schlüssel funktionieren nur, wenn der Benutzer VPC-Endpunkte verwendet. Weitere Informationen finden Sie unter Schlüssel, die für manche Services verfügbar sind im IAM-Benutzerhandbuch.

Protokollieren von Amazon CloudWatch Logs-API-Aufrufen in AWS CloudTrail

Amazon CloudWatch Logs ist in AWS CloudTrail integriert, ein Service, der die Aktionen eines Benutzers, einer Rolle oder eines AWS-Services in CloudWatch Logs aufzeichnet. CloudTrail erfasst alle API-Aufrufe von Ihrem oder für Ihr AWS-Konto. Zu den erfassten Aufrufen gehören Aufrufe von der CloudWatch-Konsole und Code-Aufrufe der CloudWatch Logs-API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail-Ereignissen an einen Amazon S3-Bucket aktivieren, einschließlich Ereignissen für CloudWatch Logs. Auch wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse in der CloudTrail-Konsole in Event history (Ereignisverlauf) anzeigen. Mit den von CloudTrail gesammelten Informationen können Sie die an CloudWatch Logs gestellte Anfrage, die IP-Adresse, von der die Anfrage gestellt wurde, den Initiator der Anfrage, den Zeitpunkt der Anfrage und weitere Angaben bestimmen.

Weitere Informationen über CloudTrail, einschließlich Konfiguration und Aktivierung, finden Sie im AWS CloudTrail User Guide.

Themen

- CloudWatch Logs-Informationen in CloudTrail (p. 149)
- Grundlagen zu Protokolldateieinträgen (p. 150)

CloudWatch Logs-Informationen in CloudTrail

CloudTrail wird beim Erstellen Ihres AWS-Kontos für Sie aktiviert. Die in CloudWatch Logs auftretenden unterstützten Aktivitäten werden als CloudTrail-Ereignis zusammen mit anderen AWS-Serviceereignissen in Event history (Ereignisverlauf) aufgezeichnet. Sie können die neusten Ereignisse in Ihr(em) AWS-Konto anzeigen, suchen und herunterladen. Weitere Informationen finden Sie unter Anzeigen von Ereignissen mit dem CloudTrail-API-Ereignisverlauf.

Erstellen Sie für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS-Konto, darunter Ereignisse für CloudWatch Logs, einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3-Bucket. Wenn Sie einen Pfad in der Konsole anlegen, gilt dieser standardmäßig für alle AWS-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS-Partition und stellt die Protokolldateien in dem Amazon S3-Bucket bereit, den Sie angeben. Darüber hinaus können Sie andere AWS-Services konfigurieren, um die in den CloudTrail-Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren. Weitere Informationen finden Sie unter:

- Übersicht zum Erstellen eines Pfads
- In CloudTrail unterstützte Services und Integrationen
- Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail-Protokolldateien aus mehreren Regionen und EmpfangenCloudTrail von Protokolldateien aus mehreren Konten

CloudWatch Logs unterstützt die Protokollierung der folgenden Aktionen als Ereignisse in CloudTrail-Protokolldateien:

- CancelExportTask
- CreateExportTask
- CreateLogGroup
- CreateLogStream
- DeleteDestination
- DeleteLogGroup
- DeleteLogStream
- DeleteMetricFilter
- DeleteRetentionPolicy
- DeleteSubscriptionFilter
- PutDestination
- PutDestinationPolicy
- PutMetricFilter
- PutRetentionPolicy
- PutSubscriptionFilter
- StartQuery
- StopQuery
- TestMetricFilter

Nur Anforderungselemente werden für diese CloudWatch Logs API-Aktionen in CloudTrail protokolliert:

- DescribeDestinations
- DescribeExportTasks
- DescribeLogGroups
- DescribeLogStreams
- DescribeMetricFilters
- DescribeQueries
- DescribeSubscriptionFilters
- GetLogGroupFields
- GetLogRecord

Jedes Event oder jeder Protokolleintrag enthält Informationen über den Ersteller der Anfrage. Anhand der Identitätsinformationen zur Benutzeridentität können Sie Folgendes bestimmen:

- Ob die Anfrage mit Root- oder AWS Identity and Access Management-Benutzeranmeldeinformationen (IAM) ausgeführt wurde.
- Ob die Anfrage mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen föderierten Benutzer ausgeführt wurde.
- Ob die Anforderung von einem anderen AWS-Service getätigt wurde.

Weitere Informationen finden Sie unter CloudTrail-Element "userIdentity".

Grundlagen zu Protokolldateieinträgen

Ein Trail ist eine Konfiguration, durch die Ereignisse an den von Ihnen angegebenen Amazon S3-Bucket übermittelt werden. CloudTrail-Protokolldateien können einen oder mehrere Einträge enthalten. Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält unter anderem

Informationen über die angeforderte Aktion, das Datum und die Uhrzeit der Aktion sowie über die Anfrageparameter. CloudTrail-Protokolldateien sind kein geordnetes Stacktrace der öffentlichen API-Aufrufe und erscheinen daher nicht in einer bestimmten Reihenfolge.

Der folgende Protokolldateieintrag zeigt, dass ein Benutzer die CloudWatch Logs-Aktion CreateExportTask aufgerufen hat.


```
{
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/someuser",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "someuser"
        },
        "eventTime": "2016-02-08T06:35:14Z",
        "eventSource": "logs.amazonaws.com",
        "eventName": "CreateExportTask",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-ruby2/2.0.0.rc4 ruby/1.9.3 x86_64-linux Seahorse/0.1.0",
        "requestParameters": {
            "destination": "yourdestination",
            "logGroupName": "yourloggroup",
            "to": 123456789012,
            "from": 0,
            "taskName": "yourtask"
        },
        "responseElements": {
            "taskId": "15e5e534-9548-44ab-a221-64d9d2b27b9b"
        },
        "requestID": "1cd74c1c-ce2e-12e6-99a9-8dbb26bd06c9",
        "eventID": "fd072859-bd7c-4865-9e76-8e364e89307c",
        "eventType": "AwsApiCall",
        "apiVersion": "20140328",
        "recipientAccountId": "123456789012"
}
```

Referenz für den CloudWatch Logs-Agenten

> **Important**
>
> Diese Referenz ist für ältere CloudWatch Logs -Agent, der sich auf dem Pfad zur Veralterung befindet. Wir empfehlen dringend, dass Sie die einheitliche CloudWatch Agenten statt. Weitere Informationen zu diesem Agenten finden Sie unter Sammeln von Metriken und Protokollen aus Amazon EC2 -Instance und lokale Server mit der CloudWatch Vertreter.

Der CloudWatch Logs-Agent bietet eine automatisierte Methode zum Senden von Protokolldaten an CloudWatch Logs von Amazon EC2-Instances aus. Der Agent enthält die folgenden Komponenten:

- Ein Plug-In in der AWS CLI, mit dem Protokolldaten an CloudWatch Logs übertragen werden.
- Ein Skript (Daemon), das dem Prozess den Anstoß gibt, Daten an CloudWatch Logs zu übertragen.
- Ein Cron-Auftrag, der sicherstellt, dass der Daemon ständig ausgeführt wird.

Agent-Konfigurationsdatei

Die CloudWatch Logs-Agenten-Konfigurationsdatei beschreibt Informationen, die für den CloudWatch Logs-Agent erforderlich sind. Im Abschnitt [general] der Agent-Konfigurationsdatei werden gängige Konfigurationen definiert, die für alle Protokoll-Streams gelten. Im Abschnitt [logstream] werden die erforderlichen Informationen zum Senden einer lokalen Datei an einen Remote-Protokoll-Stream definiert. Sie können mehrere [logstream]-Abschnitte definieren, von denen aber jeder einen eindeutigen Namen innerhalb der Konfigurationsdatei haben muss, z. B. [logstream1], [logstream2] und so weiter. Der Wert von [logstream] sowie die erste Datenzeile in der Protokolldatei definieren die Identität der Protokolldatei.

```
[general]
state_file = value
logging_config_file = value
use_gzip_http_content_encoding = [true | false]

[logstream1]
log_group_name = value
log_stream_name = value
datetime_format = value
time_zone = [LOCAL|UTC]
file = value
file_fingerprint_lines = integer | integer-integer
multi_line_start_pattern = regex | {datetime_format}
initial_position = [start_of_file | end_of_file]
encoding = [ascii|utf_8|..]
buffer_duration = integer
batch_count = integer
batch_size = integer

[logstream2]
...
```

state_file

Gibt an, wo die Zustandsdatei gespeichert ist.

logging_config_file

(Optional) Gibt an, wo sich die Konfigurationsdatei für die Agent-Protokollierung befindet. Wenn Sie keine Konfigurationsdatei für die Agent-Protokollierung angeben, wird immer die Standarddatei awslogs.conf verwendet. Der Standardspeicherort der Datei ist `/var/awslogs/etc/awslogs.conf`, wenn Sie den Agenten mit einem Skript installiert haben, und `/etc/awslogs/awslogs.conf`, wenn Sie den Agenten mit rpm installiert haben. Die Datei ist im Python-Format für Konfigurationsdateien (https://docs.python.org/2/library/logging.config.html#logging-config-fileformat) geschrieben. Logger mit den folgenden Namen können angepasst werden.

```
cwlogs.push
cwlogs.push.reader
cwlogs.push.publisher
cwlogs.push.event
cwlogs.push.batch
cwlogs.push.stream
cwlogs.push.watcher
```

Im unten stehenden Beispiel ändert sich die Ebene für Leser und Veröffentlicher auf WARNUNG, während der Standardwert INFO lautet.

```
[loggers]
keys=root,cwlogs,reader,publisher

[handlers]
keys=consoleHandler

[formatters]
keys=simpleFormatter

[logger_root]
level=INFO
handlers=consoleHandler

[logger_cwlogs]
level=INFO
handlers=consoleHandler
qualname=cwlogs.push
propagate=0

[logger_reader]
level=WARNING
handlers=consoleHandler
qualname=cwlogs.push.reader
propagate=0

[logger_publisher]
level=WARNING
handlers=consoleHandler
qualname=cwlogs.push.publisher
propagate=0

[handler_consoleHandler]
class=logging.StreamHandler
level=INFO
formatter=simpleFormatter
args=(sys.stderr,)

[formatter_simpleFormatter]
```

```
format=%(asctime)s - %(name)s - %(levelname)s - %(process)d - %(threadName)s -
 %(message)s
```

use_gzip_http_content_encoding

Wenn dieser Wert auf "true" (Standard) gesetzt ist, kann HTTP-Inhaltsverschlüsselung mit "gzip" komprimierte Nutzlasten an CloudWatch Logs senden. Dies verringert die CPU-Auslastung, reduziert NetworkOut und senkt die Put-Latenz. Um diese Funktion zu deaktivieren, fügen Sie use_gzip_http_content_encoding = false im Abschnitt [general] der CloudWatch Logs-Agenten-Konfigurationsdatei ein. Starten Sie dann den Agenten neu.

Note

Diese Einstellung ist nur awscli-cwlogs Version 1.3.3 und höher verfügbar.

log_group_name

Gibt die Ziel-Protokollgruppe an. Eine Protokollgruppe wird automatisch erstellt, sofern diese nicht bereits vorhanden ist. Protokollgruppennamen können zwischen 1 und 512 Zeichen lang sein. Zulässige Zeichen sind a – z, A – Z, 0 – 9, "_" (Unterstrich), "-" (Bindestrich), "/" (Schrägstrich) und "." (Punkt).

log_stream_name

Gibt den Ziel-Protokoll-Stream an. Sie können den Protokoll-Stream-Namen mithilfe einer Literalzeichenfolge oder den vordefinierten Variablen ({instance_id}, {hostname} und {ip_address}) oder einer Kombination aus beiden definieren. Ein Protokoll-Stream wird automatisch erstellt, sofern dieser nicht bereits vorhanden ist.

datetime_format

Gibt an, wie der Zeitstempel aus Protokollen extrahiert wird. Der Zeitstempel wird zum Abrufen von Protokollereignissen und Generieren von Metriken verwendet. Wenn datetime_format nicht angegeben ist, wird für die einzelnen Protokollereignisse die aktuelle Uhrzeit verwendet. Wenn der vorhandene Wert datetime_format für eine bestimmte Protokollnachricht ungültig ist, wird der Zeitstempel ab dem letzten Protokollereignis mit erfolgreich analysiertem Zeitstempel verwendet. Sind keine vorherigen Protokollereignisse vorhanden, wird die aktuelle Uhrzeit verwendet.

Die häufig verwendeten datetime_format-Codes sind unten aufgeführt. Sie können auch alle datetime_format-Codes verwenden, die von Python, datetime.strptime() unterstützt werden. Der Zeitzonenversatz (%z) wird ebenfalls unterstützt, wenn auch nur bis python 3.2, [+ -]HHMM ohne Doppelpunkt (:). Weitere Informationen finden Sie unter strftime()- und strptime ()-Verhalten.

y Jahr ohne Jahrhundert als mit Nullen aufgefüllte Dezimalzahl 00, 01, ..., 99

%Y : Jahr mit Jahrhundert als Dezimalzahl.1970, 1988, 2001, 2013

b Monat als abgekürzter Name des Gebietsschemas. Jan, Feb, ..., Dec (en_US);

%B Monat als vollständiger Name des Gebietsschemas. January, February, ..., December (en_US);

%-i Monat als mit Nullen aufgefüllte Dezimalzahl 01, 02, ..., 12

%-d Tag des Monats als mit Nullen aufgefüllte Dezimalzahl 01, 02, ..., 31

%H Stunde (24-Stunden-Format) als mit Nullen aufgefüllte Dezimalzahl. 00, 01, ..., 23

Prozentualer AnteilI: Stunde (12-Stunden-Uhr) als mit Nullen aufgefüllte Dezimalzahl. 01, 02, ..., 12

<p> : Regionales Äquivalent für AM oder PM.

%-M Minute als mit Nullen aufgefüllte Dezimalzahl. 00, 01, ..., 59

%-S Sekunde als eine mit Nullen aufgefüllte Dezimalzahl. 00, 01, ..., 59

f Mikrosekunde als Dezimalzahl, auf der linken Seite mit Nullen aufgefüllt. 000000, ..., 999999

%z UTC-Offset im Format +HHMM oder -HHMM. +0000, -0400, +1030

Beispielformate:

```
Syslog: '%b %d %H:%M:%S', e.g. Jan 23 20:59:29

Log4j: '%d %b %Y %H:%M:%S', e.g. 24 Jan 2014 05:00:00

ISO8601: '%Y-%m-%dT%H:%M:%S%z', e.g. 2014-02-20T05:20:20+0000
```

time_zone

Gibt die Zeitzone des Zeitstempels des Protokollereignisses an. Die beiden unterstützten Werte sind UTC und LOCAL. Standardmäßig wird LOCAL verwendet, wenn die Zeitzone nicht von datetime_format abgeleitet werden kann.

--file

Gibt die Protokolldateien an, die an CloudWatch Logs übertragen werden sollen. Die Datei kann auf eine bestimmte Datei oder mehrere Dateien (mit Platzhaltern wie /var/log/system.log*) verweisen. Nur die aktuelle Datei wird basierend auf dem Änderungsdatum der Datei an CloudWatch Logs übertragen. Wir empfehlen, dass Sie eine Reihe von Platzhaltern angeben, z. B. Dateien desselben Typs, access_log.2014-06-01-01, access_log.2014-06-01-02 und so weiter, aber nicht mehrere Arten von Dateien, wie z. B. access_log_80 und access_log_443. Wenn Sie mehrere Arten von Dateien angeben möchten, fügen Sie der Konfigurationsdatei einen anderen Protokoll-Stream-Eintrag hinzu, damit jede Art von Protokolldatei in verschiedene Protokoll-Streams gestellt wird. Komprimierte Dateien werden nicht unterstützt.

file_fingerprint_lines

Gibt den Bereich von Zeilen an, über die eine Datei identifiziert wird. Gültige Werte sind eine Zahl oder zwei durch Gedankenstriche getrennten Zahlen, wie "1", "2-5". Der Standardwert ist 1, damit die erste Zeile für die Berechnung des Fingerabdrucks verwendet wird. Fingerabdruck-Zeilen werden nicht an CloudWatch Logs gesendet, es sei denn, alle angegebenen Zeilen sind verfügbar.

multi_line_start_pattern

Gibt das Muster an, anhand dessen der Beginn einer Protokolldatei identifiziert wird. Eine Protokollmeldung besteht aus einer Zeile, die mit dem angegebenen Muster übereinstimmt, und allen folgenden Zeilen, die nicht dem Muster entsprechen. Gültige Werte sind reguläre Ausdrücke oder {datetime_format} Bei der Verwendung von {datetime_format} muss die Option "datetime_format" angegeben werden. Der Standardwert ist "^ [^\s]", sodass bei jeder Zeile, die mit Zeichen ohne Leerzeichen beginnt, die vorherige Protokollmeldung abgeschlossen wird und eine neue Protokollnachricht startet.

initial_position

Gibt an, wo der Anfang zum Lesen der Daten ist (start_of_file oder end_of_file). Der Standardwert ist start_of_file. Es wird nur verwendet, wenn für diesen Protokoll-Stream kein Zustand besteht.

encoding

Gibt die Verschlüsselung der Protokolldatei an, damit die Datei korrekt gelesen werden kann. Der Standardwert ist utf_8. Hier können von Python codecs.decode() unterstützte Verschlüsselungen verwendet werden.

> Warning
>
> Die Angabe einer fehlerhaften Kodierung kann zu Datenverlust führen, weil Zeichen, die nicht dekodiert werden können, durch ein anderes Zeichen ersetzt werden.

Im Folgenden sind einige gängige Kodierungen aufgeführt:

```
ascii, big5, big5hkscs, cp037, cp424, cp437, cp500, cp720, cp737, cp775,
cp850, cp852, cp855, cp856, cp857, cp858, cp860, cp861, cp862, cp863, cp864,
cp865, cp866, cp869, cp874, cp875, cp932, cp949, cp950, cp1006, cp1026,
cp1140, cp1250, cp1251, cp1252, cp1253, cp1254, cp1255, cp1256, cp1257,
cp1258, euc_jp, euc_jis_2004, euc_jisx0213, euc_kr, gb2312, gbk, gb18030,
hz, iso2022_jp, iso2022_jp_1, iso2022_jp_2, iso2022_jp_2004, iso2022_jp_3,
iso2022_jp_ext, iso2022_kr, latin_1, iso8859_2, iso8859_3, iso8859_4,
iso8859_5, iso8859_6, iso8859_7, iso8859_8, iso8859_9, iso8859_10,
iso8859_13, iso8859_14, iso8859_15, iso8859_16, johab, koi8_r, koi8_u,
mac_cyrillic, mac_greek, mac_iceland, mac_latin2, mac_roman, mac_turkish,
ptcp154, shift_jis, shift_jis_2004, shift_jisx0213, utf_32, utf_32_be,
utf_32_le, utf_16, utf_16_be, utf_16_le, utf_7, utf_8, utf_8_sig
```

buffer_duration

Gibt die Dauer für die Stapelverarbeitung von Protokollereignissen an. Der Mindestwert und der Standardwert ist 5 000ms.

batch_count

Gibt die maximale Anzahl der Protokollereignisse in einem Stapel an, bis zu 10.000. Der Standardwert lautet 10.000.

batch_size

Gibt die maximale Größe von Protokollereignissen in einem Stapel in Byte an, bis zu 1048576 Byte. Der Standardwert lautet 1048576 Bytes. Diese Größe ist die Summe aller Ereignismeldungen in UTF-8 plus 26 Bytes pro Protokollereignis.

Verwenden des CloudWatch Logs-Agent mit HTTP-Proxys

Sie können den CloudWatch Logs-Agent mit HTTP-Proxys verwenden.

Note

HTTP-Proxys werden in awslogs-agent-setup.py Version 1.3.8 oder höher unterstützt.

So verwenden Sie den CloudWatch Logs-Agent mit HTTP-Proxys

1. Führen Sie eine der folgenden Aufgaben aus:

   a. Für eine neue Installation des CloudWatch Logs-Agent führen Sie die folgenden Befehle aus:

   ```
   curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-
   setup.py -O
   ```

   ```
   sudo python awslogs-agent-setup.py --region us-east-1 --http-proxy http://your/
   proxy --https-proxy http://your/proxy --no-proxy 169.254.169.254
   ```

   Um den Zugriff auf die Amazon EC2-Metadatenservice für EC2-Instances zu behalten, verwenden Sie --no-proxy 169.254.169.254 (empfohlen). Weitere Informationen finden Sie unter Instance-Metadaten und Benutzerdaten im Amazon EC2-Benutzerhandbuch für Linux-Instances.

   Geben Sie in den Werten für `http-proxy` und `https-proxy` die gesamte URL ein.

b. Zum Bearbeiten einer vorhandenen Installation des CloudWatch Logs-Agent fügen Sie die Proxys in /var/awslogs/etc/proxy.conf hinzu:

```
HTTP_PROXY=
HTTPS_PROXY=
NO_PROXY=
```

2. Starten Sie den Agenten, damit die Änderungen wirksam werden:

```
sudo service awslogs restart
```

Wenn Sie Amazon Linux 2 verwenden, verwenden Sie den folgenden Befehl, um den Agenten neu zu starten:

```
sudo service awslogsd restart
```

Aufgliedern der Konfigurationsdateien für den CloudWatch Logs-Agent

Wenn Sie awslogs-agent-setup.py Version 1.3.8 oder höher mit awscli-cwlogs 1.3.3 oder höher verwenden, können Sie verschiedene Stream-Konfigurationen für verschiedene Komponenten unabhängig voneinander importieren, indem Sie zusätzliche Konfigurationsdateien im Verzeichnis /var/awslogs/etc/config/ erstellen. Wenn der CloudWatch Logs-Agent gestartet wird, umfasst er die Stream-Konfigurationen in diesen zusätzlichen Konfigurationsdateien. Konfigurationseigenschaften im Abschnitt [general] müssen in der Haupt-Konfigurationsdatei (/var/awslogs/etc/awslogs.conf) definiert werden und werden in den anderen unter /var/awslogs/etc/config/ vorhandenen Konfigurationsdateien ignoriert.

Wenn Sie kein /var/awslogs/etc/config/-Verzeichnis haben, weil Sie den Agenten mit rpm installiert haben, können Sie stattdessen das Verzeichnis /etc/awslogs/config/ verwenden.

Starten Sie den Agenten, damit die Änderungen wirksam werden:

```
sudo service awslogs restart
```

Wenn Sie Amazon Linux 2 verwenden, verwenden Sie den folgenden Befehl, um den Agenten neu zu starten:

```
sudo service awslogsd restart
```

CloudWatch Logs-Agent – Häufig gestellte Fragen

Welche Arten von Dateirotationen werden unterstützt?

Folgende Dateirotationsmechanismen werden unterstützt:

- Umbenennen vorhandener Protokolldateien mit einem numerischen Suffix, anschließendes Neuerstellen der ursprünglichen leeren Protokolldatei. Z. B. /var/log/syslog.log is renamed /var/log/syslog.log.1. Wenn /var/log/syslog.log.1 bereits aus einem vorherigen Durchgang vorhanden ist, wird es in /var/log/syslog.log.2 umbenannt.
- Kürzen der ursprünglichen Protokolldatei nach der Erstellung einer Kopie. Beispielsweise wird /var/log/syslog.log in /var/log/syslog.log.1 kopiert und /var/log/syslog.log wird gekürzt. In diesem

Fall können Datenverluste entstehen. Verwenden Sie diesen Dateirotationsmechanismus also mit Bedacht.

- Erstellen einer neuen Datei mit dem gemeinsamen Muster der alten Datenbank. Beispielsweise bleibt /var/log/syslog.log.2014-01-01 bestehen und /var/log/syslog.log.2014-01-02 wird erstellt.

Der Fingerabdruck (Quell-ID) der Datei wird berechnet, indem ein Hash für den Protokoll-Stream-Schlüssel und die erste Zeile in der Datei durchgeführt wird. Zum Überschreiben dieses Verhaltens kann die Option file_fingerprint_lines verwendet werden. Bei einer Rotation sollte die neue Datei neue Inhalte haben, und die alte Datei sollte keine angehängten Inhalte haben. Der Agent überträgt die neue Datei, wenn der Lesevorgang der alten Datei abgeschlossen ist.

Wie kann ich bestimmen, welche Agent-Version ich verwende?

Wenn Sie ein Setupskript für die Installation des CloudWatch Logs-Agenten verwendet haben, können Sie mit /var/awslogs/bin/awslogs-version.sh prüfen, welche Agent-Version Sie verwenden. Dabei werden die Version des Agent und seine großen Abhängigkeiten gedruckt. Wenn Sie den CloudWatch Logs-Agenten mit yum installiert haben, können Sie die Version des -Agent und Plug-Ins mit "yum info awslogs" und "yum info aws-cli-plugin-cloudwatch-logs"CloudWatch Logs prüfen.

Wie werden Protokolleinträge in Protokollereignisse umgewandelt?

Protokollereignisse enthalten zwei Eigenschaften: den Zeitstempel mit Datum und Uhrzeit des Ereignisses und die reine Protokollnachricht. Standardmäßig wird bei jeder Zeile , die mit Zeichen ohne Leerzeichen beginnt, die vorherige Protokollnachricht (falls vorhanden) abgeschlossen und eine neue Protokollnachricht gestartet. Um dieses Verhalten zu überschreiben, kann multi_line_start_pattern verwendet werden, und bei jeder Zeile, die mit dem Muster übereinstimmt, wird eine neue Protokollnachricht gestartet. Das Muster ist ein regulärer Ausdruck oder '{datetime_format}'. Wenn beispielsweise die erste Zeile in jeder Protokollnachricht einen Zeitstempel wie z. B."2014-01-02T13:13:01 Z" enthält, kann das multi_line_start_pattern auf "\d\{4\}-\d\{2\}-\d\{2\}T \d\{2\}:\d\{2\}:\d\{2\}Z" gesetzt werden. Zum Vereinfachen der Konfiguration, kann die Variable '{datetime_format}' verwendet werden, wenn datetime_format option angegeben ist. Für dasselbe Beispiel gilt, wenn datetime_format auf '%Y-%m-%dT%H:%M:%S%z' gesetzt ist, dann könnte multi_line_start_pattern einfach '{datetime_format}' sein.

Wenn datetime_format nicht angegeben ist, wird für die einzelnen Protokollereignisse die aktuelle Uhrzeit verwendet. Wenn der vorhandene Wert datetime_format für eine bestimmte Protokollnachricht ungültig ist, wird der Zeitstempel ab dem letzten Protokollereignis mit erfolgreich analysiertem Zeitstempel verwendet. Sind keine vorherigen Protokollereignisse vorhanden, wird die aktuelle Uhrzeit verwendet. Eine Warnmeldung wird protokolliert, wenn ein Protokollereignis auf die aktuelle Uhrzeit oder den Zeitpunkt des vorherigen Protokollereignisses zurückgreift.

Zeitstempel dienen zum Abrufen von Protokollereignissen und Generieren von Metriken. Wenn Sie das falsche Format angeben, könnten die Protokollereignisse nicht abrufbar werden, und es werden falsche Metriken generiert.

Wie werden Protokollereignisse im Stapel verarbeitet?

Eine Stapel ist voll und wird veröffentlicht, wenn eine der folgenden Bedingungen erfüllt ist:

1. Der Zeitraum buffer_duration ist abgelaufen, nachdem das erste Protokollereignis hinzugefügt wurde.
2. Weniger als batch_size der Protokollereignisse wurden akkumuliert, aber durch das Hinzufügen des neuen Protokollereignisses wird batch_size überschritten.
3. Die Anzahl der Protokollereignisse hat batch_count erreicht.
4. Protokollereignisse aus dem Stapel umfassen nie mehr als 24 Stunden. Aber durch das Hinzufügen des neuen Protokollereignisses wird die 24-Stunden-Bedingung überschritten.

Wodurch werden Protokolleinträge, Protokollereignisse oder Stapel übersprungen oder gekürzt?

Zur Einhaltung der Bedingung für die `PutLogEvents`-Operation können folgende Probleme dazu führen, dass ein Protokollereignis oder ein Stapel übersprungen wird.

Note

Der CloudWatch Logs-Agent schreibt eine Warnung in das Protokoll, wenn Daten übersprungen werden.

1. Wenn das Protokollereignis größer als 256 KB ist, wird es vollständig übersprungen.
2. Wenn der Zeitstempel des Protokollereignisses mehr als 2 Stunden in der Zukunft liegt, wird das Protokollereignis übersprungen.
3. Wenn der Zeitstempel des Protokollereignisses mehr als 14 Stunden in der Vergangenheit liegt, wird das Protokollereignis übersprungen.
4. Wenn ein Protokollereignis älter als der Aufbewahrungszeitraum der Protokollgruppe ist, wird der gesamte Stapel übersprungen.
5. Wenn der Stapel von Protokollereignissen in einer einzigen `PutLogEvents`-Anforderung mehr als 24 Stunden umfasst, schlägt die `PutLogEvents`-Operation fehl.

Führt das Anhalten des Agent zu Datenverlust/Duplikaten?

Nicht, solange die Zustandsdatei verfügbar ist und seit der letzten Ausführung keine Dateirotation stattgefunden hat. Der CloudWatch Logs-Agent kann an der Stelle fortgesetzt werden, an der er angehalten wurde, und die Protokolldaten weiterhin übertragen.

Kann ich verschiedene Protokolldateien aus demselben Host oder aus unterschiedlichen Hosts demselben Protokoll-Stream zuweisen?

Die Konfiguration von mehreren Protokollquellen, damit Daten an einen einzelnen Protokoll-Stream gesendet werden, wird nicht unterstützt.

Welche API-Aufrufe führt der Agent durch (oder welche Aktionen sollte ich meiner IAM-Richtlinie hinzufügen)?

Die Schaltfläche CloudWatch Logs Agent erfordert die `CreateLogGroup`, , und Sie haben die Möglichkeit `CreateLogStream`, , und Sie haben die Möglichkeit `DescribeLogStreams`, und `PutLogEvents` Vorgänge. Wenn Sie den neuesten Agent verwendet, wird `DescribeLogStreams` nicht mehr benötigt. Weitere Informationen finden Sie im Beispiel für eine IAM-Richtlinie weiter unten.

```
{
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogGroup",
      "logs:CreateLogStream",
      "logs:PutLogEvents",
      "logs:DescribeLogStreams"
    ],
    "Resource": [
      "arn:aws:logs:*:*:*"
    ]
  }
 ]
}
```

Ich möchte nicht, dass der CloudWatch Logs-Agent Protokollgruppen oder Protokoll-Streams automatisch erstellt. Wie kann ich verhindern, dass der Agent Protokollgruppen und Protokoll-Streams neu erstellt?

In Ihrem IAM können Sie den Agenten auf die folgenden Operationen beschränken: `DescribeLogStreams`, , und Sie haben die Möglichkeit `PutLogEvents`.

Bevor Sie die Berechtigungen `CreateLogGroup` und `CreateLogStream` vom Agenten entziehen, müssen Sie sowohl die Protokollgruppen als auch die Protokolldatenströme erstellen, die der Agent verwenden soll. Der Protokoll-Agent kann keine Protokolldatenströme in einer Protokollgruppe

erstellen, die Sie erstellt haben, es sei denn, er verfügt über die Berechtigungen `CreateLogGroup` und `CreateLogStream`.

Welche Protokolle sollte ich bei der Fehlerbehebung beachten?

Das Installationsprotokoll des Agenten finden Sie unter `/var/log/awslogs-agent-setup.log` und das Protokoll des Agenten unter `/var/log/awslogs.log`.

Überwachung der Nutzung mit CloudWatch-Metriken

CloudWatch Logs sendet einmal pro Minute Metriken an Amazon CloudWatch.

CloudWatch Logs-Metriken

Der `AWS/Logs`-Namespace enthält die folgenden Metriken.

Metrik	Beschreibung
`IncomingBytes`	Die Menge der Protokollereignisse in nicht komprimierten Byte, die auf CloudWatch Logs hochgeladen wurden. In Verbindung mit der Dimension `LogGroupName` ist dies die Menge der Protokollereignisse in nicht komprimierten Byte, die in die Protokollgruppe hochgeladen wurden. Gültige Dimensionen: LogGroupName Gültige Statistiken: Summe Einheiten: Byte
`IncomingLogEvents`	Die Anzahl der Protokollereignisse, die auf CloudWatch Logs hochgeladen wurden. In Verbindung mit der Dimension `LogGroupName` ist dies die Anzahl der Protokollereignisse, die in die Protokollgruppe hochgeladen wurden. Gültige Dimensionen: LogGroupName Gültige Statistiken: Summe Einheiten: keine
`ForwardedBytes`	Die Menge der Protokollereignisse in komprimierten Byte, die zum Abonnementziel weitergeleitet wurden. Gültige Dimensionen: LogGroupName, DestinationType, FilterName Gültige Statistiken: Summe Einheiten: Byte
`ForwardedLogEvents`	Die Zahl der Protokollereignisse, die zum Abonnementziel weitergeleitet wurden. Gültige Dimensionen: LogGroupName, DestinationType, FilterName Gültige Statistiken: Summe Einheiten: keine

Metrik	Beschreibung
`DeliveryErrors`	Die Zahl der Protokollereignisse, für die CloudWatch Logs beim Weiterleiten der Daten an das Abonnementziel eine Fehlermeldung erhalten hat. Gültige Dimensionen: LogGroupName, DestinationType, FilterName Gültige Statistiken: Summe Einheiten: keine
`DeliveryThrottling`	Die Zahl der Protokollereignisse, für die CloudWatch Logs beim Weiterleiten der Daten an das Abonnementziel gedrosselt wurde. Gültige Dimensionen: LogGroupName, DestinationType, FilterName Gültige Statistiken: Summe Einheiten: keine

Dimensionen für CloudWatch Logs-Metriken

Die Dimensionen, die Sie mit CloudWatch Logs-Metriken verwenden können, sind nachstehend aufgeführt.

Dimension	Beschreibung
`LogGroupName`	Der Name der CloudWatch Logs-Protokollgruppe, für die Metriken angezeigt werden sollen.
`DestinationType`	Das Abonnementziel für die CloudWatch Logs-Daten. Mögliche Werte sind AWS Lambda, Amazon Kinesis Data Streams und Amazon Kinesis Data Firehose.
`FilterName`	Der Name des Abonnementfilters, der Daten aus der Protokollgruppe an das Ziel weiterleitet. Der Abonnementfiltername wird von CloudWatch automatisch in ASCII konvertiert. Alle nicht unterstützten Zeichen werden dabei durch ein Fragezeichen (?) ersetzt.

CloudWatch Logs-Kontingente

CloudWatch Logs hat folgende Kontingente:

Ressource	Standardkontingent
Stapel-Größe	1 MB (maximal). Dieses Kontingent kann nicht geändert werden.
Datenarchivierung	Bis zu 5 GB kostenlose Datenarchivierung. Dieses Kontingent kann nicht geändert werden.
CreateLogStream	50 Transaktionen pro Sekunde (TPS/Konto/Region), nach denen Transaktionen gedrosselt werden. Sie können eine Kontingenterhöhung beantragen.
DescribeLogGroups	5 Transaktionen pro Sekunde (TPS/Konto/Region). Sie können eine Kontingenterhöhung beantragen.
DescribeLogStreams	5 Transaktionen pro Sekunde (TPS/Konto/Region). Sie können eine Kontingenterhöhung beantragen.
Erkannte Protokollfelder	CloudWatch Logs Insights kann maximal 1000 Protokollereignisfelder in einer Protokollgruppe ermitteln. Dieses Kontingent kann nicht geändert werden. Weitere Informationen finden Sie unter Unterstützte Protokolle und erkannte Felder (p. 36).
Extrahierte Protokollfelder in JSON-Protokollen	CloudWatch Logs Insights kann maximal 100 Protokollereignisfelder aus einem JSON-Protokoll extrahieren. Dieses Kontingent kann nicht geändert werden. Weitere Informationen finden Sie unter Unterstützte Protokolle und erkannte Felder (p. 36).
Ereignis-Größe	256 KB (maximal). Dieses Kontingent kann nicht geändert werden.
Exportaufgabe	Eine aktive (laufende oder ausstehende) Exportaufgabe pro Konto. Dieses Kontingent kann nicht geändert werden.
FilterLogEvents	5 Transaktionen pro Sekunde (TPS)/Konto/Region. Dieses Kontingent kann nicht geändert werden.
GetLogEvents	10 Anforderungen pro Sekunde pro Konto pro Region. Dieses Kontingent kann nicht geändert werden. Wir empfehlen Abonnements, wenn Sie fortwährend neue Daten verarbeiten. Wenn Sie historische Daten benötigen, empfiehlt es sich, Ihre Daten in Amazon S3 zu exportieren.
Eingehende Daten	Bis zu 5 GB eingehende Daten, kostenlos. Dieses Kontingent kann nicht geändert werden.

Ressource	Standardkontingent
Protokollgruppen	20,000 Protokollgruppen pro Konto pro Region. Sie können eine Kontingenterhöhung beantragen. Es gibt kein Kontingent dazu, wie viele Protokoll-Streams zu einer Protokollgruppe gehören können.
Metrikfilter	100 pro Protokollgruppe. Dieses Kontingent kann nicht geändert werden.
Metriken im eingebetteten Metrikformat	100 Metriken pro Protokollereignis und 9 Dimensionen pro Metrik. Weitere Informationen zum eingebetteten Metrikformat finden Sie unter Specification: Embedded Metric Format im Amazon CloudWatch-Benutzerhandbuch.
PutLogEvents	5 Anfragen pro Sekunde und Protokollstream. Darüber hinausgehende Anfragen werden gedrosselt. Dieses Kontingent kann nicht geändert werden. Die maximale Stapelgröße einer PutLogEvents-Anfrage beträgt 1 MB. 800 Transaktionen pro Sekunde pro Konto pro Region, mit Ausnahme der folgenden Regionen, in denen das Kontingent 1500 Transaktionen pro Sekunde pro Konto pro Region beträgt: USA Ost (Nord-Virginia), USA West (Oregon) und Europa (Irland). Sie können eine Kontingenterhöhung beantragen.
Zeitüberschreitung für die Abfrageausführung	Abfragen in CloudWatch Logs Insights werden nach 15 Minuten beendet. Dieses Zeitlimit kann nicht geändert werden.
Abgefragte Protokollgruppen	Es können maximal 20 Protokollgruppen in einer einzigen CloudWatch Logs Insights-Abfrage abgefragt werden. Dieses Kontingent kann nicht geändert werden.
Gleichzeitigkeit von Abfragen	Maximal 4 gleichzeitige CloudWatch Logs Insights-Abfragen, einschließlich Abfragen, die zu Dashboards hinzugefügt wurden. Sie können eine Kontingenterhöhung beantragen.
Verfügbarkeit von Abfrageergebnissen	Ergebnisse aus einer Abfrage können 7 Tage lang abgerufen werden. Dieser Verfügbarkeitszeitraum kann nicht geändert werden.
Anzeige der Abfrageergebnisse in der Konsole	Standardmäßig werden bis zu 1000 Zeilen mit Abfrageergebnissen in der Konsole angezeigt. Sie können den **`limit`**-Befehl in einer Abfrage verwenden, um dies auf bis zu 10.000 Zeilen zu erhöhen. Weitere Informationen finden Sie unter CloudWatch Logs Insights-Abfragesyntax (p. 42).
Gespeicherte Abfragen	Sie können bis zu 1000 CloudWatch Logs Insights-Abfragen pro Region und Konto speichern. Dieses Kontingent kann nicht geändert werden.
Abonnement-Filter	1 pro Protokollgruppe. Dieses Kontingent kann nicht geändert werden.

Dokumentverlauf

In der folgenden Tabelle sind wichtige Änderungen in jeder Version des CloudWatch Logs-Benutzerhandbuchs seit Juni 2018 beschrieben. Um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten, können Sie einen RSS-Feed abonnieren.

update-history-change	update-history-description	update-history-date
CloudWatch Logs-Insights freigegeben (p. 165)	Mit CloudWatch Logs Insights können Sie interaktiv Ihre Protokolldaten durchsuchen und analysieren. Weitere Informationen finden Sie unter Analysieren von Protokolldaten mit CloudWatch Logs Insights im Amazon CloudWatch Logs User Guide.	November 27, 2018
Support für Amazon VPC-Endpunkte (p. 165)	Sie können jetzt eine private Verbindung zwischen Ihrer VPC und CloudWatch Logs herstellen. Weitere Informationen finden Sie unter Verwenden von CloudWatch Logs mit Schnittstellen-VPC-Endpunkten im Amazon CloudWatch Logs User Guide.	June 28, 2018

In der folgenden Tabelle werden die wichtigen Änderungen am Amazon CloudWatch Logs-Benutzerhandbuch beschrieben.

Ändern	Beschreibung	Veröffentlichungsdatum
Schnittstellen-VPC-Endpunkte	In manchen Regionen können Sie einen Schnittstellen-VPC-Endpunkt verwenden, um zu verhindern, dass Datenverkehr zwischen Ihrer Amazon VPC und CloudWatch Logs das Amazon-Netzwerk verlässt. Weitere Informationen finden Sie unter Verwenden von CloudWatch Logs mit Schnittstellen-VPC-Endpunkten (p. 145).	7. März 2018
Route 53-DNS-Abfrageprotokolle	Sie können mit CloudWatch Logs Protokolle über die von Route 53 empfangenen DNS-Abfragen speichern. Weitere Informationen finden Sie unter Was ist Amazon CloudWatch Logs? (p. 1) oder Protokollieren von DNS-Abfragen im Entwicklerhandbuch für Amazon Route 53.	7. September 2017
Tag-Protokollgruppen	Sie können Tags verwenden, um Ihre Protokollgruppen zu kategorisieren. Weitere Informationen finden Sie unter Protokollgruppen in Amazon CloudWatch Logs kennzeichnen (p. 64).	13. Dezember 2016

Ändern	Beschreibung	Veröffentlichungsdatum
Verbesserungen an der Konsole	Sie können von Metrikdiagrammen zu den zugehörigen Protokollgruppen navigieren. Weitere Informationen finden Sie unter Wechseln von Metriken zu Protokollen (p. 63).	7. November 2016
Verbesserungen an der Benutzerfreundlichkeit der Konsole	Verbesserte Benutzererfahrung für einfacheres Suchen, Filtern und Fehlerbeheben. Beispielsweise können Sie jetzt Ihre Protokolldaten nach Datum und Uhrzeit filtern. Weitere Informationen finden Sie unter An CloudWatch Logs gesendete Protokolldaten anzeigen lassen (p. 61).	29. August 2016
AWS CloudTrail-Support für Amazon CloudWatch Logs- und neue CloudWatch Logs-Metriken hinzugefügt	AWS CloudTrail wird nun für CloudWatch Logs unterstützt. Weitere Informationen finden Sie unter Protokollieren von Amazon CloudWatch Logs-API-Aufrufen in AWS CloudTrail (p. 149).	10. März 2016
Support für CloudWatch Logs-Export nach Amazon S3 hinzugefügt.	Support für das Exportieren von CloudWatch Logs-Daten in Amazon S3 hinzugefügt. Weitere Informationen finden Sie unter Exportieren von Protokolldaten in Amazon S3 (p. 112).	7. Dezember 2015
Unterstützung für AWS CloudTrail-Ereignisprotokollierung in Amazon CloudWatch Logs hinzugefügt.	Sie können Alarme in CloudWatch erstellen und Benachrichtigungen von bestimmten von CloudTrail erfassten API-Aktivitäten erhalten und anhand der Benachrichtigungen eine Fehlerbehebung durchführen.	10. November 2014
Unterstützung für Amazon CloudWatch Logs hinzugefügt	Sie können Amazon CloudWatch Logs zur Überwachung, Speicherung und für den Zugriff auf Ihre System-, Anwendungs- und benutzerdefinierten Protokolldateien aus Amazon Elastic Compute Cloud (Amazon EC2)-Instances oder anderen Quellen verwenden. Anschließend können Sie die zugehörigen Protokolldaten aus CloudWatch Logs mit der Amazon CloudWatch-Konsole, den CloudWatch Logs-Befehlen in der AWS-CLI oder dem CloudWatch Logs-SDK abrufen. Weitere Informationen finden Sie unter Was ist Amazon CloudWatch Logs? (p. 1).	10. Juli 2014

AWS-Glossar

Die aktuelle AWS-Terminologie finden Sie im AWS-Glossar im AWS General Reference.

Sofern wir eine Übersetzung der englischsprachigen Version des Handbuchs bereitstellen, gilt im Fall von Widersprüchen die englischsprachige Version des Handbuchs. Bei der Übersetzung handelt es sich um eine maschinelle Übersetzung.