aws

Leitfaden

Amazon Simple Storage Service

API-Version 2006-03-01

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Simple Storage Service: Leitfaden

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Marken und Handelsmarken von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, die geeignet ist, Kunden irrezuführen oder Amazon in irgendeiner Weise herabzusetzen oder zu diskreditieren. Alle anderen Marken, die nicht im Besitz von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist Amazon S3? 1
Funktionen von Amazon S3 1
Speicherklassen 1
Speicherverwaltung 2
Zugriffsverwaltung und Sicherheit 3
Datenverarbeitung 4
Speicherprotokollierung und Überwachung 5
Analytik und Einblicke 5
Starke Konsistenz 6
Funktionsweise von Amazon S3 6
Buckets 7
Objekte 8
Schlüssel 8
S3-Versioning 8
Versions-ID 9
Bucket-Richtlinie 9
S3-Zugriffspunkte 10
Zugriffskontrolllisten (ACLs) 10
Regionen 11
Amazon S3-Datenkonsistenzmodell 11
Gleichzeitige Anwendungen 12
Zugehörige Services 14
Zugriff auf Amazon S3 15
AWS Management Console 15
AWS Command Line Interface 15
AWS SDKs 15
Amazon S3 REST API 16
Bezahlen für Amazon S3 16
PCIDSSEinhaltung der Vorschriften 17
Erste Schritte 18
Einrichtung 18
Melden Sie sich an für ein AWS-Konto 19
Erstellen eines Benutzers mit Administratorzugriff 19
Schritt 1: Erstellen eines Buckets 21

Schritt 2: Hochladen eines Objekts ... 28
Schritt 3: Herunterladen eines Objekts ... 29
Verwenden der S3-Konsole ... 29
Schritt 4: Kopieren eines Objekts ... 30
Schritt 5: Löschen von Objekt und Bucket ... 32
Löschen eines Objekts ... 33
Leeren des Buckets ... 33
Löschen des Buckets ... 34
Nächste Schritte ... 34
Verstehen allgemeiner Anwendungsfälle ... 35
Kontrollieren des Zugriffs auf Buckets und Objekte ... 35
Schützen und überwachen Sie Ihren Speicher ... 36
Entwickeln mit Amazon S3 ... 37
Erwerben Sie neue Kenntnisse aus Tutorials ... 38
Lernen Sie Trainings und Support kennen ... 40
Arbeiten mit Buckets ... 41
Bucket-Übersicht ... 42
Übersicht über Allzweck-Buckets ... 43
Allgemeine Bucket-Muster ... 44
Berechtigungen ... 44
Verwalten des öffentlichen Zugriffs auf Buckets ... 45
Bucket-Konfiguration ... 46
Bucket-Operationen ... 50
Allgemeine Bucket-Muster ... 50
Bucket-Muster für mehrere Mandanten ... 51
Bucket-per-use Muster ... 52
Benennungsregeln ... 53
Regeln für die Benennung von Buckets für allgemeine Zwecke ... 53
Beispiel für Namen von Buckets für allgemeine Zwecke ... 55
Bewährte Methoden ... 56
Einen Bucket erstellen, der ein GUID im Bucket-Namen verwendet ... 57
Kontingente, Einschränkungen und Beschränkungen ... 58
Bucket-Kontingente ... 58
Einschränkungen für Objekte und Buckets ... 59
Regeln für die Benennung von Buckets ... 59
Zugriff auf einen Bucket ... 59

..... 59
Virtuelles Hosting bei Buckets 63
Erstellen eines Buckets 73
Anzeigen von Bucket-Eigenschaften 87
Auflisten von Buckets 90
Leeren eines Buckets 96
Leeren eines Buckets mit den folgenden Einstellungen AWS CloudTrail 99
Löschen eines Buckets 100
Mountpoint für Amazon S3 105
Mountpoint installieren 106
Konfiguration und Verwendung von Mountpoint 111
Fehlerbehebung bei Mountpoint 116
Konfigurieren der Transferbeschleunigung 117
Gründe für die Nutzung von Amazon S3 Transfer Acceleration 117
Anforderungen für die Verwendung von Transfer Acceleration 118
Erste Schritte 119
Aktivieren der Transfer Acceleration 122
Speed Comparison Tool 129
Verwenden von Zahlung durch den Anforderer 130
Die Gebühren bei Zahlung durch den Anforderer 131
Konfigurieren von Zahlung durch den Anforderer 132
Konfiguration wird abgerufen requestPayment 134
Objekte aus Requester Pays-Buckets werden heruntergeladen 135
Arbeiten mit Objekten 137
Überblick über Objekte 138
Subressourcen 140
Benennen von Objekten 140
Auswahl von Objektschlüsselnamen 141
Richtlinien für Objektschlüsselnamen 141
Arbeiten mit Metadaten 145
Systemdefinierte Objektmetadaten 145
Benutzerdefinierte Objektmetadaten 149
Objektmetadaten bearbeiten 151
Objekte hochladen 155
Hochladen eines Objekts 156
Verhindern Sie das Hochladen von Objekten mit identischen Schlüsselnamen 170

Verwenden von mehrteiligen Uploads ... 170
Bedingte Anfragen stellen ... 247
Wie ruft man Objekte auf der Grundlage von Metadaten ab oder kopiert sie ... 248
Wie verhindert man Objektüberschreibungen ... 250
Objekte kopieren, verschieben und umbenennen ... 254
Ein Objekt kopieren ... 257
So verschieben Sie ein Objekt ... 268
Um ein Objekt umzubenennen ... 270
Herunterladen von Objekten ... 271
Herunterladen eines Objekts ... 272
Herunterladen mehrerer Objekte ... 274
Herunterladen eines Teils eines Objekts ... 276
Herunterladen eines Objekts von einem anderen AWS-Konto ... 277
Herunterladen von archivierten Objekten ... 278
Objekte auf der Grundlage von Metadaten herunterladen ... 278
Fehlerbehebung beim Herunterladen von Objekten ... 279
Überprüfung der Objektintegrität ... 279
Verwenden unterstützter Prüfsummenalgorithmen ... 279
Inhalte verwenden — MD5 beim Hochladen von Objekten ... 289
Verwenden Sie Content- MD5 und the, um hochgeladene ETag Objekte zu verifizieren ... 289
Verwenden von nachfolgenden Prüfsummen ... 290
Verwenden von Prüfsummen auf Teilebene für mehrteilige Uploads ... 290
Löschen von Objekten ... 292
Bewährte Methoden, die Sie vor dem Löschen eines Objekts beachten sollten ... 293
Löschen von Objekten aus einem Bucket mit aktivierter Versionierung ... 294
Löschen von Objekten aus einem Bucket, für das die Versionierung gesperrt wurde ... 295
Objekte aus einem Bucket ohne Versionsnummer löschen ... 295
Objekte aus einem Bucket mit MFA -aktiviertem Bucket löschen ... 295
Löschen eines einzelnen Objekts ... 296
Löschen von mehreren Objekten ... 308
Organisieren und Auflisten von Objekten ... 311
Verwenden von Präfixen ... 311
Auflisten von Objekten ... 314
Verwenden von Ordnern ... 317
Anzeigen von Objekteigenschaften ... 321
Objekte mit Tags kategorisieren ... 323

Verwenden von presigned URLs zum Herunterladen und Hochladen von Objekten ... 337
Wer kann ein vorsigniertes erstellen URL ... 338
Ablaufzeit für vorsignierte URLs ... 339
Einschränkung vordefinierter Funktionen URL ... 340
Objekte mit vorsignierten teilen URLs ... 342
Objekte mit Presigned hochladen URLs ... 345
Transformieren von Objekten ... 347
Erstellen von Objekt-Lambda-Zugriffspunkten ... 349
Verwenden von Amazon S3 Objekt-Lambda-Zugriffspunkten ... 365
Sicherheitsüberlegungen ... 369
Schreiben von Lambda-Funktionen ... 377
Verwenden von AWS eingebauten Funktionen ... 409
Bewährte Methoden und Richtlinien für S3 Object Lambda ... 411
Tutorials zu S3 Object Lambda ... 413
Problembehandlung bei S3 Object Lambda ... 452
Objektoperationen in großen Mengen durchführen ... 453
Grundlagen von BatchVorgänge ... 454
Tutorial zu S3-Batchvorgängen ... 456
Gewähren von Berechtigungen ... 456
Erstellen eines-Auftrags ... 467
Unterstützte Vorgänge ... 491
Verwalten von Aufträgen ... 537
Verfolgen von Auftragsstatus- und Abschluss ... 543
Verwenden von Markierungen ... 559
Objektsperre mit Batch-Operationen verwalten ... 576
Tutorial: Batch-Transcodierung von Videos ... 601
Problembehandlung bei Batch-Vorgängen ... 644
Daten an Ort und Stelle abfragen ... 647
Voraussetzungen und Einschränkungen ... 647
Erstellen einer Anforderung ... 648
Fehler ... 650
S3-Select-Beispiele ... 650
SQLReferenz ... 654
Arbeiten mit Verzeichnis-Buckets ... 697
Namen von Verzeichnis-Buckets ... 698
Verzeichnisse ... 698

Schlüsselnamen ... 699
Zugriffsverwaltung ... 699
Verzeichnis-Buckets erstellen und verwenden ... 699
Anwendungsfälle für Directory-Buckets ... 700
Leistungsstarke Workloads ... 700
Unterschiede bei Directory-Buckets ... 745
Unterschiede bei Directory-Buckets ... 745
APIOperationen, die für Directory-Buckets unterstützt werden ... 749
Amazon S3 S3-Funktionen, die von Directory-Buckets nicht unterstützt werden ... 750
Netzwerke für Verzeichnis-Buckets ... 751
Endpunkte ... 752
Konfiguration von Gateway-Endpunkten VPC ... 752
Regeln für die Benennung von Verzeichnis-Buckets ... 753
Anzeigen von Eigenschaften ... 754
Verwalten von Bucket-Richtlinien ... 755
Hinzufügen einer Bucket-Richtlinie ... 755
Anzeigen einer Bucket-Richtlinie ... 759
Löschen einer Bucket-Richtlinie ... 759
Leeren eines Verzeichnis-Buckets ... 760
Löschen eines Verzeichnis-Buckets ... 761
Auflisten von Verzeichnis-Buckets ... 764
Feststellen, ob Sie auf einen Bucket zugreifen können ... 767
Arbeiten mit Objekten in einem Verzeichnis-Bucket ... 768
Importieren von Objekten in einen Verzeichnis-Bucket ... 768
Arbeiten mit S3 Lifecycle ... 770
Verwenden von Batch Operations mit S3 Express One Zone ... 783
Daten an Objekte anhängen ... 787
Hochladen eines Objekts ... 789
Kopieren eines Objekts ... 823
Löschen eines Objekts ... 829
Herunterladen eines Objekts ... 833
Objektmetadaten werden abgerufen ... 835
Sicherheit für Verzeichnis-Buckets ... 836
Datenschutz und Verschlüsselung ... 837
Anfragen authentifizieren und autorisieren ... 862
Bewährte Methoden für die Sicherheit ... 886

Protokollierung mit AWS CloudTrail für Verzeichnis-Buckets ... 890
CloudTrail Verwaltungsereignisse für Directory-Buckets ... 891
CloudTrail Datenereignisse für Verzeichnis-Buckets ... 892
... 893
Optimierung der Leistung von Directory-Buckets ... 900
Leistungsrichtlinien und Entwurfsmuster ... 901
Entwickeln mit Verzeichnis-Buckets ... 902
Regionale und zonale Endpunkte für Directory-Buckets ... 903
Arbeiten mit Verzeichnis-Buckets mithilfe der S3-Konsole AWS CLI, und AWS SDKs ... 903
APIVerzeichnis-Bucket-Operationen ... 905
Zugriffskontrolle ... 907
S3-Ressourcen ... 908
Identitäten ... 913
Besitzer von Buckets oder Ressourcen ... 916
Tools zur Zugriffsverwaltung ... 916
Aktionen ... 923
Anwendungsfälle für die Zugriffsverwaltung ... 924
Fehlerbehebung bei der Zugriffsverwaltung ... 932
Identity and Access Management (IAM) ... 933
Zielgruppe ... 934
Authentifizierung mit Identitäten ... 935
Verwalten des Zugriffs mit Richtlinien ... 939
So funktioniert Amazon S3 mit IAM ... 942
Autorisierung beantragen ... 966
Erforderliche Berechtigungen für API S3-Operationen ... 976
Richtlinien und Berechtigungen ... 1013
Bucket-Richtlinien ... 1017
Identitätsbasierte Richtlinien ... 1066
Exemplarische Vorgehensweisen mithilfe von Richtlinien ... 1104
Verwenden von serviceverknüpften Rollen für Amazon S3 Storage Lens ... 1148
Problembehandlung Amazon S3 S3-Identität und -Zugriff ... 1152
AWS verwaltete Richtlinien ... 1178
Arbeiten mit Zugriffspunkten ... 1180
Beschränkungen und Einschränkungen ... 1181
IAMRichtlinien konfigurieren ... 1183
Erstellen von Zugriffspunkten ... 1191

Verwenden von Zugriffspunkten ... 1199
Verwalten des Zugriffs mit S3-Zugriffsberechtigungen ... 1211
Konzepte von S3 Access Grants ... 1213
S3 Access Grants und Unternehmensverzeichnisidentitäten ... 1218
Erste Schritte mit S3 Access Grants ... 1228
Arbeiten mit S3 Access Grants-Instanzen ... 1230
Mit S3 Access Grants-Standorten arbeiten ... 1243
Arbeiten mit Zuschüssen in S3 Access Grants ... 1266
Abrufen von S3-Daten mithilfe von Zugriffsberechtigungen ... 1281
Kontoübergreifender Zugriff mit S3 Access Grants ... 1294
Verwenden von AWS Tags mit S3 Access Grants ... 1307
Limits für S3 Access Grants ... 1310
S3-Access-Grants-Integrationen ... 1312
Verwaltung des Zugriffs mit ACLs ... 1313
ACLÜberblick ... 1314
Konfiguration ACLs ... 1336
Beispiele für Richtlinien ... 1354
Blockieren des öffentlichen Zugriffs ... 1359
Block Public Access-Einstellungen ... 1361
Durchführen von Block Public Access-Vorgänge an einem Zugriffspunkt ... 1365
Die Bedeutung von „öffentlich“ ... 1366
Verwenden von IAM Access Analyzer für S3 zur Überprüfung öffentlicher Buckets ... 1369
Berechtigungen ... 1370
Verwenden von Block Public Access ... 1371
Konfigurieren von Kontoeinstellungen ... 1371
Konfigurieren von Bucket- und Zugriffspunktseinstellungen ... 1374
Überprüfen des Bucket-Zugriffs ... 1378
Welche Informationen bietet IAM Access Analyzer for S3? ... 1380
IAMAccess Analyzer für S3 aktivieren ... 1381
Blockieren des gesamten öffentlichen Zugriffs ... 1381
Überprüfen und Ändern des Bucket-Zugriffs ... 1382
Archivieren von Bucket-Ergebnissen ... 1384
Aktivieren eines archivierten Bucket-Ergebnisses ... 1384
Anzeigen von Ergebnisdetails ... 1385
Ein IAM Access Analyzer for S3-Bericht wird heruntergeladen ... 1385
Überprüfen der Bucket-Eigentümerschaft ... 1386

Wann die Bucket-Eigentümer-Bedingung verwendet werden sollte ... 1386
Verifizieren eines Bucket-Eigentümers ... 1387
Beispiele ... 1388
Beschränkungen und Einschränkungen ... 1391
Steuern der Objekteigentümerschaft ... 1391
Einstellungen für Object Ownership ... 1393
Änderungen, die durch Deaktivierung eingeführt wurden ACLs ... 1396
Voraussetzungen für die Deaktivierung ACLs ... 1398
Berechtigungen für Object Ownership ... 1401
Deaktivierung ACLs für alle neuen Buckets ... 1402
Replikation und Object Ownership ... 1402
Einstellung von Object Ownership ... 1402
Voraussetzungen für die Deaktivierung ACLs ... 1403
Erstellen eines Buckets ... 1416
Einstellung von Object Ownership ... 1426
Anzeigen von Object Ownership-Einstellungen ... 1429
Deaktivierung ACLs für alle neuen Buckets ... 1431
Fehlerbehebung ... 1434
Sicherheit ... 1438
Bewährte Methoden für die Gewährleistung der Sicherheit ... 1440
Bewährte Methoden für die Sicherheit in Amazon S3 ... 1440
Bewährte Methoden zur Überwachung und Prüfung von Amazon S3 ... 1447
Überwachung der Datensicherheit ... 1451
Datenschutz ... 1454
Datenverschlüsselung ... 1456
Server-side encryption ... 1458
Verwenden der clientseitigen Verschlüsselung ... 1562
Richtlinie für den Datenverkehr zwischen Netzwerken ... 1563
Datenverkehr zwischen Service und lokalen Clients und Anwendungen ... 1563
Verkehr zwischen AWS Ressourcen in derselben Region ... 1564
AWS PrivateLink für Amazon S3 ... 1564
Compliance-Validierung ... 1583
Ausfallsicherheit ... 1585
Verschlüsselung von Sicherungen ... 1587
Sicherheit der Infrastruktur ... 1588
Konfigurations- und Schwachstellenanalyse ... 1589

Zugriffsverwaltung ... 1589
Datenschutz ... 1593
Objekte innerhalb und zwischen Regionen replizieren ... 1595
Warum sollten Sie Replikation verwenden? ... 1596
Verwenden der regionsübergreifenden Replikation ... 1598
Verwenden von Replikation innerhalb derselben Region ... 1598
Wann sollte die bidirektionale Replikation verwendet werden ... 1599
Wann die S3-Batch-Replikation verwendet wird ... 1599
Workload-Anforderungen und Live-Replikation ... 1600
Was wird repliziert? ... 1601
Anforderungen und Überlegungen für die Replikation ... 1605
Einrichtung der Live-Replikation ... 1609
Live-Replikation verwalten oder pausieren ... 1703
Replizieren vorhandener Objekte ... 1705
Fehlerbehebung bei einer Replikation ... 1721
Den Fortschritt überwachen und den Status abrufen ... 1730
Verwaltung des Datenverkehrs in mehreren Regionen ... 1755
Erstellen Multi-Regions-Zugriffspunkten ... 1757
Konfigurieren von Multi-Region Access Points ... 1767
Verwenden von Multi-Region Access Points ... 1772
Beibehaltung mehrerer Versionen von Objekten ... 1827
Nicht versionierte, versionings-fähige und Buckets mit ausgesetztem Versioning ... 1827
Verwenden des S3-Versioning mit dem S3-Lebenszyklus ... 1828
S3-Versioning ... 1829
Aktivieren des Versioning für Buckets ... 1834
MFALöschen konfigurieren ... 1841
Arbeiten mit versioning-fähigen Objekten ... 1844
Arbeiten mit Objekten mit ausgesetztem Versioning ... 1875
Problembehandlung bei der Versionierung ... 1879
Sperren von Objekten ... 1884
So funktioniert die S3-Objektsperre ... 1886
Überlegungen zu Object Lock ... 1890
Konfigurieren der Objektsperre ... 1896
Sichern von Daten ... 1907
Kostenoptimierung ... 1909
Fakturierungs- und Nutzungsberichte ... 1910

Verwenden von Kostenzuordnungs-Markierungen ... 1911
Fakturierungsberichte ... 1913
Nutzungsberichte ... 1916
Fakturierungs- und Nutzungsberichte verstehen ... 1919
Abrechnung für Amazon S3 S3-Fehlerantworten ... 1948
Speicherklassen verstehen und verwalten ... 1973
Häufig aufgerufene Objekte ... 1973
Automatische Optimierung von Daten mit sich ändernden oder unbekannten Zugriffsmustern ... 1974
Selten aufgerufene Objekte ... 1976
Objekte, auf die selten zugegriffen wird ... 1978
Amazon S3 in Outposts ... 1979
Vergleich der Speicherklassen ... 1980
Einrichten der Speicherklasse eines Objekts ... 1981
Speicherklassen-Analyse ... 1986
Verwaltung der Speicherkosten mit Amazon S3 Intelligent-Tiering ... 1994
Amazon S3 Glacier-Speicherklassen ... 2007
Arbeiten mit archivierten Objekten ... 2014
Verwalten des Lebenszyklus ... 2028
Verwaltung des gesamten Lebenszyklus von Objekten ... 2030
Übergang von Objekten ... 2031
Auslaufende Objekte ... 2041
Einrichten der Lebenszyklus-Konfiguration ... 2044
Verwenden anderer Bucket-Konfigurationen ... 2064
Konfiguration von S3 Lifecycle-Ereignisbenachrichtigungen ... 2067
Elemente der Lebenszyklus-Konfiguration ... 2069
Konflikte bei der Lebenszykluskonfiguration ... 2088
Beispiele für S3-Lifecycle-Konfigurationen ... 2093
Behebung von Problemen im Lebenszyklus ... 2108
Protokollierung und Überwachung ... 2116
Überwachungstools ... 2119
Automatisierte Tools ... 2119
Manuelle Tools ... 2119
Protokollierungsoptionen ... 2120
Protokollierung mit CloudTrail ... 2123

Verwenden von CloudTrail Protokollen mit Amazon S3 S3-Serverzugriffsprotokollen und CloudWatch Protokollen ... 2125
CloudTrail Nachverfolgung mit Amazon S3 SOAP API S3-Anrufen ... 2126
CloudTrail Ereignisse ... 2127
Beispiele für Protokolldateien ... 2140
Aktivieren CloudTrail ... 2144
Identifizieren von S3-Anfragen ... 2148
Protokollierungs-Serverzugriff ... 2155
Wie aktiviere ich die Protokollzustellung? ... 2155
Protokollobjekt-Schlüsselformat ... 2158
Wie werden Protokolle ausgeliefert? ... 2159
Best-Effort-Serverprotokollbereitstellung ... 2160
Statusänderungen in der Bucket-Protokollierung werden mit der Zeit wirksam ... 2161
Aktivieren der Server-Zugriffsprotokollierung ... 2161
Protokollformat ... 2184
Löschen von Protokolldateien ... 2199
Identifizieren von S3-Anfragen ... 2200
Behebung von Fehlern bei der Server-Zugriffsprotokollierung ... 2206
Überwachung von Metriken mit CloudWatch ... 2209
Metriken und Dimensionen ... 2212
Zugriff auf Metriken CloudWatch ... 2232
CloudWatch Metriken, Konfigurationen ... 2233
Amazon-S3-Ereignis-Benachrichtigungen ... 2243
Übersicht ... 2243
Benachrichtigungstypen und -ziele ... 2245
Verwenden von SQSSNS, und Lambda ... 2253
Benutzen EventBridge ... 2284
Visualisieren Sie Ihre Speicheraktivität und -nutzung ... 2295
Metriken und Funktionen von S3 Storage Lens ... 2296
Grundlegendes zu S3 Storage Lens ... 2298
Metrikglossar ... 2310
Festlegung von -Berechtigungen ... 2346
Arbeiten mit S3 Storage Lens ... 2350
Anzeigen von Speichermetriken ... 2391
Arbeiten mit Organisationen ... 2454
Arbeiten mit Storage Lens-Gruppen ... 2465

Katalogisierung und Analyse Ihrer Daten ... 2512
Amazon-S3-Inventory-Buckets ... 2513
Bestandslisten ... 2513
Konfigurieren von Amazon S3 Inventory ... 2519
Lokalisieren Ihres Bestands ... 2530
Einrichten von Benachrichtigungen für den Bestandsabschluss ... 2534
Bestandsabfrage mit Athena ... 2535
Konvertieren leerer Versions-ID-Zeichenfolgen in Null-Zeichenfolgen ... 2541
Mit dem ACL Objektfeld arbeiten ... 2544
Optimierung der Leistung ... 2547
Leistungsrichtlinien für Amazon S3 ... 2548
Messen der Leistung ... 2549
Horizontal skalieren ... 2550
Verwenden Sie Abrufe im Bytebereich ... 2550
Anfragen wiederholen ... 2550
Kombinieren Sie Amazon S3 und Amazon EC2 in derselben Region ... 2551
Verwenden Sie Transfer Acceleration, um die Latenz zu minimieren ... 2551
Verwenden Sie die neueste AWS SDKs ... 2551
Leistungsentwurfsmuster für Amazon S3 ... 2552
Zwischenspeichern von Inhalten, auf die häufig zugegriffen wird ... 2552
Timeouts und Wiederholungsversuche für latenzempfindliche Apps ... 2553
Horizontale Skalierung und Parallelisierung von Anfragen ... 2554
Beschleunigung geografisch verteilter Datenübertragungen ... 2556
Hosten einer statischen Website ... 2557
Website-Endpunkte ... 2558
Website-Endpunkt-Beispiele ... 2559
Hinzufügen eines DNS CNAME ... 2560
Verwenden einer benutzerdefinierten Domäne mit Route 53 ... 2560
Hauptunterschiede zwischen einem Website-Endpunkt und einem REST API Endpunkt ... 2561
Aktivieren des Website-Hostings ... 2562
Konfigurieren eines Indexdokuments ... 2567
Indexdokument und Ordner ... 2568
Konfigurieren eines Indexdokuments ... 2569
Konfigurieren eines benutzerdefinierten Fehlerdokuments ... 2571
Amazon S3 HTTP S3-Antwortcodes ... 2571
Konfigurieren eines benutzerdefinierten Fehlerdokuments ... 2574

Festlegen von Berechtigungen für den Website-Zugriff ... 2575
Schritt 1: Bearbeiten der S3 Block Public Access-Einstellungen ... 2576
Schritt 2: Hinzufügen einer Bucket-Richtlinie ... 2578
Objektzugriffskontrolllisten ... 2580
Protokollieren des Webdatenverkehrs ... 2581
Konfigurieren einer Umleitung ... 2582
Anforderungen an einen anderen Host umleiten. ... 2583
Konfigurieren von Umleitungsregeln ... 2584
So leiten Sie Anforderungen für ein Objekt um ... 2592
Verwenden CORS ... 2594
Cross-Origin Resource Sharing: Szenarien in Anwendungsfällen ... 2594
Wie bewertet Amazon S3 die CORS Konfiguration eines Buckets? ... 2595
So unterstützt Object Lambda Access Point CORS ... 2596
Elemente einer CORS Konfiguration ... 2596
Konfiguration CORS ... 2602
Testen CORS ... 2611
Fehlerbehebung für CORS ... 2613
Tutorials für statische Websites ... 2618
Hosting von Video-Streaming ... 2620
Konfigurieren einer statischen Website ... 2641
Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten Domäne ... 2651
Bereitstellung einer statischen Website für Amplify von Amazon S3 ... 2678
Kontingente ... 2682
Die Quote wird erhöht ... 2682
Referenz ... 2683
Dokumentverlauf ... 2684
Frühere Updates ... 2728
... mmdcclvii

Was ist Amazon S3?

Amazon Simple Storage Service (Amazon S3) ist ein Objektspeicherservice, der branchenführende Skalierbarkeit, Datenverfügbarkeit, Sicherheit und Leistung bietet. Kunden aller Größen und Branchen können Amazon S3 für die Speicherung und den Schutz beliebiger Datenmengen für eine Reihe von Anwendungsfällen verwenden, wie Data Lakes, Websites, mobile Anwendungen, Backup und Wiederherstellung, Archivierung, Unternehmensanwendungen, IoT-Geräte und Big-Data-Analysen. Amazon S3 bietet Verwaltungsfunktionen, mit denen Sie den Zugriff auf Ihre Daten optimieren, organisieren und konfigurieren können, um Ihre spezifischen geschäftlichen, organisatorischen und Compliance-Anforderungen zu erfüllen.

> **Note**
>
> Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter S3 Express One Zone und Arbeiten mit Verzeichnis-Buckets.

Themen

- Funktionen von Amazon S3
- Funktionsweise von Amazon S3
- Amazon S3-Datenkonsistenzmodell
- Zugehörige Services
- Zugriff auf Amazon S3
- Bezahlen für Amazon S3
- PCIDSSEinhaltung der Vorschriften

Funktionen von Amazon S3

Speicherklassen

Amazon S3 bietet eine Vielzahl von Speicherklassen an, die für diverse Anwendungsfälle genutzt werden können. So können Sie beispielsweise missionskritische Produktionsdaten in S3 Standard für häufigen Zugriff speichern, Kosten sparen, indem Sie selten aufgerufene Daten in S3 Standard-

IA speichern und Daten zu den niedrigsten Kosten in S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive archivieren.

Amazon S3 Express One Zone ist eine leistungsstarke Amazon-S3-Speicherklasse mit einer einzelnen Zone, die speziell für den konsistenten Datenzugriff im einstelligen Millisekundenbereich für Ihre latenzempfindlichsten Anwendungen entwickelt wurde. S3 Express One Zone ist die derzeit verfügbare Cloud-Objektspeicherklasse mit der niedrigsten Latenz. Sie bietet Datenzugriffsgeschwindigkeiten, die bis zu zehnmal schneller sind und die Anforderungskosten um 50 Prozent niedriger sind als bei S3 Standard. S3 Express One Zone ist die erste S3-Speicherklasse, bei der Sie eine einzelne Availability Zone mit der Option auswählen können, Ihren Objektspeicher gemeinsam mit Ihren Computingressourcen zu platzieren, was die höchstmögliche Zugriffsgeschwindigkeit bietet. Um die Zugriffsgeschwindigkeit weiter zu erhöhen und Hunderttausende von Anfragen pro Sekunde zu unterstützen, werden Daten außerdem in einem neuen Bucket-Typ gespeichert: einem Amazon-S3-Verzeichnis-Bucket. Weitere Informationen erhalten Sie unter S3 Express One Zone und Arbeiten mit Verzeichnis-Buckets.

Sie können Daten mit sich ändernden oder unbekannten Zugriffsmustern in S3 Intelligent-Tiering speichern, das die Speicherkosten optimiert, indem es Ihre Daten automatisch zwischen vier Zugriffsebenen verschiebt, wenn sich Ihre Zugriffsmuster ändern. Diese vier Zugriffsebenen umfassen zwei Zugriffsebenen mit geringer Latenz, die für häufigen und seltenen Zugriff optimiert sind, und zwei Opt-in-Archivzugriffsebenen, die für den asynchronen Zugriff auf selten abgerufene Daten ausgelegt sind.

Weitere Informationen finden Sie unter Amazon S3 S3-Speicherklassen verstehen und verwalten.

Speicherverwaltung

Amazon S3 hat Speicherverwaltungsfunktionen, die Sie benutzen können, um Kosten zu verwalten, gesetzliche Anforderungen zu erfüllen, Latenz zu reduzieren und mehrere verschiedene Kopien Ihrer Daten für Compliance-Anforderungen zu speichern.

- S3-Lebenszyklus – Richten Sie eine Lebenszykluskonfiguration ein, um Ihre Objekte zu verwalten und während ihres gesamten Lebenszyklus kostengünstig zu speichern. Sie können Objekte in andere S3-Speicherklassen umstellen oder Objekte ablaufen, die das Ende ihrer Lebensdauer erreichen.
- S3-Objektsperre – Verhindern Sie, dass Amazon S3-Objekte für einen bestimmten Zeitraum oder auf unbestimmte Zeit gelöscht oder überschrieben werden. Sie können Object Lock verwenden, um gesetzliche Anforderungen zu erfüllen, die Speicherplatz write-once-read-many(WORM)

erfordern, oder um einfach eine weitere Schutzebene gegen Objektänderungen und -löschungen hinzuzufügen.

- S3-Replikation — Replizieren Sie Objekte und ihre jeweiligen Metadaten und Objekt-Tags in einen oder mehrere Ziel-Buckets im gleichen oder unterschiedlichen Ziel-Buckets, AWS-Regionen um Latenz, Compliance, Sicherheit und andere Anwendungsfälle zu reduzieren.
- S3 Batch Operations — Verwalten Sie Milliarden von Objekten in großem Umfang mit einer einzigen API S3-Anfrage oder ein paar Klicks in der Amazon S3 S3-Konsole. Mithilfe von Batch-Operationen können Sie Operationen wie Kopieren, AWS Lambda-Funktion aufrufen und Wiederherstellen für Millionen oder Milliarden von Objekten ausführen.

Zugriffsverwaltung und Sicherheit

Amazon S3 bietet Funktionen für die Überwachung und Verwaltung des Zugriffs auf Ihre Buckets und Objekte. Standardmäßig werden S3-Buckets und -Objekte als privat eingestuft. Sie haben nur Zugriff auf die S3-Ressourcen, die Sie erstellen. Um detaillierte Ressourcenberechtigungen zu erteilen, die Ihren speziellen Anwendungsfall unterstützen, oder um die Berechtigungen Ihrer Amazon S3-Ressourcen zu überprüfen, können Sie die folgenden Funktionen verwenden.

- S3 öffentlichen Zugriff blockieren – Blockieren Sie den öffentlichen Zugriff auf S3-Buckets und -Objekte. Standardmäßig sind die Einstellungen für das Blockieren des öffentlichen Zugriffs auf Bucket-Ebene aktiviert. Es wird empfohlen, alle Einstellungen für das Blockieren des öffentlichen Zugriffs aktiviert zu lassen, es sei denn, Sie wissen, dass Sie eine oder mehrere dieser Einstellungen für Ihren Anwendungsfall deaktivieren müssen. Weitere Informationen finden Sie unter Konfigurieren von Block-Public-Access-Einstellungen für Ihre S3-Buckets.
- AWS Identity and Access Management (IAM) — IAM ist ein Webservice, mit dem Sie den Zugriff auf AWS Ressourcen, einschließlich Ihrer Amazon S3 S3-Ressourcen, sicher kontrollieren können. Mit können Sie Berechtigungen zentral verwaltenIAM, mit denen gesteuert wird, auf welche AWS Ressourcen Benutzer zugreifen können. Sie steuern IAM damit, wer authentifiziert (angemeldet) und autorisiert ist (über Berechtigungen verfügt), um Ressourcen zu verwenden.
- Bucket-Richtlinien — Verwenden Sie eine IAM basierte Richtliniensprache, um ressourcenbasierte Berechtigungen für Ihre S3-Buckets und die darin enthaltenen Objekte zu konfigurieren.
- Amazon-S3-Zugriffspunkte – Konfigurieren Sie benannte Netzwerkendpunkte mit dedizierten Zugriffsrichtlinien, um den Datenzugriff auf freigegebene Datensätze in Amazon S3 skalierbar zu verwalten.

- Zugriffskontrolllisten (ACLs) — Erteilen Sie autorisierten Benutzern Lese- und Schreibberechtigungen für einzelne Buckets und Objekte. In der Regel empfehlen wir, stattdessen ressourcenbasierte S3-Richtlinien (Bucket-Richtlinien und Zugriffspunktrichtlinien) oder IAM Benutzerrichtlinien für die Zugriffskontrolle zu verwenden. ACLs Richtlinien stellen eine vereinfachte und flexiblere Zugriffskontrolloption dar. Mit Bucket- und Zugriffspunktrichtlinien können Sie Regeln definieren, die allgemein für alle Anfragen an Ihre Amazon-S3-Ressourcen gelten. Weitere Informationen zu den spezifischen Fällen, in denen Sie ACLs anstelle von ressourcenbasierten Richtlinien oder IAM Benutzerrichtlinien verwenden würden, finden Sie unter. Verwaltung des Zugriffs mit ACLs
- S3 Object Ownership – Übernehmen Sie die Verantwortung für jedes Objekt in Ihrem Bucket, wodurch die Zugriffsverwaltung für in Amazon S3 gespeicherte Daten vereinfacht wird. S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, die Sie zum Deaktivieren oder Aktivieren verwenden können. ACLs Standardmäßig ACLs sind sie deaktiviert. Wenn diese ACLs Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff auf Daten ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.
- IAMAccess Analyzer for S3 — Evaluieren und überwachen Sie Ihre S3-Bucket-Zugriffsrichtlinien und stellen Sie sicher, dass die Richtlinien nur den beabsichtigten Zugriff auf Ihre S3-Ressourcen ermöglichen.

Datenverarbeitung

Um Daten zu transformieren und Workflows auszulösen, um eine Vielzahl anderer Verarbeitungsaktivitäten maßstabsgerecht zu automatisieren, können Sie die folgenden Funktionen verwenden.

- S3 Object Lambda — Fügen Sie Ihren eigenen Code zu S3 hinzu und HEAD LIST fordern Sie anGET, Daten zu ändern und zu verarbeiten, wenn sie an eine Anwendung zurückgegeben werden. Filtern Sie Zeilen, ändern Sie die Größe von Bildern dynamisch, verkleinern Sie vertrauliche Daten und vieles mehr.
- Ereignisbenachrichtigungen — Löst Workflows aus, die Amazon Simple Notification Service (AmazonSNS), Amazon Simple Queue Service (AmazonSQS) verwenden, und AWS Lambda wenn eine Änderung an Ihren S3-Ressourcen vorgenommen wird.

Speicherprotokollierung und Überwachung

Amazon S3 bietet Protokollierungs- und Überwachungstools, mit denen Sie überwachen und steuern können, wie Ihre Amazon S3-Ressourcen verwendet werden. Weitere Informationen finden Sie unter Überwachungstools.

Automatisierte Überwachungstools

- CloudWatchAmazon-Metriken für Amazon S3 — Verfolgen Sie den Betriebsstatus Ihrer S3-Ressourcen und konfigurieren Sie Abrechnungswarnungen, wenn geschätzte Gebühren einen benutzerdefinierten Schwellenwert erreichen.
- AWS CloudTrail— Zeichnet Aktionen auf, die von einem Benutzer, einer Rolle oder einem AWS-Service in Amazon S3 ausgeführt wurden. CloudTrail Protokolle bieten Ihnen eine detaillierte API Nachverfolgung von S3-Vorgängen auf Bucket- und Objektebene.

Manuelle Überwachungstools

- Server-Zugriffsprotokollierung – Abrufen detaillierter Datensätze für die Anforderungen, die an einen Bucket gestellt werden. Sie können Serverzugriffsprotokolle für viele Anwendungsfälle verwenden, z. B. für die Durchführung von Sicherheits- und Zugriffsprüfungen, das Kennenlernen Ihres Kundenstamms und das Verstehen Ihrer Amazon S3-Rechnung.
- AWS Trusted Advisor — Evaluieren Sie Ihr Konto anhand von AWS Best-Practice-Checks, um Möglichkeiten zur Optimierung Ihrer AWS Infrastruktur, Verbesserung von Sicherheit und Leistung, Kostensenkung und Überwachung der Servicekontingenten zu ermitteln. Sie können dann die Empfehlungen befolgen, um Ihre Dienste und Ressourcen zu optimieren.

Analytik und Einblicke

Amazon S3 bietet Funktionen, die Ihnen dabei helfen, Einblicke in Ihre Speicherauslastung zu erhalten, sodass Sie Ihren Speicher in großem Umfang besser verstehen, analysieren und optimieren können.

- Amazon S3 Storage Lens – Verstehen, analysieren und optimieren Sie Ihren Speicher. S3 Storage Lens bietet mehr als 60 Nutzungs- und Aktivitätsmetriken sowie interaktive Dashboards, um Daten für Ihr gesamtes Unternehmen, bestimmte Konten AWS-Regionen, Buckets oder Präfixe zu aggregieren.

- Speicherklassenanalys – Analysieren Sie Speicherzugriffsmuster, um zu entscheiden, wann es an der Zeit ist, Daten in eine kostengünstigere Speicherklasse zu verschieben.
- S3-Inventory mit Bestandsberichten – Überwachen und erstellen Sie Berichte zu Objekten und den entsprechenden Metadaten und konfigurieren Sie andere Amazon S3-Funktionen, um in Inventarberichten Maßnahmen zu ergreifen. Sie können beispielsweise einen Bericht über den Replikations- und Verschlüsselungsstatus Ihrer Objekte erstellen. Eine Liste aller verfügbaren Metadaten, die für jedes Objekt in Inventorberichten verfügbar sind, finden Sie unter Amazon S3-Bestandsliste.

Starke Konsistenz

Amazon S3 bietet insgesamt eine hohe read-after-write Konsistenz für Objekte PUT und DELETE Anfragen von Objekten in Ihrem Amazon S3 S3-Bucket AWS-Regionen. Dieses Verhalten gilt sowohl für Schreibvorgänge neuer Objekte als auch für PUT Anfragen, die bestehende Objekte und DELETE Anfragen überschreiben. Darüber hinaus sind die Lesevorgänge auf Amazon S3 Select, Amazon S3 S3-Zugriffskontrolllisten (ACLs), Amazon S3 S3-Objekt-Tags und Objektmetadaten (z. B. dem HEAD Objekt) stark konsistent. Weitere Informationen finden Sie unter Amazon S3-Datenkonsistenzmodell.

Funktionsweise von Amazon S3

Amazon S3 ist ein Objektspeicherdienst, der Daten als Objekte in Buckets speichert. Ein Objekt ist eine Datei und alle Metadaten, die diese Datei beschreiben. Ein Bucket ist ein Container für Objekte.

Um Ihre Daten in Amazon S3 zu speichern, erstellen Sie zunächst einen Bucket und geben einen Bucket-Namen und AWS-Region an. Anschließend laden Sie Ihre Daten in diesen Bucket als Objekte in Amazon S3 hoch. Jedes Objekt hat einen Schlüssel (oder Schlüsselnamen), der der eindeutige Bezeichner für das Objekt im Bucket ist.

S3 bietet Funktionen, die Sie konfigurieren können, um Ihren speziellen Anwendungsfall zu unterstützen. Beispielsweise können Sie das S3-Versioning verwenden, um mehrere Versionen eines Objekts im selben Bucket zu halten, wodurch Sie versehentlich gelöschte oder überschriebene Objekte wiederherstellen können.

Buckets und die darin enthaltenen Objekte sind privat und können nur zugegriffen werden, wenn Sie explizit Zugriffsberechtigungen erteilen. Sie können Bucket-Richtlinien, AWS Identity and Access Management (IAM) -Richtlinien, Zugriffskontrolllisten (ACLs) und S3-Zugangspunkte verwenden, um den Zugriff zu verwalten.

Themen

- Buckets
- Objekte
- Schlüssel
- S3-Versioning
- Versions-ID
- Bucket-Richtlinie
- S3-Zugriffspunkte
- Zugriffskontrolllisten (ACLs)
- Regionen

Buckets

Ein Allzweck-Bucket ist ein Container für Objekte, die in Amazon S3 gespeichert sind. Sie können eine beliebige Anzahl von Objekten in einem Bucket speichern und alle Konten haben ein Standard-Bucket-Kontingent von 10.000 Allzweck-Buckets. Um Ihre Bucket-Auslastung oder Ihr Bucket-Kontingent einzusehen oder eine Erhöhung dieses Kontingents zu beantragen, besuchen Sie die Service Quotas Quotas-Konsole.

Jedes Objekt ist in einem Bucket enthalten. Wenn das benannte Objekt beispielsweise im `amzn-s3-demo-bucket` Bucket in der Region USA West (Oregon) gespeichert `photos/puppy.jpg` ist, kann es mithilfe von adressiert werden. URL `https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/photos/puppy.jpg` Weitere Informationen siehe Zugriff auf einen Bucket.

Wenn Sie einen Bucket erstellen, geben Sie einen Bucket-Namen ein und wählen die Option AWS-Region , wo sich der Bucket befindet. Nachdem Sie einen Bucket erstellt haben, können Sie den Namen des Buckets oder seiner Region nicht mehr ändern. Bucket-Namen müssen den Regeln für die Benennung von Buckets folgen. Sie können einen Bucket auch für die Verwendung des S3-Versioning oder anderer Speicherverwaltungs-Funktionen konfigurieren.

Zudem tun Buckets Folgendes:

- Sie strukturieren den Amazon S3-Namespace auf der höchsten Ebene.
- Sie identifizieren das Konto, dem die Gebühren für Datenspeicherung und -übertragung belastet werden.

- Stellen Sie Zugriffskontrolloptionen wie Bucket-Richtlinien, Zugriffskontrolllisten (ACLs) und S3-Zugriffspunkte bereit, mit denen Sie den Zugriff auf Ihre Amazon S3 S3-Ressourcen verwalten können.
- Sie dienen im Rahmen der Erstellung von Nutzungsberichten als Auswertungseinheit.

Weitere Informationen über Buckets finden Sie unter Bucket-Übersicht.

Objekte

Objekte sind die Grundeinheiten, die in Amazon S3 gespeichert werden. Objekte bestehen aus Objekt- und Metadaten. Metadaten bestehen aus mehreren Name/Wert-Paaren, die das Objekt beschreiben. Diese Paare enthalten einige Standardmetadaten, wie das Datum der letzten Änderung, und HTTP Standardmetadaten wie`Content-Type`. Sie können bei der Speicherung des Objekts auch benutzerdefinierte Metadaten angeben.

Ein Objekt wird innerhalb eines Buckets eindeutig durch einen Schlüssel (Name) und eine Versions-ID identifiziert (wenn das S3-Versioning im Bucket aktiviert ist). Weitere Informationen über Objekte finden Sie unter Übersicht über Amazon-S3-Objekte.

Schlüssel

Ein Objektschlüssel (oder Schlüsselname) ist der eindeutige Bezeichner für ein Objekt in einem Bucket. Jedes Objekt in einem Bucket besitzt genau einen Schlüssel. Die Kombination aus Bucket, Objektschlüssel und optional Versions-ID (wenn das S3-Versioning für den Bucket aktiviert ist) identifiziert jedes Objekt eindeutig. Amazon S3 fungiert also als grundlegende Datenzuordnung zwischen „Bucket + Schlüssel + Version“ und dem Objekt selbst.

Jedes Objekt in Amazon S3 ist über eine Kombination von Webservice-Endpunkt, Bucket-Name, Schlüssel und wahlweise einer Version aufrufbar. In der *`amzn-s3-demo-bucket`* steht URL `https://`*`amzn-s3-demo-bucket`*`.s3.us-west-2.amazonaws.com/photos/puppy.jpg` beispielsweise der Name des Buckets und `photos/puppy.jpg` ist der Schlüssel.

Weitere Informationen über Objektschlüssel finden Sie unter Benennen Amazon S3 S3-Objekten.

S3-Versioning

Sie S3-Versioning verwenden, um mehrere Versionen eines Objekts im selben Bucket aufzubewahren. Mit S3-Versioning können Sie jede Version jedes in Ihren Buckets gespeicherten

Objekts beibehalten, abrufen und wiederherstellen. Sie können sowohl nach unbeabsichtigten Benutzeraktionen als auch nach Anwendungsfehlern problemlos wiederherstellen.

Weitere Informationen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

Versions-ID

Wenn Sie das S3-Versioning in einem Bucket aktivieren, generiert Amazon S3 eine eindeutige Versions-ID für jedes Objekt, das dem Bucket hinzugefügt wird. Objekte, die zum Zeitpunkt der Aktivierung des Versioning bereits im Bucket vorhanden waren, haben die Versions-ID`null` . Wenn Sie diese (oder andere) Objekte mit anderen Operationen wie CopyObjectund ändern PutObject, erhalten die neuen Objekte eine eindeutige Versions-ID.

Weitere Informationen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

Bucket-Richtlinie

Eine Bucket-Richtlinie ist eine ressourcenbasierte AWS Identity and Access Management (IAM) Richtlinie, mit der Sie Zugriffsberechtigungen für Ihren Bucket und die darin enthaltenen Objekte gewähren können. Nur der Bucket-Eigentümer kann einem Bucket eine Richtlinie zuordnen. Die dem Bucket zugeordneten Berechtigungen gelten für alle Objekte im Bucket, die dem Bucket-Eigentümer gehören. Bucket-Richtlinien sind auf eine Größe von 20 KB beschränkt.

Bucket-Richtlinien verwenden eine auf der JSON Grundlage von Zugriffsrichtlinien verwendete Sprache, die in allen Bereichen Standard ist. AWS Sie können Bucket-Richtlinien verwenden, um Berechtigungen für die Objekte in einem Bucket hinzuzufügen oder zu verweigern. Bucket-Richtlinien erlauben oder verweigern Anforderungen basierend auf den Elementen in der Richtlinie, einschließlich des Anforderers, S3-Aktionen, Ressourcen und Aspekten oder Bedingungen der Anforderung (z. B. die IP-Adresse, die für die Anforderung verwendet wird). Sie können beispielsweise eine Bucket-Richtlinie erstellen, die kontoübergreifende Berechtigungen zum Hochladen von Objekten in einen S3-Bucket gewährt, während gleichzeitig sichergestellt wird, dass der Bucket-Eigentümer die volle Kontrolle über die hochgeladenen Objekte hat. Weitere Informationen finden Sie unter Beispiele für Amazon S3 S3-Bucket-Richtlinien.

In Ihrer Bucket-Richtlinie können Sie Platzhalterzeichen für Amazon-Ressourcennamen (ARNs) und andere Werte verwenden, um einer Teilmenge von Objekten Berechtigungen zu erteilen. Sie

können beispielsweise den Zugriff auf Gruppen von Objekten steuern, die mit einem gemeinsamen Präfixbeginnen oder mit einer bestimmten Erweiterung wie `.html` enden.

S3-Zugriffspunkte

Amazon S3-Zugriffspunkte sind benannte Netzwerkendpunkte mit dedizierten Zugriffsrichtlinien, die beschreiben, wie mit diesem Endpunkt auf Daten zugegriffen werden kann. Access Points sind an Buckets angehängt, mit denen Sie S3-Objektoperationen wie und ausführen können. GetObject PutObject Zugriffspunkte vereinfachen die skalierbare Verwaltung des Datenzugriffs auf freigegebene Datensätze in Amazon S3.

Jeder Zugriffspunkt verfügt über eine eigene Zugriffspunktrichtlinie. Sie können Block Public Access-Einstellungen für jeden Zugriffspunkt konfigurieren. Um den Amazon S3 S3-Datenzugriff auf ein privates Netzwerk zu beschränken, können Sie auch jeden Access Point so konfigurieren, dass er nur Anfragen von einer virtuellen privaten Cloud akzeptiert (VPC).

Weitere Informationen finden Sie unter Verwaltung des Zugriffs auf gemeinsam genutzte Datensätze mit Access Points.

Zugriffskontrolllisten (ACLs)

Sie können ACLs es verwenden, um autorisierten Benutzern Lese- und Schreibberechtigungen für einzelne Buckets und Objekte zu gewähren. Jedem Bucket und Objekt ist eine Unterressource ACL angehängt. Das ACL definiert, welche AWS-Konten Gruppen Zugriff erhalten und welche Art von Zugriff gewährt wird. ACLssind ein Zugangskontrollmechanismus, der älter ist alsIAM. Weitere Informationen zu finden Sie ACLs unterÜbersicht über die Zugriffskontrollliste (ACL).

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie sowohl das Eigentum an den Objekten kontrollieren können, die in Ihren Bucket hochgeladen werden, als auch um sie zu deaktivieren oder zu aktivieren. ACLs Standardmäßig ist für Object Ownership die erzwungene Einstellung des Bucket-Besitzers festgelegt, und alle sind ACLs deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff darauf ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.

Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Wenn diese ACLs Option deaktiviert ist, können Sie mithilfe von Richtlinien den Zugriff auf alle Objekte in Ihrem Bucket steuern, unabhängig davon, wer die Objekte in Ihren Bucket hochgeladen hat. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

Regionen

Sie können das geografische Gebiet wählen AWS-Region , in dem Amazon S3 die von Ihnen erstellten Buckets speichert. Sie sollten eine Region im Hinblick auf Latenz, Kosten sowie Einhaltung der relevanten Vorschriften auswählen. Objekte, die in der Region gespeichert sind und diese AWS-Region nie verlassen, es sei denn, Sie übertragen oder replizieren sie ausdrücklich in eine andere Region. In der Region EU (Irland) gespeicherte Objekte verlassen diese Region nicht.

> **Note**
>
> Sie können auf Amazon S3 und seine Funktionen nur in den Bereichen zugreifen AWS-Regionen , die für Ihr Konto aktiviert sind. Weitere Informationen darüber, wie Sie einer Region die Erstellung und Verwaltung von AWS Ressourcen ermöglichen, finden Sie unter Verwaltung AWS-Regionen in der Allgemeine AWS-Referenz.

Eine Liste der Amazon-S3-Regionen und -Endpunkte finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Referenz.

Amazon S3-Datenkonsistenzmodell

Amazon S3 bietet insgesamt eine hohe read-after-write Konsistenz für Objekte PUT und DELETE Anfragen von Objekten in Ihrem Amazon S3 S3-Bucket AWS-Regionen. Dieses Verhalten gilt sowohl für Schreibvorgänge in neue Objekte als auch für PUT Anfragen, die bestehende Objekte und DELETE Anfragen überschreiben. Darüber hinaus sind die Lesevorgänge auf Amazon S3 Select, Amazon S3 S3-Zugriffskontrolllisten (ACLs), Amazon S3 S3-Objekt-Tags und Objektmetadaten (z. B. das HEAD Objekt) sehr konsistent.

Aktualisierungen an einem einzelnen Schlüssel sind unteilbar. Wenn Sie beispielsweise von einem Thread aus eine PUT Anfrage an einen vorhandenen Schlüssel stellen und gleichzeitig von einem zweiten Thread aus eine GET Anfrage für denselben Schlüssel ausführen, erhalten Sie entweder die alten oder die neuen Daten, jedoch niemals unvollständige oder beschädigte Daten.

Amazon S3 erzielt hohe Verfügbarkeit, indem die Daten innerhalb der AWS -Rechenzentren über mehrere Server repliziert werden. Wenn eine PUT Anfrage erfolgreich ist, werden Ihre Daten sicher gespeichert. Bei jedem GET LIST Lesevorgang (oder jeder Anfrage), der nach Erhalt einer erfolgreichen PUT Antwort eingeleitet wird, werden die in der PUT Anfrage geschriebenen Daten zurückgegeben. Hier finden Sie Beispiele für dieses Verhalten:

- Ein Prozess schreibt ein neues Objekt in Amazon S3 und listet sofort Schlüssel innerhalb seines Buckets auf. Das neue Objekt wird in der Liste angezeigt.
- Ein Prozess ersetzt ein vorhandenes Objekt und versucht sofort, es zu lesen. Amazon S3 gibt die neuen Daten zurück.
- Ein Prozess löscht ein vorhandenes Objekt und versucht sofort, es zu lesen. Amazon S3 gibt keine Daten zurück, da das Objekt gelöscht wurde.
- Ein Prozess löscht ein vorhandenes Objekt und listet sofort Schlüssel innerhalb seines Buckets auf. Das Objekt wird nicht im Verzeichnis angezeigt.

> **Note**
>
> - Amazon S3 unterstützt keine Objektsperre für gleichzeitige Schreibvorgänge. Wenn zwei PUT Anfragen gleichzeitig an denselben Schlüssel gestellt werden, gewinnt die Anfrage mit dem neuesten Zeitstempel. Wenn das ein Problem ist, müssen Sie in Ihre Anwendung einen Sperrmechanismus für Objekte einbauen.
> - Updates basieren auf Schlüsseln. Es gibt keine Möglichkeit für unteilbare Aktualisierungen über Schlüssel hinweg. Sie können beispielsweise die Aktualisierung eines Schlüssels nicht von der Aktualisierung eines anderen Schlüssels abhängig machen. Dazu müssten Sie diese Funktionalität in Ihrer Anwendung implementieren.

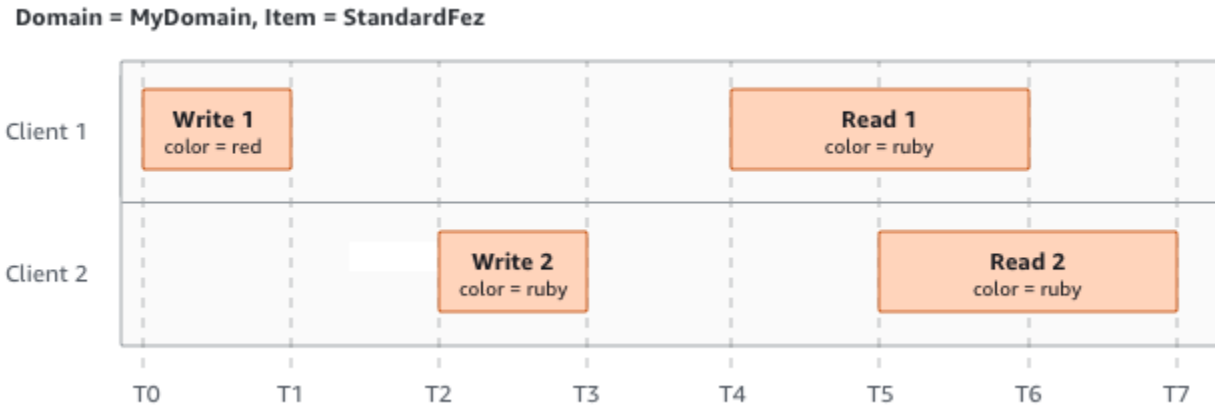
Bucket-Konfigurationen haben ein Modell mit letztendlicher Datenkonsistenz. Dies bedeutet insbesondere, dass:

- Wenn Sie einen Bucket löschen und sofort alle Buckets auflisten, wird der gelöschte Bucket möglicherweise weiterhin in der Liste angezeigt.
- Wenn Sie das Versioning für einen Bucket zum ersten Mal aktivieren, kann es einen Moment dauern, bis die Änderung vollständig verbreitet ist. Wir empfehlen, nach der Aktivierung der Versionierung 15 Minuten zu warten, bevor Sie Schreibvorgänge (PUToder DELETE Anfragen) für Objekte im Bucket ausführen.

Gleichzeitige Anwendungen

Dieser Abschnitt enthält Beispiele für das zu erwartende Verhalten von Amazon S3, wenn mehrere Kunden in dieselben Elemente schreiben.

In diesem Beispiel werden W1 (Lesen 1) und W2 (Lesen 2) abgeschlossen, bevor R1 (Lesen 1) und R2 (Lesen 2) starten. Da S3 stark konsistent ist, geben R1 und R2 beide `color = ruby` zurück.



Im nächsten Beispiel ist W2 vor dem Start von R1 nicht abgeschlossen. Daher könnte R1 `color = ruby` oder `color = garnet` zurückgeben. Da W1 und W2 jedoch vor dem Start von R2 fertig sind, gibt R2 zurück `color = garnet`.

Im letzten Beispiel beginnt W2, bevor W1 eine Bestätigung erhalten hat. Daher werden diese Schreibvorgänge als gleichzeitig betrachtet. Amazon S3 verwendet intern last-writer-wins Semantik, um zu bestimmen, welcher Schreibvorgang Vorrang hat. Die Reihenfolge, in der Amazon S3 die Anforderungen erhält, und die Reihenfolge, in der Anwendungen Bestätigungen erhalten, können aber aufgrund von verschiedenen Faktoren wie Netzwerklatenz nicht vorhergesagt werden. Beispielsweise kann W2 von einer EC2 Amazon-Instance in derselben Region initiiert werden, während W1 von einem weiter entfernten Host initiiert werden kann. Die beste Methode, um den endgültigen Wert zu bestimmen, besteht darin, einen Lesevorgang auszuführen, nachdem beide Schreibvorgänge bestätigt wurden.

Zugehörige Services

Nachdem Sie Ihre Daten in Amazon S3 geladen haben, können Sie sie mit anderen AWS Diensten verwenden. Häufig genutzte Services:

- Amazon Elastic Compute Cloud (AmazonEC2) — Bietet sichere und skalierbare Rechenkapazität in der AWS Cloud. EC2Durch die Nutzung von Amazon müssen Sie nicht im Voraus in Hardware investieren, sodass Sie Anwendungen schneller entwickeln und bereitstellen können. Sie können Amazon verwendenEC2, um so viele oder so wenige virtuelle Server zu starten, wie Sie benötigen, Sicherheit und Netzwerke zu konfigurieren und Speicher zu verwalten.
- Amazon EMR — Unterstützt Unternehmen, Forscher, Datenanalysten und Entwickler bei der einfachen und kostengünstigen Verarbeitung großer Datenmengen. Amazon EMR verwendet ein gehostetes Hadoop-Framework, das auf der Web-Scale-Infrastruktur von Amazon EC2 und Amazon S3 läuft.
- AWS Snow Family — Hilft Kunden, die Betriebsabläufe in rauen Umgebungen außerhalb von Rechenzentren und an Standorten durchführen müssen, an denen es an konsistenter Netzwerkkonnektivität mangelt. Sie können AWS Snow Family-Geräte verwenden, um lokal und kostengünstig auf die Speicher- und Rechenleistung von AWS Cloud Orten zuzugreifen, an denen eine Internetverbindung möglicherweise keine Option ist.
- AWS Transfer Family— Bietet vollständig verwaltete Unterstützung für Dateiübertragungen direkt in und aus Amazon S3 oder Amazon Elastic File System (AmazonEFS) mithilfe von Secure Shell (SSH) File Transfer Protocol (SFTP), File Transfer Protocol over SSL (FTPS) und File Transfer Protocol (FTP).

Zugriff auf Amazon S3

Sie können mit Amazon S3 auf eine der folgenden Arten arbeiten:

AWS Management Console

Die Konsole ist eine webbasierte Benutzeroberfläche für die Verwaltung von Amazon S3 und AWS Ressourcen. Wenn Sie sich für eine angemeldet haben AWS-Konto, können Sie auf die Amazon S3 S3-Konsole zugreifen, indem Sie sich bei der anmelden AWS Management Console und auf der AWS Management Console Startseite S3 auswählen.

AWS Command Line Interface

Sie können die AWS Befehlszeilentools verwenden, um Befehle auszugeben oder Skripts an der Befehlszeile Ihres Systems zu erstellen, um Aufgaben AWS (einschließlich S3) auszuführen.

Das AWS Command Line Interface (AWS CLI) bietet Befehle für eine Vielzahl von AWS-Services. Das AWS CLI wird unter Windows, MacOS und Linux unterstützt. Informationen zu den ersten Schritten finden Sie im AWS Command Line Interface -Benutzerhandbuch. Weitere Informationen über die Befehle für Amazon S3 finden Sie unter s3api und s3control in der AWS CLI -Befehlsreferenz.

AWS SDKs

AWS stellt SDKs (Software Development Kits) bereit, die aus Bibliotheken und Beispielcode für verschiedene Programmiersprachen und Plattformen bestehen (Java, Python, Ruby,. NET, iOS, Android usw.). AWS SDKsSie bieten eine bequeme Möglichkeit, programmatischen Zugriff auf S3 und AWS zu erstellen. Amazon S3 ist ein REST Service. Sie können Anfragen mithilfe der AWS SDK Bibliotheken an Amazon S3 senden, die das zugrunde liegende Amazon S3 einschließen REST API und Ihre Programmieraufgaben vereinfachen. SDKsSie kümmern sich beispielsweise um Aufgaben wie das Berechnen von Signaturen, das kryptografische Signieren von Anfragen, das Verwalten von Fehlern und das automatische Wiederholen von Anfragen. Informationen zu den AWS SDKs, einschließlich deren Herunterladen und Installieren, finden Sie unter Tools für. AWS

Jede Interaktion mit Amazon S3 erfolgt entweder authentifiziert oder anonym. Wenn Sie die verwenden AWS SDKs, berechnen die Bibliotheken die Signatur für die Authentifizierung anhand der von Ihnen bereitgestellten Schlüssel. Weitere Informationen darüber, wie Sie Anfragen an Amazon S3 stellen können, finden Sie unter Anfragen stellen.

Amazon S3 REST API

Die Amazon S3-Architektur ist so ausgelegt, dass sie unabhängig von Programmiersprachen ist und unsere von AWS unterstützten Schnittstellen verwendet, um Objekte zu speichern und abzurufen. Sie können mit Amazon S3 und AWS programmatisch auf S3 zugreifen. REST API Das REST API ist eine HTTP Schnittstelle zu Amazon S3. Mit dem verwenden Sie HTTP Standardanforderungen RESTAPI, um Buckets und Objekte zu erstellen, abzurufen und zu löschen.

Um das zu verwenden RESTAPI, können Sie jedes Toolkit verwenden, das dies unterstützt. HTTP Sie können sogar einen Browser verwenden, um Objekte zu laden, wenn diese anonym lesbar sind.

Das REST API verwendet HTTP Standardheader und Statuscodes, sodass Standardbrowser und Toolkits erwartungsgemäß funktionieren. In einigen Bereichen haben wir Funktionen hinzugefügt HTTP (zum Beispiel haben wir Header hinzugefügt, um die Zugriffskontrolle zu unterstützen). In diesen Fällen haben wir unser Bestes getan, um die neuen Funktionen so hinzuzufügen, dass sie dem Stil der HTTP Standardnutzung entsprechen.

Wenn Sie in Ihrer Anwendung direkte REST API Anrufe tätigen, müssen Sie den Code zur Berechnung der Signatur schreiben und der Anfrage hinzufügen. Weitere Informationen darüber, wie Sie Anfragen an Amazon S3 stellen, finden Sie unter Anfragen stellen in der Amazon S3 API S3-Referenz.

> **Note**
>
> SOAPAPISupport Over HTTP ist veraltet, aber es ist immer noch verfügbar. HTTPS Neuere Amazon S3 S3-Funktionen werden nicht unterstütztSOAP. Wir empfehlen Ihnen, entweder den REST API oder den zu verwenden AWS SDKs.

Bezahlen für Amazon S3

Die Preise für Amazon S3 sind so ausgelegt, dass Sie nicht im Hinblick auf die Speicheranforderungen Ihrer Anwendung planen müssen. Die meisten Speicheranbieter verlangen, dass Sie eine vorgegebene Menge an Speicherkapazität und Netzwerkübertragungskapazität erwerben. Wenn Sie in diesem Szenario diese Kapazität überschreiten, wird Ihr Service abgeschaltet, oder Sie zahlen hohe Überziehungsgebühren. Wenn Sie diese Kapazität nicht überschreiten, zahlen Sie genauso viel, als wenn Sie sie verwendet hätten.

Amazon S3 stellt Ihnen nur Gebühren für Kapazitäten in Rechnung, die Sie tatsächlich genutzt haben, ohne verborgene Kosten und Überziehungsgebühren. Dieses Modell bietet Ihnen einen Service mit variablen Kosten, der mit Ihrem Unternehmen mitwachsen kann und Ihnen gleichzeitig die Kostenvorteile der Infrastruktur bietet. AWS Weitere Informationen finden Sie unter Amazon S3 – Preise.

Wenn Sie sich für registrieren AWS, AWS-Konto ist Ihr automatisch für alle Services in angemeldet AWS, einschließlich Amazon S3. Es werden jedoch nur die Services in Rechnung gestellt, die Sie tatsächlich nutzen. Wenn Sie neuer Amazon S3-Kunde sind, können Sie kostenlos mit Amazon S3 beginnen. Weitere Informationen finden Sie unter Kostenloses Kontingent für AWS.

Um Ihre Rechnung anzuzeigen, navigieren Sie zu Fakturierungs- und Kostenverwaltungs-Dashboard in der AWS Billing and Cost Management -Konsole. Weitere Informationen zur AWS-Konto Abrechnung finden Sie im AWS Billing Benutzerhandbuch. Wenn Sie Fragen zur AWS Abrechnung und haben AWS-Konten, wenden Sie sich an den AWS Support.

PCIDSSEinhaltung der Vorschriften

Amazon S3 unterstützt die Verarbeitung, Speicherung und Übertragung von Kreditkartendaten durch einen Händler oder Dienstleister und wurde als konform mit dem Payment Card Industry (PCI) Data Security Standard (DSS) validiert. Weitere Informationen zum AWS PCI Compliance-Paket PCIDSS, einschließlich der Beantragung einer Kopie, finden Sie unter PCIDSSStufe 1.

Erste Schritte mit Amazon S3

Sie können mit Amazon S3 beginnen, indem Sie mit Buckets und Objekten arbeiten. Ein Bucket ist ein Container für Objekte. Ein Objekt ist eine Datei und alle Metadaten, die diese Datei beschreiben.

Um ein Objekt in Amazon S3 zu speichern, erstellen Sie einen Bucket und laden das Objekt dann in den Bucket hoch. Wenn sich das Objekt im Bucket befindet, können Sie es öffnen, herunterladen und verschieben. Wenn Sie kein Objekt oder einen Bucket mehr benötigen, können Sie Ihre Ressourcen aufräumen.

Mit Amazon S3 zahlen Sie nur für das, was Sie tatsächlich nutzen. Weitere Informationen zu den Funktionen und Preisen von Amazon S3 finden Sie unter Amazon S3. Wenn Sie neuer Amazon-S3-Kunde sind, können Sie kostenlos mit Amazon S3 beginnen. Weitere Informationen finden Sie unter Kostenloses Kontingent für AWS.

> **Note**
>
> Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter S3 Express One Zone und Arbeiten mit Verzeichnis-Buckets.

Video: Erste Schritte mit Amazon S3

Voraussetzungen

Bevor Sie beginnen, sollten Sie sicherstellen, dass Sie die in Einrichten von Amazon S3 beschriebenen Schritte ausgeführt haben.

Einrichten von Amazon S3

Wenn Sie sich für registrieren AWS, AWS-Konto ist Ihr automatisch für alle Services angemeldet AWS, einschließlich Amazon S3. Berechnet werden Ihnen aber nur die Services, die Sie nutzen.

Mit Amazon S3 zahlen Sie nur für das, was Sie tatsächlich nutzen. Weitere Informationen zu den Funktionen und Preisen von Amazon S3 finden Sie unter Amazon S3. Wenn Sie neuer Amazon-S3-Kunde sind, können Sie kostenlos mit Amazon S3 beginnen. Weitere Informationen finden Sie unter Kostenloses Kontingent für AWS.

Um Amazon S3 einzurichten, führen Sie die Schritte in den folgenden Abschnitten aus.

Wenn Sie sich für Amazon S3 registrieren AWS und es einrichten, können Sie optional die Anzeigesprache in der ändern AWS Management Console. Weitere Informationen finden Sie unterChanging the language of the AWS Management Console(Ändern der Sprache der Konsole) im Handbuch Erste Schritte mit AWS Management Console .

Themen

- Melden Sie sich an für ein AWS-Konto
- Erstellen eines Benutzers mit Administratorzugriff

Melden Sie sich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können Ihre aktuellen Kontoaktivitäten jederzeit einsehen und Ihr Konto verwalten, indem Sie zu https://aws.amazon.com/gehen und Mein Konto auswählen.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS Management Consoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für Ihren Root-Benutzer.

   Anweisungen finden Sie im Benutzerhandbuch unter Aktivieren eines virtuellen MFA Geräts für Ihren AWS-Konto IAM Root-Benutzer (Konsole).

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.
2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis im AWS IAM Identity Center Benutzerhandbuch.

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM Identity Center-Benutzer anzumelden, verwenden Sie die Anmeldung, URL die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM Identity Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Zugangsportal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center Benutzerhandbuch.

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center Benutzerhandbuch.

Schritt 1: Erstellen des ersten S3-Buckets

Nachdem Sie sich für angemeldet haben AWS, können Sie mit dem einen Bucket in Amazon S3 erstellen AWS Management Console. Alle Objekte in Amazon S3 werden in einem Bucket gespeichert. Bevor Sie Daten in Amazon S3 speichern können, müssen Sie einen Bucket erstellen.

> **Note**
>
> Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter S3 Express One Zone und Arbeiten mit Verzeichnis-Buckets.

> **Note**
>
> Für das Erstellen eines Buckets werden Ihnen keine Gebühren in Rechnung gestellt. Es fallen nur Gebühren für das Speichern von Objekten im Bucket und für die Übertragung von Objekten in den und aus dem Bucket an. Die Gebühren, die beim Durcharbeiten der Beispiele in diesem Handbuch anfallen, sind minimal (weniger als 1,0 USD). Weitere Information zu Speicherkosten finden Sie unter Amazon S3 – Preise.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen der aktuell angezeigten Datei aus AWS-Region. Wählen Sie als Nächstes die Region aus, in der Sie einen Bucket erstellen möchten.

> **Note**
>
> Wählen Sie eine Region in der Nähe aus, um Latenz und Kosten gering zu halten und behördliche Vorschriften zu erfüllen. In einer Region gespeicherte Objekte verbleiben so lange in der Region, bis sie explizit in eine andere Region verschoben werden. Eine Liste von Amazon S3 finden Sie AWS-Regionen unter AWS-Service Endpoints in der Allgemeine Amazon Web Services-Referenz.

3. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
4. Wählen Sie Create Bucket (Bucket erstellen) aus.

   Anschließend wird die Seite Bucket erstellen geöffnet.
5. Sehen Sie sich unter Allgemeine Konfiguration an, AWS-Region wo Ihr Bucket erstellt wird.
6. Wählen Sie unter Bucket-Typ die Option Allgemeiner Zweck aus.
7. Geben Sie unter Bucket Name (Bucket-Name) einen Namen für den Bucket ein.

   Der Bucket-Name ...:

   - Muss innerhalb einer Partition einzigartig sein. Eine Partition ist eine Gruppierung von Regionen. AWS verfügt derzeit über drei Partitionen: `aws` (Standardregionen), `aws-cn` (China-Regionen) und `aws-us-gov` (AWS GovCloud (US) Regions).
   - zwischen 3 und 63 Zeichen lang sein,
   - Darf nur aus Kleinbuchstaben, Zahlen, Punkten (.) und Bindestrichen (-) bestehen. Aus Gründen der besten Kompatibilität empfehlen wir, Punkte (.) in Bucket-Namen zu vermeiden, mit Ausnahme von Buckets, die nur für statisches Website-Hosting verwendet werden.
   - Muss mit einer Zahl oder einem Buchstaben beginnen und enden.

   Der Name eines einmal erstellter Buckets kann nicht nachträglich geändert werden. AWS-Konto Derjenige, der den Bucket erstellt, besitzt ihn. Weitere Informationen zur Benennung von Buckets finden Sie unter Allgemeine Benennungsregeln für Buckets.

> **Important**
>
> Vermeiden Sie, vertrauliche Informationen, wie Kontonummern, in den Bucket-Namen einzubeziehen. Der Bucket-Name ist in dem Punkt sichtbarURLs, auf den die Objekte im Bucket verweisen.

8. AWS Management Console ermöglicht es Ihnen, die Einstellungen eines vorhandenen Buckets in Ihren neuen Bucket zu kopieren. Wenn Sie die Einstellungen eines vorhandenen Buckets nicht kopieren möchten, fahren Sie mit dem nächsten Schritt fort.

   > **Note**
   >
   > Diese Option:
   >
   > - Ist in der nicht verfügbar AWS CLI und ist nur in der Konsole verfügbar
   > - Ist für Verzeichnis-Buckets nicht verfügbar
   > - Kopiert die Bucket-Richtlinie nicht aus dem vorhandenen Bucket in den neuen Bucket

   Um die Einstellungen eines vorhandenen Buckets zu kopieren, wählen Sie unter Einstellungen aus vorhandenem Bucket kopieren die Option Bucket auswählen aus. Das Fenster „Bucket auswählen“ wird geöffnet. Suchen Sie den Bucket mit den Einstellungen, die Sie kopieren möchten, und wählen Sie Bucket auswählen aus. Das Fenster „Bucket auswählen“ wird geschlossen, und das Fenster „Bucket erstellen“ wird erneut geöffnet.

   Unter Einstellungen aus vorhandenem Bucket kopieren wird nun der Name des ausgewählten Buckets angezeigt. Außerdem wird die Option Standardwerte wiederherstellen angezeigt, mit der Sie die kopierten Bucket-Einstellungen entfernen können. Überprüfen Sie die verbleibenden Bucket-Einstellungen auf der Seite Bucket erstellen. Sie werden sehen, dass sie jetzt mit den Einstellungen des von Ihnen ausgewählten Buckets übereinstimmen. Sie können mit dem letzten Schritt fortfahren.

9. Wählen Sie unter Object Ownership eine der folgenden Einstellungen aus, um die Inhaberschaft von Objekten, die in Ihren Bucket hochgeladen wurden, zu deaktivieren oder zu aktivieren ACLs und zu kontrollieren:

ACLsdeaktiviert

- Bucket-Besitzer erzwungen (Standard) — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLswirkt sich nicht mehr auf die Zugriffsberechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet ausschließlich Richtlinien, um die Zugriffssteuerung zu definieren.

  Standardmäßig ACLs sind sie deaktiviert. Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs. Wir empfehlen, die ACLs Option weiterhin zu deaktivieren, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

ACLsaktiviert

- Bevorzugter Bucket-Besitzer — Der Bucket-Besitzer besitzt und hat die volle Kontrolle über neue Objekte, die andere Accounts in den Bucket mit den gespeicherten Objekten `bucket-owner-full-control` schreibenACL.

  Wenn Sie die bevorzugte Einstellung des Bucket-Besitzers anwenden, sodass alle Amazon S3 S3-Uploads die gespeicherten Dateien enthalten müssen `bucket-owner-full-control`ACL, können Sie eine Bucket-Richtlinie hinzufügen, die nur Objekt-Uploads zulässt, die dies verwenden. ACL
- Objekt-Writer — Derjenige AWS-Konto , der ein Objekt hochlädt, besitzt das Objekt, hat die volle Kontrolle darüber und kann anderen Benutzern Zugriff darauf gewähren. ACLs

> **Note**
>
> Die Standardeinstellung ist Bucket-Eigentümer erzwungen. Um die Standardeinstellung anzuwenden und ACLs deaktiviert zu lassen, ist nur die `s3:CreateBucket` entsprechende Berechtigung erforderlich. Zum Aktivieren ACLs benötigen Sie die `s3:PutBucketOwnershipControls` entsprechende Genehmigung.

10. Wählen Sie unter Einstellungen "Öffentlichen Zugriff beschränken" für diesen Bucket die Einstellungen zum Beschränken des öffentlichen Zugriffs aus, die Sie auf den Bucket anwenden möchten.

Alle vier Einstellungen zum Blockieren des öffentlichen Zugriffs sind standardmäßig aktiviert. Es wird empfohlen, alle Einstellungen aktiviert zu lassen, es sei denn, Sie wissen, dass Sie eine oder mehrere dieser Einstellungen für Ihren Anwendungsfall deaktivieren müssen. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

> **Note**
>
> Zum Aktivieren aller Einstellungen zum Blockieren des öffentlichen Zugriffs ist nur die `s3:CreateBucket`-Berechtigung erforderlich. Wenn Sie eine der Einstellungen zum Blockieren des öffentlichen Zugriffs deaktivieren möchten, benötigen Sie die `s3:PutBucketPublicAccessBlock`-Berechtigung.

11. (Optional) Unter Bucket Versioning (Bucket-Versionsverwaltung) können Sie auswählen, ob Sie Varianten von Objekten in Ihrem Bucket beibehalten möchten. Weitere Informationen über das Versioning finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

    Wenn Sie die Versionsverwaltung in Ihrem Bucket deaktivieren oder aktivieren möchten, wählen Sie entweder Disable (Deaktivieren) oder Enable (Aktivieren) aus.

12. (Optional) Unter Tags können Sie auswählen, ob Sie Ihrem Bucket Tags hinzufügen möchten. Tags sind Schlüssel-Wert-Paare, die zur Kategorisierung von Speicher verwendet werden.

    Wenn Sie ein Bucket-Tag hinzuzufügen, geben Sie einen Key (Schlüssel) und optional einen Value (Wert) ein. Wählen Sie dann Add Tag (Tag hinzufügen) aus.

13. Wählen Sie unter Default encryption (Standard-Verschlüsselung) Edit (Bearbeiten) aus.

14. Wählen Sie eine der folgenden Optionen unter Verschlüsselungstyp aus, um die Standardverschlüsselung zu konfigurieren:

    - Verwalteter Amazon S3 S3-Schlüssel (SSE-S3)
    - AWS Key Management Service Schlüssel (SSE-KMS)

    > **Important**
    >
    > Wenn Sie die KMS Option SSE - für Ihre standardmäßige Verschlüsselungskonfiguration verwenden, unterliegen Sie dem Kontingent für Anfragen pro Sekunde (RPS) von AWS KMS. Weitere Informationen zu AWS KMS

Kontingenten und dazu, wie Sie eine Kontingenterhöhung beantragen können, finden Sie unter Kontingente im AWS Key Management Service Entwicklerhandbuch.

Buckets und neue Objekte werden mit serverseitiger Verschlüsselung verschlüsselt. Dabei ist ein Von Amazon S3 verwalteter Schlüssel die Grundebene der Verschlüsselungskonfiguration. Weitere Informationen zur Standardverschlüsselung finden Sie unter Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets.

Weitere Informationen zur Datenverschlüsselung mit der serverseitigen Amazon-S3-Verschlüsselung finden Sie unter Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden (SSE-S3).

15. Wenn Sie die AWS Key Management Service Taste (SSE-KMS) gewählt haben, gehen Sie wie folgt vor:

    a. Geben Sie unter AWS KMS Schlüssel Ihren KMS Schlüssel auf eine der folgenden Arten an:

    - Um aus einer Liste verfügbarer KMS Schlüssel zu wählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys und wählen Sie Ihren KMSSchlüssel aus der Liste der verfügbaren Schlüssel aus.

      Sowohl der Von AWS verwalteter Schlüssel (`aws/s3`) als auch Ihr vom Kunden verwalteter Schlüssel werden in dieser Liste angezeigt. Weitere Informationen über vom Kunden verwaltete Schlüssel finden Sie unter Kundenschlüssel und AWS -Schlüssel im Entwicklerhandbuch zu AWS Key Management Service .
    - Um den KMS Schlüssel einzugebenARN, wählen Sie Enter AWS KMS key ARN und geben Sie Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
    - Um einen neuen, vom Kunden verwalteten Schlüssel in der AWS KMS Konsole zu erstellen, wählen Sie KMS Schlüssel erstellen.

      Weitere Informationen zum Erstellen eines AWS KMS key finden Sie unter Schlüssel erstellen im AWS Key Management Service Entwicklerhandbuch.

    Important

    Sie können nur KMS Schlüssel verwenden, die im selben AWS-Region Bucket verfügbar sind. Die Amazon S3 S3-Konsole listet nur die ersten 100 KMS

Schlüssel in derselben Region wie der Bucket auf. Um einen KMS Schlüssel zu verwenden, der nicht aufgeführt ist, müssen Sie Ihren KMS Schlüssel eingebenARN. Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, benötigen Sie zunächst die Erlaubnis, den Schlüssel zu verwenden, und dann müssen Sie den KMS Schlüssel eingebenARN. Weitere Informationen zu kontoübergreifenden Berechtigungen für KMS Schlüssel finden Sie im AWS Key Management Service Entwicklerhandbuch unter KMSSchlüssel erstellen, die andere Konten verwenden können. Weitere Informationen zu SSE - finden KMS Sie unterGeben Sie die serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) an. Wenn Sie eine AWS KMS key für die serverseitige Verschlüsselung in Amazon S3 verwenden, müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel und keine asymmetrischen KMS Schlüssel. Weitere Informationen finden Sie unter Identifizieren symmetrischer und asymmetrischer KMS Schlüssel im Entwicklerhandbuch.AWS Key Management Service

Weitere Informationen zum Erstellen von finden Sie unter Creating Keys im AWS Key Management Service Developer Guide. AWS KMS key Weitere Informationen zur Verwendung AWS KMS mit Amazon S3 finden Sie unterServerseitige Verschlüsselung mit AWS KMS Schlüsseln verwenden (SSE-) KMS.

b. Wenn Sie Ihren Bucket so konfigurieren, dass er die Standardverschlüsselung mit SSE - verwendetKMS, können Sie auch S3 Bucket Keys aktivieren. S3-Bucket-Keys senken die Kosten für die Verschlüsselung, indem sie den Anforderungsverkehr von Amazon S3 zu verringern AWS KMS. Weitere Informationen finden Sie unter Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys.

   Um S3-Bucket-Schlüssel zu verwenden, wählen Sie unter Bucket Key (Bucket-Schlüssel) die Option Enable (Aktivieren).

16. (Optional) Wenn Sie die S3-Objektsperre aktivieren möchten, gehen Sie wie folgt vor:

    a. Wählen Sie Erweiterte Einstellungen aus.

> ⚠ Important
>
> Durch Aktivieren der Objektsperre wird auch die Versioning für den Bucket aktiviert. Nach dem Aktivieren müssen Sie die Standardeinstellungen für die Objektsperre im

Hinblick auf die (rechtliche) Aufbewahrung konfigurieren, um neue Objekte vor dem Löschen oder Überschreiben zu schützen.

b. Wenn Sie die Objektsperre aktivieren möchten, wählen Sie Enable (Aktivieren) aus, lesen Sie die angezeigte Warnung und bestätigen Sie sie.

Weitere Informationen finden Sie unter Sperren von Objekten mit Object Lock.

> **Note**
>
> Wenn Sie einen Bucket mit aktivierter Objektsperre erstellen möchten, benötigen Sie die folgenden Berechtigungen: `s3:CreateBucket`, `s3:PutBucketVersioning` und `s3:PutBucketObjectLockConfiguration`.

17. Wählen Sie Create Bucket (Bucket erstellen) aus.

Sie haben einen Bucket in Amazon S3 erstellt.

Nächster Schritt

Informationen zum Hinzufügen eines Objekts zu einem Bucket finden Sie unter Schritt 2: Hochladen eines Objekts in Ihren Bucket.

Schritt 2: Hochladen eines Objekts in Ihren Bucket

Nachdem Sie in Amazon S3 einen Bucket erstellt haben, können Sie ein Objekt hochladen. Ein Objekt kann eine beliebige Datei sein: eine Textdatei, ein Foto, ein Video usw.

> **Note**
>
> Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter S3 Express One Zone und Arbeiten mit Verzeichnis-Buckets.

Ein Objekt in einen Bucket hochladen

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

2. Wählen Sie in der Liste Bucket den Namen des Buckets aus, zu dem Sie Ihr Objekt hochladen möchten.
3. Wählen Sie im Tab Objects (Objekte) für Ihren Bucket die Option Upload (Hochladen) aus.
4. Wählen Sie unter Files and folders (Dateien und Ordner) die Option Add files (Dateien hinzufügen) aus.
5. Wählen Sie eine hochzuladende Datei und dann Öffnen aus.
6. Klicken Sie auf Hochladen.

Sie haben erfolgreich ein Objekt zu Ihrem Bucket hochgeladen.

Nächster Schritt

Informationen zum Anzeigen des Objekts finden Sie unter Schritt 3: Herunterladen eines Objekts.

Schritt 3: Herunterladen eines Objekts

Nachdem Sie ein Objekt in einen Bucket hochgeladen haben, können Sie Informationen über Ihr Objekt anzeigen und das Objekt auf Ihren lokalen Computer herunterladen.

> **Note**
>
> Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter S3 Express One Zone und Arbeiten mit Verzeichnis-Buckets.

Verwenden der S3-Konsole

In diesem Abschnitt erfahren Sie, wie Sie mit der Amazon-S3-Konsole ein Objekt aus einem S3-Bucket herunterladen.

> **Note**
>
> - Sie können jeweils nur ein Objekt herunterladen.
> - Wenn Sie über die Amazon-S3-Konsole ein Objekt herunterladen, dessen Schlüsselname mit Punkt (`.`) endet, wird der Punkt aus dem Schlüsselnamen des heruntergeladenen

Objekts entfernt. Um den Punkt am Ende des Namens des heruntergeladenen Objekts beizubehalten, müssen Sie AWS Command Line Interface (AWS CLI) AWS SDKs, oder Amazon S3 verwenden RESTAPI.

Ein Objekt von einem S3-Bucket herunterladen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, aus dem Sie ein Objekt herunterladen möchten.
3. Sie können ein Objekt wie folgt aus einem S3-Bucket herunterladen:
   - Aktivieren Sie das Kontrollkästchen neben dem Objekt und wählen Sie Herunterladen. Wenn Sie das Objekt in einen spezifischen Ordner herunterladen möchten, wählen Sie im Menü Aktionen die Option Herunterladen als.
   - Wenn Sie eine spezifische Version des Objekts herunterladen möchten, aktivieren Sie die Schaltfläche Versionen anzeigen (neben dem Suchfeld). Aktivieren Sie das Kontrollkästchen neben der gewünschten Version des Objekts und wählen Sie Herunterladen. Wenn Sie das Objekt in einen spezifischen Ordner herunterladen möchten, wählen Sie im Menü Aktionen die Option Herunterladen als.

Sie haben Ihr Objekt erfolgreich heruntergeladen.

Nächster Schritt

Informationen zum Kopieren und Einfügen Ihres Objekts in Amazon S3 finden Sie unter Schritt 4: Kopieren Ihres Objekts in einen Ordner.

Schritt 4: Kopieren Ihres Objekts in einen Ordner

Sie haben einem Bucket bereits ein Objekt hinzugefügt und das Objekt heruntergeladen. Jetzt erstellen Sie einen Ordner, kopieren das Objekt und fügen es in den Ordner ein.

> **Note**
>
> Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter S3 Express One Zone und Arbeiten mit Verzeichnis-Buckets.

So kopieren Sie ein Objekt in einen Ordner

1. Wählen Sie in der Liste Buckets (Buckets) den Namen Ihres Buckets.
2. Wählen Sie Create folder (Ordner erstellen) aus und konfigurieren Sie einen neuen Ordner:
   a. Geben Sie einen Ordnernamen ein (z. B. `favorite-pics`).
   b. Als Einstellung für die Ordnerverschlüsselung wählen Sie Disable (Deaktivieren) aus.
   c. Wählen Sie Save (Speichern) aus.
3. Navigieren Sie zum Amazon-S3-Bucket oder -Ordner, der die Objekte enthält, die Sie kopieren möchten.
4. Markieren Sie das Kontrollkästchen links neben den Namen der Objekte, die Sie kopieren möchten.
5. Wählen Sie Actions (Aktionen) und wählen Sie aus der angezeigten Liste der Optionen die Option Copy (Kopieren) aus.

   Oder wählen Sie aus den Optionen oben rechts die Option Copy (Kopieren) aus.
6. Wählen Sie den Zielordner aus:
   a. Wählen Sie S3 durchsuchen.
   b. Wählen Sie das Optionsfeld links vom Ordnernamen aus.

      Um in einen Ordner zu navigieren und einen Unterordner als Ziel auszuwählen, wählen Sie den Ordnernamen aus.
   c. Wählen Sie Choose destination (Ziel wählen).

   Der Pfad zu Ihrem Zielordner wird im Feld Destination (Ziel) angezeigt. In Destination können Sie alternativ Ihren Zielpfad eingeben, zum Beispiel s3://*bucket-name*/*folder-name*/.
7. Wählen Sie unten rechts Copy (Kopieren).

Amazon S3 kopiert Ihre Objekte in den Zielordner.

Nächster Schritt

Informationen zum Löschen eines Objekts und eines Buckets in Amazon S3 finden Sie unter Schritt 5: Löschen Ihrer Objekte und Buckets.

Schritt 5: Löschen Ihrer Objekte und Buckets

Wenn Sie ein Objekt oder einen Bucket mehr benötigen, sollten Sie diese löschen, um weitere Gebühren zu vermeiden. Wenn Sie diesen Walkthrough für die ersten Schritte zur Übung durchgeführt haben und Ihren Bucket oder Ihre Objekte nicht verwenden möchten, sollten Sie den Bucket und die Objekte löschen, damit keine weiteren Gebühren anfallen.

Bevor Sie Ihren Bucket löschen, leeren Sie den Bucket oder löschen Sie die Objekte im Bucket. Nachdem Sie Ihre Objekte und Ihren Bucket gelöscht haben, sind die Objekte und der Bucket nicht mehr verfügbar.

Wenn Sie denselben Bucket-Namen weiter verwenden möchten, sollten Sie die Objekte löschen oder den Bucket leeren, den Bucket selbst jedoch nicht löschen. Wenn Sie einen Bucket gelöscht haben, können Sie den Namen wiederverwenden. Ein anderer Benutzer erstellt jedoch AWS-Konto möglicherweise einen Bucket mit demselben Namen, bevor Sie ihn wiederverwenden können.

> **Note**
>
> Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter S3 Express One Zone und Arbeiten mit Verzeichnis-Buckets.

Themen

- Löschen eines Objekts
- Leeren des Buckets
- Löschen des Buckets

Löschen eines Objekts

Wenn Sie nur bestimmte Objekte löschen möchten, ohne alle Objekte im Bucket zu löschen, können Sie ein Objekt löschen.

1. Wählen Sie in der Liste Buckets (Buckets) den Namen des Buckets aus, aus dem Sie ein Objekt löschen möchten.
2. Wählen Sie das zu löschende Objekt aus.
3. Wählen Sie in den Optionen oben rechts Löschen aus.
4. Geben Sie auf der Seite Objekte löschen **`delete`** ein, um das Löschen Ihrer Objekte zu bestätigen.
5. Wählen Sie Delete objects (Objekte löschen).

Leeren des Buckets

Wenn Sie Ihren Bucket löschen möchten, müssen Sie zuerst den Bucket leeren, um alle Objekte im Bucket zu löschen.

So leeren Sie einen Bucket

1. Wählen Sie in der Liste Buckets (Buckets) den Bucket aus, den Sie leeren möchten. Wählen Sie anschließend Empty (Leeren) aus.
2. Um zu bestätigen, dass Sie den Bucket leeren und alle in diesem enthaltenen Objekte löschen möchten, geben Sie unter Bucket leeren **`permanently delete`** ein.

   > **Important**
   >
   > Das Leeren des Buckets kann nicht rückgängig gemacht werden. Dem Bucket während der Löschaktion hinzugefügte Objekte werden gelöscht.

3. Um den Bucket zu leeren und alle in diesem enthaltenen Objekte zu löschen, wählen Sie Empty (Leeren) aus.

   Anschließend wird die Seite Empty bucket: Status (Bucket leeren: Status) geöffnet, auf der Sie eine Übersicht über fehlgeschlagene und erfolgreiche Objektlöschungen anzeigen können.
4. Um zur Bucket-Liste zurückzukehren, wählen Sie Exit (Beenden) aus.

Löschen des Buckets

Nachdem Sie den Bucket geleert oder alle Objekte aus dem Bucket gelöscht haben, können Sie den Bucket löschen.

1. Um einen Bucket zu löschen, wählen Sie in der Liste Buckets (Buckets) den Bucket aus.
2. Wählen Sie Löschen.
3. Geben Sie zum Bestätigen des Löschvorgangs unter Bucket löschen den Namen des Buckets ein.

   > **Important**
   >
   > Das Löschen eines Buckets kann nicht rückgängig gemacht werden. Bucket-Namen sind eindeutig. Wenn Sie Ihren Bucket löschen, kann ein anderer AWS Benutzer den Namen verwenden. Wenn Sie denselben Bucket-Namen weiter verwenden möchten, sollten Sie den Bucket nicht löschen. Sie sollten den Bucket stattdessen leeren und behalten.

4. Um den Bucket zu löschen, wählen Sie Delete bucket (Bucket löschen) aus.

Nächste Schritte

Im vorherigen Beispiel haben Sie gelernt, wie einige der grundlegenden Aufgaben in Amazon S3 erfolgen.

Die folgenden Themen erklären die Lernpfade, die Sie benutzen können, um sich ein vertieftes Wissen über Amazon S3 zu verschaffen, sodass Sie es in Ihren Anwendungen implementieren können.

> **Note**
>
> Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter S3 Express One Zone und Arbeiten mit Verzeichnis-Buckets.

Themen

- Verstehen allgemeiner Anwendungsfälle

- Kontrollieren des Zugriffs auf Buckets und Objekte
- Schützen und überwachen Sie Ihren Speicher
- Entwickeln mit Amazon S3
- Erwerben Sie neue Kenntnisse aus Tutorials
- Lernen Sie Trainings und Support kennen

Verstehen allgemeiner Anwendungsfälle

Sie können Amazon S3 verwenden, um Ihren speziellen Anwendungsfall zu unterstützen. Die AWS Solutions Library (Bibliothek für Lösungen) und AWS -Blog bieten anwendungsfallspezifische Informationen und Tutorials. Im Folgenden sind einige häufige Anwendungsfälle für Amazon S3 zu finden:

- Backup und Speicherung – Verwenden Sie Amazon S3 Speicherverwaltungsfunktionen, um Kosten zu verwalten, gesetzliche Anforderungen zu erfüllen, Latenz zu reduzieren und mehrere verschiedene Kopien Ihrer Daten für Compliance-Anforderungen zu speichern.
- Hosting von Anwendungen – Sie können Webanwendungen, die zuverlässig, hochgradig skalierbar und kostengünstig sind, bereitstellen, installieren und verwalten. Sie können beispielsweise Ihren Amazon-S3-Bucket so konfigurieren, dass er eine statische Website hostet. Weitere Informationen finden Sie unter Hosten einer statischen Website mit Amazon S3.
- Medienhosting – Entwicklung einer hoch verfügbaren Infrastruktur, die Uploads und Downloads von Video-, Foto- oder Musikdateien hostet.
- Softwarebereitstellung – Hosting von Softwareanwendungen, die Kunden herunterladen können.

Kontrollieren des Zugriffs auf Buckets und Objekte

Amazon S3 enthält eine Vielzahl von Sicherheitsfunktionen und -tools. Eine Übersicht finden Sie unter Zugriffskontrolle in Amazon S3.

Standardmäßig werden S3-Buckets und -Objekte als privat eingestuft. Sie haben nur Zugriff auf die S3-Ressourcen, die Sie erstellen. Sie können die folgenden Funktionen verwenden, um detaillierte Ressourcenberechtigungen zu erteilen, die Ihren speziellen Anwendungsfall unterstützen, oder um die Berechtigungen Ihrer Amazon S3-Ressourcen zu überprüfen.

- S3 öffentlichen Zugriff blockieren – Blockieren Sie den öffentlichen Zugriff auf S3-Buckets und -Objekte. Standardmäßig sind die Einstellungen für das Blockieren des öffentlichen Zugriffs auf Bucket-Ebene aktiviert.
- AWS Identity and Access Management (IAM) Identitäten — Verwenden Sie IAM oder AWS IAM Identity Center erstellen Sie IAM Identitäten in Ihrem, AWS-Konto um den Zugriff auf Ihre Amazon S3 S3-Ressourcen zu verwalten. Sie können es beispielsweise IAM mit Amazon S3 verwenden, um die Art des Zugriffs zu kontrollieren, den ein Benutzer oder eine Benutzergruppe auf einen Amazon S3 S3-Bucket hat, der Ihnen AWS-Konto gehört. Weitere Informationen zu IAM Identitäten und bewährten Methoden finden Sie im Benutzerhandbuch unter IAMIdentitäten (Benutzer, Benutzergruppen und Rollen). IAM
- Bucket-Richtlinien — Verwenden Sie eine IAM basierte Richtliniensprache, um ressourcenbasierte Berechtigungen für Ihre S3-Buckets und die darin enthaltenen Objekte zu konfigurieren.
- Zugriffskontrolllisten (ACLs) — Erteilen Sie autorisierten Benutzern Lese- und Schreibberechtigungen für einzelne Buckets und Objekte. In der Regel empfehlen wir, stattdessen ressourcenbasierte S3-Richtlinien (Bucket-Richtlinien und Zugriffspunktrichtlinien) oder IAM Benutzerrichtlinien für die Zugriffskontrolle zu verwenden. ACLs Richtlinien stellen eine vereinfachte und flexiblere Zugriffskontrolloption dar. Mit Bucket- und Zugriffspunktrichtlinien können Sie Regeln definieren, die allgemein für alle Anfragen an Ihre Amazon-S3-Ressourcen gelten. Weitere Informationen zu den spezifischen Fällen, in denen Sie ACLs anstelle von ressourcenbasierten Richtlinien oder IAM Benutzerrichtlinien verwenden würden, finden Sie unter. Identity and Access Management für Amazon S3
- S3 Object Ownership – Übernehmen Sie die Verantwortung für jedes Objekt in Ihrem Bucket, wodurch die Zugriffsverwaltung für in Amazon S3 gespeicherte Daten vereinfacht wird. S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, die Sie zum Deaktivieren oder Aktivieren verwenden können. ACLs Standardmäßig ACLs sind sie deaktiviert. Wenn diese ACLs Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff auf Daten ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.
- IAMAccess Analyzer for S3 — Evaluieren und überwachen Sie Ihre S3-Bucket-Zugriffsrichtlinien und stellen Sie sicher, dass die Richtlinien nur den beabsichtigten Zugriff auf Ihre S3-Ressourcen ermöglichen.

Schützen und überwachen Sie Ihren Speicher

- Schutz Ihres Speichers — Nachdem Sie Buckets erstellt und Objekte in Amazon S3 hochgeladen haben, können Sie Ihren Objektspeicher schützen. Sie können beispielsweise S3-Versionierung,

S3-Replikation und Multi-Region-Access-Point-Failover-Kontrollen für die Notfallwiederherstellung verwenden, um Ihre Daten AWS Backup zu sichern, und S3 Object Lock, um Aufbewahrungsfristen festzulegen, Löschungen und Überschreibungen zu verhindern und Compliance-Anforderungen zu erfüllen.

- Überwachung Ihres Speichers — Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von Amazon S3 und Ihren AWS Lösungen. Sie können Speicheraktivitäten und -kosten überwachen. Wir empfehlen auch, dass Sie von allen Teilen Ihrer AWS -Lösung Überwachungsdaten sammeln, damit Sie Ausfälle, die sich über mehrere Punkte erstrecken, leichter debuggen können.

  Sie können auch Analysen und Erkenntnisse in Amazon S3 verwenden, um Ihre Speichernutzung zu verstehen, zu analysieren und zu optimieren. Verwenden Sie z. B. Amazon S3 Storage Lens, um Ihren Speicher zu verstehen, zu analysieren und zu optimieren. S3 Storage Lens bietet mehr als 29 Nutzungs- und Aktivitätsmetriken und interaktive Dashboards, um Daten für Ihre gesamte Organisation, bestimmte Konten, Regionen, Buckets oder Präfixe zu aggregieren. Verwenden Sie Speicherklassenanalysen, um Speicherzugriffsmuster zu analysieren und zu entscheiden, wann es an der Zeit ist, Ihre Daten in eine kostengünstigere Speicherklasse zu verschieben. Um Ihre Kosten zu verwalten, können Sie S3 Lifecycle verwenden.

Entwickeln mit Amazon S3

Amazon S3 ist ein REST Service. Sie können Anfragen mithilfe der Bibliotheken REST API oder der AWS SDK Bibliotheken an Amazon S3 senden, die das zugrunde liegende Amazon S3 umschließen REST API und so Ihre Programmieraufgaben vereinfachen. Sie können auch die AWS Command Line Interface (AWS CLI) verwenden, um Amazon S3 API S3-Aufrufe zu tätigen. Weitere Informationen finden Sie unter Anfragen stellen in der Amazon S3 API S3-Referenz.

Amazon S3 REST API ist eine HTTP Schnittstelle zu Amazon S3. Mit dem verwenden Sie HTTP Standardanforderungen RESTAPI, um Buckets und Objekte zu erstellen, abzurufen und zu löschen. Um das zu verwenden RESTAPI, können Sie jedes Toolkit verwenden, das dies unterstützt. HTTP Sie können sogar einen Browser verwenden, um Objekte zu laden, wenn diese anonym lesbar sind. Weitere Informationen finden Sie unter Entwickeln mit Amazon S3 in der Amazon S3 API S3-Referenz.

Damit Sie Anwendungen in der von Ihnen bevorzugten Sprache entwickeln können, stellen wir die folgenden Ressourcen zur Verfügung.

AWS CLI

Sie können auf die Funktionen von Amazon S3 mit AWS CLI zugreifen. Informationen zum Herunterladen und Konfigurieren von finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS CLI in der Amazon S3 API S3-Referenz. AWS CLI

Das AWS CLI bietet zwei Befehlsebenen für den Zugriff auf Amazon S3: Befehle auf hoher Ebene (s3) und Befehle auf API Hochebene (s3api) und Befehle. `s3control` Die High-Level-S3-Befehle vereinfachen das Ausführen häufiger Vorgänge. Zu diesen zählen das Erstellen, Bearbeiten und Löschen von Objekten und Buckets. Die Befehle s3api und s3control bieten direkten Zugriff auf alle Amazon S3 API S3-Operationen, mit denen Sie erweiterte Operationen ausführen können, die mit den allgemeinen Befehlen allein möglicherweise nicht möglich sind.

Eine Liste der Amazon S3 AWS CLI S3-Befehle finden Sie unter s3, s3api und s3control.

AWS SDKsund Entdecker

Sie können das verwenden AWS SDKs, wenn Sie Anwendungen mit Amazon S3 entwickeln. Sie AWS SDKs vereinfachen Ihre Programmieraufgaben, indem Sie die zugrunde liegenden Daten einschließen RESTAPI. Die AWS Mobile SDKs - und die JavaScript Amplify-Bibliothek stehen auch für die Erstellung verbundener Mobil- und AWS Webanwendungen zur Verfügung.

Darüber hinaus sind AWS Explorer für Visual Studio und Eclipse für Java verfügbar. AWS SDKs IDE In diesem Fall sind der Explorer SDKs und der Explorer als Toolkits gebündelt. AWS

Weitere Informationen finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS SDKs in der Amazon S3 API S3-Referenz.

Beispiel-Code und Bibliotheken

Das AWS -Entwicklerzentrum und der AWS Code-Beispiel-Katalog und Bibliotheken bieten Beispiel-Code und Bibliotheken speziell für Amazon S3. Sie können diese Codebeispiele verwenden, um zu verstehen, wie Amazon S3 implementiert wirdAPI. Sie können sich auch die Amazon Simple Storage Service API Reference ansehen, um die API Abläufe von Amazon S3 im Detail zu verstehen.

Erwerben Sie neue Kenntnisse aus Tutorials

Sie können mit step-by-step Tutorials beginnen, um mehr über Amazon S3 zu erfahren. Diese Tutorials sind für Laborumgebungen vorgesehen und sie benutzen fiktive Unternehmensnamen, Benutzernamen usw. In den Tutorials finden Sie allgemeine Anleitungen. Ohne die sorgfältige Prüfung und Anpassung an die besonderen Gegebenheiten der Umgebung Ihrer Organisation sind sie nicht zur direkten Verwendung in einer Produktionsumgebung bestimmt.

Erste Schritte

- Tutorial: Speichern und Abrufen einer Datei mit Amazon S3
- Tutorial: Erste Schritte mit S3 Intelligent-Tiering
- Tutorial: Erste Schritte mit den Amazon-S3-Glacier-Speicherklassen

Optimierung der Speicherkosten

- Tutorial: Erste Schritte mit S3 Intelligent-Tiering
- Tutorial: Erste Schritte mit den Amazon-S3-Glacier-Speicherklassen
- Tutorial: Optimieren von Kosten und Transparenz bei der Nutzung mit S3 Storage Lens

Verwalten des Speichers

- Tutorial: Erste Schritte mit Multi-Region-Zugangspunkten in Amazon S3
- Tutorial: Replizieren vorhandener Objekte in Ihren Amazon-S3-Buckets mit S3 Batch Replication

Hosten von Videos und Websites

- Tutorial: Hosten von On-Demand-Streaming-Videos mit Amazon S3 CloudFront, Amazon und Amazon Route 53
- Tutorial: Konfigurieren einer statischen Website auf Amazon S3
- Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne

Verarbeiten von Daten

- Tutorial: Transformieren von Daten für Ihre Anwendung mit S3 Object Lambda
- Tutorial: Erkennen und Redigieren von PII Daten mit S3 Object Lambda und Amazon Comprehend
- Tutorial: Verwenden von S3 Object Lambda, um Bilder beim Abrufen dynamisch mit Wasserzeichen zu versehen
- Tutorial: Batch-Transcodierung von Videos mit S3 Batch Operations

Schutz von Daten

- Tutorial: Überprüfen der Integrität von Daten in Amazon S3 mit zusätzlichen Prüfsummen
- Tutorial: Replizieren von Daten innerhalb und zwischen Daten AWS-Regionen mithilfe von S3 Replication
- Tutorial: Schutz von Daten in Amazon S3 vor dem versehentlichen Löschen oder Anwendungsfehlern mithilfe von S3 Versioning, S3 Object Lock und S3 Replication
- Tutorial: Replizieren vorhandener Objekte in Ihren Amazon S3-Buckets mit S3 Batch Replication

Lernen Sie Trainings und Support kennen

Sie können von AWS Experten lernen, um Ihre Fähigkeiten zu erweitern und Expertenunterstützung bei der Erreichung Ihrer Ziele zu erhalten.

- Training – Trainingsressourcen bieten einen praktischen Ansatz zum Erlernen von Amazon S3. Weitere Informationen finden Sie unter AWS -Schulungen und -Zertifizierung und AWS -Online-Kundendienstmitarbeiter.
- Diskussionsforen – Im Forum können Sie Beiträge überprüfen, um zu erfahren, was mit Amazon S3 alles möglich ist und was nicht. Sie können auch Ihre Fragen posten. Weitere Informationen finden Sie unter Diskussionsforen.
- Technischer Support – Wenn Sie weitere Fragen haben, können Sie sich an den Technischen Support wenden.

Erstellen, Konfigurieren und Arbeiten mit Amazon S3-Buckets

Um Ihre Daten in Amazon S3 zu speichern, arbeiten Sie mit Ressourcen, die als Buckets und Objekte bezeichnet werden. Ein Bucket ist ein Container für Objekte. Ein Objekt ist eine Datei und alle Metadaten, die diese Datei beschreiben.

Um ein Objekt in Amazon S3 zu speichern, erstellen Sie einen Bucket und laden das Objekt dann in einen Bucket hoch. Wenn sich das Objekt im Bucket befindet, können Sie es öffnen, herunterladen und verschieben. Wenn Sie kein Objekt oder einen Bucket mehr benötigen, können Sie Ihre Ressourcen aufräumen.

> **Note**
>
> Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter S3 Express One Zone und Arbeiten mit Verzeichnis-Buckets.

> **Note**
>
> Mit Amazon S3 zahlen Sie nur für das, was Sie tatsächlich nutzen. Weitere Informationen zu den Funktionen und Preisen von Amazon S3 finden Sie unter Amazon S3. Wenn Sie neuer Amazon-S3-Kunde sind, können Sie kostenlos mit Amazon S3 beginnen. Weitere Informationen finden Sie unter Kostenloses Kontingent für AWS.

Die Themen in diesem Abschnitt geben einen Überblick über die Arbeit mit Buckets in Amazon S3. Sie enthalten Informationen über das Benennen, Erstellen, Zugreifen und Löschen von Buckets. Weitere Informationen zum Anzeigen oder Auflisten von Objekten in einem Bucket finden Sie unter Organisieren, Auflisten und Arbeiten mit Ihren Objekten.

Themen

- Bucket-Übersicht
- Allgemeine Bucket-Muster für die Erstellung von Anwendungen auf Amazon S3

- Allgemeine Benennungsregeln für Buckets
- Bucket-Kontingente, Beschränkungen und Einschränkungen
- Zugreifen auf einen Amazon S3 S3-Bucket
- Erstellen eines Buckets
- Anzeigen der Eigenschaften eines S3-Buckets
- Amazon S3 S3-Buckets auflisten
- Leeren eines Buckets
- Löschen eines Buckets
- Arbeiten mit Mountpoint für Amazon S3
- Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration
- Nutzen von Buckets mit Zahlung durch den Anforderer für Speicherübertragungen und Nutzung

Bucket-Übersicht

Zum Hochladen Ihrer Daten (Fotos, Videos, Dokumente usw.) in Amazon S3 müssen Sie zunächst einen S3-Bucket in einer AWS-Regionen erstellen.

Es gibt zwei Arten von Amazon-S3-Buckets: Allzweck-Buckets und Verzeichnis-Buckets. Wählen Sie den Bucket-Typ, der am besten zu Ihren Anwendungs- und Leistungsanforderungen passt:

- Allzweck-Buckets sind der ursprüngliche S3-Bucket-Typ und werden für die meisten Anwendungsfälle und Zugriffsmuster empfohlen. Allzweck-Buckets ermöglichen auch Objekte, die in allen Speicherklassen gespeichert werden, mit Ausnahme von S3 Express One Zone.
- Directory-Buckets verwenden die Speicherklasse S3 Express One Zone. Diese wird empfohlen, wenn Ihre Anwendung leistungsempfindlich ist und von einstelligen Millisekunden und Latenzen profitiert, siehe PUT undGET. S3 Express One Zone Arbeiten mit Verzeichnis-Buckets

In den folgenden Abschnitten finden Sie weitere Informationen zu Allzweck-Buckets, einschließlich Regeln zur Benennung von Buckets, Kontingenten und Einzelheiten zur Bucket-Konfiguration. Eine Liste der Einschränkungen und Beschränkungen im Zusammenhang mit Amazon S3 S3-Buckets finden Sie unter. Bucket-Kontingente, Beschränkungen und Einschränkungen

Themen

- Übersicht über Allzweck-Buckets

- Allgemeine Bucket-Muster
- Berechtigungen
- Verwalten des öffentlichen Zugriffs auf Buckets
- Optionen für die Bucket-Konfiguration
- Bucket-Operationen

Übersicht über Allzweck-Buckets

Ein Allzweck-Bucket ist ein Container für Objekte, die in Amazon S3 gespeichert sind. Sie können eine beliebige Anzahl von Objekten in einem Bucket speichern und alle Konten haben ein Standard-Bucket-Kontingent von 10.000 Allzweck-Buckets. Um Ihre Bucket-Auslastung oder Ihr Bucket-Kontingent einzusehen oder eine Erhöhung dieses Kontingents zu beantragen, besuchen Sie die Service Quotas Quotas-Konsole.

- Allgemeine Bucket-Kontingente für kommerzielle Regionen können nur von USA Ost (Nord-Virginia) aus angezeigt und verwaltet werden.
- Allgemeine Bucket-Kontingente für AWS GovCloud (US) können nur von AWS GovCloud (USA West) aus angezeigt und verwaltet werden.

Jedes Objekt ist in einem Bucket enthalten. Wenn das benannte Objekt beispielsweise im `amzn-s3-demo-bucket` Bucket in der Region USA West (Oregon) gespeichert `photos/puppy.jpg` ist, kann es mithilfe von adressiert werden. URL `https://amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com/photos/puppy.jpg` Weitere Informationen siehe Zugriff auf einen Bucket.

In Bezug auf die Implementierung sind Buckets und Objekte AWS Ressourcen, und Amazon S3 bietet Ihnen die APIs Möglichkeit, sie zu verwalten. Sie können beispielsweise mit Amazon S3 einen Bucket erstellen und Objekte hochladenAPI. Sie können diese Vorgänge auch in der Amazon-S3-Konsole durchführen. Die Konsole verwendet Amazon S3APIs, um Anfragen an Amazon S3 zu senden.

In diesem Abschnitt wird beschrieben, wie Buckets verwendet werden. Weitere Informationen zur Arbeit mit Objekten finden Sie unter Übersicht über Amazon-S3-Objekte.

Amazon S3 unterstützt globale Buckets, was bedeutet, dass jeder Bucket-Name AWS-Regionen innerhalb einer Partition AWS-Konten insgesamt eindeutig sein muss. Eine Partition ist eine

Gruppierung von Regionen. AWS hat derzeit drei Partitionen: `aws` (Standardregionen), `aws-cn` (China-Regionen) und `aws-us-gov` (AWS GovCloud (US)).

Nachdem ein Bucket erstellt wurde, kann der Name dieses Buckets nicht von einem anderen AWS-Konto in derselben Partition verwendet werden kann, bis der Bucket gelöscht wird. Verlassen Sie sich nicht auf eine spezifische Benennungskonvention für Buckets für Verfügbarkeits- oder Sicherheitsprüfungszwecke. Namensrichtlinien für Buckets finden Sie unter Allgemeine Benennungsregeln für Buckets.

Amazon S3 erstellt Buckets in der von Ihnen angegebenen Region. Um die Latenz zu reduzieren, die Kosten zu minimieren oder regulatorische Anforderungen zu erfüllen, wählen Sie eine Lösung AWS-Region , die sich geografisch in Ihrer Nähe befindet. Wenn Sie beispielsweise in Europa ansässig sind, könnte es vorteilhaft sein, Buckets in den Regionen EU (Irland) oder EU (Frankfurt) zu erstellen. Eine Liste der Amazon-S3-Regionen finden Sie unter Regions and Endpoints (Regionen und Endpunkte) in der Allgemeinen AWS -Referenz.

> **Note**
>
> Objekte, die zu einem Bucket gehören, den Sie in einer bestimmten Region erstellen, verlassen diese Region AWS-Region nie, es sei denn, Sie übertragen sie explizit in eine andere Region. Beispielsweise verlassen in der Region EU (Irland) gespeicherte Objekte diese Region nie.

Allgemeine Bucket-Muster

Wenn Sie Anwendungen auf Amazon S3 erstellen, können Sie spezielle Allzweck-Buckets verwenden, um verschiedene Datensätze oder Workloads zu trennen. Abhängig von Ihrem Anwendungsfall gibt es unterschiedliche Entwurfsmuster und bewährte Methoden für die Verwendung von Buckets. Weitere Informationen finden Sie unter Allgemeine Bucket-Muster für die Erstellung von Anwendungen auf Amazon S3.

Berechtigungen

Sie können Ihre Root-Benutzer des AWS-Kontos Anmeldeinformationen verwenden, um einen Bucket zu erstellen und jeden anderen Amazon S3 S3-Vorgang auszuführen. Sie sollten die Stammbenutzer-Anmeldeinformationen Ihres AWS-Konto jedoch nicht verwenden, um Anforderungen zu senden, z. B. für die Erstellung eines Buckets. Erstellen Sie stattdessen einen Benutzer AWS

Identity and Access Management (IAM) und gewähren Sie diesem Benutzer vollen Zugriff (Benutzer haben standardmäßig keine Berechtigungen).

Diese Benutzer werden als Administratorenbezeichnet. Sie können die Administrator-Benutzeranmeldedaten anstelle der Root-Benutzeranmeldedaten Ihres Kontos verwenden, um mit Ihnen zu interagieren AWS und Aufgaben auszuführen, z. B. einen Bucket zu erstellen, Benutzer zu erstellen und ihnen Berechtigungen zu erteilen.

Weitere Informationen finden Sie unter Root-Benutzer des AWS-Kontos Anmeldeinformationen und IAM Benutzeranmeldedaten in der AWS Allgemeinen Referenz und Bewährte Sicherheitsmethoden IAM im IAM Benutzerhandbuch.

AWS-Konto Derjenige, der eine Ressource erstellt, besitzt diese Ressource. Wenn Sie beispielsweise einen IAM Benutzer in Ihrem erstellen AWS-Konto und diesem Benutzer die Berechtigung zum Erstellen eines Buckets erteilen, kann der Benutzer einen Bucket erstellen. Der Benutzer ist jedoch nicht der Bucket-Eigentümer. Der Bucket gehört zum AWS-Konto , dem der Benutzer angehört. Der Benutzer benötigt eine zusätzliche Berechtigung vom Ressourceneigentümer, um andere Bucket-Operation auszuführen. Weitere Informationen zum Verwalten von Berechtigungen für Ihre Amazon-S3-Ressourcen finden Sie unter Identity and Access Management für Amazon S3.

Verwalten des öffentlichen Zugriffs auf Buckets

Öffentlicher Zugriff auf Buckets und Objekte wird über Bucket-Richtlinien, Zugriffskontrolllisten (ACLs) oder beides gewährt. Um Sie bei der Verwaltung des öffentlichen Zugriffs auf Amazon-S3-Ressourcen zu unterstützen, bietet Amazon S3 Einstellungen für das Sperren des öffentlichen Zugriffs. Die Einstellungen von Amazon S3 Block Public Access können Richtlinien außer Kraft setzen ACLs und so einheitliche Beschränkungen für den öffentlichen Zugriff auf diese Ressourcen durchsetzen. Sie können Block Public Access-Einstellungen auf einzelne Buckets oder auf alle Buckets in Ihrem Konto anwenden.

Um zu gewährleisten, dass bei allen Amazon-S3-Buckets und -Objekten der öffentliche Zugriff blockiert ist, sind alle vier Einstellungen für Block Public Access standardmäßig aktiviert, wenn Sie einen neuen Bucket erstellen. Wir empfehlen, alle vier Einstellungen für Block Public Access auch für Ihr Konto zu aktivieren. Diese Einstellungen blockieren den öffentlichen Zugriff für alle aktuellen und künftigen Buckets.

Bevor Sie diese Einstellungen anwenden, verifizieren Sie, dass Ihre Anwendungen ohne öffentlichen Zugriff korrekt funktionieren. Wenn ein bestimmtes Maß an öffentlichem Zugriff auf Ihre Buckets oder Objekte nötig ist, z. B. zum Hosten einer statischen Website, wie unter Hosten einer

statischen Website mit Amazon S3 beschrieben, können Sie die einzelnen Einstellungen an Ihre Speicheranwendungsfälle anpassen. Weitere Informationen finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

Wir empfehlen jedoch dringend, Block Public Access aktiviert zu lassen. Wenn Sie alle vier Block Public Access-Einstellungen aktiviert lassen und eine statische Website hosten möchten, können Sie Amazon CloudFront Origin Access Control (OAC) verwenden. Amazon CloudFront bietet die Funktionen, die für die Einrichtung einer sicheren statischen Website erforderlich sind. Statische Amazon S3 S3-Websites unterstützen nur HTTP Endpunkte. Amazon CloudFront verwendet den dauerhaften Speicher von Amazon S3 und bietet gleichzeitig zusätzliche Sicherheitsheader, wie HTTPS z. HTTPSerhöht die Sicherheit, indem es eine normale HTTP Anfrage verschlüsselt und vor gängigen Cyberangriffen schützt.

Weitere Informationen finden Sie unter Erste Schritte mit einer sicheren statischen Website im Amazon CloudFront Developer Guide.

> **Note**
>
> Wenn beim Auflisten Ihrer Buckets mit ihren Einstellungen für den öffentlichen Zugriff die Fehlermeldung `Error` angezeigt wird, verfügen Sie möglicherweise nicht über die erforderlichen Berechtigungen. Stellen Sie sicher, dass Sie Ihrer Benutzer- oder Rollenrichtlinie die folgenden Berechtigungen hinzugefügt haben:
>
> ```
> s3:GetAccountPublicAccessBlock
> s3:GetBucketPublicAccessBlock
> s3:GetBucketPolicyStatus
> s3:GetBucketLocation
> s3:GetBucketAcl
> s3:ListAccessPoints
> s3:ListAllMyBuckets
> ```
>
> In einigen seltenen Fällen können Anfragen auch aufgrund des Ausfalls einer AWS-Region fehlschlagen.

Optionen für die Bucket-Konfiguration

Amazon S3 unterstützt Vorgänge Optionen für die Konfiguration Ihres Buckets. Sie können Ihren Bucket beispielsweise für ein Website-Hosting konfigurieren, eine Konfiguration zur Verwaltung

des Lebenszyklus von Objekten im Bucket hinzufügen und den Bucket so konfigurieren, dass alle Zugriffe protokolliert werden. Amazon S3 unterstützt Subressourcen, um die Bucket-Konfigurationsinformationen zu speichern und zu verwalten. Sie können Amazon S3 verwendenAPI, um diese Unterressourcen zu erstellen und zu verwalten. Sie können jedoch auch die Konsole oder die verwenden. AWS SDKs

> **Note**
>
> Außerdem gibt es Konfigurationen auf Objektebene. Sie können beispielsweise Berechtigungen auf Objektebene konfigurieren, indem Sie eine spezifische Zugriffskontrollliste (ACL) für dieses Objekt konfigurieren.

Man bezeichnet sie als Subressourcen, weil sie im Kontext eines spezifischen Buckets oder Objekts existieren. Die folgende Tabelle listet Subressourcen auf, die ihnen ermöglichen, für den Bucket spezifische Konfigurationen zu verwalten.

Subressource	Beschreibung
cors (ursprung sübergreifende gemeinsame Nutzung von Ressourcen)	Sie können Ihren Bucket so konfigurieren, dass er ursprungsübergreifende Anforderungen zulässt. Weitere Informationen finden Sie unter Verwendung von ursprungsübergreifendem Ressourcenaustausch () CORS.
event notification	Sie können Ihrem Bucket gestatten, Benachrichtigungen über bestimmte Bucket-Ereignisse zu senden. Weitere Informationen finden Sie unter Amazon-S3-Ereignis-Benachrichtigungen.
Lebenszyklus	Sie können Lebenszyklusregeln für Objekte in Ihrem Bucket mit definiertem Lebenszyklus definieren. Sie können z. B. eine Regel so festlegen, dass Objekte ein Jahr nach dem Erstellungsdatum archiviert werden, oder dass ein Objekt 10 Jahre nach dem Erstellen gelöscht wird. Weitere Informationen finden Sie unter Verwaltung des Lebenszyklus von Objekten.

Subressource	Beschreibung
location	Wenn Sie einen Bucket erstellen, geben Sie an, AWS-Region wo Amazon S3 den Bucket erstellen soll. Amazon S3 speichert diese Informationen in der Standort-Subressource und bietet Ihnen die MöglichkeitAPI, diese Informati onen abzurufen.
logging	Protokollierung ermöglicht Ihnen, Zugriffsanforderungen für Ihren Bucket nachzuverfolgen. Jeder Zugriffsprotokolldatensatz enthält Details über eine Zugriffsanforderung, z. B. Auftraggeber, Bucket-Name, Anforderungszeit, Anforderungsaktion, Antwortstatus und Fehlercode, falls vorhanden. Die Zugriffsprotokollinformationen können für Sicherheits- und Zugriffsüberprüfun gen nützlich sein. Außerdem erfahren Sie damit mehr über Ihren Kundensta mm und erhalten einen Überblick über Ihre Amazon-S3-Rechnung. Weitere Informationen finden Sie unter Protokollieren von Anfragen mit Server-Zugriffsprotokollierung.
Objektsperre	Zur Verwendung der S3-Objektsperre muss diese für einen Bucket aktiviert werden. Sie können optional auch einen Standardaufbewahrungsmodus und -zeitraum konfigurieren, der für neue Objekte gilt, die in den Bucket platziert werden. Weitere Informationen finden Sie unter Sperren von Objekten mit Object Lock.
Richtlinie und ACL(Zugriffs kontrollliste)	Standardmäßig sind alle Ihre Ressourcen (wie Buckets und Objekte) privat. Amazon S3 unterstützt sowohl die Optionen Bucket Policy als auch Access Control List (ACL), mit denen Sie Berechtigungen auf Bucket-Eb ene gewähren und verwalten können. Amazon S3 speichert die Berechtig ungsinformationen in den Subressourcen policy und acl. Weitere Informationen finden Sie unter Identity and Access Management für Amazon S3.

Subressource	Beschreibung
Replikation	Die Replikation ist ein automatisches, asynchrones Kopieren von Objekten über Buckets hinweg in derselben oder in verschiedenen AWS-Regionen. Weitere Informationen finden Sie unter Objekte innerhalb und zwischen Regionen replizieren.
requestPayment	Standardmäßig zahlt derjenige AWS-Konto , der den Bucket erstellt (der Bucket-Besitzer), für Downloads aus dem Bucket. Mit dieser Subressource kann der Bucket-Eigentümer angeben, dass die Person, die den Download anfordert, die Gebühren für den Download trägt. Amazon S3 bietet Ihnen eine API Möglichkeit, diese Unterressource zu verwalten. Weitere Informationen finden Sie unter Nutzen von Buckets mit Zahlung durch den Anforderer für Speicherübertragungen und Nutzung.
Markieren	Sie können Ihrem Bucket Tags für die Kostenzuweisung hinzufügen, um Ihre Kosten zu kategorisieren und nachzuverfolgen. AWS Amazon S3 unterstützt die tagging-Subressource, um Markierungen für einen Bucket zu speichern und zu verwalten. Mit Hilfe von Markierungen, die Sie auf Ihren Bucket anwenden, erzeugt AWS einen Kostenzuordnungs-Bericht mit Nutzungs- und Kostendaten, aggregiert nach Ihren Markierungen. Weitere Informationen finden Sie unter Abrechnungs- und Nutzungsberichte für Amazon S3.
transfer acceleration	Transfer Acceleration ermöglicht schnelle, einfache und sichere Übertragungen von Dateien zwischen Ihrem Client und einem S3-Bucket über große Entfernungen. Transfer Acceleration nutzt die weltweit verteilten Edge-Standorte von Amazon CloudFront. Weitere Informationen finden Sie unter Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration.

Subressource	Beschreibung
Versioning	Versioning hilft Ihnen bei einer Wiederherstellung nach einem versehentlichen Überschreiben und Löschen. Wir empfehlen das Versioning als bewährte Methode, um zu verhindern, dass Objekte versehentlich gelöscht oder überschrieben werden. Weitere Informationen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.
Website	Sie können Ihren Bucket für ein Hosting statischer Websites konfigurieren. Amazon S3 speichert diese Konfiguration, indem es eine website-Subressource erstellt. Weitere Informationen finden Sie unter Hosten einer statischen Website mit Amazon S3.

Bucket-Operationen

Das Hochverfügbarkeits-Engineering von Amazon S3 konzentriert sich auf get-, put-, list- und delete-Operationen. Da Bucket-Operationen in einem zentralen, globalen Ressourcenbereich ausgeführt werden, empfehlen wir, dass Sie keine Buckets im Hochverfügbarkeitscodepfad Ihrer Anwendung erstellen, löschen oder konfigurieren. Sinnvoller ist es, Buckets in einer separaten Initialisierungs- oder Einrichtungsroutine zu erstellen, zu löschen oder zu konfigurieren, die seltener ausgeführt wird.

Allgemeine Bucket-Muster für die Erstellung von Anwendungen auf Amazon S3

Wenn Sie Anwendungen auf Amazon S3 erstellen, können Sie spezielle Allzweck-Buckets verwenden, um verschiedene Datensätze oder Workloads zu trennen. Wenn Sie Anwendungen für Endbenutzer oder verschiedene Benutzergruppen erstellen, verwenden Sie unsere bewährten Entwurfsmuster, um Anwendungen zu entwickeln, die die Funktionen und Skalierbarkeit von Amazon S3 optimal nutzen können.

> **Important**
>
> Wir empfehlen Ihnen, Bucket-Namen zu erstellen, die nicht vorhersehbar sind. Schreiben Sie keinen Code, vorausgesetzt, der von Ihnen gewählte Bucket-Name ist verfügbar, es sei denn, Sie haben den Bucket bereits erstellt. Eine Methode zum Erstellen von Bucket-Namen, die nicht vorhersehbar sind, besteht darin, Ihrem Bucket-Namen einen Globally Unique Identifier (GUID) anzuhängen, z. B. `amzn-s3-demo-bucket-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111` Weitere Hinweise zu allgemeinen Benennungsregeln für Buckets finden Sie unterAllgemeine Benennungsregeln für Buckets.

Themen

- Bucket-Muster für mehrere Mandanten
- Bucket-per-use Muster

Bucket-Muster für mehrere Mandanten

Bei Buckets mit mehreren Mandanten erstellen Sie einen einzelnen Bucket für ein Team oder eine Arbeitslast. Sie verwenden eindeutige S3-Präfixe, um die Objekte zu organisieren, die Sie im Bucket speichern. Ein Präfix ist eine Zeichenfolge am Anfang des Objektschlüsselnamens. Ein Präfix kann eine beliebige Länge haben, abhängig von der maximalen Länge des Objektschlüsselnamens (1.024 Byte). Sie können sich Präfixe als eine Möglichkeit vorstellen, Ihre Daten ähnlich wie Verzeichnisse zu organisieren. Präfixe sind jedoch keine Verzeichnisse.

Um beispielsweise Informationen über Städte zu speichern, können Sie diese nach Kontinenten, dann nach Ländern, dann nach Provinz oder Bundesland organisieren. Diese Namen enthalten normalerweise keine Interpunktionszeichen, deshalb könnten Sie den Schrägstrich (/) als Trennzeichen verwenden. Die folgenden Beispiele zeigen, wie Präfixe verwendet werden, um Städtenamen mithilfe eines Schrägstrichs (/) nach Kontinent, Land und dann nach Provinz oder Bundesland zu organisieren.

- Europe/France/NouvelleA-Aquitaine/Bordeaux
- Norden America/Canada/Quebec/Montreal
- Norden America/USA/Washington/Bellevue
- Norden America/USA/Washington/Seattle

Dieses Muster lässt sich gut skalieren, wenn Sie Hunderte von eindeutigen Datensätzen in einem Bucket haben. Mit Präfixen können Sie diese Datensätze einfach organisieren und gruppieren.

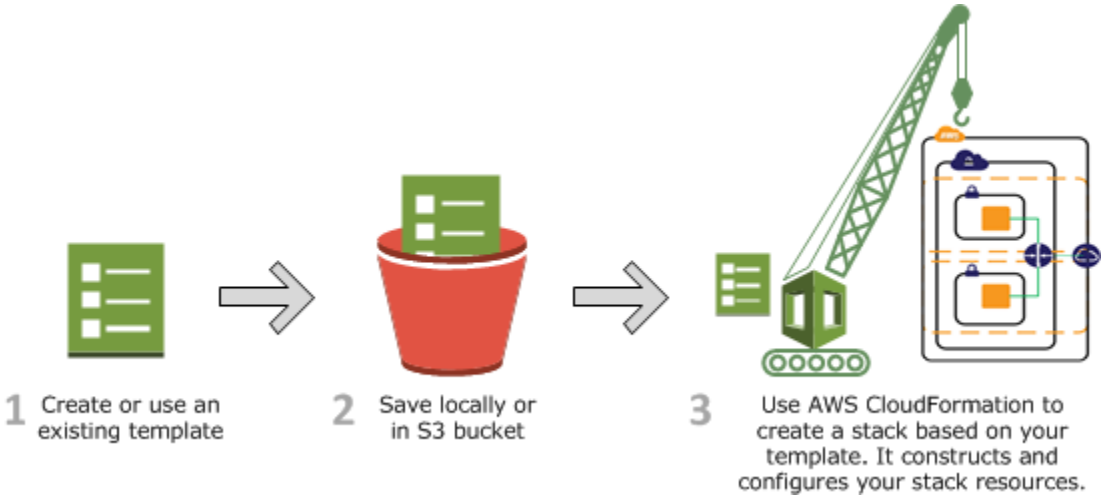
Ein potenzieller Nachteil des Bucket-Musters mit mehreren Mandanten besteht jedoch darin, dass viele Funktionen auf S3-Bucket-Ebene wie Standard-Bucket-Verschlüsselung, S3-Versionierung und S3 Requester Pays auf Bucket-Ebene und nicht auf Präfix-Ebene festgelegt sind. Wenn die verschiedenen Datensätze innerhalb des Multi-Tenant-Buckets unterschiedliche Anforderungen haben, kann die Tatsache, dass Sie nicht viele Funktionen auf S3-Bucket-Ebene auf Präfix-Ebene konfigurieren können, es Ihnen erschweren, die richtigen Einstellungen für jeden Datensatz anzugeben. Darüber hinaus kann die Kostenzuweisung in einem Multi-Tenant-Bucket komplex werden, wenn Sie daran arbeiten, den Speicher, die Anfragen und die Datenübertragung im Zusammenhang mit bestimmten Präfixen zu verstehen.

Bucket-per-use Muster

Mit dem bucket-per-use Muster erstellen Sie einen Bucket für jeden einzelnen Datensatz, jeden Endbenutzer oder jedes Team. Da Sie S3-Funktionen auf Bucket-Ebene für jeden dieser Buckets konfigurieren können, können Sie dieses Muster verwenden, um eindeutige Einstellungen auf Bucket-Ebene zu konfigurieren. Sie können beispielsweise Funktionen wie Standard-Bucket-Verschlüsselung, S3-Versionierung und S3 Requester Pays so konfigurieren, dass sie an den Datensatz in jedem Bucket angepasst sind. Die Verwendung eines Buckets für jeden einzelnen Datensatz, jeden Endbenutzer oder jedes Team kann Ihnen auch dabei helfen, sowohl Ihre Zugriffsverwaltung als auch Ihre Kostenzuweisungsstrategien zu vereinfachen.

Ein potenzieller Nachteil dieser Strategie besteht darin, dass Sie potenziell Tausende von Buckets verwalten müssen. Alle AWS-Konten haben ein Standard-Bucket-Kontingent von 10.000 Buckets für allgemeine Zwecke. Sie können das Bucket-Kontingent für ein Konto erhöhen, indem Sie eine Anfrage zur Erhöhung des Kontingents einreichen. Um eine Erhöhung der Buckets für allgemeine Zwecke zu beantragen, besuchen Sie die Service-Kontingents-Konsole.

Um Ihr bucket-per-use Muster zu verwalten und Ihr Infrastrukturmanagement zu vereinfachen, können Sie verwenden AWS CloudFormation. Sie können eine benutzerdefinierte AWS CloudFormation Vorlage für Ihr Muster erstellen, die bereits alle gewünschten Einstellungen für Ihre S3-Buckets definiert, sodass Sie alle Änderungen an Ihrer Infrastruktur problemlos bereitstellen und nachverfolgen können. Weitere Informationen finden Sie unter AWS: :S3: :Bucket im AWS CloudFormation Benutzerhandbuch.



Allgemeine Benennungsregeln für Buckets

Wenn Sie einen Allzweck-Bucket erstellen, wählen Sie seinen Namen und den, in dem er erstellt werden AWS-Region soll. Nachdem Sie einen Allzweck-Bucket erstellt haben, können Sie dessen Namen oder Region nicht mehr ändern. Die folgenden Abschnitte enthalten Informationen zur Benennung von Buckets für allgemeine Zwecke, darunter Benennungsregeln, bewährte Methoden und ein Beispiel für die Erstellung eines Allzweck-Buckets mit einem Namen, der eine global eindeutige Kennung (GUID) enthält.

Benennungsregeln für Verzeichnis-Buckets finden Sie unterRegeln für die Benennung von Verzeichnis-Buckets.

Informationen zu Objektschlüsselnamen finden Sie unter Erstellen von Objektschlüsselnamen.

Themen

- Regeln für die Benennung von Buckets für allgemeine Zwecke
- Beispiel für Namen von Buckets für allgemeine Zwecke
- Bewährte Methoden
- Einen Bucket erstellen, der ein GUID im Bucket-Namen verwendet

Regeln für die Benennung von Buckets für allgemeine Zwecke

Die folgenden Regeln gelten für die Benennung von Buckets für allgemeine Zwecke.

- Bucket-Namen müssen zwischen (min.) 3 und (max.) 63 Zeichen lang sein.

- Bucket-Namen können nur aus Kleinbuchstaben, Zahlen, Punkten (.) und Bindestrichen (-) bestehen.
- Bucket-Namen müssen mit einem Buchstaben oder einer Zahl beginnen und enden.
- Bucketnamen dürfen keine aufeinander folgenden Punkte (..) enthalten.
- Bucket-Namen dürfen nicht als IP-Adresse formatiert sein (zum Beispiel 192.168.5.4).
- Der Bucket-Name darf nicht mit dem Präfix `xn--` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `sthree-` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `sthree-configurator` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `amzn-s3-demo-` beginnen.
- Bucket-Namen dürfen nicht mit dem Suffix `-s3alias` enden. Dieses Suffix ist für Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter Verwenden eines Alias im Bucket-Stil für Ihren S3-Bucket-Zugriffspunkt.
- Bucket-Namen dürfen nicht mit dem Suffix `--ol-s3` enden. Dieses Suffix ist für Objekt-Lambda-Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter So verwenden Sie einen Alias im Bucket-Stil für den Object Lambda Access Point Ihres S3-Buckets.
- Bucket-Namen dürfen nicht mit dem Suffix `.mrap` enden. Dieses Suffix ist für Namen von Access Points mit mehreren Regionen reserviert. Weitere Informationen finden Sie unter Regeln zur Benennung von Amazon S3-Multi-Regions-Zugriffspunkten.
- Bucket-Namen dürfen nicht mit dem Suffix `--x-s3` enden. Dieses Suffix ist für Verzeichnis-Buckets reserviert. Weitere Informationen finden Sie unter Regeln für die Benennung von Verzeichnis-Buckets.
- Bucket-Namen müssen AWS-Regionen innerhalb einer AWS-Konten Partition überall eindeutig sein. Eine Partition ist eine Gruppierung von Regionen. AWS hat derzeit drei Partitionen: `aws` (Standardregionen), `aws-cn` (China-Regionen) und `aws-us-gov` (AWS GovCloud (US)).
- Ein Bucket-Name kann nicht von einem anderen AWS-Konto in derselben Partition verwendet werden, bis der Bucket gelöscht wurde.
- Buckets, die mit Amazon S3 Transfer Acceleration verwendet werden, können keine Punkte (.) in ihren Namen haben. Weitere Informationen zu Transfer Acceleration finden Sie unter Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration.

Important

- Bucket-Namen müssen AWS-Regionen innerhalb einer Partition AWS-Konten in allen Bereichen eindeutig sein. Eine Partition ist eine Gruppierung von Regionen. AWS verfügt derzeit über drei Partitionen: `aws` (Standardregionen), `aws-cn` (China-Regionen) und `aws-us-gov` (AWS GovCloud (US)).
- Ein Bucket-Name kann nicht von einem anderen AWS-Konto in derselben Partition verwendet werden, bis der Bucket gelöscht wurde. Beachten Sie nach dem Löschen eines Buckets, dass ein anderer Bucket AWS-Konto in derselben Partition denselben Bucket-Namen verwenden kann.

Note

Vor dem 1. März 2018 konnten Buckets, die in der Region USA Ost (Nord-Virginia) erstellt wurden, Namen mit bis zu 255 Zeichen und mit Großbuchstaben und Unterstrichen haben. Ab dem 1. März 2018 müssen neue Buckets in USA Ost (Nord-Virginia) den gleichen Regeln entsprechen, die in allen anderen Regionen angewendet werden.

Beispiel für Namen von Buckets für allgemeine Zwecke

Dies sind Beispiele für gültige Namen von Buckets. Sie befolgen die Empfehlungen für die Benennung von Buckets für allgemeine Zwecke:

- `docexamplebucket-1a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa`
- `amzn-s3-demo-bucket1-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111`
- `amzn-s3-demo-bucket-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222`
- `amzn-s3-demo-bucket2`

Die folgenden Beispiel-Bucket-Namen sind gültig, aber nicht für andere Verwendungszwecke als statisches Website-Hosting empfohlen:

- `example.com`
- `www.example.com`

- `my.example.s3.bucket`

Die folgenden Beispiel-Bucket-Namen sind ungültig:

- `amzn_s3_demo_bucket` (enthält Unterstriche)
- `AmznS3DemoBucket` (enthält Großbuchstaben)
- `amzn-s3-demo-bucket-` (endet mit einem Bindestrich)

Bewährte Methoden

Beachten Sie bei der Benennung Ihrer Buckets die folgenden bewährten Methoden zur Benennung von Buckets.

Wählen Sie ein Benennungsschema für Buckets, bei dem es unwahrscheinlich ist, dass es zu Namenskonflikten kommt

Wenn Ihre Anwendung automatisch Buckets erstellt, wählen Sie ein Bucket-Namensschema, das wahrscheinlich keine Namenskonflikte verursacht. Stellen Sie sicher, dass Ihre Anwendungslogik einen anderen Bucket-Namen auswählt, wenn ein Bucket-Name bereits vergeben ist.

Hängen Sie global eindeutige Bezeichner (GUID) an Bucket-Namen an

Wir empfehlen, Bucket-Namen zu erstellen, die nicht vorhersehbar sind. Schreiben Sie keinen Code, vorausgesetzt, der von Ihnen gewählte Bucket-Name ist verfügbar, es sei denn, Sie haben den Bucket bereits erstellt. Eine Methode zum Erstellen von Bucket-Namen, die nicht vorhersehbar sind, besteht darin, Ihrem Bucket-Namen einen Globally Unique Identifier (GUID) anzuhängen, z. B. `amzn-s3-demo-bucket-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111`

Vermeiden Sie die Verwendung von Punkten (.) in Bucket-Namen

Aus Gründen der besten Kompatibilität empfehlen wir, Punkte (.) in Bucket-Namen zu vermeiden, mit Ausnahme von Buckets, die nur für statisches Website-Hosting verwendet werden. Wenn Sie Punkte in den Namen eines Buckets aufnehmen, können Sie virtual-host-style Adressierung nicht verwendenHTTPS, es sei denn, Sie führen Ihre eigene Zertifikatsvalidierung durch. Dies liegt daran, dass die Sicherheitszertifikate, die für das virtuelle Hosten von Buckets verwendet werden, nicht für Buckets mit Punkten in ihren Namen funktionieren.

Diese Einschränkung wirkt sich nicht auf Buckets aus, die für statisches Website-Hosting verwendet werden, da statisches Website-Hosting nur über HTTP verfügbar ist. Weitere Informationen zur

virtual-host-style Adressierung finden Sie unterVirtuelles Hosting bei Buckets. Weitere Hinweise zum Hosten statischer Websites finden Sie unter Hosten einer statischen Website mit Amazon S3.

Wählen Sie einen relevanten Namen

Wenn Sie einem Bucket einen Namen geben, empfehlen wir Ihnen, einen Namen zu wählen, der für Sie oder Ihr Unternehmen relevant ist. Vermeiden Sie die Verwendung von Namen, die mit anderen Entitäten verbunden sind. Zum Beispiel sollten Sie es vermeiden, `AWS` oder `Amazon` in Ihrem Bucket-Namen zu verwenden.

Wiederverwenden von Bucket-Namen

Wenn ein Bucket leer ist, können Sie ihn löschen. Nachdem ein Bucket gelöscht wurde, wird der Name zur Wiederverwendung verfügbar. Nachdem Sie den Bucket gelöscht haben, können Sie den Namen aus verschiedenen Gründen möglicherweise jedoch nicht wiederverwenden.

Wenn Sie beispielsweise den Bucket löschen und der Name für die Wiederverwendung verfügbar ist, AWS-Konto kann ein anderer Benutzer einen Bucket mit diesem Namen erstellen. Außerdem kann einige Zeit vergehen, bis Sie den Namen eines gelöschten Buckets wiederverwenden können. Wenn Sie denselben Bucket-Namen verwenden möchten, empfehlen wir, den Bucket nicht zu löschen.

Einen Bucket erstellen, der ein GUID im Bucket-Namen verwendet

Die folgenden Beispiele zeigen, wie Sie einen Allzweck-Bucket erstellen, der GUID am Ende des Bucket-Namens ein verwendet.

Mit dem AWS CLI

Im folgenden AWS CLI Beispiel wird ein Bucket in der Region USA West (Nordkalifornien) (us-west-1) mit einem Beispiel-Bucket-Namen erstellt, der einen global eindeutigen Bezeichner (GUID) verwendet.

```
aws s3api create-bucket \
    --bucket amzn-s3-demo-bucket1$(uuidgen | tr -d - | tr '[:upper:]' '[:lower:]' ) \
    --region us-west-1 \
    --create-bucket-configuration LocationConstraint=us-west-1
```

Verwendung des AWS SDK for Java

Das folgende Beispiel zeigt Ihnen, wie Sie einen Bucket mit einem GUID am Ende des Bucket-Namens in der Region USA Ost (Nord-Virginia) (us-east-1) erstellen, indem Sie den verwenden. AWS

SDK for Java Weitere Informationen zu anderen finden Sie unter Tools AWS SDKs, auf denen Sie aufbauen können. AWS

```
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.Bucket;
import com.amazonaws.services.s3.model.CreateBucketRequest;

import java.util.List;
import java.util.UUID;

public class CreateBucketWithUUID {
    public static void main(String[] args) {
        final AmazonS3 s3 =
 AmazonS3ClientBuilder.standard().withRegion(Regions.US_EAST_1).build();
        String bucketName = "amzn-s3-demo-bucket" +
 UUID.randomUUID().toString().replace("-", "");
        CreateBucketRequest createRequest = new CreateBucketRequest(bucketName);
        System.out.println(bucketName);
        s3.createBucket(createRequest);
    }
}
```

Bucket-Kontingente, Beschränkungen und Einschränkungen

Ein Amazon S3 S3-Bucket gehört demjenigen AWS-Konto , der ihn erstellt hat. Die Bucket-Eigentümerschaft ist nicht auf ein anderes Konto übertragbar.

Bucket-Kontingente

Standardmäßig können Sie bis zu 10.000 Allzweck-Buckets pro AWS-Konto Bucket erstellen. Besuchen Sie die Konsole Service Quotas, um eine Erhöhung des Kontingents für allgemeine Zwecke zu beantragen.

> **Important**
>
> Wir empfehlen dringend, nur paginierte `ListBuckets` Anfragen zu verwenden. `ListBuckets`Anfragen ohne Seiten werden nur unterstützt, wenn sie auf das standardmäßige allgemeine Bucket-Kontingent von 10.000 AWS-Konten festgelegt sind.

Wenn Sie ein genehmigtes Kontingent für allgemeine Buckets von mehr als 10.000 haben, müssen Sie paginierte `ListBuckets` Anfragen senden, um die Buckets Ihres Kontos aufzulisten. Alle `ListBuckets` Anfragen ohne Seitennummerierung werden abgelehnt, wenn das Kontingent für AWS-Konten allgemeine Zwecke mehr als 10.000 beträgt.

Note

Sie müssen Folgendes verwenden, AWS-Regionen um Ihr Kontingent und Ihre Bucket-Auslastung einzusehen oder eine Erhöhung Ihrer allgemeinen Buckets in Ihrem zu beantragen. AWS-Konto

- Allgemeine Bucket-Kontingente für kommerzielle Regionen können nur von USA Ost (Nord-Virginia) aus eingesehen und verwaltet werden.
- Allgemeine Bucket-Kontingente für AWS GovCloud (US) können nur von AWS GovCloud (USA West) aus angezeigt und verwaltet werden.

Informationen zu Servicekontingenten finden Sie unter AWS Servicekontingenten im Allgemeine Amazon Web Services-Referenz.

Einschränkungen für Objekte und Buckets

Die maximale Bucket-Größe oder die Anzahl der Objekte, die Sie in einem Bucket speichern können, ist nicht begrenzt. Sie können alle Ihre Objekte in einem einzigen Bucket speichern, oder sie über mehrere Buckets verteilen. Sie können jedoch keinen Bucket über einen anderen Bucket erstellen.

Regeln für die Benennung von Buckets

Wenn Sie einen Bucket erstellen, wählen Sie seinen Namen und den Ort, in AWS-Region dem er erstellt werden soll. Name oder Region einmal erstellter Buckets können nicht nachträglich geändert werden. Weitere Informationen zur Benennung von Buckets finden Sie unter Allgemeine Benennungsregeln für Buckets.

Zugreifen auf einen Amazon S3 S3-Bucket

Sie können über die Amazon S3-Konsole,, AWS Command Line Interface oder Amazon S3 auf Ihre Amazon S3 REST API S3-Buckets zugreifen. AWS SDKs Jede Methode für den Zugriff auf einen S3-

Bucket unterstützt bestimmte Anwendungsfälle. Weitere Informationen finden Sie in den folgenden Abschnitten.

Themen

- Anwendungsfälle
- Amazon S3-Konsole
- AWS CLI
- AWS SDKs
- Amazon S3 REST API

Anwendungsfälle

Je nach Anwendungsfall des Amazon-S3-Buckets werden verschiedene Methoden für den Zugriff auf die zugrunde liegenden Daten in den Buckets empfohlen. Die folgende Liste enthält gängige Anwendungsfälle für den Zugriff auf Daten.

- Statische Websites – Sie können in Amazon S3 eine statische Website hosten. Bei diesem Anwendungsfall lässt sich der S3-Bucket so konfigurieren, dass er sich wie eine Website verhält. Ein Beispiel für Schritte zum Hosten einer Website auf Amazon S3 finden Sie unter Tutorial: Konfigurieren einer statischen Website auf Amazon S3.

  Um eine statische Website mit aktivierten Sicherheitseinstellungen wie Block Public Access zu hosten, empfehlen wir, Amazon CloudFront mit Origin Access Control (OAC) zu verwenden und zusätzliche Sicherheitsheader zu implementieren, wie HTTPS z. Weitere Informationen finden Sie unter Erste Schritte mit einer sicheren statischen Website.

  > **Note**
  >
  > Amazon S3 unterstützt sowohl den virtuellen Hosting-Stil als auch den URLsPfadstil für den statischen Zugriff auf Websites. Da auf Buckets im Pfadstil und im URLs Virtual-Hosted-Stil zugegriffen werden kann, empfehlen wir, Buckets mit -konformen Bucket-Namen zu erstellen. DNS Weitere Informationen finden Sie unter Bucket-Kontingente, Beschränkungen und Einschränkungen.

- Freigegebene Datensätze – Bei der Skalierung auf Amazon S3 wird üblicherweise ein Multi-Tenant-Modell verwendet, bei dem Sie verschiedenen Endkunden oder Geschäftseinheiten eindeutige Präfixe innerhalb eines freigegebenen Buckets zuweisen. Durch die Verwendung

von Amazon S3 Access Points können Sie eine umfangreiche Bucket-Richtlinie in separate, diskrete Zugangspunkt-Richtlinien für jede Anwendung unterteilen, die auf den freigegebenen Datensatz zugreifen muss. Mit diesem Ansatz ist es einfacher, sich auf die Erstellung der richtigen Zugriffsrichtlinie für eine Anwendung zu konzentrieren, ohne die Aktivitäten anderer Anwendungen innerhalb des freigegebenen Datensatzes zu beeinflussen. Weitere Informationen finden Sie unter Verwaltung des Zugriffs auf gemeinsam genutzte Datensätze mit Access Points.

- Workload mit hohem Durchsatz – Mountpoint für Amazon S3 ist ein Open-Source-Dateiclient mit hohem Durchsatz zum Mounten eines Amazon-S3-Bucket als lokales Dateisystem. Mit Mountpoint können Anwendungen über Dateisystem-Operationen wie Öffnen und Lesen auf Objekte zugreifen, die in Amazon S3 gespeichert sind. Mountpoint übersetzt diese Operationen automatisch in API S3-Objekt-Aufrufe, sodass Ihre Anwendungen über eine Dateischnittstelle auf den elastischen Speicher und Durchsatz von Amazon S3 zugreifen können. Weitere Informationen finden Sie unter Arbeiten mit Mountpoint für Amazon S3.
- Multi-Regions-Anwendungen – Amazon S3 Multi-Region Access Points stellen einen globalen Endpunkt bereit, mit dem Anwendungen Anforderungen von S3-Buckets ausführen können, die sich in mehreren AWS-Regionen-Regionen befinden. Sie können Multi-Regions Access Points verwenden, um Multi-Regions-Anwendungen mit derselben Architektur zu erstellen, die in einer einzelnen Region verwendet wird, und diese Anwendungen dann überall auf der Welt ausführen. Anstatt Anforderungen über das öffentliche Internet zu senden, bieten Multi-Region Access Points integrierte Netzwerkausfallsicherheit mit Beschleunigung internetbasierter Anforderungen an Amazon S3. Weitere Informationen finden Sie unter Verwaltung von regionsübergreifendem Verkehr mit multiregionalen Access Points.
- Secure Shell (SSH) File Transfer Protocol (SFTP) — Wenn Sie versuchen, vertrauliche Daten sicher über das Internet zu übertragen, können Sie einen SFTP -fähigen Server mit Ihrem Amazon S3 S3-Bucket verwenden. AWS SFTP ist ein Netzwerkprotokoll, das die volle Sicherheits- und Authentifizierungsfunktionalität von SSH unterstützt. Mit diesem Protokoll haben Sie eine genaue Kontrolle über Benutzeridentität, Berechtigungen und Schlüssel oder Sie können IAM Richtlinien verwenden, um den Zugriff zu verwalten. Um Ihrem Amazon S3 S3-Bucket einen SFTP aktivierten Server zuzuordnen, stellen Sie sicher, dass Sie zuerst Ihren SFTP -fähigen Server erstellen. Anschließend richten Sie Benutzerkonten ein und verknüpfen den Server mit einem Amazon-S3-Bucket. Eine exemplarische Vorgehensweise für diesen Prozess finden Sie unter AWS Transfer for SFTP — Fully Managed SFTP Service für Amazon S3 in AWS Blogs.

Amazon S3-Konsole

Die Konsole ist eine webbasierte Benutzeroberfläche für die Verwaltung von Amazon S3 und AWS Ressourcen. Mit der Amazon S3 S3-Konsole können Sie einfach auf einen Bucket zugreifen und dessen Eigenschaften ändern. Mit der Benutzeroberfläche der Konsole können Sie zudem fast alle Bucket-Operationen ausführen, ohne Code schreiben zu müssen.

Wenn Sie sich für eine angemeldet haben AWS-Konto, können Sie auf die Amazon S3 S3-Konsole zugreifen, indem Sie sich bei der Amazon S3-Konsole anmelden und auf der Startseite der Amazon S3-Konsole S3 auswählen. Sie können diesen Link auch verwenden, um direkt auf die zuzugreifen https://console.aws.amazon.com/s3/.

AWS CLI

Sie können den verwenden AWS CLI , um Befehle auszugeben oder Skripts in der Befehlszeile Ihres Systems zu erstellen, um Aufgaben AWS (einschließlich S3) auszuführen. Wenn Sie beispielsweise auf mehrere Buckets zugreifen müssen, können Sie Zeit sparen, indem Sie den verwenden, AWS CLI um allgemeine und sich wiederholende Aufgaben zu automatisieren. Die Skriptfähigkeit und die Wiederholbarkeit gängiger Aktionen sind wichtige Aspekte, die bei der Skalierung von Unternehmen zu berücksichtigen sind.

Das AWS CLIbietet Befehle für eine Vielzahl von. AWS-Services Das AWS CLI wird unter Windows, MacOS und Linux unterstützt. Informationen zu den ersten Schritten finden Sie im AWS Command Line Interface -Benutzerhandbuch. Weitere Informationen über die Befehle für Amazon S3 finden Sie unter s3api und s3control in der AWS CLI -Befehlsreferenz.

AWS SDKs

AWS stellt SDKs (Software Development Kits) bereit, die aus Bibliotheken und Beispielcode für verschiedene Programmiersprachen und Plattformen bestehen (Java, Python, Ruby,. NET, iOS, Android usw.). AWS SDKsSie bieten eine bequeme Möglichkeit, programmatischen Zugriff auf S3 und AWS zu erstellen. Amazon S3 ist ein REST Service. Sie können Anfragen mithilfe der AWS SDK Bibliotheken an Amazon S3 senden, die das zugrunde liegende Amazon S3 einschließen REST API und Ihre Programmieraufgaben vereinfachen. SDKsSie kümmern sich beispielsweise um Aufgaben wie das Berechnen von Signaturen, das kryptografische Signieren von Anfragen, das Verwalten von Fehlern und das automatische Wiederholen von Anfragen. Informationen zu den AWS SDKs, einschließlich deren Herunterladen und Installieren, finden Sie unter Tools für. AWS

Jede Interaktion mit Amazon S3 erfolgt entweder authentifiziert oder anonym. Wenn Sie die verwenden AWS SDKs, berechnen die Bibliotheken die Signatur für die Authentifizierung anhand der

von Ihnen bereitgestellten Schlüssel. Weitere Informationen darüber, wie Sie Anfragen an Amazon S3 stellen können, finden Sie unter Anfragen stellen.

Amazon S3 REST API

Die Amazon S3-Architektur ist so ausgelegt, dass sie unabhängig von Programmiersprachen ist und unsere von AWS unterstützten Schnittstellen verwendet, um Objekte zu speichern und abzurufen. Sie können mithilfe von Amazon S3 und AWS programmatisch auf S3 zugreifen. REST API Das REST API ist eine HTTP Schnittstelle zu Amazon S3. Mit dem verwenden Sie HTTP Standardanforderungen RESTAPI, um Buckets und Objekte zu erstellen, abzurufen und zu löschen.

Um das zu verwenden RESTAPI, können Sie jedes Toolkit verwenden, das dies unterstützt. HTTP Sie können sogar einen Browser verwenden, um Objekte zu laden, wenn diese anonym lesbar sind.

Das REST API verwendet HTTP Standardheader und Statuscodes, sodass Standardbrowser und Toolkits erwartungsgemäß funktionieren. In einigen Bereichen haben wir Funktionen hinzugefügt HTTP (zum Beispiel haben wir Header hinzugefügt, um die Zugriffskontrolle zu unterstützen). In diesen Fällen haben wir unser Bestes getan, um die neuen Funktionen so hinzuzufügen, dass sie dem Stil der HTTP Standardnutzung entsprechen.

Wenn Sie in Ihrer Anwendung direkte REST API Anrufe tätigen, müssen Sie den Code zur Berechnung der Signatur schreiben und der Anfrage hinzufügen. Weitere Informationen darüber, wie Sie Anfragen an Amazon S3 stellen, finden Sie unter Anfragen stellen in der Amazon S3 API S3-Referenz.

Virtuelles Hosting bei Buckets

Das virtuelle Hosting ist das beste Verfahren, um mehrere Websites mit einem Webserver zu unterstützen. Eine Möglichkeit, Websites in Ihren Amazon S3 REST API S3-Anfragen zu unterscheiden, besteht darin, den offensichtlichen Hostnamen der Anfrage zu verwenden — URI statt nur den Pfadnamen, der URI Teil von ist. Eine normale Amazon S3 REST S3-Anfrage spezifiziert einen Bucket, indem sie die erste durch Schrägstriche getrennte Komponente des Request-Pfads verwendet. URI Stattdessen können Sie das virtuelle Amazon S3 S3-Hosting verwenden, um einen Bucket in einem REST API Aufruf mithilfe des HTTP `Host` Headers zu adressieren. In der Praxis interpretiert Amazon S3 `Host` so, dass die meisten Buckets automatisch (für begrenzte Anfragetypen) unter `https://`*`bucket-name`*`.s3.`*`region-code`*`.amazonaws.com` zur Verfügung stehen. Eine vollständige Liste der Amazon-S3-Regionen und -Endpunkte finden Sie unter Amazon-S3-Endpunkte und -Kontingente in der Allgemeine Amazon Web Services-Referenz.

Virtuelles Hosting hat auch andere Vorteile. Indem Sie Ihren Bucket nach Ihrem registrierten Domainnamen benennen und diesen Namen zu einem DNS Alias für Amazon S3 machen, können Sie Ihre Amazon S3 S3-Ressourcen vollständig anpassen, `http://my.bucket-name.com/` z. B. URL Sie können auch im „Stammverzeichnis“ des virtuellen Servers Ihres Buckets veröffentlichen. Diese Fähigkeit kann wichtig sein, da viele vorhandene Anwendungen nach Dateien an diesem Standardspeicherort suchen. Beispielsweise wird erwartet, dass `favicon.ico`, `robots.txt` und `crossdomain.xml` im Stamm gefunden werden können.

> **Important**
>
> Wenn Sie Buckets im Virtual-Hosting-Stil mit verwenden, stimmt das SSL Platzhalterzertifikat nur mit Buckets übereinSSL, die keine Punkte () enthalten. `.` Um diese Einschränkung zu umgehen, verwenden oder schreiben Sie Ihre eigene Logik zur Zertifikatsverifizierung. HTTP Weitere Informationen finden Sie unter Amazon S3 Path Deprecation Plan im AWS News Blog.

Themen

- Anforderungen im Pfadformat
- Anforderungen im virtuellen Hosting-Format
- HTTPHostHeader-Bucket-Spezifikation
- Beispiele
- Amazon S3 URLs mit CNAME Datensätzen anpassen
- Verknüpfen eines Host-Namens mit einem Amazon-S3-Bucket
- Einschränkungen
- Abwärtskompatibilität

Anforderungen im Pfadformat

Derzeit unterstützt Amazon S3 sowohl virtuell gehosteten Zugriff als auch Pfadzugriff. URL AWS-Regionen Path-Style URLs wird jedoch in future eingestellt. Weitere Informationen finden Sie im folgenden wichtigen Hinweis.

In Amazon S3 URLs verwendet path-style das folgende Format:

```
https://s3.region-code.amazonaws.com/bucket-name/key-name
```

Wenn Sie beispielsweise einen Bucket mit dem Namen `amzn-s3-demo-bucket1` USA West (Oregon) erstellen und auf das `puppy.jpg` Objekt in diesem Bucket zugreifen möchten, können Sie den folgenden Pfadstil verwenden: URL

```
https://s3.us-west-2.amazonaws.com/amzn-s3-demo-bucket1/puppy.jpg
```

Important

Update (23. September 2020) — Um sicherzustellen, dass Kunden die Zeit haben, die sie für die Umstellung auf virtuell gehostete Systeme benötigen, haben wir beschlossenURLs, die Einstellung von Path-Style zu verzögern. URLs Weitere Informationen finden Sie unter Amazon S3 Path Deprecation Plan – The Rest of the Story im AWS News Blog.

Warning

Vermeiden Sie beim Hosten von Website-Inhalten, auf die über einen Webbrowser zugegriffen wird, die Verwendung des Pfadformats, da dies das Sicherheitsmodell des Browsers „Selbe Origin“ beeinträchtigen URLs könnte. Zum Hosten von Website-Inhalten empfehlen wir, entweder S3-Website-Endpunkte oder eine Distribution zu verwenden. CloudFront Weitere Informationen finden Sie unter Website-Endpunkte und Bereitstellen einer React-basierten Einzelseitenanwendung für Amazon S3 und CloudFront in den AWS Perspective Guidance Patterns.

Anforderungen im virtuellen Hosting-Format

Bei einem virtuell gehosteten System ist der Bucket-Name URI Teil des Domainnamens in der. URL

Verwenden Sie im virtuellen Hosted-Stil von Amazon S3 das folgende Format: URLs

```
https://bucket-name.s3.region-code.amazonaws.com/key-name
```

In diesem Beispiel ist `amzn-s3-demo-bucket1` der Bucket-Name, "US West (Oregon) (USA West (Oregon))" die Region und `puppy.png` der Schlüsselname:

```
https://amzn-s3-demo-bucket1.s3.us-west-2.amazonaws.com/puppy.png
```

HTTP`Host`Header-Bucket-Spezifikation

Solange Ihre `GET` Anfrage den SSL Endpunkt nicht verwendet, können Sie den Bucket für die Anfrage mithilfe des HTTP `Host` Headers angeben. Der `Host` Header in einer REST Anfrage wird wie folgt interpretiert:

- Wenn der `Host` Header weggelassen wird oder sein Wert ists`3.`*`region-code`*`.amazonaws.com`, ist der Bucket für die Anfrage die erste durch Schrägstriche getrennte Komponente der Anfrage-URI und der Schlüssel für die Anfrage ist der Rest der Anfrage-. URI Dies ist die übliche Methode wie im ersten und zweiten Beispiel in diesem Abschnitt dargestellt. Das Weglassen des `Host` Headers ist nur für 1.0-Anfragen gültig. HTTP
- Wenn andernfalls der Wert des `Host`-Headers mit `.s3.`*`region-code`*`.amazonaws.com` endet, ist der Bucket-Name die führende Komponente im Wert des `Host`-Headers bis zu `.s3.`*`region-code`*`.amazonaws.com`. Der Schlüssel für die Anfrage ist Request-. URI Diese Interpretation stellt Buckets als Unterdomänen von `.s3.`*`region-code`*`.amazonaws.com` bereit (vgl. das dritte und vierte Beispiel in diesem Abschnitt).
- Andernfalls ist der Bucket für die Anfrage der Kleinbuchstabenwert des `Host` Headers, und der Schlüssel für die Anfrage ist der Request-URI. Diese Interpretation ist nützlich, wenn Sie denselben DNS Namen wie Ihren Bucket-Namen registriert und diesen Namen als kanonischen Namen (CNAME) -Alias für Amazon S3 konfiguriert haben. Das Verfahren zur Registrierung von Domainnamen und zur Konfiguration von CNAME DNS Datensätzen würde den Rahmen dieses Handbuchs sprengen. Das Ergebnis wird jedoch anhand des letzten Beispiels in diesem Abschnitt veranschaulicht.

Beispiele

Dieser Abschnitt enthält Beispiele URLs und Anfragen.

Example — Pfadstil URLs und Anfragen

In diesem Beispiel wird Folgendes verwendet:

- Bucket-Name – `example.com`
- Region: USA Ost (Nord-Virginia)
- Schlüsselname – `homepage.html`

Das URL ist wie folgt:

```
http://s3.us-east-1.amazonaws.com/example.com/homepage.html
```

die Anforderung lautet wie folgt:

```
GET /example.com/homepage.html HTTP/1.1
Host: s3.us-east-1.amazonaws.com
```

Die Anfrage mit HTTP 1.0 und dem Weglassen des `Host` Headers lautet wie folgt:

```
GET /example.com/homepage.html HTTP/1.0
```

Informationen zu DNS -kompatiblen Namen finden Sie unter Einschränkungen. Weitere Informationen zu Schlüsseln finden Sie unter Schlüssel.

Example — URLs Virtual-Hosted — Stil und Anfragen

In diesem Beispiel wird Folgendes verwendet:

- Bucket-Name – `amzn-s3-demo-bucket1`
- Region - Europa (Irland)
- Schlüsselname – `homepage.html`

Das ist wie folgt: URL

```
http://amzn-s3-demo-bucket1.s3.eu-west-1.amazonaws.com/homepage.html
```

die Anforderung lautet wie folgt:

```
GET /homepage.html HTTP/1.1
Host: amzn-s3-demo-bucket1.s3.eu-west-1.amazonaws.com
```

Example — CNAME Alias-Methode

Um diese Methode zu verwenden, müssen Sie Ihren DNS Namen als CNAME Alias für konfigurieren*`bucket-name`*`.s3.us-east-1.amazonaws.com`. Weitere Informationen finden Sie unter Amazon S3 URLs mit CNAME Datensätzen anpassen.

In diesem Beispiel wird Folgendes verwendet:

- Bucket-Name – `example.com`
- Schlüsselname – `homepage.html`

Das URL ist wie folgt:

```
http://www.example.com/homepage.html
```

Das Beispiel ist wie folgt:

```
GET /homepage.html HTTP/1.1
Host: www.example.com
```

Amazon S3 URLs mit CNAME Datensätzen anpassen

Abhängig von den Anforderungen können Sie die Anzeige von `s3.`*`region-code`*`.amazonaws.com` auf der Website oder im Service gegebenenfalls unterbinden. Wenn Sie beispielsweise Websiteabbilder auf Amazon S3 hosten, bevorzugen Sie möglicherweise `http://images.example.com/` anstelle von `http://images.example.com.s3.us-east-1.amazonaws.com/`. Jeder Bucket mit einem DNS -kompatiblen Namen kann wie folgt referenziert werden: zum `http://`*`BucketName`*`.s3.`*`Region`*`.amazonaws.com/[`*`Filename`*`]` Beispiel. `http://images.example.com.s3.us-east-1.amazonaws.com/mydog.jpg` Durch die Verwendung CNAME können Sie einem Amazon S3 S3-Hostnamen zuordnen, sodass der vorherige zu werden URL `http://images.example.com/mydog.jpg` könnte. `images.example.com`

Ihr Bucket-Name muss mit dem CNAME identisch sein. Wenn Sie beispielsweise eine Zuordnung CNAME `images.example.com` zu erstellen`images.example.com.s3.us-east-1.amazonaws.com`, `http://images.example.com.s3.us-east-1.amazonaws.com/filename` werden beide `http://images.example.com/filename` und identisch sein.

Der CNAME DNS Datensatz sollte Ihren Domainnamen mit dem entsprechenden Hostnamen im virtuellen Hosting-Stil verknüpfen. Wenn sich Ihr Bucket-Name und Ihr Domainname beispielsweise in der Region USA Ost (Nord-Virginia) befinden `images.example.com` und Ihr Bucket sich in der Region USA Ost (Nord-Virginia) befindet, sollte der CNAME Datensatz als Alias lauten. `images.example.com.s3.us-east-1.amazonaws.com`

```
images.example.com CNAME    images.example.com.s3.us-east-1.amazonaws.com.
```

Amazon S3 verwendet den Host-Namen, um den Bucket-Namen zu ermitteln. Der Bucket-Name CNAME und der Bucket-Name müssen also identisch sein. Nehmen wir zum Beispiel an, dass Sie `www.example.com` als CNAME für konfiguriert habenwww.example.com.s3.us-east-1.amazonaws.com. Wenn Sie auf `http://www.example.com` zugreifen, empfängt Amazon S3 eine Anforderung, die in etwa wie folgt aussieht:

Example

```
GET / HTTP/1.1
Host: www.example.com
Date: date
Authorization: signatureValue
```

Amazon S3 sieht nur den ursprünglichen Hostnamen `www.example.com` und weiß nichts von der CNAME Zuordnung, die zur Lösung der Anfrage verwendet wurde.

Sie können jeden Amazon S3 S3-Endpunkt in einem CNAME Alias verwenden. `s3.ap-southeast-1.amazonaws.com`Kann beispielsweise in CNAME Aliasen verwendet werden. Weitere Informationen zu Endpunkten finden Sie unter Endpunkte anfordern in der Amazon S3 API S3-Referenz. Informationen zum Erstellen einer statischen Website mit einer benutzerdefinierten Domäne finden Sie unter Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne.

> **Important**
>
> Wenn Sie custom URLs with verwendenCNAMEs, müssen Sie sicherstellen, dass für jeden von Ihnen konfigurierten Datensatz CNAME oder Alias-Datensatz ein passender Bucket vorhanden ist. Wenn Sie beispielsweise DNS Einträge für Webinhalte erstellen `www.example.com` und diese mit S3 veröffentlichen `login.example.com` möchten, müssen Sie sowohl Buckets `www.example.com` als auch erstellen. `login.example.com` Wenn ein Aliaseintrag CNAME oder ein Alias konfiguriert ist, der auf einen S3-Endpunkt ohne passenden Bucket verweist, kann jeder AWS Benutzer diesen Bucket erstellen und Inhalte unter dem konfigurierten Alias veröffentlichen, auch wenn die Eigentümerschaft nicht identisch ist.
>
> Aus demselben Grund empfehlen wir, dass Sie beim Löschen eines Buckets den entsprechenden Alias CNAME oder Alias ändern oder entfernen.

Verknüpfen eines Host-Namens mit einem Amazon-S3-Bucket

So verknüpfen Sie mithilfe eines Alias einen Hostnamen mit einem Amazon S3-Bucket CNAME

1. Wählen Sie einen Host-Namen aus, der zu einer von Ihnen kontrollierten Domäne gehört.

   Dieses Beispiel verwendet die Unterdomäne `images` der Domäne `example.com`.
2. Erstellen Sie einen Bucket, der dem Host-Namen entspricht.

   In diesem Beispiel lauten der Host- und der Bucket-Name `images.example.com`. Der Bucket-Name muss exakt mit dem Host-Namen übereinstimmen.
3. Erstellen Sie einen CNAME DNS Datensatz, der den Hostnamen als Alias für den Amazon S3 S3-Bucket definiert.

   Beispielsweise:

   ```
   images.example.com CNAME images.example.com.s3.us-west-2.amazonaws.com
   ```

   > **Important**
   >
   > Aus Gründen der Weiterleitung von Anfragen muss der CNAME DNS Datensatz genau so definiert werden, wie im vorherigen Beispiel gezeigt. Andernfalls kann sich trotz richtig erscheinender Funktion unvorhersehbares Verhalten ergeben.

   Das Verfahren zur Konfiguration von CNAME DNS Datensätzen hängt von Ihrem DNS Server oder DNS Anbieter ab. Spezifische Informationen finden Sie in Ihrer Serverdokumentation oder erhalten Sie von Ihrem Anbieter.

Einschränkungen

SOAPSupport Over HTTP ist veraltet, aber SOAP immer noch verfügbar. HTTPS Neue Amazon S3 S3-Funktionen werden nicht unterstütztSOAP. Anstatt zu verwendenSOAP, empfehlen wir, entweder den REST API oder den zu verwenden AWS SDKs.

Abwärtskompatibilität

In den folgenden Abschnitten werden verschiedene Aspekte der Amazon S3 S3-Abwärtskompatibilität behandelt, die sich auf Anfragen im Pfadstil und auf virtuell gehostete Anfragen beziehen. URL

Legacy-Endpunkte

Einige Regionen unterstützen Legacy-Endpunkte. Möglicherweise finden Sie diese Endpunkte in Ihren Serverzugriffsprotokollen oder -protokollen. AWS CloudTrail Weitere Informationen finden Sie im folgenden Abschnitt. Eine vollständige Liste der Amazon-S3-Regionen und -Endpunkte finden Sie unter Endpunkte und Kontingente von Amazon S3 in der Allgemeine Amazon Web Services-Referenz.

Important

Obwohl Sie möglicherweise Legacy-Endpunkte in Ihren Protokollen sehen, empfehlen wir Ihnen, immer die Standardendpunktsyntax für den Zugriff auf Ihre Buckets zu verwenden. Verwenden Sie im virtuellen Hosted-Stil von Amazon S3 das folgende Format: URLs

```
https://bucket-name.s3.region-code.amazonaws.com/key-name
```

In Amazon S3 URLs verwendet path-style das folgende Format:

```
https://s3.region-code.amazonaws.com/bucket-name/key-name
```

S3-Region

Einige ältere Amazon-S3-Regionen unterstützen Endpunkte, die einen Bindestrich (`-`) zwischen `s3` und der Region (z. B. `s3-us-west-2`) anstelle eines Punkts (z. B. `s3.us-west-2`) enthalten. Wenn sich Ihr Bucket in einer dieser Regionen befindet, wird in Ihren Serverzugriffsprotokollen oder CloudTrail -protokollen möglicherweise das folgende Endpunktformat angezeigt:

```
https://bucket-name.s3-region-code.amazonaws.com
```

In diesem Beispiel lautet der Bucket-Name amzn-s3-demo-bucket1 und die Region ist US West (Oregon):

```
https://amzn-s3-demo-bucket1.s3-us-west-2.amazonaws.com
```

Globaler Legacy-Endpunkt

Für einige Regionen können Sie den globalen Legacy-Endpunkt verwenden, um Anforderungen zu erstellen, die keinen regionsspezifischen Endpunkt angeben. Der globale Legacy-Endpunkt lautet wie folgt:

```
bucket-name.s3.amazonaws.com
```

In Ihren Serverzugriffsprotokollen oder CloudTrail -protokollen sehen Sie möglicherweise Anfragen, die den alten globalen Endpunkt verwenden. In diesem Beispiel lautet der Bucket-Name `amzn-s3-demo-bucket1` und der globale Legacy-Endpunkt ist folgender:

```
https://amzn-s3-demo-bucket1.s3.amazonaws.com
```

Anforderungen im virtuellen Hosting-Format für USA Ost (Nord-Virginia)

Anforderungen, die über den globalen Legacy-Endpunkt gesendet werden, werden standardmäßig an USA Ost (Nord-Virginia) weitergeleitet. Daher wird manchmal der ältere globale Endpunkt anstelle des regionalen Endpunkts für USA Ost (Nord-Virginia) verwendet. Wenn Sie einen Bucket in USA Ost (Nord-Virginia) erstellen und den globalen Endpunkt verwenden, leitet Amazon S3 Ihre Anfrage standardmäßig an diese Region weiter.

Anforderungen im virtuellem Hosting-Format für andere Regionen

Der globale Legacy-Endpunkt wird auch für Anforderungen im virtuellen Hosting-Format in anderen unterstützten Regionen verwendet. Wenn Sie einen Bucket in einer Region erstellen, die vor dem 20. März 2019 gestartet wurde, und den alten globalen Endpunkt verwenden, aktualisiert Amazon S3 den DNS Datensatz, um die Anfrage an den richtigen Standort umzuleiten, was einige Zeit dauern kann. In der Zwischenzeit wird die Standardregel angewendet und Ihre Anfrage im virtuellen Hosting-Format wird an die Region USA Ost (Nord-Virginia) weitergeleitet. Amazon S3 leitet es dann mit einer temporären HTTP 307-Weiterleitung in die richtige Region weiter.

Bei S3-Buckets in Regionen, die nach dem 20. März 2019 gestartet wurden, leitet der DNS Server Ihre Anfrage nicht direkt an den Ort weiter, an AWS-Region dem sich Ihr Bucket befindet. Stattdessen wird der Fehler HTTP 400 Bad Request zurückgegeben. Weitere Informationen finden Sie unter Anfragen stellen in der Amazon S3 API S3-Referenz.

Anforderungen im Pfadformat

Für die Region USA Ost (Nord-Virginia) können Sie den globalen Legacy-Endpunkt für Anforderungen im Pfadformat verwenden.

Für alle anderen Regionen erfordert die Syntax im Pfadformat, dass beim Zugriff auf einen Bucket der regionsspezifische Endpunkt verwendet werden muss. Wenn Sie versuchen, auf einen Bucket mit dem alten globalen Endpunkt oder einem anderen Endpunkt als dem für die Region zuzugreifen, in der sich der Bucket befindet, erhalten Sie den HTTP Antwortcode 307 Temporary Redirect-Fehler und eine Meldung, die angibt, dass URI für Ihre Ressource der richtige ist. Wenn Sie beispielsweise `https://s3.amazonaws.com/bucket-name` für einen Bucket verwenden, der in der Region USA West (Oregon) erstellt wurde, erhalten Sie den Fehler HTTP 307 Temporary Redirect.

Erstellen eines Buckets

Um Ihre Daten auf Amazon S3 hochzuladen, müssen Sie zuerst einen Amazon-S3-Bucket in einer der AWS-Regionen erstellen. AWS-Konto Derjenige, der den Bucket erstellt, besitzt ihn. Wenn Sie einen Bucket erstellen, müssen Sie einen Bucket-Namen und eine Region auswählen. Sie können optional andere Speicherverwaltungsoptionen für den Bucket auswählen. Der Name eines erstellten Buckets oder Region kann nicht nachträglich geändert werden. Weitere Informationen zur Benennung von Buckets finden Sie unter Allgemeine Benennungsregeln für Buckets.

Standardmäßig können Sie bis zu 10.000 Allzweck-Buckets pro AWS-Konto Bucket erstellen. Besuchen Sie die Konsole Service Quotas, um eine Erhöhung des Kontingents für allgemeine Zwecke zu beantragen.

Sie können in einem Bucket beliebig viele Objekte speichern. Eine Liste der Einschränkungen und Beschränkungen im Zusammenhang mit Amazon S3 S3-Buckets finden Sie unter. Bucket-Kontingente, Beschränkungen und Einschränkungen

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie sowohl den Besitz von Objekten kontrollieren können, die in Ihren Bucket hochgeladen werden, als auch um Zugriffskontrolllisten zu deaktivieren oder zu aktivieren ()ACLs. Standardmäßig ist für Object Ownership die erzwungene Einstellung Bucket Owner festgelegt, und alle sind ACLs deaktiviert. Wenn diese ACLs Option deaktiviert ist, besitzt der Bucket-Besitzer jedes Objekt im Bucket und verwaltet den Zugriff auf Daten ausschließlich mithilfe von Richtlinien.

Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

Die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Basisebene der Verschlüsselungskonfiguration für jeden Bucket in Amazon S3. Alle neuen Objekte, die in einen S3-Bucket hochgeladen werden, werden automatisch mit SSE-S3 als Basisverschlüsselungseinstellung verschlüsselt. Wenn Sie eine andere Art der Standardverschlüsselung verwenden möchten, können Sie auch eine serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln (SSE-) oder vom Kunden bereitgestellten Schlüsseln (SSE-CKMS) angeben, um Ihre Daten zu verschlüsseln. Weitere Informationen finden Sie unter Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets.

Sie können die Amazon S3-Konsole, Amazon S3 oder verwenden APIs AWS CLI, AWS SDKs um einen Bucket zu erstellen. Weitere Informationen zu den für die Erstellung eines Buckets erforderlichen Berechtigungen finden Sie CreateBucketin der Amazon Simple Storage Service API Reference.

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen der aktuell angezeigten Datei aus AWS-Region. Wählen Sie als Nächstes die Region aus, in der Sie einen Bucket erstellen möchten.

   > **Note**
   >
   > Wählen Sie eine Region in der Nähe aus, um Latenz und Kosten gering zu halten und behördliche Vorschriften zu erfüllen. In einer Region gespeicherte Objekte verbleiben so lange in der Region, bis sie explizit in eine andere Region verschoben werden. Eine Liste von Amazon S3 finden Sie AWS-Regionen unter AWS-Service Endpoints in der Allgemeine Amazon Web Services-Referenz.

3. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
4. Wählen Sie Create Bucket (Bucket erstellen) aus.

   Anschließend wird die Seite Bucket erstellen geöffnet.
5. Sehen Sie sich unter Allgemeine Konfiguration an, AWS-Region wo Ihr Bucket erstellt wird.
6. Wählen Sie unter Bucket-Typ die Option Allgemeiner Zweck aus.
7. Geben Sie unter Bucket Name (Bucket-Name) einen Namen für den Bucket ein.

Der Bucket-Name ...:

- Muss innerhalb einer Partition einzigartig sein. Eine Partition ist eine Gruppierung von Regionen. AWS verfügt derzeit über drei Partitionen: `aws` (Standardregionen), `aws-cn` (China-Regionen) und `aws-us-gov` (AWS GovCloud (US) Regions).
- zwischen 3 und 63 Zeichen lang sein,
- Darf nur aus Kleinbuchstaben, Zahlen, Punkten (.) und Bindestrichen (-) bestehen. Aus Gründen der besten Kompatibilität empfehlen wir, Punkte (.) in Bucket-Namen zu vermeiden, mit Ausnahme von Buckets, die nur für statisches Website-Hosting verwendet werden.
- Muss mit einer Zahl oder einem Buchstaben beginnen und enden.

Der Name eines einmal erstellter Buckets kann nicht nachträglich geändert werden. AWS-Konto Derjenige, der den Bucket erstellt, besitzt ihn. Weitere Informationen zur Benennung von Buckets finden Sie unter Allgemeine Benennungsregeln für Buckets.

> **Important**
>
> Vermeiden Sie, vertrauliche Informationen, wie Kontonummern, in den Bucket-Namen einzubeziehen. Der Bucket-Name ist in dem Punkt sichtbarURLs, auf den die Objekte im Bucket verweisen.

8. AWS Management Console ermöglicht es Ihnen, die Einstellungen eines vorhandenen Buckets in Ihren neuen Bucket zu kopieren. Wenn Sie die Einstellungen eines vorhandenen Buckets nicht kopieren möchten, fahren Sie mit dem nächsten Schritt fort.

> **Note**
>
> Diese Option:
>
> - Ist in der nicht verfügbar AWS CLI und ist nur in der Konsole verfügbar
> - Ist für Verzeichnis-Buckets nicht verfügbar
> - Kopiert die Bucket-Richtlinie nicht aus dem vorhandenen Bucket in den neuen Bucket

Um die Einstellungen eines vorhandenen Buckets zu kopieren, wählen Sie unter Einstellungen aus vorhandenem Bucket kopieren die Option Bucket auswählen aus. Das Fenster „Bucket

auswählen“ wird geöffnet. Suchen Sie den Bucket mit den Einstellungen, die Sie kopieren möchten, und wählen Sie Bucket auswählen aus. Das Fenster „Bucket auswählen“ wird geschlossen, und das Fenster „Bucket erstellen“ wird erneut geöffnet.

Unter Einstellungen aus vorhandenem Bucket kopieren wird nun der Name des ausgewählten Buckets angezeigt. Außerdem wird die Option Standardwerte wiederherstellen angezeigt, mit der Sie die kopierten Bucket-Einstellungen entfernen können. Überprüfen Sie die verbleibenden Bucket-Einstellungen auf der Seite Bucket erstellen. Sie werden sehen, dass sie jetzt mit den Einstellungen des von Ihnen ausgewählten Buckets übereinstimmen. Sie können mit dem letzten Schritt fortfahren.

9. Wählen Sie unter Object Ownership eine der folgenden Einstellungen aus, um die Inhaberschaft von Objekten, die in Ihren Bucket hochgeladen wurden, zu deaktivieren oder zu aktivieren ACLs und zu kontrollieren:

   ACLsdeaktiviert

   - Bucket-Besitzer erzwungen (Standard) — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLswirkt sich nicht mehr auf die Zugriffsberechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet ausschließlich Richtlinien, um die Zugriffssteuerung zu definieren.

     Standardmäßig ACLs sind sie deaktiviert. Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs. Wir empfehlen, die ACLs Option weiterhin zu deaktivieren, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

   ACLsaktiviert

   - Bevorzugter Bucket-Besitzer — Der Bucket-Besitzer besitzt und hat die volle Kontrolle über neue Objekte, die andere Accounts in den Bucket mit den gespeicherten Objekten `bucket-owner-full-control` schreibenACL.

     Wenn Sie die bevorzugte Einstellung des Bucket-Besitzers anwenden, sodass alle Amazon S3 S3-Uploads die gespeicherten Dateien enthalten müssen `bucket-owner-full-control`ACL, können Sie eine Bucket-Richtlinie hinzufügen, die nur Objekt-Uploads zulässt, die dies verwenden. ACL

- Objekt-Writer — Derjenige AWS-Konto , der ein Objekt hochlädt, besitzt das Objekt, hat die volle Kontrolle darüber und kann anderen Benutzern Zugriff darauf gewähren. ACLs

> **Note**
>
> Die Standardeinstellung ist Bucket-Eigentümer erzwungen. Um die Standardeinstellung anzuwenden und ACLs deaktiviert zu lassen, ist nur die `s3:CreateBucket` entsprechende Berechtigung erforderlich. Zum Aktivieren ACLs benötigen Sie die `s3:PutBucketOwnershipControls` entsprechende Genehmigung.

10. Wählen Sie unter Einstellungen "Öffentlichen Zugriff beschränken" für diesen Bucket die Einstellungen zum Beschränken des öffentlichen Zugriffs aus, die Sie auf den Bucket anwenden möchten.

    Alle vier Einstellungen zum Blockieren des öffentlichen Zugriffs sind standardmäßig aktiviert. Es wird empfohlen, alle Einstellungen aktiviert zu lassen, es sei denn, Sie wissen, dass Sie eine oder mehrere dieser Einstellungen für Ihren Anwendungsfall deaktivieren müssen. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

    > **Note**
    >
    > Zum Aktivieren aller Einstellungen zum Blockieren des öffentlichen Zugriffs ist nur die `s3:CreateBucket`-Berechtigung erforderlich. Wenn Sie eine der Einstellungen zum Blockieren des öffentlichen Zugriffs deaktivieren möchten, benötigen Sie die `s3:PutBucketPublicAccessBlock`-Berechtigung.

11. (Optional) Unter Bucket Versioning (Bucket-Versionsverwaltung) können Sie auswählen, ob Sie Varianten von Objekten in Ihrem Bucket beibehalten möchten. Weitere Informationen über das Versioning finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

    Wenn Sie die Versionsverwaltung in Ihrem Bucket deaktivieren oder aktivieren möchten, wählen Sie entweder Disable (Deaktivieren) oder Enable (Aktivieren) aus.

12. (Optional) Unter Tags können Sie auswählen, ob Sie Ihrem Bucket Tags hinzufügen möchten. Tags sind Schlüssel-Wert-Paare, die zur Kategorisierung von Speicher verwendet werden.

Wenn Sie ein Bucket-Tag hinzuzufügen, geben Sie einen Key (Schlüssel) und optional einen Value (Wert) ein. Wählen Sie dann Add Tag (Tag hinzufügen) aus.

13. Wählen Sie unter Default encryption (Standard-Verschlüsselung) Edit (Bearbeiten) aus.
14. Wählen Sie eine der folgenden Optionen unter Verschlüsselungstyp aus, um die Standardverschlüsselung zu konfigurieren:

    - Verwalteter Amazon S3 S3-Schlüssel (SSE-S3)
    - AWS Key Management Service Schlüssel (SSE-KMS)

    > **Important**
    >
    > Wenn Sie die KMS Option SSE - für Ihre standardmäßige Verschlüsselungskonfiguration verwenden, unterliegen Sie dem Kontingent für Anfragen pro Sekunde (RPS) von AWS KMS. Weitere Informationen zu AWS KMS Kontingenten und dazu, wie Sie eine Kontingenterhöhung beantragen können, finden Sie unter Kontingente im AWS Key Management Service Entwicklerhandbuch.

    Buckets und neue Objekte werden mit serverseitiger Verschlüsselung verschlüsselt. Dabei ist ein Von Amazon S3 verwalteter Schlüssel die Grundebene der Verschlüsselungskonfiguration. Weitere Informationen zur Standardverschlüsselung finden Sie unter Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets.

    Weitere Informationen zur Datenverschlüsselung mit der serverseitigen Amazon-S3-Verschlüsselung finden Sie unter Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden (SSE-S3).
15. Wenn Sie die AWS Key Management Service Taste (SSE-KMS) gewählt haben, gehen Sie wie folgt vor:

    a. Geben Sie unter AWS KMS Schlüssel Ihren KMS Schlüssel auf eine der folgenden Arten an:

        - Um aus einer Liste verfügbarer KMS Schlüssel zu wählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys und wählen Sie Ihren KMSSchlüssel aus der Liste der verfügbaren Schlüssel aus.

          Sowohl der Von AWS verwalteter Schlüssel (`aws/s3`) als auch Ihr vom Kunden verwalteter Schlüssel werden in dieser Liste angezeigt. Weitere Informationen über vom

Kunden verwaltete Schlüssel finden Sie unter Kundenschlüssel und AWS -Schlüssel im Entwicklerhandbuch zu AWS Key Management Service .

- Um den KMS Schlüssel einzugebenARN, wählen Sie Enter AWS KMS key ARN und geben Sie Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
- Um einen neuen, vom Kunden verwalteten Schlüssel in der AWS KMS Konsole zu erstellen, wählen Sie Create a KMS key aus.

  Weitere Informationen zum Erstellen eines AWS KMS key finden Sie unter Schlüssel erstellen im AWS Key Management Service Entwicklerhandbuch.

> **Important**
>
> Sie können nur KMS Schlüssel verwenden, die im selben AWS-Region Bucket verfügbar sind. Die Amazon S3 S3-Konsole listet nur die ersten 100 KMS Schlüssel in derselben Region wie der Bucket auf. Um einen KMS Schlüssel zu verwenden, der nicht aufgeführt ist, müssen Sie Ihren KMS Schlüssel eingebenARN. Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, benötigen Sie zunächst die Erlaubnis, den Schlüssel zu verwenden, und dann müssen Sie den KMS Schlüssel eingebenARN. Weitere Informationen zu kontoübergreifenden Berechtigungen für KMS Schlüssel finden Sie im AWS Key Management Service Entwicklerhandbuch unter Erstellen von KMS Schlüsseln, die andere Konten verwenden können. Weitere Informationen zu SSE - finden KMS Sie unterGeben Sie die serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) an. Wenn Sie eine AWS KMS key für die serverseitige Verschlüsselung in Amazon S3 verwenden, müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel und keine asymmetrischen KMS Schlüssel. Weitere Informationen finden Sie unter Identifizieren symmetrischer und asymmetrischer KMS Schlüssel im Entwicklerhandbuch.AWS Key Management Service

Weitere Informationen zum Erstellen von finden Sie unter Creating Keys im AWS Key Management Service Developer Guide. AWS KMS key Weitere Informationen zur Verwendung AWS KMS mit Amazon S3 finden Sie unterServerseitige Verschlüsselung mit AWS KMS Schlüsseln verwenden (SSE-) KMS.

b. Wenn Sie Ihren Bucket so konfigurieren, dass er die Standardverschlüsselung mit SSE - verwendetKMS, können Sie auch S3 Bucket Keys aktivieren. S3-Bucket-Keys senken die Kosten für die Verschlüsselung, indem sie den Anforderungsverkehr von Amazon S3 zu verringern AWS KMS. Weitere Informationen finden Sie unter Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys.

   Um S3-Bucket-Schlüssel zu verwenden, wählen Sie unter Bucket Key (Bucket-Schlüssel) die Option Enable (Aktivieren).

16. (Optional) Wenn Sie die S3-Objektsperre aktivieren möchten, gehen Sie wie folgt vor:

    a. Wählen Sie Erweiterte Einstellungen aus.

    > **Important**
    >
    > Durch Aktivieren der Objektsperre wird auch die Versioning für den Bucket aktiviert. Nach dem Aktivieren müssen Sie die Standardeinstellungen für die Objektsperre im Hinblick auf die (rechtliche) Aufbewahrung konfigurieren, um neue Objekte vor dem Löschen oder Überschreiben zu schützen.

    b. Wenn Sie die Objektsperre aktivieren möchten, wählen Sie Enable (Aktivieren) aus, lesen Sie die angezeigte Warnung und bestätigen Sie sie.

    Weitere Informationen finden Sie unter Sperren von Objekten mit Object Lock.

    > **Note**
    >
    > Wenn Sie einen Bucket mit aktivierter Objektsperre erstellen möchten, benötigen Sie die folgenden Berechtigungen: `s3:CreateBucket`, `s3:PutBucketVersioning` und `s3:PutBucketObjectLockConfiguration`.

17. Wählen Sie Bucket erstellen aus.

Mit dem AWS SDKs

Wenn Sie den verwenden, AWS SDKs um einen Bucket zu erstellen, müssen Sie einen Client erstellen und dann den Client verwenden, um eine Anfrage zum Erstellen eines Buckets zu senden. Als bewährte Methode sollten Sie Ihren Client und Ihren Bucket in derselben AWS-

Region erstellen. Wenn Sie beim Erstellen eines Clients oder Buckets keine Region angeben, verwendet Amazon S3 die Standardregion USA Ost (Nord-Virginia). Wenn Sie die Bucket-Erstellung auf eine bestimmte AWS-Region beschränken möchten, verwenden Sie dazu den Bedingungsschlüssel `LocationConstraint`.

Um einen Client für den Zugriff auf einen Dual-Stack-Endpunkt zu erstellen, müssen Sie eine AWS-Region angeben. Weitere Informationen finden Sie unter Verwenden von Amazon S3 S3-Dual-Stack-Endpunkten in der Amazon S3 API S3-Referenz. Eine Liste der verfügbaren AWS-Regionen Regionen und Endpunkte finden Sie unter Regionen und Endpunkte in der. Allgemeine AWS-Referenz

Wenn Sie einen Client erstellen, wird die Region dem regionsspezifischen Endpunkt zugeordnet. Der Client verwendet diesen Endpunkt für die Kommunikation mit Amazon S3: `s3.region.amazonaws.com`. Wenn Ihre Region nach dem 20. März 2019 gestartet wurde, müssen sich Ihr Client und Ihr Bucket in derselben Region befinden. Sie können jedoch einen Client in der Region USA Ost (Nord-Virginia) verwenden, um einen Bucket in einer beliebigen Region zu erstellen, die vor dem 20. März 2019 gestartet wurde. Weitere Informationen finden Sie unter Legacy-Endpunkte.

Mit diesen AWS SDK Codebeispielen werden die folgenden Aufgaben ausgeführt:

- Erstellen eines Clients durch explizite Angabe einer AWS-Region – Im Beispiel verwendet der Client den Endpunkt `s3.us-west-2.amazonaws.com`, um mit Amazon S3 zu kommunizieren. Sie können eine beliebige AWS-Region angeben. Eine Liste von AWS-Regionen finden Sie unter Regionen und Endpunkte in der AWS allgemeinen Referenz.
- Senden einer Bucket-Erstellungs-Anfrage durch Angabe eines Bucket-Namens – Der Client sendet eine Anfrage an Amazon S3, um den Bucket in der Region zu erstellen, in der Sie einen Client erstellt haben.
- Abrufen von Informationen zum Standort des Buckets – Amazon S3 speichert Informationen zum Standort des Buckets in der Subressource Standort, die dem Bucket zugeordnet ist.

Java

Example Erstellen Sie einen Bucket, der einen global eindeutigen Bezeichner (GUID) im Bucket-Namen verwendet

Das folgende Beispiel zeigt Ihnen, wie Sie einen Bucket mit einem GUID am Ende des Bucket-Namens in der Region USA Ost (Nord-Virginia) (us-east-1) erstellen, indem Sie den verwenden.

AWS SDK for Java Weitere Informationen zu anderen finden Sie unter Tools AWS SDKs, auf denen Sie aufbauen können. AWS

```
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.Bucket;
import com.amazonaws.services.s3.model.CreateBucketRequest;

import java.util.List;
import java.util.UUID;

public class CreateBucketWithUUID {
    public static void main(String[] args) {
        final AmazonS3 s3 =
 AmazonS3ClientBuilder.standard().withRegion(Regions.US_EAST_1).build();
        String bucketName = "amzn-s3-demo-bucket" +
 UUID.randomUUID().toString().replace("-", "");
        CreateBucketRequest createRequest = new CreateBucketRequest(bucketName);
        System.out.println(bucketName);
        s3.createBucket(createRequest);
    }
}
```

Example Erstellen eines -Buckets

Dieses Beispiel zeigt Ihnen, wie Sie mit dem einen Amazon S3 S3-Bucket erstellen AWS SDK for Java. Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CreateBucketRequest;
import com.amazonaws.services.s3.model.GetBucketLocationRequest;

import java.io.IOException;
```

```
public class CreateBucket2 {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            if (!s3Client.doesBucketExistV2(bucketName)) {
                // Because the CreateBucketRequest object doesn't specify a region,
 the
                // bucket is created in the region specified in the client.
                s3Client.createBucket(new CreateBucketRequest(bucketName));

                // Verify that the bucket was created by retrieving it and checking
 its
                // location.
                String bucketLocation = s3Client.getBucketLocation(new
 GetBucketLocationRequest(bucketName));
                System.out.println("Bucket location: " + bucketLocation);
            }
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Informationen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter AWS SDKfür. NETAPIReferenz zu Version 3.

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using Amazon.S3.Util;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CreateBucketTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            CreateBucketAsync().Wait();
        }

        static async Task CreateBucketAsync()
        {
            try
            {
                if (!(await AmazonS3Util.DoesS3BucketExistAsync(s3Client,
 bucketName)))
                {
                    var putBucketRequest = new PutBucketRequest
                    {
                        BucketName = bucketName,
                        UseClientRegion = true
                    };

                    PutBucketResponse putBucketResponse = await
 s3Client.PutBucketAsync(putBucketRequest);
                }
                // Retrieve the bucket location.
                string bucketLocation = await FindBucketLocationAsync(s3Client);
            }
            catch (AmazonS3Exception e)
```

```
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
        static async Task<string> FindBucketLocationAsync(IAmazonS3 client)
        {
            string bucketLocation;
            var request = new GetBucketLocationRequest()
            {
                BucketName = bucketName
            };
            GetBucketLocationResponse response = await
 client.GetBucketLocationAsync(request);
            bucketLocation = response.Location.ToString();
            return bucketLocation;
        }
    }
}
```

Ruby

Informationen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter AWS SDKFür Ruby — Version 3.

Example

```
require 'aws-sdk-s3'

# Wraps Amazon S3 bucket actions.
class BucketCreateWrapper
  attr_reader :bucket

  # @param bucket [Aws::S3::Bucket] An Amazon S3 bucket initialized with a name.
 This is a client-side object until
  #                                 create is called.
  def initialize(bucket)
    @bucket = bucket
  end
```

```
  # Creates an Amazon S3 bucket in the specified AWS Region.
  #
  # @param region [String] The Region where the bucket is created.
  # @return [Boolean] True when the bucket is created; otherwise, false.
  def create?(region)
    @bucket.create(create_bucket_configuration: { location_constraint: region })
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't create bucket. Here's why: #{e.message}"
    false
  end

  # Gets the Region where the bucket is located.
  #
  # @return [String] The location of the bucket.
  def location
    if @bucket.nil?
      'None. You must create a bucket before you can get its location!'
    else
      @bucket.client.get_bucket_location(bucket: @bucket.name).location_constraint
    end
  rescue Aws::Errors::ServiceError => e
    "Couldn't get the location of #{@bucket.name}. Here's why: #{e.message}"
  end
end

# Example usage:
def run_demo
  region = "us-west-2"
  wrapper = BucketCreateWrapper.new(Aws::S3::Bucket.new("amzn-s3-demo-bucket-
#{Random.uuid}"))
  return unless wrapper.create?(region)

  puts "Created bucket #{wrapper.bucket.name}."
  puts "Your bucket's region is: #{wrapper.location}"
end

run_demo if $PROGRAM_NAME == __FILE__
```

Unter Verwendung des AWS CLI

Im folgenden AWS CLI Beispiel wird ein Bucket in der Region USA West (Nordkalifornien) (us-west-1) mit einem Beispiel-Bucket-Namen erstellt, der einen global eindeutigen Bezeichner (GUID) verwendet.

```
aws s3api create-bucket \
    --bucket amzn-s3-demo-bucket1$(uuidgen | tr -d - | tr '[:upper:]' '[:lower:]' ) \
    --region us-west-1 \
    --create-bucket-configuration LocationConstraint=us-west-1
```

Weitere Informationen finden Sie unter create-bucket in der AWS CLI -Befehlsreferenz.

Anzeigen der Eigenschaften eines S3-Buckets

Sie können Eigenschaften für jeden Amazon S3 S3-Bucket anzeigen, den Sie besitzen. Diese Einstellungen umfassen u. a. folgende:

- Bucket Versioning (Bucket-Versionssteuerung) – Behalten Sie mehrere Versionen eines Objekts in einem Bucket bei, indem Sie die Versionssteuerung verwenden. Für einen neuen Bucket ist das Versioning standardmäßig deaktiviert. Weitere Informationen über die Aktivierung des Versionings finden Sie unter Aktivieren des Versioning für Buckets.
- Tags — Bei der AWS Kostenzuweisung können Sie Bucket-Tags verwenden, um die Abrechnung für Ihre Nutzung eines Buckets mit Anmerkungen zu versehen. Ein Tag ist ein Schlüssel-Wert-Paar, das eine Bezeichnung repräsentiert, die Sie einem Bucket zuweisen. Weitere Informationen finden Sie unter Verwenden von Kostenzuordnungs-Markierungen für S3-Buckets.
- Default encryption (Standardverschlüsselung) – Die Aktivierung der Standardverschlüsselung bietet Ihnen eine automatische serverseitige Verschlüsselung. Amazon S3 verschlüsselt ein Objekt, bevor es auf einer Festplatte gespeichert wird, und entschlüsselt das Objekt beim Herunterladen. Weitere Informationen finden Sie unter Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets.
- Server access logging (Server-Zugriffsprotokollierung) – Die Server-Zugriffsprotokollierung bietet detaillierte Aufzeichnungen über die Anfragen, die an Ihren Bucket gestellt wurden. Standardmäßig erfasst Amazon S3 keine Server-Zugriffsprotokolle. Weitere Informationen zur Aktivierung der Server-Zugriffsprotokollierung finden Sie unter Aktivieren Sie die Amazon-S3-Server-Zugriffsprotokollierung.

- AWS CloudTrail Datenereignisse — Dient CloudTrail zum Protokollieren von Datenereignissen. Standardmäßig werden Datenereignisse nicht von den Trails protokolliert. Für Datenereignisse werden zusätzliche Gebühren fällig. Weitere Informationen finden Sie unter Protokollierung von Datenereignissen für Trails im AWS CloudTrail -Benutzerhandbuch.
- Event notifications (Ereignisbenachrichtigungen)Ereignisse – Sie können bestimmte Amazon S3-Bucket-Ereignisse aktivieren, um eine Benachrichtigungsmeldung an ein Ziel zu senden, wenn das Ereignis auftritt. Weitere Informationen finden Sie unter Aktivieren und Konfigurieren von Ereignis-Benachrichtigungen mit der Amazon-S3-Konsole.
- Transfer Acceleration – Ermöglicht schnelle, einfache und sichere Übertragungen von Dateien zwischen Ihrem Client und einem S3-Bucket über große Entfernungen. Weitere Informationen zur Aktivierung der Transfer Acceleration finden Sie unter Aktivieren und Verwenden von S3 Transfer Acceleration.
- Object Lock – Mit der S3-Objektsperre können Sie für einen festen Zeitraum oder auf unbegrenzte Zeit verhindern, dass ein Objekt gelöscht oder überschrieben wird. Weitere Informationen finden Sie unter Sperren von Objekten mit Object Lock.
- Requester Pays (Zahlung durch den Anforderer) – Sie können die Zahlung durch den Anforderer aktivieren, sodass der Anforderer (anstelle des Bucket-Eigentümers) für die Anforderungen und Datenübertragungen zahlt. Weitere Informationen finden Sie unter Nutzen von Buckets mit Zahlung durch den Anforderer für Speicherübertragungen und Nutzung.
- Static website hosting (Hosting statischer Websites) – Sie können in Amazon S3 eine statische Website hosten. Weitere Informationen finden Sie unter Hosten einer statischen Website mit Amazon S3.

Sie können Bucket-Eigenschaften mit dem AWS Management Console AWS CLI, oder anzeigen AWS SDKs

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets) den Namen des Buckets aus, dessen Eigenschaften Sie anzeigen wollen.
3. Wählen Sie die Registerkarte Eigenschaften aus.
4. Auf der Eigenschaftenseite können Sie die oben genannten Eigenschaften für den Bucket konfigurieren.

Mit dem AWS CLI

Bucket-Eigenschaften anzeigen mit dem AWS CLI

Die folgenden Befehle zeigen, wie Sie die verwenden können AWS CLI , um verschiedene Bucket-Eigenschaften aufzulisten.

Im Folgenden wird der Tagsatz zurückgegeben, der dem Bucket zugeordnet ist *amzn-s3-demo-bucket1*. Weitere Informationen zu Bucket-Tags finden Sie unterVerwenden von Kostenzuordnungs-Markierungen für S3-Buckets.

```
aws s3api get-bucket-tagging --bucket amzn-s3-demo-bucket1
```

Weitere Informationen und Beispiele finden Sie unter get-bucket-taggingin der AWS CLI Befehlsreferenz.

Im Folgenden wird der Versionsstatus des Buckets zurückgegeben *amzn-s3-demo-bucket1*. Hinweise zur Bucket-Versionierung finden Sie unterBeibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

```
aws s3api get-bucket-versioning --bucket amzn-s3-demo-bucket1
```

Weitere Informationen und Beispiele finden Sie unter get-bucket-versioningin der AWS CLI Befehlsreferenz.

Im Folgenden wird die Standardverschlüsselungskonfiguration für den Bucket zurückgegeben *amzn-s3-demo-bucket1*. Standardmäßig haben alle Buckets eine Standardverschlüsselungskonfiguration, die serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verwendet. Informationen zur Standardverschlüsselung des Buckets finden Sie unter. Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets

```
aws s3api get-bucket-encryption --bucket amzn-s3-demo-bucket1
```

Weitere Informationen und Beispiele finden Sie unter get-bucket-encryptionin der AWS CLI Befehlsreferenz.

Im Folgenden wird die Benachrichtigungskonfiguration des Buckets zurückgegeben *amzn-s3-demo-bucket1*. Informationen zu den Bucket-Ereignisbenachrichtigungen finden Sie unterAmazon-S3-Ereignis-Benachrichtigungen.

```
aws s3api get-bucket-notification-configuration --bucket amzn-s3-demo-bucket1
```

Weitere Informationen und Beispiele finden Sie unter get-bucket-notification-configurationin der AWS CLI Befehlsreferenz.

Im Folgenden wird der Logging-Status für den Bucket zurückgegeben *amzn-s3-demo-bucket1*. Hinweise zur Bucket-Protokollierung finden Sie unterProtokollieren von Anfragen mit Server-Zugriffsprotokollierung.

```
aws s3api get-bucket-logging --bucket amzn-s3-demo-bucket1
```

Weitere Informationen und Beispiele finden Sie unter get-bucket-loggingin der AWS CLI Befehlsreferenz.

Mit dem AWS SDKs

Beispiele für die Rückgabe von Bucket-Eigenschaften mit dem AWS SDKs, wie Versionierung, Tags und mehr, finden Sie in den Codebeispielen in der Amazon S3 API S3-Referenz.

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS SDKs in der Amazon S3 API S3-Referenz.

Amazon S3 S3-Buckets auflisten

Um eine Liste der Allzweck-Buckets zurückzugeben, die Sie besitzen, können Sie verwenden. ListBuckets Sie können Ihre Buckets mit der Amazon S3 S3-Konsole AWS Command Line Interface, dem oder dem AWS SDKs auflisten. Für `ListBuckets` Anfragen mit, und Amazon S3 AWS CLI AWS SDKs RESTAPI, AWS-Konten die das Standard-Servicekontingent für Buckets (10.000 Buckets) verwenden, werden sowohl paginierte als auch unpaginierte Anfragen unterstützt. Unabhängig davon, wie viele Buckets Sie in Ihrem Konto haben, können Sie Seitengrößen zwischen 1 und 10.000 Buckets erstellen, um alle Ihre Buckets aufzulisten. Bei paginierten Anfragen geben Anfragen sowohl die Bucket-Namen als auch die entsprechenden AWS-Regionen Namen für jeden Bucket zurück. `ListBuckets` Im Folgenden AWS Command Line Interface und AWS SDK anhand von Beispielen erfahren Sie, wie Sie die Paginierung in Ihrer `ListBuckets` Anfrage verwenden. Beachten Sie, dass einige AWS SDKs bei der Paginierung helfen.

Berechtigungen

Zum Auflisten sämtlicher Buckets benötigen Sie die Genehmigung `s3:ListAllMyBuckets`. Wenn Sie auf einen `HTTP Access Denied (403 Forbidden)` Fehler stoßen, finden Sie weitere Informationen unterProblembehandlung bei verweigertem Zugriff beheben (403 Forbidden) Fehler in Amazon S3.

> **Important**
>
> Wir empfehlen dringend, nur paginierte `ListBuckets` Anfragen zu verwenden. `ListBuckets`Anfragen ohne Seiten werden nur unterstützt, wenn sie auf das standardmäßige allgemeine Bucket-Kontingent von 10.000 AWS-Konten festgelegt sind. Wenn Sie ein genehmigtes Kontingent für allgemeine Buckets von mehr als 10.000 haben, müssen Sie paginierte `ListBuckets` Anfragen senden, um die Buckets Ihres Kontos aufzulisten. Alle `ListBuckets` Anfragen ohne Seitennummerierung werden abgelehnt, wenn das Kontingent für AWS-Konten allgemeine Zwecke mehr als 10.000 beträgt.

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Auf der Registerkarte Allzweck-Buckets sehen Sie eine Liste Ihrer Allzweck-Buckets.
4. Um Buckets nach Namen zu suchen, geben Sie einen Bucket-Namen in das Suchfeld ein.

Verwenden Sie den AWS CLI

Um mit dem eine Liste von Allzweck-Buckets AWS CLI zu generieren, können Sie die `list-buckets` Befehle `ls` oder verwenden. Die folgenden Beispiele zeigen, wie Sie eine paginierte und eine `ls` unpaginierte `list-buckets` Anfrage erstellen. Um diese Beispiele zu verwenden, ersetzen Sie das *`user input placeholders`*.

Example — Führe alle Buckets in deinem Konto auf, indem du **`ls`** (unpaginiert) verwendest

Der folgende Beispielbefehl listet alle Allzweck-Buckets in Ihrem Konto in einem einzigen Aufruf ohne Seitennummerierung auf. Dieser Aufruf gibt eine Liste aller Buckets in Ihrem Konto zurück (bis zu 10.000 Ergebnisse):

```
$ aws s3 ls
```

Weitere Informationen finden Sie unter Auflisten von Buckets und Objekten.

Example — Listet alle Buckets in Ihrem Konto auf, indem Sie **`ls`** (paginiert) verwenden

Der folgende Beispielbefehl führt einen oder mehrere paginierte Aufrufe durch, um alle Allzweck-Buckets in Ihrem Konto aufzulisten, wobei 100 Buckets pro Seite zurückgegeben werden:

```
$ aws s3 ls --page-size 100
```

Weitere Informationen finden Sie unter Auflisten von Buckets und Objekten.

Example — Listet alle Buckets in Ihrem Konto auf (paginiert)

Das folgende Beispiel enthält einen paginierten `list-buckets` Befehl, mit dem Sie alle Allzweck-Buckets in Ihrem Konto auflisten können. Die `--page-size` Optionen `--max-items` und begrenzen die Anzahl der aufgelisteten Buckets auf 100 pro Seite.

```
$ aws s3api list-buckets /
    --max-items 100 /
    --page-size 100
```

Wenn die Anzahl der Output (`--max-items`) -Elemente geringer ist als die Gesamtzahl der von den zugrunde liegenden API Aufrufen zurückgegebenen Elemente, enthält die Ausgabe ein durch das `starting-token` Argument spezifiziertes Fortsetzungstoken, das Sie an einen nachfolgenden Befehl übergeben können, um die nächste Gruppe von Elementen abzurufen. Das folgende Beispiel zeigt, wie der vom vorherigen Beispiel zurückgegebene `starting-token` Wert verwendet wird. Sie können angeben`starting-code`, dass die nächsten 100 Buckets abgerufen werden sollen.

```
$ aws s3api list-buckets /
    --max-items 100 /
    --page-size 100 /
    --starting-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ==
```

Example — Listet alle Buckets in einer AWS-Region (paginierten)

Der folgende Beispielbefehl verwendet den `--bucket-region` Parameter, um bis zu 100 Buckets in einem Konto aufzulisten, die sich in der Region befinden. `us-east-2` Anfragen an einen regionalen Endpunkt, der sich von dem im `--bucket-region` Parameter angegebenen Wert unterscheidet, werden nicht unterstützt. Wenn Sie beispielsweise die Antwort auf Ihre Buckets in beschränken möchten`us-east-2`, müssen Sie Ihre Anfrage an einen Endpunkt in `us-east-2` richten.

```
$ aws s3api list-buckets /
    --region us-east-2 /
    --max-items 100 /
    --page-size 100 /
    --bucket-region us-east-2
```

Example — Listet alle Buckets auf, die mit einem bestimmten Bucket-Namenspräfix beginnen (paginiert)

Der folgende Beispielbefehl listet bis zu 100 Buckets auf, deren Name mit dem beginnt *amzn-s3-demo-bucket* Präfix.

```
$ aws s3api list-buckets /
    --max-items 100 /
    --page-size 100 /
    --prefix amzn-s3-demo-bucket
```

Mit dem AWS SDKs

Die folgenden Beispiele zeigen Ihnen, wie Sie Ihre Allzweck-Buckets auflisten, indem Sie den AWS SDKs

SDK for Python

Example — ListBuckets Anfrage (paginiert)

```
import boto3

s3 = boto3.client('s3')
response = s3.list_buckets(MaxBuckets=100)
```

Example — ListBuckets Antwort (paginiert)

```
import boto3

s3 = boto3.client('s3')
response =
 s3.list_buckets(MaxBuckets=1,ContinuationToken="eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9
```

SDK for Java

```
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.Bucket;
import com.amazonaws.services.s3.model.ListBucketsPaginatedRequest;
import com.amazonaws.services.s3.model.ListBucketsPaginatedResult;

import java.util.List;

public class ListBuckets {
    public static void main(String[] args) {
        final AmazonS3 s3 =
 AmazonS3ClientBuilder.standard().withRegion(Regions.DEFAULT_REGION).build();
        ListBucketsPaginatedRequest listBucketsPaginatedRequest = new
 ListBucketsPaginatedRequest().withMaxBuckets(1);
        ListBucketsPaginatedResult listBucketsPaginatedResult =
 s3.listBuckets(listBucketsPaginatedRequest);
        List<Bucket> buckets = listBucketsPaginatedResult.getBuckets();
        System.out.println("Your Amazon S3 buckets are:");
        for (Bucket b : buckets) {
            System.out.println("* " + b.getName() + " region: " + b.getRegion());
        }
        System.out.println("continuation token: " +
 listBucketsPaginatedResult.getContinuationToken());
    }
}
```

SDK for Java 2.x

```
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
```

```
import software.amazon.awssdk.services.s3.model.Bucket;
import software.amazon.awssdk.services.s3.model.ListBucketsRequest;
import software.amazon.awssdk.services.s3.model.ListBucketsResponse;

import java.util.List;

public class ListBuckets {
    public static void main(String[] args) {
        // Create an S3 client
        S3Client s3 = S3Client.builder()
                .region(Region.US_EAST_1) // Replace with your preferred region
                .credentialsProvider(DefaultCredentialsProvider.create())
                .build();

        try {
            // List buckets
            ListBucketsRequest listBucketsRequest = ListBucketsRequest.builder()
                    .maxBuckets(10)
                    .build();
            ListBucketsResponse listBucketsResponse =
 s3.listBuckets(listBucketsRequest);
            List<Bucket> buckets = listBucketsResponse.buckets();

            // Print bucket names
            System.out.println("Your Amazon S3 buckets are:");
            for (Bucket bucket : buckets) {
                System.out.println(bucket.name());
                System.out.println(bucket.getBucketRegion());
            }

        } catch (Exception e) {
            System.err.println("Error listing buckets: " + e.getMessage());
            e.printStackTrace();
        } finally {
            // Close the S3 client to release resources
            s3.close();
        }
    }
}
```

SDK for Go

```
package main
```

```
import (
 "context"
 "fmt"
 "log"
 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/s3"
)
func main() {
 cfg, err := config.LoadDefaultConfig(context.TODO(), config.WithRegion("us-
east-2"))
 if err != nil {
  log.Fatal(err)
 }
 client := s3.NewFromConfig(cfg)
 maxBuckets := 1000
 resp, err := client.ListBuckets(context.TODO(), management
 portals3.ListBucketsInput{MaxBuckets: aws.Int32(int32(maxBuckets))})
 if err != nil {
  log.Fatal(err)
 }
 fmt.Println("S3 Buckets:")
 for _, bucket := range resp.Buckets {
     fmt.Println("- Name:", *bucket.Name)
     fmt.Println("-BucketRegion", *bucket.BucketRegion)
 }
 fmt.Println(resp.ContinuationToken == nil)
 fmt.Println(resp.Prefix == nil)
}
```

Leeren eines Buckets

Sie können den Inhalt eines Buckets mit der Amazon S3 S3-Konsole, AWS SDKs, oder AWS Command Line Interface (AWS CLI) leeren. Wenn Sie einen Bucket leeren, löschen Sie alle Objekte, behalten aber den Bucket. Nachdem Sie einen Bucket geleert haben, kann es nicht rückgängig gemacht werden. Dem Bucket während der Löschaktion hinzugefügte Objekte werden möglicherweise gelöscht. Alle Objekte (einschließlich aller Objektversionen und Löschmarkierungen) im Bucket müssen gelöscht werden, bevor der Bucket selbst gelöscht werden kann.

Wenn Sie einen Bucket mit aktivierter oder ausgesetzter S3-Versionsverwaltung leeren, werden alle Versionen aller Objekte im Bucket gelöscht. Weitere Informationen finden Sie unter Arbeiten mit Objekten in einem versioning-fähigen Bucket.

Sie können auch eine Lebenszyklus-Konfiguration für einen Bucket angeben, sodass Objekte ablaufen, damit Amazon S3 sie löschen kann. Weitere Informationen finden Sie unter Einstellung einer S3-Lifecycle-Konfiguration für einen Bucket. Um einen großen Bucket zu leeren, empfehlen wir, eine S3-Lifecycle-Konfigurationsregel zu verwenden. Der Ablauf des Lebenszyklus ist ein asynchroner Prozess, daher kann es einige Tage dauern, bis die Regel ausgeführt wird und der Bucket leer ist. Nachdem Amazon S3 die Regel zum ersten Mal ausführt, werden alle Objekte, die für den Ablauf in Frage kommen, zum Löschen markiert. Die Objekte, die zum Löschen markiert sind, werden Ihnen nicht mehr in Rechnung gestellt. Weitere Informationen finden Sie unter Wie leere ich einen Amazon-S3-Bucket mithilfe einer Lebenszykluskonfigurationsregel?.

Verwenden der S3-Konsole

Sie können die Amazon-S3-Konsole verwenden, um einen Bucket zu leeren, der alle Objekte im Bucket löscht, ohne den Bucket zu löschen.

Einen S3-Bucket leeren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Bucket Name (Bucket-Name) die Option neben dem Namen des Buckets aus, den Sie leeren möchten. Wählen Sie anschließend Empty (Leeren) aus.
3. Bestätigen Sie auf der Seite Empty bucket (Leerer Bucket), dass Sie den Bucket leeren möchten, indem Sie den Bucket-Namen in das Textfeld eingeben, und wählen Sie dann Empty (Leeren).
4. Überwachen Sie den Fortschritt des Vorgangs zum Entleeren von Buckets auf der Seite Empty bucket: Status (Bucket entleeren: Status).

Verwenden Sie den AWS CLI

Sie können einen Bucket AWS CLI nur dann leeren, wenn für den Bucket die Bucket-Versionierung nicht aktiviert ist. Wenn die Versionierung nicht aktiviert ist, können Sie den AWS CLI Befehl `rm` (remove) mit dem `--recursive` Parameter verwenden, um den Bucket zu leeren (oder eine Teilmenge von Objekten mit einem bestimmten Schlüsselnamenpräfix zu entfernen).

Der folgende `rm`-Befehl entfernt Objekte mit dem Schlüsselnamenpräfix `doc`, z. B. `doc/doc1` und `doc/doc2`.

```
$ aws s3 rm s3://bucket-name/doc --recursive
```

Verwenden Sie den folgenden Befehl, um alle Objekte zu entfernen, ohne ein Präfix anzugeben.

```
$ aws s3 rm s3://bucket-name --recursive
```

Weitere Informationen finden Sie unter Using High-Level S3 Commands with the AWS CLI(Verwenden von hochrangigen S3-Befehlen mit der CLI) im AWS Command Line Interface - Benutzerhandbuch.

> **Note**
>
> Sie können keine Objekte aus einem Bucket entfernen, für den die Versionierung aktiviert ist. Amazon S3 fügt eine Löschmarkierung hinzu, wenn Sie ein Objekt löschen, was dieser Befehl tut. Weitere Informationen über die S3-Bucket-Versioning finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

Verwenden Sie den AWS SDKs

Sie können den verwenden AWS SDKs, um einen Bucket zu leeren oder eine Teilmenge von Objekten zu entfernen, die ein bestimmtes Schlüsselnamenpräfix haben.

Ein Beispiel dafür, wie Sie einen Bucket mit leeren AWS SDK for Java, finden Sie unterLöschen eines Buckets. Der Code löscht alle Objekte, unabhängig davon, ob für den Bucket das Versioning aktiviert ist. Anschließend löscht er den Bucket. Um den Bucket nur zu leeren, stellen Sie sicher, dass Sie die Anweisung entfernen, die den Bucket löscht.

Weitere Informationen zur Verwendung anderer AWS SDKs finden Sie unter Tools für Amazon Web Services.

Verwenden einer Lebenszyklus-Konfiguration

Um einen großen Bucket zu leeren, empfehlen wir, eine S3-Lifecycle-Konfigurationsregel zu verwenden. Der Ablauf des Lebenszyklus ist ein asynchroner Prozess, daher kann es einige Tage dauern, bis die Regel ausgeführt wird und der Bucket leer ist. Nachdem Amazon S3 die Regel zum ersten Mal ausführt, werden alle Objekte, die für den Ablauf in Frage kommen, zum Löschen

markiert. Die Objekte, die zum Löschen markiert sind, werden Ihnen nicht mehr in Rechnung gestellt. Weitere Informationen finden Sie unter Wie leere ich einen Amazon-S3-Bucket mithilfe einer Lebenszykluskonfigurationsregel?.

Wenn Sie eine Lebenszykluskonfiguration verwenden, um Ihren Bucket zu leeren, sollte die Konfiguration aktuelle Versionen, nicht aktuelle Versionen, Löschmarkierungen und unvollständige mehrteilige Uploads enthalten.

Sie können Lebenszyklus-Konfigurationsregeln hinzufügen, sodass alle Objekte oder eine Untermenge davon mit einem spezifischen Schlüsselnamenpräfix ablaufen. Sie können z. B. eine Lebenszyklusregel so festlegen, dass Objekte einen Tag nach dem Erstellungsdatum ablaufen, um alle Objekte in einem Bucket zu entfernen.

Amazon S3 unterstützt eine Bucket-Lebenszyklusregel, mit der Sie unvollständige mehrteilige Uploads abbrechen können, die nicht innerhalb einer bestimmten Anzahl von Tagen nach der Initiierung abgeschlossen werden. Wir empfehlen, dass Sie diese Lebenszyklusregel konfigurieren, um Ihre Speicherkosten zu minimieren. Weitere Informationen finden Sie unter Konfigurieren einer Bucket-Lebenszykluskonfiguration zum Löschen unvollständiger mehrteiliger Uploads.

Weitere Informationen zur Verwendung einer Lebenszyklus-Konfiguration zum Leeren eines Buckets finden Sie unter Einstellung einer S3-Lifecycle-Konfiguration für einen Bucket und Auslaufende Objekte.

Leeren eines Buckets mit den folgenden Einstellungen AWS CloudTrail

AWS CloudTrail verfolgt Datenereignisse auf Objektebene in einem Amazon S3 S3-Bucket, z. B. das Löschen von Objekten. Wenn Sie einen Bucket als Ziel für die Protokollierung Ihrer CloudTrail Ereignisse verwenden und Objekte aus demselben Bucket löschen, erstellen Sie möglicherweise neue Objekte, während Sie Ihren Bucket leeren. Um dies zu verhindern, sollten Sie Ihre AWS CloudTrail Spuren unterbrechen. Weitere Informationen darüber, wie du verhindern kannst, dass deine CloudTrail Trails Ereignisse protokollieren, findest du im AWS CloudTrail Benutzerhandbuch unter Deaktivieren der Protokollierung für einen Trail.

Eine weitere Alternative, um zu verhindern, dass CloudTrail Trails dem Bucket hinzugefügt werden, besteht darin, Ihrer Bucket-Richtlinie eine `s3:PutObject` Ablehnungsmeldung hinzuzufügen. Wenn Sie zu einem späteren Zeitpunkt neue Objekte im Bucket speichern möchten, müssen Sie diese Deny `s3:PutObject`-Anweisung entfernen. Weitere Informationen finden Sie unter Objektoperationen und zu den IAMJSONRichtlinienelementen: Wirkung im IAMBenutzerhandbuch.

Löschen eines Buckets

Sie können einen leeren Amazon-S3-Bucket löschen. Berücksichtigen Sie Folgendes, bevor Sie einen Bucket löschen:

- Bucket-Namen sind eindeutig. Wenn Sie einen Bucket löschen, kann ein anderer AWS Benutzer den Namen verwenden.
- Wenn der Bucket eine statische Website hostet und Sie eine gehostete Zone von Amazon Route 53, wie unter Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne beschrieben, erstellt und konfiguriert haben, müssen Sie die gehosteten Route 53-Zoneneinstellungen bereinigen, die sich auf den Bucket beziehen. Weitere Informationen finden Sie unter Schritt 2: Löschen der von Route 53 gehosteten Zone.
- Wenn der Bucket Logdaten von Elastic Load Balancing (ELB) empfängt: Wir empfehlen, dass Sie die Übermittlung von ELB Logs an den Bucket beenden, bevor Sie ihn löschen. Erstellt ein anderer Benutzer nach dem Löschen des Buckets einen Bucket mit demselben Namen, dann könnten Ihre Protokolldaten potenziell an diesen Bucket übermittelt werden. Informationen zu ELB Zugriffs-Logs finden Sie unter Access Logs im User Guide for Classic Load Balancers und Access Logs im User Guide for Application Load Balancers.

Fehlerbehebung

Wenn Sie einen Amazon-S3-Bucket nicht löschen können, sollten Sie Folgendes berücksichtigen:

- Stellen Sie sicher, dass der Bucket leer ist – Sie können nur Buckets löschen, bei denen keine Objekte enthalten sind. Stellen Sie sicher, dass der Bucket leer ist.
- Stellen Sie sicher, dass keine Zugriffspunkte angefügt sind – Sie können nur Buckets löschen, denen keine Zugriffspunkte angefügt sind. Löschen Sie alle Zugriffspunkte, die dem Bucket angefügt sind, bevor Sie den Bucket löschen.
- AWS Organizations Dienststeuerungsrichtlinien (SCPs) und Ressourcensteuerungsrichtlinien (RCPs) — SCPs und RCPs kann die Löschberechtigung für einen Bucket verweigern. Weitere Informationen finden Sie unter Dienststeuerungsrichtlinien und Ressourcensteuerungsrichtlinien im AWS Organizations Benutzerhandbuch.
- s3: DeleteBucket Berechtigungen — Wenn Sie einen Bucket nicht löschen können, erkundigen Sie sich bei Ihrem IAM Administrator, ob Sie über die `s3:DeleteBucket` entsprechenden

Berechtigungen verfügen. Informationen zum Anzeigen oder Aktualisieren von IAM Berechtigungen finden Sie im IAMBenutzerhandbuch unter Ändern von Berechtigungen für einen IAM Benutzer.

- s3: Anweisung DeleteBucket verweigern — Wenn Ihre IAM Richtlinie über `s3:DeleteBucket` Berechtigungen verfügt und Sie einen Bucket nicht löschen können, enthält die Bucket-Richtlinie möglicherweise eine Ablehnungsaussage fürs`3:DeleteBucket`. Buckets, die von erstellt wurden ElasticBeanstalk , verfügen standardmäßig über eine Richtlinie, die diese Aussage enthält. Bevor Sie den Bucket löschen können, müssen Sie diese Anweisung oder die Bucket-Richtlinie löschen.

> **Important**
>
> Bucket-Namen sind eindeutig. Wenn Sie einen Bucket löschen, kann ein anderer AWS Benutzer den Namen verwenden. Wenn Sie weiterhin denselben Bucket-Namen verwenden wollen, sollten Sie den Bucket nicht löschen. Wir empfehlen, den Bucket zu leeren und beizubehalten.

Verwenden der S3-Konsole

Einen S3-Bucket löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets die Option neben dem Namen des Buckets aus, den Sie löschen möchten, und wählen Sie dann oben auf der Seite Delete (Löschen) aus.
3. Bestätigen Sie auf der Seite Delete bucket (Bucket löschen), dass Sie den Bucket löschen möchten. Geben Sie dazu den Bucket-Namen in das Textfeld ein und wählen Sie Delete bucket (Bucket löschen).

> **Note**
>
> Wenn der Bucket Objekte enthält, leeren Sie den Bucket, bevor Sie ihn löschen, indem Sie in der Fehlermeldung This bucket is not empty (Dieser Bucket ist nicht leer) den Link für eine leere Bucket-Konfiguration auswählen und die Anweisungen auf der Seite Empty bucket (Bucket leeren) befolgen. Kehren Sie dann zur Seite Delete bucket (Bucket löschen) zurück und löschen Sie den Bucket.

4. Wenn Sie überprüfen möchten, ob Sie den Bucket gelöscht haben, öffnen Sie die Buckets-Liste und geben den Namen des Buckets ein, den Sie gelöscht haben. Wenn der Buckets nicht gefunden wird, war der Löschvorgang erfolgreich.

Verwendung des AWS SDK for Java

Das folgende Beispiel zeigt Ihnen, wie Sie einen Bucket mit dem AWS SDK für Java löschen. Zuerst löscht der Code Objekte im Bucket, und dann löscht er das Bucket. Informationen zu anderen AWS SDKs finden Sie unter Tools für Amazon Web Services.

Java

Das folgende Java-Beispiel löscht einen Bucket mit Objekten. Zuerst löscht das Beispiel alle Objekte und dann den Bucket. Das Beispiel funktioniert für Buckets mit oder ohne aktiviertem Versioning.

> **Note**
>
> Bei Buckets ohne aktiviertes Versioning können Sie alle Objekte direkt löschen und danach den Bucket löschen. Bei Buckets mit aktiviertem Versioning müssen Sie zuerst alle Objektversionen löschen, bevor Sie den Bucket löschen.

Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.Iterator;

public class DeleteBucket2 {
```

```
   public static void main(String[] args) {
       Regions clientRegion = Regions.DEFAULT_REGION;
       String bucketName = "*** Bucket name ***";

       try {
           AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                   .withCredentials(new ProfileCredentialsProvider())
                   .withRegion(clientRegion)
                   .build();

           // Delete all objects from the bucket. This is sufficient
           // for unversioned buckets. For versioned buckets, when you attempt to
delete
           // objects, Amazon S3 inserts
           // delete markers for all objects, but doesn't delete the object
versions.
           // To delete objects from versioned buckets, delete all of the object
versions
           // before deleting
           // the bucket (see below for an example).
           ObjectListing objectListing = s3Client.listObjects(bucketName);
           while (true) {
               Iterator<S3ObjectSummary> objIter =
objectListing.getObjectSummaries().iterator();
               while (objIter.hasNext()) {
                   s3Client.deleteObject(bucketName, objIter.next().getKey());
               }

               // If the bucket contains many objects, the listObjects() call
               // might not return all of the objects in the first listing. Check
to
               // see whether the listing was truncated. If so, retrieve the next
page of
               // objects
               // and delete them.
               if (objectListing.isTruncated()) {
                   objectListing = s3Client.listNextBatchOfObjects(objectListing);
               } else {
                   break;
               }
           }

           // Delete all object versions (required for versioned buckets).
```

```
            VersionListing versionList = s3Client.listVersions(new
 ListVersionsRequest().withBucketName(bucketName));
            while (true) {
                Iterator<S3VersionSummary> versionIter =
 versionList.getVersionSummaries().iterator();
                while (versionIter.hasNext()) {
                    S3VersionSummary vs = versionIter.next();
                    s3Client.deleteVersion(bucketName, vs.getKey(),
 vs.getVersionId());
                }

                if (versionList.isTruncated()) {
                    versionList = s3Client.listNextBatchOfVersions(versionList);
                } else {
                    break;
                }
            }

            // After all objects and object versions are deleted, delete the bucket.
            s3Client.deleteBucket(bucketName);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
 couldn't
            // parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Mit dem AWS CLI

Sie können einen Bucket, der Objekte enthält, mit dem löschen, AWS CLI wenn für ihn die Versionsverwaltung nicht aktiviert ist. Wenn Sie einen Bucket löschen, der Objekte enthält, werden alle Objekte in dem Bucket dauerhaft gelöscht. Dazu gehören auch Objekte, die in die `S3 Glacier`-Speicherklasse übergegangen sind.

Wenn für Ihren Bucket die Versionierung nicht aktiviert ist, können Sie den AWS CLI Befehl `rb` (remove bucket) mit dem `--force` Parameter verwenden, um den Bucket und alle darin enthaltenen Objekte zu löschen. Dieser Befehl löscht zuerst alle Objekte und dann den Bucket.

Wenn das Versioning aktiviert ist, werden versionierte Objekte in diesem Prozess nicht gelöscht. Dadurch würde das Löschen des Buckets fehlschlagen, da der Bucket nicht leer wäre. Informationen zum Löschen versionierter Objekte finden Sie unter Löschen von Objektversionen.

```
$ aws s3 rb s3://bucket-name --force
```

Weitere Informationen finden Sie unter Verwenden von S3-Befehlen auf hoher Ebene mit dem AWS Command Line Interface im AWS Command Line Interface Benutzerhandbuch.

Arbeiten mit Mountpoint für Amazon S3

Mountpoint für Amazon S3 ist ein Open-Source-Dateiclient mit hohem Durchsatz, um ein Amazon-S3-Bucket als lokales Dateisystem zu mounten. Mit Mountpoint können Ihre Anwendungen über Dateisystem-Operationen wie Öffnen und Lesen auf Objekte zugreifen, die in Amazon S3 gespeichert sind. Mountpoint übersetzt diese Operationen automatisch in API S3-Objekt-Aufrufe, sodass Ihre Anwendungen über eine Dateischnittstelle auf den elastischen Speicher und Durchsatz von Amazon S3 zugreifen können.

Mountpoint für Amazon S3 ist generell für den produktiven Einsatz in Ihren großen leseintensiven Anwendungen verfügbar: Data Lakes, Training für maschinelles Lernen, Rendern von Bildern, Simulation autonomer Fahrzeuge, Extrahieren, Transformieren und Laden (ETL) und mehr.

Mountpoint unterstützt grundlegende Dateisystem-Operationen und kann Dateien mit einer Größe von bis zu 5 TB lesen. Diese Lösung kann vorhandene Dateien auflisten und lesen sowie neue erstellen. Sie kann keine vorhandenen Dateien ändern oder Verzeichnisse löschen und unterstützt keine symbolischen Links oder Dateisperren. Mountpoint ist ideal für Anwendungen, die nicht alle Funktionen eines gemeinsam genutzten Dateisystems und POSIX Berechtigungen benötigen, sondern den elastischen Durchsatz von Amazon S3 zum Lesen und Schreiben großer S3-Datensätze benötigen. Einzelheiten finden Sie unter Verhalten des Mountpoint-Dateisystems auf GitHub. Für Workloads, die volle POSIX Unterstützung benötigen, empfehlen wir Amazon FSx for Lustre und dessen Unterstützung für die Verknüpfung von S3-Buckets.

Mountpoint für Amazon S3 ist nur verfügbar für Linux Betriebssysteme. Sie können Mountpoint für den Zugriff auf S3-Objekte in allen Speicherklassen außer S3 Glacier Flexible Retrieval, S3 Glacier

Deep Archive, S3 Intelligent-Tiering Archive Access Tier und S3 Intelligent-Tiering Deep Archive Access Tier verwenden.

Themen

- Mountpoint installieren
- Konfiguration und Verwendung von Mountpoint
- Fehlerbehebung bei Mountpoint

Mountpoint installieren

Sie können vorgefertigte Pakete von Mountpoint für Amazon S3 über die Befehlszeile herunterladen und installieren. Die Anweisungen zum Herunterladen und Installieren von Mountpoint variieren je nachdem Linux Betriebssystem, das Sie verwenden.

Themen

- RPMbasierte Distributionen (Amazon Linux, Fedora, CentOS, RHEL)
- DEBbasierte Distributionen (Debian, Ubuntu)
- Sonstige Linux Verteilungen
- Überprüfen der Signatur des Pakets für Mountpoint für Amazon S3

RPMbasierte Distributionen (Amazon Linux, Fedora, CentOS, RHEL)

1. Kopieren Sie den folgenden Download URL für Ihre Architektur.

   x86_64:

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.rpm
   ```

   ARM64 (Graviton):

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.rpm
   ```

2. Laden Sie das Mountpoint für Amazon-S3-Paket herunter. *download-link*Ersetzen Sie ihn durch den entsprechenden Download URL aus dem vorherigen Schritt.

   ```
   wget download-link
   ```

3. (Optional) Überprüfen Sie die Authentizität und Integrität der heruntergeladenen Datei. Kopieren Sie zunächst die entsprechende Signatur URL für Ihre Architektur.

   x86_64:

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.rpm.asc
   ```

   ARM64 (Graviton):

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.rpm.asc
   ```

   Sehen Sie sich als Nächstes Überprüfen der Signatur des Pakets für Mountpoint für Amazon S3 an.

4. Installieren Sie das Paket mit dem folgenden Befehl:

   ```
   sudo yum install ./mount-s3.rpm
   ```

5. Stellen Sie sicher, dass Mountpoint erfolgreich installiert wurde, indem Sie den folgenden Befehl eingeben:

   ```
   mount-s3 --version
   ```

   Die Ausgabe sollte folgendermaßen oder ähnlich aussehen:

   ```
   mount-s3 1.3.1
   ```

DEBbasierte Distributionen (Debian, Ubuntu)

1. Kopieren Sie den Download URL für Ihre Architektur.

   x86_64:

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.deb
   ```

   ARM64 (Graviton):

   ```
   https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.deb
   ```

2. Laden Sie das Mountpoint für Amazon-S3-Paket herunter. *download-link*Ersetzen Sie ihn durch den entsprechenden Download URL aus dem vorherigen Schritt.

```
wget download-link
```

3. (Optional) Überprüfen Sie die Authentizität und Integrität der heruntergeladenen Datei. Kopieren Sie zunächst die Signatur URL für Ihre Architektur.

   x86_64:

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.deb.asc
```

   ARM64 (Graviton):

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.deb.asc
```

   Sehen Sie sich als Nächstes Überprüfen der Signatur des Pakets für Mountpoint für Amazon S3 an.

4. Installieren Sie das Paket mit dem folgenden Befehl:

```
sudo apt-get install ./mount-s3.deb
```

5. Stellen Sie sicher, dass Mountpoint für Amazon S3 erfolgreich installiert wurde, indem Sie den folgenden Befehl ausführen:

```
mount-s3 --version
```

   Die Ausgabe sollte folgendermaßen oder ähnlich aussehen:

```
mount-s3 1.3.1
```

Sonstige Linux Verteilungen

1. Schlagen Sie in der Dokumentation Ihres Betriebssystems nach, um die erforderlichen Pakete FUSE und `libfuse2` zu installieren.
2. Kopieren Sie den Download URL für Ihre Architektur.

x86_64:

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.tar.gz
```

ARM64 (Graviton):

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.tar.gz
```

3. Laden Sie das Mountpoint für Amazon-S3-Paket herunter. *download-link*Ersetzen Sie ihn durch den entsprechenden Download URL aus dem vorherigen Schritt.

```
wget download-link
```

4. (Optional) Überprüfen Sie die Authentizität und Integrität der heruntergeladenen Datei. Kopieren Sie zunächst die Signatur URL für Ihre Architektur.

   x86_64:

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.tar.gz.asc
```

   ARM64 (Graviton):

```
https://s3.amazonaws.com/mountpoint-s3-release/latest/arm64/mount-s3.tar.gz.asc
```

   Sehen Sie sich als Nächstes Überprüfen der Signatur des Pakets für Mountpoint für Amazon S3 an.

5. Installieren Sie das Paket mit dem folgenden Befehl:

```
sudo mkdir -p /opt/aws/mountpoint-s3 && sudo tar -C /opt/aws/mountpoint-s3 -xzf ./
mount-s3.tar.gz
```

6. Fügen Sie das `mount-s3`-Binary Ihrer `PATH`-Umgebungsvariablen hinzu. Fügen Sie in Ihrer `$HOME/.profile`-Datei die folgende Zeile an:

```
export PATH=$PATH:/opt/aws/mountpoint-s3/bin
```

   Speichern Sie die `.profile`-Datei und führen Sie den folgenden Befehl aus:

```
source $HOME/.profile
```

7. Stellen Sie sicher, dass Mountpoint für Amazon S3 erfolgreich installiert wurde, indem Sie den folgenden Befehl ausführen:

   ```
   mount-s3 --version
   ```

   Die Ausgabe sollte folgendermaßen oder ähnlich aussehen:

   ```
   mount-s3 1.3.1
   ```

Überprüfen der Signatur des Pakets für Mountpoint für Amazon S3

1. Installieren GnuPG (der gpg Befehl). Es ist erforderlich, um die Authentizität und Integrität eines heruntergeladenen Mountpoint for Amazon S3-Pakets zu überprüfen. GnuPG ist standardmäßig auf Amazon Linux Amazon Machine Images (AMIs) installiert. Nach der InstallationGnuPG, fahren Sie mit Schritt 2 fort.
2. Laden Sie den öffentlichen Mountpoint-Schlüssel herunter, indem Sie den folgenden Befehl ausführen:

   ```
   wget https://s3.amazonaws.com/mountpoint-s3-release/public_keys/KEYS
   ```

3. Importieren Sie den öffentlichen Mountpoint-Schlüssel in Ihren Schlüsselbund, indem Sie den folgenden Befehl ausführen:

   ```
   gpg --import KEYS
   ```

4. Überprüfen Sie den Fingerabdruck des öffentlichen Mountpoint-Schlüssels, indem Sie den folgenden Befehl ausführen:

   ```
   gpg --fingerprint mountpoint-s3@amazon.com
   ```

   Vergewissern Sie sich, dass die angezeigte Fingerabdruck-Zeichenfolge mit der folgenden Zeichenfolge übereinstimmt:

   ```
   673F E406 1506 BB46 9A0E  F857 BE39 7A52 B086 DA5A
   ```

Wenn die Zeichenfolge für den Fingerabdruck nicht übereinstimmt, stellen Sie die Installation von Mountpoint nicht fertig und wenden Sie sich an AWS Support.

5. Laden Sie die Paket-Signaturdatei herunter. Ersetzen Sie *signature-link* durch den entsprechenden Signaturlink aus den vorherigen Abschnitten.

```
wget signature-link
```

6. Überprüfen Sie die Signatur des heruntergeladenen Pakets, indem Sie den folgenden Befehl ausführen. Ersetzen Sie *signature-filename* durch den Dateinamen aus dem vorherigen Schritt.

```
gpg --verify signature-filename
```

   Geben Sie beispielsweise bei RPM basierten Distributionen, einschließlich Amazon Linux, den folgenden Befehl ein:

```
gpg --verify mount-s3.rpm.asc
```

7. Die Ausgabe sollte den Text `Good signature` enthalten. Wenn die Ausgabe den Text `BAD signature` enthält, laden Sie die Mountpoint-Paketdatei erneut herunter und wiederholen Sie diese Schritte. Wenn das Problem weiterhin besteht, stellen Sie die Installation von Mountpoint nicht fertig und wenden Sie sich an AWS Support.

   Die Ausgabe kann eine Warnung über eine vertrauenswürdige Signatur enthalten. Dies deutet nicht auf ein Problem hin. Es bedeutet nur, dass Sie den öffentlichen Mountpoint-Schlüssel nicht unabhängig verifiziert haben.

Konfiguration und Verwendung von Mountpoint

Um Mountpoint für Amazon S3 verwenden zu können, benötigt Ihr Host gültige AWS Anmeldeinformationen mit Zugriff auf den Bucket oder die Buckets, die Sie mounten möchten. Informationen zu verschiedenen Authentifizierungsmethoden finden Sie unter Mountpoint Credentials auf AWS GitHub.

Sie können zu diesem Zweck beispielsweise einen neuen Benutzer und eine neue Rolle AWS Identity and Access Management (IAM) erstellen. Stellen Sie sicher, dass diese Rolle Zugriff auf den Bucket

oder die Buckets hat, die Sie mounten möchten. Sie können die IAM Rolle mit einem EC2 Instance-Profil an Ihre Amazon-Instance übergeben.

Themen

- Verwenden von Mountpoint für Amazon S3
- Konfigurieren von Caching in Mountpoint

Verwenden von Mountpoint für Amazon S3

Verwenden Sie Mountpoint für Amazon S3, um Folgendes zu tun:

1. Mounten Sie Buckets mit dem `mount-s3`-Befehl.

   Ersetzen Sie im folgenden Beispiel *`amzn-s3-demo-bucket`* durch den Namen Ihres S3-Buckets und ersetzen Sie *`~/mnt`* durch das Verzeichnis auf Ihrem Host, in dem Ihr S3-Bucket gemountet werden soll.

   ```
   mkdir ~/mnt
   mount-s3 amzn-s3-demo-bucket ~/mnt
   ```

   Da der Mountpoint-Client standardmäßig im Hintergrund ausgeführt wird, bietet Ihnen das *`~/mnt`*-Verzeichnis jetzt Zugriff auf die Objekte in Ihrem S3-Bucket.

2. Greifen Sie über Mountpoint auf die Objekte in Ihrem Bucket zu.

   Nachdem Sie Ihren Bucket lokal bereitgestellt haben, können Sie Common verwenden Linux Befehle wie `cat` oder`ls`, um mit Ihren S3-Objekten zu arbeiten. Mountpoint für Amazon S3 interpretiert Schlüssel in Ihrem S3-Bucket als Dateisystempfade. Dazu werden sie anhand des Schrägstrichs (`/`) getrennt. Wenn Sie zum Beispiel den Objektschlüssel `Data/2023-01-01.csv` in Ihrem Bucket haben, finden Sie ein Verzeichnis mit dem Namen `Data` in Ihrem Mountpoint-Dateisystem mit einer Datei namens `2023-01-01.csv` darin.

   Mountpoint for Amazon S3 implementiert bewusst nicht die vollständige POSIXStandardspezifikation für Dateisysteme. Mountpoint ist für Workloads optimiert, die über eine Dateisystemschnittstelle Lese- und Schreibzugriff mit hohem Durchsatz auf Daten benötigen, die in Amazon S3 gespeichert aber ansonsten nicht auf Dateisystemfunktionen angewiesen sind. Weitere Informationen finden Sie unter Dateisystemverhalten von Mountpoint for Amazon S3 auf GitHub. Kunden, die eine umfassendere Dateisystemsemantik benötigen,

sollten andere AWS Dateidienste wie Amazon Elastic File System (AmazonEFS) oder Amazon in Betracht ziehen. FSx

3. Heben Sie das Mounting Ihres Bucket mit dem `umount`-Befehl auf. Mit diesem Befehl wird Mounting Ihres S3-Bucket aufgehoben und Mountpoint wird beendet.

   Um den folgenden Beispielbefehl zu verwenden, ersetzen Sie *~/mnt* durch das Verzeichnis auf Ihrem Host, in dem Ihr S3-Bucket gemountet ist.

   ```
   umount ~/mnt
   ```

   > **Note**
   >
   > Wenn Sie eine Liste der Optionen für diesen Befehl erhalten möchten, führen Sie `umount --help` aus.

Weitere Informationen zur Mountpoint-Konfiguration finden Sie unter S3-Bucket-Konfiguration und Dateisystemkonfiguration auf GitHub.

Konfigurieren von Caching in Mountpoint

Mountpoint für Amazon S3 unterstützt verschiedene Arten von Daten-Caching. Um wiederholte Leseanfragen zu beschleunigen, können Sie sich für Folgendes entscheiden:

- Lokaler Cache — Sie können einen lokalen Cache in Ihrem EC2 Amazon-Instance-Speicher oder einem Amazon Elastic Block Store-Volume verwenden. Wenn Sie wiederholt dieselben Daten aus derselben Compute-Instance lesen und Sie in Ihrem lokalen Instance-Speicher ungenutzten Speicherplatz für den wiederholt gelesenen Datensatz haben, sollten Sie sich für einen lokalen Cache entscheiden.
- Gemeinsamer Cache — Sie können einen gemeinsam genutzten Cache auf S3 Express One Zone verwenden. Wenn Sie wiederholt kleine Objekte aus mehreren Recheninstanzen lesen oder wenn Sie die Größe Ihres wiederholt gelesenen Datensatzes nicht kennen und von der Elastizität der Cachegröße profitieren möchten, sollten Sie sich für den Shared Cache entscheiden. Sobald Sie sich dafür entschieden haben, speichert Mountpoint Objekte mit einer Größe von bis zu einem Megabyte in einem Verzeichnis-Bucket, der S3 Express One Zone verwendet.
- Kombinierter lokaler und gemeinsam genutzter Cache — Wenn Sie ungenutzten Speicherplatz in Ihrem lokalen Cache haben, aber auch einen gemeinsamen Cache für mehrere Instanzen haben

möchten, können Sie sich sowohl für einen lokalen Cache als auch für einen gemeinsamen Cache entscheiden.

Das Caching in Mountpoint ist ideal für Anwendungsfälle, in denen Sie wiederholt dieselben Daten lesen, die sich während der mehrfachen Lesevorgänge nicht ändern. Sie können das Caching beispielsweise bei Machine-Learning-Trainingsaufgaben verwenden, bei denen ein Trainingsdatensatz mehrmals gelesen werden muss, um die Modellgenauigkeit zu verbessern.

Weitere Informationen zur Konfiguration von Caching in Mountpoint finden Sie in den folgenden Beispielen.

Themen

- Lokaler Cache
- Gemeinsamer Cache
- Kombinierter lokaler und gemeinsam genutzter Cache

Lokaler Cache

Mit dem Flag können Sie sich für einen lokalen Cache entscheiden. `--cache` *CACHE_PATH* Ersetzen Sie im folgenden Beispiel *CACHE_PATH* durch den Dateipfad zu dem Verzeichnis, in dem Sie Ihre Daten zwischenspeichern möchten. Ersetzen *amzn-s3-demo-bucket* mit dem Namen Ihres S3-Buckets und ersetzen *~/mnt* mit dem Verzeichnis auf Ihrem Host, in dem Ihr S3-Bucket gemountet werden soll.

```
mkdir ~/mnt
mount-s3 --cache CACHE_PATH amzn-s3-demo-bucket ~/mnt
```

Wenn Sie sich beim Mounten eines S3-Buckets für lokales Caching entscheiden, erstellt Mountpoint ein leeres Unterverzeichnis am konfigurierten Cache-Speicherort, falls dieses Unterverzeichnis noch nicht existiert. Wenn Sie einen Bucket zum ersten Mal mounten und wenn Sie ihn unmounten, löscht Mountpoint den Inhalt des lokalen Caches.

> **Important**
>
> Wenn Sie das lokale Caching aktivieren, speichert Mountpoint unverschlüsselten Objektinhalt aus Ihrem gemounteten S3-Bucket an dem beim Mount angegebenen lokalen Cache-

Speicherort. Um Ihre Daten zu schützen, sollten Sie den Zugriff auf den Datencache-Speicherort mithilfe von Zugriffskontrollmechanismen für das Dateisystem einschränken.

Gemeinsamer Cache

Wenn Sie wiederholt kleine Objekte (bis zu 1 MB) aus mehreren Compute-Instances lesen oder die Größe des Datensatzes, den Sie wiederholt lesen, häufig die Größe Ihres lokalen Caches überschreitet, sollten Sie einen gemeinsamen Cache in S3 Express One Zone verwenden. Wenn Sie dieselben Daten wiederholt aus mehreren Instanzen lesen, verbessert dies die Latenz, da redundante Anfragen an Ihren bereitgestellten S3-Bucket vermieden werden.

Sobald Sie sich für den Shared Cache entschieden haben, zahlen Sie für die Daten, die in Ihrem Directory-Bucket in S3 Express One Zone zwischengespeichert sind. Sie zahlen auch für Anfragen, die sich auf Ihre Daten im Verzeichnis-Bucket in S3 Express One Zone beziehen. Weitere Informationen finden Sie unter Amazon S3 – Preise. Mountpoint löscht niemals zwischengespeicherte Objekte aus Verzeichnis-Buckets. Um Ihre Speicherkosten zu verwalten, sollten Sie eine Lifecycle-Richtlinie für Ihren Verzeichnis-Bucket einrichten, sodass Amazon S3 die zwischengespeicherten Daten in S3 Express One Zone nach einem von Ihnen angegebenen Zeitraum abläuft. Weitere Informationen finden Sie unter Mountpoint für Amazon S3 S3-Caching-Konfiguration unter GitHub.

Wenn Sie beim Mounten eines Allzweck-Buckets in Ihrer Compute-Instance das Caching in S3 Express One Zone aktivieren möchten, verwenden Sie das `--cache-xz` Flag und geben Sie einen Verzeichnis-Bucket als Cache-Speicherort an. Ersetzen Sie im folgenden Beispiel den *`user input placeholders`*.

```
mount-s3 amzn-s3-demo-bucket ~/mnt --cache-xz amzn-s3-demo-bucket--usw2-az1--x-s3
```

Kombinierter lokaler und gemeinsam genutzter Cache

Wenn Sie über ungenutzten Speicherplatz auf Ihrer Instance verfügen, aber auch einen gemeinsamen Cache für mehrere Instances verwenden möchten, können Sie sich sowohl für einen lokalen Cache als auch für einen gemeinsamen Cache entscheiden. Mit dieser Caching-Konfiguration können Sie redundante Leseanforderungen von derselben Instanz an den gemeinsam genutzten Cache im Verzeichnis-Bucket vermeiden, wenn die erforderlichen Daten im lokalen Speicher zwischengespeichert werden. Dadurch können die Anforderungskosten gesenkt und die Leistung verbessert werden.

Um sich beim Mounten eines S3-Buckets sowohl für einen lokalen Cache als auch für einen gemeinsam genutzten Cache zu entscheiden, geben Sie beide Cache-Speicherorte mithilfe der `--cache-xz` Flags `--cache` und an. Um das folgende Beispiel zu verwenden, um sich sowohl für einen lokalen als auch für einen gemeinsamen Cache zu entscheiden, ersetzen Sie das *`user input placeholders`*.

```
mount -s3 amzn-s3-demo-bucket ~/mnt --cache /path/to/mountpoint/cache --cache -xz amzn-s3-demo-bucket--usw2-az1--x-s3
```

Weitere Informationen finden Sie unter Mountpoint für Amazon S3 S3-Caching-Konfiguration unter GitHub.

> **Important**
>
> Wenn Sie Shared Caching aktivieren, kopiert Mountpoint den Objektinhalt aus Ihrem bereitgestellten S3-Bucket in den S3-Verzeichnis-Bucket, den Sie als gemeinsamen Cache-Speicherort angeben, sodass er für jeden Anrufer mit Zugriff auf den S3-Verzeichnis-Bucket zugänglich ist. Um Ihre zwischengespeicherten Daten zu schützen, sollten Sie die folgenden Regeln beachten, Bewährte Methoden für die Sicherheit in Amazon S3 um sicherzustellen, dass Ihre Buckets die richtigen Richtlinien verwenden und nicht öffentlich zugänglich sind. Sie sollten einen Verzeichnis-Bucket verwenden, der dem gemeinsamen Caching von Mountpoint gewidmet ist, und nur Mountpoint-Clients Zugriff gewähren.

Fehlerbehebung bei Mountpoint

Mountpoint für Amazon S3 wird unterstützt von. AWS Support Wenn Sie Hilfe benötigen, wenden Sie sich an das AWS Support -Zentrum.

Sie können Mountpoint-Probleme auch unter überprüfen und einreichen GitHub.

Wenn Sie in diesem Projekt ein potenzielles Sicherheitsproblem entdecken, bitten wir Sie über unsere Seite zur Meldung von Schwachstelle AWS Security darüber zu informieren. Erstellen Sie kein öffentliches GitHub Problem.

Wenn sich Ihre Anwendung mit Mountpoint unerwartet verhält, können Sie Ihre Protokollinformationen überprüfen, um das Problem zu diagnostizieren.

Protokollierung

Standardmäßig sendet Mountpoint Protokollinformationen mit hohem Schweregrad an syslog.

Um Protokolle auf den modernsten Geräten anzuzeigen Linux Distributionen, einschließlich Amazon Linux, führen den folgenden `journald` Befehl aus:

```
journalctl -e SYSLOG_IDENTIFIER=mount-s3
```

Auf anderen Linux Auf Systemen werden `syslog` Einträge wahrscheinlich in eine Datei wie geschrieben`/var/log/syslog`.

Sie können diese Protokolle verwenden, um Fehler in Ihrer Anwendung zu beheben. Wenn Ihre Anwendung beispielsweise versucht, eine vorhandene Datei zu überschreiben, schlägt der Vorgang fehl und im Protokoll wird eine Zeile angezeigt, die der folgenden ähnelt:

```
[WARN] open{req=12 ino=2}: mountpoint_s3::fuse: open failed: inode error: inode 2 (full
 key "README.md") is not writable
```

Weitere Informationen finden Sie unter Mountpoint for Amazon S3 Logging on GitHub.

Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration

Amazon S3 Transfer Acceleration ist eine Funktion auf Bucket-Ebene, die die schnelle, einfache und sichere Übertragung von Dateien über große Entfernungen zwischen Ihrem Client und einem S3-Bucket ermöglicht. Transfer Acceleration wurde entwickelt, um Übertragungsgeschwindigkeiten aus der ganzen Welt in S3-Buckets zu optimieren. Transfer Acceleration nutzt die weltweit verteilten Edge-Standorte von Amazon CloudFront. Sobald die Daten an einem Edge-Standort eingehen, werden sie über einen optimierten Netzwerkpfad an Ihren Amazon S3-Bucket weitergeleitet.

Wenn Sie Transfer Acceleration verwenden, fallen möglicherweise zusätzliche Gebühren für die Datenübertragung an. Weitere Informationen zu Preisen finden Sie unter Amazon S3-Preise.

Gründe für die Nutzung von Amazon S3 Transfer Acceleration

Es gibt verschiedene Gründe für die Verwendung von Transfer Acceleration auf einem Bucket:

- Ihre Kunden laden in einen zentralen Bucket aus der ganzen Welt hoch.
- Sie übertragen regelmäßig mehrere Gigabyte bis Terabyte von Daten über mehrere Kontinente hinweg.

- Sie können beim Hochladen in Amazon S3 nicht die gesamte verfügbare Bandbreite über das Internet nutzen.

Weitere Informationen darüber, wann Sie Transfer Acceleration verwenden sollten, finden Sie unter Amazon S3 FAQs.

Anforderungen für die Verwendung von Transfer Acceleration

Folgendes ist erforderlich, wenn Sie Transfer Acceleration für einen S3-Bucket verwenden:

- Transfer Acceleration wird nur bei virtuell gehosteten Anfragen unterstützt. Weitere Informationen zu Anfragen im virtuell gehosteten Stil finden Sie unter Anfragen stellen mithilfe von REST API in der Amazon S3 API S3-Referenz.
- Der Name des für Transfer Acceleration verwendeten Buckets muss DNS -konform sein und darf keine Punkte („.“) enthalten.
- Transfer Acceleration muss im Bucket aktiviert sein. Weitere Informationen finden Sie unter Aktivieren und Verwenden von S3 Transfer Acceleration.

  Nach der Aktivierung von Transfer Acceleration für einen Bucket kann es bis zu 20 Minuten dauern, bis sich die Datenübertragungsgeschwindigkeit in den Bucket erhöht.

  > **Note**
  >
  > Transfer Acceleration wird zurzeit für Buckets in den folgenden Regionen unterstützt:
  >
  > - Asien-Pazifik (Tokyo) (ap-northeast-1)
  > - Asien-Pazifik (Seoul): (ap-northeast-2)
  > - Asien-Pazifik (Mumbai): (ap-south-1)
  > - Asien-Pazifik (Singapur): (ap-southeast-1)
  > - Asien-Pazifik (Sydney): (ap-southeast-2)
  > - Kanada (Zentral): (ca-central-1)
  > - Europa (Frankfurt) (eu-central-1)
  > - Europa (Irland) (eu-west-1)
  > - Europa (London) (eu-west-2)
  > - Europa (Paris) (eu-west-3)
  > - Südamerika (São Paulo) (sa-east-1)

- USA Ost (Nord-Virginia): (us-east-1)
- USA Ost (Ohio): (us-east-2)
- USA West (Nordkalifornien) (us-west-1)
- USA West (Oregon): (us-west-2)

- Um auf den Bucket zuzugreifen, der für Transfer Acceleration konfiguriert ist, müssen Sie den Endpunkt *bucketname*`.s3-accelerate.amazonaws.com` verwenden. Oder verwenden Sie den Dual-Stack-Endpunkt*bucketname*`.s3-accelerate.dualstack.amazonaws.com`, um eine Verbindung zum aktivierten Bucket herzustellen. IPv6 Sie können weiterhin die regulären Endpunkte für die Standarddatenübertragung verwenden.
- Sie müssen der Bucket-Eigentümer sein, um den Transfer Acceleration-Status festlegen zu können. Der Bucket-Eigentümer kann anderen Benutzern Berechtigungen erteilen, um ihnen zu gestatten, den Beschleunigungsstatus für einen Bucket einzurichten. Die `s3:PutAccelerateConfiguration`-Berechtigung gestattet Benutzern, Transfer Acceleration für einen Bucket zu aktivieren oder zu deaktivieren. Mit dieser `s3:GetAccelerateConfiguration` Berechtigung können Benutzer den Status Transfer Acceleration eines Buckets zurückgeben, der entweder oder `Enabled` `Suspended`.

In den folgenden Abschnitten wird beschrieben, wie Sie beginnen und Amazon S3 Transfer Acceleration für die Übertragung von Daten verwenden.

Themen

- Erste Schritte mit Amazon S3 Transfer Acceleration
- Aktivieren und Verwenden von S3 Transfer Acceleration
- Verwenden des Amazon S3 Transfer Acceleration Speed Comparison-Tools

Erste Schritte mit Amazon S3 Transfer Acceleration

Sie können Amazon S3 Transfer Acceleration für die schnelle, einfache und sichere Übertragung von Dateien über große Entfernungen zwischen Ihrem Client und einem S3-Bucket nutzen. Transfer Acceleration nutzt die weltweit verteilten Edge-Standorte in Amazon CloudFront. Sobald die Daten an einem Edge-Standort eingehen, werden sie über einen optimierten Netzwerkpfad an Ihren Amazon S3-Bucket weitergeleitet.

Um mit der Verwendung von Amazon S3 Transfer Acceleration zu beginnen, führen Sie die folgenden Schritte aus:

1. Aktivieren der Transfer Acceleration auf einem Bucket

   Sie können Transfer Acceleration für einen Bucket auf eine der folgenden Arten aktivieren:

   - Verwenden der Amazon S3-Konsole
   - Verwenden Sie den REST API PUTBucket Accelerate-Vorgang.
   - Verwenden Sie das AWS CLI und AWS SDKs. Weitere Informationen finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS SDKs in der Amazon S3 API S3-Referenz.

   Weitere Informationen finden Sie unter Aktivieren und Verwenden von S3 Transfer Acceleration.

   > **Note**
   >
   > Damit Ihr Bucket mit der Übertragungsbeschleunigung funktioniert, muss der Bucket-Name den DNS Benennungsanforderungen entsprechen und darf keine Punkte („.“) enthalten.

2. Übertragen von Daten in den und aus dem beschleunigungsfähigen Bucket

   Verwenden Sie einen der folgenden s3-beschleunigten Endpunkt-Domänennamen:

   - Um auf einen beschleunigungsfähigen Bucket zuzugreifen, nutzen Sie *`bucketname`*`.s3-accelerate.amazonaws.com`.
   - Um auf einen Bucket mit aktivierter Beschleunigung zuzugreifen, verwenden Sie. IPv6 *`bucketname`*`.s3-accelerate.dualstack.amazonaws.com`

     Amazon S3 S3-Dual-Stack-Endpunkte unterstützen Anfragen an S3-Buckets über und. IPv6 IPv4 Der Transfer Acceleration-Dual-Stack-Endpunkt verwendet nur den virtuellen Hosting-Endpunktnamen. Weitere Informationen finden Sie unter Anfragen an Amazon S3 stellen IPv6 in der Amazon API S3-Referenz und Verwenden von Amazon S3 S3-Dual-Stack-Endpunkten in der Amazon S3 API S3-Referenz.

   > **Note**
   >
   > Ihre Anwendung zur Datenübertragung muss eine der beiden folgenden Endpunkt-Typen verwenden, um auf den Bucket zugreifen und die Datenübertragung

beschleunigen zu können: `.s3-accelerate.amazonaws.com` oder `.s3-accelerate.dualstack.amazonaws.com` für den Dual-Stack-Endpunkt. Wenn Sie die Standarddatenübertragung verwenden möchten, können Sie weiterhin die regulären Endpunkte nutzen.

Sie können Ihr Amazon S3 PUT GET S3-Objekt und Ihre Objektanforderungen auf den Domainnamen des `s3-accelerate` Endpunkts verweisen, nachdem Sie Transfer Acceleration aktiviert haben. Nehmen wir zum Beispiel an, Sie haben derzeit eine REST API Anwendung, die PUTObject verwendet und den Hostnamen `mybucket.s3.us-east-1.amazonaws.com` in der PUT Anfrage verwendet. Um PUT zu beschleunigen, ändern Sie den Hostnamen in Ihrer Anfrage in `mybucket.s3-accelerate.amazonaws.com`. Um wieder die Standard-Upload-Geschwindigkeit zu verwenden, ändern Sie den Namen zurück in `mybucket.s3.us-east-1.amazonaws.com`.

Nachdem Transfer Acceleration aktiviert wurde, kann es bis zu 20 Minuten dauern, bis Sie den Leistungsvorteil wahrnehmen. Der beschleunigte Endpunkt steht zur Verfügung, sobald Sie Transfer Acceleration aktivieren.

Sie können den Accelerate-Endpunkt in den AWS CLI AWS SDKs, und anderen Tools verwenden, die Daten zu und von Amazon S3 übertragen. Wenn Sie den verwenden AWS SDKs, verwenden einige der unterstützten Sprachen ein Accelerate-Endpoint-Client-Konfigurationsflag, sodass Sie den Endpunkt für Transfer Acceleration nicht explizit auf festlegen müssen*bucketname*`.s3-accelerate.amazonaws.com`. Beispiele für das Flag für eine Client-Konfiguration mit beschleunigtem Endpunkt finden Sie unter Aktivieren und Verwenden von S3 Transfer Acceleration.

Sie können alle Amazon S3-Vorgänge über die Endpunkte der Transfer Acceleration mit Ausnahme der Folgenden verwenden:

- GETService (Buckets auflisten)
- PUTBucket (Bucket erstellen)
- DELETEEimer

Außerdem unterstützt Amazon S3 Transfer Acceleration keine regionsübergreifenden Kopien mit PUTObject — Copy.

Aktivieren und Verwenden von S3 Transfer Acceleration

Sie können Amazon S3 Transfer Acceleration verwenden, um Dateien schnell und sicher über große Entfernungen zwischen Ihrem Client und einem S3-Bucket zu übertragen. Sie können Transfer Acceleration mit der S3-Konsole, dem AWS Command Line Interface (AWS CLI) oder dem aktivieren AWS SDKs. API

Dieser Abschnitt bietet Beispiele für die Aktivierung von Amazon S3 Transfer Acceleration und die Verwendung des beschleunigten Endpunkts für den aktivierten Bucket.

Weitere Informationen zu den Anforderungen für Transfer Acceleration finden Sie unter Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration.

Verwenden der S3-Konsole

> **Note**
>
> Wenn Sie beschleunigte und nicht beschleunigte Upload-Geschwindigkeiten vergleichen möchten, öffnen Sie das Amazon S3 Transfer Acceleration Speed Comparison-Tool.
> Das Tool zum Geschwindigkeitsvergleich verwendet mehrteiligen Upload, um eine Datei AWS-Regionen mit und ohne Amazon S3 S3-Übertragungsbeschleunigung von Ihrem Browser in verschiedene zu übertragen. Sie können die Upload-Geschwindigkeit für direkte Uploads vergleichen und beschleunigte Uploads nach Region übertragen.

Die Transfer Acceleration für einen S3-Bucket aktivieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie Transfer Acceleration aktivieren möchten.
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie unter Transfer acceleration (Beschleunigung übertragen) die Option Edit (Bearbeiten) aus.
5. Wählen Sie Enable (Aktivieren) und wählen Sie Save changes (Änderungen speichern) aus.

Zugriff auf beschleunigte Datenübertragungen

1. Nachdem Amazon S3 die Übertragungsbeschleunigung für Ihren Bucket aktiviert hat, sehen Sie sich den Tab Properties (Eigenschaften) für den Bucket an.
2. Unter Transfer acceleration (Übertragungsbeschleunigung) wird unter Accelerated endpoint (beschleunigter Endpunkt) der Endpunkt der Übertragungsbeschleunigung für Ihren Bucket angezeigt. Verwenden Sie diesen Endpunkt, um auf beschleunigte Datenübertragungen von und zu Ihrem Bucket zuzugreifen.

   Wenn Sie die Transfer Acceleration aussetzen, funktioniert der beschleunigte Endpunkt nicht mehr.

Verwenden Sie den AWS CLI

Im Folgenden finden Sie Beispiele für AWS CLI Befehle, die für die Übertragungsbeschleunigung verwendet werden. Anweisungen zur Einrichtung von finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS CLI in der Amazon S3 API S3-Referenz. AWS CLI

Aktivieren der Transfer Acceleration auf einem Bucket

Verwenden Sie den AWS CLI put-bucket-accelerate-configurationBefehl, um Transfer Acceleration für einen Bucket zu aktivieren oder zu unterbrechen.

Im folgenden Beispiel wird `Status=Enabled` zur Aktivierung von Transfer Acceleration für einen Bucket festgelegt. Sie verwenden `Status=Suspended`, um Transfer Acceleration auszusetzen.

Example

```
$ aws s3api put-bucket-accelerate-configuration --bucket bucketname --accelerate-
configuration Status=Enabled
```

Nutzen der Transfer Acceleration

Sie können alle Amazon S3 S3-Anfragen, die mit den AWS CLI Befehlen s3 und s3api gestellt werden, an den Accelerate-Endpunkt weiterleiten:. `s3-accelerate.amazonaws.com` Setzen Sie dazu den Konfigurationswert `use_accelerate_endpoint true` in einem Profil in Ihrer AWS Config-Datei auf. Transfer Acceleration muss für Ihren Bucket aktiviert sein, um den beschleunigten Endpunkt nutzen zu können.

Alle Anforderungen werden mit der virtuellen Bucket-Adressierung gesendet: `my-bucket.s3-accelerate.amazonaws.com`. `ListBuckets`-, `CreateBucket`-, und `DeleteBucket`-Anfragen werden nicht an den beschleunigten Endpunkt gesendet, da der Endpunkt diese Vorgänge nicht unterstützt.

Weitere Informationen zu `use_accelerate_endpoint` finden Sie in der AWS CLI -S3-Konfiguration in der AWS CLI -Befehlsreferenz.

Das folgende Beispiel setzt `use_accelerate_endpoint` im Standardprofil auf `true`.

Example

```
$ aws configure set default.s3.use_accelerate_endpoint true
```

Wenn Sie den Accelerate-Endpunkt für einige AWS CLI Befehle verwenden möchten, für andere jedoch nicht, können Sie eine der folgenden beiden Methoden verwenden:

- Verwenden Sie den beschleunigten Endpunkt für jeden s3- oder s3api-Befehl, indem Sie den Parameter `--endpoint-url` auf `https://s3-accelerate.amazonaws.com` setzen.
- Richten Sie separate Profile in Ihrer AWS Config-Datei ein. Legen Sie beispielsweise ein Profil an, das `use_accelerate_endpoint` auf `true` setzt, und ein Profil, das `use_accelerate_endpoint` nicht setzt. Wenn Sie einen Befehl ausführen, geben Sie an, welches Profil Sie verwenden möchten, abhängig davon, ob Sie den beschleunigten Endpunkt verwenden möchten.

Hochladen eines Objekts in einen Bucket, der für Transfer Acceleration geeignet ist

Das folgende Beispiel lädt eine Datei in einen Bucket hoch, der für Transfer Acceleration konfiguriert ist. Dazu verwendet es das Standardprofil, das für die Verwendung des beschleunigten Endpunkts konfiguriert wurde.

Example

```
$ aws s3 cp file.txt s3://bucketname/keyname --region region
```

Das folgende Beispiel lädt eine Datei in einen Bucket hoch, der für Transfer Acceleration konfiguriert ist. Dazu verwendet es den Parameter `--endpoint-url` für die Angabe des beschleunigten Endpunkts.

Example

```
$ aws configure set s3.addressing_style virtual
$ aws s3 cp file.txt s3://bucketname/keyname --region region --endpoint-url https://s3-accelerate.amazonaws.com
```

Mit dem AWS SDKs

Im Folgenden finden Sie Beispiele für die Verwendung von Transfer Acceleration zum Hochladen von Objekten auf Amazon S3 mithilfe von AWS SDK. Einige der AWS SDK unterstützten Sprachen (z. B. Java und. NET) verwenden Sie ein Client-Konfigurationsflag für Accelerate Endpoint, sodass Sie den Endpunkt für Transfer Acceleration nicht explizit auf festlegen müssen *bucketname*.s3-accelerate.amazonaws.com.

Java

Example

Das folgende Beispiel zeigt, wie Sie einen beschleunigten Endpunkt für das Hochladen eines Objekts in Amazon S3 verwenden. Das Beispiel erledigt Folgendes:

- Erstellt einen `AmazonS3Client`, der für die Verwendung von beschleunigten Endpunkten konfiguriert ist. Für alle Buckets, auf die der Client zugreift, muss Transfer Acceleration aktiviert sein.
- Aktiviert Transfer Acceleration auf einem angegebenen Bucket. Dieser Schritt ist nur erforderlich, wenn für den von Ihnen angegebenen Bucket nicht bereits Transfer Acceleration aktiviert ist.
- Überprüft, ob Transfer Acceleration für den angegebenen Bucket aktiviert ist.
- Lädt ein neues Objekt in den angegebenen Bucket hoch und verwendet dazu den beschleunigten Endpunkt des Buckets.

Weitere Informationen zur Verwendung von Transfer Acceleration finden Sie unter Erste Schritte mit Amazon S3 Transfer Acceleration. Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter Erste Schritte im Entwicklerhandbuch. AWS SDK for Java

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
```

```
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketAccelerateConfiguration;
import com.amazonaws.services.s3.model.BucketAccelerateStatus;
import com.amazonaws.services.s3.model.GetBucketAccelerateConfigurationRequest;
import com.amazonaws.services.s3.model.SetBucketAccelerateConfigurationRequest;

public class TransferAcceleration {
    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ***";

        try {
            // Create an Amazon S3 client that is configured to use the accelerate
 endpoint.
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withRegion(clientRegion)
                    .withCredentials(new ProfileCredentialsProvider())
                    .enableAccelerateMode()
                    .build();

            // Enable Transfer Acceleration for the specified bucket.
            s3Client.setBucketAccelerateConfiguration(
                    new SetBucketAccelerateConfigurationRequest(bucketName,
                            new BucketAccelerateConfiguration(
                                    BucketAccelerateStatus.Enabled)));

            // Verify that transfer acceleration is enabled for the bucket.
            String accelerateStatus = s3Client.getBucketAccelerateConfiguration(
                    new GetBucketAccelerateConfigurationRequest(bucketName))
                    .getStatus();
            System.out.println("Bucket accelerate status: " + accelerateStatus);

            // Upload a new object using the accelerate endpoint.
            s3Client.putObject(bucketName, keyName, "Test object for transfer
 acceleration");
            System.out.println("Object \"" + keyName + "\" uploaded with transfer
 acceleration.");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
```

```
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Das folgende Beispiel zeigt, wie Sie mit dem AWS SDK for .NET Transfer Acceleration in einem Bucket aktivieren können. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem AWS SDK for. NETim AWS SDKfür. NETLeitfaden für Entwickler.

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class TransferAccelerationTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            EnableAccelerationAsync().Wait();
        }

        static async Task EnableAccelerationAsync()
        {
                try
                {
```

```
                    var putRequest = new PutBucketAccelerateConfigurationRequest
                    {
                        BucketName = bucketName,
                        AccelerateConfiguration = new AccelerateConfiguration
                        {
                            Status = BucketAccelerateStatus.Enabled
                        }
                    };
                    await
 s3Client.PutBucketAccelerateConfigurationAsync(putRequest);

                    var getRequest = new GetBucketAccelerateConfigurationRequest
                    {
                        BucketName = bucketName
                    };
                    var response = await
 s3Client.GetBucketAccelerateConfigurationAsync(getRequest);

                    Console.WriteLine("Acceleration state = '{0}' ",
 response.Status);
                }
                catch (AmazonS3Exception amazonS3Exception)
                {
                    Console.WriteLine(
                        "Error occurred. Message:'{0}' when setting transfer
 acceleration",
                        amazonS3Exception.Message);
                }
        }
    }
}
```

Wenn Sie ein Objekt in einen Bucket hochladen, für den Transfer Acceleration aktiviert ist, verwenden Sie für die Angabe des beschleunigten Endpunkts den Zeitpunkt, an dem ein Client erstellt wurde.

```
var client = new AmazonS3Client(new AmazonS3Config
            {
                RegionEndpoint = TestRegionEndpoint,
                UseAccelerateEndpoint = true
            }
```

Javascript

Ein Beispiel für die Aktivierung der Übertragungsbeschleunigung mithilfe von AWS SDK for JavaScript finden Sie unter Calling the putBucketAccelerate Configuration operation in der AWS SDKfor JavaScript API Reference.

Python (Boto)

Ein Beispiel für die Aktivierung von Transfer Acceleration mithilfe von SDK for Python finden Sie unter put_bucket_accelerate_configuration in der for Python (Boto3) -Referenz.AWS SDK API

Other

Informationen zur Verwendung von anderen finden Sie unter Beispielcode und Bibliotheken. AWS SDKs

Verwenden von REST API

Verwenden Sie den REST API `PutBucketAccelerateConfiguration` Vorgang, um die beschleunigte Konfiguration für einen vorhandenen Bucket zu aktivieren.

Weitere Informationen finden Sie PutBucketAccelerateConfigurationin der Amazon Simple Storage Service API Reference.

Verwenden des Amazon S3 Transfer Acceleration Speed Comparison-Tools

Sie können das Amazon S3 Transfer Acceleration Speed Comparison Tool verwenden, um beschleunigte und nicht beschleunigte Upload-Geschwindigkeiten in allen Amazon S3-Regionen zu vergleichen. Das Speed Comparison-Tool verwendet mehrteilige Uploads, um eine Datei von Ihrem Browser in verschiedene Amazon S3-Regionen mit und ohne die Verwendung von Transfer Acceleration zu übertragen.

Sie können mit einer der folgenden Methoden auf das Speed Comparison-Tool zugreifen:

- Kopieren Sie den folgenden Text URL in Ihr Browserfenster und ersetzen Sie *`region`* mit dem AWS-Region , was Sie verwenden (zum Beispiel`us-west-2`) und *`yourBucketName`* mit dem Namen des Buckets, den Sie auswerten möchten:

```
https://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/
accelerate-speed-comparsion.html?
region=region&origBucketName=yourBucketName
```

Eine Liste der von Amazon S3 unterstützten Regionen finden Sie unter Endpunkte und Kontingente von Amazon S3 in der Allgemeine AWS-Referenz.

- Verwenden der Amazon S3-Konsole

Nutzen von Buckets mit Zahlung durch den Anforderer für Speicherübertragungen und Nutzung

Im Allgemeinen zahlen Bucket-Eigentümer alle Amazon-S3-Speicher- und -Datenübertragungskosten, die ihrem Bucket zuzuordnen sind. Sie können jedoch einen Bucket als Bucket mit Zahlung durch den Anforderer konfigurieren. Bei Buckets mit Zahlung durch den Anforderer zahlt der Auftraggeber statt des Bucket-Eigentümers die Kosten für die Anforderung und den Daten-Download aus dem Bucket. Der Bucket-Eigentümer zahlt immer die Kosten für das Speichern der Daten.

In der Regel konfigurieren Sie Buckets dann als Buckets mit Zahlung durch den Anforderer, wenn Sie Daten freigeben möchten, aber keine Gebühren dafür übernehmen wollen, wenn andere auf die Daten zugreifen. Sie können beispielsweise Buckets mit Zahlung durch den Anforderer verwenden, wenn Sie große Datasets bereitstellen, beispielsweise Postleitzahlenverzeichnisse, Referenzdaten, Geodateninformationen oder Web-Crawling-Daten.

> **Important**
>
> Wenn Sie die Zahlung durch den Anforderer für einen Bucket aktivieren, ist kein anonymer Zugriff auf den Bucket zulässig.

Sie müssen alle Anforderungen für Buckets mit Zahlung durch den Anforderer authentifizieren. Die Anfrageauthentifizierung ermöglicht Amazon S3, den Auftraggeber zu identifizieren und ihm seine Verwendung des Buckets mit Zahlung durch den Anforderer in Rechnung zu stellen.

Wenn der Anforderer die Rolle AWS Identity and Access Management (IAM) annimmt, bevor er seine Anfrage stellt, wird die Anfrage dem Konto, zu dem die Rolle gehört, in Rechnung gestellt. Weitere Informationen zu IAM Rollen finden Sie im IAMBenutzerhandbuch unter IAMRollen.

Nachdem Sie einen Bucket als Bucket für „Requester Pays“ konfiguriert haben, müssen die Anforderer nachweisen, dass sie sich darüber im Klaren sind, dass ihnen die Anfrage und der Datendownload in Rechnung gestellt werden. Um nachzuweisen, dass sie die Gebühren akzeptieren, müssen die Anforderer entweder `x-amz-request-payer` als Header in ihre API Anfrage fürDELETE,, GET HEADPOST, und PUT -Anfragen aufnehmen oder den `RequestPayer` Parameter in ihrer Anfrage hinzufügen. REST Für CLI Anfragen können Anforderer den `--request-payer` Parameter verwenden.

Example — Die Verwendung von Requester Pays beim Löschen eines Objekts

Um Folgendes zu verwenden DeleteObjectVersionAPIErsetzen Sie beispielsweise die *`user input placeholders`* durch Ihre eigenen Informationen.

```
DELETE /Key+?versionId=VersionId HTTP/1.1
Host: Bucket.s3.amazonaws.com
x-amz-mfa: MFA
x-amz-request-payer: RequestPayer
x-amz-bypass-governance-retention: BypassGovernanceRetention
x-amz-expected-bucket-owner: ExpectedBucketOwner
```

Wenn der Anforderer Objekte mithilfe der wiederherstellt RestoreObjectAPI, Requester Pays wird unterstützt, solange der `x-amz-request-payer` Header oder der `RequestPayer` Parameter in der Anfrage enthalten sind. Der Anforderer zahlt jedoch nur für die Kosten der Anfrage. Der Bucket-Besitzer zahlt die Abrufgebühren.

Buckets mit Zahlung durch den Anforderer unterstützen die folgenden Funktionen nicht:

- Anonyme Anforderungen
- SOAPAnfragen
- Die Verwendung eines Buckets mit Zahlung durch den Anforderer als Ziel-Bucket für die Endbenutzer-Protokollierung, oder umgekehrt. Sie können jedoch die Endbenutzer-Protokollierung für einen Bucket mit Zahlung durch den Anforderer aktivieren, wenn es sich beim Ziel-Bucket nicht um einen Bucket mit Zahlung durch den Anforderer handelt.

Die Gebühren bei Zahlung durch den Anforderer

Die Verteilung der Gebühren für erfolgreiche Anforderungen mit Zahlung durch den Anforderer ist ganz einfach: Der Anforderer zahlt für die Datenübertragung und die Anforderung, der Bucket-

Eigentümer für die Speicherung der Daten. Dem Bucket-Eigentümer werden jedoch nur unter den folgenden Bedingungen Gebühren für die Anforderung in Rechnung gestellt:

- Die Anfrage gibt einen Fehler `AccessDenied` (HTTP`403 Forbidden`) zurück und die Anfrage wird innerhalb des individuellen AWS Kontos oder der AWS Organisation des Bucket-Besitzers initiiert.
- Die Anfrage ist eine SOAP Anfrage.

Weitere Informationen zur Zahlung durch den Anforderer finden Sie in den folgenden Themen.

Themen

- Konfigurieren von Zahlung durch den Anforderer für einen Bucket
- Die Konfiguration wird mit dem requestPayment abgerufen REST API
- Objekte aus Requester Pays-Buckets werden heruntergeladen

Konfigurieren von Zahlung durch den Anforderer für einen Bucket

Sie können einen Amazon-S3-Bucket als Bucket mit Zahlung durch den Anforderer konfigurieren, damit anstelle des Bucket-Eigentümers der Anforderer die Kosten für die Anforderung und den Datendownload übernimmt.

Dieser Abschnitt enthält Beispiele für die Konfiguration von Requester Pays in einem Amazon S3 S3-Bucket mithilfe der Konsole und der RESTAPI.

Verwenden der S3-Konsole

So aktivieren Sie die Zahlung durch den Anforderer für einen S3-Bucket:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie die Zahlung durch den Anforderer aktivieren möchten.
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie unter Requester pays (Zahlung durch Anforderer) die Option Edit (Bearbeiten).
5. Wählen Sie Enable (Aktivieren) und wählen Sie Save changes (Änderungen speichern) aus.

Amazon S3 aktiviert die Zahlung durch den Anforderer für Ihren Bucket und zeigt Ihre Bucket-Übersichtan. Unter Zahlung durch den Anforderer sehen Sie Aktiviert

Verwenden Sie den REST API

Nur der Bucket-Eigentümer kann den Konfigurationswert `RequestPaymentConfiguration.payer` eines Buckets auf `BucketOwner` (den Standardwert) oder `Requester` setzen. Die Einrichtung der Ressource `requestPayment` ist optional. Standardmäßig ist der Bucket kein Bucket mit Zahlung durch den Anforderer.

Um einen Bucket mit Zahlung durch den Anforderer in einen regulären Bucket umzuwandeln, verwenden Sie den Wert `BucketOwner`. In der Regel verwenden Sie `BucketOwner` zum Hochladen von Daten in den Amazon-S3-Bucket. Dann würden Sie den Wert auf `Requester` setzen, bevor Sie die Objekte in dem Bucket veröffentlichen.

Zum Einstellen requestPayment

- Verwenden Sie eine PUT-Anforderung, um den Wert `Payer` für einen bestimmten Bucket auf `Requester` zu setzen.

```
PUT ?requestPayment HTTP/1.1
Host: [BucketName].s3.amazonaws.com
Content-Length: 173
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS [Signature]

<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Payer>Requester</Payer>
</RequestPaymentConfiguration>
```

War die Anfrage erfolgreich, gibt Amazon S3 eine Antwort zurück, die etwa wie folgt aussieht.

```
HTTP/1.1 200 OK
x-amz-id-2: [id]
x-amz-request-id: [request_id]
Date: Wed, 01 Mar 2009 12:00:00 GMT
Content-Length: 0
Connection: close
Server: AmazonS3
```

```
x-amz-request-charged:requester
```

Sie können Zahlung durch den Anforderer nur auf Bucket-Ebene festlegen. Sie können Zahlung durch den Anforderer nicht für bestimmte Objekte innerhalb des Buckets festlegen.

Sie können einen Bucket jederzeit als `BucketOwner` oder `Requester` konfigurieren. Es kann jedoch einige Minuten dauern, bis der neue Konfigurationswert aktiv ist.

> **Note**
>
> Besitzer von Buckets, die vorsigniert ausgeben, URLs sollten sorgfältig überlegen, bevor sie einen Bucket als Requester Pays konfigurieren, vor allem, wenn der Bucket eine lange Lebensdauer URL hat. Dem Bucket-Besitzer wird jedes Mal eine Gebühr in Rechnung gestellt, wenn der Anforderer einen vorsignierten Bucket verwendet, der URL die Anmeldeinformationen des Bucket-Besitzers verwendet.

Die Konfiguration wird mit dem requestPayment abgerufen REST API

Sie können den `Payer`-Wert ermitteln, der für einen Bucket eingerichtet ist, indem Sie die Ressource `requestPayment` abrufen.

Um die Ressource zurückzugeben requestPayment

- Verwenden Sie eine GET Anfrage, um die `requestPayment` Ressource abzurufen, wie in der folgenden Anfrage gezeigt.

```
GET ?requestPayment HTTP/1.1
Host: [BucketName].s3.amazonaws.com
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS [Signature]
```

War die Anfrage erfolgreich, gibt Amazon S3 eine Antwort zurück, die etwa wie folgt aussieht.

```
HTTP/1.1 200 OK
x-amz-id-2: [id]
x-amz-request-id: [request_id]
Date: Wed, 01 Mar 2009 12:00:00 GMT
```

```
Content-Type: [type]
Content-Length: [length]
Connection: close
Server: AmazonS3

<?xml version="1.0" encoding="UTF-8"?>
<RequestPaymentConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<Payer>Requester</Payer>
</RequestPaymentConfiguration>
```

Diese Antwort zeigt, dass der `payer`-Wert auf `Requester` gesetzt ist.

Objekte aus Requester Pays-Buckets werden heruntergeladen

Den Anforderern werden die Gebühren für das Herunterladen von Daten aus Buckets mit Zahlung durch den Anforderer in Rechnung gestellt. Deshalb müssen die Anforderungen einen speziellen Parameter enthalten, `x-amz-request-payer`, der bestätigt, dass der Anforderer weiß, dass er Gebühren für den Download zahlen muss. Um auf Objekte in Buckets mit Zahlung durch den Anforderer zuzugreifen, müssen die Anforderungen eine der folgenden Informationen enthalten.

- FürDELETE,GET, und PUT Anfragen HEADPOST, fügen Sie sie `x-amz-request-payer : requester` in die Kopfzeile ein
- Wenn signiertURLs, fügen Sie es `x-amz-request-payer=requester` in die Anfrage ein

Ist die Anforderung erfolgreich und dem Auftraggeber werden Gebühren in Rechnung gestellt, enthält die Antwort den Header `x-amz-request-charged:requester`. Ist `x-amz-request-payer` nicht in der Anfrage enthalten, gibt Amazon S3 einen 403-Fehler zurück und stellt dem Bucket-Eigentümer die Gebühren für die Anfrage in Rechnung.

> **Note**
>
> Bucket-Eigentümer müssen ihren Anforderungen `x-amz-request-payer` nicht hinzufügen. Stellen Sie sicher, dass Sie `x-amz-request-payer` und seinen Wert in die Berechnung Ihrer Signatur aufgenommen haben. Weitere Informationen finden Sie unter Verwenden eines Autorisierungsheaders in der Amazon S3 API S3-Referenz.

Verwenden von REST API

Herunterladen von Objekten aus einem Bucket mit Zahlung durch den Anforderer

- Verwenden Sie eine GET-Anforderung, um ein Objekt aus einem Bucket mit Zahlung durch den Anforderer herunterzuladen, wie in der folgenden Anforderung gezeigt.

```
GET / [destinationObject] HTTP/1.1
Host: [BucketName].s3.amazonaws.com
x-amz-request-payer : requester
Date: Wed, 01 Mar 2009 12:00:00 GMT
Authorization: AWS [Signature]
```

Wenn die GET Anfrage erfolgreich ist und dem Anfragenden eine Gebühr berechnet wird, umfasst die Antwort Folgendes: `x-amz-request-charged:requester`

Amazon S3 kann für Anfragen, die versuchen, Objekte aus einem Bucket mit Zahlung durch den Anforderer abzurufen, einen `Access Denied`-Fehler zurückgeben. Weitere Informationen finden Sie unter Fehlerantworten in der Amazon Simple Storage Service API Reference.

Unter Verwendung des AWS CLI

Um Objekte mithilfe von aus einem Requester Pays-Bucket herunterzuladen AWS CLI, geben Sie dies im `--request-payer requester` Rahmen Ihrer `get-object` Anfrage an. Weitere Informationen finden Sie unter get-object in der AWS CLI -Referenz.

Arbeiten mit Objekten in Amazon S3

Um Ihre Daten in Amazon S3 zu speichern, arbeiten Sie mit Ressourcen, die als Buckets und Objekte bezeichnet werden. Ein Bucket ist ein Container für Objekte. Ein Objekt ist eine Datei und alle Metadaten, die diese Datei beschreiben.

Um ein Objekt in Amazon S3 zu speichern, erstellen Sie einen Bucket und laden das Objekt dann in einen Bucket hoch. Wenn sich das Objekt im Bucket befindet, können Sie es öffnen, herunterladen und kopieren. Wenn Sie ein Objekt oder einen Bucket nicht mehr benötigen, können Sie diese Ressourcen aufräumen.

> **Note**
>
> Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter S3 Express One Zone und Arbeiten mit Verzeichnis-Buckets.

> **Important**
>
> Wenn Sie in der Amazon S3 S3-Konsole für ein Objekt „Öffnen“ oder „Herunterladen als“ wählen, werden diese Operationen vorsigniertURLs. Für die Dauer von fünf Minuten ist Ihr Objekt für jeden zugänglich, der Zugriff auf diese URLs vorsignierten Objekte hat. Weitere Informationen zu Presigned finden Sie unter Presigned URLs verwenden. URLS

Mit Amazon S3 zahlen Sie nur für das, was Sie tatsächlich nutzen. Weitere Informationen zu den Funktionen und Preisen von Amazon S3 finden Sie unter Amazon S3. Wenn Sie neuer Amazon-S3-Kunde sind, können Sie kostenlos mit Amazon S3 beginnen. Weitere Informationen finden Sie unter Kostenloses Kontingent für AWS.

Themen

- Übersicht über Amazon-S3-Objekte
- Benennen Amazon S3 S3-Objekten
- Arbeiten mit Objekt-Metadaten
- Objekte hochladen

- Fügen Sie Vorbedingungen zu S3-Vorgängen mit bedingten Anfragen hinzu
- Objekte kopieren, verschieben und umbenennen
- Herunterladen von Objekten
- Überprüfung der Objektintegrität
- Löschen von Amazon-S3-Objekten
- Organisieren, Auflisten und Arbeiten mit Ihren Objekten
- Laden Sie Objekte mit vorsignierten Objekten herunter und laden Sie sie hoch URLs
- Transformieren von Objekten mit S3 Object Lambda
- Massenausführung von Objektoperationen mit Batch Operations
- Abfragen vorhandener Daten mit Amazon S3 Select

Übersicht über Amazon-S3-Objekte

Amazon S3 ist ein Objektspeicher, der eindeutige Schlüsselwerte verwendet, um so viele Objekte zu speichern, wie Sie möchten. Sie speichern diese Objekte in einem oder mehreren Buckets. Jedes Objekt kann eine Größe von bis zu 5 TB haben. Ein Objekt besteht aus Folgendem:

Schlüssel

Der Name, den Sie einem Objekt zuweisen. Zum Abrufen des Objekts verwenden Sie den Objektschlüssel. Weitere Informationen finden Sie unter Arbeiten mit Objektmetadaten.

Versions-ID

In einem Bucket wird ein Objekt durch einen Schlüssel und eine Versions-ID eindeutig identifiziert. Die Versions-ID ist eine Zeichenfolge, die Amazon S3 generiert, wenn Sie das Objekt einem Bucket hinzufügen. Weitere Informationen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

Value

Der Inhalt, den Sie speichern.

Ein Objektwert kann eine beliebige Bytefolge sein. Objekte können von 0 bis 5 TB groß sein. Weitere Informationen finden Sie unter Objekte hochladen.

Metadaten

Ein Satz von Name-Wert-Paaren, mit dem Sie Informationen zum Objekt speichern können. Sie können Ihren Objekten in Amazon S3 Metadaten zuweisen, auch als benutzerdefinierte Metadaten bezeichnet. Amazon S3 weist diesen Objekten auch System-Metadatenhinzu, die es für die Verwaltung von Objekten verwendet. Weitere Informationen finden Sie unter Arbeiten mit Objektmetadaten.

Subressourcen

Amazon S3 verwendet den Subressourcen-Mechanismus, um objektspezifische zusätzliche Informationen zu speichern. Subressourcen sind Objekten untergeordnet, deshalb sind werden immer einer anderen Entity zugeordnet, wie beispielsweise einem Objekt oder einem Bucket. Weitere Informationen finden Sie unter Objekt-Subressourcen.

Informationen zur Zugriffskontrolle

Sie können den Zugriff auf Objekte kontrollieren, die Sie in Amazon S3 speichern. Amazon S3 unterstützt sowohl die ressourcenbasierte Zugriffskontrolle, wie z. B. eine Zugriffskontrollliste (ACL) und Bucket-Richtlinien, als auch die benutzerbasierte Zugriffskontrolle. Weitere Informationen über die Kontrolle des Zugriffs über finden Sie im:

- Zugriffskontrolle in Amazon S3
- Identity and Access Management für Amazon S3
- Konfiguration ACLs

Ihre Amazon-S3-Ressourcen (wie z. B. Buckets und Objekte) sind standardmäßig privat. Sie müssen anderen ausdrücklich eine Berechtigung erteilen, damit sie auf diese Ressourcen zugreifen dürfen. Weitere Informationen über das Teilen von Objekten finden Sie unter Objekte mit vorsignierten teilen URLs.

Tags

Sie können Tags verwenden, um Ihre gespeicherten Objekte zu kategorisieren, für die Zugriffskontrolle oder die Kostenzuordnung. Weitere Informationen finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen.

Objekt-Subressourcen

Amazon S3 definiert eine Menge von Subressourcen, die Buckets und Objekten zugeordnet sind. Subressourcen sind Objekten untergeordnet. Dies bedeutet, dass Subressourcen nicht alleine existieren. Sie sind immer einer anderen Entität zugeordnet, z. B. einem Objekt oder einem Bucket.

Die folgende Tabelle listet die Subressourcen auf, die Amazon-S3-Objekten zugeordnet sind.

Subressource	Beschreibung
acl	Enthält eine Liste der Rechte, wobei die Empfänger und die erteilten Berechtigungen identifizieren werden. Wenn Sie ein Objekt erstellen, identifiziert die `acl` den Objekteigentümer, der volle Kontrolle über das Objekt hat. Sie können ein Objekt abrufen ACL oder es durch eine aktualisierte Liste von Zuschüssen ersetzen. Bei jeder Aktualisierung eines ACL müssen Sie das vorhandene ersetzenACL. Weitere Informationen zu finden ACLs Sie unterÜbersicht über die Zugriffskontrollliste (ACL).

Benennen Amazon S3 S3-Objekten

Der Objektschlüssel (oder Schlüsselname) identifiziert das Objekt in einem Amazon-S3-Bucket eindeutig. Wenn Sie ein Objekt erstellen, geben Sie den Schlüsselnamen an. Wenn Sie beispielsweise auf der Amazon S3 S3-Konsole einen Bucket auswählen, wird eine Liste der Objekte in Ihrem Bucket angezeigt. Diese Namen sind die Objektschlüssel.

Der Objektschlüsselname ist eine Folge von Unicode-Zeichen mit einer Kodierung von UTF-8 und einer Länge von bis zu 1.024 Byte. Bei den Objektschlüsselnamen muss die Groß- und Kleinschreibung beachtet werden. Der folgende Abschnitt enthält Einschränkungen bei Objektschlüsselnamen und Anleitungen zur Auswahl von Schlüsselnamen.

> **Note**
>
> Objektschlüsselnamen mit dem Wert „soap“ werden für virtual-hosted-style Anfragen nicht unterstützt. Für Objektschlüsselnamenwerte, bei denen „soap“ verwendet wird, URL muss stattdessen ein Pfadstil verwendet werden.

Auswahl von Objektschlüsselnamen

Das Amazon-S3-Datenmodell ist eine flache Struktur: Sie erstellen einen Bucket und der Bucket speichert Objekte. Es gibt keine Hierarchie von Unterbuckets oder Unterordnern. Sie können jedoch mit den Schlüsselnamenpräfixen und Trennzeichen eine logische Hierarchie erschließen, wie in der Amazon-S3-Konsole. Die Amazon-S3-Konsole unterstützt ein Ordnerkonzept. Weitere Informationen zum Bearbeiten von Metadaten über die Amazon-S3-Konsole finden Sie unter Bearbeiten von Objektmetadaten in der Amazon-S3-Konsole.

Angenommen, Ihr Bucket (`admin-created`) enthält vier Objekte mit den folgenden Objektschlüsseln:

`Development/Projects.xls`

`Finance/statement1.pdf`

`Private/taxdocument.pdf`

`s3-dg.pdf`

Die Konsole verwendet die Schlüsselnamenpräfixe (`Development/`, `Finance/` und `Private/`) und das Trennzeichen ("/"), um eine Ordnerstruktur wie dargestellt anzuzeigen. Der `s3-dg.pdf`-Schlüssel hat kein Präfix, deshalb erscheint sein Objekt direkt auf Root-Ebene des Buckets. Wenn Sie den Ordner `Development/` öffnen, werden Sie darin das Objekt `Projects.xlsx` sehen.

- Amazon S3 unterstützt Buckets und Objekt. Es gibt keine Hierarchie in Amazon S3. Durch die Verwendung von Präfixen und Trennzeichen in einem Objektschlüsselnamen AWS SDKs können die Amazon S3-Konsole und die Amazon S3-Konsole jedoch eine Hierarchie ableiten und das Konzept von Ordnern einführen.
- Die Amazon-S3-Konsole implementiert die Ordnerobjekterstellung, indem sie ein Null-Byte-Objekt mit dem Wert des Ordners für Präfix und Trennzeichen als Schlüssel erstellt. Diese Ordnerobjekte werden nicht in der Konsole angezeigt. Ansonsten verhalten sie sich wie alle anderen Objekte und können mit Hilfe von, und betrachtet und manipuliert werden. REST API AWS CLI AWS SDKs

Richtlinien für Objektschlüsselnamen

Sie können ein beliebiges UTF -8-Zeichen in einem Objektschlüsselnamen verwenden. Die Verwendung bestimmter Zeichen in Schlüsselnamen kann jedoch bei manchen Anwendungen und

Protokollen zu Problemen führen. Die folgenden Richtlinien helfen Ihnen dabei, die Einhaltung von websicheren ZeichenDNS, XML Parsern und anderen zu maximieren. APIs

Sichere Zeichen

Die folgenden Zeichensätze sind allgemein sicher für die Verwendung in Schlüsselnamen.

Alphanumerische Zeichen	• 0-9 • a-z • A-Z
Sonderzeichen	• Ausrufezeichen (!) • Bindestrich (-) • Unterstrich (_) • Punkt (.) • Sternchen (*) • Einzelnes Anführungszeichen (') • Öffnende Klammer (() • Schließende Klammer ())

Nachfolgend finden Sie Beispiele für gültige Objektschlüsselnamen:

- `4my-organization`
- `my.great_photos-2014/jan/myvacation.jpg`
- `videos/2014/birthday/video1.wmv`

> **Note**
>
> Bei Objekten mit Schlüsselnamen, die mit Punkt(en) "." enden, die über die Amazon-S3-Konsole heruntergeladen werden, werden sämtliche Punkte "." aus dem Schlüsselnamen des heruntergeladenen Objekts entfernt. Um ein Objekt herunterzuladen, dessen Schlüsselname auf Punkt (en) „.“ endet und im heruntergeladenen Objekt beibehalten wird, müssen Sie AWS Command Line Interface (AWS CLI) AWS SDKs, oder verwenden. REST API
> Darüber hinaus sollten Sie folgende Präfixbeschränkungen beachten:

- Objekte mit dem Präfix „./“ müssen mit dem AWS Command Line Interface (AWS CLI) AWS SDKs, oder hoch- oder REST API heruntergeladen werden. Sie können die Amazon-S3-Konsole nicht verwenden.
- Objekte mit dem Präfix „../“ können nicht mit AWS Command Line Interface (AWS CLI) oder der Amazon-S3-Konsole hochgeladen werden.

Zeichen, die möglicherweise eine Sonderverarbeitung benötigen

Die folgenden Zeichen in einem Schlüsselnamen erfordern möglicherweise zusätzliche Codebehandlung und müssen wahrscheinlich als URL kodiert oder referenziert werden. HEX Einige davon sind nicht darstellbare Zeichen, und Ihr Browser kann sie ggf. nicht verarbeiten, was zudem einer speziellen Vorgehensweise bedarf:

- Ampersand ("&")
- Dollar ("$")
- ASCIIZeichenbereiche 00—1F hex (0—31 dezimal) und 7F (127 dezimal)
- 'At'-Symbol ("@")
- Gleichheitszeichen ("=")
- Semikolon (";")
- Schrägstrich ("/")
- Doppelpunkt (":")
- Plus ("+")
- Leerzeichen – Wichtige Leerzeichenfolgen gehen möglicherweise bei bestimmten Verwendungszwecken verloren (insbesondere Mehrfachleerzeichen).
- Komma (",")
- Fragezeichen ("?")

Zeichen, die Sie vermeiden sollten

Wir empfehlen, die folgenden Zeichen nicht in einem Schlüsselnamen zu verwenden, da die Behandlung von Sonderzeichen häufig vorkommt und diese nicht in allen Anwendungen einheitlich ist.

- Umgekehrter Schrägstrich ("\")
- Linke geschweifte Klammer ("{")
- Nicht druckbare ASCII Zeichen (128—255 Dezimalzeichen)
- Caret ("^")
- Rechte geschweifte Klammer ("}")
- Prozentzeichen ("%")
- Accent Grave ("`")
- Rechte eckige Klammer ("]")
- Anführungszeichen
- Größersymbol (">")
- Linke eckige Klammer ("[")
- Tilde ("~")
- Kleiner als-Zeichen („<“)
- Pfundzeichen ("#")
- Vertikaler Strich ("|")

XMLSchlüsseleinschränkungen für verwandte Objekte

Gemäß der XMLNorm zur end-of-line Verarbeitung ist der gesamte XML Text normalisiert, sodass Zeilenumbrüche (ASCIICode 13) und Zeilenumbrüche, denen unmittelbar ein Zeilenvorschub folgt (ASCIICode 10), durch ein einzelnes Zeilenvorschubzeichen ersetzt werden. Um die korrekte Analyse von Objektschlüsseln in XML Anfragen zu gewährleisten, müssen Zeilenumbrüche und andere Sonderzeichen durch ihren entsprechenden XML Entitätscode ersetzt werden, wenn sie in XML Tags eingefügt werden. Im Folgenden finden Sie eine Liste solcher Sonderzeichen und ihrer entsprechenden Entitätscodes:

- ' wie `'`
- ” wie `"`
- & wie `&`
- < wie `<`
- > wie `>`
- \r wie `
` oder `
`
- \n wie `
` oder `
`

Example

Das folgende Beispiel veranschaulicht die Verwendung eines XML Entitätscodes als Ersatz für einen Zeilenumbruch. Diese `DeleteObjects`-Anforderung löscht ein Objekt mit dem `key`-Parameter: `/some/prefix/objectwith\rcarriagereturn` (wobei\ r die Zeilenumschaltung ist).

```
<Delete xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Object>
    <Key>/some/prefix/objectwith&#13;carriagereturn</Key>
  </Object>
</Delete>
```

Arbeiten mit Objekt-Metadaten

Sie können Objekt-Metadaten in Amazon S3 zum Zeitpunkt des Hochladens des Objekts festlegen. Objekt-Metadaten bestehen aus Name/Wert-Paaren. Nachdem Sie das Objekt hochgeladen haben, können Sie Objekt-Metadaten nicht mehr ändern. Die einzige Methode, wie Sie Objekt-Metadaten ändern können, ist es, eine Kopie des Objekts anzulegen und die Metadaten festzulegen.

Wenn Sie ein Objekt erstellen, geben Sie auch den Schlüsselnamen an, der das Objekt in dem Bucket eindeutig identifiziert. Der Objektschlüssel (oder Schlüsselname) identifiziert das Objekt in einem Amazon-S3-Bucket eindeutig. Weitere Informationen finden Sie unter Benennen Amazon S3 S3-Objekten.

Es gibt zweierlei Arten von Metadaten in Amazon S3: systemdefinierte und benutzerdefinierte Metadaten. In den folgenden Abschnitten finden Sie weitere Informationen zu systemdefinierten und benutzerdefinierten Metadaten. Weitere Informationen zum Bearbeiten von Metadaten mit der Amazon-S3-Konsole finden Sie unter Bearbeiten von Objektmetadaten in der Amazon-S3-Konsole.

Systemdefinierte Objektmetadaten

Amazon S3 verwaltet für jedes in einem Bucket gespeicherte Objekt einen Satz Systemmetadaten. Amazon S3 verarbeitet diese Systemmetadaten nach Bedarf. Beispielsweise verwaltet Amazon S3 das Erstellungsdatum eines Objekts und seine Größe als Metadaten und verwendet dies Information für die Objektverwaltung.

Es gibt zwei Kategorien von Systemmetadaten:

- Systemdefiniert – Metadaten wie beispielsweise das Erstellungsdatum eines Objekts werden vom System definiert. Dies bedeutet, dass nur Amazon S3 den Wert ändern kann.

- Benutzerdefiniert – Andere Systemmetadaten, wie beispielsweise die für das Objekt konfigurierte Speicherklasse und die Angabe, ob für das Objekt eine serverseitige Verschlüsselung aktiviert ist, sind Beispiele für Metadaten, deren Werte Sie definieren können. Wenn Ihr Bucket als Website konfiguriert ist, möchten Sie manchmal eine Seitenanfrage auf eine andere oder externe Seite umleitenURL. In diesem Fall ist eine Webseite ein Objekt in Ihrem Bucket. Amazon S3 speichert den Wert für die Seitenumleitung als Systemmetadaten, deren Wert Sie steuern können.

  Wenn Sie Objekte erstellen, können Sie die Werte dieser Systemmetadaten konfigurieren oder sie nach Bedarf aktualisieren. Weitere Informationen über Speicherklassen finden Sie unter Amazon S3 S3-Speicherklassen verstehen und verwalten.

  Amazon S3 verwendet AWS KMS Schlüssel, um Ihre Amazon S3 S3-Objekte zu verschlüsseln. AWS KMS verschlüsselt nur die Objektdaten. Die Prüfsumme wird zusammen mit dem angegebenen Algorithmus als Teil der Metadaten des Objekts gespeichert. Wenn für das Objekt eine serverseitige Verschlüsselung angefordert wird, wird die Prüfsumme in verschlüsselter Form gespeichert. Weitere Informationen zur serverseitigen Verschlüsselung finden Sie unter Datenschutz durch Verschlüsselung.

> **Note**
>
> Der PUT-Anforderungs-Header ist auf eine Größe von 8 KB begrenzt. Innerhalb des PUT-Anforderungs-Headers sind die systemdefinierten Metadaten auf eine Größe von 2 KB begrenzt. Die Größe systemdefinierter Metadaten wird anhand der Summe der Anzahl der Byte in der ASCII US-Kodierung jedes Schlüssels und Werts gemessen.

Die folgenden Tabelle enthält eine Liste der vom System definierten Metadaten, und gibt an, ob Sie sie aktualisieren können.

Name	Beschreibung	Kann der Benutzer den Wert ändern?
`Date`	Aktuelles Datum und Uhrzeit	Nein

Name	Beschreibung	Kann der Benutzer den Wert ändern?
`Cache-Control`	Ein allgemeines Header-Feld zum Angeben von Caching-Richtlinien.	Ja
`Content-Disposition`	Darstellungsinformationen zum Objekt.	Ja
`Content-Length`	Die Objektgröße in Bytes	Nein
`Content-Type`	Der Typ des Objekts.	Ja
`Last-Modified`	Das Datum, an dem das Objekt erstellt wurde, oder das letzte Änderungsdatum, je nachdem, welcher Wert aktueller ist Bei mehrteiligen Uploads entspricht das Objekterstellungsdatum dem Startdatum des mehrteiligen Uploads.	Nein
`ETag`	Ein Entity-Tag (ETag), das eine bestimmte Version eines Objekts darstellt. Bei Objekten, die nicht als mehrteiliger Upload hochgeladen werden und entweder unverschlüsselt oder durch serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt sind, ETag handelt es sich um eine MD5 Zusammenfassung der Daten.	Nein
`x-amz-server-side-encryption`	Ein Header, der angibt, ob die serverseitige Verschlüsselung für das Objekt aktiviert ist und ob diese Verschlüsselung die AWS Key Management Service (AWS KMS) -Schlüssel (SSE-KMS) oder die von Amazon S3 verwalteten Verschlüsselungsschlüssel (SSE-S3) verwendet. Weitere Informationen finden Sie unter Schützen von Daten mit serverseitiger Verschlüsselung.	Ja

Name	Beschreibung	Kann der Benutzer den Wert ändern?
`x-amz-checksum-crc32` , `x-amz-checksum-crc32c` , `x-amz-checksum-sha1` , `x-amz-checksum-sha256`	Header, die die Prüfsumme oder den Digest des Objekts enthalten. Je nach Prüfsummenalgorithmus, den Amazon S3 verwenden soll, wird höchstens jeweils einer dieser Header festgelegt. Weitere Informationen zur Auswahl des Prüfsummenalgorithmus finden Sie unter Überprüfung der Objektintegrität.	Nein
`x-amz-version-id`	Die Objekt-Versionsverwaltung. Wenn Sie die Versionsverwaltung für einen Bucket aktivieren, weist Amazon S3 allen Objekten, die dem Bucket hinzugefügt werden, eine Versions-ID zu. Weitere Informationen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.	Nein
`x-amz-delete-marker`	Ein boolesches Kennzeichen, das angibt, ob das Objekt eine Löschmarkierung ist. Dieses Kennzeichen wird nur in Buckets verwendet, für die die Versionsverwaltung aktiviert ist.	Nein
`x-amz-storage-class`	Die Speicherklasse, für die Speicherung des Objekts verwendet wird. Weitere Informationen finden Sie unter Amazon S3 S3-Speicherklassen verstehen und verwalten .	Ja
`x-amz-website-redirect-location`	Ein Header, der Anfragen für das zugehörige Objekt an ein anderes Objekt im selben Bucket oder an ein externes Objekt weiterleitet. URL Weitere Informationen finden Sie unter (Optional) Konfigurieren einer Webseiten umleitung.	Ja

Name	Beschreibung	Kann der Benutzer den Wert ändern?
`x-amz-server-side-encryption-aws-kms-key-id`	Ein Header, der die ID des AWS KMS symmetrischen KMS Verschlüsselungsschlüssels angibt, der zur Verschlüsselung des Objekts verwendet wurde. Dieser Header wird nur verwendet, wenn der Header `x-amz-server-side-encryption` vorhanden ist und den Wert `aws:kms` aufweist.	Ja
`x-amz-server-side-encryption-customer-algorithm`	Ein Header, der angibt, ob die serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) aktiviert ist. Weitere Informationen finden Sie unter Verwenden Sie serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE.	Ja
`x-amz-tagging`	Das Tag-Set für das Objekt. Das Tag-Set muss als Abfrageparameter codiert werden. URL	Ja

Benutzerdefinierte Objektmetadaten

Wenn Sie ein Objekt hochladen, können Sie ihm Metadaten zuweisen. Sie geben diese optionalen Informationen als Name-Wert-Paar (Schlüssel-Wert) an, wenn Sie eine `PUT`- oder `POST`-Anforderung senden, um das Objekt zu erstellen. Wenn Sie Objekte mithilfe von hochladen RESTAPI, müssen die optionalen benutzerdefinierten Metadatennamen mit beginnen, um sie von anderen `x-amz-meta-` Headern zu unterscheiden. HTTP Wenn Sie das Objekt mithilfe von abrufen RESTAPI, wird dieses Präfix zurückgegeben. Wenn Sie Objekte mit dem hochladen SOAPAPI, ist das Präfix nicht erforderlich. Wenn Sie das Objekt mithilfe von abrufen SOAPAPI, wird das Präfix entfernt, unabhängig davon, welches API Sie zum Hochladen des Objekts verwendet haben.

> **Note**
>
> SOAPSupport Over HTTP ist veraltet, aber SOAP weiterhin über verfügbar. HTTPS Neue Amazon S3 S3-Funktionen werden nicht unterstütztSOAP. Anstatt zu verwendenSOAP, empfehlen wir, entweder den REST API oder den zu verwenden AWS SDKs.

Wenn Metadaten über die abgerufen werden RESTAPI, fasst Amazon S3 Header mit demselben Namen (ohne Berücksichtigung der Groß- und Kleinschreibung) zu einer kommagetrennten Liste zusammen. Wenn einige Metadaten nicht darstellbare Zeichen enthalten, werden sie nicht zurückgegeben. Stattdessen wird der `x-amz-missing-meta`-Header mit der Anzahl der nicht darstellbaren Metadateneinträge als Wert zurückgegeben. Die `HeadObject`-Aktion ruft Metadaten von einem Objekt ab, ohne das Objekt selbst zurückzugeben. Dieser Vorgang ist nützlich, wenn Sie nur an den Metadaten eines Objekts interessiert sind. Um `HEAD` verwenden zu können, müssen Sie `READ`-Zugriff auf das Objekt haben. Weitere Informationen finden Sie HeadObjectin der Amazon Simple Storage Service API Reference.

Benutzerdefinierte Metadaten sind Schlüssel/Wert-Paare. Amazon S3 speichert benutzerdefinierte Metadatenschlüssel in Kleinbuchstaben.

Amazon S3 erlaubt beliebige Unicode-Zeichen in Ihren Metadatenwerten.

Um Probleme bei der Darstellung dieser Metadatenwerte zu vermeiden, sollten Sie bei der Verwendung von ASCII US-Zeichen REST und bei der Verwendung von -8 bei der Verwendung von SOAP oder browserbasierten Uploads über die Zeichen UTF -8 einhalten. `POST`

Wenn Sie in Ihren Metadatenwerten ASCII Nicht-US-Zeichen verwenden, wird die bereitgestellte Unicode-Zeichenfolge auf Nicht-US-Zeichen untersucht. ASCII Die Werte solcher Header werden vor dem Speichern gemäß 2047 dekodiert und gemäß RFC2047 codiert, um sie vor der Rückgabe mailsicher zu machen. RFC Wenn die Zeichenfolge nur ASCII US-Zeichen enthält, wird sie unverändert dargestellt.

Im Folgenden wird ein Beispiel gezeigt.

```
PUT /Key HTTP/1.1
Host: amzn-s3-demo-bucket.s3.amazonaws.com
x-amz-meta-nonascii: ÄMÄZÕÑ S3
```

```
HEAD /Key HTTP/1.1
Host: amzn-s3-demo-bucket.s3.amazonaws.com
x-amz-meta-nonascii: =?UTF-8?B?w4PChE3Dg8KEWsODwpXDg8KRIFMz?=

PUT /Key HTTP/1.1
Host: amzn-s3-demo-bucket.s3.amazonaws.com
x-amz-meta-ascii: AMAZONS3

HEAD /Key HTTP/1.1
Host: amzn-s3-demo-bucket.s3.amazonaws.com
x-amz-meta-ascii: AMAZONS3
```

> **Note**
>
> Der PUT-Anforderungs-Header ist auf eine Größe von 8 KB begrenzt. Innerhalb des PUT-Anforderungs-Headers sind die benutzerdefinierten Metadaten auf eine Größe von 2 KB begrenzt. Die Größe benutzerdefinierter Metadaten wird anhand der Summe der Byteanzahl in der UTF -8-Kodierung jedes Schlüssels und Werts gemessen.

Informationen zum Ändern der Metadaten Ihres Objekts, nachdem das Objekt hochgeladen wurde, indem eine Kopie des Objekts erstellt und geändert und das alte Objekt ersetzt wurde oder indem eine neue Version erstellt wurde, finden Sie unter Bearbeiten von Objektmetadaten in der Amazon-S3-Konsole.

Bearbeiten von Objektmetadaten in der Amazon-S3-Konsole

Sie können die Amazon S3 S3-Konsole verwenden, um Metadaten für bestehende S3-Objekte zu bearbeiten, indem Sie die Aktion Kopieren verwenden. Um Metadaten zu bearbeiten, kopieren Sie Objekte an dasselbe Ziel und geben die neuen Metadaten an, die Sie anwenden möchten. Diese ersetzen die alten Metadaten für das Objekt. Einige Metadaten werden von Amazon S3 festgelegt, wenn Sie das Objekt hochladen. Beispielsweise sind `Content-Length` und `Last-Modified` systemdefinierte Objektmetadatenfelder, die von einem Benutzer nicht geändert werden können.

Sie können auch benutzerdefinierte Metadaten festlegen, wenn Sie das Objekt hochladen, und sie ersetzen, wenn sich Ihre Anforderungen ändern. Beispielsweise haben Sie vielleicht eine Reihe von Objekten, die Sie ursprünglich in der `STANDARD`-Speicherklasse gespeichert haben. Im Laufe der Zeit benötigen Sie diese Daten möglicherweise nicht mehr, um hochverfügbar zu sein. Sie können

also die Speicherklasse auf ändern, `GLACIER` indem Sie den Wert des `x-amz-storage-class` Schlüssels von `STANDARD` bis ersetzen`GLACIER`.

> **Note**
>
> Beachten Sie Folgendes, wenn Sie Objektmetadaten in Amazon S3 ersetzen:
>
> - Sie müssen vorhandene Metadaten angeben, die Sie behalten möchten, Metadaten, die Sie hinzufügen möchten, und Metadaten, die Sie bearbeiten möchten.
> - Wenn Ihr Objekt weniger als 5 GB groß ist, können Sie die Aktion Kopieren in der S3-Konsole verwenden, um Objektmetadaten zu ersetzen. Wenn Ihr Objekt größer als 5 GB ist, können Sie die Objektmetadaten ersetzen, wenn Sie ein Objekt mit mehrteiligem Upload kopieren, indem Sie das AWS CLIoder AWS SDKsverwenden. Weitere Informationen finden Sie unter Kopieren eines Objekts mit Multipart-Upload.
> - Eine Liste der zusätzlichen Berechtigungen, die zum Ersetzen von Metadaten erforderlich sind, finden Sie unterthe section called “Erforderliche Berechtigungen für API S3-Operationen”. Richtlinien, die diese Berechtigung gewähren, finden Sie beispielsweise unterthe section called “Beispiele für identitätsbasierte Richtlinien”.
> - Dieser Vorgang erstellt eine Kopie des Objekts mit aktualisierten Einstellungen und dem Datum der letzten Änderung. Wenn S3-Versioning aktiviert ist, wird eine neue Version des Objekts erstellt, und das vorhandene Objekt wird zu einer älteren Version. Wenn das S3-Versioning nicht aktiviert ist, ersetzt eine neue Kopie des Objekts das ursprüngliche Objekt. Derjenige, der der IAM Rolle AWS-Konto zugeordnet ist, die die Eigenschaft ändert, wird auch der Besitzer des neuen Objekts oder (Objektversion).
> - Durch das Bearbeiten von Metadaten werden Werte für bestehende Schlüsselnamen ersetzt.
> - Objekte, die mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) verschlüsselt wurden, können nicht mit der Konsole kopiert werden. Sie müssen den AWS CLI, AWS SDK, oder den Amazon S3 verwenden RESTAPI.

> **Warning**
>
> Warten Sie beim Ersetzen von Metadaten für Ordner, bis die Aktion Kopieren abgeschlossen ist, bevor Sie dem Ordner neue Objekte hinzufügen. Andernfalls könnten auch neue Objekte bearbeitet werden.

In den folgenden Themen wird beschrieben, wie Metadaten für ein Objekt mithilfe der Aktion Kopieren in der Amazon S3 S3-Konsole ersetzt werden.

Ersetzen systemdefinierter Metadaten

Sie können einige systemdefinierte Metadaten für ein S3-Objekt ersetzen. Eine Liste der systemdefinierten Metadaten und Werte, die Sie ändern können, finden Sie unter. Systemdefinierte Objektmetadaten

So ersetzen Sie systemdefinierte Metadaten eines Objekts

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im Navigationsbereich Buckets und dann die Registerkarte Allzweck-Buckets oder Directory-Buckets aus. Navigieren Sie zu dem Amazon S3 S3-Bucket oder -Ordner, der die Objekte enthält, die Sie ändern möchten.
3. Aktivieren Sie das Kontrollkästchen für die Objekte, die Sie ändern möchten.
4. Wählen Sie im Menü Aktionen aus der daraufhin angezeigten Optionsliste die Option Kopieren aus.
5. Um den Zielpfad anzugeben, wählen Sie Browse S3, navigieren Sie zu demselben Ziel wie die Quellobjekte und aktivieren Sie das Kontrollkästchen Ziel. Wählen Sie unten rechts Choose destination (Ziel auswählen) aus.

   Geben Sie alternativ den Zielpfad ein.
6. Wenn Sie die Bucket-Versionierung nicht aktiviert haben, wird eine Warnung angezeigt, in der empfohlen wird, die Bucket-Versionierung zu aktivieren, um sich vor dem unbeabsichtigten Überschreiben oder Löschen von Objekten zu schützen. Wenn Sie alle Versionen von Objekten in diesem Bucket behalten möchten, wählen Sie Enable Bucket Versioning (Bucket-Versionierung aktivieren). Sie können die Standardeinstellungen für Verschlüsselung und Objektsperre auch in den Zieldetails einsehen.
7. Wählen Sie unter Zusätzliche Kopiereinstellungen die Option Einstellungen angeben aus, um Einstellungen für Metadaten festzulegen.
8. Scrollen Sie zum Abschnitt Metadaten und wählen Sie dann Alle Metadaten ersetzen aus.
9. Wählen Sie Metadaten hinzufügen aus.
10. Wählen Sie für Type(Typ) der Metadaten, System-defined (systemdefiniert) aus.
11. Geben Sie einen eindeutigen Key (Schlüssel) und den Value (Wert) der Metadaten an.

12. Um weitere Metadaten zu bearbeiten, wählen Sie Add metadata (Metadaten hinzufügen). Sie können auch „Entfernen“ wählen, um einen Satz von zu entfernen type-key-values.
13. Wählen Sie die Option Kopieren aus. Amazon S3 speichert Ihre Metadatenänderungen.

Ersetzen benutzerdefinierter Metadaten

Sie können benutzerdefinierte Metadaten eines Objekts ersetzen, indem Sie das Metadatenpräfix, und einen Namen kombinieren`x-amz-meta-`, den Sie wählen, um einen benutzerdefinierten Schlüssel zu erstellen. Wenn Sie beispielsweise den benutzerdefinierten Namen `alt-name` hinzufügen, wäre der Metadaten-Schlüssel `x-amz-meta-alt-name`.

Benutzerdefinierte Metadaten können insgesamt bis zu 2 KB umfassen. Um die Gesamtgröße der benutzerdefinierten Metadaten zu berechnen, addieren Sie die Anzahl der Byte in der UTF -8-Kodierung für jeden Schlüssel und Wert. Beide Schlüssel und ihre Werte müssen den US-Standards entsprechen. ASCII Weitere Informationen finden Sie unter Benutzerdefinierte Objektmetadaten.

Um benutzerdefinierte Metadaten eines Objekts zu ersetzen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im Navigationsbereich Buckets und dann die Registerkarte Allzweck-Buckets oder Directory-Buckets aus. Navigieren Sie zu dem Amazon S3 S3-Bucket oder -Ordner, der die Objekte enthält, die Sie ändern möchten.
3. Aktivieren Sie das Kontrollkästchen für die Objekte, die Sie ändern möchten.
4. Wählen Sie im Menü Aktionen aus der daraufhin angezeigten Optionsliste die Option Kopieren aus.
5. Um den Zielpfad anzugeben, wählen Sie Browse S3, navigieren Sie zu demselben Ziel wie die Quellobjekte und aktivieren Sie das Kontrollkästchen Ziel. Wählen Sie Choose destination (Ziel wählen).

   Geben Sie alternativ den Zielpfad ein.
6. Wenn Sie die Bucket-Versionierung nicht aktiviert haben, wird eine Warnung angezeigt, in der empfohlen wird, die Bucket-Versionierung zu aktivieren, um sich vor dem unbeabsichtigten Überschreiben oder Löschen von Objekten zu schützen. Wenn Sie alle Versionen von Objekten in diesem Bucket behalten möchten, wählen Sie Enable Bucket Versioning (Bucket-Versionierung aktivieren). Sie können die Standardeinstellungen für Verschlüsselung und Objektsperre auch in den Zieldetails einsehen.

7. Wählen Sie unter Zusätzliche Kopiereinstellungen die Option Einstellungen angeben aus, um Einstellungen für Metadaten festzulegen.
8. Scrollen Sie zum Abschnitt Metadaten und wählen Sie dann Alle Metadaten ersetzen aus.
9. Wählen Sie Metadaten hinzufügen aus.
10. Wählen Sie für Type (Typ) der Metadaten, User-defined (benutzerdefiniert) aus.
11. Geben Sie einen auf `x-amz-meta-` folgenden eindeutigen, benutzerdefinierten Key (Schlüssel) ein. Geben Sie ebenfalls einen Value (Wert) der Metadaten ein.
12. Um zusätzliche Metadaten hinzuzufügen, wählen Sie Add metadata (Metadaten hinzufügen). Sie können auch „Entfernen“ wählen, um einen Satz von zu entfernen type-key-values.
13. Wählen Sie die Option Kopieren aus. Amazon S3 speichert Ihre Metadatenänderungen.

Objekte hochladen

Wenn Sie eine Datei Amazon S3 hochladen, wird sie als S3-Objekt gespeichert. Objekte umfassen die Datei und die Metadaten, die das Objekt beschreiben. Ein Bucket kann eine unbegrenzte Anzahl von Objekten aufnehmen. Bevor Sie Dateien zu Amazon S3 hochladen können, benötigen Sie die Schreibberechtigungen für den Bucket. Weitere Informationen zu den Zugriffsberechtigungen finden Sie unter Identity and Access Management für Amazon S3.

Sie können beliebige Dateitypen – Bilder, Backups, Daten, Filme usw. – in einen S3-Bucket hochladen. Die maximale Größe einer Datei, die Sie über die Amazon-S3-Konsole hochladen können, ist 160 GB. Um eine Datei mit mehr als 160 GB hochzuladen, verwenden Sie AWS Command Line Interface (AWS CLI) AWS SDKs, oder Amazon S3 RESTAPI.

Wenn Sie ein Objekt mit einem Schlüsselnamen hochladen, der bereits in einem Bucket mit aktiviertem Versioning vorhanden ist, erstellt Amazon S3 eine weitere Version des Objekts, statt das vorhandene Objekt zu ersetzen. Weitere Informationen zur Aktivierung der Versionierung finden Sie unterAktivieren des Versioning für Buckets.

Abhängig von der Größe der Daten, die Sie hochladen, bietet Amazon S3 die folgenden Optionen:

- Laden Sie ein Objekt in einem einzigen Vorgang hoch, indem Sie AWS SDKs RESTAPI, oder AWS CLI verwenden. Mit einem einzigen PUT Vorgang können Sie ein einzelnes Objekt mit einer Größe von bis zu 5 GB hochladen.
- Hochladen eines einzelnen Objekts mit der Amazon-S3-Konsole – Mit der Amazon-S3-Konsole können Sie ein einzelnes Objekt mit einer Größe von bis zu 160 GB hochladen.

- Laden Sie ein Objekt in Teilen hoch, indem Sie AWS SDKs, RESTAPI, oder AWS CLI — Mithilfe des mehrteiligen API Upload-Vorgangs können Sie ein einzelnes großes Objekt mit einer Größe von bis zu 5 TB hochladen.

  Der API Vorgang zum mehrteiligen Hochladen wurde entwickelt, um das Hochladen größerer Objekte zu verbessern. Sie können ein Objekt in Teilen hochladen. Diese Objektteile können unabhängig, in jeder beliebigen Reihenfolge und parallel hochgeladen werden. Sie können einen mehrteiligen Upload verwenden, um Objekte mit einer Größe von 5 MB bis 5 TB hochzuladen. Weitere Informationen finden Sie unter Hochladen und Kopieren von Objekten mit mehrteiligen Uploads.

Wenn Sie ein Objekt hochladen, wird das Objekt standardmäßig automatisch mit serverseitiger Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt. Wenn Sie es herunterladen, wird das Objekt entschlüsselt. Weitere Informationen erhalten Sie unter Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets und Datenschutz durch Verschlüsselung.

Wenn Sie ein Objekt hochladen und eine andere Art der Standardverschlüsselung verwenden möchten, können Sie auch die serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-KMS) in Ihren PUT S3-Anfragen angeben oder die Standardverschlüsselungskonfiguration im Ziel-Bucket so einstellen, dass SSE - KMS zur Verschlüsselung Ihrer Daten verwendet wird. Weitere Hinweise zu SSE - KMS finden Sie unter. Geben Sie die serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) an Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, müssen Sie über die entsprechende Berechtigung verfügen. Weitere Informationen zu kontoübergreifenden Berechtigungen für KMS Schlüssel finden Sie im AWS Key Management Service Entwicklerhandbuch unter Erstellen von KMS Schlüsseln, die andere Konten verwenden können.

Wenn Sie in Amazon S3 auf den Fehler Access Denied (403 Forbidden) stoßen, finden Sie weitere Informationen Problembehandlung bei verweigertem Zugriff beheben (403 Forbidden) Fehler in Amazon S3 zu den häufigsten Ursachen unter.

Hochladen eines Objekts

Verwenden der S3-Konsole

In diesem Verfahren wird erläutert, wie Sie Objekte und Ordner mithilfe der Konsole in einen Amazon-S3-Bucket hochladen.

Wenn Sie ein Objekt hochladen, ist der Name des Objektschlüssels der Dateiname und etwaige optionale Präfixe. In der Amazon-S3-Konsole können Sie Ordner erstellen, um Ihre Objekte zu organisieren. In Amazon S3 werden Ordner als Präfixe dargestellt, die im Namen des Objektschlüssels angezeigt werden. Wenn Sie ein einzelnes Objekt in einen Ordner in der Amazon-S3-Konsole hochladen, wird der Ordnername im Namen des Objektschlüssels aufgenommen.

Wenn Sie beispielsweise ein Objekt mit dem Namen `sample1.jpg` in einen Ordner namens `backup` hochladen, lautet der Schlüsselname `backup/sample1.jpg`. Allerdings wird das Objekt in der Konsole als `sample1.jpg` im Ordner `backup` angezeigt. Weitere Informationen zu Schlüsselnamen finden Sie unter Arbeiten mit Objekt-Metadaten.

> **Note**
>
> Wenn Sie ein Objekt umbenennen oder eine der Eigenschaften in der Amazon-S3-Konsole ändern, z. B. Speicherklasse, Verschlüsselung oder Metadaten, wird ein neues Objekt erstellt, das das alte ersetzt. Wenn S3-Versioning aktiviert ist, wird eine neue Version des Objekts erstellt, und das vorhandene Objekt wird zu einer älteren Version. Die Rolle, die die Eigenschaft ändert, wird auch Besitzer des neuen Objekts (oder der neuen Objektversion).

Wenn Sie einen Ordner hochladen, lädt Amazon S3 alle Dateien und Unterordner aus dem angegebenen Ordner in Ihren Bucket hoch. Es wird dann ein Objektschlüsselname zugewiesen, der eine Kombination aus dem hochgeladenen Dateinamen und dem Ordnernamen ist. Wenn Sie beispielweise einen Ordner mit dem Namen `/images` hochladen, der zwei Dateien, `sample1.jpg` und `sample2.jpg`, enthält, lädt Amazon S3 die Dateien hoch und weist ihnen die jeweiligen Schlüsselnamen `images/sample1.jpg` und `images/sample2.jpg` zu. Der Schlüsselname enthält den Ordnernamen als Präfix. Die Amazon-S3-Konsole zeigt nur den Teil des Schlüsselnamens nach dem letzten `/` an. Beispielsweise werden in einem `images`-Ordner die Objekte `images/sample1.jpg` und `images/sample2.jpg` als `sample1.jpg` und `sample2.jpg` angezeigt.

So laden Sie Ordner und Dateien in einen S3-Bucket hoch

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, in den Ihre Ordner oder Dateien hochgeladen werden sollen.

4. Klicken Sie auf Upload.
5. Führen Sie im Fenster Upload einen der folgenden Schritte aus:
   - Ziehen Sie Dateien und Ordner in das Fenster Upload (Hochladen) und legen Sie sie dort ab.
   - Wählen Sie Datei hinzufügen oder Ordner hinzufügen aus, wählen Sie die hochzuladenden Dateien oder Ordner und anschließend Öffnen aus.
6. Um die Versioning zu aktivieren, wählen Sie unter Destination (Ziel) die Option Enable Bucket Versioning (Bucket-Versioning aktivieren) aus.
7. Um die aufgelisteten Dateien und Ordner hochzuladen, ohne zusätzliche Upload-Optionen zu konfigurieren, wählen Sie unten auf der Seite Upload aus.

   Amazon S3 lädt Ihre Objekte und Ordner hoch. Wenn der Upload abgeschlossen ist, wird auf der Seite Upload: Status eine Erfolgsmeldung angezeigt.

So konfigurieren Sie zusätzliche Objekteigenschaften

1. Um Zugriffssteuerungslisten-Berechtigungen zu ändern, wählen Sie Permissions (Berechtigungen) aus.
2. Bearbeiten Sie unter Zugriffskontrollliste (ACL) die Berechtigungen.

   Weitere Informationen zu den Zugriffsberechtigungen für Objekte finden Sie unter Verwenden Sie die S3-Konsole, um ACL Berechtigungen für ein Objekt festzulegen. Sie können der Öffentlichkeit Lesezugriff auf Ihre Objekte erteilen, und zwar für alle Dateien, die Sie hochladen. Wir empfehlen allerdings, die Standardeinstellung für den öffentlichen Lesezugriff nicht zu ändern. Die Erteilung von öffentlichem Lesezugriff ist nur für wenige Anwendungsfälle sinnvoll, beispielsweise, wenn Buckets für Websites verwendet werden. Nachdem Sie das Objekt hochgeladen haben, können Sie jederzeit Änderungen an den Objektberechtigungen vornehmen.
3. Um andere zusätzliche Eigenschaften zu konfigurieren, wählen Sie Properties (Eigenschaften) aus.
4. Wählen Sie im Abschnitt Speicherklasse die Speicherklasse für die Dateien aus, die Sie hochladen.

   Weitere Informationen über Speicherklassen finden Sie unter Amazon S3 S3-Speicherklassen verstehen und verwalten.

5. Um die Verschlüsselungseinstellungen für Ihre Objekte zu aktualisieren, gehen Sie unter Server-side encryption settings (Serverseitige Verschlüsselungseinstellungen) wie folgt vor.

   a. Klicken Sie auf Specify an encryption key (Verschlüsselungsschlüssel angeben).

   b. Wählen Sie unter Verschlüsselungseinstellungen die Option Verwenden von Bucket-Einstellungen für die Standardverschlüsselung oder Überschreiben der Bucket-Einstellungen für die Standardverschlüsselung aus.

   c. Wenn Sie Überschreiben der Bucket-Einstellungen für die Standardverschlüsselung ausgewählt haben, müssen Sie die folgenden Verschlüsselungseinstellungen konfigurieren.

      - Um die hochgeladenen Dateien mithilfe von Schlüsseln zu verschlüsseln, die von Amazon S3 verwaltet werden, wählen Sie Amazon S3 managed key (SSE-S3).

        Weitere Informationen finden Sie unter Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden (SSE-S3).

      - Um die hochgeladenen Dateien mit den in AWS Key Management Service (AWS KMS) gespeicherten Schlüsseln zu verschlüsseln, wählen Sie AWS Key Management Service Schlüssel (SSE-). KMS Wählen Sie dann eine der folgenden Optionen für AWS KMS - Schlüssel aus:

        - Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys und wählen Sie dann Ihren KMSSchlüssel aus der Liste der verfügbaren Schlüssel aus.

          Sowohl der Von AWS verwalteter Schlüssel (`aws/s3`) als auch Ihr vom Kunden verwalteter Schlüssel werden in dieser Liste angezeigt. Weitere Informationen über vom Kunden verwaltete Schlüssel finden Sie unter Kundenschlüssel und AWS -Schlüssel im Entwicklerhandbuch zu AWS Key Management Service .

        - Um den KMS Schlüssel einzugebenARN, wählen Sie Enter AWS KMS key ARN und geben Sie dann Ihren KMS Schlüssel ARN in das angezeigte Feld ein.

        - Um einen neuen, vom Kunden verwalteten Schlüssel in der AWS KMS Konsole zu erstellen, wählen Sie Create a KMS key aus.

          Weitere Informationen zum Erstellen eines AWS KMS key finden Sie unter Schlüssel erstellen im AWS Key Management Service Entwicklerhandbuch.

> **Important**
>
> Sie können nur KMS Schlüssel verwenden, die im selben AWS-Region Bucket verfügbar sind. Die Amazon S3 S3-Konsole listet nur die ersten 100 KMS Schlüssel in derselben Region wie der Bucket auf. Um einen KMS Schlüssel zu verwenden, der nicht aufgeführt ist, müssen Sie Ihren KMS Schlüssel eingebenARN. Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, benötigen Sie zunächst die Erlaubnis, den Schlüssel zu verwenden, und dann müssen Sie den KMS Schlüssel eingebenARN. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel und keine asymmetrischen KMS Schlüssel. Weitere Informationen finden Sie unter Identifizieren symmetrischer und asymmetrischer KMS Schlüssel im Entwicklerhandbuch.AWS Key Management Service

6. Wählen Sie On (Ein) aus, um zusätzliche Prüfsummen zu verwenden. Wählen Sie dann für Checksum function (Prüfsummenfunktion) die Funktion aus, die Sie verwenden möchten. Amazon S3 berechnet und speichert den Prüfsummenwert, nachdem es das gesamte Objekt erhalten hat. Sie können das Feld Precalculated value (Vorberechneter Wert) verwenden, um einen vorberechneten Wert anzugeben. In diesem Fall vergleicht Amazon S3 den von Ihnen angegebenen Wert mit dem eigenen berechneten Wert. Wenn die beiden Werte nicht übereinstimmen, generiert Amazon S3 einen Fehler.

   Mit zusätzlichen Prüfsummen können Sie den Prüfsummenalgorithmus angeben, den Sie zur Überprüfung Ihrer Daten verwenden möchten. Weitere Informationen zu zusätzlichen Prüfsummen finden Sie unter Überprüfung der Objektintegrität.

7. Um allen hochgeladenen Objekten Markierungen hinzuzufügen, wählen Sie Add tag (Tag hinzufügen). Geben Sie einen Tag-Namen in das Feld Schlüssel ein. Geben Sie einen Wert für das Tag ein.

   Das Markieren von Objekten ermöglicht Ihnen, Speicher zu kategorisieren. Jeder Tag ist ein Schlüssel/Wert-Paar. Bei Schlüssel- und Tag-Werten wird die Groß-/Kleinschreibung berücksichtigt. Sie können über bis zu 10 Markierungen pro Objekt verfügen. Ein Tag-Schlüssel kann maximal 128 Unicode-Zeichen und die Tag-Werte bis zu 255 Unicode-Zeichen lang sein. Weitere Informationen über Objekt-Markierungen finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen.

8. Um Metadaten hinzuzufügen, wählen Sie Add metadata (Metadaten hinzufügen).

a. Wählen Sie unter Typdie Option System defined (System definiert) oder User defined (Benutzerdefiniert) aus.

   Für systemdefinierte Metadaten können Sie allgemeine HTTP Header wie Content-Type und Content-Disposition auswählen. Eine Liste systemdefinierter Metadaten und Informationen dazu, ob Sie Werte hinzufügen können, finden Sie unter Systemdefinierte Objektmetadaten. Metadaten, die mit dem Präfix `x-amz-meta-` beginnen, werden als benutzerdefinierte Metadaten behandelt. Benutzerdefinierte-Metadaten werden mit dem Objekt gespeichert und beim Download mit ihm zurückgegeben. Sowohl die Schlüssel als auch ihre Werte müssen den US-Standards entsprechen. ASCII Benutzerdefinierte Metadaten können bis zu 2 KB umfassen. Weitere Informationen zu systemdefinierten und benutzerdefinierten Metadaten finden Sie unter Arbeiten mit Objekt-Metadaten.

b. Wählen Sie für Key einen Schlüssel aus.

c. Geben Sie einen Wert für den Schlüssel ein.

9. Um Ihre Objekte hochzuladen, wählen Sie Upload (Hochladen) aus.

   Amazon S3 lädt Ihr Objekt hoch. Wenn der Upload abgeschlossen ist, wird auf der Seite Upload: status eine Erfolgsmeldung angezeigt.

10. Wählen Sie Exit (Beenden) aus.

Mit dem AWS CLI

Sie können eine PUT-Anforderung zum Hochladen eines Objekts von bis zu 5 GB in einem einzigen Vorgang senden. Weitere Informationen finden Sie im `PutObject`-Beispiel in der AWS CLI - Befehlsreferenz.

Mit dem REST API

Sie können REST Anfragen zum Hochladen eines Objekts senden. Sie können eine PUT-Anforderung senden, um Daten in einer einzigen Operation hochzuladen. Weitere Informationen finden Sie unter PUTObjekt.

Mit dem AWS SDKs

Sie können das verwenden AWS SDKs, um Objekte in Amazon S3 hochzuladen. SDKsSie bieten Wrapper-Bibliotheken, mit denen Sie Daten einfach hochladen können. Weitere Informationen finden Sie in der Liste der unterstützten SDKs Programme.

Hier sind einige Beispiele mit einigen ausgewähltenSDKs:

.NET

Das folgende C#-Codebeispiel erstellt zwei Objekte mit zwei `PutObjectRequest`-Anforderungen:

- Die erste `PutObjectRequest`-Anforderung speichert als Beispielobjektdaten eine Textzeichenfolge. Sie gibt zudem Bucket- und Objektschlüsselnamen an.
- Die zweite `PutObjectRequest`-Anforderung lädt eine Datei durch Angabe des Dateinamens hoch. Diese Anforderung gibt außerdem den `ContentType`-Header und optionale Objektmetadaten (einen Titel) ein.

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem AWS SDK for. NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadObjectTest
    {
        private const string bucketName = "*** bucket name ***";
        // For simplicity the example creates two objects from the same file.
        // You specify key names for these objects.
        private const string keyName1 = "*** key name for first object created ***";
        private const string keyName2 = "*** key name for second object created
 ***";
        private const string filePath = @"*** file path ***";
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.EUWest1;

        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
```

```
            WritingAnObjectAsync().Wait();
        }

        static async Task WritingAnObjectAsync()
        {
            try
            {
                // 1. Put object-specify only key name for the new object.
                var putRequest1 = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName1,
                    ContentBody = "sample text"
                };

                PutObjectResponse response1 = await
client.PutObjectAsync(putRequest1);

                // 2. Put the object-set ContentType and add metadata.
                var putRequest2 = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName2,
                    FilePath = filePath,
                    ContentType = "text/plain"
                };

                putRequest2.Metadata.Add("x-amz-meta-title", "someTitle");
                PutObjectResponse response2 = await
client.PutObjectAsync(putRequest2);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine(
                        "Error encountered ***. Message:'{0}' when writing an
object"
                        , e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine(
                    "Unknown encountered on server. Message:'{0}' when writing an
object"
                    , e.Message);
```

```
            }
        }
    }
}
```

Java

Das folgende Beispiel erstellt zwei Objekte. Das erste Objekt besitzt eine Textzeichenfolge als Daten, und das zweite Objekt ist eine Datei. Das Beispiel erstellt das erste Objekt durch Angabe des Bucket-Namen, des Objektschlüssels und von Textdaten direkt in einem Aufruf an `AmazonS3Client.putObject()`. Das Beispiel erstellt das zweite Objekt mittels einer `PutObjectRequest`, die den Bucket-Namen, den Objektschlüssel und den Dateipfad angibt. Die `PutObjectRequest` gibt zudem den `ContentType`-Header und Titelmetadaten an.

Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ObjectMetadata;
import com.amazonaws.services.s3.model.PutObjectRequest;

import java.io.File;
import java.io.IOException;

public class UploadObject {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String stringObjKeyName = "*** String object key name ***";
        String fileObjKeyName = "*** File object key name ***";
        String fileName = "*** Path to file to upload ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
```

```
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withRegion(clientRegion)
                    .build();

            // Upload a text string as a new object.
            s3Client.putObject(bucketName, stringObjKeyName, "Uploaded String
 Object");

            // Upload a file as a new object with ContentType and title specified.
            PutObjectRequest request = new PutObjectRequest(bucketName,
 fileObjKeyName, new File(fileName));
            ObjectMetadata metadata = new ObjectMetadata();
            metadata.setContentType("plain/text");
            metadata.addUserMetadata("title", "someTitle");
            request.setMetadata(metadata);
            s3Client.putObject(request);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

JavaScript

Im folgenden Beispiel wird eine vorhandene Datei in einen Amazon-S3-Bucket in einer spezifischen Region hochgeladen.

```
import { readFile } from "node:fs/promises";

import {
  PutObjectCommand,
  S3Client,
  S3ServiceException,
} from "@aws-sdk/client-s3";

/**
```

```
 * Upload a file to an S3 bucket.
 * @param {{ bucketName: string, key: string, filePath: string }}
 */
export const main = async ({ bucketName, key, filePath }) => {
  const client = new S3Client({});
  const command = new PutObjectCommand({
    Bucket: bucketName,
    Key: key,
    Body: await readFile(filePath),
  });

  try {
    const response = await client.send(command);
    console.log(response);
  } catch (caught) {
    if (
      caught instanceof S3ServiceException &&
      caught.name === "EntityTooLarge"
    ) {
      console.error(
        `Error from S3 while uploading object to ${bucketName}. \
The object was too large. To upload objects larger than 5GB, use the S3 console
 (160GB max) \
or the multipart upload API (5TB max).`,
      );
    } else if (caught instanceof S3ServiceException) {
      console.error(
        `Error from S3 while uploading object to ${bucketName}.  ${caught.name}:
 ${caught.message}`,
      );
    } else {
      throw caught;
    }
  }
};
```

PHP

Dieses Beispiel führt Sie durch die Verwendung von Klassen aus AWS SDK for PHP , um ein Objekt mit einer Größe von bis zu 5 GB hochzuladen. Für größere Dateien müssen Sie den mehrteiligen API Upload-Vorgang verwenden. Weitere Informationen finden Sie unter Hochladen und Kopieren von Objekten mit mehrteiligen Uploads.

Weitere Informationen über die AWS SDK für Ruby API finden Sie unter AWS SDKfür Ruby — Version 2.

Example – Erstellen eines Objekts in einem Amazon-S3-Bucket, indem Daten hochgeladen werden

Im folgenden PHP Beispiel wird ein Objekt in einem angegebenen Bucket erstellt, indem Daten mithilfe der `putObject()` Methode hochgeladen werden.

```
 require 'vendor/autoload.php';

use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

try {
    // Upload data.
    $result = $s3->putObject([
        'Bucket' => $bucket,
        'Key'    => $keyname,
        'Body'   => 'Hello, world!',
        'ACL'    => 'public-read'
    ]);

    // Print the URL to the object.
    echo $result['ObjectURL'] . PHP_EOL;
} catch (S3Exception $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

Ruby

Die AWS SDK for Ruby - Version 3 bietet zwei Möglichkeiten, ein Objekt auf Amazon S3 hochzuladen. Bei der ersten wird ein verwalteter Datei-Uploader verwendet, mit dem sich Dateien

beliebiger Größe einfacher von einer Festsplatte hochladen lassen. So verwenden Sie die Methode des verwalteten Datei-Uploaders:

1. Erstellen Sie eine Instance der `Aws::S3::Resource`-Klasse.
2. Verweisen Sie per Bucket-Name und Schlüssel auf das Ziel-Objekt. Objekte befinden sich in einem Bucket und haben eindeutige Schlüssel, die jedes Objekt identifizieren.
3. Rufen Sie`#upload_file` für das Objekt auf.

Example

```
require 'aws-sdk-s3'

# Wraps Amazon S3 object actions.
class ObjectUploadFileWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Uploads a file to an Amazon S3 object by using a managed uploader.
  #
  # @param file_path [String] The path to the file to upload.
  # @return [Boolean] True when the file is uploaded; otherwise false.
  def upload_file(file_path)
    @object.upload_file(file_path)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't upload file #{file_path} to #{@object.key}. Here's why:
 #{e.message}"
    false
  end
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-uploaded-file"
  file_path = "object_upload_file.rb"
```

```
  wrapper = ObjectUploadFileWrapper.new(Aws::S3::Object.new(bucket_name,
 object_key))
  return unless wrapper.upload_file(file_path)

  puts "File #{file_path} successfully uploaded to #{bucket_name}:#{object_key}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Die zweite Methode, mit der AWS SDK for Ruby — Version 3 ein Objekt hochladen kann, verwendet die `#put` Methode von`Aws::S3::Object`. Das ist praktisch, wenn das Objekt eine Zeichenfolge oder ein I/O-Objekt ist, bei dem es sich nicht um eine Datei auf einem Datenträger handelt. So verwenden Sie diese Methode:

1. Erstellen Sie eine Instance der `Aws::S3::Resource`-Klasse.
2. Verweisen Sie per Bucket-Name und Schlüssel auf das Ziel-Objekt.
3. Rufen Sie `#put` auf und übergeben Sie die Zeichenfolge oder das I/O-Objekt.

Example

```
require 'aws-sdk-s3'

# Wraps Amazon S3 object actions.
class ObjectPutWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  def put_object(source_file_path)
    File.open(source_file_path, 'rb') do |file|
      @object.put(body: file)
    end
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't put #{source_file_path} to #{object.key}. Here's why:
 #{e.message}"
    false
  end
```

```
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-object-key"
  file_path = "my-local-file.txt"

  wrapper = ObjectPutWrapper.new(Aws::S3::Object.new(bucket_name, object_key))
  success = wrapper.put_object(file_path)
  return unless success

  puts "Put file #{file_path} into #{object_key} in #{bucket_name}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Verhindern Sie das Hochladen von Objekten mit identischen Schlüsselnamen

Sie können überprüfen, ob ein Objekt in Ihrem Bucket vorhanden ist, bevor Sie es mithilfe von bedingten Schreibvorgängen beim Hochladen erstellen. Dadurch kann verhindert werden, dass vorhandene Daten überschrieben werden. Durch bedingte Schreibvorgänge wird bestätigt, dass sich beim Hochladen kein Objekt mit demselben Schlüsselnamen bereits in Ihrem Bucket befindet.

Sie können bedingte Schreibvorgänge verwenden für PutObjectoder CompleteMultipartUploadAnfragen.

Weitere Informationen zu bedingten Anfragen finden Sie unterFügen Sie Vorbedingungen zu S3-Vorgängen mit bedingten Anfragen hinzu.

Hochladen und Kopieren von Objekten mit mehrteiligen Uploads

Mit dem mehrteiligen Upload können Sie ein einzelnes Objekt als Satz aus mehreren Teilen hochladen. Jeder Teil ist ein zusammenhängender Teil der Daten des Objekts. Sie können diese Objektteile unabhängig und in beliebiger Reihenfolge hochladen. Wenn die Übertragung eines Teils fehlschlägt, können Sie das Teil erneut übertragen, ohne dass dies Auswirkungen auf andere Teile hat. Nachdem alle Teile Ihres Objekts hochgeladen sind, fügt Amazon S3 diese Teile zusammen und erstellt das Objekt. Wenn Ihre Objektgröße 100 MB erreicht, sollten Sie in der Regel mehrteilige Uploads verwenden, anstatt das Objekt in einem einzigen Vorgang hochzuladen.

Die Nutzung mehrteiliger Uploads bietet die folgenden Vorteile:

- Verbesserter Durchsatz – Sie können die Teile parallel hochladen, um den Durchsatz zu erhöhen.
- Schnelle Wiederherstellung bei Netzwerkproblemen – Die kleinere Teilegröße minimiert die Auswirkungen eines Neustarts eines fehlgeschlagenen Uploads aufgrund eines Netzwerkfehlers.
- Anhalten und Fortsetzen von Objekt-Uploads – Sie können Objektteile mit der Zeit hochladen. Nachdem Sie einen mehrteiligen Upload initiiert haben, gibt es kein Ablaufdatum. Sie müssen den mehrteiligen Upload ausdrücklich abschließen oder abbrechen.
- Starten Sie einen Upload, bevor Sie die endgültige Objektgröße kennen – Sie können ein Objekt hochladen, während Sie es noch erstellen.

Sie sollten den mehrteiligen Upload wie folgt verwenden:

- Wenn Sie große Objekte über ein stabiles Netzwerk mit hoher Bandbreite hochladen, können Sie einen mehrteiligen Upload verwenden, um die Nutzung der verfügbaren Bandbreite zu maximieren. Hierzu laden Sie Objektteile parallel hoch, um von einer Multi-Threading-Leistung zu profitieren.
- Wenn Sie einen Upload über ein instabiles Netzwerk ausführen, können Sie einen mehrteiligen Upload verwenden, um die Resilienz in Bezug auf Netzwerkfehler durch Vermeidung von Neustarts der Uploads zu vermeiden. Wenn Sie mehrteilige Uploads verwenden, müssen Sie nur die Teile erneut hochladen, deren Upload unterbrochen wurde. Sie müssen nicht das gesamte Objekt von Anfang an neu hochladen.

> **Note**
>
> Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter S3 Express One Zone und Arbeiten mit Verzeichnis-Buckets. Weitere Informationen zur Verwendung mehrteiliger Uploads mit S3 Express One Zone und Verzeichnis-Buckets finden Sie unter Mehrteilige Uploads mit Verzeichnis-Buckets verwenden.

Mehrteiliger Upload-Prozess

Der mehrteilige Upload ist ein Prozess in drei Schritten: Sie beginnen den Upload, laden die Teile des Objekts hoch und schließen den mehrteiligen Upload ab, wenn alle Teile hochgeladen wurden. Nach dem vollständigen Eingang der Daten aus dem mehrteiligen Upload erstellt Amazon S3 das Objekt

anhand der hochgeladenen Teile. Dann können Sie auf das Objekt wie auf jedes andere Objekt in Ihrem Bucket zugreifen.

Sie können alle laufenden mehrteiligen Uploads auflisten oder eine Liste der Teile anfordern, die Sie für einen bestimmten Multipart-Upload hochgeladen haben. Alle Vorgänge werden in diesem Abschnitt erklärt.

Initiieren des mehrteiligen Uploads

Wenn Sie eine Anfrage zum Initiieren eines mehrteiligen Uploads senden, gibt Amazon S3 eine Antwort mit einer Upload-ID zurück, als eindeutige Kennung für den Multipart-Upload. Sie müssen diese Upload-ID immer angeben, wenn Sie Teile hochladen, die Teile auflisten, einen Upload abschließen oder einen Upload abbrechen. Wenn Sie Metadaten bereitstellen möchten, die das hochzuladende Objekt beschreiben, müssen sie in der Anforderung auf Initiierung des mehrteiligen Uploads angegeben werden.

Teile hochladen

Beim Hochladen eines Teils müssen Sie zusätzlich zur Upload-ID eine Teilenummer angeben. Sie können jede Teilenummer zwischen 1 und 10.000 wählen. Die Teilenummer identifiziert eindeutig einen Teil und seine Position im Objekt, das Sie hochladen. Die von Ihnen gewählte Teilenummer muss nicht fortlaufend sein (möglich sind z. B. 1, 5 und 14). Wenn Sie einen neuen Teil mit derselben Teilenummer hochladen wie bereits einmal zuvor, wird der früher hochgeladene Teil überschrieben.

Wenn Sie ein Teil hochladen, gibt Amazon S3 ein Entitäts-Tag (ETag) für das Teil als Header in der Antwort zurück. Für jeden Upload eines Teils müssen Sie die Artikelnummer und den ETag Wert aufzeichnen. Sie müssen diese Werte in die spätere Anforderung einschließen, um den mehrteiligen Upload abzuschließen. Jedes Teil wird zum ETag Zeitpunkt des Uploads sein eigenes haben. Sobald der mehrteilige Upload jedoch abgeschlossen und alle Teile konsolidiert sind, werden alle Teile ETag als Prüfsumme von Prüfsummen unter eins angezeigt.

> **Note**
>
> Nachdem Sie einen mehrteiligen Upload gestartet und einen oder mehrere Teile hochgeladen haben, müssen Sie den mehrteiligen Upload abschließen oder abbrechen, damit keine Gebühren für die Speicherung der hochgeladenen Teile anfallen. Erst nach dem Abschluss oder Abbruch eines mehrteiligen Uploads gibt Amazon S3 den Speicher für die Teile frei und stoppt die Berechnung von Gebühren für die Speicherung der Teile.
> Wenn Sie einen mehrteiligen Upload abbrechen, können Sie mit dieser Upload-ID keine Teile mehr hochladen. Wenn der mehrteilige Upload abgebrochen wird, während Teile

hochgeladen werden, können diese Uploads auch nach dem Abbruch des Uploads erfolgreich abgeschlossen werden oder fehlschlagen. Um sicherzustellen, dass der von allen Teilen verbrauchte Speicherplatz freigegeben wird, dürfen Sie einen mehrteiligen Upload erst dann abbrechen, wenn alle Uploads abgeschlossen wurden.

Abschließen eines mehrteiligen Uploads

Wenn Sie einen mehrteiligen Upload abschließen, erstellt Amazon S3 ein Objekt, indem die Teile in aufsteigender Reihenfolge auf Grundlage der Teilenummer verkettet werden. Wenn Sie Metadaten für das Objekt bei der Initiierung des mehrteiligen Uploads bereitgestellt haben, verknüpft Amazon S3 die Metadaten mit dem Objekt. Nach einer erfolgreich ausgeführten Abschlussanforderung sind die Teile nicht mehr vorhanden.

Ihre vollständige mehrteilige Upload-Anfrage muss die Upload-ID und eine Liste der beiden Artikelnummern und der entsprechenden Werte enthalten. ETag Die Amazon S3 S3-Antwort enthält eineETag, die die kombinierten Objektdaten eindeutig identifiziert. Dies ETag ist nicht unbedingt ein MD5 Hash der Objektdaten.

Beispielaufrufe mehrteiliger Uploads

Für dieses Beispiel nehmen wir an, dass Sie einen mehrteiligen Upload für eine 100-GB-Datei generieren. In diesem Fall hätten Sie die folgenden API Aufrufe für den gesamten Prozess. Es würde insgesamt 1002 API Anrufe geben.

- Ein `CreateMultipartUpload`-Aufruf zum Starten des Prozesses
- 1 000 individuelle `UploadPart`-Aufrufe, mit denen jeweils ein Teil von 100 MB für die Gesamtgröße von 100 GB hochgeladen wird
- Ein `CompleteMultipartUpload`-Aufruf zum Beenden des Prozesses

Auflistungen mehrteiliger Uploads

Sie können alle Teile eines bestimmten Multipart-Uploads oder alle laufenden mehrteiligen Uploads auflisten. Die Operation für die Teileauflistung gibt die Teileinformationen zurück, die Sie für einen bestimmten mehrteiligen Upload hochgeladen haben. Für jeden Abruf einer Teileauflistung gibt Amazon S3 die Teileinformationen für einen angegebenen mehrteiligen Upload bis zu maximal 1.000 Teilen zurück. Wenn im Multipart-Upload mehr als 1.000 Teile vorhanden sind, müssen Sie eine Reihe von Anforderungen auf Teileauflistung senden, um alle Teile abzurufen. Beachten Sie,

dass die zurückgegebene Teileauflistung keine Teile enthält, die noch nicht vollständig hochgeladen wurden. Bei Verwendung der Operation Mehrteilige Uploads auflisten können Sie eine Liste aller mehrteiligen Uploads erhalten, die sich in Bearbeitung befinden.

Ein mehrteiliger Upload in Verarbeitung ist ein Upload, den Sie gestartet haben, der aber noch nicht abgeschlossen ist oder abgebrochen wurde. Jeder Anforderung gibt bis zu 1.000 mehrteilige Uploads zurück. Wenn mehr als 1 000 mehrteilige Uploads vorhanden sind, müssen Sie zusätzliche Anforderungen senden, um die verbleibenden mehrteiligen Uploads abzurufen. Verwenden Sie die zurückgegebene Liste nur zur Überprüfung. Verwenden Sie das Ergebnis dieser Auflistung nicht, wenn Sie eine Anforderung für den Abschluss eines mehrteiligen Uploads senden. Pflegen Sie stattdessen Ihre eigene Liste mit den Artikelnummern, die Sie beim Hochladen von Teilen angegeben haben, und den entsprechenden ETag Werten, die Amazon S3 zurückgibt.

Prüfsummen mit mehrteiligen Upload-Operationen

Wenn Sie ein Objekt auf Amazon S3 hochladen, können Sie einen Prüfsummenalgorithmus angeben, den Amazon S3 verwenden soll. Amazon S3 verwendet MD5 standardmäßig zur Überprüfung der Datenintegrität. Sie können jedoch einen zusätzlichen Prüfsummenalgorithmus angeben, der verwendet werden soll. Bei Verwendung MD5 berechnet Amazon S3 die Prüfsumme des gesamten mehrteiligen Objekts, nachdem der Upload abgeschlossen ist. Diese Prüfsumme ist keine Prüfsumme des gesamten Objekts, sondern eine Prüfsumme der Prüfsummen für jeden einzelnen Teil.

Wenn Sie Amazon S3 anweisen, zusätzliche Prüfsummen zu verwenden, berechnet Amazon S3 den Prüfsummenwert für jeden Teil und speichert die Werte. Sie können das API oder verwendenSDK, um den Prüfsummenwert für einzelne Teile abzurufen, indem Sie oder verwenden. `GetObject` `HeadObject` Wenn Sie die Prüfsummenwerte für einzelne Teile von mehrteiligen Uploads abrufen möchten, die noch in Bearbeitung sind, können Sie `ListParts` verwenden.

> **Important**
>
> Wenn Sie einen mehrteiligen Upload mit zusätzlichen Prüfsummen verwenden, müssen die mehrteiligen Teilenummern aufeinander folgende Teilenummern sein. Wenn Sie zusätzliche Prüfsummen verwenden und versuchen, eine mehrteilige Upload-Anfrage mit nicht aufeinanderfolgenden Artikelnummern abzuschließen, generiert Amazon S3 einen Fehler. HTTP `500 Internal Server Error`

Weitere Informationen darüber, wie Prüfsummen bei mehrteiligen Upload-Objekten funktionieren, finden Sie unter. Überprüfung der Objektintegrität

Ein end-to-end Verfahren zum Hochladen eines Objekts mit mehrteiligem Upload mit einer zusätzlichen Prüfsumme finden Sie unter. Tutorial: Laden Sie ein Objekt durch mehrteiligen Upload hoch und überprüfen Sie seine Datenintegrität

Gleichzeitige mehrteilige Upload-Vorgänge

In einer verteilten Entwicklungsumgebung ist es für Ihre Anwendung möglich, mehrere Updates gleichzeitig für dasselbe Objekt zu initiieren. Ihre Anwendung kann möglicherweise mehrere Multipart-Uploads mit demselben Objektschlüssel initiieren. Für jeden dieser Uploads kann Ihre Anwendung Teile hochladen und eine Anfrage auf Abschluss des Uploads an Amazon S3 senden, um das Objekt zu erstellen. Wenn die Buckets die S3-Versioning aktiviert haben, wird beim Abschluss eines mehrteiligen Uploads immer eine neue Version erstellt. Wenn Sie mehrere mehrteilige Uploads initiieren, die denselben Objektschlüssel in einem Bucket mit aktivierter Versionierung verwenden, wird die aktuelle Version des Objekts anhand der zuletzt gestarteten Version des Objekts bestimmt (). `createdDate` Nehmen wir zum Beispiel an, dass Sie um 10:00 Uhr eine CreateMultipartUpload Anfrage für ein Objekt starten. Dann reichen Sie um 11:00 Uhr eine zweite `CreateMultipartUpload` Anfrage für dasselbe Objekt ein. Da die zweite Anfrage zuletzt eingereicht wurde, entspricht das mit der Anfrage um 11:00 Uhr hochgeladene Objekt der aktuellen Version, auch wenn der erste Upload nach dem zweiten abgeschlossen ist. Bei Buckets, für die keine Versioning aktiviert ist, kann es sein, dass zwischen den Zeitpunkten der Initiierung bis zum Abschluss eines Multipart-Uploads eine andere Anforderung vorrangig ist.

> **Note**
>
> Es ist möglich, dass eine andere Anforderung zwischen dem Zeitpunkt der Initiierung und jenem des Abschlusses eines Multipart-Uploads stattgefunden hat. Wenn z. B. ein anderer Vorgang einen Schlüssel löscht, nachdem Sie einen mehrteiligen Upload mit diesem Schlüssel initiiert haben, aber bevor Sie ihn abschließen, kann die Antwort für den Abschluss des Multipart-Uploads möglicherweise eine erfolgreiche Objekterstellung anzeigen, ohne dass Sie das Objekt je zu Gesicht bekommen haben.

Verhindern Sie, dass beim mehrteiligen Upload Objekte mit identischen Schlüsselnamen hochgeladen werden

Sie können überprüfen, ob ein Objekt in Ihrem Bucket vorhanden ist, bevor Sie es mithilfe von bedingten Schreibvorgängen beim Hochladen erstellen. Dadurch kann verhindert werden, dass

vorhandene Daten überschrieben werden. Durch bedingte Schreibvorgänge wird bestätigt, dass sich beim Hochladen kein Objekt mit demselben Schlüsselnamen bereits in Ihrem Bucket befindet.

Sie können bedingte Schreibvorgänge verwenden für PutObjectoder CompleteMultipartUploadAnfragen.

Weitere Informationen zu bedingten Anfragen finden Sie unterFügen Sie Vorbedingungen zu S3-Vorgängen mit bedingten Anfragen hinzu.

Mehrteiliger Upload und Preise

Nachdem Sie einen mehrteiligen Upload gestartet haben, behält Amazon S3 alle Teile bei, bis Sie den Upload abschließen oder abbrechen. Während seiner gesamten Lebensdauer werden Ihnen der gesamte Speicher, die Bandbreite und die Anforderungen für diesen mehrteiligen Upload und die zugehörigen Teile in Rechnung gestellt.

Diese Teile werden gemäß der Speicherklasse berechnet, die beim Hochladen der Teile angegeben wurde. Eine Ausnahme bilden Teile, die in S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive hochgeladen wurden. In Bearbeitung befindliche mehrteilige Teile für die Speicherklasse S3 Glacier Flexible Retrieval werden PUT bis zum Abschluss des Uploads als S3 Glacier Flexible Retrieval Staging Storage zu den Speichertarifen S3 Standard abgerechnet. Darüber hinaus werden sowohl als auch zu den CreateMultipartUpload S3-Standardtarifen UploadPart abgerechnet. Nur die CompleteMultipartUpload Anfrage wird zum Tarif S3 Glacier Flexible Retrieval abgerechnet. In ähnlicher Weise werden in Bearbeitung befindliche mehrteilige Teile für eine Speicherklasse PUT bis zur Speicherklasse S3 Glacier Deep Archive als S3 Glacier Flexible Retrieval Staging Storage zu den S3-Standardspeichertarifen abgerechnet, bis der Upload abgeschlossen ist, wobei nur die CompleteMultipartUpload Anfrage zu den S3 Glacier Deep Archive Archive-Tarifen berechnet wird.

Wenn Sie den mehrteiligen Upload abbrechen, löscht Amazon S3 Upload-Artefakte und alle Teile, die Sie hochgeladen haben. Die Berechnung wird eingestellt. Für das Löschen unvollständiger mehrteiliger Uploads fallen keine Gebühren für vorzeitiges Löschen an, unabhängig von der angegebenen Speicherklasse. Weitere Informationen zu Preisen finden Sie unter Amazon-S3-Preise.

> **Note**
>
> Wir empfehlen, eine Lebenszyklusregel zu konfigurieren, durch die unvollständige mehrteilige Uploads nach einer bestimmten Anzahl von Tagen mit der Aktion `AbortIncompleteMultipartUpload` gelöscht werden, um Ihre Speicherkosten gering zu halten. Weitere Informationen zum Erstellen einer Lebenszyklusregel zum

Löschen unvollständiger mehrteiliger Uploads finden Sie unter Konfigurieren einer Bucket-Lebenszykluskonfiguration zum Löschen unvollständiger mehrteiliger Uploads.

API-Unterstützung für mehrteilige Uploads

Diese Bibliotheken bieten eine Abstraktion auf hoher Ebene, die das Hochladen von mehrteiligen Upload-Objekten vereinfacht. Wenn Ihre Anwendung dies erfordert, können Sie das jedoch direkt verwenden. REST API In den folgenden Abschnitten der Amazon Simple Storage Service API Reference wird das REST API für mehrteilige Uploads beschrieben.

Eine exemplarische Vorgehensweise für mehrteilige Uploads, die AWS Lambda-Funktionen verwendet, finden Sie unter Hochladen großer Objekte auf Amazon S3 mithilfe der mehrteiligen Upload- und Übertragungsbeschleunigung.

- Multipart-Upload erstellen
- Upload Part
- Teil hochladen (kopieren)
- Abschließen eines mehrteiligen Uploads
- Abort Multipart Upload
- Teile auflisten
- List Multipart Uploads

AWS Command Line Interface Unterstützung für mehrteiliges Hochladen

In den folgenden Themen werden die Vorgänge für den mehrteiligen Upload AWS Command Line Interface beschrieben.

- Initiieren eines mehrteiligen Uploads
- Upload Part
- Teil hochladen (kopieren)
- Abschließen eines mehrteiligen Uploads
- Abort Multipart Upload
- Teile auflisten
- List Multipart Uploads

AWS SDK-Unterstützung für mehrteilige Uploads

Sie können eine verwenden AWS SDKs, um ein Objekt in Teilen hochzuladen. Eine Liste der AWS SDKs unterstützten API Aktionen finden Sie unter:

- Multipart-Upload erstellen
- Upload Part
- Teil hochladen (kopieren)
- Abschließen eines mehrteiligen Uploads
- Abort Multipart Upload
- Teile auflisten
- List Multipart Uploads

Mehrteiliger Upload API und Berechtigungen

Sie müssen über die erforderlichen Berechtigungen verfügen, um die Multipart-Upload-Vorgänge zu verwenden. Sie können Zugriffskontrolllisten (ACLs), die Bucket-Richtlinie oder die Benutzerrichtlinie verwenden, um Einzelpersonen Berechtigungen zur Ausführung dieser Operationen zu gewähren. In der folgenden Tabelle sind die erforderlichen Berechtigungen für verschiedene mehrteilige Upload-Operationen aufgeführtACLs, wenn Sie eine Bucket-Richtlinie oder eine Benutzerrichtlinie verwenden.

Aktion	Erforderliche Berechtigungen
Multipart-Upload erstellen	Sie müssen die Aktion `s3:PutObject` für ein Objekt ausführen können, um einen Multipart-Upload zu erstellen. Der Bucket-Eigentümer kann anderen Prinzipalen erlauben, die Aktion `s3:PutObject` auszuführen.
Initiieren eines mehrteiligen Uploads	Sie müssen die Aktion `s3:PutObject` für ein Objekt ausführen können, um einen Multipart-Upload zu initiieren. Der Bucket-Eigentümer kann anderen Prinzipalen erlauben, die Aktion `s3:PutObject` auszuführen.

Aktion	Erforderliche Berechtigungen
Initiator	Containerelement, das identifiziert, wer den Multipart-Upload initiiert hat. Wenn es sich bei dem Initiator um einen handelt AWS-Konto, liefert dieses Element dieselben Informationen wie das Owner-Element. Wenn der Initiator ein IAM Benutzer ist, stellt dieses Element den Benutzer ARN und den Anzeigenamen bereit.
Upload Part	Sie müssen die Aktion `s3:PutObject` für ein Objekt ausführen können, um einen Teil hochzuladen. Der Bucket-Eigentümer muss dem Initiator erlauben, die Aktion `s3:PutObject` für ein Objekt auszuführen, damit der Initiator einen Teil für dieses Objekt hochladen kann.
Teil hochladen (kopieren)	Sie müssen die Aktion `s3:PutObject` für ein Objekt ausführen können, um einen Teil hochzuladen. Da Sie einen Teil eines vorhandenen Objekts hochladen, müssen Sie die Berechtigung `s3:GetObject` für das Quellobjekt besitzen. Damit der Initiator einen Teil eines Objekts hochladen kann, muss der Bucket-Besitzer dem Initiator eine Genehmigung für die Ausführung der Aktion `s3:PutObject` für das Objekt erteilen.
Abschließen eines mehrteiligen Uploads	Sie müssen die Aktion `s3:PutObject` für ein Objekt ausführen können, um einen Multipart-Upload abzuschließen. Der Bucket-Eigentümer muss dem Initiator erlauben, die Aktion `s3:PutObject` für ein Objekt auszuführen, damit der Initiator den mehrteiligen Upload für dieses Objekt abschließen kann.

Aktion	Erforderliche Berechtigungen
Mehrteiligen Upload abbrechen	Sie müssen die Aktion `s3:AbortMultipartUpload` ausführen dürfen, um einen mehrteiligen Upload abzubrechen. Standardmäßig dürfen der Bucket-Besitzer und der Initiator des mehrteiligen Uploads diese Aktion im Rahmen der IAM Bucket-Richtlinien ausführen. Wenn der Initiator ein IAM Benutzer ist, darf dieser Benutzer den AWS-Konto mehrteiligen Upload auch beenden. Bei VPC Endpunktrichtlinien erhält der Initiator des mehrteiligen Uploads nicht automatisch die Erlaubnis, die Aktion auszuführen. `s3:AbortMultipartUpload` Zusätzlich zu diesen Standardberechtigungen, kann der Bucket-Eigentümer anderen Prinzipalen erlauben, die Aktion `s3:AbortMultipartUpload` auszuführen. Der Bucket-Eigentümer kann jedem Prinzipal verbieten, die Aktion `s3:AbortMultipartUpload` auszuführen.
Teile auflisten	Sie müssen die Aktion `s3:ListMultipartUploadParts` ausführen können, um Teileauflistungen in einem mehrteiligen Upload anzufragen. Standardmäßig kann der Bucket-Eigentümer Teileauflistungen von Multipart-Uploads in seinem Bucket anfordern. Der Initiator des mehrteiligen Uploads hat die Berechtigung, Teileauflistungen des spezifischen Multipart-Uploads aufzulisten. Wenn der Initiator des mehrteiligen Uploads ein IAM Benutzer ist, hat der Benutzer, der die AWS-Konto Kontrolle hat, IAM auch die Berechtigung, Teile dieses Uploads aufzulisten. Zusätzlich zu diesen Standardberechtigungen, kann der Bucket-Eigentümer anderen Prinzipalen erlauben, die Aktion `s3:ListMultipartUploadParts` auszuführen. Der Bucket-Eigentümer kann jedem Prinzipal auch verbieten, die Aktion `s3:ListMultipartUploadParts` auszuführen.

Aktion	Erforderliche Berechtigungen
List Multipart Uploads	Sie müssen die Aktion `s3:ListBucketMultipartUploads` für einen Bucket ausführen können, um eine Multipart-Uploads-Auflistung für diesen Bucket anzufragen. Zusätzlich zu diesen Standardberechtigungen, kann der Bucket-Eigentümer anderen Prinzipalen erlauben, die Aktion `s3:ListBucketMultipartUploads` für ein Bucket auszuführen.
AWS KMS Die damit verbundenen Berechtigungen verschlüsseln und entschlüsseln	Um einen mehrteiligen Upload mit Verschlüsselung unter Verwendung eines AWS Key Management Service (AWS KMS) KMS -Schlüssels durchzuführen, muss der Anforderer über Berechtigungen für die `kms:GenerateDataKey` Aktionen `kms:Decrypt` und für den Schlüssel verfügen. Der Anforderer muss außerdem über die Berechtigungen für die `kms:GenerateDataKey` Aktion verfügen CreateMultipartUploadAPI. Dann benötigt der Anforderer Berechtigungen für die `kms:Decrypt` Aktion auf der UploadPart und UploadPartCopyAPIs. Diese Berechtigungen sind erforderlich, da Amazon S3 Daten aus den verschlüsselten Teilen der Datei entschlüsseln und lesen muss, bevor es den Multipart-Upload vornehmen kann. Wenn Ihr IAM Benutzer oder Ihre Rolle dem KMS Schlüssel entspricht AWS-Konto , müssen Sie über diese Berechtigungen für die Schlüsselrichtlinie verfügen. Wenn Ihr IAM Benutzer oder Ihre Rolle zu einem anderen Konto als dem KMS Schlüssel gehört, müssen Sie über die erforderlichen Berechtigungen sowohl für die Schlüsselrichtlinie als auch für Ihren IAM Benutzer oder Ihre Rolle verfügen.

Informationen zur Beziehung zwischen ACL Berechtigungen und Berechtigungen in Zugriffsrichtlinien finden Sie unterZuordnung von ACL Berechtigungen und Zugriffsrichtlinienberechtigungen. Informationen zu IAM Benutzern, Rollen und bewährten Methoden finden Sie im Benutzerhandbuch unter IAMIdentitäten (Benutzer, Benutzergruppen und Rollen). IAM

Themen

- Konfigurieren einer Bucket-Lebenszykluskonfiguration zum Löschen unvollständiger mehrteiliger Uploads
- Hochladen eines Objekts mit Multipart-Upload

- Hochladen eines Verzeichnisses mithilfe der High-Level-Methode. NET TransferUtility Klasse
- Auflisten von mehrteiligen Uploads
- Verfolgen eines mehrteiligen Uploads mit dem AWS SDKs
- Abbrechen eines mehrteiligen Uploads
- Kopieren eines Objekts mit Multipart-Upload
- Tutorial: Laden Sie ein Objekt durch mehrteiligen Upload hoch und überprüfen Sie seine Datenintegrität
- EInschränkungen mehrteiliger Amazon-S3-Uploads

Konfigurieren einer Bucket-Lebenszykluskonfiguration zum Löschen unvollständiger mehrteiliger Uploads

Als bewährte Methode empfehlen wir Ihnen, eine Lebenszyklusregel mit der Aktion `AbortIncompleteMultipartUpload` zu konfigurieren, um Ihre Speicherkosten zu minimieren. Weitere Informationen zum Abbruch eines mehrteiligen Uploads finden Sie unter Abbrechen eines mehrteiligen Uploads.

Amazon S3 unterstützt eine Bucket-Lebenszyklusregel, mit der Sie Amazon S3 anweisen können, unvollständige mehrteilige Uploads abzubrechen, die nicht innerhalb einer bestimmten Anzahl von Tagen nach der Initiierung abgeschlossen werden. Wenn ein mehrteiliger Upload nicht innerhalb des angegebenen Zeitrahmens abgeschlossen wird, wird er für eine Abbruchoperation zugelassen. Amazon S3 bricht den mehrteiligen Upload dann ab und löscht alle diesem mehrteiligen Upload zugeordneten Teile. Diese Regel gilt sowohl für bestehende mehrteilige Uploads als auch für solche, die Sie später erstellen.

Das folgende Beispiel zeigt eine Lebenszykluskonfiguration, die eine Regel mit der Aktion `AbortIncompleteMultipartUpload` spezifiziert.

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
        <Prefix></Prefix>
        <Status>Enabled</Status>
        <AbortIncompleteMultipartUpload>
          <DaysAfterInitiation>7</DaysAfterInitiation>
        </AbortIncompleteMultipartUpload>
    </Rule>
```

```
</LifecycleConfiguration>
```

In dem Beispiel gibt die Regel keinen Wert für das `Prefix`-Element (Objektschlüsselnamen-Präfix) an. Daher gilt die Regel für alle Objekte im Bucket, für die Sie mehrteilige Uploads initiiert haben. Alle mehrteiligen Uploads, die initiiert wurden und nicht innerhalb von sieben Tagen abgeschlossen wurden, werden für eine Abbruchoperation zugelassen. Die Abbruchaktion hat keine Auswirkung auf abgeschlossene mehrteilige Uploads.

Weitere Informationen zur Bucket-Lebenszykluskonfiguration finden Sie unter Verwaltung des Lebenszyklus von Objekten.

> **Note**
>
> Wenn der mehrteilige Upload innerhalb der in der Regel angegebenen Anzahl von Tagen abgeschlossen wurde, gilt die Lebenszyklusaktion `AbortIncompleteMultipartUpload` nicht (das heißt, Amazon S3 führt keine Aktion aus). Außerdem gilt diese Aktion nicht für Objekte. Durch diese Lebenszyklus-Aktion werden keine Objekte gelöscht. Darüber hinaus fallen keine Gebühren für das vorzeitige Löschen für den S3-Lebenszyklus an, wenn Sie Teile von unvollständigen mehrteiligen Uploads entfernen.

Verwenden der S3-Konsole

Um unvollständige mehrteilige Uploads automatisch zu verwalten, können Sie die S3-Konsole verwenden, um eine Lebenszyklusrichtlinie zu erstellen und unvollständige mehrteilige Upload-Bytes aus dem Bucket nach einer bestimmten Anzahl von Tagen ablaufen zu lassen. Im folgenden Verfahren wird gezeigt, wie Sie eine Lebenszyklusregel hinzufügen, um unvollständige mehrteilige Uploads nach 7 Tagen zu löschen. Weitere Informationen über das Hinzufügen von Lebenszyklusregeln finden Sie unter Einstellung einer S3-Lifecycle-Konfiguration für einen Bucket.

So fügen Sie eine Lebenszyklusregel hinzu, um unvollständige mehrteilige Uploads, die älter als 7 Tage sind, abzubrechen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie eine Lebenszyklusregel erstellen möchten.

3. Wählen Sie den Tab Management (Verwaltung) und dann die Option Create lifecycle rule (Lebenszyklusregel erstellen).
4. Geben Sie unter Lifecycle rule name (Name der Lebenszyklusregel) einen Namen für Ihre Regel ein.

   Der Name muss innerhalb des Buckets eindeutig sein.
5. Wählen Sie den Umfang der Lebenszyklusregel:

   - Wenn Sie eine Lebenszyklusregel für alle Objekte mit einem bestimmten Präfix erstellen möchten, wählen Sie Limit the scope of this rule using one or more filters (Den Geltungsbereich dieser Regel mithilfe eines oder mehrerer Filter einschränken) aus und geben Sie das Präfix im Feld Prefix (Präfix) ein.
   - Wenn Sie diese Lebenszyklusregel auf alle Objekte im Bucket anwenden möchten, wählen Sie This rule applies to all objects in the bucket (Diese Regel gilt für alle Objekte in dem Bucket) aus und klicken Sie auf I acknowledge that this rule applies to all objects in the bucket (Ich bestätige, dass diese Regel für alle Objekte in dem Bucket gilt).
6. Wählen Sie unter Lifecycle rule actions (Lebenszyklusregelaktionen) die Option Delete expired object delete markers or incomplete multipart uploads (Abgelaufene Objektlöschmarken oder unvollständige mehrteilige Uploads löschen) aus.
7. Wählen Sie unter Delete expired object delete markers or incomplete multipart uploads (Abgelaufene Objektlöschmarken oder unvollständige mehrteilige Uploads löschen) die Option Delete incomplete multipart uploads (Unvollständige mehrteilige Uploads löschen) aus.
8. Geben Sie im Feld Number of days (Anzahl der Tage) die Anzahl der Tage ein, nach denen unvollständige mehrteilige Uploads gelöscht werden sollen (in diesem Beispiel 7 Tage).
9. Wählen Sie Regel erstellen aus.

Verwenden Sie den AWS CLI

Der folgende Befehl `put-bucket-lifecycle-configuration`AWS Command Line Interface (AWS CLI) fügt die Lebenszykluskonfiguration für den angegebenen Bucket hinzu. Zum Verwenden dieses Befehls ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws s3api put-bucket-lifecycle-configuration  \
        --bucket amzn-s3-demo-bucket  \
        --lifecycle-configuration filename-containing-lifecycle-configuration
```

Das folgende Beispiel zeigt, wie Sie eine Lebenszyklusregel hinzufügen, um unvollständige mehrteilige Uploads mithilfe der AWS CLI abzubrechen. Er enthält ein Beispiel für eine JSON Lebenszykluskonfiguration zum Abbrechen unvollständiger mehrteiliger Uploads, die älter als 7 Tage sind.

Um die CLI Befehle in diesem Beispiel zu verwenden, ersetzen Sie die durch Ihre Informationen*user input placeholders*.

So fügen Sie eine Lebenszyklusregel hinzu, um unvollständige mehrteilige Uploads abzubrechen

1. Richten Sie das ein AWS CLI. Anweisungen finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS CLI in der Amazon S3 API S3-Referenz.
2. Speichern Sie die folgende Beispiel-Lebenszykluskonfiguration in einer Datei (zum Beispiel *lifelycle.json*). Diese Beispielkonfiguration spezifiziert ein leeres Präfix und gilt daher für alle Objekte im Bucket. Sie können ein Präfix angeben, um die Konfiguration auf eine Teilmenge von Objekten zu beschränken.

```
{
    "Rules": [
        {
            "ID": "Test Rule",
            "Status": "Enabled",
            "Filter": {
                "Prefix": ""
            },
            "AbortIncompleteMultipartUpload": {
                "DaysAfterInitiation": 7
            }
        }
    ]
}
```

3. Führen Sie den folgenden CLI Befehl aus, um diese Lebenszykluskonfiguration für Ihren Bucket festzulegen.

```
aws s3api put-bucket-lifecycle-configuration   \
--bucket amzn-s3-demo-bucket  \
--lifecycle-configuration file://lifecycle.json
```

4. Um zu überprüfen, ob die Lebenszykluskonfiguration für Ihren Bucket festgelegt wurde, rufen Sie die Lebenszykluskonfiguration mit dem folgenden `get-bucket-lifecycle`-Befehl ab.

```
aws s3api get-bucket-lifecycle  \
--bucket amzn-s3-demo-bucket
```

5. Verwenden Sie zum Löschen der Lebenszykluskonfiguration den folgenden `delete-bucket-lifecycle`-Befehl.

```
aws s3api delete-bucket-lifecycle \
--bucket amzn-s3-demo-bucket
```

Hochladen eines Objekts mit Multipart-Upload

Sie können den Multipart-Upload verwenden, um ein einzelnes Objekt programmgesteuert auf Amazon S3 hochzuladen. Jedes Objekt wird als Satz von Teilen hochgeladen. Jeder Teil ist ein zusammenhängender Teil der Daten des Objekts. Sie können diese Objektteile unabhängig und in beliebiger Reihenfolge hochladen. Wenn die Übertragung eines Teils fehlschlägt, können Sie das Teil erneut übertragen, ohne dass dies Auswirkungen auf andere Teile hat. Nachdem alle Teile Ihres Objekts hochgeladen sind, fügt Amazon S3 diese Teile zusammen und erstellt das Objekt.

Eine Anleitung zum end-to-end Hochladen eines Objekts mit mehrteiligem Upload mit zusätzlicher Prüfsumme finden Sie unter. Tutorial: Laden Sie ein Objekt durch mehrteiligen Upload hoch und überprüfen Sie seine Datenintegrität

Im folgenden Abschnitt wird gezeigt, wie der mehrteilige Upload mit den Zeichen, und verwendet wird. AWS Command Line Interface AWS SDKs

Verwenden der S3-Konsole

Sie können beliebige Dateitypen – Bilder, Backups, Daten, Filme usw. – in einen S3-Bucket hochladen. Die maximale Größe einer Datei, die Sie über die Amazon-S3-Konsole hochladen können, ist 160 GB. Um eine Datei mit mehr als 160 GB hochzuladen, verwenden Sie AWS Command Line Interface (AWS CLI) AWS SDKs, oder Amazon S3 RESTAPI.

Anweisungen zum Hochladen eines Objekts über den finden Sie AWS Management Console unterObjekte hochladen.

Mit dem AWS CLI

Im Folgenden werden die Amazon S3 S3-Operationen für den mehrteiligen Upload mit dem AWS CLI beschrieben.

- Initiieren eines mehrteiligen Uploads
- Upload Part
- Teil hochladen (kopieren)
- Abschließen eines mehrteiligen Uploads
- Abort Multipart Upload
- Teile auflisten
- List Multipart Uploads

Unter Verwendung der REST API

In den folgenden Abschnitten der Amazon Simple Storage Service API Reference wird das REST API für mehrteilige Uploads beschrieben.

- Initiieren eines mehrteiligen Uploads
- Upload Part
- Abschließen eines mehrteiligen Uploads
- Mehrteiligen Upload abbrechen
- Teile auflisten
- List Multipart Uploads

Verwenden von AWS SDKs (auf hoher EbeneAPI)

Einige bieten AWS SDKs eine höhere Ebene an, API die den mehrteiligen Upload vereinfacht, indem die verschiedenen API Operationen, die zum Abschließen eines mehrteiligen Uploads erforderlich sind, zu einem einzigen Vorgang zusammengefasst werden. Weitere Informationen finden Sie unter Hochladen und Kopieren von Objekten mit mehrteiligen Uploads.

Wenn Sie mehrteilige Uploads anhalten und fortsetzen müssen, die Teilgrößen während des Uploads variieren müssen oder die Größe der Daten nicht im Voraus kennen, verwenden Sie die Low-Level-Methoden. API Die API Low-Level-Methoden für mehrteilige Uploads bieten zusätzliche Funktionen. Weitere Informationen finden Sie unter. Verwenden von AWS SDKs (APILow-Level)

.NET

Um eine Datei in einen S3-Bucket hochzuladen, verwenden Sie die Klasse `TransferUtility`. Beim Hochladen von Daten aus einer Datei müssen Sie den Schlüsselnamen des Objekts

angeben. Wenn Sie dies nicht tun, API verwendet der den Dateinamen als Schlüsselnamen. Beim Hochladen von Daten aus einem Stream müssen Sie den Schlüsselnamen des Objekts angeben.

Verwenden Sie die Klasse, um erweiterte Upload-Optionen festzulegen, z. B. die Größe des Teils, die Anzahl der Threads beim gleichzeitigen Hochladen der Teile, Metadaten, die Speicherklasse oder. ACL `TransferUtilityUploadRequest`

> **Note**
>
> Wenn Sie einen Stream für die Datenquelle verwenden, führt die `TransferUtility`-Klasse keine gleichzeitigen Uploads durch.

Im folgenden C#-Beispiel wird eine Datei in mehreren Teilen in einen Amazon-S3-Bucket hochgeladen. Es veranschaulicht, wie Sie zahlreiche `TransferUtility.Upload`-Überladungen zum Hochladen einer Datei verwenden. Jeder nachfolgende Aufruf zum Hochladen ersetzt den vorherigen Upload. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem for. AWS SDK NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.IO;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadFileMPUHighLevelAPITest
    {
        private const string bucketName = "*** provide bucket name ***";
        private const string keyName = "*** provide a name for the uploaded object
 ***";
        private const string filePath = "*** provide the full path name of the file
 to upload ***";
       // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
```

```
        {
            s3Client = new AmazonS3Client(bucketRegion);
            UploadFileAsync().Wait();
        }

        private static async Task UploadFileAsync()
        {
            try
            {
                var fileTransferUtility =
                    new TransferUtility(s3Client);

                // Option 1. Upload a file. The file name is used as the object key
name.
                await fileTransferUtility.UploadAsync(filePath, bucketName);
                Console.WriteLine("Upload 1 completed");

                // Option 2. Specify object key name explicitly.
                await fileTransferUtility.UploadAsync(filePath, bucketName,
keyName);
                Console.WriteLine("Upload 2 completed");

                // Option 3. Upload data from a type of System.IO.Stream.
                using (var fileToUpload =
                    new FileStream(filePath, FileMode.Open, FileAccess.Read))
                {
                    await fileTransferUtility.UploadAsync(fileToUpload,
                                               bucketName, keyName);
                }
                Console.WriteLine("Upload 3 completed");

                // Option 4. Specify advanced settings.
                var fileTransferUtilityRequest = new TransferUtilityUploadRequest
                {
                    BucketName = bucketName,
                    FilePath = filePath,
                    StorageClass = S3StorageClass.StandardInfrequentAccess,
                    PartSize = 6291456, // 6 MB.
                    Key = keyName,
                    CannedACL = S3CannedACL.PublicRead
                };
                fileTransferUtilityRequest.Metadata.Add("param1", "Value1");
                fileTransferUtilityRequest.Metadata.Add("param2", "Value2");
```

```
                await fileTransferUtility.UploadAsync(fileTransferUtilityRequest);
                Console.WriteLine("Upload 4 completed");
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }

        }
    }
}
```

JavaScript

Example

Laden Sie eine große Datei hoch.

```
import { S3Client } from "@aws-sdk/client-s3";
import { Upload } from "@aws-sdk/lib-storage";

import {
  ProgressBar,
  logger,
} from "@aws-doc-sdk-examples/lib/utils/util-log.js";

const twentyFiveMB = 25 * 1024 * 1024;

export const createString = (size = twentyFiveMB) => {
  return "x".repeat(size);
};

/**
 * Create a 25MB file and upload it in parts to the specified
 * Amazon S3 bucket.
 * @param {{ bucketName: string, key: string }}
 */
export const main = async ({ bucketName, key }) => {
```

```
  const str = createString();
  const buffer = Buffer.from(str, "utf8");
  const progressBar = new ProgressBar({
    description: `Uploading "${key}" to "${bucketName}"`,
    barLength: 30,
  });

  try {
    const upload = new Upload({
      client: new S3Client({}),
      params: {
        Bucket: bucketName,
        Key: key,
        Body: buffer,
      },
    });

    upload.on("httpUploadProgress", ({ loaded, total }) => {
      progressBar.update({ current: loaded, total });
    });

    await upload.done();
  } catch (caught) {
    if (caught instanceof Error && caught.name === "AbortError") {
      logger.error(`Multipart upload was aborted. ${caught.message}`);
    } else {
      throw caught;
    }
  }
};
```

Example

Laden Sie eine große Datei herunter.

```
import { GetObjectCommand, NoSuchKey, S3Client } from "@aws-sdk/client-s3";
import { createWriteStream, rmSync } from "node:fs";

const s3Client = new S3Client({});
const oneMB = 1024 * 1024;

export const getObjectRange = ({ bucket, key, start, end }) => {
  const command = new GetObjectCommand({
```

```
    Bucket: bucket,
    Key: key,
    Range: `bytes=${start}-${end}`,
  });

  return s3Client.send(command);
};

/**
 * @param {string | undefined} contentRange
 */
export const getRangeAndLength = (contentRange) => {
  const [range, length] = contentRange.split("/");
  const [start, end] = range.split("-");
  return {
    start: Number.parseInt(start),
    end: Number.parseInt(end),
    length: Number.parseInt(length),
  };
};

export const isComplete = ({ end, length }) => end === length - 1;

const downloadInChunks = async ({ bucket, key }) => {
  const writeStream = createWriteStream(
    fileURLToPath(new URL(`./${key}`, import.meta.url)),
  ).on("error", (err) => console.error(err));

  let rangeAndLength = { start: -1, end: -1, length: -1 };

  while (!isComplete(rangeAndLength)) {
    const { end } = rangeAndLength;
    const nextRange = { start: end + 1, end: end + oneMB };

    const { ContentRange, Body } = await getObjectRange({
      bucket,
      key,
      ...nextRange,
    });
    console.log(`Downloaded bytes ${nextRange.start} to ${nextRange.end}`);

    writeStream.write(await Body.transformToByteArray());
    rangeAndLength = getRangeAndLength(ContentRange);
  }
```

```
};

/**
 * Download a large object from and Amazon S3 bucket.
 *
 * When downloading a large file, you might want to break it down into
 * smaller pieces. Amazon S3 accepts a Range header to specify the start
 * and end of the byte range to be downloaded.
 *
 * @param {{ bucketName: string, key: string }}
 */
export const main = async ({ bucketName, key }) => {
  try {
    await downloadInChunks({
      bucket: bucketName,
      key: key,
    });
  } catch (caught) {
    if (caught instanceof NoSuchKey) {
      console.error(`Failed to download object. No such key "${key}".`);
      rmSync(key);
    }
  }
};
```

Go

Weitere Informationen zum Go-Codebeispiel für mehrteiligen Upload finden Sie unter Hochladen oder Herunterladen großer Dateien zu und von Amazon S3 mit einem AWS SDK.

Example

Laden Sie ein großes Objekt hoch, indem Sie einen Upload-Manager verwenden, um die Daten in Teile zu zerlegen und sie gleichzeitig hochzuladen.

```
import (
 "bytes"
 "context"
 "errors"
 "fmt"
 "io"
 "log"
```

```
 "os"
 "time"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/feature/s3/manager"
 "github.com/aws/aws-sdk-go-v2/service/s3"
 "github.com/aws/aws-sdk-go-v2/service/s3/types"
 "github.com/aws/smithy-go"
)

// BucketBasics encapsulates the Amazon Simple Storage Service (Amazon S3) actions
// used in the examples.
// It contains S3Client, an Amazon S3 service client that is used to perform bucket
// and object actions.
type BucketBasics struct {
 S3Client *s3.Client
}
```

```
// UploadLargeObject uses an upload manager to upload data to an object in a bucket.
// The upload manager breaks large data into parts and uploads the parts
 concurrently.
func (basics BucketBasics) UploadLargeObject(ctx context.Context, bucketName string,
 objectKey string, largeObject []byte) error {
 largeBuffer := bytes.NewReader(largeObject)
 var partMiBs int64 = 10
 uploader := manager.NewUploader(basics.S3Client, func(u *manager.Uploader) {
  u.PartSize = partMiBs * 1024 * 1024
 })
 _, err := uploader.Upload(ctx, &s3.PutObjectInput{
  Bucket: aws.String(bucketName),
  Key:    aws.String(objectKey),
  Body:   largeBuffer,
 })
 if err != nil {
  var apiErr smithy.APIError
  if errors.As(err, &apiErr) && apiErr.ErrorCode() == "EntityTooLarge" {
   log.Printf("Error while uploading object to %s. The object is too large.\n"+
    "The maximum size for a multipart upload is 5TB.", bucketName)
  } else {
   log.Printf("Couldn't upload large object to %v:%v. Here's why: %v\n",
    bucketName, objectKey, err)
```

```
  }
 } else {
  err = s3.NewObjectExistsWaiter(basics.S3Client).Wait(
   ctx, &s3.HeadObjectInput{Bucket: aws.String(bucketName), Key:
 aws.String(objectKey)}, time.Minute)
  if err != nil {
   log.Printf("Failed attempt to wait for object %s to exist.\n", objectKey)
  }
 }

 return err
}
```

Example

Laden Sie ein großes Objekt herunter, indem Sie einen Download-Manager verwenden, um die Daten in Teilen abzurufen und sie gleichzeitig herunterzuladen.

```
// DownloadLargeObject uses a download manager to download an object from a bucket.
// The download manager gets the data in parts and writes them to a buffer until all
 of
// the data has been downloaded.
func (basics BucketBasics) DownloadLargeObject(ctx context.Context, bucketName
 string, objectKey string) ([]byte, error) {
 var partMiBs int64 = 10
 downloader := manager.NewDownloader(basics.S3Client, func(d *manager.Downloader) {
  d.PartSize = partMiBs * 1024 * 1024
 })
 buffer := manager.NewWriteAtBuffer([]byte{})
 _, err := downloader.Download(ctx, buffer, &s3.GetObjectInput{
  Bucket: aws.String(bucketName),
  Key:    aws.String(objectKey),
 })
 if err != nil {
  log.Printf("Couldn't download large object from %v:%v. Here's why: %v\n",
   bucketName, objectKey, err)
 }
 return buffer.Bytes(), err
}
```

PHP

In diesem Thema wird erklärt, wie Sie die `Aws\S3\Model\MultipartUpload\UploadBuilder` High-Level-Klasse aus dem AWS SDK for PHP für mehrteilige Datei-Uploads verwenden. Weitere Informationen zu AWS SDK for Ruby finden Sie API unter [AWS SDKfor Ruby — Version 2.]

Das folgende PHP Beispiel lädt eine Datei in einen Amazon S3 S3-Bucket hoch. Das Beispiel veranschaulicht, wie Sie Parameter für das `MultipartUploader`-Objekt festlegen.

```
 require 'vendor/autoload.php';

use Aws\Exception\MultipartUploadException;
use Aws\S3\MultipartUploader;
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Prepare the upload parameters.
$uploader = new MultipartUploader($s3, '/path/to/large/file.zip', [
    'bucket' => $bucket,
    'key'    => $keyname
]);

// Perform the upload.
try {
    $result = $uploader->upload();
    echo "Upload complete: {$result['ObjectURL']}" . PHP_EOL;
} catch (MultipartUploadException $e) {
    echo $e->getMessage() . PHP_EOL;
}
```

Python

Im folgenden Beispiel wird ein Objekt mithilfe des mehrteiligen Uploads auf hoher Ebene von Python API (der `TransferManager` Klasse) geladen.

```
import sys
import threading

import boto3
from boto3.s3.transfer import TransferConfig


MB = 1024 * 1024
s3 = boto3.resource("s3")


class TransferCallback:
    """
    Handle callbacks from the transfer manager.

    The transfer manager periodically calls the __call__ method throughout
    the upload and download process so that it can take action, such as
    displaying progress to the user and collecting data about the transfer.
    """

    def __init__(self, target_size):
        self._target_size = target_size
        self._total_transferred = 0
        self._lock = threading.Lock()
        self.thread_info = {}

    def __call__(self, bytes_transferred):
        """
        The callback method that is called by the transfer manager.

        Display progress during file transfer and collect per-thread transfer
        data. This method can be called by multiple threads, so shared instance
        data is protected by a thread lock.
        """
        thread = threading.current_thread()
        with self._lock:
            self._total_transferred += bytes_transferred
            if thread.ident not in self.thread_info.keys():
                self.thread_info[thread.ident] = bytes_transferred
            else:
                self.thread_info[thread.ident] += bytes_transferred

            target = self._target_size * MB
```

```
            sys.stdout.write(
                f"\r{self._total_transferred} of {target} transferred "
                f"({(self._total_transferred / target) * 100:.2f}%)."
            )
            sys.stdout.flush()


def upload_with_default_configuration(
    local_file_path, bucket_name, object_key, file_size_mb
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, using the default
    configuration.
    """
    transfer_callback = TransferCallback(file_size_mb)
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def upload_with_chunksize_and_meta(
    local_file_path, bucket_name, object_key, file_size_mb, metadata=None
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, setting a
    multipart chunk size and adding metadata to the Amazon S3 object.

    The multipart chunk size controls the size of the chunks of data that are
    sent in the request. A smaller chunk size typically results in the transfer
    manager using more threads for the upload.

    The metadata is a set of key-value pairs that are stored with the object
    in Amazon S3.
    """
    transfer_callback = TransferCallback(file_size_mb)

    config = TransferConfig(multipart_chunksize=1 * MB)
    extra_args = {"Metadata": metadata} if metadata else None
    s3.Bucket(bucket_name).upload_file(
        local_file_path,
        object_key,
        Config=config,
        ExtraArgs=extra_args,
```

```
        Callback=transfer_callback,
    )
    return transfer_callback.thread_info


def upload_with_high_threshold(local_file_path, bucket_name, object_key,
 file_size_mb):
    """
    Upload a file from a local folder to an Amazon S3 bucket, setting a
    multipart threshold larger than the size of the file.

    Setting a multipart threshold larger than the size of the file results
    in the transfer manager sending the file as a standard upload instead of
    a multipart upload.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(multipart_threshold=file_size_mb * 2 * MB)
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, Config=config, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def upload_with_sse(
    local_file_path, bucket_name, object_key, file_size_mb, sse_key=None
):
    """
    Upload a file from a local folder to an Amazon S3 bucket, adding server-side
    encryption with customer-provided encryption keys to the object.

    When this kind of encryption is specified, Amazon S3 encrypts the object
    at rest and allows downloads only when the expected encryption key is
    provided in the download request.
    """
    transfer_callback = TransferCallback(file_size_mb)
    if sse_key:
        extra_args = {"SSECustomerAlgorithm": "AES256", "SSECustomerKey": sse_key}
    else:
        extra_args = None
    s3.Bucket(bucket_name).upload_file(
        local_file_path, object_key, ExtraArgs=extra_args,
 Callback=transfer_callback
    )
    return transfer_callback.thread_info
```

```
def download_with_default_configuration(
    bucket_name, object_key, download_file_path, file_size_mb
):
    """
    Download a file from an Amazon S3 bucket to a local folder, using the
    default configuration.
    """
    transfer_callback = TransferCallback(file_size_mb)
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def download_with_single_thread(
    bucket_name, object_key, download_file_path, file_size_mb
):
    """
    Download a file from an Amazon S3 bucket to a local folder, using a
    single thread.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(use_threads=False)
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, Config=config, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def download_with_high_threshold(
    bucket_name, object_key, download_file_path, file_size_mb
):
    """
    Download a file from an Amazon S3 bucket to a local folder, setting a
    multipart threshold larger than the size of the file.

    Setting a multipart threshold larger than the size of the file results
    in the transfer manager sending the file as a standard download instead
    of a multipart download.
    """
    transfer_callback = TransferCallback(file_size_mb)
    config = TransferConfig(multipart_threshold=file_size_mb * 2 * MB)
```

```
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, Config=config, Callback=transfer_callback
    )
    return transfer_callback.thread_info


def download_with_sse(
    bucket_name, object_key, download_file_path, file_size_mb, sse_key
):
    """
    Download a file from an Amazon S3 bucket to a local folder, adding a
    customer-provided encryption key to the request.

    When this kind of encryption is specified, Amazon S3 encrypts the object
    at rest and allows downloads only when the expected encryption key is
    provided in the download request.
    """
    transfer_callback = TransferCallback(file_size_mb)

    if sse_key:
        extra_args = {"SSECustomerAlgorithm": "AES256", "SSECustomerKey": sse_key}
    else:
        extra_args = None
    s3.Bucket(bucket_name).Object(object_key).download_file(
        download_file_path, ExtraArgs=extra_args, Callback=transfer_callback
    )
    return transfer_callback.thread_info
```

Verwenden von AWS SDKs (APILow-Level)

Das AWS SDK macht ein Low-Level verfügbarAPI, das Amazon S3 REST API für mehrteilige Uploads sehr ähnlich ist (siehe. Hochladen und Kopieren von Objekten mit mehrteiligen Uploads Verwenden Sie das Low-LevelAPI, wenn Sie mehrteilige Uploads anhalten und fortsetzen müssen, die Teilgrößen während des Uploads variieren müssen oder wenn Sie die Größe der Upload-Daten nicht im Voraus kennen. Wenn Sie diese Anforderungen nicht erfüllen, verwenden Sie den High-Level-Modus API (siehe). Verwenden von AWS SDKs (auf hoher EbeneAPI)

Java

Im folgenden Beispiel wird gezeigt, wie Sie mithilfe der Low-Level-Java-Klassen eine Datei hochladen. Es führt die folgenden Schritte aus:

- Startet einen mehrteiligen Upload mit der Methode `AmazonS3Client.initiateMultipartUpload()` und übergibt ein `InitiateMultipartUploadRequest`-Objekt.
- Speichern Sie die Upload-ID, die von der Methode `AmazonS3Client.initiateMultipartUpload()` zurückgegeben wird. Sie geben diese Upload-ID bei jeder nachfolgenden mehrteiligen Upload-Operation an.
- Lädt die Teile des Objekts hoch. Für jedes Teil rufen Sie die Methode `AmazonS3Client.uploadPart()` auf. Sie stellen die Informationen für das Hochladen von Teilen in einem `UploadPartRequest`-Objekt bereit.
- Speichert für jeden Teil die Antwort der `AmazonS3Client.uploadPart()` Methode in einer Liste. ETag Sie verwenden die ETag Werte, um den mehrteiligen Upload abzuschließen.
- Ruft die Methode `AmazonS3Client.completeMultipartUpload()` auf, um den mehrteiligen Upload fertigzustellen.

Example

Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class LowLevelMultipartUpload {
```

```
   public static void main(String[] args) throws IOException {
       Regions clientRegion = Regions.DEFAULT_REGION;
       String bucketName = "*** Bucket name ***";
       String keyName = "*** Key name ***";
       String filePath = "*** Path to file to upload ***";

       File file = new File(filePath);
       long contentLength = file.length();
       long partSize = 5 * 1024 * 1024; // Set part size to 5 MB.

       try {
           AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                   .withRegion(clientRegion)
                   .withCredentials(new ProfileCredentialsProvider())
                   .build();

           // Create a list of ETag objects. You retrieve ETags for each object
part
           // uploaded,
           // then, after each individual part has been uploaded, pass the list of
ETags to
           // the request to complete the upload.
           List<PartETag> partETags = new ArrayList<PartETag>();

           // Initiate the multipart upload.
           InitiateMultipartUploadRequest initRequest = new
InitiateMultipartUploadRequest(bucketName, keyName);
           InitiateMultipartUploadResult initResponse =
s3Client.initiateMultipartUpload(initRequest);

           // Upload the file parts.
           long filePosition = 0;
           for (int i = 1; filePosition < contentLength; i++) {
               // Because the last part could be less than 5 MB, adjust the part
size as
               // needed.
               partSize = Math.min(partSize, (contentLength - filePosition));

               // Create the request to upload a part.
               UploadPartRequest uploadRequest = new UploadPartRequest()
                       .withBucketName(bucketName)
                       .withKey(keyName)
                       .withUploadId(initResponse.getUploadId())
```

```
                        .withPartNumber(i)
                        .withFileOffset(filePosition)
                        .withFile(file)
                        .withPartSize(partSize);

                // Upload the part and add the response's ETag to our list.
                UploadPartResult uploadResult = s3Client.uploadPart(uploadRequest);
                partETags.add(uploadResult.getPartETag());

                filePosition += partSize;
            }

            // Complete the multipart upload.
            CompleteMultipartUploadRequest compRequest = new
 CompleteMultipartUploadRequest(bucketName, keyName,
                    initResponse.getUploadId(), partETags);
            s3Client.completeMultipartUpload(compRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Das folgende C#-Beispiel zeigt, wie Sie den AWS SDK for .NET mehrteiligen Low-Level-Upload verwenden, API um eine Datei in einen S3-Bucket hochzuladen. Weitere Informationen über mehrteilige Uploads zu Amazon S3 finden Sie unter Hochladen und Kopieren von Objekten mit mehrteiligen Uploads.

> **Note**
>
> Wenn Sie das AWS SDK for .NET API zum Hochladen großer Objekte verwenden, kann es beim Schreiben von Daten in den Anforderungsstream zu einem Timeout kommen. Sie können mit `UploadPartRequest` ein explizites Timeout festlegen.

Im folgenden C#-Beispiel wird eine Datei mithilfe des mehrteiligen Low-Level-Uploads in einen S3-Bucket hochgeladen. API Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem for. AWS SDK NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
using Amazon;
using Amazon.Runtime;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.IO;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadFileMPULowLevelAPITest
    {
        private const string bucketName = "*** provide bucket name ***";
        private const string keyName = "*** provide a name for the uploaded object
 ***";
        private const string filePath = "*** provide the full path name of the file
 to upload ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            Console.WriteLine("Uploading an object");
            UploadObjectAsync().Wait();
        }

        private static async Task UploadObjectAsync()
        {
            // Create list to store upload part responses.
            List<UploadPartResponse> uploadResponses = new
 List<UploadPartResponse>();

            // Setup information required to initiate the multipart upload.
```

```
            InitiateMultipartUploadRequest initiateRequest = new
InitiateMultipartUploadRequest
            {
                BucketName = bucketName,
                Key = keyName
            };

            // Initiate the upload.
            InitiateMultipartUploadResponse initResponse =
                await s3Client.InitiateMultipartUploadAsync(initiateRequest);

            // Upload parts.
            long contentLength = new FileInfo(filePath).Length;
            long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB

            try
            {
                Console.WriteLine("Uploading parts");

                long filePosition = 0;
                for (int i = 1; filePosition < contentLength; i++)
                {
                    UploadPartRequest uploadRequest = new UploadPartRequest
                        {
                            BucketName = bucketName,
                            Key = keyName,
                            UploadId = initResponse.UploadId,
                            PartNumber = i,
                            PartSize = partSize,
                            FilePosition = filePosition,
                            FilePath = filePath
                        };

                    // Track upload progress.
                    uploadRequest.StreamTransferProgress +=
                        new
EventHandler<StreamTransferProgressArgs>(UploadPartProgressEventCallback);

                    // Upload a part and add the response to our list.
                    uploadResponses.Add(await
s3Client.UploadPartAsync(uploadRequest));

                    filePosition += partSize;
                }
```

```
                // Setup to complete the upload.
                CompleteMultipartUploadRequest completeRequest = new
 CompleteMultipartUploadRequest
                    {
                        BucketName = bucketName,
                        Key = keyName,
                        UploadId = initResponse.UploadId
                     };
                completeRequest.AddPartETags(uploadResponses);

                // Complete the upload.
                CompleteMultipartUploadResponse completeUploadResponse =
                    await s3Client.CompleteMultipartUploadAsync(completeRequest);
            }
            catch (Exception exception)
            {
                Console.WriteLine("An AmazonS3Exception was thrown: { 0}",
 exception.Message);

                // Abort the upload.
                AbortMultipartUploadRequest abortMPURequest = new
 AbortMultipartUploadRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
                    UploadId = initResponse.UploadId
                };
               await s3Client.AbortMultipartUploadAsync(abortMPURequest);
            }
        }
        public static void UploadPartProgressEventCallback(object sender,
 StreamTransferProgressArgs e)
        {
            // Process event.
            Console.WriteLine("{0}/{1}", e.TransferredBytes, e.TotalBytes);
        }
    }
}
```

PHP

In diesem Thema wird gezeigt, wie Sie die `uploadPart` Low-Level-Methode aus Version 3 von verwenden AWS SDK for PHP , um eine Datei in mehreren Teilen hochzuladen. Weitere Informationen über die AWS SDK für Ruby API finden Sie unter AWS SDKfür Ruby — Version 2.

Das folgende PHP Beispiel lädt eine Datei mithilfe des PHP API mehrteiligen Low-Level-Uploads in einen Amazon S3 S3-Bucket hoch.

```
 require 'vendor/autoload.php';

use Aws\S3\Exception\S3Exception;
use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';
$filename = '*** Path to and Name of the File to Upload ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

$result = $s3->createMultipartUpload([
    'Bucket'       => $bucket,
    'Key'          => $keyname,
    'StorageClass' => 'REDUCED_REDUNDANCY',
    'Metadata'     => [
        'param1' => 'value 1',
        'param2' => 'value 2',
        'param3' => 'value 3'
    ]
]);
$uploadId = $result['UploadId'];

// Upload the file in parts.
try {
    $file = fopen($filename, 'r');
    $partNumber = 1;
    while (!feof($file)) {
        $result = $s3->uploadPart([
            'Bucket'     => $bucket,
            'Key'        => $keyname,
```

```
            'UploadId'   => $uploadId,
            'PartNumber' => $partNumber,
            'Body'       => fread($file, 5 * 1024 * 1024),
        ]);
        $parts['Parts'][$partNumber] = [
            'PartNumber' => $partNumber,
            'ETag' => $result['ETag'],
        ];
        $partNumber++;

        echo "Uploading part $partNumber of $filename." . PHP_EOL;
    }
    fclose($file);
} catch (S3Exception $e) {
    $result = $s3->abortMultipartUpload([
        'Bucket'   => $bucket,
        'Key'      => $keyname,
        'UploadId' => $uploadId
    ]);

    echo "Upload of $filename failed." . PHP_EOL;
}

// Complete the multipart upload.
$result = $s3->completeMultipartUpload([
    'Bucket'   => $bucket,
    'Key'      => $keyname,
    'UploadId' => $uploadId,
    'MultipartUpload'    => $parts,
]);
$url = $result['Location'];

echo "Uploaded $filename to $url." . PHP_EOL;
```

Unter Verwendung des AWS SDK for Ruby

Die AWS SDK for Ruby Version 3 unterstützt mehrteilige Amazon S3 S3-Uploads auf zwei Arten. Bei der ersten Option können Sie verwaltete Datei-Uploads verwenden. Weitere Informationen finden Sie unter Uploading Files to Amazon S3 im AWS -Entwickler-Blog. Verwaltete Datei-Uploads sind die empfohlene Methode zum Hochladen von Dateien zu einem Bucket. Sie bieten die folgenden Vorteile:

- Verwaltet mehrteilige Uploads von Objekten über 15 MB.
- Dateien werden im Binärmodus korrekt geöffnet, um Codierungsprobleme zu vermeiden.
- Verwendet mehrere Threads zum parallelen Hochladen von Teilen großer Objekte.

Alternativ können Sie die folgenden Multipart-Upload-Client-Vorgänge direkt verwenden:

- create_multipart_upload – Initiiert einen mehrteiligen Upload und gibt eine Upload-ID zurück.
- upload_part – Lädt einen Teil eines mehrteiligen Uploads hoch.
- upload_part_copy – Lädt einen Teil durch Kopieren von Daten aus einem vorhandenen Objekt hoch, das als Datenquelle dient.
- complete_multipart_upload – Schließt einen mehrteiligen Upload durch das Zusammenfügen zuvor hochgeladener Teile ab.
- abort_multipart_upload – Bricht einen mehrteiligen Upload ab.

Hochladen eines Verzeichnisses mithilfe der High-Level-Methode. NET TransferUtility Klasse

Mit der Klasse `TransferUtility` können Sie ein gesamtes Verzeichnis hochladen. Standardmäßig werden nur die Dateien im Stammverzeichnis des angegebenen Verzeichnisses API hochgeladen. Sie können jedoch festlegen, dass die Dateien in allen Unterverzeichnissen rekursiv hochgeladen werden sollen.

Um Dateien im angegebenen Verzeichnis basierend auf Filterkriterien anzugeben, geben Sie Filterausdrücke an. Wenn Sie z. B. nur die .pdf-Dateien aus einem Verzeichnis hochladen möchten, geben Sie den Filterausdruck `"*.pdf"` ein.

Wenn Sie Dateien aus einem Verzeichnis hochladen, geben Sie nicht die Schlüsselnamen der resultierenden Objekte an. Amazon S3 setzt die Schlüsselnamen aus dem Original-Dateipfad zusammen. Angenommen, ein Verzeichnis mit dem Namen `c:\myfolder` besitzt die folgende Verzeichnisstruktur:

Example

```
C:\myfolder
      \a.txt
      \b.pdf
      \media\
```

```
            An.mp3
```

Wenn Sie dieses Verzeichnis hochladen, verwendet Amazon S3 die folgenden Schlüsselnamen:

Example

```
a.txt
b.pdf
media/An.mp3
```

Example

Im folgenden C#-Beispiel wird ein Verzeichnis in einen Amazon-S3-Bucket hochgeladen. Es veranschaulicht, wie Sie zahlreiche `TransferUtility.UploadDirectory`-Überladungen zum Hochladen des Verzeichnisses verwenden. Jeder nachfolgende Aufruf zum Hochladen ersetzt den vorherigen Upload. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem AWS SDK for. NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.IO;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class UploadDirMPUHighLevelAPITest
    {
        private const string existingBucketName = "*** bucket name ***";
        private const string directoryPath = @"*** directory path ***";
        // The example uploads only .txt files.
        private const string wildCard = "*.txt";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion = RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            UploadDirAsync().Wait();
        }
```

```
        private static async Task UploadDirAsync()
        {
            try
            {
                var directoryTransferUtility =
                    new TransferUtility(s3Client);

                // 1. Upload a directory.
                await directoryTransferUtility.UploadDirectoryAsync(directoryPath,
                    existingBucketName);
                Console.WriteLine("Upload statement 1 completed");

                // 2. Upload only the .txt files from a directory
                //    and search recursively.
                await directoryTransferUtility.UploadDirectoryAsync(
                                                directoryPath,
                                                existingBucketName,
                                                wildCard,
                                                SearchOption.AllDirectories);
                Console.WriteLine("Upload statement 2 completed");

                // 3. The same as Step 2 and some optional configuration.
                //    Search recursively for .txt files to upload.
                var request = new TransferUtilityUploadDirectoryRequest
                {
                    BucketName = existingBucketName,
                    Directory = directoryPath,
                    SearchOption = SearchOption.AllDirectories,
                    SearchPattern = wildCard
                };

                await directoryTransferUtility.UploadDirectoryAsync(request);
                Console.WriteLine("Upload statement 3 completed");
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine(
                        "Error encountered ***. Message:'{0}' when writing an object",
e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine(
```

```
                    "Unknown encountered on server. Message:'{0}' when writing an
 object", e.Message);
                }
            }
        }
    }
```

Auflisten von mehrteiligen Uploads

Sie können, oder AWS CLI, verwenden REST API AWS SDKs, um eine Liste der laufenden mehrteiligen Uploads in Amazon S3 abzurufen. Sie können den Multipart-Upload verwenden, um ein einzelnes Objekt programmgesteuert auf Amazon S3 hochzuladen. Bei mehrteiligen Uploads werden Objekte in Amazon S3 verschoben, indem jeweils ein Teil der Daten eines Objekts verschoben wird. Weitere allgemeine Informationen zu mehrteiligen Uploads finden Sie unter. Hochladen und Kopieren von Objekten mit mehrteiligen Uploads

Ein end-to-end Verfahren zum Hochladen eines Objekts mit mehrteiligem Upload mit einer zusätzlichen Prüfsumme finden Sie unter. Tutorial: Laden Sie ein Objekt durch mehrteiligen Upload hoch und überprüfen Sie seine Datenintegrität

Im folgenden Abschnitt wird gezeigt, wie Sie in Bearbeitung befindliche mehrteilige Uploads mit dem AWS Command Line Interface, Amazon S3 REST API und auflisten. AWS SDKs

Auflisten mehrteiliger Uploads mit dem AWS CLI

In den folgenden Abschnitten werden die Vorgänge zum Auflisten mehrteiliger Uploads AWS Command Line Interface beschrieben.

- list-parts – listet die hochgeladenen Teile für einen bestimmten mehrteiligen Upload auf.
- list-multipart-uploads-listet in Bearbeitung befindliche mehrteilige Uploads auf.

Auflisten von mehrteiligen Uploads mit dem REST API

In den folgenden Abschnitten der Amazon Simple Storage Service API Reference wird beschrieben, wie Sie mehrteilige Uploads auflisten können: REST API

- ListParts-listet die hochgeladenen Teile für einen bestimmten mehrteiligen Upload auf.
- ListMultipartUploads-listet laufende mehrteilige Uploads auf.

Auflisten mehrteiliger Uploads mithilfe der (Low-Level) AWS SDK API

Java

Die folgenden Aufgaben führen Sie durch die Verwendung von Java Low-Level-Klassen, um alle laufenden mehrteiligen Uploads für einen Bucket aufzulisten.

Auflisten mehrteiliger Uploads auf niedriger Ebene API

1	Erstellen Sie eine Instance der `ListMultipartUploadsRequest` -Klasse und stellen Sie den Bucket-Namen bereit.
2	Führen Sie die `AmazonS3Client.listMultipartUploads` -Methode aus. Die Methode gibt eine Instance der Klasse `MultipartUploadListing` zurück, die Ihnen Informationen über die laufenden mehrteiligen Uploads bereitstellt.

Im folgenden Java-Codebeispiel werden die vorherigen Aufgaben veranschaulicht.

Example

```
ListMultipartUploadsRequest allMultpartUploadsRequest =
     new ListMultipartUploadsRequest(existingBucketName);
MultipartUploadListing multipartUploadListing =
     s3Client.listMultipartUploads(allMultpartUploadsRequest);
```

.NET

Um alle laufenden mehrteiligen Uploads in einem bestimmten Bucket aufzulisten, verwenden Sie die Klasse des mehrteiligen Uploads auf niedriger Ebene. AWS SDK for .NET API `ListMultipartUploadsRequest` Die Methode `AmazonS3Client.ListMultipartUploads` gibt eine Instance der Klasse `ListMultipartUploadsResponse` zurück, die Informationen über die laufenden mehrteiligen Uploads bereitstellt.

Ein laufender mehrteiliger Upload ist ein mehrteiliger Upload, der von der Anfrage für das Initiieren mehrteiliger Uploads initiiert, aber weder abgeschlossen noch abgebrochen wurde. Weitere Informationen über mehrteilige Amazon-S3-Uploads finden Sie unter Hochladen und Kopieren von Objekten mit mehrteiligen Uploads.

Das folgende C#-Beispiel zeigt, wie Sie mithilfe von alle laufenden mehrteiligen Uploads in einem AWS SDK for .NET Bucket auflisten können. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem for. AWS SDK NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
ListMultipartUploadsRequest request = new ListMultipartUploadsRequest
{
  BucketName = bucketName // Bucket receiving the uploads.
};

ListMultipartUploadsResponse response = await
 AmazonS3Client.ListMultipartUploadsAsync(request);
```

PHP

In diesem Thema wird gezeigt, wie Sie die API Low-Level-Klassen aus Version 3 von verwenden, AWS SDK for PHP um alle laufenden mehrteiligen Uploads in einem Bucket aufzulisten. Weitere Informationen zu for Ruby finden Sie unter AWS SDK for API Ruby —AWS SDK Version 2.

Das folgende PHP Beispiel zeigt die Auflistung aller laufenden mehrteiligen Uploads in einem Bucket.

```
 require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Retrieve a list of the current multipart uploads.
$result = $s3->listMultipartUploads([
    'Bucket' => $bucket
]);

// Write the list of uploads to the page.
print_r($result->toArray());
```

Verfolgen eines mehrteiligen Uploads mit dem AWS SDKs

Sie können den Upload-Fortschritt eines Objekts auf Amazon S3 mit einer Listen-Schnittstelle verfolgen. Der mehrteilige Upload auf hoher Ebene API bietet eine solche Listen-Schnittstelle namens`ProgressListener`. Fortschrittsereignisse treten periodisch auf und benachrichtigen den Listener, dass Bytes übertragen wurden. Weitere allgemeine Informationen zu mehrteiligen Uploads finden Sie unter. Hochladen und Kopieren von Objekten mit mehrteiligen Uploads

Ein end-to-end Verfahren zum Hochladen eines Objekts mit mehrteiligem Upload mit einer zusätzlichen Prüfsumme finden Sie unter. Tutorial: Laden Sie ein Objekt durch mehrteiligen Upload hoch und überprüfen Sie seine Datenintegrität

Der folgende Abschnitt zeigt, wie Sie einen mehrteiligen Upload mit dem verfolgen können. AWS SDKs

Java

Example

```
TransferManager tm = new TransferManager(new ProfileCredentialsProvider());

PutObjectRequest request = new PutObjectRequest(
    existingBucketName, keyName, new File(filePath));

// Subscribe to the event and provide event handler.
request.setProgressListener(new ProgressListener() {
   public void progressChanged(ProgressEvent event) {
    System.out.println("Transferred bytes: " +
      event.getBytesTransfered());
            }
});
```

Example

Der folgende Java-Code lädt eine Datei hoch und verwendet den `ProgressListener`, um den Upload-Fortschritt zu verfolgen. Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
import java.io.File;

import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
```

```
import com.amazonaws.event.ProgressEvent;
import com.amazonaws.event.ProgressListener;
import com.amazonaws.services.s3.model.PutObjectRequest;
import com.amazonaws.services.s3.transfer.TransferManager;
import com.amazonaws.services.s3.transfer.Upload;

public class TrackMPUProgressUsingHighLevelAPI {

    public static void main(String[] args) throws Exception {
        String existingBucketName = "*** Provide bucket name ***";
        String keyName            = "*** Provide object key ***";
        String filePath           = "*** file to upload ***";

        TransferManager tm = new TransferManager(new ProfileCredentialsProvider());


        // For more advanced uploads, you can create a request object
        // and supply additional request parameters (ex: progress listeners,
        // canned ACLs, etc.)
        PutObjectRequest request = new PutObjectRequest(
          existingBucketName, keyName, new File(filePath));

        // You can ask the upload for its progress, or you can
        // add a ProgressListener to your request to receive notifications
        // when bytes are transferred.
        request.setGeneralProgressListener(new ProgressListener() {
   @Override
   public void progressChanged(ProgressEvent progressEvent) {
    System.out.println("Transferred bytes: " +
      progressEvent.getBytesTransferred());
   }
  });

        // TransferManager processes all transfers asynchronously,
        // so this call will return immediately.
        Upload upload = tm.upload(request);

        try {
         // You can block and wait for the upload to finish
         upload.waitForCompletion();
        } catch (AmazonClientException amazonClientException) {
         System.out.println("Unable to upload file, upload aborted.");
         amazonClientException.printStackTrace();
        }
```

```
    }
}
```

.NET

Das folgende C#-Beispiel lädt eine Datei unter Verwendung der Klasse `TransferUtility` in einen S3-Bucket hoch und verfolgt den Fortschritt des Uploads. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem AWS SDK for. NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class TrackMPUUsingHighLevelAPITest
    {
        private const string bucketName = "*** provide the bucket name ***";
        private const string keyName = "*** provide the name for the uploaded object
 ***";
        private const string filePath = " *** provide the full path name of the file
 to upload **";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;


        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            TrackMPUAsync().Wait();
        }

        private static async Task TrackMPUAsync()
        {
            try
            {
                var fileTransferUtility = new TransferUtility(s3Client);
```

```
                // Use TransferUtilityUploadRequest to configure options.
                // In this example we subscribe to an event.
                var uploadRequest =
                    new TransferUtilityUploadRequest
                    {
                        BucketName = bucketName,
                        FilePath = filePath,
                        Key = keyName
                    };

                uploadRequest.UploadProgressEvent +=
                    new EventHandler<UploadProgressArgs>
                        (uploadRequest_UploadPartProgressEvent);

                await fileTransferUtility.UploadAsync(uploadRequest);
                Console.WriteLine("Upload completed");
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }

        static void uploadRequest_UploadPartProgressEvent(object sender,
 UploadProgressArgs e)
        {
            // Process event.
            Console.WriteLine("{0}/{1}", e.TransferredBytes, e.TotalBytes);
        }
    }
}
```

Abbrechen eines mehrteiligen Uploads

Nachdem Sie einen mehrteiligen Upload initiiert haben, beginnen Sie mit dem Hochladen von Teilen. Amazon S3 speichert diese Teile und erstellt das Objekt erst, nachdem Sie alle Teile hochgeladen

und eine Anfrage zum Abschluss des mehrteiligen Uploads gesendet haben. Nach dem Empfang der Anfrage auf Abschluss des Multipart-Uploads erzeugt Amazon S3 aus den Teilen ein Objekt. Wenn Sie die vollständige mehrteilige Upload-Anfrage nicht erfolgreich senden, baut S3 die Teile nicht zusammen und erstellt auch kein Objekt. Wenn Sie einen mehrteiligen Upload nach dem Hochladen von Teilen nicht abschließen möchten, sollten Sie den mehrteiligen Upload abbrechen.

Ihnen wird der gesamte Speicher in Rechnung gestellt, der mit hochgeladenen Teilen verknüpft ist. Es wird empfohlen, entweder den mehrteiligen Upload immer abzuschließen oder den mehrteiligen Upload zu beenden, um alle hochgeladenen Teile zu entfernen. Weitere Informationen zu Preisen erhalten Sie unter Mehrteiliger Upload und Preise.

Sie können einen unvollständigen mehrteiligen Upload auch mithilfe einer Bucket-Lebenszykluskonfiguration beenden. Weitere Informationen finden Sie unter Konfigurieren einer Bucket-Lebenszykluskonfiguration zum Löschen unvollständiger mehrteiliger Uploads.

Der folgende Abschnitt zeigt, wie Sie einen laufenden mehrteiligen Upload in Amazon S3 mithilfe von AWS Command Line Interface RESTAPI, oder beenden können. AWS SDKs

Mit dem AWS CLI

Weitere Informationen zur Verwendung von AWS CLI zum Beenden eines mehrteiligen Uploads finden Sie abort-multipart-uploadin der AWS CLI Befehlsreferenz.

Verwenden von REST API

Weitere Informationen zur Verwendung von REST API zum Stoppen eines mehrteiligen Uploads finden Sie AbortMultipartUploadin der Amazon Simple Storage Service API Reference.

Verwenden von AWS SDKs (auf hoher EbeneAPI)

Java

Die Klasse `TransferManager` bietet die Methode `abortMultipartUploads`, um laufende Multipart-Uploads zu stoppen. Ein Upload wird als laufend betrachtet, nachdem Sie ihn initiiert haben und bis er abgeschlossen ist oder Sie ihn abbrechen. Sie geben einen `Date` Wert an, und dadurch werden alle mehrteiligen Uploads für diesen Bucket API gestoppt, die vor dem angegebenen Wert initiiert wurden `Date` und noch im Gange sind.

Die folgenden Aufgaben führen Sie durch die Verwendung von High-Level Java-Klassen für das Abbrechen mehrteiliger Uploads.

Prozess zum Stoppen API mehrteiliger Uploads auf hoher Ebene

1	Erstellen Sie eine Instance der `TransferManager` -Klasse.
2	Führen Sie die `TransferManager.abortMultipartUploads` -Methode aus, indem Sie den Bucket-Namen und einen `Date`-Wert übergeben.

Der folgende Java-Code bricht alle laufenden mehrteiligen Uploads ab, die vor mehr als einer Woche für einen bestimmten Bucket initiiert wurden. Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
import java.util.Date;

import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3.transfer.TransferManager;

public class AbortMPUUsingHighLevelAPI {

    public static void main(String[] args) throws Exception {
        String existingBucketName = "*** Provide existing bucket name ***";

        TransferManager tm = new TransferManager(new ProfileCredentialsProvider());


        int sevenDays = 1000 * 60 * 60 * 24 * 7;
  Date oneWeekAgo = new Date(System.currentTimeMillis() - sevenDays);

        try {
         tm.abortMultipartUploads(existingBucketName, oneWeekAgo);
        } catch (AmazonClientException amazonClientException) {
         System.out.println("Unable to upload file, upload was aborted.");
         amazonClientException.printStackTrace();
        }
    }
}
```

> **Note**
>
> Sie können auch einen bestimmten mehrteiligen Upload abbrechen. Weitere Informationen finden Sie unter Verwenden des AWS SDKs (Low-LevelAPI).

.NET

Das folgende C#-Bespiel bricht alle laufenden mehrteiligen Uploads ab, die vor mehr als einer Woche für einen bestimmten Bucket initiiert wurden. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem AWS SDK for. NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Transfer;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class AbortMPUUsingHighLevelAPITest
    {
        private const string bucketName = "*** provide bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            AbortMPUAsync().Wait();
        }

        private static async Task AbortMPUAsync()
        {
            try
            {
                var transferUtility = new TransferUtility(s3Client);
```

```
                // Abort all in-progress uploads initiated before the specified
 date.
                await transferUtility.AbortMultipartUploadsAsync(
                    bucketName, DateTime.Now.AddDays(-7));
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
    }
}
```

> **Note**
>
> Sie können auch einen bestimmten mehrteiligen Upload abbrechen. Weitere Informationen finden Sie unter Verwenden des AWS SDKs (Low-LevelAPI).

Verwenden des AWS SDKs (Low-LevelAPI)

Durch Aufruf der Methode `AmazonS3.abortMultipartUpload` können Sie einen mehrteiligen Upload in Bearbeitung abbrechen. Diese Methode löscht alle Teile, die in Amazon S3 hochgeladen wurden, und gibt die Ressourcen frei. Sie müssen die Upload-ID, den Bucket-Namen und die Schlüsselnamen bereitstellen. Das folgende Java-Codebeispiel zeigt, wie ein laufender mehrteiliger Upload abgebrochen wird.

Um einen mehrteiligen Upload abzubrechen, stellen Sie die Upload-ID bereit sowie die Bucket- und Schlüsselnamen, die im Upload verwendet werden. Nachdem ein mehrteiliger Upload abgebrochen wurde, können Sie mit dieser Upload-ID keinen zusätzlichen Teile mehr hochladen. Weitere Informationen über mehrteilige Amazon-S3-Uploads finden Sie unter Hochladen und Kopieren von Objekten mit mehrteiligen Uploads.

Java

Im folgenden Java-Codebeispiel wird ein laufender mehrteiliger Upload gestoppt.

Example

```
InitiateMultipartUploadRequest initRequest =
    new InitiateMultipartUploadRequest(existingBucketName, keyName);
InitiateMultipartUploadResult initResponse =
               s3Client.initiateMultipartUpload(initRequest);

AmazonS3 s3Client = new AmazonS3Client(new ProfileCredentialsProvider());
s3Client.abortMultipartUpload(new AbortMultipartUploadRequest(
            existingBucketName, keyName, initResponse.getUploadId()));
```

> **Note**
>
> Statt einen spezifischen mehrteiligen Upload können Sie auch alle Ihre mehrteiligen Uploads abbrechen, die vor einem bestimmten Zeitpunkt initiiert wurden und immer noch laufen. Diese Bereinigungsoperation ist praktisch, um alte mehrteilige Uploads abzubrechen, die Sie initiiert haben, aber die nicht abgeschlossen oder abgebrochen wurden. Weitere Informationen finden Sie unter Verwenden von AWS SDKs (auf hoher EbeneAPI).

.NET

Das folgende C#-Beispiel veranschaulicht, wie Sie einen mehrteiligen Upload abbrechen. Ein vollständiges C#-Beispiel mit dem folgenden Code finden Sie unter Verwenden von AWS SDKs (APILow-Level).

```
AbortMultipartUploadRequest abortMPURequest = new AbortMultipartUploadRequest
{
    BucketName = existingBucketName,
    Key = keyName,
    UploadId = initResponse.UploadId
};
await AmazonS3Client.AbortMultipartUploadAsync(abortMPURequest);
```

Sie können auch alle laufenden mehrteiligen Uploads abbrechen, die vor einem bestimmten Zeitpunkt initiiert wurden. Diese Bereinigungsoperation ist praktisch, um mehrteilige Uploads

abzubrechen, die nicht abgeschlossen oder abgebrochen wurden. Weitere Informationen finden Sie unter Verwenden von AWS SDKs (auf hoher EbeneAPI).

PHP

Dieses Beispiel zeigt, wie Sie eine Klasse aus Version 3 des AWS SDK for PHP für PHP verwenden, um einen laufenden mehrteiligen Upload abzubrechen. Weitere Informationen zu AWS SDK for Ruby API finden Sie unter AWS SDKfor Ruby — Version 2. Das Beispiel der Methode `abortMultipartUpload()`.

Weitere Informationen über die AWS SDK für Ruby API finden Sie unter AWS SDKfür Ruby — Version 2.

```
 require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';
$uploadId = '*** Upload ID of upload to Abort ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Abort the multipart upload.
$s3->abortMultipartUpload([
    'Bucket'   => $bucket,
    'Key'      => $keyname,
    'UploadId' => $uploadId,
]);
```

Kopieren eines Objekts mit Multipart-Upload

Mit dem mehrteiligen Upload können Sie Objekte als Gruppe von Teilen kopieren. Die Beispiele in diesem Abschnitt zeigen Ihnen, wie Sie Objekte mit mehr als 5 GB mithilfe des mehrteiligen Uploads kopieren können. API Informationen zu mehrteiligen Uploads finden Sie unter. Hochladen und Kopieren von Objekten mit mehrteiligen Uploads

Sie können Objekte mit weniger als 5 GB in einem einzigen Vorgang kopieren, ohne den mehrteiligen Upload zu verwenden. API Sie können Objekte mit weniger als 5 GB mit dem AWS Management Console, AWS CLI RESTAPI, oder AWS SDKs kopieren. Weitere Informationen finden Sie unter Objekte kopieren, verschieben und umbenennen.

Eine Anleitung zum end-to-end Hochladen eines Objekts mit mehrteiligem Upload mit zusätzlicher Prüfsumme finden Sie unter. Tutorial: Laden Sie ein Objekt durch mehrteiligen Upload hoch und überprüfen Sie seine Datenintegrität

Im folgenden Abschnitt wird gezeigt, wie Sie ein Objekt beim mehrteiligen Upload mit dem oder kopieren. REST API AWS SDKs

Mit dem REST API

In den folgenden Abschnitten der Amazon Simple Storage Service API Reference wird das REST API für mehrteilige Uploads beschrieben. Um ein vorhandenes Objekt zu kopieren, verwenden Sie den Upload Part (Copy) API und geben Sie das Quellobjekt an, indem Sie Ihrer `x-amz-copy-source` Anfrage den Anforderungsheader hinzufügen.

- Initiieren eines mehrteiligen Uploads
- Upload Part
- Teil hochladen (kopieren)
- Abschließen eines mehrteiligen Uploads
- Abort Multipart Upload
- Teile auflisten
- List Multipart Uploads

Sie können diese verwendenAPIs, um Ihre eigenen REST Anfragen zu stellen, oder Sie können eine der von SDKs uns bereitgestellten verwenden. Weitere Informationen zur Verwendung von Multipart Upload mit dem finden Sie AWS CLI unterMit dem AWS CLI. Weitere Informationen zu dem finden Sie SDKs unterAWS SDK-Unterstützung für mehrteilige Uploads.

Mit dem AWS SDKs

Gehen Sie wie folgt vor, um ein Objekt mithilfe der API Low-Level-Methode zu kopieren:

- Initiieren eines mehrteiligen Uploads durch Aufrufen der Methode `AmazonS3Client.initiateMultipartUpload()`.

- Speichern Sie die Upload-ID aus dem Antwortobjekt, das die Methode `AmazonS3Client.initiateMultipartUpload()` zurückgibt. Sie geben diese Upload-ID bei jeder Teiloperation mehrteiliger Uploads an.
- Kopieren Sie alle Teile. Erstellen Sie für jeden Teil, den Sie kopieren müssen, eine neue Instance der Klasse `CopyPartRequest`. Geben Sie die teilspezifischen Informationen an, darunter Quell- und Zielbucket-Namen, Quell- und Ziel-Objektschlüssel, Upload-ID, Stelle des ersten und des letzten Byte des Teils sowie die Nummer des Teils.
- Speichern Sie die Antworten der `AmazonS3Client.copyPart()`-Methodenaufrufe. Jede Antwort enthält den Wert für `ETag` und die Teilenummer des hochgeladenen Teils. Sie benötigen diese Informationen, um den mehrteiligen Upload abzuschließen.
- Rufen Sie die Methode `AmazonS3Client.completeMultipartUpload()` auf, um die Kopieroperation abzuschließen.

Java

Example

Das folgende Beispiel zeigt, wie Sie Amazon S3 Low-Level-Java verwendenAPI, um eine mehrteilige Kopie durchzuführen. Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class LowLevelMultipartCopy {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String sourceBucketName = "*** Source bucket name ***";
        String sourceObjectKey = "*** Source object key ***";
```

```
        String destBucketName = "*** Target bucket name ***";
        String destObjectKey = "*** Target object key ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Initiate the multipart upload.
            InitiateMultipartUploadRequest initRequest = new
InitiateMultipartUploadRequest(destBucketName,
                    destObjectKey);
            InitiateMultipartUploadResult initResult =
s3Client.initiateMultipartUpload(initRequest);

            // Get the object size to track the end of the copy operation.
            GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest(sourceBucketName, sourceObjectKey);
            ObjectMetadata metadataResult =
s3Client.getObjectMetadata(metadataRequest);
            long objectSize = metadataResult.getContentLength();

            // Copy the object using 5 MB parts.
            long partSize = 5 * 1024 * 1024;
            long bytePosition = 0;
            int partNum = 1;
            List<CopyPartResult> copyResponses = new ArrayList<CopyPartResult>();
            while (bytePosition < objectSize) {
                // The last part might be smaller than partSize, so check to make
sure
                // that lastByte isn't beyond the end of the object.
                long lastByte = Math.min(bytePosition + partSize - 1, objectSize -
1);

                // Copy this part.
                CopyPartRequest copyRequest = new CopyPartRequest()
                        .withSourceBucketName(sourceBucketName)
                        .withSourceKey(sourceObjectKey)
                        .withDestinationBucketName(destBucketName)
                        .withDestinationKey(destObjectKey)
                        .withUploadId(initResult.getUploadId())
                        .withFirstByte(bytePosition)
                        .withLastByte(lastByte)
```

```
                        .withPartNumber(partNum++);
                copyResponses.add(s3Client.copyPart(copyRequest));
                bytePosition += partSize;
            }

            // Complete the upload request to concatenate all uploaded parts and
 make the
            // copied object available.
            CompleteMultipartUploadRequest completeRequest = new
 CompleteMultipartUploadRequest(
                    destBucketName,
                    destObjectKey,
                    initResult.getUploadId(),
                    getETags(copyResponses));
            s3Client.completeMultipartUpload(completeRequest);
            System.out.println("Multipart copy complete.");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }

    // This is a helper function to construct a list of ETags.
    private static List<PartETag> getETags(List<CopyPartResult> responses) {
        List<PartETag> etags = new ArrayList<PartETag>();
        for (CopyPartResult response : responses) {
            etags.add(new PartETag(response.getPartNumber(), response.getETag()));
        }
        return etags;
    }
}
```

.NET

Das folgende C#-Beispiel zeigt, wie Sie mithilfe von ein Amazon S3 S3-Objekt, das größer als 5 GB ist, von einem Quellspeicherort zu einem anderen kopieren, z. B. von einem Bucket in einen anderen. AWS SDK for .NET Um Objekte zu kopieren, die kleiner als 5 GB sind, verwenden Sie

das Kopierverfahren in einer einzigen Operation, das unter Unter Verwendung der AWS SDKs beschrieben wird. Weitere Informationen über mehrteilige Amazon-S3-Uploads finden Sie unter Hochladen und Kopieren von Objekten mit mehrteiligen Uploads.

Dieses Beispiel zeigt, wie ein Amazon S3 S3-Objekt, das größer als 5 GB ist, mithilfe des AWS SDK for .NET mehrteiligen Uploads API von einem S3-Bucket in einen anderen kopiert wird.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CopyObjectUsingMPUapiTest
    {
        private const string sourceBucket = "*** provide the name of the bucket with
 source object ***";
        private const string targetBucket = "*** provide the name of the bucket to
 copy the object to ***";
        private const string sourceObjectKey = "*** provide the name of object to
 copy ***";
        private const string targetObjectKey = "*** provide the name of the object
 copy ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            Console.WriteLine("Copying an object");
            MPUCopyObjectAsync().Wait();
        }
        private static async Task MPUCopyObjectAsync()
        {
            // Create a list to store the upload part responses.
            List<UploadPartResponse> uploadResponses = new
 List<UploadPartResponse>();
            List<CopyPartResponse> copyResponses = new List<CopyPartResponse>();
```

```
            // Setup information required to initiate the multipart upload.
            InitiateMultipartUploadRequest initiateRequest =
                new InitiateMultipartUploadRequest
                {
                    BucketName = targetBucket,
                    Key = targetObjectKey
                };

            // Initiate the upload.
            InitiateMultipartUploadResponse initResponse =
                await s3Client.InitiateMultipartUploadAsync(initiateRequest);

            // Save the upload ID.
            String uploadId = initResponse.UploadId;

            try
            {
                // Get the size of the object.
                GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest
                {
                    BucketName = sourceBucket,
                    Key = sourceObjectKey
                };

                GetObjectMetadataResponse metadataResponse =
                    await s3Client.GetObjectMetadataAsync(metadataRequest);
                long objectSize = metadataResponse.ContentLength; // Length in
bytes.

                // Copy the parts.
                long partSize = 5 * (long)Math.Pow(2, 20); // Part size is 5 MB.

                long bytePosition = 0;
                for (int i = 1; bytePosition < objectSize; i++)
                {
                    CopyPartRequest copyRequest = new CopyPartRequest
                    {
                        DestinationBucket = targetBucket,
                        DestinationKey = targetObjectKey,
                        SourceBucket = sourceBucket,
                        SourceKey = sourceObjectKey,
                        UploadId = uploadId,
```

```
                        FirstByte = bytePosition,
                        LastByte = bytePosition + partSize - 1 >= objectSize ?
 objectSize - 1 : bytePosition + partSize - 1,
                        PartNumber = i
                    };

                    copyResponses.Add(await s3Client.CopyPartAsync(copyRequest));

                    bytePosition += partSize;
                }

                // Set up to complete the copy.
                CompleteMultipartUploadRequest completeRequest =
                new CompleteMultipartUploadRequest
                {
                    BucketName = targetBucket,
                    Key = targetObjectKey,
                    UploadId = initResponse.UploadId
                };
                completeRequest.AddPartETags(copyResponses);

                // Complete the copy.
                CompleteMultipartUploadResponse completeUploadResponse =
                    await s3Client.CompleteMultipartUploadAsync(completeRequest);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
    }
}
```

Tutorial: Laden Sie ein Objekt durch mehrteiligen Upload hoch und überprüfen Sie seine Datenintegrität

Mit dem mehrteiligen Upload können Sie ein einzelnes Objekt als Satz aus mehreren Teilen hochladen. Jeder Teil ist ein zusammenhängender Teil der Daten des Objekts. Sie können diese Objektteile unabhängig und in beliebiger Reihenfolge hochladen. Wenn die Übertragung eines Teils fehlschlägt, können Sie das Teil erneut übertragen, ohne dass dies Auswirkungen auf andere Teile hat. Nachdem alle Teile Ihres Objekts hochgeladen sind, fügt Amazon S3 diese Teile zusammen und erstellt das Objekt. Wenn Ihre Objektgröße 100 MB erreicht, sollten Sie in der Regel mehrteilige Uploads verwenden, anstatt das Objekt in einem einzigen Vorgang hochzuladen. Weitere Informationen zu mehrteiligen Uploads finden Sie unter. Hochladen und Kopieren von Objekten mit mehrteiligen Uploads Einschränkungen im Zusammenhang mit mehrteiligen Uploads finden Sie unter. EInschränkungen mehrteiliger Amazon-S3-Uploads

Sie können Prüfsummen verwenden, um sicherzustellen, dass Assets beim Kopieren nicht verändert werden. Das Durchführen einer Prüfsumme besteht darin, mithilfe eines Algorithmus sequentiell über jedes Byte in einer Datei zu iterieren. Amazon S3 bietet mehrere Prüfsummenoptionen zur Überprüfung der Integrität von Daten. Wir empfehlen Ihnen, diese Integritätsprüfungen als bewährte Methode für die Haltbarkeit durchzuführen und um sicherzustellen, dass jedes Byte unverändert übertragen wird. Amazon S3 unterstützt auch die folgenden Algorithmen: SHA -1, SHA -256CRC32, undCRC32C. Amazon S3 verwendet einen oder mehrere dieser Algorithmen, um einen zusätzlichen Prüfsummenwert zu berechnen und ihn als Teil der Objektmetadaten zu speichern. Weitere Informationen zu Prüfsummen finden Sie unter Überprüfung der Objektintegrität.

Ziel

In diesem Tutorial erfahren Sie, wie Sie ein Objekt mithilfe eines mehrteiligen Uploads und einer zusätzlichen SHA -256-Prüfsumme über die AWS Befehlszeilenschnittstelle () auf Amazon S3 hochladen.AWS CLI Sie erfahren auch, wie Sie die Datenintegrität des Objekts überprüfen, indem Sie den MD5 Hash und die SHA -256-Prüfsumme des hochgeladenen Objekts berechnen.

Themen

- Voraussetzungen
- Schritt 1: Erstellen Sie eine große Datei
- Schritt 2: Teilen Sie die Datei in mehrere Dateien auf
- Schritt 3: Erstellen Sie den mehrteiligen Upload mit einer zusätzlichen Prüfsumme
- Schritt 4: Laden Sie die Teile Ihres mehrteiligen Uploads hoch

- Schritt 5: Listet alle Teile eures mehrteiligen Uploads auf
- Schritt 6: Schließen Sie den mehrteiligen Upload ab
- Schritt 7: Vergewissern Sie sich, dass das Objekt in Ihren Bucket hochgeladen wurde
- Schritt 8: Überprüfen Sie die Objektintegrität mit einer MD5 Prüfsumme
- Schritt 9: Überprüfen Sie die Objektintegrität mit einer zusätzlichen Prüfsumme
- Schritt 10: Bereinigen Ihrer Ressourcen

Voraussetzungen

- Bevor Sie mit diesem Tutorial beginnen, stellen Sie sicher, dass Sie Zugriff auf einen Amazon S3 S3-Bucket haben, in den Sie hochladen können. Weitere Informationen finden Sie unter Erstellen eines Buckets.
- Sie müssen das AWS CLI installiert und konfiguriert haben. Wenn Sie das nicht AWS CLI installiert haben, finden Sie im AWS Command Line Interface Benutzerhandbuch weitere Informationen unter Installieren oder Aktualisieren AWS CLI auf die neueste Version von.
- Alternativ können Sie AWS CLI Befehle von der Konsole aus ausführen, indem Sie AWS CloudShell AWS CloudShell ist eine browserbasierte, vorab authentifizierte Shell, die Sie direkt von der aus starten können. AWS Management ConsoleWeitere Informationen finden Sie unter Was ist? CloudShell und Erste Schritte mit AWS CloudShell im AWS CloudShell Benutzerhandbuch.

Schritt 1: Erstellen Sie eine große Datei

Wenn Sie bereits eine Datei zum Hochladen bereit haben, können Sie die Datei für dieses Tutorial verwenden. Andernfalls erstellen Sie mit den folgenden Schritten eine 15-MB-Datei. Informationen zu Beschränkungen im Zusammenhang mit mehrteiligen Uploads finden Sie unter. EInschränkungen mehrteiliger Amazon-S3-Uploads

Um eine große Datei zu erstellen

Verwenden Sie einen der folgenden Befehle, um Ihre Datei zu erstellen, je nachdem, welches Betriebssystem Sie verwenden.

Linux oder macOS

Um eine 15-MB-Datei zu erstellen, öffnen Sie Ihr lokales Terminal und führen Sie den folgenden Befehl aus:

```
dd if=/dev/urandom of=census-data.bin bs=1M count=15
```

Dieser Befehl erstellt eine mit zufälligen Bytes `census-data.bin` gefüllte Datei mit einer Größe von 15 MB.

Windows

Um eine 15-MB-Datei zu erstellen, öffnen Sie Ihr lokales Terminal und führen Sie den folgenden Befehl aus:

```
fsutil file createnew census-data.bin 15728640
```

Dieser Befehl erstellt eine Datei mit einem Namen `census-data.bin` mit einer Größe von 15 MB an beliebigen Daten (15728640 Byte).

Schritt 2: Teilen Sie die Datei in mehrere Dateien auf

Um den mehrteiligen Upload durchzuführen, müssen Sie Ihre große Datei in kleinere Teile aufteilen. Anschließend können Sie die kleineren Teile mithilfe des mehrteiligen Upload-Prozesses hochladen. In diesem Schritt wird gezeigt, wie Sie die in Schritt 1 erstellte große Datei in kleinere Teile aufteilen. Im folgenden Beispiel wird eine 15-MB-Datei mit dem Namen verwendet`census-data.bin`.

Um eine große Datei in Teile aufzuteilen

Linux oder macOS

Verwenden Sie den `split` Befehl, um die große Datei in 5 MB-Teile zu unterteilen. Öffnen Sie Ihr Terminal und führen Sie Folgendes aus:

```
split -b 5M -d census-data.bin census-part
```

Dieser Befehl teilt sich `census-data.bin` in 5 MB große Teile mit dem Namen auf`census-part**`, wobei `**` ein numerisches Suffix beginnt. `00`

Windows

Um die große Datei aufzuteilen, verwenden Sie. PowerShell Öffnen Sie Powershell und führen Sie das folgende Skript aus:

```
$inputFile = "census-data.bin"
$outputFilePrefix = "census-part"
$chunkSize = 5MB

$fs = [System.IO.File]::OpenRead($inputFile)
$buffer = New-Object byte[] $chunkSize
$fileNumber = 0

while ($fs.Position -lt $fs.Length) {
$bytesRead = $fs.Read($buffer, 0, $chunkSize)
$outputFile = "{0}{1:D2}" -f $outputFilePrefix, $fileNumber
$fileStream = [System.IO.File]::Create($outputFile)
$fileStream.Write($buffer, 0, $bytesRead)
$fileStream.Close()
$fileNumber++
}

$fs.Close()
```

Dieses PowerShell Skript liest die große Datei in Blöcken von 5 MB und schreibt jeden Abschnitt in eine neue Datei mit einem numerischen Suffix.

Nachdem Sie den entsprechenden Befehl ausgeführt haben, sollten Sie die Teile in dem Verzeichnis sehen, in dem Sie den Befehl ausgeführt haben. Jeder Teil hat ein Suffix, das seiner Artikelnummer entspricht, zum Beispiel:

```
census-part00 census-part01 census-part02
```

Schritt 3: Erstellen Sie den mehrteiligen Upload mit einer zusätzlichen Prüfsumme

Um den mehrteiligen Upload-Vorgang zu starten, müssen Sie die mehrteilige Upload-Anfrage erstellen. In diesem Schritt wird der mehrteilige Upload initiiert und eine zusätzliche Prüfsumme für die Datenintegrität angegeben. Im folgenden Beispiel wird die Prüfsumme -256 verwendet. SHA Wenn Sie Metadaten angeben möchten, die das hochgeladene Objekt beschreiben, müssen Sie diese in der Anforderung angeben, um den mehrteiligen Upload zu initiieren.

> **Note**
>
> In diesem Schritt und den nachfolgenden Schritten verwendet dieses Tutorial den zusätzlichen Algorithmus SHA -256. Sie können optional eine weitere zusätzliche Prüfsumme

für diese Schritte verwenden, z. B. CRC32CRC32C, oder SHA -1. Wenn Sie einen anderen Algorithmus verwenden, müssen Sie ihn in allen Schritten des Tutorials verwenden.

Um den mehrteiligen Upload zu starten

Verwenden Sie in Ihrem Terminal den folgenden `create-multipart-upload` Befehl, um einen mehrteiligen Upload für Ihren Bucket zu starten. Ersetzen Sie es *amzn-s3-demo-bucket1* durch Ihren tatsächlichen Bucket-Namen. Ersetzen Sie außerdem den `census_data_file` durch den von Ihnen gewählten Dateinamen. Dieser Dateiname wird zum Objektschlüssel, wenn der Upload abgeschlossen ist.

```
aws s3api create-multipart-upload --bucket amzn-s3-demo-bucket1 --key
 'census_data_file' --checksum-algorithm sha256
```

Wenn Ihre Anfrage erfolgreich ist, erhalten Sie eine JSON Ausgabe wie die folgende:

```
{
    "ServerSideEncryption": "AES256",
    "ChecksumAlgorithm": "SHA256",
    "Bucket": "amzn-s3-demo-bucket1",
    "Key": "census_data_file",
    "UploadId":
 "cNV6KCSNANFZapz1LUGPC5XwUVi1n6yUoIeSP138sNOKPeMhpKQRrbT9k0ePmgoOTCj9K83T4e2Gb5hQvNoNpCKqyb8m3
}
```

Note

Wenn Sie eine Anfrage zum Initiieren eines mehrteiligen Uploads senden, gibt Amazon S3 eine Antwort mit einer Upload-ID zurück, als eindeutige Kennung für den Multipart-Upload. Sie müssen diese Upload-ID immer angeben, wenn Sie Teile hochladen, die Teile auflisten, einen Upload abschließen oder einen Upload abbrechen. Sie müssen die `Bucket` Werte`UploadId`, und für spätere Schritte verwenden`Key`, also stellen Sie sicher, dass Sie diese speichern.

Wenn Sie den mehrteiligen Upload mit zusätzlichen Prüfsummen verwenden, müssen die Artikelnummern außerdem fortlaufend sein. Wenn Sie nicht aufeinander folgende

Artikelnummern verwenden, kann die `complete-multipart-upload` Anfrage zu einer führen. HTTP `500 Internal Server Error`

Schritt 4: Laden Sie die Teile Ihres mehrteiligen Uploads hoch

In diesem Schritt laden Sie die Teile Ihres mehrteiligen Uploads in Ihren S3-Bucket hoch. Verwenden Sie den `upload-part` Befehl, um jeden Teil einzeln hochzuladen. Dieser Vorgang erfordert die Angabe der Upload-ID, der Artikelnummer und der hochzuladenden Datei für jedes Teil.

Um die Teile hochzuladen

1. Wenn Sie ein Teil hochladen, müssen Sie zusätzlich zur Upload-ID eine Artikelnummer mithilfe des `--part-number` Arguments angeben. Sie können jede Teilenummer zwischen 1 und 10.000 wählen. Die Teilenummer identifiziert eindeutig einen Teil und seine Position im Objekt, das Sie hochladen. Die von Ihnen gewählte Artikelnummer muss in einer aufeinanderfolgenden Reihenfolge angegeben werden (sie kann beispielsweise 1, 2 oder 3 sein). Wenn Sie einen neuen Teil mit derselben Teilenummer hochladen wie bereits einmal zuvor, wird der früher hochgeladene Teil überschrieben.
2. Verwenden Sie den `upload-part` Befehl, um jeden Teil Ihres mehrteiligen Uploads hochzuladen. Der `--upload-id` ist derselbe wie in der Ausgabe, die mit dem `create-multipart-upload` Befehl in Schritt 3 erstellt wurde. Verwenden Sie den folgenden Befehl, um den ersten Teil Ihrer Daten hochzuladen:

   ```
   aws s3api upload-part --bucket amzn-s3-demo-bucket1 --key
    'census_data_file' --part-number 1 --body census-part00 --upload-id
    "cNV6KCSNANFZapz1LUGPC5XwUVi1n6yUoIeSP138sNOKPeMhpKQRrbT9k0ePmgoOTCj9K83T4e2Gb5hQvNoNpCKqy
    --checksum-algorithm SHA256
   ```

   Nach Abschluss jedes `upload-part` Befehls sollten Sie eine Ausgabe wie das folgende Beispiel sehen:

   ```
   {
       "ServerSideEncryption": "AES256",
       "ETag": "\"e611693805e812ef37f96c9937605e69\"",
       "ChecksumSHA256": "QLl8R4i4+SaJlrl8ZIcutc5TbZtwt2NwB8lTXkd3GH0="
   }
   ```

3. Für nachfolgende Teile erhöhen Sie die Artikelnummer entsprechend:

```
aws s3api upload-part --bucket amzn-s3-demo-bucket1 --key 'census_data_file' --
part-number <part-number> --body <file-path> --upload-id "<your-upload-id>" --
checksum-algorithm SHA256
```

   Verwenden Sie beispielsweise den folgenden Befehl, um den zweiten Teil hochzuladen:

```
aws s3api upload-part --bucket amzn-s3-demo-bucket1 --key
 'census_data_file' --part-number 2 --body census-part01 --upload-id
 "cNV6KCSNANFZapz1LUGPC5XwUVi1n6yUoIeSP138sNOKPeMhpKQRrbT9k0ePmgoOTCj9K83T4e2Gb5hQvNoNpCKqy
 --checksum-algorithm SHA256
```

   Amazon S3 gibt ein Entitäts-Tag (ETag) und zusätzliche Prüfsummen für jeden hochgeladenen Teil als Header in der Antwort zurück.

4. Verwenden Sie den `upload-part` Befehl weiter, bis Sie alle Teile Ihres Objekts hochgeladen haben.

Schritt 5: Listet alle Teile eures mehrteiligen Uploads auf

Um den mehrteiligen Upload abzuschließen, benötigen Sie eine Liste aller Teile, die für diesen speziellen mehrteiligen Upload hochgeladen wurden. Die Ausgabe des `list-parts` Befehls enthält Informationen wie Bucket-Namen, Schlüssel, Upload-ID, ArtikelnummereTag, zusätzliche Prüfsummen und mehr. Es ist hilfreich, diese Ausgabe in einer Datei zu speichern, sodass Sie sie für den nächsten Schritt verwenden können, wenn Sie den mehrteiligen Upload-Vorgang abschließen. Sie können eine JSON Ausgabedatei erstellen, die mit `parts.json` der folgenden Methode aufgerufen wird.

Um eine Datei zu erstellen, die alle Teile auflistet

1. Verwenden Sie den folgenden `list-parts` Befehl, um eine JSON Datei mit den Details aller hochgeladenen Teile zu generieren. ***amzn-s3-demo-bucket1***Ersetzen Sie es durch Ihren tatsächlichen Bucket-Namen und **<your-upload-id>** durch die Upload-ID, die Sie in Schritt 3 erhalten haben. Weitere Informationen zu dem `list-parts` Befehl finden Sie unter list-parts im AWS Command Line Interface -Benutzerhandbuch.

```
aws s3api list-parts --bucket amzn-s3-demo-bucket1 --key 'census_data_file' --
upload-id <your-upload-id> --query '{Parts: Parts[*].{PartNumber: PartNumber, ETag:
 ETag, ChecksumSHA256: ChecksumSHA256}}' --output json > parts.json
```

Eine neue Datei namens `parts.json` wird generiert. Die Datei enthält die JSON formatierten Informationen für alle Ihre hochgeladenen Teile. Die `parts.json` Datei enthält wichtige Informationen für jeden Teil Ihres mehrteiligen Uploads, wie z. B. die Artikelnummern und die entsprechenden ETag Werte, die für den Abschluss des mehrteiligen Uploads erforderlich sind.

2. Öffnen Sie `parts.json` sie mit einem beliebigen Texteditor oder über das Terminal. Hier ist die Beispielausgabe:

```
{
    "Parts": [
        {
            "PartNumber": 1,
            "ETag": "\"3c3097f89e2a2fece47ac54b243c9d97\"",
            "ChecksumSHA256": "fTPVHfyNHdv5VkR4S3EewdyioXECv7JBxN+d4FXYYTw="
        },
        {
            "PartNumber": 2,
            "ETag": "\"03c71cc160261b20ab74f6d2c476b450\"",
            "ChecksumSHA256": "VDWTa8enjOvULBAO3W2a6C+5/7ZnNjrnLApa1QVc3FE="
        },
        {
            "PartNumber": 3,
            "ETag": "\"81ae0937404429a97967dffa7eb4affb\"",
            "ChecksumSHA256": "cVVkXehUlzcwrBrXgPIM+EKQXPUvWist8mlUTCs4bg8="
        }
    ]
}
```

Schritt 6: Schließen Sie den mehrteiligen Upload ab

Nachdem Sie alle Teile Ihres mehrteiligen Uploads hochgeladen und aufgelistet haben, besteht der letzte Schritt darin, den mehrteiligen Upload abzuschließen. In diesem Schritt werden alle hochgeladenen Teile zu einem einzigen Objekt in Ihrem S3-Bucket zusammengeführt.

> **Note**
>
> Sie können die Objekt-Prüfsumme vor dem Aufruf berechnen, `complete-multipart-upload` indem Sie sie `--checksum-sha256` in Ihre Anfrage aufnehmen. Wenn die Prüfsummen nicht übereinstimmen, schlägt Amazon S3 die Anfrage fehl. Weitere Informationen finden Sie unter complete-multipart-upload im AWS Command Line Interface - Benutzerhandbuch.

Um den mehrteiligen Upload abzuschließen

Verwenden Sie den Befehl, um den mehrteiligen Upload abzuschließen. `complete-multipart-upload` Für diesen Befehl sind die in Schritt 5 erstellte `parts.json` Datei, Ihr Bucket-Name und die Upload-ID erforderlich. **<*amzn-s3-demo-bucket1*>**Ersetzen Sie es durch Ihren Bucket-Namen und **<your-upload-id>** durch die Upload-ID von`parts.json`.

```
aws s3api complete-multipart-upload --multipart-upload file://parts.json --bucket amzn-s3-demo-bucket1 --key 'census_data_file' --upload-id <your-upload-id>
```

Hier ist die Beispielausgabe:

```
{
    "ServerSideEncryption": "AES256",
    "Location": "https://amzn-s3-demo-bucket1.s3.us-east-2.amazonaws.com/
census_data_file",
    "Bucket": "amzn-s3-demo-bucket1",
    "Key": "census_data_file",
    "ETag": "\"f453c6dccca969c457efdf9b1361e291-3\"",
    "ChecksumSHA256": "aI8EoktCdotjU8Bq46DrPCxQCGuGcPIhJ51noWs6hvk=-3"
}
```

> **Note**
>
> Löschen Sie die einzelnen Teildateien noch nicht. Sie benötigen die einzelnen Teile, damit Sie Prüfsummen auf sie anwenden können, um die Integrität des zusammengeführten Objekts zu überprüfen.

Schritt 7: Vergewissern Sie sich, dass das Objekt in Ihren Bucket hochgeladen wurde

Nach Abschluss des mehrteiligen Uploads können Sie überprüfen, ob das Objekt erfolgreich in Ihren S3-Bucket hochgeladen wurde. Verwenden Sie den Befehl, um die Objekte in Ihrem Bucket aufzulisten und das Vorhandensein Ihrer neu hochgeladenen Datei zu bestätigen `list-objects-v2`

Um das hochgeladene Objekt aufzulisten

Um die Objekte in Ihrem aufzulisten, verwenden Sie den `list-objects-v2` Befehl bucket. ***amzn-s3-demo-bucket1***Ersetzen Sie es durch Ihren tatsächlichen Bucket-Namen:

```
aws s3api list-objects-v2 --bucket amzn-s3-demo-bucket1
```

Dieser Befehl gibt eine Liste der Objekte in Ihrem Bucket zurück. Suchen Sie in der Objektliste nach Ihrer hochgeladenen Datei (zum Beispiel`census_data_file`).

Weitere Informationen finden Sie im Abschnitt Beispiele für den `list-objects-v2` Befehl im AWS Command Line Interface Benutzerhandbuch.

Schritt 8: Überprüfen Sie die Objektintegrität mit einer MD5 Prüfsumme

Wenn Sie ein Objekt hochladen, können Sie einen Prüfsummenalgorithmus angeben, den Amazon S3 verwenden soll. Standardmäßig speichert Amazon S3 den MD5 Byte-Digest als den des ETag Objekts. Bei mehrteiligen Uploads ETag handelt es sich nicht um die Prüfsumme für das gesamte Objekt, sondern um eine Zusammenstellung von Prüfsummen für jeden einzelnen Teil.

Um die Objektintegrität mithilfe einer Prüfsumme zu überprüfen MD5

1. Führen Sie eine `head-object` Anfrage ETag aus, um das hochgeladene Objekt abzurufen:

   ```
   aws s3api head-object --bucket amzn-s3-demo-bucket1 --key census_data_file
   ```

   Hier ist die Beispielausgabe:

   ```
   {
       "AcceptRanges": "bytes",
       "LastModified": "2024-07-26T19:04:13+00:00",
       "ContentLength": 16106127360,
       "ETag": "\"f453c6dccca969c457efdf9b1361e291-3\"",
   ```

```
    "ContentType": "binary/octet-stream",
    "ServerSideEncryption": "AES256",
    "Metadata": {}
}
```

An das Ende dieses Etags ist „-3" angehängt. Dies weist darauf hin, dass das Objekt in drei Teilen mithilfe eines mehrteiligen Uploads hochgeladen wurde.

2. Als Nächstes berechnen Sie die MD5 Prüfsumme der einzelnen Teile mit dem `md5sum` Befehl. Stellen Sie sicher, dass Sie den richtigen Pfad zu Ihren Bauteildateien angeben:

```
md5sum census-part*
```

Hier ist die Beispielausgabe:

```
e611693805e812ef37f96c9937605e69 census-part00
63d2d5da159178785bfd6b6a5c635854 census-part01
95b87c7db852451bb38b3b44a4e6d310 census-part02
```

3. Kombinieren Sie für diesen Schritt die MD5 Hashes manuell zu einer Zeichenfolge. Führen Sie dann den folgenden Befehl aus, um die Zeichenfolge in eine Binärdatei zu konvertieren und die MD5 Prüfsumme des Binärwerts zu berechnen:

```
echo
 "e611693805e812ef37f96c9937605e6963d2d5da159178785bfd6b6a5c63585495b87c7db852451bb38b3b44a
 | xxd -r -p | md5sum
```

Hier ist die Beispielausgabe:

```
f453c6dccca969c457efdf9b1361e291 -
```

Dieser Hashwert sollte mit dem Hashwert des ursprünglichen ETag Werts in Schritt 1 übereinstimmen, der die Integrität des `census_data_file` Objekts bestätigt.

Wenn Sie Amazon S3 anweisen, zusätzliche Prüfsummen zu verwenden, berechnet Amazon S3 den Prüfsummenwert für jeden Teil und speichert die Werte. Wenn Sie die Prüfsummenwerte für

einzelne Teile von mehrteiligen Uploads abrufen möchten, die noch in Bearbeitung sind, können Sie Folgendes verwenden. `list-parts`

Weitere Hinweise zur Funktionsweise von Prüfsummen mit mehrteiligen Upload-Objekten finden Sie unter. Überprüfung der Objektintegrität

Schritt 9: Überprüfen Sie die Objektintegrität mit einer zusätzlichen Prüfsumme

In diesem Schritt verwendet dieses Tutorial SHA -256 als zusätzliche Prüfsumme zur Überprüfung der Objektintegrität. Wenn Sie eine andere zusätzliche Prüfsumme verwendet haben, verwenden Sie stattdessen diesen Prüfsummenwert.

Um die Objektintegrität zu überprüfen mit SHA256

1. Führen Sie den folgenden Befehl in Ihrem Terminal aus, einschließlich des `--checksum-mode enabled` Arguments, um den `ChecksumSHA256` Wert Ihres Objekts anzuzeigen:

   ```
   aws s3api head-object --bucket amzn-s3-demo-bucket1 --key census_data_file --
   checksum-mode enabled
   ```

   Hier ist die Beispielausgabe:

   ```
   {
       "AcceptRanges": "bytes",
       "LastModified": "2024-07-26T19:04:13+00:00",
       "ContentLength": 16106127360,
       "ChecksumSHA256": "aI8EoktCdotjU8Bq46DrPCxQCGuGcPIhJ51noWs6hvk=-3",
       "ETag": "\"f453c6dccca969c457efdf9b1361e291-3\"",
       "ContentType": "binary/octet-stream",
       "ServerSideEncryption": "AES256",
       "Metadata": {}
   }
   ```

2. Verwenden Sie die folgenden Befehle, um die `ChecksumSHA256` Werte für die einzelnen Teile in Base64 zu dekodieren und sie in einer Binärdatei namens zu speichern. `outfile` Diese Werte finden Sie in Ihrer `parts.json` Datei. Ersetzen Sie die Base64-Beispiel-Strings durch Ihre tatsächlichen `ChecksumSHA256` Werte.

```
echo "QLl8R4i4+SaJlrl8ZIcutc5TbZtwt2NwB8lTXkd3GH0=" | base64 --decode >> outfile
echo "xCdgs1K5Bm4jWETYw/CmGYr+m6O2DcGfpckx5NVokvE=" | base64 --decode >> outfile
echo "f5wsfsa5bB+yXuwzqG1Bst91uYneqGD3CCidpb54mAo=" | base64 --decode >> outfile
```

3. Führen Sie den folgenden Befehl aus, um die SHA256 Prüfsumme von zu berechnen: `outfile`

```
sha256sum outfile
```

Hier ist die Beispielausgabe:

```
688f04a24b42768b6353c06ae3a0eb3c2c50086b8670f221279d67a16b3a86f9 outfile
```

Nehmen Sie im nächsten Schritt den Hashwert und konvertieren Sie ihn in einen Binärwert. Dieser Binärwert sollte mit dem `ChecksumSHA256` Wert aus Schritt 1 übereinstimmen.

4. Wandeln Sie die SHA256 Prüfsumme aus Schritt 3 in eine Binärdatei um und codieren Sie sie dann nach Base64, um zu überprüfen, ob sie mit dem `ChecksumSHA256` Wert aus Schritt 1 übereinstimmt:

```
echo "688f04a24b42768b6353c06ae3a0eb3c2c50086b8670f221279d67a16b3a86f9" | xxd -r -p
 | base64
```

Hier ist die Beispielausgabe:

```
aI8EoktCdotjU8Bq46DrPCxQCGuGcPIhJ51noWs6hvk=
```

Diese Ausgabe sollte bestätigen, dass die Base64-Ausgabe mit dem `ChecksumSHA256` Wert aus der `head-object` Befehlsausgabe übereinstimmt. Wenn die Ausgabe mit dem Prüfsummenwert übereinstimmt, ist das Objekt gültig.

Important

- Wenn Sie Amazon S3 anweisen, zusätzliche Prüfsummen zu verwenden, berechnet Amazon S3 die Prüfsummenwerte für jeden Teil und speichert diese Werte.
- Wenn Sie die Prüfsummenwerte für einzelne Teile von mehrteiligen Uploads abrufen möchten, die noch in Bearbeitung sind, können Sie den Befehl verwenden. `list-parts`

Schritt 10: Bereinigen Ihrer Ressourcen

Wenn Sie die in diesem Tutorial erstellten Dateien bereinigen möchten, verwenden Sie die folgende Methode. Anweisungen zum Löschen der in Ihren S3-Bucket hochgeladenen Dateien finden Sie unterLöschen von Amazon-S3-Objekten.

Löschen Sie die in Schritt 1 erstellten lokalen Dateien:

Um die Dateien zu entfernen, die Sie für Ihren mehrteiligen Upload erstellt haben, führen Sie den folgenden Befehl in Ihrem Arbeitsverzeichnis aus:

```
rm census-data.bin census-part* outfile parts.json
```

EInschränkungen mehrteiliger Amazon-S3-Uploads

Mit dem mehrteiligen Upload können Sie ein einzelnes Objekt als Satz aus mehreren Teilen hochladen. Jeder Teil ist ein zusammenhängender Teil der Daten des Objekts. Nachdem alle Teile Ihres Objekts hochgeladen sind, fügt Amazon S3 diese Teile zusammen und erstellt das Objekt. Wenn Ihre Objektgröße 100 MB erreicht, sollten Sie in der Regel mehrteilige Uploads verwenden, anstatt das Objekt in einem einzigen Vorgang hochzuladen. Weitere Informationen über mehrteilige Uploads finden Sie unter Hochladen und Kopieren von Objekten mit mehrteiligen Uploads.

Die folgende Tabelle enthält die Core-Spezifikationen für den mehrteiligen Upload. Dazu gehören die maximale Objektgröße, die maximale Anzahl von Teilen, die maximale Bauteilgröße und mehr. Es gibt keine minimale Größenbeschränkung für den letzten Teil Ihres mehrteiligen Uploads.

Item	Spezifikation
Maximale Objektgröße	5 TiB
Maximale Anzahl von Teilen pro Upload	10.000
Teilenummern	1 bis 10.000 (inklusive)
Teilegröße	5 MiB bis 5 GiB. Es gibt keine minimale Größenbeschränkung für den letzten Teil Ihres mehrteiligen Uploads.

Item	Spezifikation
Maximale Anzahl der zurückgeg ebenen Teile bei einer Anforderung zum Auflisten der Teile	1000
Maximale Anzahl der zurückgeg ebenen mehrteiligen Uploads bei einer Anforderung zum Auflisten mehrteiliger Uploads	1000

Fügen Sie Vorbedingungen zu S3-Vorgängen mit bedingten Anfragen hinzu

Sie können bedingte Anfragen verwenden, um Ihren S3-Vorgängen Vorbedingungen hinzuzufügen. Um bedingte Anfragen zu verwenden, fügen Sie Ihrem Amazon S3 API S3-Vorgang einen zusätzlichen Header hinzu. Dieser Header spezifiziert eine Bedingung, die, wenn sie nicht erfüllt wird, dazu führt, dass der S3-Vorgang fehlschlägt.

Bedingte Lesevorgänge werden für `GET`, `HEAD`, und `COPY` Anfragen unterstützt. Sie können Vorbedingungen hinzufügen, um ein Objekt auf der Grundlage seines Entitäts-Tags (ETag) oder des Datums der letzten Änderung zurückzugeben oder zu kopieren. Dadurch kann ein S3-Vorgang auf Objekte beschränkt werden, die seit einem bestimmten Datum aktualisiert wurden. Sie können einen S3-Vorgang auch auf einen bestimmten Wert beschränkenETag. Dadurch könnte sichergestellt werden, dass Sie nur eine bestimmte Objektversion zurückgeben oder kopieren. Weitere Informationen zu Objekt-Metadaten erhalten Sie unter Arbeiten mit Objekt-Metadaten.

Durch bedingte Schreibvorgänge kann sichergestellt werden, dass während des `PUT` Betriebs kein Objekt mit demselben Schlüsselnamen in Ihrem Bucket vorhanden ist. Dadurch wird verhindert, dass vorhandene Objekte mit identischen Schlüsselnamen überschrieben werden. Sie können bedingte Schreibvorgänge verwenden für PutObjectoder CompleteMultipartUploadAnfragen. Weitere Informationen zu Schlüsselnamen finden Sie unter Benennen Amazon S3 S3-Objekten.

Für bedingte Lesevorgänge oder bedingte Schreibvorgänge fallen keine zusätzlichen Gebühren an. Ihnen werden nur die bestehenden Tarife für die entsprechenden Anfragen berechnet, auch für fehlgeschlagene Anfragen. Informationen zu den Funktionen und Preisen von Amazon S3 finden Sie unter Amazon-S3-Preise.

Themen

- Wie ruft man Objekte auf der Grundlage von Metadaten mit bedingten Lesevorgängen ab oder kopiert sie
- Wie verhindert man das Überschreiben von Objekten durch bedingte Schreibvorgänge

Wie ruft man Objekte auf der Grundlage von Metadaten mit bedingten Lesevorgängen ab oder kopiert sie

Bei bedingten Lesevorgängen können Sie Ihrer Leseanforderung einen zusätzlichen Header hinzufügen, um Ihrem S3-Vorgang Vorbedingungen hinzuzufügen. Wenn diese Voraussetzungen nicht erfüllt sind, schlägt die Leseanforderung fehl.

Sie können bedingte Lesevorgänge für`GET`, oder `COPY` -Anfragen verwenden`HEAD`, um ein Objekt nur auf der Grundlage seiner Metadaten zurückzugeben.

Wenn Sie ein Objekt hochladen, erstellt Amazon S3 systemgesteuerte Metadaten, die nur von S3 geändert werden können. Entity Tags (ETags) und Last-Modified sind Beispiele für systemgesteuerte Metadaten. Bei einem Objekt ETag handelt es sich um eine Zeichenfolge, die eine bestimmte Version eines Objekts darstellt. Bei dem Datum der letzten Änderung handelt es sich um Metadaten, die das Erstellungsdatum oder das Datum der letzten Änderung eines Objekts angeben, je nachdem, welches Datum das letzte ist.

Bei bedingten Lesevorgängen können Sie ein Objekt auf der Grundlage des Objekts ETag oder des Datums der letzten Änderung zurückgeben. Sie können in Ihrer Anfrage einen ETag Wert angeben und das Objekt nur zurückgeben, wenn der ETag Wert übereinstimmt. Dadurch kann sichergestellt werden, dass Sie nur eine bestimmte Version eines Objekts zurückgeben oder kopieren. Sie können in Ihrer Leseanforderung den Wert Letzte Änderung angeben und ein Objekt nur zurückgeben, wenn dieses Objekt seit einem von Ihnen angegebenen Datum geändert wurde.

Unterstützt APIs

Die folgende APIs S3-Unterstützung mit bedingten Lesevorgängen:

- GetObject
- HeadObject
- CopyObject

Sie können die folgenden Header verwenden, um ein Objekt zurückzugeben, das vom Entity-Tag (ETag) oder dem Datum der letzten Änderung abhängig ist. Weitere Hinweise zu Objektmetadaten wie ETags Last-Modified finden Sie unter. the section called “Systemdefinierte Objektmetadaten”

GetObject

- `If-Match`— Gibt das Objekt nur zurück, wenn es ETag mit dem angegebenen übereinstimmt.
- `If-Modified-Since`— Gibt das Objekt nur zurück, wenn es seit dem angegebenen Zeitpunkt geändert wurde.
- `If-None-Match`— Gibt das Objekt nur zurück, wenn ETag es nicht mit dem angegebenen übereinstimmt.
- `If-Unmodified-Since`— Gibt das Objekt nur zurück, wenn es seit dem angegebenen Zeitpunkt nicht geändert wurde.

Weitere Hinweise zu diesen Headern, den zurückgegebenen Fehlern und der Reihenfolge, in der S3 mehrere bedingte Header in einer einzigen Anfrage verarbeitet, finden Sie unter GetObjectin der Amazon Simple Storage Service API Reference.

HeadObject

- `If-Match`— Gibt das Objekt nur zurück, wenn es ETag mit dem angegebenen übereinstimmt.
- `If-Modified-Since`— Gibt das Objekt nur zurück, wenn es seit dem angegebenen Zeitpunkt geändert wurde.
- `If-None-Match`— Gibt das Objekt nur zurück, wenn ETag es nicht mit dem angegebenen übereinstimmt.
- `If-Unmodified-Since`— Gibt das Objekt nur zurück, wenn es seit dem angegebenen Zeitpunkt nicht geändert wurde.

Weitere Hinweise zu diesen Headern, den zurückgegebenen Fehlern und der Reihenfolge, in der S3 mehrere bedingte Header in einer einzigen Anfrage verarbeitet, finden Sie unter HeadObjectin der Amazon Simple Storage Service API Reference.

CopyObject

- `x-amz-copy-source-if-match`— Kopiert das Quellobjekt nur, wenn es ETag mit dem angegebenen übereinstimmt.
- `x-amz-copy-source-if-modified-since`— Kopiert das Quellobjekt nur, wenn es seit dem angegebenen Zeitpunkt geändert wurde.
- `x-amz-copy-source-if-none-match`— Kopiert das Quellobjekt nur, wenn ETag es nicht mit dem angegebenen übereinstimmt.
- `x-amz-copy-source-if-unmodified-since`— Kopiert das Quellobjekt nur, wenn es seit dem angegebenen Zeitpunkt nicht geändert wurde.

Weitere Hinweise zu diesen Headern, den zurückgegebenen Fehlern und der Reihenfolge, in der S3 mehrere bedingte Header in einer einzigen Anfrage verarbeitet, finden Sie unter CopyObjectin der Amazon Simple Storage Service API Reference.

Wie verhindert man das Überschreiben von Objekten durch bedingte Schreibvorgänge

Bei bedingten Schreibvorgängen können Sie Ihrer Schreibanforderung einen zusätzlichen Header hinzufügen, um Vorbedingungen für Ihren S3-Vorgang festzulegen. Auf diese Weise können Sie das Überschreiben vorhandener Daten verhindern, indem überprüft wird, ob sich in Ihrem Bucket noch kein Objekt mit demselben Schlüsselnamen befindet. Bedingte Schreibvorgänge funktionieren für API Anfragen in Allzweck-Buckets und Verzeichnis-Buckets.

Wenn Sie ein Objekt auf Amazon S3 hochladen, geben Sie den Schlüsselnamen an. Der Schlüsselname ist eine eindeutige Kennung eines Objekts in einem Bucket, bei der Groß- und Kleinschreibung berücksichtigt wird. Wenn Sie ein Objekt mit einem identischen Schlüsselnamen in einen Bucket hochladen, für den keine Version verwendet wurde oder dessen Version gesperrt wurde, wird das Objekt überschrieben. In einem versionierten Bucket wird das zuletzt hochgeladene Objekt zur aktuellen Version des Objekts.

Bei bedingten Schreibvorgängen wird während des `WRITE` Vorgangs geprüft, ob ein Objekt vorhanden ist. Wenn im Bucket ein identischer Schlüsselname gefunden wird, schlägt der `WRITE` Vorgang fehl. Bei bedingten Schreibvorgängen können mehrere Clients in denselben Bucket schreiben, ohne dass die Möglichkeit besteht, bestehende Objekte zu überschreiben.

Um bedingte Schreibvorgänge ausführen zu können, benötigen Sie die `s3:PutObject` entsprechende Genehmigung. Diese Berechtigung ermöglicht es dem Aufrufer, zu überprüfen, ob

Objekte im Bucket vorhanden sind. Sie können bedingte Schreibvorgänge mit vorsigniertem mit URLs dem verwenden. AWS SDKs

> **Note**
>
> Um bedingte Schreibvorgänge zu verwenden, müssen Sie die Anfragen über HTTPS (TLS) stellen oder AWS Signature Version 4 verwenden, um die Anfrage zu signieren.

Unterstützt APIs

Die folgende APIs S3-Unterstützung mit bedingten Schreibvorgängen:

- PutObject
- CompleteMultipartUpload

Sie können die folgenden Header verwenden, um ein Objekt zu schreiben, das vom Objektschlüsselnamen abhängig ist. Hinweise zu Objektschlüsselnamen finden Sie unter,the section called “Benennen von Objekten”.

PutObject

- `If-None-Match`— Lädt das Objekt nur hoch, wenn im angegebenen Bucket noch kein Objekt mit demselben Schlüsselnamen existiert. Sie müssen den Wert * (Sternchen) mit diesem Parameter verwenden.

Verwenden Sie den AWS CLI

Der folgende `put-object` Beispielbefehl zeigt, wie Sie mithilfe des AWS CLI `if-none-match` Parameters ein Objekt mit einem bedingten Schreib-Header hochladen können.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key dir-1/my_images.tar.bz2 --body
 my_images.tar.bz2 --if-none-match "*"
```

Weitere Informationen finden Sie unter put-objectin der AWS CLI Befehlsreferenz.

Informationen zu dem AWS CLI finden Sie unter Was ist der AWS Command Line Interface? im AWS Command Line Interface Benutzerhandbuch.

Weitere Informationen zu diesen Headern finden Sie unter PutObjectin der Amazon Simple Storage Service API Reference.

CompleteMultipartUpload

- `If-None-Match`— Schließt den Upload nur ab, wenn im angegebenen Bucket noch kein Objekt mit demselben Schlüsselnamen vorhanden ist. Sie müssen den Wert * (Sternchen) für diesen Parameter verwenden.

Verwenden Sie den AWS CLI

Der folgende `complete-multipart-upload` Beispielbefehl zeigt, wie Sie mithilfe des AWS CLI `if-none-match` Parameters einen mehrteiligen Upload mit einem Header für bedingtes Schreiben abschließen können.

```
aws s3api complete-multipart-upload --multipart-upload file://mpustruct --bucket amzn-s3-demo-bucket --key dir-1/my_images.tar.bz2 --upload-id uploadID --if-none-match "*"
```

Weitere Informationen finden Sie unter complete-multipart-uploadin der AWS CLI Befehlsreferenz.

Informationen zu dem AWS CLI finden Sie unter Was ist der AWS Command Line Interface? im AWS Command Line Interface Benutzerhandbuch.

Weitere Informationen zu diesen Headern finden Sie unter CompleteMultipartUploadin der Amazon Simple Storage Service API Reference.

Bedingtes Schreibverhalten

Bedingte Schreibvorgänge werden anhand vorhandener Objekte in einem Bucket ausgewertet. Wenn im Bucket kein Objekt mit demselben Schlüsselnamen vorhanden ist, ist der Schreibvorgang erfolgreich, was zu einer `200` Antwort führt. Wenn ein Objekt vorhanden ist, schlägt der Schreibvorgang fehl, was zu einer `412 Precondition Failed` Antwort führt. Bei Buckets mit aktivierter Versionierung prüft S3 im Rahmen der bedingten Bewertung, ob eine aktuelle Objektversion mit demselben Namen vorhanden ist. Wenn es keine aktuelle Objektversion mit demselben Namen gibt oder wenn es sich bei der aktuellen Objektversion um eine Löschmarke handelt, ist der Schreibvorgang erfolgreich. Andernfalls führt dies zu einem fehlgeschlagenen Schreibvorgang mit einer `412 Precondition Failed` Antwort.

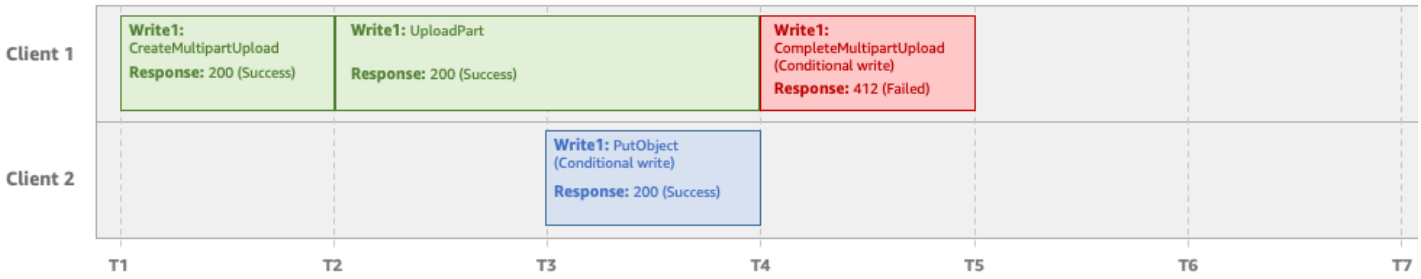
Wenn mehrere bedingte Schreibvorgänge für denselben Objektnamen ausgeführt werden, ist der erste abgeschlossene Schreibvorgang erfolgreich. Amazon S3 schlägt dann nachfolgende Schreibvorgänge mit einer `412 Precondition Failed` Antwort fehl.

Sie können auch bei gleichzeitigen Anfragen eine `409 Conflict` Antwort erhalten, wenn eine Löschanforderung für ein Objekt erfolgreich ist, bevor ein bedingter Schreibvorgang für dieses Objekt abgeschlossen werden kann. Das liegt daran, dass die Löschanforderung Vorrang vor dem zuvor initiierten bedingten Schreibvorgang hat. Wenn Sie bedingte Schreibvorgänge mit verwenden`PutObject`, werden Uploads möglicherweise erneut versucht, nachdem ein Fehler aufgetreten ist. `409` Bei Verwendung `CompleteMultipartUpload` muss der mehrteilige Upload erneut initiiert werden, um das Objekt nach Erhalt eines Fehlers erneut hochzuladen. `CreateMultipartUpload` `409`

Stellen Sie sich die folgenden Szenarien vor, in denen zwei Clients Operationen auf demselben Bucket ausführen.

4.1.2 Vorbedingung: Fehlgeschlagene Antwort

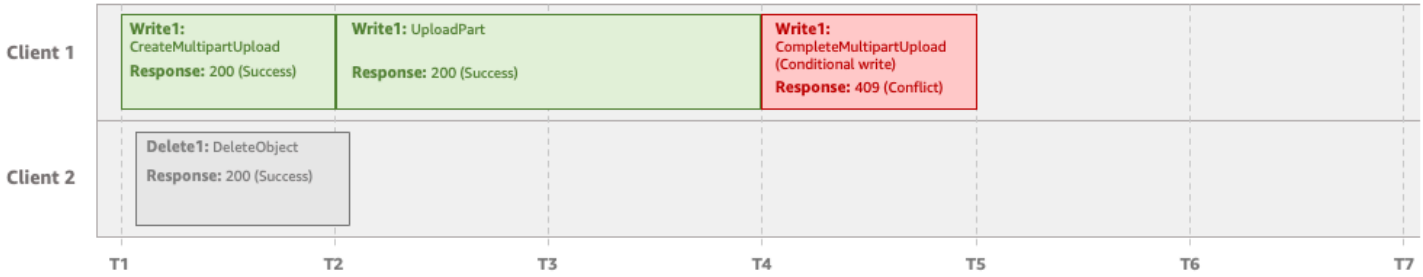
Bei bedingten Schreibvorgängen werden Anfragen für mehrteilige Uploads nicht berücksichtigt, da es sich bei diesen noch nicht um vollständig geschriebene Objekte handelt. Stellen Sie sich das folgende Beispiel vor, in dem Client 1 ein Objekt mithilfe eines mehrteiligen Uploads hochlädt. Während des mehrteiligen Uploads kann Client 2 dasselbe Objekt mit dem bedingten Schreibvorgang erfolgreich schreiben. Wenn Client 1 anschließend versucht, den mehrteiligen Upload mithilfe eines bedingten Schreibvorgangs abzuschließen, schlägt der Upload fehl, da das Objekt bereits existiert.



409 Reaktion auf Konflikte

Wenn eine Löschanforderung erfolgreich ist, bevor eine bedingte Schreibanforderung abgeschlossen werden kann, gibt Amazon S3 eine `409 Conflict` Antwort für den Schreibvorgang zurück. Das liegt daran, dass die zuvor initiierte Löschanforderung Vorrang vor dem bedingten Schreibvorgang hat. Betrachten Sie das folgende Beispiel, in dem eine Datei in einem Bucket `puppy.txt` vorhanden ist. Client 1 startet einen mehrteiligen Upload für eine andere Datei, die ebenfalls benannt ist,

`puppy.txt` mit der Absicht, den mehrteiligen Upload mit einem bedingten Schreibvorgang abzuschließen. Während des Uploads löscht Client 2 `puppy.txt` aus dem Bucket. Wenn Client 1 versucht, seine eigene `puppy.txt` Datei mit einem bedingten Schreibvorgang hochzuladen, schlägt dies fehl und es wird eine `409 Conflict` Antwort angezeigt. `CompleteMultipartUpload` In solchen Fällen müssen Sie einen neuen mehrteiligen Upload initiieren.



> **Note**
>
> Wir empfehlen, eine Lebenszyklusregel zu konfigurieren, durch die unvollständige mehrteilige Uploads nach einer bestimmten Anzahl von Tagen mit der Aktion `AbortIncompleteMultipartUpload` gelöscht werden, um Ihre Speicherkosten gering zu halten. Weitere Informationen zum Erstellen einer Lebenszyklusregel zum Löschen unvollständiger mehrteiliger Uploads finden Sie unter Konfigurieren einer Bucket-Lebenszykluskonfiguration zum Löschen unvollständiger mehrteiliger Uploads.

Objekte kopieren, verschieben und umbenennen

Der `CopyObject` Vorgang erstellt eine Kopie eines Objekts, das bereits in Amazon S3 gespeichert ist.

Sie können in einem einzigen atomaren Vorgang eine Kopie eines Objekts mit einer Größe von bis zu 5 GB erstellen. Um jedoch ein Objekt zu kopieren, das größer als 5 GB ist, müssen Sie einen mehrteiligen Upload mit dem AWS CLI oder AWS SDKs verwenden. Weitere Informationen finden Sie unter the section called “Kopieren eines Objekts mit Multipart-Upload”.

Mit der `CopyObject`-Operation können Sie:

- Erstellen Sie zusätzliche Kopien von Objekten.
- Benennen Sie Objekte um, indem Sie sie kopieren und die Originale löschen.

- Kopieren oder verschieben Sie Objekte von einem Bucket in einen anderen, auch quer AWS-Regionen (z. B. von `us-west-1` bis `eu-west-2`). Wenn Sie ein Objekt verschieben, kopiert Amazon S3 das Objekt an das angegebene Ziel und löscht dann das Quellobjekt.

  > **Note**
  >
  > Beim Kopieren oder Verschieben von Objekten fallen AWS-Regionen Bandbreitengebühren an. Weitere Informationen finden Sie unter Amazon S3 – Preise.

- Objektmetadaten ändern. Jedes Amazon-S3-Objekt hat Metadaten. Bei diesen Metadaten handelt es sich um eine Reihe von Name-Wert-Paaren. Sie können Objektmetadaten festlegen, wenn Sie ein Objekt hochladen. Nachdem Sie das Objekt hochgeladen haben, können Sie die Objektmetadaten nicht mehr ändern. Die einzige Methode, wie Sie Objekt-Metadaten ändern können, ist es, eine Kopie des Objekts anzulegen und die Metadaten festzulegen. Stellen Sie dazu beim Kopiervorgang dasselbe Objekt als Quelle und Ziel ein.

  Einige Objektmetadaten sind Systemmetadaten und andere sind benutzerdefiniert. Sie können einige der Systemmetadaten steuern. Sie können beispielsweise die Speicherklasse und die Art der serverseitigen Verschlüsselung steuern, die für das Objekt verwendet werden sollen. Wenn Sie ein Objekt kopieren, werden auch die Systemmetadaten und benutzerdefinierte Metadaten kopiert. Amazon S3 setzt die vom System gesteuerten Metadaten zurück. Wenn Sie beispielsweise ein Objekt kopieren, setzt Amazon S3 das Erstellungsdatum des kopierten Objekts zurück. Sie müssen in Ihrer Kopieranforderung keinen dieser systemgesteuerten Metadatenwerte festlegen.

  Wenn Sie ein Objekt kopieren, wollen Sie möglicherweise einige der Metadatenwerte aktualisieren. Wenn Ihr Quellobjekt beispielsweise für die Verwendung des S3-Standardspeichers konfiguriert ist, könnten Sie entscheiden, für die Objektkopie S3 Intelligent Tiering zu verwenden. Außerdem könnten Sie entscheiden, einige der benutzerdefinierten Metadatenwerte für das Quellobjekt zu aktualisieren. Wenn Sie entschieden haben, beim Kopieren vom Benutzer des Objekts konfigurierbare Metadaten zu aktualisieren (System oder benutzerdefiniert), müssen Sie explizit alle vom Benutzer konfigurierbaren Metadaten angeben, die im Quellobjekt Ihrer Anforderung vorhanden sind, selbst wenn Sie nur einen der Metadatenwerte ändern.

  Weitere Informationen zu Objekt-Metadaten erhalten Sie unter Arbeiten mit Objekt-Metadaten.

Archivierte und wiederhergestellte Objekte werden kopiert

Wenn das Quellobjekt in S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive archiviert ist, müssen Sie zuerst eine temporäre Kopie wiederherstellen, bevor Sie das Objekt in einen anderen Bucket kopieren können. Weitere Informationen über das Archivieren von Objekten finden Sie unter Arbeiten mit archivierten Objekten.

Der Kopiervorgang in der Amazon S3 S3-Konsole wird für wiederhergestellte Objekte in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive nicht unterstützt. Um diese wiederhergestellten Objekte zu kopieren, verwenden Sie den AWS Command Line Interface (AWS CLI) AWS SDKs, den oder den Amazon S3 RESTAPI.

Verschlüsselte Objekte werden kopiert

Amazon S3 verschlüsselt alle neuen Objekte, die in einen S3-Bucket kopiert werden, automatisch. Wenn Sie in Ihrer Kopieranforderung keine Verschlüsselungsinformationen angeben, wird die Verschlüsselungseinstellung des Zielobjekts auf die Standardverschlüsselungskonfiguration des Ziel-Buckets festgelegt. Standardmäßig haben alle Buckets eine grundlegende Verschlüsselungskonfiguration, die serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verwendet. Wenn der Ziel-Bucket über eine Standardverschlüsselungskonfiguration verfügt, die serverseitige Verschlüsselung mit einem AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) oder einem vom Kunden bereitgestellten Verschlüsselungsschlüssel (SSE-C) verwendet, verwendet Amazon S3 den entsprechenden KMS Schlüssel oder einen vom Kunden bereitgestellten Schlüssel, um die Zielobjektkopie zu verschlüsseln.

Wenn Sie beim Kopieren eines Objekts eine andere Art der Verschlüsselungseinstellung für das Zielobjekt verwenden möchten, können Sie verlangen, dass Amazon S3 das Zielobjekt mit einem KMS Schlüssel, einem von Amazon S3 verwalteten Schlüssel oder einem vom Kunden bereitgestellten Schlüssel verschlüsselt. Wenn sich die Verschlüsselungseinstellung in Ihrer Anforderung von der Standardverschlüsselungskonfiguration des Ziel-Buckets unterscheidet, hat die Verschlüsselungseinstellung in Ihrer Anfrage Vorrang. Wenn das Quellobjekt für die Kopie mit SSE-C verschlüsselt ist, müssen Sie in Ihrer Anfrage die erforderlichen Verschlüsselungsinformationen angeben, damit Amazon S3 das Objekt zum Kopieren entschlüsseln kann. Weitere Informationen finden Sie unter Datenschutz durch Verschlüsselung.

Verwendung von Prüfsummen beim Kopieren von Objekten

Beim Kopieren von Objekten können Sie einen anderen Prüfsummenalgorithmus für das Objekt verwenden. Unabhängig davon, ob Sie denselben oder einen neuen Algorithmus verwenden, berechnet Amazon S3 einen neuen Prüfsummenwert, nachdem das Objekt kopiert wurde. Amazon

S3 kopiert den Wert der Prüfsumme nicht direkt. Der Prüfsummenwert von Objekten, die mithilfe von mehrteiligen Uploads geladen wurden, kann sich ändern. Weitere Informationen zum Berechnen der Prüfsumme finden Sie unter Verwenden von Prüfsummen auf Teilebene für mehrteilige Uploads.

Kopieren mehrerer Objekte in einer einzigen Anfrage

Um mehr als ein Amazon S3 S3-Objekt mit einer einzigen Anfrage zu kopieren, können Sie auch S3 Batch Operations verwenden. Sie stellen S3 Batch Operations eine Liste von Objekten zur Verfügung, für die Vorgänge ausgeführt werden sollen. S3 Batch Operations ruft die entsprechende API Operation auf, um die angegebene Operation auszuführen. Ein einzelner Batch-Vorgangsauftrag kann die angegebene Operation für Milliarden von Objekten ausführen, die Exabytes von Daten enthalten.

Die Funktion „S3-Batchoperationen“ verfolgt den Fortschritt, versendet Benachrichtigungen und speichert einen detaillierten Abschlussbericht zu allen Aktionen. Sie profitieren von einer vollständig verwalteten, prüfbaren und serverlosen Umgebung. Sie können S3 Batch Operations über die Amazon S3 S3-Konsole AWS CLI, AWS SDKs, oder verwenden RESTAPI. Weitere Informationen finden Sie unter the section called “Grundlagen von BatchVorgänge”.

Objekte in Verzeichnis-Buckets kopieren

Hinweise zum Kopieren eines Objekts in einen Verzeichnis-Bucket finden Sie unterObjekte aus oder in einen Verzeichnis-Bucket kopieren. Informationen zur Verwendung der Amazon S3 Express One Zone-Speicherklasse mit Verzeichnis-Buckets finden Sie unter S3 Express One Zone undArbeiten mit Verzeichnis-Buckets.

Ein Objekt kopieren

Verwenden Sie die folgenden Methoden, um ein Objekt zu kopieren.

Verwenden der S3-Konsole

> **Note**
>
> Die Einschränkungen und Einschränkungen beim Kopieren eines Objekts mit der Konsole lauten wie folgt:
>
> - Sie können ein Objekt kopieren, wenn Ihr Objekt weniger als 5 GB groß ist. Wenn Ihr Objekt größer als 5 GB ist, müssen Sie das AWS CLIoder verwenden, AWS SDKsum ein Objekt zu kopieren.

- Eine Liste der zusätzlichen Berechtigungen, die zum Kopieren von Objekten erforderlich sind, finden Sie unterthe section called “Erforderliche Berechtigungen für API S3-Operationen”. Richtlinien, die diese Berechtigung gewähren, finden Sie beispielsweise unterthe section called “Beispiele für identitätsbasierte Richtlinien”.
- Die `Copy` Aktion gilt für alle Objekte in den angegebenen Ordnern (Präfixe). Objekte, die diesen Ordnern hinzugefügt wurden, während die Aktion ausgeführt wird, können betroffen sein.
- Regionsübergreifendes Kopieren von Objekten, die mit SSE — verschlüsselt sind, KMS wird von der Amazon S3 S3-Konsole nicht unterstützt. Verwenden Sie Amazon S3, um Objekte, die mit SSE — verschlüsselt sind AWS CLI AWS SDK, KMS über Regionen hinweg zu kopieren RESTAPI.
- Objekte, die mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) verschlüsselt wurden, können nicht mithilfe der S3-Konsole kopiert werden. Verwenden Sie, oder Amazon S3 RESTAPI, um mit SSE -C verschlüsselte Objekte zu kopieren. AWS CLI AWS SDK
- Bei kopierten Objekten werden die Objektsperren-Einstellungen der Originalobjekte nicht beibehalten.
- Wenn der Bucket, aus dem Sie Objekte kopieren, die vom Bucket-Besitzer erzwungene Einstellung für S3-Objekteigentum verwendet, ACLs wird das Objekt nicht an das angegebene Ziel kopiert.
- Wenn Sie Objekte in einen Bucket kopieren möchten, der die vom Bucket-Besitzer erzwungene Einstellung für S3-Objekteigentum verwendet, stellen Sie sicher, dass der Quell-Bucket auch die erzwungene Einstellung des Bucket-Besitzers verwendet, oder entfernen Sie alle ACL Objektzuweisungen an andere AWS Konten und Gruppen.

Ein Objekt kopieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets und dann den Tab Buckets für allgemeine Zwecke aus. Navigieren Sie zum Amazon-S3-Bucket oder -Ordner, der die Objekte enthält, die Sie kopieren möchten.
3. Markieren Sie das Kontrollkästchen links neben den Namen der Objekte, die Sie kopieren möchten.

4. Wählen Sie im Menü Aktionen aus der daraufhin angezeigten Optionsliste die Option Kopieren aus.
5. Wählen Sie den Zieltyp und das Zielkonto aus. Um den Zielpfad anzugeben, wählen Sie Browse S3 (S3 durchsuchen), navigieren Sie zum Ziel und markieren Sie das Kontrollkästchen links von dem Ziel. Wählen Sie unten rechts Choose destination (Ziel auswählen) aus.

   Geben Sie alternativ den Zielpfad ein.
6. Wenn Sie die Bucket-Versionierung nicht aktiviert haben, wird eine Warnung angezeigt, in der empfohlen wird, die Bucket-Versionierung zu aktivieren, um sich vor dem unbeabsichtigten Überschreiben oder Löschen von Objekten zu schützen. Wenn Sie alle Versionen von Objekten in diesem Bucket behalten möchten, wählen Sie Enable Bucket Versioning (Bucket-Versionierung aktivieren). Sie können die Standardeigenschaften für Verschlüsselung und S3 Object Lock auch in den Zieldetails einsehen.
7. Wählen Sie unter Zusätzliche Kopiereinstellungen aus, ob Sie Quelleinstellungen kopieren, Keine Einstellungen angeben oder Einstellungen angeben möchten. Quelleinstellungen kopieren ist die Standardoption. Wenn Sie das Objekt nur ohne die Quelleinstellungsattribute kopieren möchten, wählen Sie „Keine Einstellungen angeben“. Wählen Sie „Einstellungen angeben“, um Einstellungen für SpeicherklasseACLs, Objekt-Tags, Metadaten, serverseitige Verschlüsselung und zusätzliche Prüfsummen anzugeben.
8. Wählen Sie unten rechts Copy (Kopieren) aus. Amazon S3 kopiert Ihre Objekte in den Zielordner.

Unter Verwendung der AWS SDKs

Die Beispiele in diesem Abschnitt zeigen, wie Objekte mit bis zu 5 GB in einer einzigen Operation kopiert werden können. Um Objekte zu kopieren, die größer als 5 GB sind, müssen Sie einen mehrteiligen Upload verwenden. Weitere Informationen finden Sie unter Kopieren eines Objekts mit Multipart-Upload.

Java

Example

Das folgende Beispiel kopiert ein Objekt mit AWS SDK for Java in Amazon S3. Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CopyObjectRequest;

import java.io.IOException;

public class CopyObjectSingleOperation {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String sourceKey = "*** Source object key *** ";
        String destinationKey = "*** Destination object key ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Copy the object into a new object in the same bucket.
            CopyObjectRequest copyObjRequest = new CopyObjectRequest(bucketName,
 sourceKey, bucketName, destinationKey);
            s3Client.copyObject(copyObjRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Im folgenden C#-Beispiel wird High-Level verwendet, AWS SDK for .NET um Objekte mit einer Größe von bis zu 5 GB in einem einzigen Vorgang zu kopieren. Für Objekte mit mehr als 5 GB müssen Sie das in Kopieren eines Objekts mit Multipart-Upload beschriebene Kopierbeispiel für mehrteilige Uploads verwenden.

In diesem Beispiel wird eine Kopie eines Objekts bis zu einer Größe von 5 GB erstellt. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem AWS SDK for. NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CopyObjectTest
    {
        private const string sourceBucket = "*** provide the name of the bucket with
 source object ***";
        private const string destinationBucket = "*** provide the name of the bucket
 to copy the object to ***";
        private const string objectKey = "*** provide the name of object to copy
 ***";
        private const string destObjectKey = "*** provide the destination object key
 name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            Console.WriteLine("Copying an object");
            CopyingObjectAsync().Wait();
        }

        private static async Task CopyingObjectAsync()
        {
```

```
            try
            {
                CopyObjectRequest request = new CopyObjectRequest
                {
                    SourceBucket = sourceBucket,
                    SourceKey = objectKey,
                    DestinationBucket = destinationBucket,
                    DestinationKey = destObjectKey
                };
                CopyObjectResponse response = await
 s3Client.CopyObjectAsync(request);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
    }
}
```

PHP

Dieses Thema führt Sie durch die Verwendung von Klassen aus Version 3 von AWS SDK for PHP , um ein einzelnes Objekt und mehrere Objekte innerhalb von Amazon S3, von einem Bucket in einen anderen oder innerhalb desselben Buckets zu kopieren.

Weitere Informationen zu AWS SDK for Ruby API finden Sie unter AWS SDKfor Ruby — Version 2.

Das folgende PHP Beispiel veranschaulicht die Verwendung der `copyObject()` Methode zum Kopieren eines einzelnen Objekts innerhalb von Amazon S3. Es zeigt auch, wie Sie mehrere Kopien eines Objekts erstellen können, indem Sie eine Reihe von Aufrufen `CopyObject` mit der `getcommand()` Methode verwenden.

Objekte kopieren

1	Erstellen Sie mit dem Klassenkonstruktor `Aws\S3\S3Client` eine Instance eines Amazon-S3-Clients.
2	Um mehrere Kopien eines Objekts zu erstellen, führen Sie eine Reihe von Aufrufen an den Amazon S3 S3-Client aus. getCommand()Methode, die von der geerbt wird Aws\CommandInterfaceKlasse. Sie stellen den `CopyObject` -Befehl als erstes Argument, sowie ein Array mit dem Quell-Bucket, dem Quellschlüsselnamen, dem Ziel-Bucket und dem Zielschlüsselnamen als zweites Argument bereit.

```
 require 'vendor/autoload.php';

use Aws\CommandPool;
use Aws\Exception\AwsException;
use Aws\ResultInterface;
use Aws\S3\S3Client;

$sourceBucket = '*** Your Source Bucket Name ***';
$sourceKeyname = '*** Your Source Object Key ***';
$targetBucket = '*** Your Target Bucket Name ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region' => 'us-east-1'
]);

// Copy an object.
$s3->copyObject([
    'Bucket' => $targetBucket,
    'Key' => "$sourceKeyname-copy",
    'CopySource' => "$sourceBucket/$sourceKeyname",
]);

// Perform a batch of CopyObject operations.
$batch = array();
for ($i = 1; $i <= 3; $i++) {
    $batch[] = $s3->getCommand('CopyObject', [
        'Bucket' => $targetBucket,
        'Key' => "{targetKeyname}-$i",
        'CopySource' => "$sourceBucket/$sourceKeyname",
```

```
    ]);
}
try {
    $results = CommandPool::batch($s3, $batch);
    foreach ($results as $result) {
        if ($result instanceof ResultInterface) {
            // Result handling here
        }
        if ($result instanceof AwsException) {
            // AwsException handling here
        }
    }
} catch (Exception $e) {
    // General error handling here
}
```

Python

```
class ObjectWrapper:
    """Encapsulates S3 object actions."""

    def __init__(self, s3_object):
        """
        :param s3_object: A Boto3 Object resource. This is a high-level resource in
 Boto3
                          that wraps object actions in a class-like structure.
        """
        self.object = s3_object
        self.key = self.object.key
```

```
    def copy(self, dest_object):
        """
        Copies the object to another bucket.

        :param dest_object: The destination object initialized with a bucket and
 key.
                            This is a Boto3 Object resource.
        """
        try:
            dest_object.copy_from(
```

```
                CopySource={"Bucket": self.object.bucket_name, "Key":
self.object.key}
            )
            dest_object.wait_until_exists()
            logger.info(
                "Copied object from %s:%s to %s:%s.",
                self.object.bucket_name,
                self.object.key,
                dest_object.bucket_name,
                dest_object.key,
            )
        except ClientError:
            logger.exception(
                "Couldn't copy object from %s/%s to %s/%s.",
                self.object.bucket_name,
                self.object.key,
                dest_object.bucket_name,
                dest_object.key,
            )
            raise
```

Ruby

Die folgenden Aufgaben führen Sie durch die Verwendung des Ruby Klassen zum Kopieren eines Objekts in Amazon S3 von einem Bucket in einen anderen oder innerhalb desselben Buckets.

Objekte kopieren

1	Verwenden Sie das modularisierte Amazon S3 S3-Gem für Version 3 von AWS SDK for Ruby`aws-sdk-s3` , fordern Sie an und geben Sie Ihre AWS Anmeldein formationen ein. Weitere Informationen zur Angabe Ihrer Anmeldeinformationen finden Sie unter Anfragen mit AWS Konto- oder IAM Benutzeranmeldedaten stellen in der Amazon S3 API S3-Referenz.
2	Geben Sie die Anforderungsinformationen wie den Quell-Bucket-Namen, den Quellschlüsselnamen, den Ziel-Bucket-Namen und den Zielschlüssel an.

Folgendes Ruby Das Codebeispiel veranschaulicht die vorherigen Aufgaben, indem die `#copy_object` Methode verwendet wird, um ein Objekt von einem Bucket in einen anderen zu kopieren.

```
require 'aws-sdk-s3'

# Wraps Amazon S3 object actions.
class ObjectCopyWrapper
  attr_reader :source_object

  # @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is
 used as the source object for
  #                                       copy actions.
  def initialize(source_object)
    @source_object = source_object
  end

  # Copy the source object to the specified target bucket and rename it with the
 target key.
  #
  # @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
 object is copied.
  # @param target_object_key [String] The key to give the copy of the object.
  # @return [Aws::S3::Object, nil] The copied object when successful; otherwise,
 nil.
  def copy_object(target_bucket, target_object_key)
    @source_object.copy_to(bucket: target_bucket.name, key: target_object_key)
    target_bucket.object(target_object_key)
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's why:
 #{e.message}"
  end
end

# Example usage:
def run_demo
  source_bucket_name = "amzn-s3-demo-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "amzn-s3-demo-bucket2"
  target_key = "my-target-file.txt"

  source_bucket = Aws::S3::Bucket.new(source_bucket_name)
  wrapper = ObjectCopyWrapper.new(source_bucket.object(source_key))
  target_bucket = Aws::S3::Bucket.new(target_bucket_name)
  target_object = wrapper.copy_object(target_bucket, target_key)
  return unless target_object
```

```
  puts "Copied #{source_key} from #{source_bucket_name} to
 #{target_object.bucket_name}:#{target_object.key}."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Unter Verwendung des REST API

In diesem Beispiel wird beschrieben, wie ein Objekt mithilfe von Amazon S3 kopiert wird RESTAPI. Weitere Informationen über das finden REST API Sie unter CopyObject.

Dieses Beispiel kopiert das `flotsam`-Objekt aus dem *amzn-s3-demo-bucket1*-Bucket in das `jetsam`-Objekt des *amzn-s3-demo-bucket2*-Buckets, wobei seine Metadaten beibehalten werden.

```
PUT /jetsam HTTP/1.1
Host: amzn-s3-demo-bucket2.s3.amazonaws.com
x-amz-copy-source: /amzn-s3-demo-bucket1/flotsam
Authorization: AWS AKIAIOSFODNN7EXAMPLE:ENoSbxYByFA0UGLZUqJN5EUnLDg=
Date: Wed, 20 Feb 2008 22:12:21 +0000
```

Die Signatur wurde aus den folgenden Informationen generiert.

```
PUT\r\n
\r\n
\r\n
Wed, 20 Feb 2008 22:12:21 +0000\r\n

x-amz-copy-source:/amzn-s3-demo-bucket1/flotsam\r\n
/amzn-s3-demo-bucket2/jetsam
```

Amazon S3 gibt die folgende Antwort zurück, die das ETag Objekt und das Datum der letzten Änderung angibt.

```
HTTP/1.1 200 OK
x-amz-id-2: Vyaxt7qEbzv34BnSu5hctyyNSlHTYZFMWK4FtzO+iX8JQNyaLdTshL0KxatbaOZt
x-amz-request-id: 6B13C3C5B34AF333
Date: Wed, 20 Feb 2008 22:13:01 +0000

Content-Type: application/xml
Transfer-Encoding: chunked
```

```
Connection: close
Server: AmazonS3
<?xml version="1.0" encoding="UTF-8"?>

<CopyObjectResult>
   <LastModified>2008-02-20T22:13:01</LastModified>
   <ETag>"7e9c608af58950deeb370c98608ed097"</ETag>
</CopyObjectResult>
```

Mit dem AWS CLI

Sie können auch das AWS Command Line Interface (AWS CLI) verwenden, um ein S3-Objekt zu kopieren. Weitere Informationen finden Sie unter copy-objectin der AWS CLI Befehlsreferenz.

Informationen zu dem AWS CLI finden Sie unter Was ist der AWS Command Line Interface? im AWS Command Line Interface Benutzerhandbuch.

So verschieben Sie ein Objekt

Verwenden Sie die folgenden Methoden, um ein Objekt zu verschieben.

Verwenden der S3-Konsole

> **Note**
>
> - Sie können ein Objekt verschieben, wenn Ihr Objekt weniger als 5 GB groß ist. Wenn Ihr Objekt größer als 5 GB ist, müssen Sie das AWS CLIoder verwenden, AWS SDKsum ein Objekt zu verschieben.
> - Eine Liste der zusätzlichen Berechtigungen, die zum Verschieben von Objekten erforderlich sind, finden Sie unterthe section called “Erforderliche Berechtigungen für API S3-Operationen”. Richtlinien, die diese Berechtigung gewähren, finden Sie beispielsweise unterthe section called “Beispiele für identitätsbasierte Richtlinien”.
> - Objekte, die mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) verschlüsselt wurden, können nicht mithilfe der Amazon S3 S3-Konsole verschoben werden. Verwenden Sie, oder Amazon S3 RESTAPI, um mit SSE -C verschlüsselte Objekte zu verschieben. AWS CLI AWS SDKs
> - Warten Sie beim Verschieben von Ordnern, bis der Verschiebevorgang abgeschlossen ist, bevor Sie weitere Änderungen an den Ordnern vornehmen.

- Sie können S3-Zugriffspunkt-Aliase nicht als Quelle oder Ziel für Move-Operationen in der Amazon S3 S3-Konsole verwenden.

So verschieben Sie ein Objekt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im Navigationsbereich Buckets und dann den Tab Allzweck-Buckets aus. Navigieren Sie zum Amazon-S3-Bucket oder -Ordner, der die Objekte enthält, die Sie verschieben möchten.
3. Aktivieren Sie das Kontrollkästchen für die Objekte, die Sie verschieben möchten.
4. Wählen Sie im Menü Aktionen die Option Verschieben aus.
5. Um den Zielpfad anzugeben, wählen Sie Browse S3, navigieren Sie zum Ziel und aktivieren Sie das Kontrollkästchen Ziel. Wählen Sie Choose destination (Ziel wählen).

   Geben Sie alternativ den Zielpfad ein.
6. Wenn Sie die Bucket-Versionierung nicht aktiviert haben, wird eine Warnung angezeigt, in der empfohlen wird, die Bucket-Versionierung zu aktivieren, um sich vor dem unbeabsichtigten Überschreiben oder Löschen von Objekten zu schützen. Wenn Sie alle Versionen von Objekten in diesem Bucket behalten möchten, wählen Sie Enable Bucket Versioning (Bucket-Versionierung aktivieren). Sie können die Standardeinstellungen für Verschlüsselung und Objektsperre auch in den Zieldetails einsehen.
7. Wählen Sie unter Zusätzliche Kopiereinstellungen aus, ob Sie Quelleinstellungen kopieren, Keine Einstellungen angeben oder Einstellungen angeben möchten. Quelleinstellungen kopieren ist die Standardoption. Wenn Sie das Objekt nur ohne die Quelleinstellungsattribute kopieren möchten, wählen Sie „Keine Einstellungen angeben“. Wählen Sie „Einstellungen angeben“, um Einstellungen für SpeicherklasseACLs, Objekt-Tags, Metadaten, serverseitige Verschlüsselung und zusätzliche Prüfsummen anzugeben.
8. Wählen Sie unten rechts Move (Verschieben) aus. Amazon S3 verschiebt Ihre Objekte in den Zielordner.

Unter Verwendung der AWS CLI

Sie können auch die AWS Command Line Interface (AWS CLI) verwenden, um ein S3-Objekt zu verschieben. Weitere Informationen finden Sie unter mvin der AWS CLI Befehlsreferenz.

Informationen zu dem AWS CLI finden Sie unter Was ist der AWS Command Line Interface? im AWS Command Line Interface Benutzerhandbuch.

Um ein Objekt umzubenennen

Gehen Sie wie folgt vor, um ein Objekt umzubenennen.

> **Note**
>
> - Sie können ein Objekt umbenennen, wenn Ihr Objekt weniger als 5 GB groß ist. Um Objekte umzubenennen, die größer als 5 GB sind, müssen Sie das AWS CLIoder verwenden, AWS SDKsum Ihr Objekt mit einem neuen Namen zu kopieren und dann das ursprüngliche Objekt zu löschen.
> - Eine Liste der zusätzlichen Berechtigungen, die zum Kopieren von Objekten erforderlich sind, finden Sie unterthe section called “Erforderliche Berechtigungen für API S3-Operationen”. Richtlinien, die diese Berechtigung gewähren, finden Sie beispielsweise unterthe section called “Beispiele für identitätsbasierte Richtlinien”.
> - Beim Umbenennen eines Objekts wird eine Kopie des Objekts mit einem neuen Datum der letzten Änderung erstellt. Anschließend wird dem ursprünglichen Objekt eine Löschmarke hinzugefügt.
> - Bucket-Einstellungen für die Standardverschlüsselung werden automatisch auf jedes angegebene unverschlüsselte Objekt angewendet.
> - Sie können die Amazon S3 S3-Konsole nicht verwenden, um Objekte mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) umzubenennen. Um mit SSE - C verschlüsselte Objekte umzubenennen, verwenden Sie AWS CLI AWS SDKs, oder Amazon S3, REST API um diese Objekte mit neuen Namen zu kopieren.
> - Wenn dieser Bucket die vom Bucket Owner erzwungene Einstellung für S3 Object Ownership verwendet, werden Objektzugriffskontrolllisten (ACLs) nicht kopiert.

Um ein Objekt umzubenennen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

2. Wählen Sie im Navigationsbereich Buckets und dann den Tab Allzweck-Buckets aus. Navigieren Sie zu dem Amazon S3 S3-Bucket oder -Ordner, der das Objekt enthält, das Sie umbenennen möchten.
3. Aktivieren Sie das Kontrollkästchen für das Objekt, das Sie umbenennen möchten.
4. Wählen Sie im Menü Aktionen die Option Objekt umbenennen.
5. Geben Sie im Feld Neuer Objektname den neuen Namen für das Objekt ein.
6. Wählen Sie unter Zusätzliche Kopiereinstellungen aus, ob Sie Quelleinstellungen kopieren, Keine Einstellungen angeben oder Einstellungen angeben möchten. Quelleinstellungen kopieren ist die Standardoption. Wenn Sie das Objekt nur ohne die Quelleinstellungsattribute kopieren möchten, wählen Sie „Keine Einstellungen angeben“. Wählen Sie „Einstellungen angeben“, um Einstellungen für SpeicherklasseACLs, Objekt-Tags, Metadaten, serverseitige Verschlüsselung und zusätzliche Prüfsummen anzugeben.
7. Wählen Sie Änderungen speichern. Amazon S3 benennt Ihr Objekt um.

Herunterladen von Objekten

In diesem Abschnitt wird erläutert, wie Sie Objekte aus einem S3-Bucket herunterladen. Mit Amazon S3 können Sie diese Objekte in einem oder mehreren Buckets speichern. Jedes Objekt kann eine Größe von bis zu 5 TB haben. Der Zugriff auf Amazon-S3-Objekte, die nicht archiviert sind, ist in Echtzeit möglich. Archivierte Objekte müssen jedoch wiederhergestellt werden, bevor sie heruntergeladen werden können. Weitere Informationen zum Herunterladen von archivierten Objekten finden Sie unter the section called “Herunterladen von archivierten Objekten”.

Sie können ein einzelnes Objekt mithilfe der Amazon S3-Konsole AWS Command Line Interface (AWS CLI) oder Amazon S3 herunterladen RESTAPI. AWS SDKs Verwenden Sie die S3-Konsole, um ein Objekt von S3 herunterzuladen, ohne Code schreiben oder Befehle ausführen zu müssen. Weitere Informationen finden Sie unter the section called “Herunterladen eines Objekts”.

Um mehrere Objekte herunterzuladen, verwenden Sie AWS CloudShell AWS CLI, die, oder AWS SDKs. Weitere Informationen finden Sie unter the section called “Herunterladen mehrerer Objekte”.

Wenn Sie einen Teil eines Objekts herunterladen müssen, verwenden Sie zusätzliche Parameter mit dem AWS CLI oder, REST API um nur die Bytes anzugeben, die Sie herunterladen möchten. Weitere Informationen finden Sie unter the section called “Herunterladen eines Teils eines Objekts”.

Wenn Sie ein Objekt herunterladen müssen, das Sie nicht besitzen, bitten Sie den Objekteigentümer, ein vorsigniertes Objekt zu generierenURL, mit dem Sie das Objekt herunterladen können. Weitere

Informationen finden Sie unter the section called “Herunterladen eines Objekts von einem anderen AWS-Konto”.

Wenn Sie Objekte außerhalb des AWS Netzwerks herunterladen, fallen Gebühren für die Datenübertragung an. Die Datenübertragung innerhalb des AWS Netzwerks ist innerhalb desselben kostenlos AWS-Region, für GET Anfragen werden Ihnen jedoch Gebühren berechnet. Weitere Informationen zu den Gebühren für Datenübertragungen und Datenabrufe finden Sie unter Preise für Amazon S3.

Themen

- Herunterladen eines Objekts
- Herunterladen mehrerer Objekte
- Herunterladen eines Teils eines Objekts
- Herunterladen eines Objekts von einem anderen AWS-Konto
- Herunterladen von archivierten Objekten
- Objekte auf der Grundlage von Metadaten herunterladen
- Fehlerbehebung beim Herunterladen von Objekten

Herunterladen eines Objekts

Sie können ein Objekt mithilfe der Amazon S3 S3-Konsole, AWS CLI AWS SDKs, oder herunterladen RESTAPI.

Verwenden der S3-Konsole

In diesem Abschnitt erfahren Sie, wie Sie mit der Amazon-S3-Konsole ein Objekt aus einem S3-Bucket herunterladen.

> **Note**
>
> - Sie können jeweils nur ein Objekt herunterladen.
> - Wenn Sie über die Amazon-S3-Konsole ein Objekt herunterladen, dessen Schlüsselname mit Punkt (.) endet, wird der Punkt aus dem Schlüsselnamen des heruntergeladenen Objekts entfernt. Um den Punkt am Ende des Namens des heruntergeladenen Objekts beizubehalten, müssen Sie AWS Command Line Interface (AWS CLI) AWS SDKs, oder Amazon S3 verwenden RESTAPI.

Ein Objekt von einem S3-Bucket herunterladen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, aus dem Sie ein Objekt herunterladen möchten.
3. Sie können ein Objekt wie folgt aus einem S3-Bucket herunterladen:
   - Aktivieren Sie das Kontrollkästchen neben dem Objekt und wählen Sie Herunterladen. Wenn Sie das Objekt in einen spezifischen Ordner herunterladen möchten, wählen Sie im Menü Aktionen die Option Herunterladen als.
   - Wenn Sie eine spezifische Version des Objekts herunterladen möchten, aktivieren Sie die Schaltfläche Versionen anzeigen (neben dem Suchfeld). Aktivieren Sie das Kontrollkästchen neben der gewünschten Version des Objekts und wählen Sie Herunterladen. Wenn Sie das Objekt in einen spezifischen Ordner herunterladen möchten, wählen Sie im Menü Aktionen die Option Herunterladen als.

Verwenden Sie den AWS CLI

Das folgende Beispiel für `get-object` veranschaulicht, wie Sie die AWS CLI verwenden können, um ein Objekt von Amazon S3 herunterzuladen. Mit diesem Befehl wird das Objekt *`folder/my_image`* aus dem Bucket *`amzn-s3-demo-bucket1`* abgerufen. Das Objekt wird in eine Datei mit dem Namen *`my_downloaded_image`* heruntergeladen.

```
aws s3api get-object --bucket amzn-s3-demo-bucket1 --key folder/
my_image my_downloaded_image
```

Weitere Informationen und Beispiele finden Sie unter get-objectin der AWS CLI Befehlsreferenz.

Mit dem AWS SDKs

Beispiele dafür, wie Sie ein Objekt mit dem herunterladen können AWS SDKs, finden Sie in den Codebeispielen in der Amazon S3 API S3-Referenz.

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS SDKs in der Amazon S3 API S3-Referenz.

Mit dem REST API

Sie können den verwenden RESTAPI, um Objekte aus Amazon S3 abzurufen. Weitere Informationen finden Sie unter GetObjectin der Amazon Simple Storage Service API Reference.

Herunterladen mehrerer Objekte

Sie können mehrere Objekte herunterladen AWS CloudShell, indem Sie AWS CLI, oder den verwenden AWS SDKs.

Verwenden Sie AWS CloudShell in der AWS Management Console

AWS CloudShell ist eine browserbasierte, vorauthentifizierte Shell, die Sie direkt von der aus starten können. AWS Management Console

Weitere Informationen zu finden Sie unter AWS CloudShell Was ist? CloudShell im AWS CloudShell Benutzerhandbuch.

> **Important**
>
> Mit AWS CloudShell, Ihr Home-Verzeichnis hat Speicherplatz von bis zu 1 GB pro. AWS-Region Daher können Sie keine Buckets mit Objekten synchronisieren, deren Gesamtwert diesen Wert überschreitet. Weitere Einschränkungen finden Sie unter Service quotas and restrictions im Benutzerhandbuch zu AWS CloudShell .

Um Objekte herunterzuladen, verwenden Sie AWS CloudShell

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die CloudShell Konsole unter https://console.aws.amazon.com/cloudshell/.
2. Führen Sie den folgenden Befehl aus, um Objekte in Ihrem Bucket mit zu synchronisieren CloudShell. Der folgende Befehl synchronisiert Objekte aus dem Bucket mit dem Namen *amzn-s3-demo-bucket1* und erstellt einen Ordner mit dem Namen *temp* in CloudShell. CloudShell synchronisiert Ihre Objekte mit diesem Ordner. Zur Verwendung dieses Befehls ersetzen Sie *user input placeholders* durch eigene Informationen.

```
aws s3 sync s3://amzn-s3-demo-bucket1 ./temp
```

> **Note**
>
> Wenn Sie einen Musterabgleich durchführen möchten, um bestimmte Objekte aus- oder einzuschließen, können Sie die Parameter `--exclude "value"` und `--include "value"` mit dem `sync`-Befehl verwenden.

3. Führen Sie den folgenden Befehl aus, um Ihre Objekte im Ordner *temp* in eine Datei mit dem Namen *temp.zip* zu komprimieren.

```
zip temp.zip -r temp/
```

4. Wählen Sie Aktionen und anschließend Datei herunterladen aus.
5. Geben Sie den Dateinamen **temp.zip** ein und wählen Sie anschließend Herunterladen aus.
6. (Optional) Löschen Sie die *temp.zip* Datei und die Objekte, die mit dem *temp* Ordner in synchronisiert wurden. CloudShell Mit AWS CloudShell haben Sie persistenten Speicher von bis zu 1 GB pro AWS-Region.

   Sie können den folgenden Beispielbefehl verwenden, um die `.zip`-Datei und den Ordner zu löschen. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
rm temp.zip && rm -rf temp/
```

Mit dem AWS CLI

Das folgende Beispiel zeigt, wie Sie mit AWS CLI dem alle Dateien oder Objekte unter dem angegebenen Verzeichnis oder Präfix herunterladen können. Mit diesem Befehl werden alle Objekte aus dem Bucket *amzn-s3-demo-bucket1* in Ihr aktuelles Verzeichnis kopiert. Wenn Sie diesen Beispielbefehl verwenden möchten, verwenden Sie anstelle von *amzn-s3-demo-bucket1* den Namen Ihres Buckets.

```
aws s3 cp s3://amzn-s3-demo-bucket1 . --recursive
```

Mit dem folgenden Befehl werden alle Objekte unter dem Präfix *logs* im Bucket *amzn-s3-demo-bucket1* in Ihr aktuelles Verzeichnis heruntergeladen. Darüber hinaus werden die Parameter `--exclude` und `--include` verwendet, damit nur Objekte mit dem Suffix *.log* kopiert werden. Wenn

Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3 cp s3://amzn-s3-demo-bucket1/logs/ . --recursive --exclude "*" --include "*.log"
```

Weitere Informationen und Beispiele finden Sie unter cpin der AWS CLI Befehlsreferenz.

Mit dem AWS SDKs

Beispiele dafür, wie Sie alle Objekte in einem Amazon S3 S3-Bucket mit dem herunterladen können AWS SDKs, finden Sie in den Codebeispielen in der Amazon S3 API S3-Referenz.

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS SDKs in der Amazon S3 API S3-Referenz.

Herunterladen eines Teils eines Objekts

Sie können einen Teil eines Objekts herunterladen, indem Sie das AWS CLI oder verwenden RESTAPI. Dazu geben Sie mithilfe zusätzlicher Parameter an, welcher Teil eines Objekts heruntergeladen werden soll.

Mit dem AWS CLI

Der folgende Beispielbefehl führt eine GET-Anforderung für einen Bytebereich im Objekt *folder/my_data* im Bucket *amzn-s3-demo-bucket1* aus. In der Anforderung muss dem Bytebereich das Präfix `bytes=` vorangestellt werden. Das Teilobjekt wird in die Ausgabedatei mit dem Namen *my_data_range* heruntergeladen. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3api get-object --bucket amzn-s3-demo-bucket1 --key folder/my_data --range
 bytes=0-500 my_data_range
```

Weitere Informationen und Beispiele finden Sie unter get-objectin der AWS CLI Befehlsreferenz.

Weitere Informationen zum HTTP `Range` Header finden Sie unter RFC9110 auf der RFC Editor-Website.

> **Note**
>
> Amazon S3 unterstützt nicht das Abrufen mehrerer Datenbereiche in einer einzelnen GET-Anforderung.

Unter Verwendung des REST API

Sie können die `Range` Parameter `partNumber` und in verwenden RESTAPI, um Objektteile aus Amazon S3 abzurufen. Weitere Informationen finden Sie unter GetObjectin der Amazon Simple Storage Service API Reference.

Herunterladen eines Objekts von einem anderen AWS-Konto

Sie können einen vorsignierten Code verwendenURL, um anderen zeitlich begrenzten Zugriff auf Ihre Objekte zu gewähren, ohne Ihre Bucket-Richtlinie zu aktualisieren.

Das Vorsignierte URL kann in einem Browser eingegeben oder von einem Programm zum Herunterladen eines Objekts verwendet werden. Die von der verwendeten Anmeldeinformationen URL sind die des AWS Benutzers, der die URL generiert hat. Nachdem URL das erstellt wurde, URL kann jeder, der das vorsignierte Objekt besitzt, das entsprechende Objekt herunterladen, bis es URL abläuft.

Verwenden eines in der S3-Konsole URL vorsignierten

Sie können die Amazon S3 S3-Konsole verwenden, um ein Objekt zu generieren, das URL für die gemeinsame Nutzung vorsigniert ist, indem Sie die folgenden Schritte ausführen. Wenn Sie die Konsole verwenden, beträgt die maximale Ablaufzeit für ein vorsigniertes URL Objekt 12 Stunden ab dem Zeitpunkt der Erstellung.

Um mit der Amazon S3 S3-Konsole ein URL vorsigniertes zu generieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Buckets-Liste den Namen des Buckets aus, der das Objekt enthält, für das Sie eine URL Vorsignierung wünschen.
4. Wählen Sie in der Objektliste das Objekt aus, für das Sie ein URL vorsigniertes Objekt erstellen möchten.

5. Wählen Sie im Menü „Objektaktionen“ die Option „Mit einem URL vorsignierten Objekt teilen“.
6. Geben Sie an, wie lange das Vorsignierte gültig sein URL soll.
7. Wählen Sie „URLVorsigniertes erstellen“.
8. Wenn eine Bestätigungsmeldung angezeigt wird, wird diese automatisch in Ihre Zwischenablage kopiert. URL Sie werden eine Schaltfläche sehen, mit der Sie das Vorsignierte kopieren könnenURL, falls Sie es erneut kopieren müssen.
9. Um das Objekt herunterzuladen, fügen Sie es URL in einen beliebigen Browser ein. Das Objekt versucht dann, es herunterzuladen.

Weitere Hinweise zu vorsignierten URLs und anderen Methoden zu ihrer Erstellung finden Sie unterLaden Sie Objekte mit vorsignierten Objekten herunter und laden Sie sie hoch URLs.

Herunterladen von archivierten Objekten

Sie können Objekte, auf die selten zugegriffen wird, archivieren, um Ihre Speicherkosten dafür zu senken. Wenn Sie ein Objekt archivieren, wird es in einen kostengünstigen Speicher verschoben. Sie können also nicht in Echtzeit darauf zugreifen. Wenn Sie ein archiviertes Objekt herunterladen möchten, müssen Sie es zunächst wiederherstellen.

Je nach Speicherklasse können archivierte Objekte innerhalb von Minuten oder Stunden wiederhergestellt werden. Sie können ein archiviertes Objekt mithilfe der Amazon S3-Konsole, S3 Batch Operations, Amazon S3 REST API AWS SDKs, The und AWS Command Line Interface (AWS CLI) wiederherstellen.

Detaillierte Anweisungen finden Sie unter Wiederherstellen eines archivierten Objekts. Nachdem Sie das archivierte Objekt wiederhergestellt haben, können Sie es herunterladen.

Objekte auf der Grundlage von Metadaten herunterladen

Sie können Vorbedingungen hinzufügen, um ein Objekt anhand seiner Metadaten mithilfe einer bedingten Leseanforderung herunterzuladen. Sie können ein Objekt auf der Grundlage seines Entity-Tags (ETag) oder des Datums der letzten Änderung zurückgeben. Dies kann eine S3-Operation auf Objekte beschränken, die seit einem bestimmten Datum aktualisiert wurden, oder nur eine bestimmte Objektversion zurückgeben.

Sie können bedingte Schreibvorgänge verwenden für GetObject oder HeadObjectAnfragen.

Weitere Informationen zu bedingten Anfragen finden Sie unterFügen Sie Vorbedingungen zu S3-Vorgängen mit bedingten Anfragen hinzu.

Fehlerbehebung beim Herunterladen von Objekten

Unzureichende Berechtigungen oder falsche Bucket- oder AWS Identity and Access Management (IAM) Benutzerrichtlinien können zu Fehlern führen, wenn Sie versuchen, Objekte von Amazon S3 herunterzuladen. Diese Probleme können häufig den Fehler Zugriff verweigert (403 Verboten) hervorrufen. Dies bedeutet, dass Amazon S3 den Zugriff auf eine Ressource nicht gewähren kann.

Weitere Informationen zu den häufigsten Ursachen für den Fehler Zugriff verweigert (403 Verboten) finden Sie unter Problembehandlung bei verweigertem Zugriff beheben (403 Forbidden) Fehler in Amazon S3.

Überprüfung der Objektintegrität

Amazon S3 verwendet Prüfsummenwerte, um die Integrität von Daten zu verifizieren, die Sie auf Amazon S3 hochladen oder von Amazon S3 herunterladen. Darüber hinaus können Sie anfordern, dass ein weiterer Prüfsummenwert für jedes Objekt berechnet wird, das Sie in Amazon S3 speichern. Sie können einen von mehreren Prüfsummenalgorithmen auswählen, der beim Hochladen oder Kopieren Ihrer Daten verwendet werden soll. Amazon S3 berechnet mit diesem Algorithmus einen zusätzlichen Prüfsummenwert und speichert ihn als Teil der Objektmetadaten. Weitere Informationen zur Verwendung zusätzlicher Prüfsummen zur Überprüfung der Datenintegrität finden Sie im Tutorial: Überprüfen der Integrität von Daten in Amazon S3 mit zusätzlichen Prüfsummen.

Wenn Sie ein Objekt hochladen, können Sie optional eine vorberechnete Prüfsumme als Teil Ihrer Anforderung aufnehmen. Amazon S3 vergleicht die bereitgestellte Prüfsumme mit der Prüfsumme, die mit Ihrem angegebenen Algorithmus berechnet wird. Wenn die beiden Werte nicht übereinstimmen, meldet Amazon S3 einen Fehler.

Verwenden unterstützter Prüfsummenalgorithmen

Amazon S3 bietet Ihnen die Möglichkeit, den Prüfsummenalgorithmus auszuwählen, der zur Validierung Ihrer Daten beim Hoch- oder Herunterladen verwendet wird. Sie können einen der folgenden Algorithmen zur Prüfung der Datenintegrität zwischen Secure Hash Algorithms (SHA) oder Cyclic Redundancy Check (CRC) wählen:

- CRC-32

- CRC-32 °C
- SHA-1
- SHA-256

Wenn Sie ein Objekt hochladen, können Sie den Algorithmus angeben, den Sie verwenden möchten:

- Wenn Sie den verwenden AWS Management Console, wählen Sie den Prüfsummenalgorithmus aus, den Sie verwenden möchten. In diesem Fall können Sie den Prüfsummenwert des Objekts optional angeben. Wenn Amazon S3 das Objekt erhält, berechnet es die Prüfsumme mithilfe des von Ihnen angegebenen Algorithmus. Wenn die beiden Prüfsummenwerte nicht übereinstimmen, generiert Amazon S3 einen Fehler.
- Wenn Sie einen verwenden SDK, können Sie den Wert des `ChecksumAlgorithm` Parameters auf den Algorithmus setzen, den Amazon S3 bei der Berechnung der Prüfsumme verwenden soll. Amazon S3 berechnet den Prüfsummenwert automatisch.
- Wenn Sie den verwenden REST API, verwenden Sie den `x-amz-sdk-checksum-algorithm` Parameter nicht. Stattdessen verwenden Sie einen der algorithmenspezifischen Header (z. B. `x-amz-checksum-crc32`).

Weitere Informationen zum Hochladen von Objekten finden Sie unter Objekte hochladen.

Um einen dieser Prüfsummenwerte auf Objekte anzuwenden, die bereits auf Amazon S3 hochgeladen wurden, können Sie das Objekt kopieren. Wenn Sie ein Objekt kopieren, können Sie angeben, ob Sie den vorhandenen oder einen neuen Prüfsummenalgorithmus verwenden möchten. Sie können einen Prüfsummenalgorithmus angeben, wenn Sie eine unterstützte Methode zum Kopieren von Objekten verwenden, einschließlich S3-Batch-Operationen. Weitere Informationen über S3-Batch-Vorgänge finden Sie unter Massenausführung von Objektoperationen mit Batch Operations.

> **Important**
>
> Wenn Sie einen mehrteiligen Upload mit zusätzlichen Prüfsummen verwenden, müssen die mehrteiligen Teilenummern aufeinander folgende Teilenummern sein. Wenn Sie zusätzliche Prüfsummen verwenden und versuchen, eine mehrteilige Upload-Anfrage mit nicht aufeinanderfolgenden Artikelnummern abzuschließen, generiert Amazon S3 einen Fehler. HTTP `500 Internal Server Error`

Nach dem Hochladen von Objekten können Sie den Prüfsummenwert abrufen und mit einem vorberechneten oder zuvor gespeicherten Prüfsummenwert vergleichen, der mit demselben Algorithmus berechnet wurde.

Verwenden der S3-Konsole

Weitere Informationen zur Verwendung der Konsole und zum Angeben von Prüfsummenalgorithmen, die beim Hochladen von Objekten verwendet werden, finden Sie unter Objekte hochladen und unter Tutorial: Überprüfen der Integrität von Daten in Amazon S3 mit zusätzlichen Prüfsummen.

Unter Verwendung der AWS SDKs

Das folgende Beispiel zeigt, wie Sie mit dem AWS SDKs eine große Datei mit mehrteiligem Upload hochladen, eine große Datei herunterladen und eine mehrteilige Upload-Datei validieren können, wobei Sie jeweils SHA -256 für die Dateiüberprüfung verwenden.

Java

Example Beispiel: Eine große Datei mit -256 hochladen, herunterladen und verifizieren SHA

Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
import software.amazon.awssdk.auth.credentials.AwsCredentials;
import software.amazon.awssdk.auth.credentials.AwsCredentialsProvider;
import software.amazon.awssdk.core.ResponseInputStream;
import software.amazon.awssdk.core.sync.RequestBody;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.AbortMultipartUploadRequest;
import software.amazon.awssdk.services.s3.model.ChecksumAlgorithm;
import software.amazon.awssdk.services.s3.model.ChecksumMode;
import software.amazon.awssdk.services.s3.model.CompleteMultipartUploadRequest;
import software.amazon.awssdk.services.s3.model.CompleteMultipartUploadResponse;
import software.amazon.awssdk.services.s3.model.CompletedMultipartUpload;
import software.amazon.awssdk.services.s3.model.CompletedPart;
import software.amazon.awssdk.services.s3.model.CreateMultipartUploadRequest;
import software.amazon.awssdk.services.s3.model.CreateMultipartUploadResponse;
import software.amazon.awssdk.services.s3.model.GetObjectAttributesRequest;
import software.amazon.awssdk.services.s3.model.GetObjectAttributesResponse;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.GetObjectTaggingRequest;
```

```
    import software.amazon.awssdk.services.s3.model.ObjectAttributes;
    import software.amazon.awssdk.services.s3.model.PutObjectTaggingRequest;
    import software.amazon.awssdk.services.s3.model.Tag;
    import software.amazon.awssdk.services.s3.model.Tagging;
    import software.amazon.awssdk.services.s3.model.UploadPartRequest;
    import software.amazon.awssdk.services.s3.model.UploadPartResponse;

    import java.io.File;
    import java.io.FileInputStream;
    import java.io.FileOutputStream;
    import java.io.IOException;
    import java.io.InputStream;
    import java.io.OutputStream;
    import java.nio.ByteBuffer;
    import java.security.MessageDigest;
    import java.security.NoSuchAlgorithmException;
    import java.util.ArrayList;
    import java.util.Base64;
    import java.util.List;

    public class LargeObjectValidation {
        private static String FILE_NAME = "sample.file";
        private static String BUCKET = "sample-bucket";
        //Optional, if you want a method of storing the full multipart object
 checksum in S3.
        private static String CHECKSUM_TAG_KEYNAME = "fullObjectChecksum";
        //If you have existing full-object checksums that you need to validate
 against, you can do the full object validation on a sequential upload.
        private static String SHA256_FILE_BYTES = "htCM5g7ZNdoSw8bN/
mkgiAhXt5MFoVowVg+LE9aIQmI=";
        //Example Chunk Size - this must be greater than or equal to 5MB.
        private static int CHUNK_SIZE = 5 * 1024 * 1024;

        public static void main(String[] args) {
            S3Client s3Client = S3Client.builder()
                    .region(Region.US_EAST_1)
                    .credentialsProvider(new AwsCredentialsProvider() {
                        @Override
                        public AwsCredentials resolveCredentials() {
                            return new AwsCredentials() {
                                @Override
                                public String accessKeyId() {
                                    return Constants.ACCESS_KEY;
                                }
```

```
                            @Override
                            public String secretAccessKey() {
                                return Constants.SECRET;
                            }
                        };
                    }
                })
                .build();
        uploadLargeFileBracketedByChecksum(s3Client);
        downloadLargeFileBracketedByChecksum(s3Client);
        validateExistingFileAgainstS3Checksum(s3Client);
    }

    public static void uploadLargeFileBracketedByChecksum(S3Client s3Client) {
        System.out.println("Starting uploading file validation");
        File file = new File(FILE_NAME);
        try (InputStream in = new FileInputStream(file)) {
            MessageDigest sha256 = MessageDigest.getInstance("SHA-256");
            CreateMultipartUploadRequest createMultipartUploadRequest =
CreateMultipartUploadRequest.builder()
                    .bucket(BUCKET)
                    .key(FILE_NAME)
                    .checksumAlgorithm(ChecksumAlgorithm.SHA256)
                    .build();
            CreateMultipartUploadResponse createdUpload =
s3Client.createMultipartUpload(createMultipartUploadRequest);
            List<CompletedPart> completedParts = new ArrayList<CompletedPart>();
            int partNumber = 1;
            byte[] buffer = new byte[CHUNK_SIZE];
            int read = in.read(buffer);
            while (read != -1) {
                UploadPartRequest uploadPartRequest =
UploadPartRequest.builder()

.partNumber(partNumber).uploadId(createdUpload.uploadId()).key(FILE_NAME).bucket(BUCKET).ch
                UploadPartResponse uploadedPart =
s3Client.uploadPart(uploadPartRequest,
RequestBody.fromByteBuffer(ByteBuffer.wrap(buffer, 0, read)));
                CompletedPart part =
CompletedPart.builder().partNumber(partNumber).checksumSHA256(uploadedPart.checksumSHA256()
                completedParts.add(part);
                sha256.update(buffer, 0, read);
                read = in.read(buffer);
```

```
                    partNumber++;
                }
                String fullObjectChecksum =
Base64.getEncoder().encodeToString(sha256.digest());
                if (!fullObjectChecksum.equals(SHA256_FILE_BYTES)) {
                    //Because the SHA256 is uploaded after the part is uploaded; the
upload is bracketed and the full object can be fully validated.

s3Client.abortMultipartUpload(AbortMultipartUploadRequest.builder().bucket(BUCKET).key(FILE
                    throw new IOException("Byte mismatch between stored checksum and
upload, do not proceed with upload and cleanup");
                }
                CompletedMultipartUpload completedMultipartUpload =
CompletedMultipartUpload.builder().parts(completedParts).build();
                CompleteMultipartUploadResponse completedUploadResponse =
s3Client.completeMultipartUpload(

CompleteMultipartUploadRequest.builder().bucket(BUCKET).key(FILE_NAME).uploadId(createdUplo
                Tag checksumTag =
Tag.builder().key(CHECKSUM_TAG_KEYNAME).value(fullObjectChecksum).build();
                //Optionally, if you need the full object checksum stored with the
file; you could add it as a tag after completion.

s3Client.putObjectTagging(PutObjectTaggingRequest.builder().bucket(BUCKET).key(FILE_NAME).t
            } catch (IOException | NoSuchAlgorithmException e) {
                e.printStackTrace();
            }
            GetObjectAttributesResponse
                    objectAttributes =
s3Client.getObjectAttributes(GetObjectAttributesRequest.builder().bucket(BUCKET).key(FILE_N
                    .objectAttributes(ObjectAttributes.OBJECT_PARTS,
ObjectAttributes.CHECKSUM).build());
            System.out.println(objectAttributes.objectParts().parts());
            System.out.println(objectAttributes.checksum().checksumSHA256());
        }

        public static void downloadLargeFileBracketedByChecksum(S3Client s3Client) {
            System.out.println("Starting downloading file validation");
            File file = new File("DOWNLOADED_" + FILE_NAME);
            try (OutputStream out = new FileOutputStream(file)) {
                GetObjectAttributesResponse
                        objectAttributes =
s3Client.getObjectAttributes(GetObjectAttributesRequest.builder().bucket(BUCKET).key(FILE_N
```

```
                        .objectAttributes(ObjectAttributes.OBJECT_PARTS,
ObjectAttributes.CHECKSUM).build());
                //Optionally if you need the full object checksum, you can grab a
tag you added on the upload
                List<Tag> objectTags =
s3Client.getObjectTagging(GetObjectTaggingRequest.builder().bucket(BUCKET).key(FILE_NAME).b
                String fullObjectChecksum = null;
                for (Tag objectTag : objectTags) {
                    if (objectTag.key().equals(CHECKSUM_TAG_KEYNAME)) {
                        fullObjectChecksum = objectTag.value();
                        break;
                    }
                }
                MessageDigest sha256FullObject =
MessageDigest.getInstance("SHA-256");
                MessageDigest sha256ChecksumOfChecksums =
MessageDigest.getInstance("SHA-256");

                //If you retrieve the object in parts, and set the ChecksumMode to
enabled, the SDK will automatically validate the part checksum
                for (int partNumber = 1; partNumber <=
objectAttributes.objectParts().totalPartsCount(); partNumber++) {
                    MessageDigest sha256Part = MessageDigest.getInstance("SHA-256");
                    ResponseInputStream<GetObjectResponse> response =
s3Client.getObject(GetObjectRequest.builder().bucket(BUCKET).key(FILE_NAME).partNumber(part
                    GetObjectResponse getObjectResponse = response.response();
                    byte[] buffer = new byte[CHUNK_SIZE];
                    int read = response.read(buffer);
                    while (read != -1) {
                        out.write(buffer, 0, read);
                        sha256FullObject.update(buffer, 0, read);
                        sha256Part.update(buffer, 0, read);
                        read = response.read(buffer);
                    }
                    byte[] sha256PartBytes = sha256Part.digest();
                    sha256ChecksumOfChecksums.update(sha256PartBytes);
                    //Optionally, you can do an additional manual validation again
the part checksum if needed in addition to the SDK check
                    String base64PartChecksum =
Base64.getEncoder().encodeToString(sha256PartBytes);
                    String base64PartChecksumFromObjectAttributes =
objectAttributes.objectParts().parts().get(partNumber - 1).checksumSHA256();
```

```
                    if (!
base64PartChecksum.equals(getObjectResponse.checksumSHA256()) || !
base64PartChecksum.equals(base64PartChecksumFromObjectAttributes)) {
                        throw new IOException("Part checksum didn't match for the
 part");
                    }
                    System.out.println(partNumber + " " + base64PartChecksum);
                }
                //Before finalizing, do the final checksum validation.
                String base64FullObject =
 Base64.getEncoder().encodeToString(sha256FullObject.digest());
                String base64ChecksumOfChecksums =
 Base64.getEncoder().encodeToString(sha256ChecksumOfChecksums.digest());
                if (fullObjectChecksum != null && !
fullObjectChecksum.equals(base64FullObject)) {
                    throw new IOException("Failed checksum validation for full
 object");
                }
                System.out.println(fullObjectChecksum);
                String base64ChecksumOfChecksumFromAttributes =
 objectAttributes.checksum().checksumSHA256();
                if (base64ChecksumOfChecksumFromAttributes != null && !
base64ChecksumOfChecksums.equals(base64ChecksumOfChecksumFromAttributes)) {
                    throw new IOException("Failed checksum validation for full
 object checksum of checksums");
                }
                System.out.println(base64ChecksumOfChecksumFromAttributes);
                out.flush();
            } catch (IOException | NoSuchAlgorithmException e) {
                //Cleanup bad file
                file.delete();
                e.printStackTrace();
            }
        }

        public static void validateExistingFileAgainstS3Checksum(S3Client s3Client)
 {
            System.out.println("Starting existing file validation");
            File file = new File("DOWNLOADED_" + FILE_NAME);
            GetObjectAttributesResponse
                    objectAttributes =
 s3Client.getObjectAttributes(GetObjectAttributesRequest.builder().bucket(BUCKET).key(FILE_N
                    .objectAttributes(ObjectAttributes.OBJECT_PARTS,
 ObjectAttributes.CHECKSUM).build());
```

```
            try (InputStream in = new FileInputStream(file)) {
                MessageDigest sha256ChecksumOfChecksums =
 MessageDigest.getInstance("SHA-256");
                MessageDigest sha256Part = MessageDigest.getInstance("SHA-256");
                byte[] buffer = new byte[CHUNK_SIZE];
                int currentPart = 0;
                int partBreak =
 objectAttributes.objectParts().parts().get(currentPart).size();
                int totalRead = 0;
                int read = in.read(buffer);
                while (read != -1) {
                    totalRead += read;
                    if (totalRead >= partBreak) {
                        int difference = totalRead - partBreak;
                        byte[] partChecksum;
                        if (totalRead != partBreak) {
                            sha256Part.update(buffer, 0, read - difference);
                            partChecksum = sha256Part.digest();
                            sha256ChecksumOfChecksums.update(partChecksum);
                            sha256Part.reset();
                            sha256Part.update(buffer, read - difference,
 difference);
                        } else {
                            sha256Part.update(buffer, 0, read);
                            partChecksum = sha256Part.digest();
                            sha256ChecksumOfChecksums.update(partChecksum);
                            sha256Part.reset();
                        }
                        String base64PartChecksum =
 Base64.getEncoder().encodeToString(partChecksum);
                        if (!
base64PartChecksum.equals(objectAttributes.objectParts().parts().get(currentPart).checksumSH
 {
                            throw new IOException("Part checksum didn't match S3");
                        }
                        currentPart++;
                        System.out.println(currentPart + " " + base64PartChecksum);
                        if (currentPart <
 objectAttributes.objectParts().totalPartsCount()) {
                            partBreak +=
 objectAttributes.objectParts().parts().get(currentPart - 1).size();
                        }
                    } else {
                        sha256Part.update(buffer, 0, read);
```

```
                    }
                    read = in.read(buffer);
                }
                if (currentPart != objectAttributes.objectParts().totalPartsCount())
 {
                    currentPart++;
                    byte[] partChecksum = sha256Part.digest();
                    sha256ChecksumOfChecksums.update(partChecksum);
                    String base64PartChecksum =
 Base64.getEncoder().encodeToString(partChecksum);
                    System.out.println(currentPart + " " + base64PartChecksum);
                }

                String base64CalculatedChecksumOfChecksums =
 Base64.getEncoder().encodeToString(sha256ChecksumOfChecksums.digest());
                System.out.println(base64CalculatedChecksumOfChecksums);
                System.out.println(objectAttributes.checksum().checksumSHA256());
                if (!
base64CalculatedChecksumOfChecksums.equals(objectAttributes.checksum().checksumSHA256()))
 {
                    throw new IOException("Full object checksum of checksums don't
 match S3");
                }

            } catch (IOException | NoSuchAlgorithmException e) {
                e.printStackTrace();
            }
        }
    }
```

Mit dem REST API

Sie können REST Anfragen zum Hochladen eines Objekts mit einem Prüfsummenwert senden, um die Integrität der Daten mit PutObjectzu überprüfen. Sie können den Prüfsummenwert für Objekte auch mit GetObjectoder abrufen. HeadObject

Mit dem AWS CLI

Sie können eine PUT-Anforderung zum Hochladen eines Objekts von bis zu 5 GB in einem einzigen Vorgang senden. Weitere Informationen finden Sie unter `PutObject` in der AWS CLI -Befehlszeilenreferenz. Sie können auch `get-object` und `head-object` verwenden, um die

Prüfsumme eines bereits hochgeladenen Objekts abzurufen und die Integrität der Daten zu überprüfen.

Weitere Informationen finden Sie unter Amazon S3 CLI FAQ im AWS Command Line Interface Benutzerhandbuch.

Inhalte verwenden — MD5 beim Hochladen von Objekten

Eine weitere Möglichkeit, die Integrität Ihres Objekts nach dem Hochladen zu überprüfen, besteht darin, MD5 beim Hochladen eine Zusammenfassung des Objekts bereitzustellen. Wenn Sie den MD5 Digest für Ihr Objekt berechnen, können Sie den Digest mit dem PUT Befehl bereitstellen, indem Sie den Header verwenden. `Content-MD5`

Nach dem Hochladen des Objekts berechnet Amazon S3 den MD5 Digest des Objekts und vergleicht ihn mit dem von Ihnen angegebenen Wert. Die Anforderung ist nur erfolgreich, wenn die beiden Digests übereinstimmen.

Die Angabe eines MD5 Digest ist nicht erforderlich, Sie können ihn jedoch verwenden, um die Integrität des Objekts im Rahmen des Upload-Vorgangs zu überprüfen.

Verwenden Sie Content- MD5 und the, um hochgeladene ETag Objekte zu verifizieren

Das Entity-Tag (ETag) für ein Objekt steht für eine bestimmte Version dieses Objekts. Beachten Sie, dass das nur Änderungen am Inhalt eines Objekts ETag widerspiegelt, nicht an seinen Metadaten. Wenn sich nur die Metadaten eines Objekts ändern, ETag bleiben die gleichen.

Je nach Objekt kann es sich ETag bei dem Objekt um eine MD5 Zusammenfassung der Objektdaten handeln:

- Wenn ein Objekt durch die `CopyObject` Operation`PutObject`,`PostObject`, oder durch die AWS Management Console, erstellt wird und dieses Objekt auch Klartext oder durch serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt ist, hat ETag dieses Objekt eine MD5 Zusammenfassung seiner Objektdaten.
- Wenn ein Objekt durch die `CopyObject` Operation`PutObject`,`PostObject`, oder durch die AWS Management Console, erstellt wird und dieses Objekt durch serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) oder serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln (SSE-KMS) verschlüsselt wird, hat dieses Objekt eine, ETag die keine MD5 Zusammenfassung seiner Objektdaten ist.

- Wenn ein Objekt entweder durch den mehrteiligen Upload-Prozess oder durch den Vorgang erstellt wird, ETag handelt es sich bei dem Objekt unabhängig von der `UploadPartCopy` Verschlüsselungsmethode nicht um einen MD5 Digest. Wenn ein Objekt größer als 16 MB ist, wird dieses Objekt als mehrteiliger AWS Management Console Upload hochgeladen oder kopiert, es handelt sich also ETag nicht um einen Digest. MD5

Bei Objekten, bei denen ETag es sich um den `Content-MD5` Digest des Objekts handelt, können Sie den ETag Wert des Objekts mit einem berechneten oder zuvor gespeicherten Digest vergleichen. `Content-MD5`

Verwenden von nachfolgenden Prüfsummen

Wenn Sie Objekte auf Amazon S3 hochladen, können Sie entweder eine vorberechnete Prüfsumme für das Objekt angeben oder eine verwenden, AWS SDK um automatisch abschließende Prüfsummen in Ihrem Namen zu erstellen. Wenn Sie sich für eine nachfolgende Prüfsumme entscheiden, generiert Amazon S3 die Prüfsumme automatisch mithilfe des von Ihnen angegebenen Algorithmus und überprüft damit die Integrität des Objekts während des Uploads.

Um bei der Verwendung von eine eine abschließende Prüfsumme zu erstellen AWS SDK, füllen Sie den Parameter mit Ihrem bevorzugten Algorithmus. `ChecksumAlgorithm` Der berechnet SDK anhand dieses Algorithmus die Prüfsumme für Ihr Objekt (oder Objektteile) und hängt sie automatisch an das Ende Ihrer Upload-Anfrage an. Dank dieser Funktionsweise sparen Sie Zeit, da Amazon S3 sowohl die Überprüfung als auch das Hochladen Ihrer Daten in einem einzigen Durchgang durchführt.

> **Important**
>
> Wenn Sie S3 Object Lambda verwenden, werden alle Anfragen an S3 Object Lambda mit `s3-object-lambda` anstelle von `s3` signiert. Dieses Verhalten wirkt sich auf die Signatur der nachfolgenden Prüfsummenwerte aus. Weitere Informationen zu S3 Object Lambda finden Sie unter Transformieren von Objekten mit S3 Object Lambda.

Verwenden von Prüfsummen auf Teilebene für mehrteilige Uploads

Wenn Objekte auf Amazon S3 hochgeladen werden, können sie entweder als einzelnes Objekt oder durch den mehrteiligen Upload-Prozess hochgeladen werden. Objekte, die größer als 16 MB sind und über die Konsole hochgeladen werden, werden automatisch mit mehrteiligen Uploads hochgeladen.

Weitere Informationen über mehrteilige Uploads finden Sie unter Hochladen und Kopieren von Objekten mit mehrteiligen Uploads.

Wenn ein Objekt als mehrteiliger Upload hochgeladen wird, handelt es sich ETag bei dem Objekt nicht um eine MD5 Zusammenfassung des gesamten Objekts. Amazon S3 berechnet den MD5 Digest jedes einzelnen Teils beim Hochladen. Die MD5 Digests werden verwendet, um den Wert ETag für das endgültige Objekt zu bestimmen. Amazon S3 verkettet die Bytes für die MD5 Digests miteinander und berechnet dann den Digest dieser verketteten MD5 Werte. Der letzte Schritt bei der Erstellung von ETag besteht darin, dass Amazon S3 am Ende einen Bindestrich mit der Gesamtzahl der Teile hinzufügt.

Stellen Sie sich zum Beispiel ein Objekt vor, das mit einem mehrteiligen Upload hochgeladen wurde und den Wert ETag von `C9A5A6878D97B48CC965C1E41859F034-14` hat. In diesem Fall `C9A5A6878D97B48CC965C1E41859F034` ist der MD5 Digest aller Digests miteinander verkettet. Der Wert `-14` gibt an, dass mit dem mehrteiligen Upload dieses Objekts 14 Teile verknüpft sind.

Wenn Sie zusätzliche Prüfsummenwerte für Ihr mehrteiliges Objekt aktiviert haben, berechnet Amazon S3 die Prüfsumme für jeden einzelnen Teil mithilfe des angegebenen Prüfsummenalgorithmus. Die Prüfsumme für das abgeschlossene Objekt wird auf die gleiche Weise berechnet, wie Amazon S3 den MD5 Digest für den mehrteiligen Upload berechnet. Mithilfe dieser Prüfsumme können Sie die Integrität des Objekts überprüfen.

Um Informationen über das Objekt abzurufen, einschließlich der Anzahl der Teile, aus denen das gesamte Objekt besteht, können Sie die Operation verwenden. GetObjectAttributes Mit zusätzlichen Prüfsummen können Sie auch Informationen für jeden einzelnen Teil wiederherstellen, der den Prüfsummenwert jedes Teils enthält.

Bei abgeschlossenen Uploads können Sie die Prüfsumme eines einzelnen Teils abrufen, indem Sie die HeadObjectOperationen GetObjectoder verwenden und eine Bauteilnummer oder einen Bytebereich angeben, der einem einzelnen Teil entspricht. Wenn Sie die Prüfsummenwerte für einzelne Teile von mehrteiligen Uploads abrufen möchten, die noch in Bearbeitung sind, können Sie Folgendes verwenden. ListParts

Aufgrund der Art und Weise, wie Amazon S3 die Prüfsumme für mehrteilige Upload-Objekte berechnet, kann sich der Prüfsummenwert für das Objekt ändern, wenn Sie es kopieren. Wenn Sie ein SDK oder das verwenden REST API und Sie aufrufen CopyObject, kopiert Amazon S3 jedes Objekt bis zu den Größenbeschränkungen des `CopyObject` API Vorgangs. Amazon S3 führt diese Kopie als einzelne Aktion aus, unabhängig davon, ob das Objekt in einer einzigen Anforderung oder im Rahmen eines mehrteiligen Uploads hochgeladen wurde. Mit einem Kopierbefehl ist die

Prüfsumme des Objekts eine direkte Prüfsumme des vollständigen Objekts. Wenn das Objekt ursprünglich mit einem mehrteiligen Upload hochgeladen wurde, ändert sich der Prüfsummenwert, auch wenn die Daten unverändert bleiben.

> **Note**
>
> Objekte, die die Größenbeschränkungen des `CopyObject` API Vorgangs überschreiten, müssen mehrteilige Kopierbefehle verwenden.

> **Important**
>
> Wenn Sie einige Operationen mit dem ausführen AWS Management Console, verwendet Amazon S3 einen mehrteiligen Upload, wenn das Objekt größer als 16 MB ist. In diesem Fall ist die Prüfsumme keine direkte Prüfsumme des vollständigen Objekts, sondern eine Berechnung, die auf den Prüfsummenwerten jeden einzelnen Teils basiert.
> Stellen Sie sich zum Beispiel ein Objekt mit einer Größe von 100 MB vor, das Sie als einteiligen direkten Upload mit dem hochgeladen haben. REST API Die Prüfsumme ist in diesem Fall eine Prüfsumme des gesamten Objekts. Wenn Sie das Objekt später mit der Konsole umbenennen, es kopieren, die Speicherklasse ändern oder die Metadaten bearbeiten, verwendet Amazon S3 die mehrteilige Upload-Funktion, um das Objekt zu aktualisieren. Infolgedessen erstellt Amazon S3 einen neuen Prüfsummenwert für das Objekt, der basierend auf den Prüfsummenwerten der einzelnen Teile berechnet wird.
> Die obige Liste der Konsolenoperationen ist keine vollständige Liste aller möglichen Aktionen, die Sie ergreifen können, wenn sie dazu führen AWS Management Console , dass Amazon S3 das Objekt mithilfe der mehrteiligen Upload-Funktion aktualisiert. Beachten Sie, dass der Prüfsummenwert möglicherweise nicht die Prüfsumme des gesamten Objekts ist, wenn Sie die Konsole für Objekte mit einer Größe über 16 MB verwenden.

Löschen von Amazon-S3-Objekten

Sie können mit der Amazon S3-Konsole,, AWS SDKs AWS Command Line Interface (AWS CLI) oder ein oder mehrere Objekte direkt aus Amazon S3 löschen RESTAPI. Wenn Sie beispielsweise Protokolldateien sammeln, sollten Sie sie unbedingt löschen, wenn sie nicht mehr benötigt werden. Sie können eine S3-Lebenszyklusregel einrichten, um Objekte wie Protokolldateien automatisch zu löschen.

Um ein Objekt zu löschen, können Sie eine der folgenden API Operationen verwenden:

- Ein einzelnes Objekt löschen — Amazon S3 bietet die API Operation `DELETE` (`DeleteObject`), mit der Sie ein Objekt in einer einzigen HTTP Anfrage löschen können.
- Mehrere Objekte löschen — Amazon S3 bietet den Multi-Object Delete (`DeleteObjects`) API - Vorgang, mit dem Sie bis zu 1.000 Objekte in einer einzigen HTTP Anfrage löschen können.

Wenn Sie Objekte aus einem Bucket löschen, für den die Versionierung nicht aktiviert ist, geben Sie nur den Namen des Objektschlüssels an. Wenn Sie jedoch Objekte aus einem Bucket mit aktivierter Versionierung löschen, können Sie die Versions-ID des Objekts angeben, um eine bestimmte Version des Objekts zu löschen.

Bewährte Methoden, die Sie vor dem Löschen eines Objekts beachten sollten

Bevor Sie ein Objekt löschen, sollten Sie die folgenden bewährten Methoden beachten:

- Aktivieren Sie die Bucket-Versionierung. Die S3-Versionierung bietet zusätzlichen Schutz vor einfachen `DeleteObject` Anfragen, um versehentliche Löschungen zu verhindern. Wenn Sie bei versionierten Buckets die aktuelle Version eines Objekts löschen oder wenn eine Löschanforderung keine bestimmte Versions-ID angibt, löscht Amazon S3 das Objekt nicht dauerhaft. Stattdessen fügt S3 eine Löschmarkierung hinzu, wodurch das Objekt automatisch gelöscht wird. Die Löschmarke wird dann zur aktuellen (oder neuesten) Version des Objekts mit einer neuen Versions-ID. Weitere Informationen finden Sie unter Löschen von Objektversionen aus einem Bucket mit aktivierter Versionierung.
- Wenn Sie eine große Anzahl von Objekten löschen oder Objekte auf der Grundlage des Objekterstellungsdatums programmgesteuert löschen möchten, legen Sie eine S3-Lifecycle-Konfiguration für Ihren Bucket fest. Um diese Löschungen zu überwachen, empfehlen wir Ihnen, eine S3 Lifecycle-Ereignisbenachrichtigung zu verwenden. Wenn Sie S3 Lifecycle-Benachrichtigungen konfigurieren, benachrichtigt Sie der `s3:LifecycleExpiration:Delete` Ereignistyp, wenn ein Objekt in einem Bucket gelöscht wird. Es benachrichtigt Sie auch, wenn eine Objektversion durch eine S3-Lebenszyklus-Konfiguration dauerhaft gelöscht wird. Der `s3:LifecycleExpiration:DeleteMarkerCreated` Ereignistyp benachrichtigt Sie, wenn S3 Lifecycle eine Löschmarkierung erstellt. Eine Löschmarkierung wird erstellt, wenn eine aktuelle Version eines Objekts in einem versionierten Bucket gelöscht wird.

- Bevor Sie Ihre S3-Lifecycle-Konfiguration aktualisieren, vergewissern Sie sich, dass Lifecycle die Aktionen für alle vorgesehenen Objekte abgeschlossen hat. Weitere Informationen finden Sie im Abschnitt Aktualisieren, Deaktivieren oder Löschen von Lebenszyklusregeln unter Einrichten einer S3-Lifecycle-Konfiguration für einen Bucket.

  > **Note**
  >
  > Die S3-Lebenszyklusregeln müssen für die richtige Teilmenge von Objekten gelten, um unbeabsichtigte Löschungen zu verhindern. Sie können Objekte bei der Erstellung der Lebenszyklusregeln nach Präfixen, Objekt-Tags oder Objektgrößen filtern.

- Erwägen Sie, Benutzern das Entfernen oder Löschen von Objekten aus Ihrem Bucket zu verbieten. Um Benutzer einzuschränken, müssen Sie Benutzern in Ihren Amazon S3 S3-Bucket-Richtlinien ausdrücklich die Berechtigungen für die folgenden Aktionen verweigern:
  - `s3:DeleteObject`, `s3:DeleteObjectVersion` (um mithilfe von API Anfragen zu kontrollieren, wer Objekte löschen kann)
  - `s3:PutLifecycleConfiguration`(um zu kontrollieren, wer S3-Lifecycle-Ablaufregeln hinzufügen kann)
- Erwägen Sie, S3 Replication zu verwenden, um mehrere Kopien Ihrer Daten zu erstellen und diese an mehreren Standorten gleichzeitig zu replizieren. Sie können so viele Ziel-Buckets wie nötig auswählen. Wenn ein Objekt versehentlich gelöscht wird, haben Sie außerdem immer noch eine Kopie der Daten.

Löschen von Objekten aus einem Bucket mit aktivierter Versionierung

Wenn Ihr Bucket versionierungsfähig ist, können mehrere Versionen desselben Objekts im Bucket vorhanden sein. Bei der Arbeit mit Buckets mit aktivierter Versionierung ermöglichen die `Delete` API Operationen die folgenden Optionen:

- Eine Löschanforderung ohne Versions-ID angeben – Geben Sie nur den Schlüssel des Objekts an, nicht die Versions-ID. In diesem Fall erstellt Amazon S3 eine Löschmarkierung über der aktuellen Version des Objekts und gibt seine Versions-ID in der Antwort zurück. Dadurch verschwindet Ihr Objekt aus dem Bucket. Weitere Informationen zum Objekt-Versioning und zum Konzept der Löschmarkierung finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

- Geben Sie eine versionierte Löschanforderung an — Geben Sie sowohl den Schlüssel als auch die Versions-ID an. In diesem Fall sind die folgenden Ergebnisse möglich:
  - Wenn die Versions-ID einer spezifischen Objektversion zugeordnet ist, löscht Amazon S3 die spezifische Version des Objekts.
  - Wenn die Versions-ID der Löschmarkierung eines Objekts zugeordnet ist, löscht Amazon S3 die Löschmarkierung. Wenn die Löschmarke gelöscht wird, erscheint das Objekt wieder in Ihrem Bucket.

Löschen von Objekten aus einem Bucket, für das die Versionierung gesperrt wurde

Wenn Ihr Bucket aufgrund der Versionierung gesperrt ist, verhalten sich die `Delete` API Operationen bei Buckets mit aktivierter Versionierung genauso (außer wenn die aktuelle Version eine Null-Versions-ID hat). Weitere Informationen finden Sie unter Löschen von Objekten aus Buckets mit ausgesetztem Versioning.

Objekte aus einem Bucket ohne Versionsnummer löschen

Wenn Ihr Bucket keine Version hat, können Sie den Schlüssel des Objekts in den `Delete` API Vorgängen angeben und Amazon S3 löscht das Objekt dauerhaft. Um das permanente Löschen eines Objekts zu verhindern, aktivieren Sie die Bucket-Versionierung.

Objekte aus einem Bucket mit MFA -aktiviertem Bucket löschen

Beachten Sie beim Löschen von Objekten aus einem Bucket mit aktivierter Multi-Faktor-Authentifizierung (MFA) Folgendes:

- Wenn Sie ein ungültiges MFA Token angeben, schlägt die Anfrage immer fehl.
- Wenn Sie einen Bucket mit MFA aktiviertem Bucket haben und eine versionierte Löschanforderung stellen (Sie geben einen Objektschlüssel und eine Versions-ID an), schlägt die Anfrage fehl, wenn Sie kein MFA gültiges Token angeben. Außerdem schlägt die gesamte Anforderung fehl, wenn Sie den `Delete` API Vorgang mit mehreren Objekten für einen Bucket mit MFA aktivierter Option verwenden und es sich bei einer der Löschungen um eine versionierte Löschanforderung handelt (d. h. Sie geben einen Objektschlüssel und eine Versions-ID an), wenn Sie kein Token angeben. MFA

In den folgenden Fällen ist die Anfrage jedoch erfolgreich:

- Wenn Sie einen Bucket mit MFA aktiviertem Bucket haben und eine nicht versionierte Löschanforderung stellen (Sie löschen kein versioniertes Objekt) und Sie kein Token angeben, ist der Löschvorgang erfolgreich. MFA
- Wenn Sie eine Anfrage zum Löschen mehrerer Objekte haben, in der nur nicht versionierte Objekte aus einem Bucket mit MFA aktiviertem Bucket gelöscht werden sollen, und Sie kein Token angeben, sind die Löschungen erfolgreich. MFA

Informationen zum Löschen finden Sie unter. MFA MFALöschen konfigurieren

Themen

- Löschen eines einzelnen Objekts
- Löschen von mehreren Objekten

Löschen eines einzelnen Objekts

Sie können die Amazon S3 S3-Konsole oder die verwenden DELETEAPI, um ein einzelnes vorhandenes Objekt aus einem S3-Bucket zu löschen. Weitere Informationen zum Löschen von Objekten in Amazon S3 finden Sie unter Löschen von Amazon-S3-Objekten.

Für alle Objekte, die Sie in Ihrem S3-Bucket aufbewahren, entstehen Speicherkosten, deshalb sollten Sie Objekte löschen, die Sie nicht mehr brauchen. Wenn Sie beispielsweise Protokolldateien sammeln, sollten Sie sie unbedingt löschen, wenn sie nicht mehr benötigt werden. Sie können einen Lebenszyklusregel definieren, um Objekte wie Protokolldateien automatisch zu löschen. Weitere Informationen finden Sie unter the section called “Einrichten der Lebenszyklus-Konfiguration”.

Informationen zu den Funktionen und Preisen von Amazon S3 finden Sie unter Amazon-S3-Preise.

Verwenden der S3-Konsole

Befolgen Sie diese Schritte, um mithilfe der Amazon-S3-Konsole ein einzelnes Objekt aus einem Bucket zu löschen.

> **Warning**
>
> Wenn Sie ein Objekt oder eine bestimmte Objektversion dauerhaft in der Amazon S3 S3-Konsole löschen, kann der Löschvorgang nicht rückgängig gemacht werden.

Um ein Objekt zu löschen, für das die Versionsverwaltung aktiviert oder ausgesetzt wurde

> **Note**
>
> Wenn die Versions-ID für ein Objekt in einem Bucket, für das die Versionierung gesperrt wurdeNULL, als markiert ist, löscht S3 das Objekt dauerhaft, da keine früheren Versionen existieren. Wenn jedoch eine gültige Versions-ID für das Objekt in einem Bucket aufgeführt ist, für das die Versionierung gesperrt wurde, erstellt S3 eine Löschmarkierung für das gelöschte Objekt, wobei die vorherigen Versionen des Objekts beibehalten werden.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Bucket Name (Bucket-Name) den Namen des Buckets aus, aus dem Sie ein Objekt löschen möchten.
3. Wählen Sie das Objekt aus und wählen Sie dann Löschen.
4. Um das Löschen der Objektliste unter Bestimmte Objekte im Bereich Objekte löschen zu bestätigen? Geben Sie in das Textfeld ein**`delete`**.

Um eine bestimmte Objektversion in einem Bucket mit aktivierter Versionierung dauerhaft zu löschen

> **Warning**
>
> Wenn Sie eine bestimmte Objektversion in Amazon S3 dauerhaft löschen, kann der Löschvorgang nicht rückgängig gemacht werden.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Bucket Name (Bucket-Name) den Namen des Buckets aus, aus dem Sie ein Objekt löschen möchten.
3. Wählen Sie das zu löschende Objekt aus.
4. Wählen Sie den Schalter Versionen anzeigen.
5. Wählen Sie die Objektversion aus und klicken Sie dann auf Löschen.

6. Um das dauerhafte Löschen der spezifischen Objektversionen zu bestätigen, die unter Bestimmte Objekte im Bereich Objekte löschen? aufgeführt sind Geben Sie im Textfeld Dauerhaft löschen ein. Amazon S3 löscht die spezifische Objektversion dauerhaft.

Um ein Objekt in einem Amazon S3 S3-Bucket, für den die Versionierung nicht aktiviert ist, dauerhaft zu löschen

> **Warning**
>
> Wenn Sie ein Objekt in Amazon S3 dauerhaft löschen, kann der Löschvorgang nicht rückgängig gemacht werden. Außerdem sind Löschungen für alle Buckets ohne aktivierte Versionierung dauerhaft.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Bucket Name (Bucket-Name) den Namen des Buckets aus, aus dem Sie ein Objekt löschen möchten.
3. Wählen Sie das Objekt aus und wählen Sie dann Löschen.
4. Um das dauerhafte Löschen des Objekts zu bestätigen, das unter Bestimmte Objekte im Feld Objekte löschen? aufgeführt ist Geben Sie im Textfeld Dauerhaft löschen ein.

> **Note**
>
> Falls beim Löschen Ihres Objekts Probleme auftreten, finden Sie weitere Informationen unterIch möchte versionierte Objekte dauerhaft löschen.

Verwenden des AWS CLI

Um ein Objekt pro Anfrage zu löschen, verwenden Sie die `DELETE`API. Weitere Informationen finden Sie unter DELETEObjekt. Weitere Hinweise zur Verwendung von CLI zum Löschen eines Objekts finden Sie unter delete-object.

Verwenden von REST API

Sie können den verwenden AWS SDKs, um ein Objekt zu löschen. Wenn Ihre Anwendung dies jedoch erfordert, können Sie REST Anfragen direkt senden. Weitere Informationen finden Sie unter DELETEObjekt in der Amazon Simple Storage Service API Reference.

Mit dem AWS SDKs

Die folgenden Beispiele zeigen, wie Sie mit AWS SDKs dem ein Objekt aus einem Bucket löschen können. Weitere Informationen finden Sie unter DELETEObjekt in der Amazon Simple Storage Service API Reference

Wenn für den Bucket das S3-Versioning aktiviert ist, stehen Ihnen die folgenden Optionen zur Verfügung:

- Löschen einer spezifischen Objektversion durch Angabe einer Versions-ID.
- Löschen eines Objekts ohne Angabe einer Version-ID. In diesem Fall fügt Amazon S3 dem Objekt eine Löschmarkierung hinzu.

Weitere Informationen über das S3-Versioning finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

Java

Example Beispiel 1: Löschen eines Objekts (nicht versionierter Bucket)

Im folgenden Beispiel wird davon ausgegangen, dass der Bucket nicht versionierungsfähig ist und das Objekt keine Version hat. IDs In der Löschanfrage geben Sie nur den Objektschlüssel und keine Versions-ID an.

Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.DeleteObjectRequest;
```

```
import java.io.IOException;

public class DeleteObjectNonVersionedBucket {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ****";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            s3Client.deleteObject(new DeleteObjectRequest(bucketName, keyName));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example Beispiel 2: Löschen eines Objekts (versionierter Bucket)

Das folgende Beispiel löscht ein Objekt aus einem versionierten Bucket. Das Beispiel löscht eine spezifische Objektversion durch Angabe des Namens des Objektschlüssels und der Versions-ID.

Das Beispiel erledigt Folgendes:

1. Fügt dem Bucket ein Beispielobjekt hinzu. Amazon S3 gibt die die Versions-ID des neu hinzugefügten Objekts zurück. Das Beispiel verwendet diese Versions-ID in der Löschanfrage.
2. Löscht die Objektversion durch Angabe sowohl des Objektschlüsselnamens als auch einer Versions-ID. Wenn keine anderen Versionen dieses Objekts vorhanden sind, löscht Amazon S3 das gesamte Objekt. Andernfalls löscht Amazon S3 die angegebene Version.

> **Note**
>
> Sie können die Version IDs eines Objekts abrufen, indem Sie eine `ListVersions` Anfrage senden.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketVersioningConfiguration;
import com.amazonaws.services.s3.model.DeleteVersionRequest;
import com.amazonaws.services.s3.model.PutObjectResult;

import java.io.IOException;

public class DeleteObjectVersionEnabledBucket {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ****";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Check to ensure that the bucket is versioning-enabled.
            String bucketVersionStatus =
 s3Client.getBucketVersioningConfiguration(bucketName).getStatus();
            if (!bucketVersionStatus.equals(BucketVersioningConfiguration.ENABLED))
 {
                System.out.printf("Bucket %s is not versioning-enabled.",
 bucketName);
            } else {
                // Add an object.
```

```
                PutObjectResult putResult = s3Client.putObject(bucketName, keyName,
                        "Sample content for deletion example.");
                System.out.printf("Object %s added to bucket %s\n", keyName,
 bucketName);

                // Delete the version of the object that we just created.
                System.out.println("Deleting versioned object " + keyName);
                s3Client.deleteVersion(new DeleteVersionRequest(bucketName, keyName,
 putResult.getVersionId()));
                System.out.printf("Object %s, version %s deleted\n", keyName,
 putResult.getVersionId());
            }
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Das folgende Beispiel veranschaulicht, wie Sie ein Objekt sowohl aus versionierten als auch aus nicht-versionierten Buckets löschen. Weitere Informationen über das S3-Versioning finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

Example Löschen eines Objekts aus einem nicht-versionierten Bucket

Das folgende C#-Beispiel löscht ein Objekt aus einem nicht-versionierten Bucket. In dem Beispiel wird davon ausgegangen, dass die Objekte keine Version habenIDs, sodass Sie keine Version angebenIDs. Sie geben nur den Objektschlüssel ein.

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem AWS SDK for. NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
```

```
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class DeleteObjectNonVersionedBucketTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** object key ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            DeleteObjectNonVersionedBucketAsync().Wait();
        }
        private static async Task DeleteObjectNonVersionedBucketAsync()
        {
            try
            {
                var deleteObjectRequest = new DeleteObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName
                };

                Console.WriteLine("Deleting an object");
                await client.DeleteObjectAsync(deleteObjectRequest);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 deleting an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 deleting an object", e.Message);
            }
        }
    }
```

```
}
```

Example Löschen eines Objekts aus einem versionierten Bucket

Das folgende C#-Beispiel löscht ein Objekt aus einem versionierten Bucket. Es löscht eine bestimmte Objektversion durch Angabe des Namens des Objektschlüssels und der Versions-ID.

Der Code führt die folgenden Aufgaben durch:

1. Aktiviert das S3-Versioning für einen Bucket, den Sie angeben (wenn S3-Versioning bereits aktiviert ist, hat dies keine Auswirkungen).
2. Fügt dem Bucket ein Beispielobjekt hinzu. In der Antwort gibt Amazon S3 die Versions-ID des neu hinzugefügten Objekts zurück. Das Beispiel verwendet diese Versions-ID in der Löschanfrage.
3. Löscht das Beispielobjekt durch Angabe sowohl des Objektschlüsselnamens als auch einer Versions-ID.

> **Note**
>
> Sie können die Version-IDs eines Objekts durch Senden einer `ListVersions`-Anfrage anfordern.
>
> ```
> var listResponse = client.ListVersions(new ListVersionsRequest { BucketName
>  = bucketName, Prefix = keyName });
> ```

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem AWS SDK for. NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class DeleteObjectVersion
    {
        private const string bucketName = "*** versioning-enabled bucket name ***";
```

```
        private const string keyName = "*** Object Key Name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            CreateAndDeleteObjectVersionAsync().Wait();
        }

        private static async Task CreateAndDeleteObjectVersionAsync()
        {
            try
            {
                // Add a sample object.
                string versionID = await PutAnObject(keyName);

                // Delete the object by specifying an object key and a version ID.
                DeleteObjectRequest request = new DeleteObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
                    VersionId = versionID
                };
                Console.WriteLine("Deleting an object");
                await client.DeleteObjectAsync(request);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
deleting an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
deleting an object", e.Message);
            }
        }

        static async Task<string> PutAnObject(string objectKey)
        {
            PutObjectRequest request = new PutObjectRequest
```

```
            {
                BucketName = bucketName,
                Key = objectKey,
                ContentBody = "This is the content body!"
            };
            PutObjectResponse response = await client.PutObjectAsync(request);
            return response.VersionId;
        }
    }
}
```

PHP

Dieses Beispiel zeigt, wie Klassen aus Version 3 von verwendet werden, AWS SDK for PHP um ein Objekt aus einem Bucket ohne Versionsnummer zu löschen. Weitere Informationen über das Löschen eines Objekts aus einem versionsfähigen Bucket finden Sie unter Verwenden von REST API.

Weitere Informationen zu AWS SDK for Ruby finden Sie API unter AWS SDKfor Ruby — Version 2.

Im folgenden PHP Beispiel wird ein Objekt aus einem Bucket gelöscht. Da dieses Beispiel zeigt, wie Objekte aus nicht versionsgesteuerten Buckets gelöscht werden, gibt es in der Löschanfrage nur den Bucket-Namen und Objektschlüssel (keine Versions-ID) an.

```
<?php

require 'vendor/autoload.php';

use Aws\S3\S3Client;
use Aws\S3\Exception\S3Exception;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// 1. Delete the object from the bucket.
try
```

```
{
    echo 'Attempting to delete ' . $keyname . '...' . PHP_EOL;

    $result = $s3->deleteObject([
        'Bucket' => $bucket,
        'Key'    => $keyname
    ]);

    if ($result['DeleteMarker'])
    {
        echo $keyname . ' was deleted or does not exist.' . PHP_EOL;
    } else {
        exit('Error: ' . $keyname . ' was not deleted.' . PHP_EOL);
    }
}
catch (S3Exception $e) {
    exit('Error: ' . $e->getAwsErrorMessage() . PHP_EOL);
}

// 2. Check to see if the object was deleted.
try
{
    echo 'Checking to see if ' . $keyname . ' still exists...' . PHP_EOL;

    $result = $s3->getObject([
        'Bucket' => $bucket,
        'Key'    => $keyname
    ]);

    echo 'Error: ' . $keyname . ' still exists.';
}
catch (S3Exception $e) {
    exit($e->getAwsErrorMessage());
}
```

Javascript

```
import { DeleteObjectCommand } from "@aws-sdk/client-s3";
import { s3Client } from "./libs/s3Client.js" // Helper function that creates Amazon
 S3 service client module.

export const bucketParams = { Bucket: "BUCKET_NAME", Key: "KEY" };
```

```
export const run = async () => {
  try {
    const data = await s3Client.send(new DeleteObjectCommand(bucketParams));
    console.log("Success. Object deleted.", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Löschen von mehreren Objekten

Für alle Objekte, die Sie in Ihrem S3-Bucket aufbewahren, entstehen Speicherkosten, deshalb sollten Sie Objekte löschen, die Sie nicht mehr brauchen. Wenn Sie beispielsweise Protokolldateien sammeln, sollten Sie sie unbedingt löschen, wenn sie nicht mehr benötigt werden. Sie können einen Lebenszyklusregel definieren, um Objekte wie Protokolldateien automatisch zu löschen. Weitere Informationen finden Sie unter the section called “Einrichten der Lebenszyklus-Konfiguration”.

Informationen zu den Funktionen und Preisen von Amazon S3 finden Sie unter Amazon-S3-Preise.

Sie können die Amazon S3 S3-Konsole oder die verwenden AWS SDKs, REST API um mehrere Objekte gleichzeitig aus einem S3-Bucket zu löschen.

Verwenden der S3-Konsole

Befolgen Sie diese Schritte, um mithilfe der Amazon-S3-Konsole mehrere Objekte aus einem Bucket zu löschen.

> **Warning**
>
> - Das Löschen eines angegebenen Objekts kann nicht rückgängig gemacht werden.
> - Diese Aktion löscht alle angegebenen Objekte. Warten Sie beim Löschen von Ordnern, bis die Löschaktion abgeschlossen ist, bevor Sie dem Ordner neue Objekte hinzufügen. Andernfalls könnten auch neue Objekte gelöscht werden.
> - Wenn Objekte in einem Bucket ohne aktivierte Versionierung gelöscht werden, löscht Amazon S3 die Objekte dauerhaft.

- Beim Löschen von Objekten in einem Bucket mit aktivierter oder angehaltener Bucket-Versionierung erstellt Amazon S3 Löschmarkierungen. Weitere Informationen finden Sie unter Arbeiten mit Löschmarkierungen.

Um Objekte zu löschen, für die die Versionsverwaltung aktiviert oder gesperrt ist

> **Note**
>
> Wenn die Version IDs für das Objekt in einem Bucket, für das die Versionierung gesperrt wurdeNULL, als markiert sind, löscht S3 die Objekte dauerhaft, da keine früheren Versionen existieren. Wenn jedoch eine gültige Versions-ID für die Objekte in einem Bucket aufgeführt ist, für das die Versionierung gesperrt wurde, erstellt S3 Löschmarkierungen für die gelöschten Objekte, wobei die vorherigen Versionen der Objekte beibehalten werden.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste der Bucket-Namen den Namen des Buckets aus, aus dem Sie die Objekte löschen möchten.
3. Wählen Sie die Objekte aus und klicken Sie dann auf Löschen.
4. Um das Löschen der Objektliste unter Bestimmte Objekte im Bereich Objekte löschen zu bestätigen? Geben Sie in das Textfeld ein**`delete`**.

Um bestimmte Objektversionen in einem Bucket mit aktivierter Versionierung dauerhaft zu löschen

> **Warning**
>
> Wenn Sie bestimmte Objektversionen in Amazon S3 dauerhaft löschen, kann der Löschvorgang nicht rückgängig gemacht werden.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste der Bucket-Namen den Namen des Buckets aus, aus dem Sie die Objekte löschen möchten.

3. Wählen Sie das zu löschende -Objekt aus.
4. Wählen Sie den Schalter Versionen anzeigen.
5. Wählen Sie die Objektversionen aus und wählen Sie dann Löschen.
6. Um das dauerhafte Löschen der spezifischen Objektversionen zu bestätigen, die unter Bestimmte Objekte im Bereich Objekte löschen? aufgeführt sind Geben Sie im Textfeld Dauerhaft löschen ein. Amazon S3 löscht die spezifischen Objektversionen dauerhaft.

Um die Objekte in einem Amazon S3 S3-Bucket, für die die Versionierung nicht aktiviert ist, dauerhaft zu löschen

> **Warning**
>
> Wenn Sie ein Objekt in Amazon S3 dauerhaft löschen, kann der Löschvorgang nicht rückgängig gemacht werden. Außerdem sind Löschungen für alle Buckets ohne aktivierte Versionierung dauerhaft.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste der Bucket-Namen den Namen des Buckets aus, aus dem Sie die Objekte löschen möchten.
3. Wählen Sie die Objekte aus und klicken Sie dann auf Löschen.
4. Um das dauerhafte Löschen der unter Bestimmte Objekte aufgelisteten Objekte im Abschnitt Objekte löschen? zu bestätigen Geben Sie im Textfeld Dauerhaft löschen ein.

> **Note**
>
> Falls beim Löschen Ihrer Objekte Probleme auftreten, finden Sie weitere Informationen unterIch möchte versionierte Objekte dauerhaft löschen.

Verwenden des AWS SDKs

Beispiele für das Löschen mehrerer Objekte mit dem finden Sie in den AWS SDKs Codebeispielen in der Amazon S3 API S3-Referenz.

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS SDKs in der Amazon S3 API S3-Referenz.

Mit dem REST API

Sie können den verwenden AWS SDKs, um mehrere Objekte mithilfe der Option Löschen mehrerer Objekte zu löschenAPI. Wenn Ihre Anwendung dies jedoch erfordert, können Sie REST Anfragen direkt senden.

Weitere Informationen finden Sie unter Mehrere Objekte löschen in der Amazon Simple Storage API Service-Referenz.

Organisieren, Auflisten und Arbeiten mit Ihren Objekten

In Amazon S3 können Sie Präfixe verwenden, um Ihren Speicher zu organisieren. Ein Präfix ist eine logische Gruppierung der Objekte in einem Bucket. Der Präfixwert ist mit einem Verzeichnisnamen vergleichbar, der Ihnen ermöglicht, ähnliche Daten in demselben Verzeichnis eines Buckets zu speichern. Wenn Sie Objekte programmgesteuert hochladen, können Sie Präfixe verwenden, um Ihre Daten zu organisieren.

In der Amazon-S3-Konsole werden Präfixe als Ordner bezeichnet. Sie können alle Ihre Objekte und Ordner in der S3-Konsole anzeigen, indem Sie zu einem Bucket navigieren. Sie können auch Informationen zu jedem Objekt anzeigen, einschließlich Objekteigenschaften.

Weitere Informationen zum Auflisten und Organisieren Ihrer Daten in Amazon S3 finden Sie in den folgenden Themen.

Themen

- Organisieren von Objekten mit Präfixen
- Programmgesteuertes Auflisten von Objektschlüsseln
- Organisieren von Objekten in der Amazon S3-Konsole mithilfe von Ordnern
- Anzeigen von Objekteigenschaften in der Amazon-S3-Konsole
- Kategorisieren des Speichers mithilfe von Markierungen

Organisieren von Objekten mit Präfixen

Sie können Präfixe verwenden, um die Daten zu organisieren, die Sie in Amazon-S3-Buckets speichern. Ein Präfix ist eine Zeichenfolge am Anfang des Objektschlüsselnamens. Ein Präfix kann

eine beliebige Länge haben, abhängig von der maximalen Länge des Objektschlüsselnamens (1.024 Byte). Sie können sich Präfixe als eine Möglichkeit vorstellen, Ihre Daten ähnlich wie Verzeichnisse zu organisieren. Präfixe sind jedoch keine Verzeichnisse.

Die Suche nach Präfix begrenzt die Ergebnisse auf die Schlüssel, die mit dem angegebenen Präfix beginnen. Das Trennzeichen veranlasst Listenoperationen, alle Schlüssel mit gemeinsamem Präfix in einer Zusammenfassungsliste als Ergebnis bereitzustellen.

Zweck der Präfix- und Trennzeichen-Parameter ist es, Ihnen dabei zu helfen, Ihre Schlüssel hierarchisch zu organisieren und dann zu durchsuchen. Dazu wählen Sie zuerst ein Trennzeichen für Ihren Bucket aus, wie beispielsweise einen Schrägstrich (/), das in Ihren Schlüsselnamen voraussichtlich nicht vorkommt. Sie können ein anderes Zeichen als Trennzeichen verwenden. Der Schrägstrich (/) hat keine besonderen Eigenschaften, wird aber sehr häufig als Präfix-Trennzeichen verwendet. Anschließend konstruieren Sie Ihre Schlüsselnamen, indem Sie alle Ebenen der Hierarchie verknüpfen und jede Ebene mit dem Trennzeichen abtrennen.

Wenn Sie beispielsweise Informationen über Städte speichern, könnten Sie diese natürlich nach dem Kontinent, dann nach dem Land und dann nach der Provinz oder dem Staat anordnen. Diese Namen enthalten normalerweise keine Interpunktionszeichen, deshalb könnten Sie den Schrägstrich (/) als Trennzeichen verwenden. Die folgenden Beispiele verwenden einen Schrägstrich (/) als Trennzeichen.

- Europe/France/Nouvelle-Aquitaine/Bordeaux
- Norden America/Canada/Quebec/Montreal
- Norden America/USA/Washington/Bellevue
- Norden America/USA/Washington/Seattle

Wenn Sie auf diese Weise Daten für jede Stadt in der Welt gespeichert haben, ist es relativ mühsam, einen flachen Schlüssel-Namespace zu verwalten. Durch Verwendung von `Prefix` und `Delimiter` für die Listenvorgänge können Sie die Hierarchie verwenden, die Sie zum Auflisten Ihrer Daten erstellt haben. Um beispielsweise alle Bundesstaaten aufzulistenUSA, legen Sie `Delimiter='/'` und fest`Prefix='North America/USA/'`. Um alle Provinzen in Kanada aufzulisten, für die Sie Daten haben, legen Sie `Delimiter='/'` und `Prefix='North America/Canada/'` fest.

Weitere Informationen zu Trennzeichen, Präfixen und verschachtelten Ordnern finden Sie unter Unterschied zwischen Präfixen und verschachtelten Ordnern.

Auflisten von Objekten mit Präfixen und Trennzeichen

Wenn Sie eine Listenanforderung mit einem Trennzeichen ausgeben, können Sie Ihre Hierarchie nur auf einer Ebene durchsuchen und die (möglicherweise Millionen von) Schlüsseln, die auf tieferen Ebenen verschachtelt sind, überspringen und zusammenfassen. Nehmen wir zum Beispiel an, Sie haben einen Bucket (*amzn-s3-demo-bucket*) mit den folgenden Schlüsseln:

`sample.jpg`

`photos/2006/January/sample.jpg`

`photos/2006/February/sample2.jpg`

`photos/2006/February/sample3.jpg`

`photos/2006/February/sample4.jpg`

Der Beispiel-Bucket hat nur das `sample.jpg`-Objekt auf der Root-Ebene. Um nur die Objekte auf Stammebene im Bucket aufzulisten, senden Sie eine GET Anfrage für den Bucket mit dem Schrägstrich (`/`) als Trennzeichen. In der Antwort gibt Amazon S3 den `sample.jpg`-Objektschlüssel zurück, weil er das Trennzeichen `/` nicht enthält. Alle anderen Schlüssel enthalten das Trennzeichen. Amazon S3 gruppiert diese Schlüssel und gibt ein einziges `CommonPrefixes`-Element mit dem Präfix-Wert `photos/` zurück, eine Unterzeichenfolge vom Anfang dieser Schlüssel bis zum Auftreten des angegebenen Trennzeichens.

Example

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>amzn-s3-demo-bucket</Name>
  <Prefix></Prefix>
  <Marker></Marker>
  <MaxKeys>1000</MaxKeys>
  <Delimiter>/</Delimiter>
  <IsTruncated>false</IsTruncated>
  <Contents>
    <Key>sample.jpg</Key>
    <LastModified>2011-07-24T19:39:30.000Z</LastModified>
    <ETag>&quot;d1a7fb5eab1c16cb4f7cf341cf188c3d&quot;</ETag>
    <Size>6</Size>
    <Owner>
      <ID>75cc57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
      <DisplayName>displayname</DisplayName>
```

```
    </Owner>
    <StorageClass>STANDARD</StorageClass>
  </Contents>
  <CommonPrefixes>
    <Prefix>photos/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```

Weitere Hinweise zum programmgesteuerten Auflisten von Objektschlüsseln finden Sie unter Programmgesteuertes Auflisten von Objektschlüsseln.

Programmgesteuertes Auflisten von Objektschlüsseln

In Amazon S3 können Schlüssel nach Präfix aufgelistet werden. Sie können ein allgemeines Präfix für die Namen verwandter Schlüssel wählen und diese Schlüssel mit einem Sonderzeichen markieren, das die Hierarchie begrenzt. Sie können dann die Listenoperation verwenden, um Schlüssel hierarchisch auszuwählen und zu durchsuchen. In ähnlicher Weise werden Dateien in einem Dateisystem in Ordnern gespeichert.

Amazon S3 stellt eine Listenoperation bereit, die Ihnen ermöglicht, die in einem Bucket enthaltenen Schlüssel aufzulisten. Schlüssel werden nach Bucket und Präfix für die Auflistung ausgewählt. Angenommen, wir haben einen Bucket namens "`dictionary`", der einen Schlüssel für jedes englische Wort enthält. Sie können einen Aufruf ausführen, um alle Schlüssel in dem Bucket aufzulisten, die mit dem Buchstaben "q" beginnen. Listenergebnisse werden immer in binärer Reihenfolge von UTF -8 zurückgegeben.

SOAPSowohl die Operationen als auch die REST Listenoperation geben ein XML Dokument zurück, das die Namen der passenden Schlüssel und Informationen über das Objekt enthält, das durch die einzelnen Schlüssel identifiziert wird.

> **Note**
>
> SOAPSupport Over HTTP ist veraltet, aber SOAP immer noch verfügbar. HTTPS Neue Amazon S3 S3-Funktionen werden nicht unterstütztSOAP. Anstatt zu verwendenSOAP, empfehlen wir, entweder den REST API oder den zu verwenden AWS SDKs.

Gruppen von Schlüsseln, die ein gemeinsames Präfix haben, abgeschlossen mit einem speziellen Trennzeichen, können in einer Auflistung nach diesem gemeinsamen Präfix ausgegeben werden.

Auf diese Weise können Anwendungen ihre Schlüssel hierarchisch organisieren und durchsuchen, ähnlich wie bei der Organisation Ihrer Dateien in Verzeichnissen in einem Dateisystem.

Um beispielsweise den dictionary-Bucket zu erweitern, um mehr als nur englische Wörter aufzunehmen, könnten Sie Schlüssel bilden, indem Sie jedem Wort ein Präfix für seine Sprache und ein Trennzeichen voranstellen, wie beispielsweise "`French/logical`". Mit Hilfe dieses Namensschemas und der Funktion zur hierarchischen Auflistung könnten Sie eine Liste nur mit französischen Wörtern abrufen. Sie könnten auch auf oberster Ebene eine Liste verfügbarer Sprachen durchsuchen, ohne alle lexikographischen Schlüssel durchlaufen zu müssen. Weitere Informationen über diesen Aspekt der Auflistung finden Sie unter Organisieren von Objekten mit Präfixen.

REST API

Wenn Ihre Bewerbung dies erfordert, können Sie REST Anfragen direkt senden. Sie können eine GET Anfrage senden, um einige oder alle Objekte in einem Bucket zurückzugeben, oder Sie können Auswahlkriterien verwenden, um eine Teilmenge der Objekte in einem Bucket zurückzugeben. Weitere Informationen finden Sie unter GETBucket (List Objects) Version 2 in der Amazon Simple Storage Service API Reference.

Effiziente Implementierung von Listen

Die Listenleistung wird von der Gesamtzahl der Schlüssel in Ihrem Bucket nicht wesentlich beeinflusst. Es ist auch nicht von der Anwesenheit oder Abwesenheit der Argumente `prefix`, `marker`, `maxkeys` oder `delimiter` betroffen.

Durchlaufen mehrseitiger Ergebnisse

Buckets können nahezu unbegrenzt viele Schlüssel enthalten, die vollständigen Ergebnisse einer Listenabfrage können deshalb extrem umfangreich sein. Um große Ergebnismengen zu verwalten, API unterstützt Amazon S3 die Paginierung, um sie in mehrere Antworten aufzuteilen. Jede Antwort für Listenschlüssel gibt eine Seite mit bis zu 1000 Schlüsseln zurück, mit einer Angabe, ob die Antwort gekürzt wurde. Sie senden eine Reihe von Anfragen nach Listenschlüsseln, bis Sie alle Schlüssel erhalten haben. AWS SDKWrapper-Bibliotheken bieten dieselbe Paginierung.

Beispiele

Sollen alle Objekte im Bucket aufgelistet werden, müssen Sie über die Berechtigung `s3:ListBucket` verfügen.

CLI

Objekte auflisten

Im folgenden Beispiel wird der `list-objects` Befehl verwendet, um die Namen aller Objekte im angegebenen Bucket anzuzeigen:

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size: Size}'
```

Im Beispiel wird das `--query` Argument verwendet, um die Ausgabe von `list-objects` nach Schlüsselwert und Größe für jedes Objekt zu filtern

Weitere Informationen zu Objekten finden Sie unter Working with Amazon S3 Objects im Amazon S3 Developer Guide.

- APIEinzelheiten finden Sie ListObjectsunter AWS CLI Befehlsreferenz.

ls

Im folgenden Beispiel werden alle Objekte und Präfixe in einem Bucket mithilfe des `ls` Befehls aufgelistet.

Um diesen Beispielbefehl zu verwenden, ersetzen *amzn-s3-demo-bucket* mit dem Namen Ihres Buckets.

```
$ aws s3 ls s3://amzn-s3-demo-bucket
```

- Weitere Informationen zum Befehl `ls` auf hoher Ebene finden Sie im AWS Command Line Interface Benutzerhandbuch unter Buckets und Objekte auflisten.

PowerShell

Tools für PowerShell

Beispiel 1: Dieser Befehl ruft die Informationen über alle Elemente im Bucket „test-files“ ab.

```
Get-S3Object -BucketName amzn-s3-demo-bucket
```

Beispiel 2: Dieser Befehl ruft die Informationen über das Objekt "sample.txt" aus dem Bucket „test-files“ ab.

```
Get-S3Object -BucketName amzn-s3-demo-bucket -Key sample.txt
```

Beispiel 3: Dieser Befehl ruft die Informationen über alle Elemente mit dem Präfix „sample“ aus dem Bucket „test-files“ ab.

```
Get-S3Object -BucketName amzn-s3-demo-bucket -KeyPrefix sample
```

- APIEinzelheiten finden Sie unter ListObjectsCmdlet-Referenz.AWS Tools for PowerShell

Organisieren von Objekten in der Amazon S3-Konsole mithilfe von Ordnern

In Amazon S3 sind Buckets und Objekte die primären Ressourcen, in denen Objekte in Buckets gespeichert werden. Amazon S3 besitzt eine flache Struktur statt einer Hierarchie, wie Sie sie in der Regel aus Dateisystemen kennen. Um eine möglichst einfache Organisation zu ermöglichen, unterstützt die Amazon-S3-Konsole jedoch das Ordner-Konzept, um Objekte gruppieren zu können. Die Konsole verwendet dazu ein gemeinsames Namenspräfix für die gruppierten Objekte. Mit anderen Worten, die gruppierten Objekte haben Namen, die mit einer bestimmten Zeichenfolge beginnen. Diese gemeinsame Zeichenfolge oder das gemeinsame Präfix ist der Ordnername. Objektnamen werden auch als Schlüsselnamen bezeichnet.

Beispielsweise können Sie einen Ordner namens `photos` in der Konsole erstellen und ein Objekt namens `myphoto.jpg` darin speichern. Das Objekt wird dann mit dem Schlüsselnamen `photos/myphoto.jpg` gespeichert, wobei `photos/` das Präfix ist.

Nachfolgend zwei weitere Beispiele:

- Wenn Sie drei Objekte in Ihrem Bucket haben – `logs/date1.txt`, `logs/date2.txt` und `logs/date3.txt`–, zeigt die Konsole einen Ordner namens `logs` an. Wenn Sie den Ordner in der Konsole öffnen, sehen Sie drei Objekte: `date1.txt`, `date2.txt` und `date3.txt`.
- Wenn Sie ein Objekt namens `photos/2017/example.jpg` haben, zeigt die Konsole den Ordner `photos` an, der den Ordner `2017` enthält. Der Ordner `2017` enthält das Objekt `example.jpg`.

Sie können Ordner innerhalb von Ordnern anlegen, aber keine Buckets innerhalb von Buckets. Sie können Objekte direkt in einen Ordner hochladen und kopieren. Ordner können erstellt, gelöscht und veröffentlicht werden, sie können jedoch nicht umbenannt werden. Objekte können von einem Ordner in einen anderen kopiert werden.

> **Important**
>
> Wenn Sie einen Ordner in Amazon S3 anlegen, erstellt S3 ein 0-Byte-Objekt mit einem Schlüssel, der auf den von Ihnen angegebenen Ordnernamen festgelegt ist. Wenn Sie beispielsweise einen Ordner mit dem Namen `photos` in Ihrem Bucket erstellen, erstellt die Amazon-S3-Konsole ein 0-Byte-Objekt mit dem Schlüssel `photos/`. Die Konsole erstellt dieses Objekt, um das Ordnerkonzept zu unterstützen.
> Die Amazon-S3-Konsole behandelt alle Objekte mit einem Schrägstrich (/) als letztes Zeichen im Schlüsselnamen als Ordner (z. B. `examplekeyname/`). Es ist nicht möglich, ein Objekt über die Amazon-S3-Konsole hochzuladen, dessen Schlüsselname mit einem / endet. Sie können jedoch Objekte, die mit einem abschließenden Namen benannt sind, / mit Amazon S3 hochladen, API indem Sie AWS Command Line Interface (AWS CLI) AWS SDKs, oder REST API verwenden.
> Ein Objekt mit einem angehängten / im Namen wird in der Amazon-S3-Konsole als Ordner angezeigt. Die Amazon-S3-Konsole zeigt für diese Objekte keinen Inhalt und keine Metadaten an. Wenn Sie ein Objekt mit einem angehängten / im Namen kopieren, wird am Zielstandort ein neuer Ordner erstellt, aber die Daten und Metadaten des Objekts werden nicht kopiert.

Themen

- Erstellen eines Ordners
- Veröffentlichen von Ordnern
- Berechnen der Ordnergröße
- Löschen von Ordnern

Erstellen eines Ordners

In diesem Abschnitt wird beschrieben, wie Sie mit der Amazon-S3-Konsole einen Ordner erstellen.

> **Important**
>
> Wenn Ihre Bucket-Richtlinie das Hochladen von Objekten in diesen Bucket ohne Tags, Metadaten oder Empfänger der Zugriffskontrollliste (ACL) verhindert, können Sie mit dem folgenden Verfahren keinen Ordner erstellen. Laden Sie stattdessen einen leeren Ordner hoch und geben Sie die folgenden Einstellungen in der Upload-Konfiguration an.

Einen Ordner erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, in dem Sie einen Ordner erstellen möchten.
4. Wenn Ihre Bucket-Richtlinie das Hochladen von Objekten ohne Verschlüsselung in diesen Bucket verhindert, müssen Sie Enable (Aktivieren) unter Server-side encryption (Serverseitige Verschlüsselung) wählen.
5. Wählen Sie Create folder.
6. Geben Sie einen Namen für den Ordner ein (z. B. **`favorite-pics`**). Wählen Sie dann Create folder (Ordner erstellen).

Veröffentlichen von Ordnern

Wir empfehlen, jeden öffentlichen Zugriff auf Ihre Amazon-S3-Ordner und Buckets zu blockieren, sofern Sie nicht speziell einen öffentlichen Ordner oder Bucket benötigen. Wenn Sie einen Ordner öffentlich machen, kann jeder Benutzer im Internet alle im Ordner gruppierten Objekte Anzeigen.

In der Amazon-S3-Konsole können Sie einen Ordner öffentlich machen. Außerdem können Sie einen Ordner öffentlich machen, indem Sie eine Bucket-Richtlinie erstellen, die den Datenzugriff durch ein Präfix einschränkt. Weitere Informationen finden Sie unter Identity and Access Management für Amazon S3.

> **Warning**
>
> Nachdem Sie einen Ordner in der Amazon-S3-Konsole öffentlich gemacht haben, kann er nicht mehr auf privat gesetzt werden. Stattdessen müssen Sie Berechtigungen für jedes einzelne Objekt im öffentlichen Ordner festlegen, damit die Objekte keinen öffentlichen Zugriff haben. Weitere Informationen finden Sie unter Konfiguration ACLs.

Themen

- Berechnen der Ordnergröße
- Löschen von Ordnern

Berechnen der Ordnergröße

In diesem Abschnitt wird beschrieben, wie Sie mit der Amazon-S3-Konsole die Größe eines Ordners berechnen.

So berechnen Sie die Größe eines Ordners

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, in dem Ihr Ordner gespeichert ist.
4. Aktivieren Sie in der Liste Objects (Objekte) das Kontrollkästchen neben dem Namen des Ordners.
5. Wählen Sie Actions (Aktionen) und dann Calculate total size (Gesamtgröße berechnen) aus.

> **Note**
>
> Wenn Sie die Seite verlassen, sind die Ordnerinformationen (einschließlich der Gesamtgröße) danach nicht mehr verfügbar. Sie müssen die Gesamtgröße erneut berechnen, wenn Sie sie erneut sehen möchten.

> **Important**
>
> Wenn Sie die Aktion Calculate total size (Gesamtgröße berechnen) für bestimmte Objekte oder Ordner in Ihrem Bucket verwenden, berechnet Amazon S3 die Gesamtzahl der Objekte und die Gesamtspeichergröße. Unvollständige oder in Bearbeitung befindliche mehrteilige Uploads und vorherige oder nicht aktuelle Versionen werden jedoch nicht auf die Gesamtzahl der Objekte oder die Gesamtgröße angerechnet. Diese Aktion berechnet nur die Gesamtzahl der Objekte und die Gesamtgröße für die aktuelle oder neueste Version jedes Objekts, das im Bucket gespeichert ist.
>
> Wenn es beispielsweise zwei Versionen eines Objekts in Ihrem Bucket gibt, zählt der Speicherrechner in Amazon S3 diese nur als ein Objekt. Daher kann die Gesamtzahl der Objekte, die in der Amazon S3-Konsole berechnet wird, von der in S3 Storage Lens angezeigten Metrik Object Count und von der in der CloudWatch Amazon-Metrik, gemeldeten Anzahl abweichen`NumberOfObjects`. Ebenso kann die Gesamtspeichergröße auch von der

Metrik „Gesamtspeicher“ in S3 Storage Lens und von der in angezeigten `BucketSizeBytes` Metrik abweichen CloudWatch.

Löschen von Ordnern

In diesem Abschnitt erfahren Sie, wie Sie mit der Amazon-S3-Konsole Ordner aus einem S3-Bucket löschen.

Informationen zu den Funktionen und Preisen von Amazon S3 finden Sie unter Amazon S3.

Ordner aus einem S3-Bucket löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, aus dem Sie Ordner löschen möchten.
3. Aktivieren Sie in der Liste Objects (Objekte) das Kontrollkästchen neben den Ordnern und Objekten, die Sie löschen möchten.
4. Wählen Sie Löschen.
5. Stellen Sie auf der Seite Delete objects (Objekte löschen) sicher, dass die Namen der Ordner, die Sie zum Löschen ausgewählt haben, aufgeführt sind.
6. Geben Sie im Feld Delete objects (Objekte löschen) ein **`delete`**, und wählen Sie Delete objects (Objekte löschen) aus.

> **Warning**
>
> Diese Aktion löscht alle angegebenen Objekte. Warten Sie beim Löschen von Ordnern, bis die Löschaktion abgeschlossen ist, bevor Sie dem Ordner neue Objekte hinzufügen. Andernfalls könnten auch neue Objekte gelöscht werden.

Anzeigen von Objekteigenschaften in der Amazon-S3-Konsole

Sie können die Amazon-S3-Konsole verwenden, um die Eigenschaften eines Objekts anzuzeigen, einschließlich Speicherklasse, Verschlüsselungs-Einstellungen, Markierungen und Metadaten.

Die Eigenschaften für ein Objekt anzeigen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.
3. Wählen Sie in der Liste Objects (Objekte) den Namen des Objekts aus, dessen Eigenschaften Sie anzeigen wollen.

   Die Object overview (Objektübersicht) für Ihr Objekt wird geöffnet. Sie können nach unten scrollen, um die Objekteigenschaften anzuzeigen.
4. Auf der Seite Object overview (Objektübersicht) können Sie die folgenden Eigenschaften für das Objekt konfigurieren.

   > **Note**
   >
   > - Wenn Sie die Eigenschaften Storage Class (Speicherklasse), Encryption (Verschlüsselung) oder Metadata (Metadaten) ändern, wird ein neues Objekt erstellt, um das alte zu ersetzen. Wenn S3-Versioning aktiviert ist, wird eine neue Version des Objekts erstellt, und das vorhandene Objekt wird zu einer älteren Version. Die Rolle, die die Eigenschaft ändert, wird auch Besitzer des neuen Objekts (oder der neuen Objektversion).
   > - Wenn Sie die Eigenschaften Speicherklasse, Verschlüsselung oder Metadaten für ein Objekt mit benutzerdefinierten Tags ändern, benötigen Sie die `s3:GetObjectTagging` entsprechende Berechtigung. Wenn Sie diese Eigenschaften für ein Objekt ändern, das keine benutzerdefinierten Tags hat, aber eine Größe von mehr als 16 MB hat, benötigen Sie auch die `s3:GetObjectTagging` entsprechende Berechtigung.
   >
   >   Wenn die Ziel-Bucket-Richtlinie die `s3:GetObjectTagging` Aktion ablehnt, werden diese Eigenschaften für das Objekt aktualisiert, aber die benutzerdefinierten Tags werden aus dem Objekt entfernt, und Sie erhalten eine Fehlermeldung.

   a. Storage class (Speicherklasse) – Jedem Objekt in Amazon S3 ist eine Speicherklasse zugeordnet. Welche Speicherklasse Sie wählen, ist davon abhängig, wie häufig Sie auf das Objekt zugreifen. Die Standardspeicherklasse für S3-Objekte ist. STANDARD Sie wählen, welche Speicherlasse verwendet werden soll, wenn Sie ein Objekt hochladen. Weitere

Informationen über Speicherklassen finden Sie unter Amazon S3 S3-Speicherklassen verstehen und verwalten.

Um die Speicherklasse nach dem Hochladen eines Objekts zu ändern, wählen Sie Storage class (Speicherklasse). Wählen Sie die gewünschte Speicherklasse aus, und wählen Sie dann Save (Speichern).

b. Serverseitige Verschlüsselungseinstellungen — Sie können serverseitige Verschlüsselung verwenden, um Ihre S3-Objekte zu verschlüsseln. Weitere Informationen finden Sie unter Geben Sie die serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) an oder Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) angeben.

c. Metadata (Metadaten) – Jedes Objekt in Amazon S3 weist eine Gruppe von Namen-Wert-Paaren auf, die seine Metadaten repräsentieren. Weitere Informationen zum Hinzufügen von Metadaten zu einem S3-Objekt finden Sie unter Bearbeiten von Objektmetadaten in der Amazon-S3-Konsole.

d. Tags – Sie kategorisieren den Speicher, indem Sie einem S3-Objekt Tags hinzufügen. Weitere Informationen finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen.

e. Objektsperre, gesetzliche Aufbewahrung und Aufbewahrung — Sie können verhindern, dass ein Objekt gelöscht wird. Weitere Informationen finden Sie unter Sperren von Objekten mit Object Lock.

Kategorisieren des Speichers mithilfe von Markierungen

Markieren Sie Objekte, um Speicher zu kategorisieren. Jeder Tag ist ein Schlüssel/Wert-Paar.

Sie können neuen Objekten Markierungen hinzufügen, wenn Sie sie hochladen, Sie können Markierungen aber auch vorhandenen Objekten hinzufügen.

- Sie können einem Objekt bis zu 10 Markierungen zuordnen. Einem Objekt zugeordnete Markierungen müssen eindeutige Tag-Schlüssel haben.
- Ein Tag-Schlüssel kann maximal 128 Unicode-Zeichen lang sein, und die Tag-Werte können bis zu 256 Unicode-Zeichen lang sein. Amazon S3 S3-Objekt-Tags werden intern in UTF -16 dargestellt. Beachten Sie, dass in UTF -16 Zeichen entweder 1 oder 2 Zeichenpositionen einnehmen.
- Bei Schlüsseln und Werten wird die Groß-/Kleinschreibung berücksichtigt.

- Weitere Informationen zu Tagbeschränkungen finden Sie unter Benutzerdefinierte Tagbeschränkungen im AWS Billing and Cost Management-Benutzerhandbuch. Grundlegende Tag-Einschränkungen finden Sie unter Tag-Einschränkungen im EC2Amazon-Benutzerhandbuch.

Beispiele

Betrachten Sie die folgenden Beispiele für die Markierung:

Example PHIInformationen

Angenommen, ein Objekt enthält geschützte Gesundheitsinformationen (PHI). Sie könnten das Objekt unter Verwendung des folgenden-Schlüssel-Wert-Paares markieren:

```
PHI=True
```

oder

```
Classification=PHI
```

Example Projektdateien

Angenommen, Sie speichern Projektdateien in Ihrem S3-Bucket. Sie könnten diese Objekte mit einem Schlüssel namens `Project` und einem Wert markieren, wie nachfolgend gezeigt:

```
Project=Blue
```

Example Mehrere Markierungen

Sie können einem Objekt mehrere Markierungen hinzufügen, wie nachfolgend gezeigt:

```
Project=x
Classification=confidential
```

Schlüsselnamen-Präfixe und -Markierungen

Mit Schlüsselnamenpräfixe können Sie auch Speicher kategorisieren. Allerdings sind Präfix-basierte Kategorisierungen eindimensional. Sehen Sie sich die folgenden Objektschlüsselnamen an:

```
photos/photo1.jpg
```

```
project/projectx/document.pdf
project/projecty/document2.pdf
```

Dieses Schlüsselnamen haben die Präfixe `photos/`, `project/projectx/` und `project/projecty/`. Diese Präfixe unterstützen eine eindimensionale Kategorisierung. Das bedeutet, alles unter einem Präfix ist eine Kategorie. Beispielsweise identifiziert das Präfix `project/projectx` alle Dokumente, die zu Projekt X gehören.

Mit der Markierung erhalten Sie jetzt eine weitere Dimension. Wenn Sie photo1 in der Kategorie project x anlegen wollen, können Sie das Objekt entsprechend markieren.

Zusätzliche Vorteile

Neben der Datenklassifizierung bietet die Markierung auch noch weitere Vorteile.

- Objekt-Markierungen bieten eine differenzierte Zugriffskontrolle für Berechtigungen. Sie könnten z. B. einem Benutzer Berechtigungen erteilen, nur Objekte mit bestimmten Markierungen zu lesen.
- Objekt-Markierungen unterstützen ein differenziertes Objektlebenszyklusmanagement, bei dem Sie in einer Lebenszyklusregel zusätzlich zum Schlüsselnamenpräfix einen auf Markierungen basierenden Filter angeben können.
- Wenn Sie Amazon-S3-Analysen verwenden, können Sie Filter konfigurieren, um Objekte für die Analyse nach Objekt-Markierungen, nach Schlüsselnamen-Präfix oder nach Präfix und Markierungen zu gruppieren.
- Sie können CloudWatch Amazon-Metriken auch so anpassen, dass Informationen anhand bestimmter Tag-Filter angezeigt werden. Die folgenden Abschnitte stellen Details bereit.

> **Important**
>
> Es ist zulässig, Tags zur Kennzeichnung von Objekten zu verwenden, die vertrauliche Daten enthalten, wie z. B. personenbezogene Daten (PII) oder geschützte Gesundheitsinformationen (PHI). Diese Markierungen sollten jedoch selbst keine vertraulichen Daten enthalten.

Hinzufügen von Objekt-Tag-Sätzen zu mehreren Amazon-S3-Objekten mit einer einzigen Anfrage

Zum Hinzufügen von Objekt-Tag-Mengen zu mehr als einem Amazon-S3-Objekt mit einer einzelnen Anforderung können Sie S3-Batchoperationen verwenden. Sie stellen S3-Batchoperationen eine Liste

von Objekten zur Verfügung, für die Operationen ausgeführt werden sollen. S3 Batch Operations ruft die entsprechende API Operation auf, um die angegebene Operation auszuführen. Ein einzelner Batch-Vorgangsauftrag kann die angegebene Operation für Milliarden von Objekten ausführen, die Exabytes von Daten enthalten.

Die Funktion „S3-Batchoperationen“ verfolgt den Fortschritt, versendet Benachrichtigungen und speichert einen detaillierten Abschlussbericht zu allen Aktionen. Sie profitieren von einer vollständig verwalteten, prüfbaren und serverlosen Umgebung. Sie können S3 Batch Operations über die Amazon S3 S3-Konsole AWS CLI, AWS SDKs, oder verwenden RESTAPI. Weitere Informationen finden Sie unter the section called “Grundlagen von BatchVorgänge”.

Weitere Informationen über Objekt-Markierungen finden Sie unter Verwalten von Objekt-Markierungen.

APIOperationen im Zusammenhang mit Objekt-Tagging

Amazon S3 unterstützt die folgenden API Operationen, die speziell für das Tagging von Objekten bestimmt sind:

Objektoperationen API

- PUTObjekt-Tagging — Ersetzt Tags auf einem Objekt. Sie geben die Markierungen im Anfragerumpf an. Es gibt zwei unterschiedliche Szenarien, in denen diese Methode für die Verwaltung von Objekt-Tags verwendet wirdAPI.
  - Objekt hat keine Tags — Damit können API Sie einem Objekt eine Reihe von Tags hinzufügen (das Objekt hat keine vorherigen Tags).
  - Das Objekt hat einen Satz vorhandener Tags — Um den vorhandenen Tagsatz zu ändern, müssen Sie zuerst den vorhandenen Tagsatz abrufen, ihn auf der Client-Seite ändern und diesen dann verwenden, API um den Tagsatz zu ersetzen.

    > **Note**
    >
    > Wenn Sie diese Anforderung mit einer leeren Tag-Menge senden, löscht Amazon S3 die vorhandene Tag-Menge für das Objekt. Wenn Sie diese Methode verwenden, wird Ihnen eine Tier-1-Anfrage (PUT) in Rechnung gestellt. Weitere Informationen finden Sie unter Amazon S3 – Preise.
    > Die DELETEObjekt-Tagging-Anfrage wird bevorzugt, da sie dasselbe Ergebnis erzielt, ohne dass Gebühren anfallen.

- GETObjekt-Tagging — Gibt den Tagsatz zurück, der einem Objekt zugeordnet ist. Amazon S3 gibt die Objekt-Markierungen im Antworttext zurück.
- DELETEObjekt-Tagging — Löscht den Tagsatz, der einem Objekt zugeordnet ist.

Andere API Operationen, die Tagging unterstützen

- PUTObjekt erstellen und mehrteiligen Upload initiieren — Sie können Tags angeben, wenn Sie Objekte erstellen. Sie geben Markierungen unter Verwendung des Anfrage-Headers `x-amz-tagging` an.
- GETObjekt — Anstatt den Tag-Satz zurückzugeben, gibt Amazon S3 die Anzahl der Objekt-Tags im `x-amz-tag-count` Header zurück (nur wenn der Anforderer die Berechtigung hat, Tags zu lesen), da die Größe der Header-Antwort auf 8 KB begrenzt ist. Wenn Sie die Tags anzeigen möchten, stellen Sie eine weitere Anfrage für den GETObjekt-Tagging-VorgangAPI.
- POSTObjekt — Sie können in Ihrer POST Anfrage Tags angeben.

  Solange die Tags in Ihrer Anfrage die Größenbeschränkung für den HTTP Anforderungsheader von 8 KB nicht überschreiten, können Sie sie verwenden, `PUT Object` API um Objekte mit Tags zu erstellen. Wenn die von Ihnen angegebenen Tags die Größenbeschränkung für Header überschreiten, können Sie diese POST Methode verwenden, bei der Sie die Tags in den Hauptteil aufnehmen.

  PUTObjekt — Kopieren — Sie können `x-amz-tagging-directive` in Ihrer Anfrage angeben, dass Amazon S3 entweder die Tags kopiert (Standardverhalten) oder die Tags durch einen neuen Satz von Tags ersetzt, die in der Anfrage bereitgestellt werden.

Beachten Sie Folgendes:

- Die S3-Objektmarkierung ist hochgradig konsistent. Weitere Informationen finden Sie unter Amazon S3-Datenkonsistenzmodell.

Zusätzliche Konfigurationen

Dieser Abschnitt erklärt, was die Objektmarkierung für andere Konfigurationen bedeutet.

Objektmarkierung und Lebenszyklusverwaltung

In der Bucket-Lebenszyklus-Konfiguration können Sie einen Filter angeben, um eine Untermenge von Objekten auszuwählen, auf die die Regel anzuwenden ist. Sie können einen Filter basierend auf den Schlüsselnamenpräfixen, Objekt-Markierungen oder beidem angeben.

Angenommen, Sie speichern Fotos (Rohdaten und im fertigen Format) in Ihrem Amazon-S3-Bucket. Sie könnten diese Objekte wie folgt markieren:

```
phototype=raw
or
phototype=finished
```

Sie könnten festlegen, dass die Rohdaten der Fotos irgendwann nach der Erstellung in S3 Glacier archiviert werden. Sie können eine Lebenszyklusregel mit einem Filter konfigurieren, der die Untermenge der Objekte mit dem Schlüsselnamenpräfix (`photos/`) identifiziert, die ein spezifisches Tag (`phototype=raw`) haben.

Weitere Informationen finden Sie unter Verwaltung des Lebenszyklus von Objekten.

Objektmarkierung und -replikation

Wenn Sie auf Ihrem Bucket die Replikation konfiguriert haben, repliziert Amazon S3 Markierungen, sofern Sie Amazon S3 die Berechtigung zum Lesen der Markierungen erteilt haben. Weitere Informationen finden Sie unter Überblick über die Einrichtung der Live-Replikation.

Ereignisbenachrichtigungen einer Objektmarkierung

Sie können eine Amazon-S3-Ereignisbenachrichtigung einrichten, um benachrichtigt zu werden, wenn eine Objektmarkierung hinzugefügt oder aus einem Objekt gelöscht wird. Der `s3:ObjectTagging:Put` Ereignistyp benachrichtigt Sie, wenn sich ein Tag PUT auf einem Objekt befindet oder wenn ein vorhandenes Tag aktualisiert wird. Der `s3:ObjectTagging:Delete`-Ereignistyp benachrichtigt Sie, wenn ein Tag aus einem Objekt entfernt wird. Weitere Informationen finden Sie unter Aktivieren von Ereignisbenachrichtigungen.

Weitere Informationen über die Objektmarkierung finden Sie in den folgenden Themen:

Themen

- Markierungs- und Zugriffskontrollrichtlinien
- Verwalten von Objekt-Markierungen

Markierungs- und Zugriffskontrollrichtlinien

Sie können außerdem Berechtigungsrichtlinien (Bucket- und Benutzerrichtlinien) verwenden, um Berechtigungen für das Objekt-Tagging zu verwalten. Informationen über Richtlinienaktionen finden Sie in den folgenden Themen:

- Objektoperationen
- Bucket-Operationen

Objekt-Markierungen bieten eine differenzierte Zugriffskontrolle für die Verwaltung von Berechtigungen. Sie können bedingte Berechtigungen basierend auf Objekt-Markierungen erteilen. Amazon S3 unterstützt die folgenden Bedingungsschlüssel, die Sie verwenden können, um bedingte Berechtigungen basierend auf Objekt-Markierungen zu erteilen.

- `s3:ExistingObjectTag/<tag-key>` – Verwenden Sie diesen Bedingungsschlüssel, um zu überprüfen, ob ein vorhandenes Objekt-Tag den spezifischen Tag-Schlüssel und -Wert besitzt.

  > **Note**
  >
  > Wenn Sie Berechtigungen für die `PUT Object`- und `DELETE Object`-Operationen erteilen, wird dieser Bedingungsschlüssel nicht unterstützt. Dies bedeutet, dass Sie keine Richtlinie erstellen können, um einem Benutzer zu gestatten, ein Objekt basierend auf seinen vorhandenen Markierungen zu löschen oder zu überschreiben.

- `s3:RequestObjectTagKeys` – Verwenden Sie diesen Bedingungsschlüssel, um die Tag-Schlüssel einzuschränken, die Sie für Objekte zulassen wollen. Dies ist nützlich, wenn Sie Objekten mithilfe der PutObjectTagging POST Objektanforderungen und und Tags hinzufügen. PutObject
- `s3:RequestObjectTag/<tag-key>` – Verwenden Sie diesen Bedingungsschlüssel, um die Tag-Schlüssel und -Werte einzuschränken, die Sie für Objekte zulassen wollen. Dies ist nützlich, wenn Sie Objekten mithilfe der Anfragen PutObjectTagging and und und PutObject POST Bucket Tags hinzufügen.

Eine vollständige Liste der für den Amazon-S3-Service spezifischen Bedingungsschlüssel finden Sie unter Beispiele für Bucket-Richtlinien mit Bedingungsschlüsseln. Die folgenden Berechtigungsrichtlinien zeigen, wie die Objektmarkierung eine differenzierte Zugriffsberechtigungsverwaltung ermöglicht.

Example 1: Einem Benutzer nur das Lesen von Objekten gestatten, die einen bestimmten Tag-Schlüssel und -Wert besitzen

Die folgende Berechtigungsrichtlinie beschränkt einen Benutzer darauf, nur Objekte zu lesen, die den Tag-Schlüssel und -Wert `environment: production` haben. Diese Richtlinie verwendet den `s3:ExistingObjectTag`-Bedingungsschlüssel, um den Tag-Schlüssel und -Wert anzugeben.

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Principal": {
      "AWS": [
        "arn:aws:iam::111122223333:role/JohnDoe"
      ]
    },
    "Effect": "Allow",
    "Action": ["s3:GetObject", "s3:GetObjectVersion"],
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
    "Condition": {
      "StringEquals":
        {"s3:ExistingObjectTag/environment": "production"}
    }
  }
  ]
}
```

Example 2: Einschränken, welche Objekt-Tag-Schlüssel Benutzer hinzufügen können

Die folgende Berechtigungsrichtlinie erteilt einem Benutzer die Berechtigungen, die `s3:PutObjectTagging`-Aktion auszuführen, die dem Benutzer gestattet, einem vorhandenen Objekt Markierungen hinzuzufügen. Die Bedingung verwendet den Bedingungsschlüssel `s3:RequestObjectTagKeys`, um die zulässigen Tag-Schlüssel zu spezifizieren, z. B. `Owner` oder `CreationDate`. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Erstellen einer Bedingung, die mehrere Schlüsselwerte testet.

Die Richtlinie stellt sicher, dass jeder in der Anfrage angegebene Tag-Schlüssel ein autorisierter Tag-Schlüssel ist. Der `ForAnyValue`-Qualifizierer in der Bedingung stellt sicher, dass mindestens einer der spezifizierten Schlüssel in der Anfrage enthalten ist.

```
{
```

```
   "Version": "2012-10-17",
  "Statement": [
    {"Principal":{"AWS":[
            "arn:aws:iam::111122223333:role/JohnDoe"
         ]
       },
 "Effect": "Allow",
      "Action": [
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ],
      "Condition": {"ForAnyValue:StringEquals": {"s3:RequestObjectTagKeys": [
            "Owner",
            "CreationDate"
          ]
        }
      }
    }
  ]
}
```

Example 3: Einen spezifischen Tag-Schlüssel und -Wert verlangen, wenn Benutzern erlaubt wird, Objekt-Tags hinzuzufügen

Die folgende Beispielrichtlinie erteilt einem Benutzer die Berechtigungen, die `s3:PutObjectTagging`-Aktion auszuführen, die dem Benutzer gestattet, einem vorhandenen Objekt Markierungen hinzuzufügen. Die Bedingung verlangt, dass der Benutzer einen spezifischen Tag-Schlüssel (z. B. *`Project`*) mit dem Wert *`X`* angibt.

```
{
   "Version": "2012-10-17",
  "Statement": [
    {"Principal":{"AWS":[
       "arn:aws:iam::111122223333:user/JohnDoe"
        ]
       },
      "Effect": "Allow",
      "Action": [
        "s3:PutObjectTagging"
      ],
      "Resource": [
```

```
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ],
      "Condition": {"StringEquals": {"s3:RequestObjectTag/Project": "X"
        }
      }
    }
  ]
}
```

Verwalten von Objekt-Markierungen

In diesem Abschnitt wird erklärt, wie Sie Objekt-Tags mit dem AWS SDKs für Java und verwalten können. NEToder die Amazon S3 S3-Konsole.

Das Markieren von Objekten ermöglicht Ihnen, Speicher zu kategorisieren. Jedes Tag ist ein Schlüssel-Wert-Paar, für das folgende Regeln gelten:

- Sie können einem Objekt bis zu 10 Markierungen zuordnen. Einem Objekt zugeordnete Markierungen müssen eindeutige Tag-Schlüssel haben.
- Ein Tag-Schlüssel kann maximal 128 Unicode-Zeichen lang sein, und die Tag-Werte können bis zu 256 Unicode-Zeichen lang sein. Amazon S3 S3-Objekt-Tags werden intern in UTF -16 dargestellt. Beachten Sie, dass in UTF -16 Zeichen entweder 1 oder 2 Zeichenpositionen einnehmen.
- Bei Schlüsseln und Werten wird die Groß-/Kleinschreibung berücksichtigt.

Weitere Informationen über Objekt-Markierungen finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen. Weitere Informationen zu Tag-Einschränkungen finden Sie unter Einschränkungen benutzerdefinierter Markierungen im AWS Billing and Cost Management Benutzerhandbuch.

Verwenden der S3-Konsole

Hinzufügen von Markern zu einem Objekt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets und dann den Tab Buckets für allgemeine Zwecke aus. Navigieren Sie zu dem Amazon S3 S3-Bucket oder -Ordner, der die Objekte enthält, die Sie ändern möchten.

3. Aktivieren Sie das Kontrollkästchen links neben den Namen der Objekte, die Sie ändern möchten.
4. Wählen Sie im Menü Aktionen die Option Markierungen bearbeiten.
5. Überprüfen Sie die aufgelisteten Objekte und wählen Sie Markierungen hinzufügen.
6. Jedes Objekt-Tag ist ein Schlüssel-Wert-Paar. Geben Sie einen Key (Schlüssel) und einen Value (Wert) ein. Um ein weiteres Tag hinzuzufügen, wählen Sie Add Tag (Tag hinzufügen).

   Sie können bis zu 10 Marker für ein Objekt eingeben.
7. Wählen Sie Save Changes (Änderungen speichern).

   Amazon S3 fügt die Markierungen zu den angegebenen Objekten hinzu.

Weitere Informationen finden Sie auch unter Anzeigen von Objekteigenschaften in der Amazon-S3-Konsole und Objekte hochladen in diesem Handbuch.

Verwenden Sie den AWS SDKs

Java

Das folgende Beispiel zeigt, wie Sie mithilfe von Tags für ein neues Objekt festlegen und Tags für ein vorhandenes Objekt abrufen oder ersetzen können. AWS SDK for Java Weitere Informationen über das Markieren von Objekten finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen. Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

public class ManagingObjectTags {
```

```
   public static void main(String[] args) {
       Regions clientRegion = Regions.DEFAULT_REGION;
       String bucketName = "*** Bucket name ***";
       String keyName = "*** Object key ***";
       String filePath = "*** File path ***";

       try {
           AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                   .withCredentials(new ProfileCredentialsProvider())
                   .withRegion(clientRegion)
                   .build();

           // Create an object, add two new tags, and upload the object to Amazon
S3.
           PutObjectRequest putRequest = new PutObjectRequest(bucketName, keyName,
new File(filePath));
           List<Tag> tags = new ArrayList<Tag>();
           tags.add(new Tag("Tag 1", "This is tag 1"));
           tags.add(new Tag("Tag 2", "This is tag 2"));
           putRequest.setTagging(new ObjectTagging(tags));
           PutObjectResult putResult = s3Client.putObject(putRequest);

           // Retrieve the object's tags.
           GetObjectTaggingRequest getTaggingRequest = new
GetObjectTaggingRequest(bucketName, keyName);
           GetObjectTaggingResult getTagsResult =
s3Client.getObjectTagging(getTaggingRequest);

           // Replace the object's tags with two new tags.
           List<Tag> newTags = new ArrayList<Tag>();
           newTags.add(new Tag("Tag 3", "This is tag 3"));
           newTags.add(new Tag("Tag 4", "This is tag 4"));
           s3Client.setObjectTagging(new SetObjectTaggingRequest(bucketName,
keyName, new ObjectTagging(newTags)));
       } catch (AmazonServiceException e) {
           // The call was transmitted successfully, but Amazon S3 couldn't process
           // it, so it returned an error response.
           e.printStackTrace();
       } catch (SdkClientException e) {
           // Amazon S3 couldn't be contacted for a response, or the client
           // couldn't parse the response from Amazon S3.
           e.printStackTrace();
       }
   }
```

```
}
```

.NET

Das folgende Beispiel zeigt, wie Sie mithilfe von AWS SDK for .NET die Tags für ein neues Objekt festlegen und die Tags für ein vorhandenes Objekt abrufen oder ersetzen können. Weitere Informationen über das Markieren von Objekten finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen.

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem AWS SDK for. NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    public class ObjectTagsTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** key name for the new object ***";
        private const string filePath = @"*** file path ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            PutObjectWithTagsTestAsync().Wait();
        }

        static async Task PutObjectWithTagsTestAsync()
        {
            try
            {
                // 1. Put an object with tags.
```

```
                var putRequest = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
                    FilePath = filePath,
                    TagSet = new List<Tag>{
                        new Tag { Key = "Keyx1", Value = "Value1"},
                        new Tag { Key = "Keyx2", Value = "Value2" }
                    }
                };

                PutObjectResponse response = await
client.PutObjectAsync(putRequest);
                // 2. Retrieve the object's tags.
                GetObjectTaggingRequest getTagsRequest = new GetObjectTaggingRequest
                {
                    BucketName = bucketName,
                    Key = keyName
                };

                GetObjectTaggingResponse objectTags = await
client.GetObjectTaggingAsync(getTagsRequest);
                for (int i = 0; i < objectTags.Tagging.Count; i++)
                    Console.WriteLine("Key: {0}, Value: {1}",
objectTags.Tagging[i].Key, objectTags.Tagging[i].Value);


                // 3. Replace the tagset.

                Tagging newTagSet = new Tagging();
                newTagSet.TagSet = new List<Tag>{
                    new Tag { Key = "Key3", Value = "Value3"},
                    new Tag { Key = "Key4", Value = "Value4" }
                };


                PutObjectTaggingRequest putObjTagsRequest = new
PutObjectTaggingRequest()
                {
                    BucketName = bucketName,
                    Key = keyName,
                    Tagging = newTagSet
                };
```

```
                PutObjectTaggingResponse response2 = await
 client.PutObjectTaggingAsync(putObjTagsRequest);

                // 4. Retrieve the object's tags.
                GetObjectTaggingRequest getTagsRequest2 = new
 GetObjectTaggingRequest();
                getTagsRequest2.BucketName = bucketName;
                getTagsRequest2.Key = keyName;
                GetObjectTaggingResponse objectTags2 = await
 client.GetObjectTaggingAsync(getTagsRequest2);
                for (int i = 0; i < objectTags2.Tagging.Count; i++)
                    Console.WriteLine("Key: {0}, Value: {1}",
 objectTags2.Tagging[i].Key, objectTags2.Tagging[i].Value);

            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine(
                        "Error encountered ***. Message:'{0}' when writing an
 object"
                        , e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine(
                    "Encountered an error. Message:'{0}' when writing an object"
                    , e.Message);
            }
        }
    }
}
```

Laden Sie Objekte mit vorsignierten Objekten herunter und laden Sie sie hoch URLs

Sie können presigned verwendenURLs, um zeitlich begrenzten Zugriff auf Objekte in Amazon S3 zu gewähren, ohne Ihre Bucket-Richtlinie zu aktualisieren. Ein Presigned URL kann in einem Browser eingegeben oder von einem Programm zum Herunterladen eines Objekts verwendet werden. Die vom Vorsignierten verwendeten Anmeldeinformationen URL sind die des AWS Benutzers, der das generiert hat. URL

Sie können Presigned auch verwendenURLs, um es jemandem zu ermöglichen, ein bestimmtes Objekt in Ihren Amazon S3 S3-Bucket hochzuladen. Dies ermöglicht einen Upload, ohne dass eine andere Partei über AWS Sicherheitsanmeldedaten oder -berechtigungen verfügen muss. Wenn ein Objekt mit demselben Schlüssel bereits im Bucket vorhanden ist, wie im vorsignierten Bucket angegebenURL, ersetzt Amazon S3 das vorhandene Objekt durch das hochgeladene Objekt.

Sie können das vorsignierte URL Objekt bis zum Ablaufdatum und der Uhrzeit mehrfach verwenden.

Wenn Sie ein vorsigniertes Dokument erstellenURL, müssen Sie Ihre Sicherheitsanmeldedaten angeben und anschließend Folgendes angeben:

- Ein Amazon-S3-Bucket
- Ein Objektschlüssel (Herunterladen: dieses Objekt wird sich in Ihrem Amazon-S3-Bucket befinden; Hochladen: der Name der hochzuladenden Datei)
- Eine HTTP Methode (`GET`zum Herunterladen von Objekten oder `PUT` zum Hochladen)
- Ein Ablaufzeitintervall

Derzeit unterstützt Amazon S3 URLs Presigned die Verwendung der folgenden Algorithmen zur Prüfung der Datenintegrität nicht (CRC32, CRC32C, SHA-1, SHA-256), wenn Sie Objekte hochladen. Um die Integrität Ihres Objekts nach dem Hochladen zu überprüfen, können Sie eine MD5 Zusammenfassung des Objekts, wenn Sie es mit einem vorsignierten Objekt hochladen. URL Weitere Informationen über die Objektintegrität finden Sie unter Überprüfung der Objektintegrität.

Themen

- Wer kann ein vorsigniertes erstellen URL
- Ablaufzeit für vorsignierte URLs
- Einschränkung vordefinierter Funktionen URL
- Objekte mit vorsignierten teilen URLs
- Objekte mit vorsigniertem Inhalt hochladen URLs

Wer kann ein vorsigniertes erstellen URL

Jeder mit gültigen Sicherheitsanmeldedaten kann eine URL vorsignierte erstellen. Damit jedoch jemand erfolgreich auf ein Objekt zugreifen kann, URL muss das vorsignierte Objekt von einer Person erstellt worden sein, die berechtigt ist, den Vorgang auszuführen, auf dem das Vorsignierte URL basiert.

Die folgenden Typen von Anmeldeinformationen können Sie verwenden, um ein vorsigniertes Objekt zu erstellen: URL

- IAMInstanzprofil — Gültig bis zu 6 Stunden.
- AWS Security Token Service: Bei einer Unterzeichnung mit langfristigen Sicherheitsanmeldedaten bis zu 36 Stunden oder für die Dauer der temporären Anmeldeinformationen gültig, je nachdem, was zuerst endet.
- IAMuser — Gültig bis zu 7 Tage, wenn Sie AWS Signature Version 4 verwenden.

  Um ein vorsigniertes Dokument zu erstellenURL, das bis zu 7 Tage gültig ist, delegieren Sie zunächst die IAM Benutzeranmeldedaten (den Zugriffsschlüssel und den geheimen Schlüssel) an die Methode, mit der Sie das Vorsignierte erstellen. URL

> **Note**
>
> Wenn Sie eine vorsignierte Datei URL mit temporären Anmeldeinformationen erstellt haben, URL läuft diese ab, wenn die Anmeldeinformationen ablaufen. Im Allgemeinen URL läuft ein vorsigniertes ab, wenn die Anmeldeinformationen, mit denen Sie es erstellt haben, gesperrt, gelöscht oder deaktiviert werden. Dies gilt auch dann, wenn das mit einer URL späteren Ablaufzeit erstellt wurde. Informationen zur Gültigkeitsdauer temporärer Sicherheitsanmeldedaten finden Sie unter AWS STS APIOperationen vergleichen im IAMBenutzerhandbuch.

Ablaufzeit für vorsignierte URLs

Ein vorsigniertes URL Dokument bleibt für den Zeitraum gültig, der bei der URL Generierung angegeben wurde. Wenn Sie ein URL mit der Amazon S3 S3-Konsole vorsigniertes erstellen, kann die Ablaufzeit zwischen 1 Minute und 12 Stunden festgelegt werden. Wenn Sie das AWS CLI oder verwenden AWS SDKs, kann die Ablaufzeit auf bis zu 7 Tage festgelegt werden.

Wenn Sie ein vorsigniertes URL mithilfe eines temporären Tokens erstellt haben, URL läuft das ab, wenn das Token abläuft. Im Allgemeinen URL läuft ein vorsigniertes ab, wenn die Anmeldeinformationen, mit denen Sie es erstellt haben, gesperrt, gelöscht oder deaktiviert werden. Dies gilt auch dann, wenn das mit einer URL späteren Ablaufzeit erstellt wurde. Weitere Informationen darüber, wie sich die von Ihnen verwendeten Anmeldeinformationen auf die Ablaufzeit auswirken, finden Sie unter Wer kann ein vorsigniertes erstellen URL.

Amazon S3 überprüft das Ablaufdatum und die Uhrzeit eines Signierten zum URL Zeitpunkt der HTTP Anfrage. Beginnt ein Client beispielsweise mit dem Herunterladen einer großen Datei unmittelbar vor der Ablaufzeit, wird der Download auch dann fortgesetzt, wenn die Ablaufzeit während des Downloads verstreicht. Wenn die Verbindung jedoch unterbrochen wird und der Client versucht, den Download nach Ablauf der Zeit erneut zu starten, schlägt der Download fehl.

Einschränkung vordefinierter Funktionen URL

Die Funktionen einer vorsignierten Datei URL werden durch die Berechtigungen des Benutzers begrenzt, der sie erstellt hat. Im Wesentlichen URLs handelt es sich bei vorsignierten Token um Inhaber-Token, die denjenigen Zugriff gewähren, die sie besitzen. Daher empfehlen wir Ihnen, sie angemessen zu schützen. Im Folgenden finden Sie einige Methoden, mit denen Sie die Verwendung Ihrer vorsignierten Daten einschränken können. URLs

AWS Signatur Version 4 (SigV4)

Um ein bestimmtes Verhalten bei der Authentifizierung vorsignierter URL Anfragen mithilfe von AWS Signature Version 4 (Sigv4) durchzusetzen, können Sie Bedingungsschlüssel in Bucket-Richtlinien und Access-Point-Richtlinien verwenden. Die folgende Bucket-Richtlinie verwendet die `s3:signatureAge` Bedingung beispielsweise, um jede vorsignierte Amazon S3 URL S3-Anfrage für Objekte in der *`amzn-s3-demo-bucket`* Bucket, wenn die Signatur älter als 10 Minuten ist. Um dieses Beispiel zu verwenden, ersetzen Sie *`Platzhalter für Benutzereingaben`* mit Ihren eigenen Informationen.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Deny a presigned URL request if the signature is more than 10 min
 old",
         "Effect": "Deny",
         "Principal": {"AWS":"*"},
         "Action": "s3:*",
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
         "Condition": {
            "NumericGreaterThan": {
               "s3:signatureAge": 600000
            }
         }
      }
   ]
```

```
}
```

Weitere Informationen zu Richtlinienschlüsseln im Zusammenhang mit AWS Signature Version 4 finden Sie unter AWS Signature Version 4-Authentifizierung in der Amazon Simple Storage API Service-Referenz.

Beschränkung der Netzwege

Wenn Sie die Nutzung des vorsignierten URLs und des gesamten Amazon S3 S3-Zugriffs auf bestimmte Netzwerkpfade einschränken möchten, können Sie Richtlinien AWS Identity and Access Management (IAM) schreiben. Sie können diese Richtlinien für den IAM Principal, der den Anruf tätigt, für den Amazon S3 S3-Bucket oder für beide festlegen.

Eine Netzwerkpfadbeschränkung für den IAM Prinzipal erfordert, dass der Benutzer mit diesen Anmeldeinformationen Anfragen vom angegebenen Netzwerk aus stellt. Eine Einschränkung des Buckets oder des Zugriffspunkts erfordert, dass alle Anfragen an diese Ressource aus dem angegebenen Netz stammen. Diese Einschränkungen gelten auch außerhalb des URL vordefinierten Szenarios.

Welchen IAM globalen Bedingungsschlüssel Sie verwenden, hängt vom Typ des Endpunkts ab. Wenn Sie den öffentlichen Endpunkt für Amazon S3 verwenden, verwenden Sie `aws:SourceIp`. Wenn Sie einen Virtual Private Cloud (VPC) -Endpunkt für Amazon S3 verwenden, verwenden Sie `aws:SourceVpc` oder`aws:SourceVpce`.

Die folgende IAM Richtlinienerklärung verlangt, dass der Principal AWS nur aus dem angegebenen Netzwerkbereich zugreift. Mit dieser Richtlinie müssen alle Zugriffe von diesem Bereich ausgehen. Dies gilt auch für den Fall, dass jemand ein URL für Amazon S3 vorsigniertes Gerät verwendet. Um dieses Beispiel zu verwenden, ersetzen Sie *`Platzhalter für Benutzereingaben`* mit Ihren eigenen Informationen.

```
{
    "Sid": "NetworkRestrictionForIAMPrincipal",
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
    "Condition": {
        "NotIpAddressIfExists": {"aws:SourceIp": "IP-address-range"},
        "BoolIfExists": {"aws:ViaAWSService": "false"}
    }
}
```

Objekte mit vorsignierten teilen URLs

Standardmäßig sind alle Amazon-S3-Objekte privat und nur der Objekteigentümer hat die Berechtigung, darauf zuzugreifen. Der Objekteigentümer kann jedoch Objekte mit anderen teilen, indem er ein URL vorsigniertes Objekt erstellt. Ein vorsignierter URL Benutzer verwendet Sicherheitsanmeldedaten, um zeitlich begrenzte Berechtigungen zum Herunterladen von Objekten zu gewähren. Das URL kann in einem Browser eingegeben oder von einem Programm zum Herunterladen des Objekts verwendet werden. Die vom Vorsignierten verwendeten Anmeldeinformationen URL sind die des AWS Benutzers, der das URL generiert hat.

Allgemeine Informationen zu Presigned URLs finden Sie unter. Laden Sie Objekte mit vorsignierten Objekten herunter und laden Sie sie hoch URLs

Sie können mithilfe der Amazon S3 S3-Konsole, des AWS Explorers URL für Visual Studio (Windows) oder AWS Toolkit for Visual Studio Code ein Objekt erstellen, das für die gemeinsame Nutzung eines Objekts vorsigniert ist, ohne Code schreiben zu müssen. Sie können ein vorsigniertes Objekt auch URL programmgesteuert generieren, indem Sie die AWS Command Line Interface ()AWS CLI oder die verwenden. AWS SDKs

Verwenden der S3-Konsole

Sie können die Amazon S3 S3-Konsole verwenden, um ein Objekt zu generieren, das URL für die gemeinsame Nutzung vorsigniert ist, indem Sie die folgenden Schritte ausführen. Wenn Sie die Konsole verwenden, beträgt die maximale Ablaufzeit für ein vorsigniertes URL Objekt 12 Stunden ab dem Zeitpunkt der Erstellung.

Um mit der Amazon S3 S3-Konsole ein URL vorsigniertes zu generieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Buckets-Liste den Namen des Buckets aus, der das Objekt enthält, für das Sie eine URL Vorsignierung wünschen.
4. Wählen Sie in der Objektliste das Objekt aus, für das Sie ein URL vorsigniertes Objekt erstellen möchten.
5. Wählen Sie im Menü „Objektaktionen“ die Option „Mit einem URL vorsignierten Objekt teilen“.
6. Geben Sie an, wie lange das Vorsignierte gültig sein URL soll.

7. Wählen Sie „URLVorsigniertes erstellen“.
8. Wenn eine Bestätigung erscheint, wird diese automatisch in Ihre Zwischenablage kopiert. URL Sie werden eine Schaltfläche sehen, mit der Sie das Vorsignierte kopieren könnenURL, falls Sie es erneut kopieren müssen.

Mit dem AWS CLI

Der folgende AWS CLI Beispielbefehl generiert ein URL für die gemeinsame Nutzung eines Objekts vorsigniertes Objekt aus einem Amazon S3 S3-Bucket. Wenn Sie den verwenden AWS CLI, beträgt die maximale Ablaufzeit für ein vorsigniertes Objekt URL 7 Tage ab dem Zeitpunkt der Erstellung. Um dieses Beispiel zu verwenden, ersetzen Sie *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
aws s3 presign s3://amzn-s3-demo-bucket/mydoc.txt --expires-in 604800
```

> **Note**
>
> Für alle, die nach dem 20. März 2019 AWS-Regionen gestartet wurden, müssen Sie das `endpoint-url` und `AWS-Region` zusammen mit der Anfrage angeben. Eine Liste aller Amazon-S3-Regionen und Endpunkte finden Sie unter Regionen und Endpunkte in der Allgemeinen AWS -Referenz.

```
aws s3 presign s3://amzn-s3-demo-bucket/mydoc.txt --expires-in 604800 --region af-south-1 --endpoint-url https://s3.af-south-1.amazonaws.com
```

Weitere Informationen finden Sie unter presignin der AWS CLI Befehlsreferenz.

Mit dem AWS SDKs

Beispiele für die Verwendung von AWS SDKs zum Generieren eines Vorsignierten URL für die gemeinsame Nutzung eines Objekts finden Sie unter Erstellen eines URL für Amazon S3 vorsignierten Objekts mithilfe von. AWS SDK

Wenn Sie das verwenden AWS SDKs, um ein vorsigniertes Objekt zu generierenURL, beträgt die maximale Ablaufzeit 7 Tage ab dem Zeitpunkt der Erstellung.

> **Note**
>
> Für alle, die nach dem 20. März 2019 AWS-Regionen gestartet wurden, müssen Sie das `endpoint-url` und `AWS-Region` zusammen mit der Anfrage angeben. Eine Liste aller Amazon-S3-Regionen und Endpunkte finden Sie unter Regionen und Endpunkte in der Allgemeinen AWS -Referenz.

> **Note**
>
> Wenn Sie das verwenden AWS SDKs, muss das Tagging-Attribut ein Header und kein Abfrageparameter sein. Alle anderen Attribute können als Parameter für das URL vorsignierte Attribut übergeben werden.

Verwenden von AWS Toolkit for Visual Studio (Windows)

> **Note**
>
> Derzeit AWS Toolkit for Visual Studio unterstützt der Visual Studio für Mac nicht.

1. Installieren Sie das AWS Toolkit for Visual Studio mithilfe der folgenden Anweisungen unter Installation und Einrichtung des Toolkit for Visual Studio im AWS Toolkit for Visual Studio Benutzerhandbuch.
2. Connect Sie AWS mithilfe der folgenden Schritte eine Verbindung zu her: Herstellen einer Verbindung zu AWS im AWS Toolkit for Visual Studio Benutzerhandbuch.
3. Doppelklicken Sie in der linken Seitenleiste mit der Bezeichnung AWS Explorer auf den Bucket, der Ihr Objekt enthält.
4. Klicken Sie mit der rechten Maustaste auf das Objekt, für das Sie ein vorsigniertes Objekt URL generieren lassen möchten, und wählen Sie Vorsigniert erstellen... URL .
5. Geben Sie im Popup-Fenster das Ablaufdatum und die Uhrzeit für Ihr vorsigniertes URL Produkt ein.
6. Der Objektschlüssel sollte auf der Grundlage des ausgewählten Objekts vorab ausgefüllt werden.
7. Geben Sie GETan, dass dieser vorsignierte Wert für das Herunterladen eines Objekts verwendet URL werden soll.

8. Wählen Sie die Schaltfläche Generate (Generieren) aus.
9. Um das in die Zwischenablage URL zu kopieren, wählen Sie Kopieren.
10. Um das generierte, vorsignierte Objekt zu verwendenURL, fügen Sie es URL in einen beliebigen Browser ein.

Verwenden AWS Toolkit for Visual Studio Code

Wenn Sie Visual Studio Code verwenden, können Sie einen Code generieren, der vorsigniert ist, um ein Objekt gemeinsam URL zu nutzen, ohne Code schreiben zu müssen, indem Sie. AWS Toolkit for Visual Studio Code Allgemeine Informationen finden Sie unter AWS Toolkit for Visual Studio Code im AWS Toolkit for Visual Studio Code -Benutzerhandbuch.

Anweisungen zur Installation von finden Sie AWS Toolkit for Visual Studio Code unter Installation von AWS Toolkit for Visual Studio Code im AWS Toolkit for Visual Studio Code Benutzerhandbuch.

1. Connect Sie AWS mithilfe der folgenden Schritte eine Verbindung zu her: Herstellen einer Verbindung zu AWS Toolkit for Visual Studio Code im AWS Toolkit for Visual Studio Code Benutzerhandbuch.
2. Wählen Sie das AWS Logo im linken Bereich in Visual Studio Code aus.
3. Wählen Sie EXPLORERunter S3 aus.
4. Wählen Sie einen Bucket und eine Datei und öffnen Sie das Kontextmenü (rechte Maustaste).
5. Wählen Sie Generate Presigned URL aus und legen Sie dann die Ablaufzeit (in Minuten) fest.
6. Drücken Sie die Eingabetaste, und das Vorsignierte URL wird in Ihre Zwischenablage kopiert.

Objekte mit vorsigniertem Inhalt hochladen URLs

Sie können presigned verwenden, um jemandem URLs zu ermöglichen, ein Objekt in Ihren Amazon S3 S3-Bucket hochzuladen. Wenn Sie ein vorsigniertes Objekt verwenden, können URL Sie es hochladen, ohne dass eine andere Partei über AWS Sicherheitsanmeldedaten oder -berechtigungen verfügen muss. Eine vorsignierte URL Datei ist durch die Berechtigungen des Benutzers begrenzt, der sie erstellt. Das heißt, wenn Sie ein für das Hochladen eines Objekts vorsigniertes URL Dokument erhalten, können Sie ein Objekt nur hochladen, wenn der Ersteller des Objekts URL über die erforderlichen Berechtigungen zum Hochladen dieses Objekts verfügt.

Wenn jemand das verwendet, URL um ein Objekt hochzuladen, erstellt Amazon S3 das Objekt im angegebenen Bucket. Wenn ein Objekt mit demselben Schlüssel, der im vorsignierten Objekt

angegeben ist, URL bereits im Bucket vorhanden ist, ersetzt Amazon S3 das vorhandene Objekt durch das hochgeladene Objekt. Nach dem Upload gehört das Objekt dem Bucket-Besitzer.

Allgemeine Informationen zu Presigned URLs finden Sie unter. Laden Sie Objekte mit vorsignierten Objekten herunter und laden Sie sie hoch URLs

Mithilfe des AWS Explorers URL für Visual Studio können Sie ein vorsigniertes Objekt für das Hochladen eines Objekts erstellen, ohne Code schreiben zu müssen. Sie können ein vorsigniertes Objekt auch URL programmgesteuert generieren, indem Sie den verwenden. AWS SDKs

Verwenden von (Windows AWS Toolkit for Visual Studio)

> **Note**
>
> Derzeit AWS Toolkit for Visual Studio unterstützt der Visual Studio für Mac nicht.

1. Installieren Sie das AWS Toolkit for Visual Studio mithilfe der folgenden Anweisungen unter Installation und Einrichtung des Toolkit for Visual Studio im AWS Toolkit for Visual Studio Benutzerhandbuch.
2. Connect Sie AWS mithilfe der folgenden Schritte eine Verbindung zu her: Herstellen einer Verbindung zu AWS im AWS Toolkit for Visual Studio Benutzerhandbuch.
3. Klicken Sie im linken Bereich mit der Bezeichnung AWS Explorer mit der rechten Maustaste auf den Bucket, in den Sie ein Objekt hochladen möchten.
4. Wählen Sie Vorsigniert URL erstellen... .
5. Geben Sie im Popup-Fenster das Ablaufdatum und die Uhrzeit für Ihr vorsigniertes URL Produkt ein.
6. Geben Sie für Object Key den Namen der hochzuladenden Datei ein. Die Datei, die Sie hochladen, muss genau diesem Namen entsprechen. Wenn ein Objekt mit demselben Objektschlüssel bereits im Bucket vorhanden ist, ersetzt Amazon S3 das vorhandene Objekt durch das neu hochgeladene Objekt.
7. Geben Sie PUTan, dass dieses vorsignierte URL Objekt für das Hochladen eines Objekts verwendet wird.
8. Wählen Sie die Schaltfläche Generate (Generieren) aus.
9. Um das in die Zwischenablage URL zu kopieren, wählen Sie Kopieren.
10. Um dies zu verwendenURL, können Sie eine PUT Anfrage mit dem `curl` Befehl senden. Geben Sie den vollständigen Pfad zu Ihrer Datei und die vorsignierte Datei URL an.

```
curl -X PUT -T "/path/to/file" "presigned URL"
```

Mit dem AWS SDKs

Beispiele für die Verwendung von AWS SDKs zum Generieren eines Vorsignierten URL für das Hochladen eines Objekts finden Sie unter Erstellen eines URL für Amazon S3 vorsignierten mithilfe von. AWS SDK

Wenn Sie das verwenden AWS SDKs, um ein vorsigniertes Objekt zu generierenURL, beträgt die maximale Ablaufzeit 7 Tage ab dem Zeitpunkt der Erstellung.

> **Note**
>
> Für alle, die nach dem 20. März 2019 AWS-Regionen gestartet wurden, müssen Sie das `endpoint-url` und `AWS-Region` zusammen mit der Anfrage angeben. Eine Liste aller Amazon-S3-Regionen und Endpunkte finden Sie unter Regionen und Endpunkte in der Allgemeinen AWS -Referenz.

Transformieren von Objekten mit S3 Object Lambda

Mit S3 Object Lambda können Sie `GET`-, `LIST`- und `HEAD`-Anforderungen von Amazon S3 eigenen Code hinzufügen, um Daten zu ändern und zu verarbeiten, wenn sie an eine Anwendung zurückgegeben werden. Sie können mit benutzerdefiniertem Code die von S3-`GET`-Anforderungen zurückgegebenen Daten ändern, um Zeilen zu filtern, Bilder dynamisch in der Größe zu ändern und mit Wassermarken zu versehen, vertrauliche Daten zu redigieren und mehr. Sie können S3 Object Lambda auch verwenden, um die Ausgabe von S3-`LIST`-Anforderungen zu ändern und eine benutzerdefinierte Ansicht aller Objekte in einem Bucket zu erstellen, und von S3-`HEAD`-Anforderungen zum Ändern von Objektmetadaten wie Objektname und -größe. Sie können S3 Object Lambda als Quelle für Ihre CloudFront Amazon-Distribution verwenden, um Daten auf Endbenutzer zuzuschneiden, z. B. die automatische Größenänderung von Bildern, die Transcodierung älterer Formate (wie von JPEG zu WebP) oder das Entfernen von Metadaten. Weitere Informationen finden Sie im AWS Blogbeitrag Verwenden Sie Amazon S3 Object Lambda mit Amazon CloudFront. Ihr Code wird von AWS Lambda-Funktionen unterstützt und läuft auf einer Infrastruktur, die vollständig von AWS verwaltet wird. Mit S3 Object Lambda wird die Notwendigkeit reduziert, abgeleitete Kopien

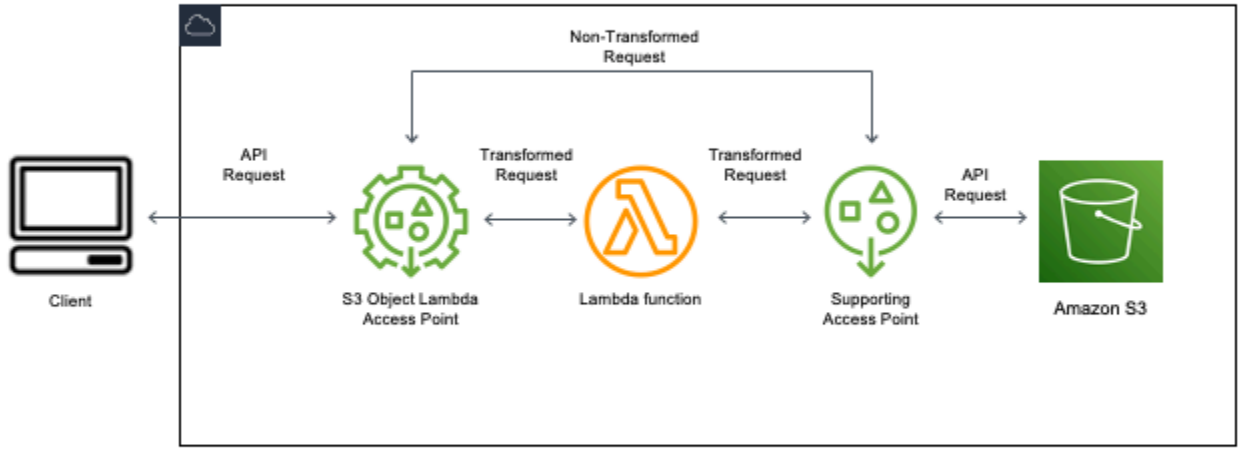
Ihrer Daten zu erstellen und zu speichern oder Proxys auszuführen. Dabei sind keine Änderungen an Ihren Anwendungen erforderlich.

So funktioniert S3 Object Lambda

S3 Object Lambda verwendet AWS Lambda Funktionen, um die Ausgabe von Standard-S3-`GET`, `LIST`, oder `HEAD` -Anfragen automatisch zu verarbeiten. AWS Lambda ist ein serverloser Rechendienst, der kundendefinierten Code ausführt, ohne dass die Verwaltung der zugrunde liegenden Rechenressourcen erforderlich ist. Sie können Ihre eigenen benutzerdefinierten Lambda-Funktionen erstellen und ausführen und die Datentransformation an Ihren spezifischen Anwendungsfall anpassen.

Nachdem Sie eine Lambda-Funktion konfiguriert haben, fügen Sie sie an einen Serviceendpunkt von S3 Object Lambda-an, der als Object Lambda Access Point bezeichnet wird. Der Object Lambda Access Point verwendet für den Zugriff auf Amazon S3 einen standardmäßigen S3-Zugriffspunkt, der als unterstützender Zugriffspunkt bezeichnet wird.

Wenn Sie eine Anforderung an Ihren Object Lambda Access Point senden, ruft Amazon S3 Ihre Lambda-Funktion automatisch auf. Dann geben alle Daten, die mit der S3-Anforderung `GET`, `LIST` oder `HEAD` über den Object Lambda Access Point abgerufen werden, ein transformiertes Ergebnis an die Anwendung zurück. Alle anderen Anforderungen werden wie gewohnt verarbeitet, wie im folgenden Diagramm dargestellt.



Die Themen in diesem Abschnitt beschreiben, wie Sie mit S3 Object Lambda arbeiten.

Themen

- Erstellen von Objekt-Lambda-Zugriffspunkten
- Verwenden von Amazon S3 Objekt-Lambda-Zugriffspunkten
- Sicherheitsüberlegungen für S3 Object Lambda-Zugriffspunkte
- Schreiben von Lambda-Funktionen für S3 Object Lambda Access Points
- Verwenden von AWS integrierten Lambda-Funktionen
- Bewährte Methoden und Richtlinien für S3 Object Lambda
- Tutorials zu S3 Object Lambda
- Debuggen und Problembehandlung bei S3 Object Lambda

Erstellen von Objekt-Lambda-Zugriffspunkten

Ein Object Lambda Access Point ist mit genau einem Standard-Zugriffspunkt und folglich mit einem Amazon-S3-Bucket verknüpft. Zum Erstellen eines Object Lambda Access Point benötigen Sie folgende Ressourcen:

- Ein Amazon-S3-Bucket Informationen zum Erstellen von Buckets finden Sie unter the section called “Erstellen eines Buckets”.
- Ein standardmäßiger S3-Zugriffspunkt Wenn Sie mit Object Lambda Access Point arbeiten, wird dieser Standardzugriffspunkt als unterstützender Zugriffspunkt bezeichnet. Informationen zum Erstellen von Standardzugriffspunkten finden Sie unter the section called “Erstellen von Zugriffspunkten”.
- Eine AWS Lambda Funktion. Sie können entweder Ihre eigene Lambda-Funktion erstellen oder eine vordefinierte Funktion verwenden. Weitere Informationen zum Erstellen von Lambda-Funktionen finden Sie unter the section called “Schreiben von Lambda-Funktionen”. Weitere Hinweise zu vorgefertigten Funktionen finden Sie unter Verwenden von AWS integrierten Lambda-Funktionen.
- (Optional) Eine AWS Identity and Access Management (IAM) Richtlinie. Amazon S3 Access Points unterstützen IAM Ressourcenrichtlinien, mit denen Sie die Nutzung des Access Points nach Ressourcen, Benutzern oder anderen Bedingungen steuern können. Weitere Informationen zum Erstellen dieser Richtlinien finden Sie unter the section called “Richtlinien konfigurieren IAM”.

In den folgenden Abschnitten wird beschrieben, wie Sie einen Object Lambda Access Point mit Folgendem erstellen:

- Die AWS Management Console
- Das AWS Command Line Interface (AWS CLI)
- Eine AWS CloudFormation Vorlage
- Die AWS Cloud Development Kit (AWS CDK)

Informationen zum Erstellen eines Object Lambda Access Points mithilfe von finden Sie `CreateAccessPointForObjectLambda`in der REST API Amazon Simple Storage Service API Reference.

Erstellen eines Object Lambda Access Point

Erstellen Sie Ihren Object Lambda Access Point mit einem der folgenden Verfahren.

Verwenden der S3-Konsole

So erstellen Sie einen Object Lambda Access Point mit der Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Navigationsleiste den Namen der aktuell angezeigten Datei aus AWS-Region. Wählen Sie als Nächstes die Region aus, zu der Sie wechseln möchten.
3. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
4. Wählen Sie auf der Seite Object Lambda access points (Objekt-Lambda-Zugriffspunkte) die Option Create Object Lambda access point (Objekt-Lambda-Zugriffspunkt erstellen).
5. Geben Sie unter Object Lambda access point name (Name des Objekt-Lambda-Zugriffspunkts) den Namen ein, den Sie für den Zugriffspunkt verwenden möchten.

   Wie bei Standard-Zugriffspunkten gibt es Regeln für die Benennung von Object Lambda Access Points. Weitere Informationen finden Sie unter Regeln zur Benennung von Amazon S3-Zugriffspunkten.
6. Geben Sie für Supporting Access Point (Unterstützender Access Point) den Standard-Zugriffspunkt ein, den Sie verwenden möchten, oder navigieren Sie zu diesem. Der Access Point muss sich in derselben Umgebung befinden AWS-Region wie die Objekte, die Sie transformieren möchten. Informationen zum Erstellen von Standardzugriffspunkten finden Sie unter the section called “Erstellen von Zugriffspunkten”.
7. Unter Transformationskonfiguration können Sie eine Funktion hinzufügen, die Ihre Daten für den Object Lambda Access Point transformiert. Führen Sie eine der folgenden Aktionen aus:

   - Wenn Sie bereits eine AWS Lambda Funktion in Ihrem Konto haben, können Sie diese unter Lambda-Funktion aufrufen auswählen. Hier können Sie den Amazon-Ressourcennamen (ARN) einer Lambda-Funktion in Ihre eingeben AWS-Konto oder eine Lambda-Funktion aus dem Drop-down-Menü auswählen.
   - Wenn Sie eine AWS eingebaute Funktion verwenden möchten, wählen Sie den Funktionsnamen unter AWS Built-Funktion und wählen Sie Lambda-Funktion erstellen aus. Dadurch gelangen Sie zur Lambda-Konsole, wo Sie eine integrierte Funktion in Ihrem AWS-Konto implementieren können. Weitere Hinweise zu integrierten Funktionen finden Sie unter Verwenden von AWS integrierten Lambda-Funktionen.

Wählen Sie unter S3 eine oder mehrere API Operationen ausAPIs, die aufgerufen werden sollen. Für jede API Auswahl müssen Sie eine Lambda-Funktion angeben, die aufgerufen werden soll.

8. (Optional) Fügen Sie unter Payload JSON Text hinzu, den Sie Ihrer Lambda-Funktion als Eingabe zur Verfügung stellen möchten. Sie können Nutzlasten mit verschiedenen Parametern für verschiedene Object Lambda Access Points konfigurieren, die dieselbe Lambda-Funktion aufrufen, wodurch die Flexibilität Ihrer Lambda-Funktion erhöht wird.

   > **Important**
   >
   > Wenn Sie Object Lambda Access Points verwenden, stellen Sie sicher, dass die Nutzlast keine vertraulichen Informationen enthält.

9. (Optional) Für Range and part number (Bereichs- und Teilenummer) müssen Sie diese Option aktivieren, wenn Sie `GET`- und `HEAD`-Anforderungen mit Bereichs- und Teilenummern-Headern bearbeiten möchten. Durch Aktivieren dieser Option wird bestätigt, dass Ihre Lambda-Funktion diese Anforderungen erkennen und bearbeiten kann. Weitere Informationen zu Bereichsüberschriften und Teilenummern finden Sie unter Arbeitet mit Range and partNumber Header.
10. (Optional) Wählen Sie für Anforderungsmetriken die Option Aktivieren oder Deaktivieren aus, um Ihrem Object Lambda Access Point Amazon-S3-Überwachung hinzuzufügen. Anforderungsmetriken werden zum CloudWatch Standardtarif von Amazon abgerechnet.
11. (Optional) Legen Sie unter Object Lambda Access Point policy (Objekt-Lambda-Access-Point-Richtlinie) eine Ressourcenrichtlinie fest. Ressourcenrichtlinien erteilen Berechtigungen für den angegebenen Object Lambda Access Point und können die Verwendung des Zugriffspunkts nach Ressource, Benutzer oder anderen Bedingungen steuern. Weitere Hinweise zu Ressourcenrichtlinien für Objekt-Lambda-Zugriffspunkten finden Sie unter Konfiguration von IAM Richtlinien für Object Lambda Access Points.
12. Wählen Sie unter Block Public Access settings for this Object Lambda Access Point (Einstellungen für die Blockierung des öffentlichen Zugriffs für diesen Objekt-Lambda-Zugriffspunkt) die Einstellungen für die Blockierung des öffentlichen Zugriffs aus, die Sie anwenden möchten. Alle Einstellungen für die Blockierung des öffentlichen Zugriffs sind standardmäßig für neue Objekt-Lambda-Zugriffspunkte aktiviert. Wir empfehlen Ihnen, die Standardeinstellungen aktiviert zu lassen. Amazon S3 unterstützt derzeit nicht das Ändern der Einstellungen für die Blockierung des öffentlichen Zugriffs eines Objekt-Lambda-Zugriffspunkts, nachdem der Objekt-Lambda-Zugriffspunkt erstellt wurde.

Weitere Informationen über die Blockierung des öffentlichen Zugriffs in Amazon S3 finden Sie unter Verwalten des öffentlichen Zugriffs auf Zugriffspunkte.

13. Wählen Sie Create Object Lambda Access Point (Objekt-Lambda-Access-Point erstellen) aus.

Mit dem AWS CLI

So erstellen Sie einen Object Lambda Access Point mithilfe einer Vorlage AWS CloudFormation

> **Note**
>
> Wenn Sie die folgenden Befehle verwenden, ersetzen Sie *user input placeholders* durch eigene Informationen.

1. Laden Sie das AWS Lambda Funktionsbereitstellungspaket `s3objectlambda_deployment_package.zip` in der Standardkonfiguration von S3 Object Lambda herunter.
2. Führen Sie den folgenden `put-object`-Befehl aus, um das Paket in einen Amazon-S3-Bucket hochzuladen.

```
aws s3api put-object --bucket Amazon S3 bucket name --key
 s3objectlambda_deployment_package.zip --body release/
s3objectlambda_deployment_package.zip
```

3. Laden Sie die AWS CloudFormation Vorlage `s3objectlambda_defaultconfig.yaml` in der Standardkonfiguration von S3 Object Lambda herunter.
4. Führen Sie den folgenden `deploy`-Befehl aus, um die Vorlage in Ihrem AWS-Konto bereitzustellen.

```
aws cloudformation deploy --template-file s3objectlambda_defaultconfig.yaml \
 --stack-name AWS CloudFormation stack name \
 --parameter-overrides ObjectLambdaAccessPointName=Object Lambda Access Point name
 \
  SupportingAccessPointName=Amazon S3 access point S3BucketName=Amazon S3 bucket \
  LambdaFunctionS3BucketName=Amazon S3 bucket containing your Lambda package \
  LambdaFunctionS3Key=Lambda object key LambdaFunctionS3ObjectVersion=Lambda object
 version \
  LambdaFunctionRuntime=Lambda function runtime --capabilities capability_IAM
```

Sie können diese AWS CloudFormation Vorlage so konfigurieren, dass Lambda für `GETHEAD`, und `LIST` API -Operationen aufgerufen wird. Weitere Informationen zum Ändern der Standardkonfiguration der Vorlage finden Sie unter the section called “Automatisieren Sie die Einrichtung von S3 Object Lambda mit AWS CloudFormation”.

So erstellen Sie einen Object Lambda Access Point mit dem AWS CLI

> **Note**
>
> Wenn Sie die folgenden Befehle verwenden, ersetzen Sie *`user input placeholders`* durch eigene Informationen.

Im folgenden Beispiel wird ein Object Lambda Access Point mit dem Namen erstellt *`my-object-lambda-ap`* für den Bucket *`amzn-s3-demo-bucket1`* auf dem Konto *`111122223333`*. In diesem Beispiel wird davon ausgegangen, dass ein Standardzugriffspunkt mit dem Namen *`Beispiel-AP`* wurde bereits erstellt. Informationen zum Erstellen eines Standardzugriffspunkts finden Sie unter the section called “Erstellen von Zugriffspunkten”.

In diesem Beispiel wird die AWS vorgefertigte Funktion `decompress` verwendet. Weitere Hinweise zu vorgefertigten Funktionen finden Sie unter the section called “Verwenden von AWS eingebauten Funktionen”.

1. Erstellen Sie einen Bucket. In diesem Beispiel verwenden wir *`amzn-s3-demo-bucket1`*. Hinweise zum Erstellen von Buckets finden Sie unterthe section called “Erstellen eines Buckets”.
2. Erstellen Sie einen Standardzugriffspunkt und hängen Sie ihn an Ihren Bucket an. In diesem Beispiel verwenden wir *`Beispiel-AP`*. Informationen zum Erstellen von Standardzugriffspunkten finden Sie unterthe section called “Erstellen von Zugriffspunkten”.
3. Führen Sie eine der folgenden Aktionen aus:

   - Erstellen Sie in Ihrem Konto eine Lambda-Funktion, mit der Sie Ihr Amazon-S3-Objekt transformieren möchten. Weitere Informationen zum Erstellen von Lambda-Funktionen finden Sie unter the section called “Schreiben von Lambda-Funktionen”. Informationen zur Verwendung Ihrer benutzerdefinierten Funktion mit dem AWS CLI finden Sie unter Verwenden von Lambda mit dem AWS CLI im AWS Lambda Entwicklerhandbuch.
   - Verwenden Sie eine AWS vorgefertigte Lambda-Funktion. Weitere Hinweise zu vorgefertigten Funktionen finden Sie unter Verwenden von AWS integrierten Lambda-Funktionen.

4. Erstellen Sie eine JSON Konfigurationsdatei mit dem Namen. `my-olap-configuration.json` Geben Sie in dieser Konfiguration den unterstützenden Access Point und den Amazon-Ressourcennamen (ARN) für die Lambda-Funktion an, die Sie in den vorherigen Schritten erstellt haben, oder ARN für die vordefinierte Funktion, die Sie verwenden.

   Example

   ```
   {
       "SupportingAccessPoint" : "arn:aws:s3:us-
   east-1:111122223333:accesspoint/example-ap",
       "TransformationConfigurations": [{
           "Actions" : ["GetObject", "HeadObject", "ListObjects", "ListObjectsV2"],
           "ContentTransformation" : {
               "AwsLambda": {
                   "FunctionPayload" : "{\"compressionType\":\"gzip\"}",
                   "FunctionArn" : "arn:aws:lambda:us-east-1:111122223333:function/
   compress"
               }
           }
       }]
   }
   ```

5. Führen Sie den Befehl `create-access-point-for-object-lambda` aus, um den Object Lambda Access Point zu erstellen.

   ```
   aws s3control create-access-point-for-object-lambda --account-id 111122223333 --
   name my-object-lambda-ap --configuration file://my-olap-configuration.json
   ```

6. (Optional) Erstellen Sie eine JSON Richtliniendatei mit dem Namen. `my-olap-policy.json`

   Durch Hinzufügen einer Ressourcenrichtlinie für Objekt-Lambda-Zugriffspunkte lässt sich die Verwendung des Zugriffspunkts nach Ressource, Benutzer oder anderen Bedingungen steuern. Diese Ressourcenrichtlinie gewährt die `GetObject` Berechtigung für das Konto *444455556666* zum angegebenen Object Lambda Access Point.

   Example

   ```
   {
       "Version": "2008-10-17",
       "Statement": [
   ```

```
        {
            "Sid": "Grant account 444455556666 GetObject access",
            "Effect": "Allow",
            "Action": "s3-object-lambda:GetObject",
            "Principal": {
                "AWS": "arn:aws:iam::444455556666:root"
            },
            "Resource": "your-object-lambda-access-point-arn"
        }
    ]
}
```

7. (Optional) Führen Sie den Befehl `put-access-point-policy-for-object-lambda` aus, um Ihre Ressourcenrichtlinie festzulegen.

```
aws s3control put-access-point-policy-for-object-lambda --account-id 111122223333
 --name my-object-lambda-ap --policy file://my-olap-policy.json
```

8. (Optional) Geben Sie eine Nutzlast an.

   Eine Nutzlast ist optional JSON und kann als Eingabe für Ihre AWS Lambda Funktion bereitgestellt werden. Sie können Nutzlasten mit verschiedenen Parametern für verschiedene Object Lambda Access Points konfigurieren, die dieselbe Lambda-Funktion aufrufen, wodurch die Flexibilität Ihrer Lambda-Funktion erhöht wird.

   Die folgende Konfiguration des Object Lambda Access Point zeigt eine Nutzlast mit zwei Parametern.

```
{
 "SupportingAccessPoint": "AccessPointArn",
 "CloudWatchMetricsEnabled": false,
 "TransformationConfigurations": [{
  "Actions": ["GetObject", "HeadObject", "ListObjects", "ListObjectsV2"],
  "ContentTransformation": {
   "AwsLambda": {
    "FunctionArn": "FunctionArn",
    "FunctionPayload": "{\"res-x\": \"100\",\"res-y\": \"100\"}"
   }
  }
 }]
}
```

Die folgende Konfiguration des Object Lambda Access Point zeigt eine Nutzlast mit einem Parameter und den aktivierten Einstellungen `GetObject-Range`, `GetObject-PartNumber`, `HeadObject-Range` und `HeadObject-PartNumber`.

```
{
    "SupportingAccessPoint":"AccessPointArn",
    "CloudWatchMetricsEnabled": false,
    "AllowedFeatures": ["GetObject-Range", "GetObject-PartNumber", "HeadObject-
Range", "HeadObject-PartNumber"],
    "TransformationConfigurations": [{
        "Action": ["GetObject", "HeadObject", "ListObjects", "ListObjectsV2"],
        "ContentTransformation": {
            "AwsLambda": {
                "FunctionArn":"FunctionArn",
                "FunctionPayload": "{\"compression-amount\": \"5\"}"
            }
        }
    }]
}
```

> **Important**
>
> Wenn Sie Object Lambda Access Points verwenden, stellen Sie sicher, dass die Nutzlast keine vertraulichen Informationen enthält.

Verwenden der AWS CloudFormation Konsole und der Vorlage

Sie können einen Object Lambda Access Point mithilfe der von Amazon S3 bereitgestellten Standardkonfiguration erstellen. Sie können eine AWS CloudFormation Vorlage und den Quellcode einer Lambda-Funktion aus dem GitHub Repository herunterladen und diese Ressourcen bereitstellen, um einen funktionierenden Object Lambda Access Point einzurichten.

Informationen zum Ändern der Standardkonfiguration der AWS CloudFormation Vorlage finden Sie unter. the section called “Automatisieren Sie die Einrichtung von S3 Object Lambda mit AWS CloudFormation”

Informationen zur Konfiguration von Object Lambda Access Points AWS CloudFormation ohne Vorlage finden Sie `AWS::S3ObjectLambda::AccessPoint`im AWS CloudFormation Benutzerhandbuch.

So laden Sie das Bereitstellungspaket für die Lambda-Funktion hoch

1. Laden Sie das AWS Lambda Funktionsbereitstellungspaket `s3objectlambda_deployment_package.zip` in der Standardkonfiguration von S3 Object Lambda herunter.
2. Hochladen des Pakets in einen Amazon S3 Bucket

So erstellen Sie einen Object Lambda Access Point mithilfe der Konsole AWS CloudFormation

1. Laden Sie die AWS CloudFormation Vorlage `s3objectlambda_defaultconfig.yaml` in der Standardkonfiguration von S3 Object Lambda herunter.
2. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS CloudFormation Konsole unter https://console.aws.amazon.com/cloudformation.
3. Führen Sie eine der folgenden Aktionen aus:
   - Wenn Sie es noch nie AWS CloudFormation zuvor verwendet haben, wählen Sie auf der AWS CloudFormation Startseite Stack erstellen aus.
   - Wenn Sie es schon AWS CloudFormation einmal verwendet haben, wählen Sie im linken Navigationsbereich Stacks aus. Wählen Sie Create stack (Stack erstellen) und dann With new resources (standard) (Mit neuen Ressourcen (Standard)) aus.
4. Wählen Sie für Voraussetzung - Vorlage vorbereiten die Option Vorlage ist bereit aus.
5. Wählen Sie unter Specify template (Vorlage angeben) die Option Upload a template file (Vorlagendatei hochladen) aus und laden Sie `s3objectlambda_defaultconfig.yaml` hoch.
6. Wählen Sie Weiter.
7. Geben Sie auf der Seite Stackdetails angeben einen Namen für den Stack ein.
8. Geben Sie im Abschnitt Parameters (Parameter) die folgenden Parameter an, die in der Stack-Vorlage definiert sind:
   a. Führen Sie für CreateNewSupportingAccessPointeinen der folgenden Schritte aus:
      - Wenn Sie bereits einen unterstützenden Zugriffspunkt für den S3-Bucket haben, in den Sie die Vorlage hochgeladen haben, wählen Sie false (falsch) aus.
      - Wenn Sie einen neuen Zugriffspunkt für diesen Bucket erstellen möchten, wählen Sie true (wahr) aus.
   b. Wählen Sie für EnableCloudWatchMonitoringwahr oder falsch, je nachdem, ob Sie CloudWatch Amazon-Anforderungsmetriken und Alarme aktivieren möchten.

c. (Optional) Fügen Sie JSON Text hinzu LambdaFunctionPayload, den Sie Ihrer Lambda-Funktion als Eingabe zur Verfügung stellen möchten. Sie können Nutzlasten mit verschiedenen Parametern für verschiedene Object Lambda Access Points konfigurieren, die dieselbe Lambda-Funktion aufrufen, wodurch die Flexibilität Ihrer Lambda-Funktion erhöht wird.

> **Important**
>
> Wenn Sie Object Lambda Access Points verwenden, stellen Sie sicher, dass die Nutzlast keine vertraulichen Informationen enthält.

d. Geben Sie für LambdaFunctionRuntimeIhre bevorzugte Laufzeit für die Lambda-Funktion ein. Die verfügbaren Auswahlmöglichkeiten sind `nodejs14.x`, `python3.9`, `java11`.

e. Geben Sie für LambdaFunctionS3 BucketName den Amazon S3 S3-Bucket-Namen ein, in den Sie das Bereitstellungspaket hochgeladen haben.

f. Geben Sie für LambdaFunctionS3Key den Amazon S3 S3-Objektschlüssel ein, in den Sie das Bereitstellungspaket hochgeladen haben.

g. Geben Sie für LambdaFunctionS3 die Amazon S3 S3-Objektversion einObjectVersion, in die Sie das Bereitstellungspaket hochgeladen haben.

h. Geben Sie für ObjectLambdaAccessPointNameeinen Namen für Ihren Object Lambda Access Point ein.

i. Geben Sie für S3 BucketName den Amazon S3 S3-Bucket-Namen ein, der Ihrem Object Lambda Access Point zugeordnet wird.

j. Geben Sie für SupportingAccessPointNameden Namen Ihres unterstützenden Access Points ein.

> **Note**
>
> Dies ist ein Zugriffspunkt, der mit dem Amazon-S3-Bucket verknüpft ist, den Sie im vorherigen Schritt ausgewählt haben. Wenn Ihrem Amazon S3 S3-Bucket keine Access Points zugeordnet sind, können Sie die Vorlage so konfigurieren, dass sie einen für Sie erstellt, indem Sie true for wählen CreateNewSupportingAccessPoint.

9. Wählen Sie Weiter.
10. Wählen Sie auf der Seite Configure stack options (Stack-Optionen konfigurieren) Next (Weiter) aus.

Weitere Informationen zu den optionalen Einstellungen auf dieser Seite finden Sie unter Einstellen von AWS CloudFormation -Stack-Optionen im AWS CloudFormation - Benutzerhandbuch.

11. Wählen Sie auf der Seite Prüfen Stack erstellen aus.

Mit dem AWS Cloud Development Kit (AWS CDK)

Weitere Informationen zur Konfiguration von Object Lambda Access Points mithilfe von finden Sie unter `AWS::S3ObjectLambdaConstruct Library` in der AWS Cloud Development Kit (AWS CDK) APIReferenz. AWS CDK

Automatisieren Sie die Einrichtung von S3 Object Lambda mit einer Vorlage CloudFormation

Sie können eine AWS CloudFormation Vorlage verwenden, um schnell einen Amazon S3 Object Lambda Access Point zu erstellen. Die CloudFormation Vorlage erstellt automatisch relevante Ressourcen, konfiguriert AWS Identity and Access Management (IAM) -Rollen und richtet eine AWS Lambda Funktion ein, die Anfragen automatisch über den Object Lambda Access Point verarbeitet. Mit der CloudFormation Vorlage können Sie bewährte Verfahren implementieren, Ihre Sicherheitslage verbessern und Fehler reduzieren, die durch manuelle Prozesse verursacht werden.

Dieses GitHub Repository enthält die CloudFormation Vorlage und den Quellcode der Lambda-Funktion. Anweisungen zur Verwendung der Vorlage finden Sie unter the section called “Erstellen von Objekt-Lambda-Zugriffspunkten”.

Die in der Vorlage bereitgestellte Lambda-Funktion führt keine Transformation aus. Stattdessen gibt sie Ihre Objekte im Ist-Zustand aus Ihrem S3-Bucket zurück. Sie können die Funktion klonen und Ihren eigenen Transformationscode hinzufügen, um Daten zu ändern und zu verarbeiten, wenn sie an eine Anwendung zurückgegeben werden. Weitere Informationen zum Anpassen Ihrer Funktion finden Sie unter the section called “Ändern der Lambda-Funktion” und the section called “Schreiben von Lambda-Funktionen”.

Ändern der Vorlage

Erstellen eines neuen unterstützenden Zugriffspunkts

S3 Object Lambda verwendet zwei Zugriffspunkte, einen Object Lambda Access Point und einen standardmäßigen S3-Zugriffspunkt, der als unterstützender Zugriffspunkt bezeichnet wird. Wenn Sie eine Anforderung an einen Object Lambda Access Point ausführen, ruft S3 entweder Lambda

in Ihrem Namen auf oder delegiert die Anforderung an den unterstützenden Zugriffspunkt, abhängig von der Konfiguration von S3 Object Lambda. Sie können einen neuen unterstützenden Zugriffspunkt erstellen, indem Sie den folgenden Parameter als Teil des `aws cloudformation deploy`-Befehls, der beim Bereitstellen der Vorlage erstellt werden soll, übergeben.

```
CreateNewSupportingAccessPoint=true
```

Konfigurieren einer Nutzlast-Funktion

Sie können eine Nutzlast konfigurieren, um der Lambda-Funktion zusätzliche Daten bereitzustellen, indem Sie beim Bereitstellen der Vorlage den folgenden Parameter als Teil des `aws cloudformation deploy`-Befehls übergeben.

```
LambdaFunctionPayload="format=json"
```

CloudWatch Amazon-Überwachung aktivieren

Sie können die CloudWatch Überwachung aktivieren, indem Sie bei der Bereitstellung der Vorlage den folgenden Parameter als Teil des `aws cloudformation deploy` Befehls übergeben.

```
EnableCloudWatchMonitoring=true
```

Dieser Parameter aktiviert die Anforderungsmetriken Ihres Object Lambda Access Point für Amazon S3 und erstellt zwei CloudWatch Alarme zur Überwachung von clientseitigen und serverseitigen Fehlern.

> **Note**
>
> Für die CloudWatch Nutzung durch Amazon fallen zusätzliche Kosten an. Weitere Informationen zu Amazon-S3-Anforderungsmetriken finden Sie unter Überwachen und Protokollieren von Zugriffspunkten.
> Details zu den Preisen finden Sie unter CloudWatch -Preise.

Konfigurieren von Provisioned Concurrency

Zur Reduzierung der Latenz können Sie bereitgestellte Parallelität für die Lambda-Funktion konfigurieren, die den Object Lambda Access Point unterstützt, indem Sie die Vorlage so bearbeiten, dass sie die folgenden Zeilen unter `Resources` einschließt.

```
LambdaFunctionVersion:
      Type: AWS::Lambda::Version
      Properties:
        FunctionName: !Ref LambdaFunction
        ProvisionedConcurrencyConfig:
            ProvisionedConcurrentExecutions: Integer
```

> **Note**
>
> Für die Bereitstellung von Nebenläufigkeit fallen zusätzliche Gebühren an. Weitere Informationen zur bereitgestellten Parallelität finden Sie unter Verwalten der von Lambda bereitgestellten Parallelität im AWS Lambda -Entwicklerhandbuch.
> Details zu den Preisen finden Sie unter AWS Lambda -Preise.

Ändern der Lambda-Funktion

Ändern der Header-Werte für eine **`GetObject`**-Anforderung

Standardmäßig leitet die Lambda-Funktion alle Header außer `Content-Length` und `ETag` von der vorsignierten URL Anfrage an den Client weiter. `GetObject` Basierend auf Ihrem Transformationscode in der Lambda-Funktion können Sie wählen, ob Sie neue Header-Werte an den `GetObject`-Client senden.

Sie können Ihre Lambda-Funktion aktualisieren, um neue Header-Werte zu senden, indem Sie sie in der `WriteGetObjectResponse` API Operation übergeben.

Wenn Ihre Lambda-Funktion beispielsweise den Text in Amazon-S3-Objekten in eine andere Sprache übersetzt, können Sie einen neuen Wert im `Content-Language`-Header bestätigen. Ändern Sie die `writeResponse`-Funktion dazu wie folgt:

```
async function writeResponse (s3Client: S3, requestContext: GetObjectContext,
 transformedObject: Buffer,
  headers: Headers): Promise<PromiseResult<{}, AWSError>> {
  const { algorithm, digest } = getChecksum(transformedObject);

  return s3Client.writeGetObjectResponse({
    RequestRoute: requestContext.outputRoute,
    RequestToken: requestContext.outputToken,
    Body: transformedObject,
```

```
    Metadata: {
      'body-checksum-algorithm': algorithm,
      'body-checksum-digest': digest
    },
    ...headers,
    ContentLanguage: 'my-new-language'
  }).promise();
}
```

Eine vollständige Liste der unterstützten Header finden Sie WriteGetObjectResponsein der Amazon Simple Storage Service API Reference.

Zurückgeben von Metadaten Headern

Sie können Ihre Lambda-Funktion aktualisieren, um neue Header-Werte zu senden, indem Sie sie in der WriteGetObjectResponseAPIOperationsanforderung übergeben.

```
async function writeResponse (s3Client: S3, requestContext: GetObjectContext,
 transformedObject: Buffer,
  headers: Headers): Promise<PromiseResult<{}, AWSError>> {
  const { algorithm, digest } = getChecksum(transformedObject);

  return s3Client.writeGetObjectResponse({
    RequestRoute: requestContext.outputRoute,
    RequestToken: requestContext.outputToken,
    Body: transformedObject,
    Metadata: {
      'body-checksum-algorithm': algorithm,
      'body-checksum-digest': digest,
      'my-new-header': 'my-new-value'
    },
    ...headers
  }).promise();
}
```

Einen neuen Statuscode zurückgeben

Sie können einen benutzerdefinierten Statuscode an den `GetObject` Client zurückgeben, indem Sie ihn in der WriteGetObjectResponseAPIBetriebsanforderung übergeben.

```
async function writeResponse (s3Client: S3, requestContext: GetObjectContext,
 transformedObject: Buffer,
```

```
  headers: Headers): Promise<PromiseResult<{}, AWSError>> {
  const { algorithm, digest } = getChecksum(transformedObject);

  return s3Client.writeGetObjectResponse({
    RequestRoute: requestContext.outputRoute,
    RequestToken: requestContext.outputToken,
    Body: transformedObject,
    Metadata: {
      'body-checksum-algorithm': algorithm,
      'body-checksum-digest': digest
    },
    ...headers,
    StatusCode: Integer
  }).promise();
}
```

Eine vollständige Liste der unterstützten Statuscodes finden Sie `WriteGetObjectResponse`in der Amazon Simple Storage Service API Reference.

Anwenden der Parameter **`Range`** und **`partNumber`** auf das Quellobjekt

Standardmäßig kann der von der CloudFormation Vorlage erstellte Object Lambda Access Point die `partNumber` Parameter `Range` und verarbeiten. Die Lambda-Funktion wendet den Bereich oder die Teilenummer an, die auf das transformierte Objekt angefordert wird. Dazu muss die Funktion das gesamte Objekt herunterladen und die Transformation ausführen. In einigen Fällen werden Ihre transformierten Objektbereiche möglicherweise genau Ihren Quellobjektbereichen zugeordnet. Dies bedeutet, dass das Anfordern des Bytebereichs A-B für Ihr Quellobjekt und das Ausführen der Transformation zum gleichen Ergebnis führen kann, wie das Anfordern des gesamten Objekts, das Ausführen der Transformation und das Zurückgeben des Bytebereichs A-B für das transformierte Objekt.

In solchen Fällen können Sie die Implementierung der Lambda-Funktion ändern, um den Bereich oder die Teilenummer direkt auf das Quellobjekt anzuwenden. Dieser Ansatz verbessert die allgemeine Funktionslatenz und den erforderlichen Speicher. Weitere Informationen finden Sie unter the section called “Arbeitet mit Range and partNumber Header”.

Deaktivieren der Verarbeitung von **`Range`** und **`partNumber`**

Standardmäßig kann der von der CloudFormation Vorlage erstellte Object Lambda Access Point die `partNumber` Parameter `Range` und verarbeiten. Wenn Sie dieses Verhalten nicht benötigen, können Sie es deaktivieren, indem Sie die folgenden Zeilen aus der Vorlage entfernen.

```
AllowedFeatures:
  - GetObject-Range
  - GetObject-PartNumber
  - HeadObject-Range
  - HeadObject-PartNumber
```

Transformieren von großen Objekten

Standardmäßig verarbeitet die Lambda-Funktion das gesamte Objekt im Speicher, bevor es mit dem Streamen der Antwort auf S3 Object Lambda beginnen kann. Sie können die Funktion ändern, um die Antwort zu streamen, während sie die Transformation durchführt. Dies hilft, die Transformationslatenz und die Speichergröße der Lambda-Funktion zu reduzieren. Eine Beispielimplementierung finden Sie unter Beispiel für komprimierte Inhalte streamen.

Verwenden von Amazon S3 Objekt-Lambda-Zugriffspunkten

Sie können Anforderungen über Amazon S3 Object Lambda Access Points ebenso vornehmen wie über andere Zugriffspunkte. Weitere Informationen darüber, wie Sie Anforderungen über einen Zugriffspunkt vornehmen können, finden Sie unter Verwenden von Zugriffspunkten. Sie können Anfragen über Object Lambda Access Points stellen, indem Sie die Amazon S3-Konsole, AWS Command Line Interface (AWS CLI) AWS SDKs, oder Amazon S3 REST API verwenden.

> **Important**
>
> Die Amazon-Ressourcennamen (ARNs) für Object Lambda Access Points verwenden den Dienstnamen. `s3-object-lambda` Somit ARNs beginnt Object Lambda Access Point mit dem`arn:aws::s3-object-lambda`, was mit anderen Access Points verwendet wird`arn:aws::s3`, und nicht mit dem, was mit anderen Access Points verwendet wird.

So finden Sie den Lambda Access Point ARN für Ihr Objekt

Um einen Object Lambda Access Point mit dem AWS CLI oder zu verwenden AWS SDKs, müssen Sie den Amazon-Ressourcennamen (ARN) des Object Lambda Access Points kennen. Die folgenden Beispiele zeigen, wie Sie den ARN für einen Object Lambda Access Point mithilfe der Amazon S3 S3-Konsole oder AWS CLI finden.

Verwenden der S3-Konsole

So finden Sie den Lambda Access Point ARN für Ihren Object mithilfe der Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
3. Wählen Sie das Optionsfeld neben dem Object Lambda Access Point, dessen ARN Sie kopieren möchten.
4. Wählen Sie die Option CopyARN Kopieren aus.

Verwenden Sie den AWS CLI

Um den Lambda Access Point ARN für Ihren Object zu finden, verwenden Sie den AWS CLI

1. Führen Sie den folgenden Befehl aus, um eine Liste der Object Lambda Access Points abzurufen, die mit Ihrem AWS-Konto verbunden sind. Bevor Sie den Befehl ausführen, ersetzen Sie die Konto-ID *111122223333* mit Ihrer AWS-Konto ID.

```
aws s3control list-access-points-for-object-lambda --account-id 111122223333
```

2. Überprüfen Sie die Befehlsausgabe, um den Object Lambda Access Point zu findenARN, den Sie verwenden möchten. Die Ausgabe des vorherigen Befehls sollte dem folgenden Beispiel gleichen.

```
{
    "ObjectLambdaAccessPointList": [
        {
            "Name": "my-object-lambda-ap",
            "ObjectLambdaAccessPointArn": "arn:aws:s3-object-lambda:us-
east-1:111122223333:accesspoint/my-object-lambda-ap"
        },
        ...
    ]
}
```

So verwenden Sie einen Alias im Bucket-Stil für den Object Lambda Access Point Ihres S3-Buckets

Wenn Sie einen Object Lambda Access Point erstellen, generiert Amazon S3 automatisch einen eindeutigen Alias für den Object Lambda Access Point. Sie können diesen Alias anstelle eines Amazon S3 S3-Bucket-Namens oder des Object Lambda Access Point Amazon Resource Name (ARN) in einer Anfrage für Access Point-Datenebenenoperationen verwenden. Eine Liste dieser Vorgänge finden Sie unter Kompatibilität des Access Points mit Diensten AWS.

Ein Aliasname eines Object Lambda Access Point wird innerhalb desselben Namespace wie ein Amazon-S3-Bucket erstellt. Dieser Aliasname wird automatisch generiert und kann nicht geändert werden. Für einen vorhandenen Object Lambda Access Point wird automatisch ein Alias zur Verwendung zugewiesen. Ein Aliasname eines Object Lambda Access Point erfüllt alle Anforderungen eines gültigen Amazon-S3-Bucketnamens und besteht aus den folgenden Teilen:

Object Lambda Access Point name prefix-*metadata*--ol-s3

> **Note**
>
> Das Suffix `--ol-s3` ist für Aliasnamen von Object Lambda Access Points reserviert und kann nicht für die Bucket-Namen oder die Namen von Object Lambda Access Points verwendet werden. Weitere Informationen zu Amazon-S3-Bucket-Benennungsregeln finden Sie unter Allgemeine Benennungsregeln für Buckets.

Die folgenden Beispiele zeigen den ARN und den Object Lambda Access Point-Alias für einen Object Lambda Access Point mit dem Namen: *my-object-lambda-access-point*

- ARN – arn:aws:s3-object-lambda:*region*:*account-id*:accesspoint/*my-object-lambda-access-point*
- Alias des Object Lambda Access Point – *my-object-lambda-acc*-*1a4n8yjrb3kda96f67zwrwiiuse1a*--ol-s3

Wenn Sie einen Object Lambda Access Point verwenden, können Sie den Aliasnamen des Object Lambda Access Point verwenden, ohne dass umfangreiche Codeänderungen erforderlich sind.

Wenn Sie einen Object Lambda Access Point löschen, wird der Aliasname des Object Lambda Access Point inaktiv und die Bereitstellung wird aufgehoben.

So finden Sie den Alias für Ihren Object Lambda Access Point

Verwenden der S3-Konsole

So finden Sie den Alias für Ihren Object Lambda Access Point mithilfe der Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
3. Kopieren Sie den Wert unter Alias des Objekt-Lambda-Zugriffspunkts für den Object Lambda Access Point, den Sie verwenden möchten.

Verwenden Sie den AWS CLI

Wenn Sie einen Object Lambda Access Point erstellen, generiert Amazon S3 automatisch einen Aliasnamen für den Object Lambda Access Point wie im folgenden Beispielbefehl gezeigt. Zum Ausführen dieses Befehls ersetzen Sie *`user input placeholders`* durch Ihre Informationen. Informationen zum Erstellen eines Object Lambda Access Points mithilfe von finden Sie AWS CLI unterSo erstellen Sie einen Object Lambda Access Point mit dem AWS CLI.

```
aws s3control create-access-point-for-object-lambda --account-id 111122223333 --
name my-object-lambda-access-point --configuration file://my-olap-configuration.json
{
    "ObjectLambdaAccessPointArn": "arn:aws:s3:region:111122223333:accesspoint/my-
access-point",
    "Alias": {
        "Value": "my-object-lambda-acc-1a4n8yjrb3kda96f67zwrwiiuse1a--ol-s3",
        "Status": "READY"
    }
}
```

Der generierte Aliasname des Object Lambda Access Point besteht aus zwei Feldern:

- Das Feld `Value` ist der Aliaswert des Object Lambda Access Point.
- Das Feld `Status` ist der Status des Alias des Object Lambda Access Point. Wenn der Status `PROVISIONING` lautet, stellt Amazon S3 den Alias des Object Lambda Access Point bereit und der Alias kann noch nicht verwendet werden. Wenn der Status `READY` lautet, wurde der Alias des Object Lambda Access Point erfolgreich bereitgestellt und kann verwendet werden.

Weitere Informationen zum `ObjectLambdaAccessPointAlias` Datentyp in der finden Sie REST API unter CreateAccessPointForObjectLambda und ObjectLambdaAccessPointAliasin der Amazon Simple Storage Service API Reference.

So verwenden Sie den Alias des Object Lambda Access Point

Sie können einen Alias eines Object Lambda Access Point anstelle eines Amazon-S3-Bucketnamens für die unter Kompatibilität des Access Points mit Diensten AWS aufgeführten Operationen verwenden.

Im folgenden AWS CLI Beispiel für den `get-bucket-location` Befehl wird der Access Point-Alias des Buckets verwendet, um den Wert zurückzugeben AWS-Region , in dem sich der Bucket befindet. Zum Ausführen dieses Befehls ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws s3api get-bucket-location --bucket my-object-lambda-
acc-w7i37nq6xuzgax3jw3oqtifiusw2a--ol-s3

{
    "LocationConstraint": "us-west-2"
}
```

Wenn der Alias des Object Lambda Access Point in einer Anforderung nicht gültig ist, wird der Fehlercode `InvalidAccessPointAliasError` zurückgegeben. Weitere Informationen zu `InvalidAccessPointAliasError` finden Sie in der Liste der Fehlercodes in der Amazon Simple Storage Service API Reference.

Die Einschränkungen eines Alias eines Object Lambda Access Point sind die gleichen wie bei einem Zugriffspunkt-Alias. Weitere Informationen zu den Einschränkungen eines Zugriffspunkt-Alias finden Sie unter Einschränkungen.

Sicherheitsüberlegungen für S3 Object Lambda-Zugriffspunkte

Mit Amazon S3 Object Lambda können Sie benutzerdefinierte Transformationen an Daten durchführen, wenn diese Amazon S3 verlassen, indem Sie den Umfang und die Flexibilität von AWS Lambda als Rechenplattform nutzen. S3 und Lambda bleiben standardmäßig sicher, aber besondere Rücksicht vom Lambda-Funktionsautor ist erforderlich, um diese Sicherheit zu gewährleisten. S3 Object Lambda erfordert, dass der gesamte Zugriff über authentifizierte Prinzipale (kein anonymer Zugriff) und mehr erfolgt. HTTPS

Es wird Folgendes empfohlen, um Sicherheitsrisiken zu minimieren:

- Beschränken Sie die Lambda-Ausführungsrolle auf den geringsten Satz von Berechtigungen.
- Stellen Sie nach Möglichkeit sicher, dass Ihre Lambda-Funktion über das bereitgestellte Presigned auf Amazon S3 zugreift. URL

Richtlinien konfigurieren IAM

S3-Zugriffspunkte unterstützen Ressourcenrichtlinien AWS Identity and Access Management (IAM), mit denen Sie die Nutzung des Access Points nach Ressourcen, Benutzern oder anderen Bedingungen steuern können. Weitere Informationen finden Sie unter Konfiguration von IAM Richtlinien für Object Lambda Access Points.

Verhalten der Verschlüsselung

Da Object Lambda Access Points sowohl Amazon S3 als auch verwenden AWS Lambda, gibt es Unterschiede im Verschlüsselungsverhalten. Weitere Informationen zum standardmäßigen S3-Verschlüsselungsverhalten finden Sie unter Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets.

- Wenn Sie die serverseitige Verschlüsselung in S3 mit Object Lambda Access Points verwenden, wird das Objekt entschlüsselt, bevor es an Lambda gesendet wird. Nachdem das Objekt an Lambda gesendet wurde, wird es unverschlüsselt verarbeitet (im Fall einer `GET`- oder `HEAD`-Anforderung).
- Um zu verhindern, dass der Verschlüsselungsschlüssel protokolliert wird, lehnt `GET` S3 `HEAD` Anfragen nach Objekten ab, die mithilfe einer serverseitigen Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) verschlüsselt wurden. SSE Die Lambda-Funktion kann diese Objekte jedoch weiterhin abrufen, sofern sie Zugriff auf den vom Client bereitgestellten Schlüssel hat.
- Wenn Sie die clientseitige Verschlüsselung in S3 mit Object Lambda Access Points verwenden, stellen Sie sicher, dass Lambda Zugriff auf den Verschlüsselungsschlüssel hat, damit es das Objekt entschlüsseln und erneut verschlüsseln kann.

Sicherheit für Zugriffspunkte

S3 Object Lambda verwendet zwei Zugriffspunkte, einen Object Lambda Access Point und einen standardmäßigen S3-Zugriffspunkt, der als unterstützender Zugriffspunkt bezeichnet wird. Wenn Sie eine Anforderung an einen Object Lambda Access Point ausführen, ruft S3 entweder Lambda in

Ihrem Namen auf oder delegiert die Anforderung an den unterstützenden Zugriffspunkt, abhängig von der Konfiguration von S3 Object Lambda. Wenn Lambda für eine Anfrage aufgerufen wird, generiert S3 in Ihrem Namen über URL den unterstützenden Access Point ein Ihrem Objekt vorsigniertes Objekt. Ihre Lambda-Funktion erhält dies URL als Eingabe, wenn die Funktion aufgerufen wird.

Sie können Ihre Lambda-Funktion so einstellen, dass sie dieses vorsignierte Objekt verwendet, URL um das ursprüngliche Objekt abzurufen, anstatt S3 direkt aufzurufen. Mit diesem Modell können Sie bessere Sicherheitsgrenzen auf Ihre Objekte anwenden. Sie können den direkten Objektzugriff über S3-Buckets oder S3-Zugriffspunkte auf eine begrenzte Anzahl von IAM Rollen oder Benutzern beschränken. Dieser Ansatz schützt Ihre Lambda-Funktionen auch davor, dem Problem des verwirrten Stellvertreters ausgesetzt zu sein, bei dem eine falsch konfigurierte Funktion mit anderen Berechtigungen als Ihr Aufrufer den Zugriff auf Objekte zulassen oder verweigern könnte, wenn dies nicht der Fall ist.

Öffentlicher Zugriff auf Objekt-Lambda-Zugriffspunkt

S3 Object Lambda erlaubt keinen anonymen oder öffentlichen Zugriff, da Amazon S3 Ihre Identität autorisieren muss, um eine S3-Object-Lambda-Anforderung abzuschließen. Wenn Sie Anforderungen über einen Object Lambda Access Point aufrufen, benötigen Sie die Berechtigung `lambda:InvokeFunction` für die konfigurierte Lambda-Funktion. Ebenso müssen Sie über die erforderlichen `s3:*` Berechtigungen verfügen, wenn Sie andere API Operationen über einen Object Lambda Access Point aufrufen.

Ohne diese Berechtigungen schlagen Anfragen zum Aufrufen von Lambda oder zum Delegieren an S3 mit HTTP 403-Fehlern (verboten) fehl. Jeder Zugriff muss von authentifizierten Prinzipalen erfolgen. Wenn Sie einen öffentlichen Zugang benötigen, können Sie Lambda@Edge als mögliche Alternative verwenden. Weitere Informationen finden Sie unter Customizing at the Edge with Lambda @Edge im Amazon CloudFront Developer Guide.

IP-Adressen von Object Lambda Access Points

Die `describe-managed-prefix-lists` Subnetze unterstützen Gateway-Endpunkte für virtuelle private Clouds (VPC) und beziehen sich auf die Routing-Tabelle der Endpunkte. VPC Da Object Lambda Access Point kein Gateway unterstützt, fehlen VPC seine IP-Bereiche. Die fehlenden Bereiche gehören zu Amazon S3, werden aber von VPC Gateway-Endpunkten nicht unterstützt. Weitere Informationen zu `describe-managed-prefix-lists` finden Sie DescribeManagedPrefixListsin der EC2APIAmazon-Referenz und AWS IP-Adressbereiche in der Allgemeine AWS-Referenz.

Konfiguration von IAM Richtlinien für Object Lambda Access Points

Amazon S3 Access Points unterstützen Ressourcenrichtlinien AWS Identity and Access Management (IAM), mit denen Sie die Nutzung des Access Points nach Ressourcen, Benutzern oder anderen Bedingungen steuern können. Sie können den Zugriff über eine optionale Ressourcenrichtlinie auf Ihrem Object Lambda Access Point oder eine Ressourcenrichtlinie für unterstützende Zugriffspunkte steuern. step-by-stepBeispiele finden Sie unter Tutorial: Transformieren von Daten für Ihre Anwendung mit S3 Object Lambda undTutorial: Erkennen und Redigieren von PII Daten mit S3 Object Lambda und Amazon Comprehend.

Die vier folgenden Ressourcen müssen über Berechtigungen verfügen, um mit Object Lambda Access Points zu arbeiten:

- Die IAM Identität, z. B. Benutzer oder Rolle. Weitere Informationen zu IAM Identitäten und bewährten Methoden finden Sie im Benutzerhandbuch unter IAMIdentitäten (Benutzer, Benutzergruppen und Rollen). IAM
- Der Bucket und der zugehörige Standardzugriffspunkt Wenn Sie mit Object Lambda Access Point arbeiten, wird dieser Standardzugriffspunkt als unterstützender Zugriffspunkt bezeichnet.
- Der Object Lambda Access Point.
- Die AWS Lambda Funktion.

> **Important**
>
> Bevor Sie Ihre Richtlinie speichern, stellen Sie sicher, dass Sie Sicherheitswarnungen, Fehler, allgemeine Warnungen und Vorschläge von behoben haben AWS Identity and Access Management Access Analyzer. IAMAccess Analyzer führt Richtlinienprüfungen durch, um Ihre Richtlinie anhand der IAM Richtliniengrammatik und der bewährten Methoden zu überprüfen. Diese Prüfungen generieren Ergebnisse und bieten umsetzbare Empfehlungen, die Sie beim Erstellen von Richtlinien unterstützen, die funktionsfähig sind und den bewährten Methoden für Sicherheit entsprechen.
>
> Weitere Informationen zur Überprüfung von Richtlinien mithilfe von IAM Access Analyzer finden Sie unter IAMAccess Analyzer-Richtlinienvalidierung im IAMBenutzerhandbuch. Eine Liste der Warnungen, Fehler und Vorschläge, die von Access Analyzer zurückgegeben werden, finden Sie unter IAM IAMAccess Analyzer-Referenz zur Richtlinienüberprüfung.

In den nachstehenden Richtlinienbeispielen wird davon ausgegangen, dass Sie die folgenden Ressourcen haben:

- Ein Amazon S3 S3-Bucket mit dem folgenden Amazon-Ressourcennamen (ARN):

  `arn:aws:s3:::amzn-s3-demo-bucket1`
- Ein Amazon S3 S3-Standardzugriffspunkt in diesem Bucket mit den folgenden EigenschaftenARN:

  `arn:aws:s3:us-east-1:111122223333:accesspoint/my-access-point`
- Ein Object Lambda Access Point mit den folgenden Eigenschaften: ARN

  `arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/my-object-lambda-ap`
- Eine AWS Lambda Funktion mit den folgenden Eigenschaften: ARN

  `arn:aws:lambda:us-east-1:111122223333:function:MyObjectLambdaFunction`

> **Note**
>
> Wenn Sie eine Lambda-Funktion von Ihrem Konto aus verwenden, müssen Sie die spezifische Funktionsversion in Ihre Richtlinienerklärung aufnehmen. Im folgenden Beispiel wird ARN die Version wie folgt angegeben *1*:
> `arn:aws:lambda:us-east-1:111122223333:function:MyObjectLambdaFunction:1`
> Lambda unterstützt das Hinzufügen von IAM Richtlinien zur Version $LATEST nicht. Weitere Informationen zu Lambda-Funktionsversionen finden Sie unter Lambda-Funktionsversionen im AWS Lambda -Entwicklerhandbuch.

Example – Bucket-Richtlinie zum Delegieren der Zugriffskontrolle an Standardzugriffspunkte

Das folgende Beispiel für eine S3-Bucket-Richtlinie delegiert die Zugriffskontrolle für einen Bucket an die Standardzugriffspunkte des Buckets. Diese Richtlinie ermöglicht Vollzugriff auf alle Zugriffspunkte, die dem Konto des Bucket-Eigentümers gehören. Somit wird der gesamte Zugriff auf diesen Bucket durch die an seine Zugriffspunkte angehängten Richtlinien gesteuert. Benutzer können nur über einen Zugriffspunkt aus dem Bucket lesen, was bedeutet, dass Operationen nur über Zugriffspunkte aufgerufen werden können. Weitere Informationen finden Sie unter Delegieren der Zugangskontrolle an Zugriffspunkte.

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS":"account-ARN"},
        "Action" : "*",
        "Resource" : [
            "arn:aws:s3:::amzn-s3-demo-bucket1",
            "arn:aws:s3:::amzn-s3-demo-bucket1/*"
        ],
        "Condition": {
            "StringEquals" : { "s3:DataAccessPointAccount" : "Bucket owner's account
 ID" }
        }
    }]
}
```

Example — IAM Richtlinie, die einem Benutzer die erforderlichen Berechtigungen zur Verwendung eines Object Lambda Access Points gewährt

Die folgende IAM Richtlinie gewährt einem Benutzer Berechtigungen für die Lambda-Funktion, den Standard-Access Point und den Object Lambda Access Point.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowLambdaInvocation",
      "Action": [
        "lambda:InvokeFunction"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:lambda:us-
east-1:111122223333:function:MyObjectLambdaFunction:1",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:CalledVia": [
            "s3-object-lambda.amazonaws.com"
          ]
        }
      }
    },
```

```
    {
      "Sid": "AllowStandardAccessPointAccess",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:us-east-1:111122223333:accesspoint/my-access-point/*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:CalledVia": [
            "s3-object-lambda.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "AllowObjectLambdaAccess",
      "Action": [
        "s3-object-lambda:Get*",
        "s3-object-lambda:List*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3-object-lambda:us-east-1:111122223333:accesspoint/my-
object-lambda-ap"
    }
  ]
}
```

Aktivieren von Berechtigungen für Lambda-Ausführungsrollen

Wenn `GET` Anfragen an einen Object Lambda Access Point gestellt werden, benötigt Ihre Lambda-Funktion die Erlaubnis, Daten an den S3 Object Lambda Access Point zu senden. Diese Berechtigung wird durch Aktivieren der `s3-object-lambda:WriteGetObjectResponse`-Berechtigung für die Ausführungsrolle Ihrer Lambda-Funktion bereitgestellt. Sie können eine neue Ausführungsrolle erstellen oder eine vorhandene Rolle aktualisieren.

> **Note**
>
> Ihre Funktion benötigt die `s3-object-lambda:WriteGetObjectResponse`-Berechtigung nur, wenn Sie eine `GET`-Anforderung stellen.

Um eine Ausführungsrolle in der Konsole zu erstellen IAM

1. Öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Roles aus.
3. Wählen Sie Rolle erstellen.
4. Wählen Sie unter Häufige Anwendungsfälle die Option Lambda aus.
5. Wählen Sie Weiter.
6. Suchen Sie auf der Seite „Berechtigungen hinzufügen“ nach der AWS verwalteten Richtlinie `AmazonS3ObjectLambdaExecutionRolePolicy`und aktivieren Sie dann das Kontrollkästchen neben dem Richtliniennamen.

   Diese Richtlinie sollte die Aktion `s3-object-lambda:WriteGetObjectResponse` enthalten.
7. Wählen Sie Weiter.
8. Geben Sie auf der Seite Name, review, and create (Name, prüfen und erstellen) für Role name (Rollenname) **`s3-object-lambda-role`** ein.
9. (Optional) Fügen Sie eine Beschreibung und Tags für diese Rolle hinzu.
10. Wählen Sie Rolle erstellen.
11. Wenden Sie die neu erstellte **`s3-object-lambda-role`** als Ausführungsrolle Ihrer Lambda-Funktion. Dies kann während oder nach der Erstellung der Lambda-Funktion in der Lambda-Konsole erfolgen.

Weitere Informationen zu Ausführungsrollen finden Sie unter Lambda-Ausführungsrolle im Entwicklerhandbuch für AWS Lambda .

Verwenden von Kontextschlüsseln mit Object Lambda Access Points

S3 Object Lambda wertet Kontextschlüssel wie `s3-object-lambda:TlsVersion` oder `s3-object-lambda:AuthType` im Zusammenhang mit der Verbindung oder dem Signieren der Anfrage aus. Alle anderen Kontextschlüssel, wie z. B. `s3:prefix`, werden von Amazon S3 ausgewertet.

Unterstützung für Object Lambda Access Point CORS

Wenn S3 Object Lambda eine Anforderung von einem Browser empfängt oder die Anforderung einen `Origin`-Header enthält, fügt S3 Object Lambda immer das Header-Feld `“AllowedOrigins":"*"` hinzu.

Weitere Informationen finden Sie unter Verwendung von ursprungsübergreifendem Ressourcenaustausch () CORS.

Schreiben von Lambda-Funktionen für S3 Object Lambda Access Points

In diesem Abschnitt wird beschrieben, wie AWS Lambda Funktionen für die Verwendung mit Amazon S3 Object Lambda Access Points geschrieben werden.

Informationen zu den vollständigen end-to-end Verfahren für einige S3 Object Lambda-Aufgaben finden Sie im Folgenden:

- Tutorial: Transformieren von Daten für Ihre Anwendung mit S3 Object Lambda
- Tutorial: Erkennen und Redigieren von PII Daten mit S3 Object Lambda und Amazon Comprehend
- Tutorial: Verwenden von S3 Object Lambda, um Bilder beim Abrufen dynamisch mit Wasserzeichen zu versehen

Themen

- Arbeiten mit GetObject-Anforderungen in Lambda
- Arbeiten mit HeadObject-Anforderungen in Lambda
- Arbeiten mit ListObjects-Anforderungen in Lambda
- Arbeiten mit ListObjectsV2-Anforderungen in Lambda
- Format und Verwendung des Ereigniskontexts
- Arbeitet mit Range and partNumber Header

Arbeiten mit **`GetObject`**-Anforderungen in Lambda

In diesem Abschnitt wird davon ausgegangen, dass Ihr Object Lambda Access Point für den Aufruf der Lambda-Funktion für `GetObject` konfiguriert ist. S3 Object Lambda umfasst den Amazon S3 API S3-Vorgang`WriteGetObjectResponse`, der es der Lambda-Funktion ermöglicht, dem Anrufer benutzerdefinierte Daten und Antwortheader bereitzustellen. `GetObject`

`WriteGetObjectResponse` bietet Ihnen eine umfassende Kontrolle über den Statuscode, die Antwort-Header und den Antworttext, basierend auf Ihren Verarbeitungsanforderungen. Sie können `WriteGetObjectResponse` verwenden, um mit dem gesamten transformierten Objekt, Teilen des transformierten Objekts oder anderen Antworten basierend auf dem Kontext Ihrer Anwendung zu

antworten. Der folgende Abschnitt zeigt einzigartige Beispiele für die Verwendung des Vorgangs. `WriteGetObjectResponse` API

- Beispiel 1: Antworten Sie mit dem HTTP Statuscode 403 (Forbidden)
- Beispiel 2: Reagieren Sie mit einem transformierten Bild
- Beispiel 3: Komprimierte Inhalte streamen

Beispiel 1: Antworten Sie mit dem HTTP Statuscode 403 (Forbidden)

Sie können dies verwenden`WriteGetObjectResponse`, um mit dem HTTP Statuscode 403 (Forbidden) zu antworten, basierend auf dem Inhalt des Objekts.

Java

```
package com.amazon.s3.objectlambda;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.events.S3ObjectLambdaEvent;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.WriteGetObjectResponseRequest;

import java.io.ByteArrayInputStream;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class Example1 {

    public void handleRequest(S3ObjectLambdaEvent event, Context context) throws
 Exception {
        AmazonS3 s3Client = AmazonS3Client.builder().build();

        // Check to see if the request contains all of the necessary information.
        // If it does not, send a 4XX response and a custom error code and message.
        // Otherwise, retrieve the object from S3 and stream it
        // to the client unchanged.
        var tokenIsNotPresent = !
event.getUserRequest().getHeaders().containsKey("requiredToken");
        if (tokenIsNotPresent) {
```

```
            s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
                    .withRequestRoute(event.outputRoute())
                    .withRequestToken(event.outputToken())
                    .withStatusCode(403)
                    .withContentLength(0L).withInputStream(new
 ByteArrayInputStream(new byte[0]))
                    .withErrorCode("MissingRequiredToken")
                    .withErrorMessage("The required token was not present in the
 request."));
            return;
        }

        // Prepare the presigned URL for use and make the request to S3.
        HttpClient httpClient = HttpClient.newBuilder().build();
        var presignedResponse = httpClient.send(
                HttpRequest.newBuilder(new URI(event.inputS3Url())).GET().build(),
                HttpResponse.BodyHandlers.ofInputStream());

        // Stream the original bytes back to the caller.
        s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
                .withRequestRoute(event.outputRoute())
                .withRequestToken(event.outputToken())
                .withInputStream(presignedResponse.body()));
    }
}
```

Python

```
import boto3
import requests

def handler(event, context):
    s3 = boto3.client('s3')

    """
    Retrieve the operation context object from the event. This object indicates
 where the WriteGetObjectResponse request
    should be delivered and contains a presigned URL in 'inputS3Url' where we can
 download the requested object from.
    The 'userRequest' object has information related to the user who made this
 'GetObject' request to
    S3 Object Lambda.
    """
```

```
    get_context = event["getObjectContext"]
    user_request_headers = event["userRequest"]["headers"]

    route = get_context["outputRoute"]
    token = get_context["outputToken"]
    s3_url = get_context["inputS3Url"]

    # Check for the presence of a 'CustomHeader' header and deny or allow based on
 that header.
    is_token_present = "SuperSecretToken" in user_request_headers

    if is_token_present:
        # If the user presented our custom 'SuperSecretToken' header, we send the
 requested object back to the user.
        response = requests.get(s3_url)
        s3.write_get_object_response(RequestRoute=route, RequestToken=token,
 Body=response.content)
    else:
        # If the token is not present, we send an error back to the user.
        s3.write_get_object_response(RequestRoute=route, RequestToken=token,
 StatusCode=403,
        ErrorCode="NoSuperSecretTokenFound", ErrorMessage="The request was not
 secret enough.")

    # Gracefully exit the Lambda function.
    return { 'status_code': 200 }
```

Node.js

```
const { S3 } = require('aws-sdk');
const axios = require('axios').default;

exports.handler = async (event) => {
    const s3 = new S3();

    // Retrieve the operation context object from the event. This object indicates
 where the WriteGetObjectResponse request
    // should be delivered and contains a presigned URL in 'inputS3Url' where we can
 download the requested object from.
    // The 'userRequest' object has information related to the user who made this
 'GetObject' request to S3 Object Lambda.
    const { userRequest, getObjectContext } = event;
    const { outputRoute, outputToken, inputS3Url } = getObjectContext;
```

```
    // Check for the presence of a 'CustomHeader' header and deny or allow based on
 that header.
    const isTokenPresent = Object
        .keys(userRequest.headers)
        .includes("SuperSecretToken");

    if (!isTokenPresent) {
        // If the token is not present, we send an error back to the user. The
 'await' in front of the request
        // indicates that we want to wait for this request to finish sending before
 moving on.
        await s3.writeGetObjectResponse({
            RequestRoute: outputRoute,
            RequestToken: outputToken,
            StatusCode: 403,
            ErrorCode: "NoSuperSecretTokenFound",
            ErrorMessage: "The request was not secret enough.",
        }).promise();
    } else {
        // If the user presented our custom 'SuperSecretToken' header, we send the
 requested object back to the user.
        // Again, note the presence of 'await'.
        const presignedResponse = await axios.get(inputS3Url);
        await s3.writeGetObjectResponse({
            RequestRoute: outputRoute,
            RequestToken: outputToken,
            Body: presignedResponse.data,
        }).promise();
    }

    // Gracefully exit the Lambda function.
    return { statusCode: 200 };
}
```

Beispiel 2: Reagieren Sie mit einem transformierten Bild

Wenn Sie eine Bildtransformation durchführen, stellen Sie möglicherweise fest, dass Sie alle Bytes des Quellobjekts benötigen, bevor Sie mit der Verarbeitung beginnen können. In diesem Fall gibt Ihre `WriteGetObjectResponse`-Anforderung das gesamte Objekt in einem Aufruf an die anfragende Anwendung zurück.

Java

```
package com.amazon.s3.objectlambda;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.events.S3ObjectLambdaEvent;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.WriteGetObjectResponseRequest;

import javax.imageio.ImageIO;
import java.awt.image.BufferedImage;
import java.awt.Image;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class Example2 {

    private static final int HEIGHT = 250;
    private static final int WIDTH = 250;

    public void handleRequest(S3ObjectLambdaEvent event, Context context) throws
 Exception {
        AmazonS3 s3Client = AmazonS3Client.builder().build();
        HttpClient httpClient = HttpClient.newBuilder().build();

        // Prepare the presigned URL for use and make the request to S3.
        var presignedResponse = httpClient.send(
                HttpRequest.newBuilder(new URI(event.inputS3Url())).GET().build(),
                HttpResponse.BodyHandlers.ofInputStream());

        // The entire image is loaded into memory here so that we can resize it.
        // Once the resizing is completed, we write the bytes into the body
        // of the WriteGetObjectResponse request.
        var originalImage = ImageIO.read(presignedResponse.body());
        var resizingImage = originalImage.getScaledInstance(WIDTH, HEIGHT,
 Image.SCALE_DEFAULT);
        var resizedImage = new BufferedImage(WIDTH, HEIGHT,
 BufferedImage.TYPE_INT_RGB);
```

```
        resizedImage.createGraphics().drawImage(resizingImage, 0, 0, WIDTH, HEIGHT,
 null);

        var baos = new ByteArrayOutputStream();
        ImageIO.write(resizedImage, "png", baos);

        // Stream the bytes back to the caller.
        s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
                .withRequestRoute(event.outputRoute())
                .withRequestToken(event.outputToken())
                .withInputStream(new ByteArrayInputStream(baos.toByteArray())));
    }
}
```

Python

```
import boto3
import requests
import io
from PIL import Image

def handler(event, context):
    """
    Retrieve the operation context object from the event. This object indicates
 where the WriteGetObjectResponse request
    should be delivered and has a presigned URL in 'inputS3Url' where we can
 download the requested object from.
    The 'userRequest' object has information related to the user who made this
 'GetObject' request to
    S3 Object Lambda.
    """
    get_context = event["getObjectContext"]
    route = get_context["outputRoute"]
    token = get_context["outputToken"]
    s3_url = get_context["inputS3Url"]

    """
    In this case, we're resizing .png images that are stored in S3 and are
 accessible through the presigned URL
    'inputS3Url'.
    """
    image_request = requests.get(s3_url)
    image = Image.open(io.BytesIO(image_request.content))
```

```
    image.thumbnail((256,256), Image.ANTIALIAS)

    transformed = io.BytesIO()
    image.save(transformed, "png")

    # Send the resized image back to the client.
    s3 = boto3.client('s3')
    s3.write_get_object_response(Body=transformed.getvalue(), RequestRoute=route,
 RequestToken=token)

    # Gracefully exit the Lambda function.
    return { 'status_code': 200 }
```

Node.js

```
const { S3 } = require('aws-sdk');
const axios = require('axios').default;
const sharp = require('sharp');

exports.handler = async (event) => {
    const s3 = new S3();

    // Retrieve the operation context object from the event. This object indicates
 where the WriteGetObjectResponse request
    // should be delivered and has a presigned URL in 'inputS3Url' where we can
 download the requested object from.
    const { getObjectContext } = event;
    const { outputRoute, outputToken, inputS3Url } = getObjectContext;

    // In this case, we're resizing .png images that are stored in S3 and are
 accessible through the presigned URL
    // 'inputS3Url'.
    const { data } = await axios.get(inputS3Url, { responseType: 'arraybuffer' });

    // Resize the image.
    const resized = await sharp(data)
        .resize({ width: 256, height: 256 })
        .toBuffer();

    // Send the resized image back to the client.
    await s3.writeGetObjectResponse({
        RequestRoute: outputRoute,
        RequestToken: outputToken,
```

```
        Body: resized,
    }).promise();

    // Gracefully exit the Lambda function.
    return { statusCode: 200 };
}
```

Beispiel 3: Komprimierte Inhalte streamen

Beim Komprimieren von Objekten werden komprimierte Daten inkrementell erzeugt. Folglich können Sie Ihre `WriteGetObjectResponse`-Anforderung verwenden, um die komprimierten Daten zurückzugeben, sobald sie bereit sind. Wie in diesem Beispiel gezeigt, ist es nicht notwendig, die Länge der abgeschlossenen Transformation zu kennen.

Java

```
package com.amazon.s3.objectlambda;

import com.amazonaws.services.lambda.runtime.events.S3ObjectLambdaEvent;
import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.WriteGetObjectResponseRequest;

import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class Example3 {

    public void handleRequest(S3ObjectLambdaEvent event, Context context) throws
 Exception {
        AmazonS3 s3Client = AmazonS3Client.builder().build();
        HttpClient httpClient = HttpClient.newBuilder().build();

        // Request the original object from S3.
        var presignedResponse = httpClient.send(
                HttpRequest.newBuilder(new URI(event.inputS3Url())).GET().build(),
                HttpResponse.BodyHandlers.ofInputStream());
```

```
        // Consume the incoming response body from the presigned request,
        // apply our transformation on that data, and emit the transformed bytes
        // into the body of the WriteGetObjectResponse request as soon as they're
 ready.
        // This example compresses the data from S3, but any processing pertinent
        // to your application can be performed here.
        var bodyStream = new GZIPCompressingInputStream(presignedResponse.body());

        // Stream the bytes back to the caller.
        s3Client.writeGetObjectResponse(new WriteGetObjectResponseRequest()
                .withRequestRoute(event.outputRoute())
                .withRequestToken(event.outputToken())
                .withInputStream(bodyStream));
    }

}
```

Python

```
import boto3
import requests
import zlib
from botocore.config import Config


"""
A helper class to work with content iterators. Takes an interator and compresses the
 bytes that come from it. It
implements 'read' and '__iter__' so that the SDK can stream the response.
"""
class Compress:
    def __init__(self, content_iter):
        self.content = content_iter
        self.compressed_obj = zlib.compressobj()

    def read(self, _size):
        for data in self.__iter__()
            return data

    def __iter__(self):
        while True:
            data = next(self.content)
            chunk = self.compressed_obj.compress(data)
```

```
            if not chunk:
                break

            yield chunk

        yield self.compressed_obj.flush()


def handler(event, context):
    """
    Setting the 'payload_signing_enabled' property to False allows us to send a
 streamed response back to the client.
    in this scenario, a streamed response means that the bytes are not buffered into
 memory as we're compressing them,
    but instead are sent straight to the user.
    """
    my_config = Config(
        region_name='eu-west-1',
        signature_version='s3v4',
        s3={
            "payload_signing_enabled": False
        }
    )
    s3 = boto3.client('s3', config=my_config)

    """
    Retrieve the operation context object from the event. This object indicates
 where the WriteGetObjectResponse request
    should be delivered and has a presigned URL in 'inputS3Url' where we can
 download the requested object from.
    The 'userRequest' object has information related to the user who made this
 'GetObject' request to S3 Object Lambda.
    """
    get_context = event["getObjectContext"]
    route = get_context["outputRoute"]
    token = get_context["outputToken"]
    s3_url = get_context["inputS3Url"]

    # Compress the 'get' request stream.
    with requests.get(s3_url, stream=True) as r:
        compressed = Compress(r.iter_content())

        # Send the stream back to the client.
```

```
        s3.write_get_object_response(Body=compressed, RequestRoute=route,
 RequestToken=token, ContentType="text/plain",
                                     ContentEncoding="gzip")

    # Gracefully exit the Lambda function.
    return {'status_code': 200}
```

Node.js

```
const { S3 } = require('aws-sdk');
const axios = require('axios').default;
const zlib = require('zlib');

exports.handler = async (event) => {
    const s3 = new S3();

    // Retrieve the operation context object from the event. This object indicates
 where the WriteGetObjectResponse request
    // should be delivered and has a presigned URL in 'inputS3Url' where we can
 download the requested object from.
    const { getObjectContext } = event;
    const { outputRoute, outputToken, inputS3Url } = getObjectContext;

    // Download the object from S3 and process it as a stream, because it might be a
 huge object and we don't want to
    // buffer it in memory. Note the use of 'await' because we want to wait for
 'writeGetObjectResponse' to finish
    // before we can exit the Lambda function.
    await axios({
        method: 'GET',
        url: inputS3Url,
        responseType: 'stream',
    }).then(
        // Gzip the stream.
        response => response.data.pipe(zlib.createGzip())
    ).then(
        // Finally send the gzip-ed stream back to the client.
        stream => s3.writeGetObjectResponse({
            RequestRoute: outputRoute,
            RequestToken: outputToken,
            Body: stream,
            ContentType: "text/plain",
            ContentEncoding: "gzip",
```

```
        }).promise()
    );

    // Gracefully exit the Lambda function.
    return { statusCode: 200 };
}
```

> **Note**
>
> Obwohl S3 Object Lambda ermöglicht, bis zu 60 Sekunden eine vollständige Antwort an den Aufrufer über die `WriteGetObjectResponse`-Anforderung zu senden, ist die tatsächlich verfügbare Zeit möglicherweise geringer. Zum Beispiel kann Ihr Timeout für Lambda-Funktionen weniger als 60 Sekunden betragen. In anderen Fällen kann der Aufrufer strengere Zeitbeschränkungen haben.

Damit der ursprüngliche Anrufer eine andere Antwort als HTTP den Statuscode 500 (Interner Serverfehler) erhält, muss der `WriteGetObjectResponse` Anruf abgeschlossen sein. Wenn die Lambda-Funktion mit einer Ausnahme oder auf andere Weise zurückkehrt, bevor die `WriteGetObjectResponse` API Operation aufgerufen wird, erhält der ursprüngliche Aufrufer eine Antwort von 500 (Internal Server Error). Ausnahmen, die während der Zeit ausgelöst werden, die zum Abschließen der Antwort benötigt wird, führen zu verkürzten Antworten an den Aufrufer. Wenn die Lambda-Funktion vom `WriteGetObjectResponse` API Anruf eine Antwort mit dem HTTP Statuscode 200 (OK) erhält, hat der ursprüngliche Aufrufer die vollständige Anfrage gesendet. Die Antwort der Lambda-Funktion, wird von S3 Object Lambda, unabhängig davon, ob eine Ausnahme ausgelöst wird oder nicht, ignoriert.

Beim Aufrufen des `WriteGetObjectResponse` API Vorgangs benötigt Amazon S3 das Route- und Anforderungstoken aus dem Ereigniskontext. Weitere Informationen finden Sie unter Format und Verwendung des Ereigniskontexts.

Die Routen- und Anforderungstokenparameter sind erforderlich, um die `WriteGetObjectResult`-Antwort mit dem ursprünglichen Aufrufer zu verbinden. Obwohl es immer angemessen ist, 500 Antworten (Interner Serverfehler) erneut zu versuchen, da es sich bei dem Anforderungstoken um ein einmalig verwendbares Token handelt, können nachfolgende Versuche, es zu Antworten mit dem HTTP Statuscode 400 (Bad Request) führen. Obwohl der Aufruf an `WriteGetObjectResponse` mit den Routen- und Anforderungstoken nicht von der aufgerufenen Lambda-Funktion erfolgen muss, ist

hierfür eine Identität im selben Konto erforderlich. Der Aufruf muss außerdem abgeschlossen sein, bevor die Lambda-Funktion die Ausführung beendet.

Arbeiten mit **`HeadObject`**-Anforderungen in Lambda

In diesem Abschnitt wird davon ausgegangen, dass Ihr Object Lambda Access Point für den Aufruf der Lambda-Funktion für `HeadObject` konfiguriert ist. Lambda erhält eine JSON Nutzlast, die einen Schlüssel namens enthält. `headObjectContext` Innerhalb des Kontextes wird eine einzelne Eigenschaft aufgerufen`inputS3Url`, die URL für den unterstützenden Zugriffspunkt vorsigniert ist. `HeadObject`

Die vorsignierte Eigenschaft enthält URL die folgenden Eigenschaften, sofern sie angegeben sind:

- `versionId` (in den Abfrageparametern)
- `requestPayer` (im `x-amz-request-payer`-Header)
- `expectedBucketOwner` (im `x-amz-expected-bucket-owner`-Header)

Andere Eigenschaften werden nicht vorsigniert und daher nicht berücksichtigt. Nicht signierte Optionen, die als Header gesendet werden, können manuell zur Anfrage hinzugefügt werden, wenn die vorsignierte Option aufgerufen wirdURL, die sich in den Headern befindet. `userRequest` Serverseitige Verschlüsselungsoptionen werden für `HeadObject` nicht unterstützt.

Die URI Syntaxparameter für Anfragen finden Sie `HeadObject`in der Amazon Simple Storage Service API Reference.

Das folgende Beispiel zeigt eine JSON Lambda-Eingabenutzlast für. `HeadObject`

```
{
  "xAmzRequestId": "requestId",
  "**headObjectContext**": {
    "**inputS3Url**": "https://my-s3-ap-111122223333.s3-accesspoint.us-
east-1.amazonaws.com/example?X-Amz-Security-Token=<snip>"
  },
  "configuration": {
        "accessPointArn": "arn:aws:s3-object-lambda:us-
east-1:111122223333:accesspoint/example-object-lambda-ap",
        "supportingAccessPointArn": "arn:aws:s3:us-
east-1:111122223333:accesspoint/example-ap",
        "payload": "{}"
  },
```

```
  "userRequest": {
       "url": "https://object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com/example",
       "headers": {
           "Host": "object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com",
           "Accept-Encoding": "identity",
           "X-Amz-Content-SHA256": "e3b0c44298fc1example"
       }
   },
   "userIdentity": {
       "type": "AssumedRole",
       "principalId": "principalId",
       "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
       "accountId": "111122223333",
       "accessKeyId": "accessKeyId",
       "sessionContext": {
            "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
       },
       "sessionIssuer": {
            "type": "Role",
            "principalId": "principalId",
            "arn": "arn:aws:iam::111122223333:role/Admin",
            "accountId": "111122223333",
            "userName": "Admin"
            }
       }
    },
  "protocolVersion": "1.00"
}
```

Ihre Lambda-Funktion sollte ein JSON Objekt zurückgeben, das die Header und Werte enthält, die für den `HeadObject` Aufruf zurückgegeben werden.

Das folgende Beispiel zeigt die Struktur der Lambda-Antwort JSON für`HeadObject`.

```
{
    "statusCode": <number>; // Required
    "errorCode": <string>;
    "errorMessage": <string>;
    "headers": {
```

```
        "Accept-Ranges": <string>,
        "x-amz-archive-status": <string>,
        "x-amz-server-side-encryption-bucket-key-enabled": <boolean>,
        "Cache-Control": <string>,
        "Content-Disposition": <string>,
        "Content-Encoding": <string>,
        "Content-Language": <string>,
        "Content-Length": <number>, // Required
        "Content-Type": <string>,
        "x-amz-delete-marker": <boolean>,
        "ETag": <string>,
        "Expires": <string>,
        "x-amz-expiration": <string>,
        "Last-Modified": <string>,
        "x-amz-missing-meta": <number>,
        "x-amz-object-lock-mode": <string>,
        "x-amz-object-lock-legal-hold": <string>,
        "x-amz-object-lock-retain-until-date": <string>,
        "x-amz-mp-parts-count": <number>,
        "x-amz-replication-status": <string>,
        "x-amz-request-charged": <string>,
        "x-amz-restore": <string>,
        "x-amz-server-side-encryption": <string>,
        "x-amz-server-side-encryption-customer-algorithm": <string>,
        "x-amz-server-side-encryption-aws-kms-key-id": <string>,
        "x-amz-server-side-encryption-customer-key-MD5": <string>,
        "x-amz-storage-class": <string>,
        "x-amz-tagging-count": <number>,
        "x-amz-version-id": <string>,
        <x-amz-meta-headers>: <string>, // user-defined metadata
        "x-amz-meta-meta1": <string>, // example of the user-defined metadata header,
 it will need the x-amz-meta prefix
        "x-amz-meta-meta2": <string>
        ...
    };
}
```

Das folgende Beispiel zeigt, wie Sie Presigned verwenden, um Ihre Antwort aufzufüllenURL, indem Sie die Header-Werte nach Bedarf ändern, bevor Sie die zurückgeben. JSON

Python

```
import requests
```

```
def lambda_handler(event, context):
    print(event)

    # Extract the presigned URL from the input.
    s3_url = event["headObjectContext"]["inputS3Url"]

    # Get the head of the object from S3.
    response = requests.head(s3_url)

    # Return the error to S3 Object Lambda (if applicable).
    if (response.status_code >= 400):
        return {
            "statusCode": response.status_code,
            "errorCode": "RequestFailure",
            "errorMessage": "Request to S3 failed"
    }

    # Store the headers in a dictionary.
    response_headers = dict(response.headers)

    # This obscures Content-Type in a transformation, it is optional to add
    response_headers["Content-Type"] = ""

    # Return the headers to S3 Object Lambda.
    return {
        "statusCode": response.status_code,
        "headers": response_headers
        }
```

Arbeiten mit **`ListObjects`**-Anforderungen in Lambda

In diesem Abschnitt wird davon ausgegangen, dass Ihr Object Lambda Access Point für den Aufruf der Lambda-Funktion für `ListObjects` konfiguriert ist. Lambda empfängt die JSON Nutzlast mit einem neuen Objekt namens. `listObjectsContext` `listObjectsContext`enthält eine einzelne Eigenschaft`inputS3Url`, die URL für den unterstützenden Access Point vorsigniert ist. `ListObjects`

Im Gegensatz zu `GetObject` und `HeadObject` enthält das Presigned URL die folgenden Eigenschaften, wenn sie angegeben sind:

- Alle Abfrageparameter

- `requestPayer` (im `x-amz-request-payer`-Header)
- `expectedBucketOwner` (im `x-amz-expected-bucket-owner`-Header)

Die URI Syntaxparameter für Anfragen finden Sie `ListObjects`in der Amazon Simple Storage Service API Reference.

> **Important**
>
> Wir empfehlen, bei der Entwicklung von Anwendungen die neuere Version ListObjectsV2 zu verwenden. Aus Gründen der Abwärtskompatibilität unterstützt Amazon S3 weiterhin `ListObjects`.

Das folgende Beispiel zeigt die JSON Lambda-Eingabenutzlast für. `ListObjects`

```
{
    "xAmzRequestId": "requestId",
     "**listObjectsContext**": {
     "**inputS3Url**": "https://my-s3-ap-111122223333.s3-accesspoint.us-
east-1.amazonaws.com/?X-Amz-Security-Token=<snip>",
     },
    "configuration": {
        "accessPointArn": "arn:aws:s3-object-lambda:us-
east-1:111122223333:accesspoint/example-object-lambda-ap",
        "supportingAccessPointArn": "arn:aws:s3:us-
east-1:111122223333:accesspoint/example-ap",
        "payload": "{}"
    },
    "userRequest": {
        "url": "https://object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com/example",
        "headers": {
            "Host": "object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com",
            "Accept-Encoding": "identity",
            "X-Amz-Content-SHA256": "e3b0c44298fc1example"
        }
    },
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "principalId",
```

```
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
        "accountId": "111122223333",
        "accessKeyId": "accessKeyId",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "principalId",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            }
        }
    },
  "protocolVersion": "1.00"
}
```

Ihre Lambda-Funktion sollte ein JSON Objekt zurückgeben, das den Statuscode, das XML Listenergebnis oder die Fehlerinformationen enthält, die von S3 Object Lambda zurückgegeben werden.

S3 Object Lambda verarbeitet oder validiert `listResultXml` nicht, sondern leitet es an den `ListObjects`-Aufrufer weiter. Für `listBucketResult` erwartet S3 Object Lambda, dass bestimmte Eigenschaften von einem speziellen Typ sind, und löst Ausnahmen aus, wenn es sie nicht analysieren kann. `listResultXml` und `listBucketResult` können nicht gleichzeitig bereitgestellt werden.

Das folgende Beispiel zeigt, wie Sie das Presigned verwendenURL, um Amazon S3 aufzurufen und das Ergebnis zu verwenden, um eine Antwort zu füllen, einschließlich Fehlerprüfung.

Python

```
import requests
import xmltodict

def lambda_handler(event, context):
    # Extract the presigned URL from the input.
    s3_url = event["listObjectsContext"]["inputS3Url"]
```

```
    # Get the head of the object from Amazon S3.
    response = requests.get(s3_url)

    # Return the error to S3 Object Lambda (if applicable).
    if (response.status_code >= 400):
        error = xmltodict.parse(response.content)
        return {
            "statusCode": response.status_code,
            "errorCode": error["Error"]["Code"],
            "errorMessage": error["Error"]["Message"]
        }

    # Store the XML result in a dict.
    response_dict = xmltodict.parse(response.content)

    # This obscures StorageClass in a transformation, it is optional to add
    for item in response_dict['ListBucketResult']['Contents']:
        item['StorageClass'] = ""

    # Convert back to XML.
    listResultXml = xmltodict.unparse(response_dict)

    # Create response with listResultXml.
    response_with_list_result_xml = {
        'statusCode': 200,
        'listResultXml': listResultXml
    }

    # Create response with listBucketResult.
    response_dict['ListBucketResult'] =
 sanitize_response_dict(response_dict['ListBucketResult'])
    response_with_list_bucket_result = {
        'statusCode': 200,
        'listBucketResult': response_dict['ListBucketResult']
    }

    # Return the list to S3 Object Lambda.
    # Can return response_with_list_result_xml or response_with_list_bucket_result
    return response_with_list_result_xml

# Converting the response_dict's key to correct casing
def sanitize_response_dict(response_dict: dict):
    new_response_dict = dict()
```

```
    for key, value in response_dict.items():
        new_key = key[0].lower() + key[1:] if key != "ID" else 'id'
        if type(value) == list:
            newlist = []
            for element in value:
                if type(element) == type(dict()):
                    element = sanitize_response_dict(element)
                newlist.append(element)
            value = newlist
        elif type(value) == dict:
            value = sanitize_response_dict(value)
        new_response_dict[new_key] = value
    return new_response_dict
```

Das folgende Beispiel zeigt die Struktur der Lambda-Antwort JSON für`ListObjects`.

```
{
  "statusCode": <number>; // Required
  "errorCode": <string>;
  "errorMessage": <string>;
  "listResultXml": <string>; // This can also be Error XML string in case S3 returned
 error response when calling the pre-signed URL

  "listBucketResult": {  // listBucketResult can be provided instead of listResultXml,
 however they can not both be provided in the JSON response
        "name": <string>,  // Required for 'listBucketResult'
        "prefix": <string>,
        "marker": <string>,
        "nextMarker": <string>,
        "maxKeys": <int>,   // Required for 'listBucketResult'
        "delimiter": <string>,
        "encodingType": <string>
        "isTruncated": <boolean>,  // Required for 'listBucketResult'
        "contents": [  {
            "key": <string>,  // Required for 'content'
            "lastModified": <string>,
            "eTag": <string>,
            "checksumAlgorithm": <string>,   // CRC32,  CRC32C,  SHA1,  SHA256
            "size": <int>,   // Required for 'content'
            "owner": {
                "displayName": <string>,  // Required for 'owner'
                "id": <string>,  // Required for 'owner'
```

```
            },
            "storageClass": <string>
            },
        ...
        ],
        "commonPrefixes": [  {
            "prefix": <string>   // Required for 'commonPrefix'
        },
        ...
        ],
    }
}
```

Arbeiten mit **`ListObjectsV2`**-Anforderungen in Lambda

In diesem Abschnitt wird davon ausgegangen, dass Ihr Object Lambda Access Point für den Aufruf der Lambda-Funktion für `ListObjectsV2` konfiguriert ist. Lambda empfängt die JSON Nutzlast mit einem neuen Objekt namens. `listObjectsV2Context` `listObjectsV2Context`enthält eine einzelne Eigenschaft`inputS3Url`, die URL für den unterstützenden Access Point vorsigniert ist. `ListObjectsV2`

Im Gegensatz zu `GetObject` und `HeadObject` enthält das Presigned URL die folgenden Eigenschaften, sofern sie angegeben sind:

- Alle Abfrageparameter
- `requestPayer` (im `x-amz-request-payer`-Header)
- `expectedBucketOwner` (im `x-amz-expected-bucket-owner`-Header)

Die URI Syntaxparameter für Anfragen finden Sie `ListObjectsV2`in der Amazon Simple Storage Service API Reference.

Das folgende Beispiel zeigt die JSON Lambda-Eingabenutzlast für. `ListObjectsV2`

```
{
    "xAmzRequestId": "requestId",
     "**listObjectsV2Context**": {
     "**inputS3Url**": "https://my-s3-ap-111122223333.s3-accesspoint.us-
east-1.amazonaws.com/?list-type=2&X-Amz-Security-Token=<snip>",
     },
    "configuration": {
```

```
        "accessPointArn": "arn:aws:s3-object-lambda:us-
east-1:111122223333:accesspoint/example-object-lambda-ap",
        "supportingAccessPointArn": "arn:aws:s3:us-
east-1:111122223333:accesspoint/example-ap",
        "payload": "{}"
    },
    "userRequest": {
        "url": "https://object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com/example",
        "headers": {
            "Host": "object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com",
            "Accept-Encoding": "identity",
            "X-Amz-Content-SHA256": "e3b0c44298fc1example"
        }
    },
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "principalId",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
        "accountId": "111122223333",
        "accessKeyId": "accessKeyId",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "principalId",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            }
        }
    },
  "protocolVersion": "1.00"
}
```


Ihre Lambda-Funktion sollte ein JSON Objekt zurückgeben, das den Statuscode, das XML Listenergebnis oder die Fehlerinformationen enthält, die von S3 Object Lambda zurückgegeben werden.

S3 Object Lambda verarbeitet oder validiert `listResultXml` nicht, sondern leitet es an den `ListObjectsV2`-Aufrufer weiter. Für `listBucketResult` erwartet S3 Object Lambda, dass bestimmte Eigenschaften von einem speziellen Typ sind, und löst Ausnahmen aus, wenn es sie nicht analysieren kann. `listResultXml` und `listBucketResult` können nicht gleichzeitig bereitgestellt werden.

Das folgende Beispiel zeigt, wie Sie das Presigned verwendenURL, um Amazon S3 aufzurufen und das Ergebnis zu verwenden, um eine Antwort zu füllen, einschließlich Fehlerprüfung.

Python

```
import requests
import xmltodict

def lambda_handler(event, context):
    # Extract the presigned URL from the input.
    s3_url = event["listObjectsV2Context"]["inputS3Url"]


    # Get the head of the object from Amazon S3.
    response = requests.get(s3_url)

    # Return the error to S3 Object Lambda (if applicable).
    if (response.status_code >= 400):
        error = xmltodict.parse(response.content)
        return {
            "statusCode": response.status_code,
            "errorCode": error["Error"]["Code"],
            "errorMessage": error["Error"]["Message"]
        }

    # Store the XML result in a dict.
    response_dict = xmltodict.parse(response.content)

    # This obscures StorageClass in a transformation, it is optional to add
    for item in response_dict['ListBucketResult']['Contents']:
        item['StorageClass'] = ""

    # Convert back to XML.
    listResultXml = xmltodict.unparse(response_dict)

    # Create response with listResultXml.
    response_with_list_result_xml = {
```

```
        'statusCode': 200,
        'listResultXml': listResultXml
    }

    # Create response with listBucketResult.
    response_dict['ListBucketResult'] =
 sanitize_response_dict(response_dict['ListBucketResult'])
    response_with_list_bucket_result = {
        'statusCode': 200,
        'listBucketResult': response_dict['ListBucketResult']
    }

    # Return the list to S3 Object Lambda.
    # Can return response_with_list_result_xml or response_with_list_bucket_result
    return response_with_list_result_xml

# Converting the response_dict's key to correct casing
def sanitize_response_dict(response_dict: dict):
    new_response_dict = dict()
    for key, value in response_dict.items():
        new_key = key[0].lower() + key[1:] if key != "ID" else 'id'
        if type(value) == list:
            newlist = []
            for element in value:
                if type(element) == type(dict()):
                    element = sanitize_response_dict(element)
                newlist.append(element)
            value = newlist
        elif type(value) == dict:
            value = sanitize_response_dict(value)
        new_response_dict[new_key] = value
    return new_response_dict
```

Das folgende Beispiel zeigt die Struktur der Lambda-Antwort JSON für`ListObjectsV2`.

```
{
    "statusCode": <number>; // Required
    "errorCode": <string>;
    "errorMessage": <string>;
    "listResultXml": <string>; // This can also be Error XML string in case S3 returned
 error response when calling the pre-signed URL
```

```
    "listBucketResult": {  // listBucketResult can be provided instead of
 listResultXml, however they can not both be provided in the JSON response
        "name": <string>, // Required for 'listBucketResult'
        "prefix": <string>,
        "startAfter": <string>,
        "continuationToken": <string>,
        "nextContinuationToken": <string>,
        "keyCount": <int>, // Required for 'listBucketResult'
        "maxKeys": <int>, // Required for 'listBucketResult'
        "delimiter": <string>,
        "encodingType": <string>
        "isTruncated": <boolean>, // Required for 'listBucketResult'
        "contents": [ {
            "key": <string>, // Required for 'content'
            "lastModified": <string>,
            "eTag": <string>,
            "checksumAlgorithm": <string>, // CRC32, CRC32C, SHA1, SHA256
            "size": <int>, // Required for 'content'
            "owner": {
                "displayName": <string>, // Required for 'owner'
                "id": <string>, // Required for 'owner'
            },
            "storageClass": <string>
            },
            ...
        ],
        "commonPrefixes": [ {
            "prefix": <string> // Required for 'commonPrefix'
            },
        ...
        ],
    }
}
```

Format und Verwendung des Ereigniskontexts

Amazon S3 Object Lambda bietet Kontext zu der Anfrage, die in dem Ereignis gestellt wird, das an Ihre AWS Lambda Funktion übergeben wird. Nachstehend finden Sie eine Beispielanforderung: Beschreibungen der Felder folgen nach dem Beispiel.

```
{
    "xAmzRequestId": "requestId",
    "getObjectContext": {
```

```
        "inputS3Url": "https://my-s3-ap-111122223333.s3-accesspoint.us-
east-1.amazonaws.com/example?X-Amz-Security-Token=<snip>",
        "outputRoute": "io-use1-001",
        "outputToken": "OutputToken"
    },
    "configuration": {
        "accessPointArn": "arn:aws:s3-object-lambda:us-
east-1:111122223333:accesspoint/example-object-lambda-ap",
        "supportingAccessPointArn": "arn:aws:s3:us-
east-1:111122223333:accesspoint/example-ap",
        "payload": "{}"
    },
    "userRequest": {
        "url": "https://object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com/example",
        "headers": {
            "Host": "object-lambda-111122223333.s3-object-lambda.us-
east-1.amazonaws.com",
            "Accept-Encoding": "identity",
            "X-Amz-Content-SHA256": "e3b0c44298fc1example"
        }
    },
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "principalId",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/example",
        "accountId": "111122223333",
        "accessKeyId": "accessKeyId",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "Wed Mar 10 23:41:52 UTC 2021"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "principalId",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            }
        }
    },
    "protocolVersion": "1.00"
```

```
}
```

Die folgenden Felder sind in der Anforderung enthalten:

- `xAmzRequestId` – Die Amazon-S3-Anforderungs-ID für diese Anforderung. Wir empfehlen, diesen Wert zu protokollieren, um beim Debuggen zu helfen.
- `getObjectContext` – Die Eingabe- und Ausgabedetails für Verbindungen zu Amazon S3 und S3 Object Lambda.
  - `inputS3Url`— Eine vorsignierteURL, die verwendet werden kann, um das Originalobjekt von Amazon S3 abzurufen. Der URL ist mit der Identität des ursprünglichen Anrufers signiert, und die Berechtigungen dieses Benutzers gelten, wenn der URL verwendet wird. Wenn der signierte Header enthältURL, muss die Lambda-Funktion diese Header in den Aufruf von Amazon S3 einbeziehen, mit Ausnahme des Headers. `Host`
  - `outputRoute`— Ein Routing-Token, das dem S3 Object Lambda hinzugefügt wird, URL wenn die Lambda-Funktion aufgerufen wird. `WriteGetObjectResponse`
  - `outputToken` – Ein undurchsichtiges Token, das von S3 Object Lambda verwendet wird, um den `WriteGetObjectResponse`-Aufruf mit dem ursprünglichen Aufrufer abzugleichen.
- `configuration` – Informationen zur Konfiguration des Object Lambda Access Point.
  - `accessPointArn`— Der Amazon-Ressourcenname (ARN) des Object Lambda Access Points, der diese Anfrage erhalten hat.
  - `supportingAccessPointArn`— Der ARN des unterstützenden Access Points, der in der Object Lambda Access Point-Konfiguration angegeben ist.
  - `payload` – Benutzerdefinierte Daten, die auf die Konfiguration des Object Lambda Access Point angewendet werden. S3 Object Lambda behandelt diese Daten als eine undurchsichtige Zeichenfolge, daher muss sie möglicherweise vor der Verwendung dekodiert werden.
- `userRequest` – Informationen über den ursprünglichen Aufruf von S3 Object Lambda.
  - `url`— Die Dekodierung URL der Anfrage, wie sie von S3 Object Lambda empfangen wurde, mit Ausnahme aller autorisierungsbezogenen Abfrageparameter.
  - `headers`— Eine Zuordnung von Zeichenketten zu Zeichenketten, die die HTTP Header und ihre Werte aus dem ursprünglichen Aufruf enthalten, ausgenommen alle autorisierungsbezogenen Header. Wenn derselbe Header mehrfach erscheint, werden die Werte von jeder Instance desselben Header zu einer durch Kommata getrennten Liste zusammengefasst. Der Fall der ursprünglichen Header wird in dieser Zuordnung beibehalten.

- `userIdentity` – Details zur die Identität, die den Aufruf von S3 Object Lambda getätigt hat. Weitere Informationen finden Sie unter Protokollieren von Datenereignissen für Trails im AWS CloudTrail -Benutzerhandbuch.
  - `type` – Die Art der Identität.
  - `accountId`— Der, AWS-Konto zu dem die Identität gehört.
  - `userName` – Der Anzeigename der Identität, von der der Aufruf stammt.
  - `principalId` – Die eindeutige ID für die Entität, von der der Aufruf stammt.
  - `arn`— Der ARN des Schulleiters, der den Anruf getätigt hat. Der letzte Abschnitt von ARN enthält den Benutzer oder die Rolle, die den Anruf getätigt hat.
  - `sessionContext` – Erfolgte die Abfrage mittels temporärer Sicherheitsanmeldeinformationen, stellt dieses Element Informationen über die Sitzung bereit, das für diese Anmeldeinformationen erstellt wurde.
  - `invokedBy`— Der Name desjenigen AWS-Service , der die Anfrage gestellt hat, z. B. Amazon EC2 Auto Scaling oder AWS Elastic Beanstalk.
  - `sessionIssuer` – Erfolgte die Abfrage mittels temporärer Sicherheitsanmeldeinformationen, gibt dieses Element Auskunft darüber, wie die Anmeldeinformationen erhalten wurden.
- `protocolVersion` – Die Versions-ID des bereitgestellten Kontextes. Das Format des Felds ist `{Major Version}.{Minor Version}`. Die Nebenversionsnummern sind immer zweistellige Zahlen. Jede Entfernung oder Änderung der Semantik eines Feldes erfordert einen Sprung der Hauptversion und erfordert ein aktives Opt-In. Amazon S3 kann jederzeit neue Felder hinzufügen. Zu diesem Zeitpunkt kann es zu einem Sprung der Nebenversion kommen. Aufgrund der Art der Software-Einführung sehen Sie möglicherweise die Verwendung mehrerer Nebenversionen gleichzeitig.

Arbeitet mit Range and partNumber Header

Wenn Sie in Amazon S3 Object Lambda mit großen Objekten arbeiten, können Sie den `Range` HTTP Header verwenden, um einen bestimmten Bytebereich von einem Objekt herunterzuladen. Sie können gleichzeitige Verbindungen zu Amazon S3 verwenden, um verschiedene Bytebereiche aus demselben Objekt abzurufen. Sie können auch den `partNumber`-Parameter (eine Ganzzahl zwischen 1 und 10 000) verwenden, der eine „ranged“ Anforderung für das angegebene Teil des Objekts ausführt.

Weil es mehrere Möglichkeiten gibt, eine Anforderung zu bearbeiten, die den `Range`- oder `partNumber`-Parameter beinhaltet, wendet S3 Object Lambda diese Parameter nicht auf das

transformierte Objekt an. Stattdessen muss Ihre AWS Lambda Funktion diese Funktionalität nach Bedarf für Ihre Anwendung implementieren.

Wenn Sie die Parameter `Range` und `partNumber` mit S3 Object Lambda verwenden möchten, gehen Sie wie folgt vor:

- Aktivieren Sie diese Parameter in der Konfiguration Ihres Object Lambda Access Point.
- Schreiben Sie eine Lambda-Funktion, die Anforderungen verarbeiten kann, die diese Parameter enthalten.

In den folgenden Schritten wird beschrieben, wie Sie dies erreichen.

Schritt 1: Konfigurieren Ihres Object Lambda Access Point

Standardmäßig antworten Object Lambda Access Points mit dem HTTP Statuscode 501 (Nicht implementiert) auf jede `HeadObject` OR-Anfrage, die einen `GetObject` `partNumber` OR-Parameter enthält, entweder in den Headern `Range` oder in den Abfrageparametern.

Damit ein Object-Lambda-Zugriffspunkt solche Anforderungen annehmen kann, müssen Sie `GetObject-Range`, `GetObject-PartNumber`, `HeadObject-Range` oder `HeadObject-PartNumber` in den `AllowedFeatures`-Abschnitt die Konfiguration Ihres Object-Lambda-Zugangspunkts einschließen. Weitere Informationen zur Aktualisierung der Konfiguration Ihres Object Lambda Access Point finden Sie unter Erstellen von Objekt-Lambda-Zugriffspunkten.

Schritt 2: Implementieren Sie das **`Range`** oder **`partNumber`** in Ihrer Lambda-Funktion

Wenn Ihr Object Lambda Access Point Ihre Lambda-Funktion mit einer `GetObject`- oder `HeadObject`-Bereichsanforderung aufruft, ist der `Range`- oder `partNumber`-Parameter im Ereigniskontext enthalten. Die Position des Parameters im Ereigniskontext hängt davon ab, welcher Parameter verwendet wurde und wie er in die ursprüngliche Anforderung an den Object Lambda Access Point aufgenommen wurde, wie in der folgenden Tabelle erläutert.

Parameter	Ort des Ereigniskontexts
`Range` (Header)	`userRequest.headers.Range`
`Range` (Abfragep arameter)	`userRequest.url` (Abfrageparameter `Range`)

Parameter	Ort des Ereigniskontexts
`partNumber`	`userRequest.url` (Abfrageparameter `partNumber`)

> **Important**
>
> Das bereitgestellte, URL für Ihren Object Lambda Access Point vorsignierte Objekt enthält nicht den `partNumber` Parameter `Range` oder aus der ursprünglichen Anfrage. In den folgenden Optionen erfahren Sie, wie Sie diese Parameter in Ihrer AWS Lambda Funktion handhaben können.

Nachdem Sie den `Range`- oder `partNumber`-Wert extrahiert haben, können Sie einen der folgenden Ansätze wählen, basierend auf den Anforderungen Ihrer Anwendung:

A. Ordnen Sie den angeforderten **`Range`**- oder **`partNumber`**-Wert dem transformierten Objekt zu (empfohlen).

Die zuverlässigste Art, `Range`- oder `partNumber`-Anforderungen zu bearbeiten, ist folgende:

- Rufen Sie das vollständige Objekt aus Amazon S3 ab.
- Transformieren Sie das Objekt.
- Wenden Sie die angeforderten `Range`- oder `partNumber`-Parameter auf das transformierte Objekt an.

Verwenden Sie dazu das bereitgestellte Presigned, URL um das gesamte Objekt von Amazon S3 abzurufen und das Objekt dann nach Bedarf zu verarbeiten. Ein Beispiel für eine Lambda-Funktion, die einen `Range` Parameter auf diese Weise verarbeitet, finden Sie in diesem Beispiel im AWS GitHub Samples-Repository.

B. Ordnen Sie das Angeforderte dem **`Range`** URL Vorsignierten zu.

In einigen Fällen kann Ihre Lambda-Funktion die Anfrage `Range` direkt der vorsignierten zuordnen, URL um nur einen Teil des Objekts aus Amazon S3 abzurufen. Dieser Ansatz ist nur dann geeignet, wenn Ihre Transformation die beiden folgenden Kriterien erfüllt:

1. Ihre Transformationsfunktion kann auf partielle Objektbereiche angewendet werden.
2. Das Anwenden der `Range`-Parameter vor oder nach der Transformationsfunktion führt zu demselben transformierten Objekt.

Beispielsweise erfüllt eine Transformationsfunktion, die alle Zeichen in einem ASCII -codierten Objekt in Großbuchstaben umwandelt, beide der oben genannten Kriterien. Die Transformation kann auf einen Teil eines Objekts angewendet werden, und durch Anwenden des `Range`-Parameters vor der Transformation wird das gleiche Ergebnis erzielt wie beim Anwenden des Parameters nach der Transformation.

Im Gegensatz dazu erfüllt eine Funktion, die die Zeichen in einem ASCII -codierten Objekt umkehrt, diese Kriterien nicht. Eine solche Funktion erfüllt Kriterium 1, da sie auf partielle Objektbereiche angewendet werden kann. Sie erfüllt jedoch nicht Kriterium 2, da das Anwenden des `Range`-Parameters vor der Transformation andere Ergebnisse erzielt als das Anwenden des Parameters nach der Transformation.

Betrachten Sie eine Anforderung, die Funktion auf die ersten drei Zeichen eines Objekts mit dem Inhalt `abcdefg` anzuwenden. Das Anwenden des `Range`-Parameters vor der Transformation ruft nur `abcab` und kehrt dann die Daten um und gibt `cba` zurück. Wenn der Parameter jedoch nach der Transformation angewendet wird, ruft die Funktion das gesamte Objekt ab, kehrt es um und wendet dann den `Range`-Parameter, um `gfe` zurückzugeben. Da diese Ergebnisse unterschiedlich sind, sollte diese Funktion den `Range`-Parameter beim Abrufen des Objekts von Amazon S3 nicht anwenden. Stattdessen sollte es das gesamte Objekt abrufen, die Transformation durchführen und erst dann den `Range`-Parameter anwenden.

> **Warning**
>
> In vielen Fällen führt die Anwendung des `Range` Parameters auf den vorsignierten Client URL zu unerwartetem Verhalten der Lambda-Funktion oder des anfordernden Clients. Wenn Sie nicht sicher sind, dass Ihre Anwendung ordnungsgemäß funktioniert, wenn Sie nur ein Teilobjekt von Amazon S3 abrufen, empfehlen wir Ihnen, vollständige Objekte abzurufen und zu transformieren, wie zuvor in Ansatz A beschrieben.

Wenn Ihre Anwendung die zuvor in Ansatz B beschriebenen Kriterien erfüllt, können Sie Ihre AWS Lambda Funktion vereinfachen, indem Sie nur den angeforderten Objektbereich abrufen und dann Ihre Transformation für diesen Bereich ausführen.

Im folgenden Java-Codebeispiel wird folgende Vorgehensweise gezeigt:

- Rufen Sie den `Range`-Header aus der `GetObject`-Anforderung ab.

- Fügen Sie dem Vorsignierten den `Range` Header hinzuURL, den Lambda verwenden kann, um den angeforderten Bereich von Amazon S3 abzurufen.

```
private HttpRequest.Builder applyRangeHeader(ObjectLambdaEvent event,
 HttpRequest.Builder presignedRequest) {
    var header = event.getUserRequest().getHeaders().entrySet().stream()
            .filter(e -> e.getKey().toLowerCase(Locale.ROOT).equals("range"))
            .findFirst();

    // Add check in the query string itself.
    header.ifPresent(entry -> presignedRequest.header(entry.getKey(),
 entry.getValue()));
    return presignedRequest;
}
```

Verwenden von AWS integrierten Lambda-Funktionen

AWS bietet einige vorgefertigte AWS Lambda Funktionen, die Sie mit Amazon S3 Object Lambda verwenden können, um personenbezogene Daten (PII) zu erkennen und zu redigieren und S3-Objekte zu dekomprimieren. Diese Lambda-Funktionen sind in AWS Serverless Application Repository verfügbar. Sie können diese Funktionen über die AWS Management Console auswählen, wenn Sie Ihren Object Lambda Access Point erstellen.

Weitere Informationen zur Bereitstellung serverloser Anwendungen finden Sie unter Deploying Applications im AWS Serverless Application Repository Developer Guide.AWS Serverless Application Repository

> **Note**
>
> Die folgenden Beispiele können nur mit `GetObject`-Anforderungen verwendet werden.

Beispiel 1: PII Zugriffskontrolle

Diese Lambda-Funktion verwendet Amazon Comprehend, einen Service zur Verarbeitung natürlicher Sprache (NLP), der maschinelles Lernen nutzt, um Erkenntnisse und Zusammenhänge im Text zu finden. Diese Funktion erkennt automatisch personenbezogene Daten (PII) wie Namen, Adressen, Daten, Kreditkartennummern und Sozialversicherungsnummern in Dokumenten in Ihrem Amazon S3 S3-Bucket. Wenn Sie Dokumente in Ihrem Bucket haben, die Folgendes beinhaltenPII, können

Sie die PII Zugriffskontrollfunktion so konfigurieren, dass sie diese PII Entitätstypen erkennt und den Zugriff auf nicht autorisierte Benutzer einschränkt.

Stellen Sie zunächst die folgende Lambda-Funktion in Ihrem Konto bereit und fügen Sie den Amazon-Ressourcennamen (ARN) für die Funktion zu Ihrer Object Lambda Access Point-Konfiguration hinzu.

Im Folgenden finden Sie ein Beispiel ARN für diese Funktion:

```
arn:aws:serverlessrepo:us-east-1:111122223333:applications/
ComprehendPiiAccessControlS3ObjectLambda
```

Sie können diese Funktion auf dem hinzufügen oder anzeigen, AWS Management Console indem Sie den folgenden AWS Serverless Application Repository Link verwenden: ComprehendPiiAccessControlS3 ObjectLambda.

Informationen zur Anzeige dieser Funktion finden Sie unter Amazon Comprehend S3 Object Lambda. GitHub

Beispiel 2: Schwärzung PII

Diese Lambda-Funktion verwendet Amazon Comprehend, einen Service zur Verarbeitung natürlicher Sprache (NLP), der maschinelles Lernen nutzt, um Erkenntnisse und Zusammenhänge im Text zu finden. Diese Funktion löscht automatisch personenbezogene Daten (PII), wie Namen, Adressen, Daten, Kreditkartennummern und Sozialversicherungsnummern, aus Dokumenten in Ihrem Amazon S3 S3-Bucket.

Wenn Sie Dokumente in Ihrem Bucket haben, die Informationen wie Kreditkartennummern oder Bankkontoinformationen enthalten, können Sie die PII Redaction S3 Object Lambda-Funktion so konfigurieren, dass sie diese Dokumente, in denen PII Entitätstypen geschwärzt sind, erkennt PII und anschließend eine Kopie zurückgibt.

Stellen Sie zunächst die folgende Lambda-Funktion in Ihrem Konto bereit und fügen Sie die ARN für die Funktion zu Ihrer Object Lambda Access Point-Konfiguration hinzu.

Im Folgenden finden Sie ein Beispiel ARN für diese Funktion:

```
arn:aws:serverlessrepo:us-east-1:111122223333::applications/
ComprehendPiiRedactionS3ObjectLambda
```

Sie können diese Funktion auf dem hinzufügen oder anzeigen, AWS Management Console indem Sie den folgenden AWS Serverless Application Repository Link verwenden: ComprehendPiiRedactionS3 ObjectLambda.

Informationen zur Anzeige dieser Funktion finden Sie unter Amazon Comprehend S3 Object Lambda. GitHub

Vollständige end-to-end Verfahren für einige S3 Object Lambda-Aufgaben in PII Redaction finden Sie unter. Tutorial: Erkennen und Redigieren von PII Daten mit S3 Object Lambda und Amazon Comprehend

Beispiel 3: Dekompression

Die Lambda-Funktion `S3ObjectLambdaDecompression` kann Objekte, die in Amazon S3 gespeichert sind, in einem von sechs komprimierten Dateiformaten dekomprimieren: `bzip2`, `gzip`, `snappy`, `zlib`, `zstandard` und `ZIP`.

Stellen Sie zunächst die folgende Lambda-Funktion in Ihrem Konto bereit und fügen Sie die ARN für die Funktion zu Ihrer Object Lambda Access Point-Konfiguration hinzu.

Im Folgenden finden Sie ein Beispiel ARN für diese Funktion:

```
arn:aws:serverlessrepo:us-east-1:111122223333::applications/S3ObjectLambdaDecompression
```

Sie können diese Funktion auf dem hinzufügen oder anzeigen, AWS Management Console indem Sie den folgenden AWS Serverless Application Repository Link verwenden: S3 ObjectLambdaDecompression.

Informationen zur Aktivierung dieser Funktion finden Sie GitHub unter S3 Object Lambda Decompression.

Bewährte Methoden und Richtlinien für S3 Object Lambda

Befolgen Sie bei der Verwendung von S3 Object Lambda diesen bewährten Methoden und Richtlinien, um den Betrieb und die Leistung zu optimieren.

Themen

- Arbeiten mit S3 Object Lambda
- AWS-Services wird in Verbindung mit S3 Object Lambda verwendet

- Range- und partNumber-Header
- Transformieren von expiry-date
- Arbeiten mit dem und AWS CLIAWS SDKs

Arbeiten mit S3 Object Lambda

S3 Object Lambda unterstützt nur die Verarbeitung von `GET`-, `LIST`- und `HEAD`-Anforderungen. Alle anderen Anfragen rufen keine standardmäßigen, nicht transformierten Antworten auf AWS Lambda und geben stattdessen zurück. API Sie können maximal 1.000 Object Lambda Access Points AWS-Konto pro Region erstellen. Die AWS Lambda Funktion, die Sie verwenden, muss sich in derselben AWS-Konto Region wie der Object Lambda Access Point befinden.

S3 Object Lambda lässt bis zu 60 Sekunden zu, um eine vollständige Antwort zu seinen Aufrufer zu streamen. Ihre Funktion unterliegt außerdem AWS Lambda Standardkontingenten. Weitere Informationen finden Sie unter Lambda quotas (Lambda-Kontingente) im AWS Lambda - Entwicklerhandbuch.

Mit S3 Object Lambda wird Ihre angegebene Lambda-Funktion aufgerufen. Sie sind dafür verantwortlich, dass alle Daten, die von Ihrer angegebenen Lambda-Funktion oder -Anwendung aus S3 überschrieben oder gelöscht werden, beabsichtigt und korrekt sind.

Sie können S3 Object Lambda nur verwenden, um Operationen auf Objekten durchzuführen. Sie können S3 Object Lambda nicht verwenden, um andere Amazon-S3-Operationen auszuführen, z. B. das Ändern oder Löschen von Buckets. Eine vollständige Liste der S3-Vorgänge, die Zugriffspunkte unterstützen, finden Sie unter Zugriffspunkt-Kompatibilität mit S3-Vorgänge.

Zusätzlich zu dieser Liste unterstützen Object Lambda Access Points die `SelectObjectContent`APIOperationen `POST Object`, `CopyObject`(als Quelle) und nicht.

AWS-Services wird in Verbindung mit S3 Object Lambda verwendet

S3 Object Lambda verbindet Amazon S3 und optional andere AWS-Services Ihrer Wahl, um Objekte bereitzustellen, die für die anfragenden Anwendungen relevant sind. AWS Lambda Alle, die mit S3 Object Lambda AWS-Services verwendet werden, unterliegen ihren jeweiligen Service Level Agreements (SLAs). Wenn beispielsweise einer seiner AWS-Service Serviceverpflichtungen nicht nachkommt, haben Sie Anspruch auf eine Servicegutschrift, wie in den Serviceverträgen dokumentiert. SLA

`Range`- und **`partNumber`**-Header

Wenn Sie mit großen Objekten arbeiten, können Sie den `Range` HTTP Header verwenden, um einen bestimmten Bytebereich von einem Objekt herunterzuladen. Wenn Sie den `Range`-Header verwenden, ruft Ihre Anforderung nur den angegebenen Teil des Objekts ab. Mit dem `partNumber`-Header können Sie auch eine Bereichsanforderung für das angegebene Teil aus dem Objekt ausführen.

Weitere Informationen finden Sie unter Arbeitet mit Range and partNumber Header.

Transformieren von **`expiry-date`**

Sie können transformierte Objekte von Ihrem Object Lambda Access Point auf dem AWS Management Consoleöffnen oder herunterladen. Diese Objekte dürfen nicht abgelaufen sein. Wenn Ihre Lambda-Funktion das `expiry-date` Ihrer Objekte transformiert, sehen Sie möglicherweise abgelaufene Objekte, die nicht geöffnet oder heruntergeladen werden können. Dieses Verhalten gilt nur für wiederhergestellte Objekte von S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive.

Arbeiten mit dem und AWS CLIAWS SDKs

AWS Command Line Interface (AWS CLI) S3-Unterbefehle (`cp`, `mv`, und`sync`) und die Verwendung der AWS SDK for Java `TransferManager` Klasse werden für die Verwendung mit S3 Object Lambda nicht unterstützt.

Tutorials zu S3 Object Lambda

In den folgenden Tutorials end-to-end werden vollständige Verfahren für einige S3 Object Lambda-Aufgaben vorgestellt.

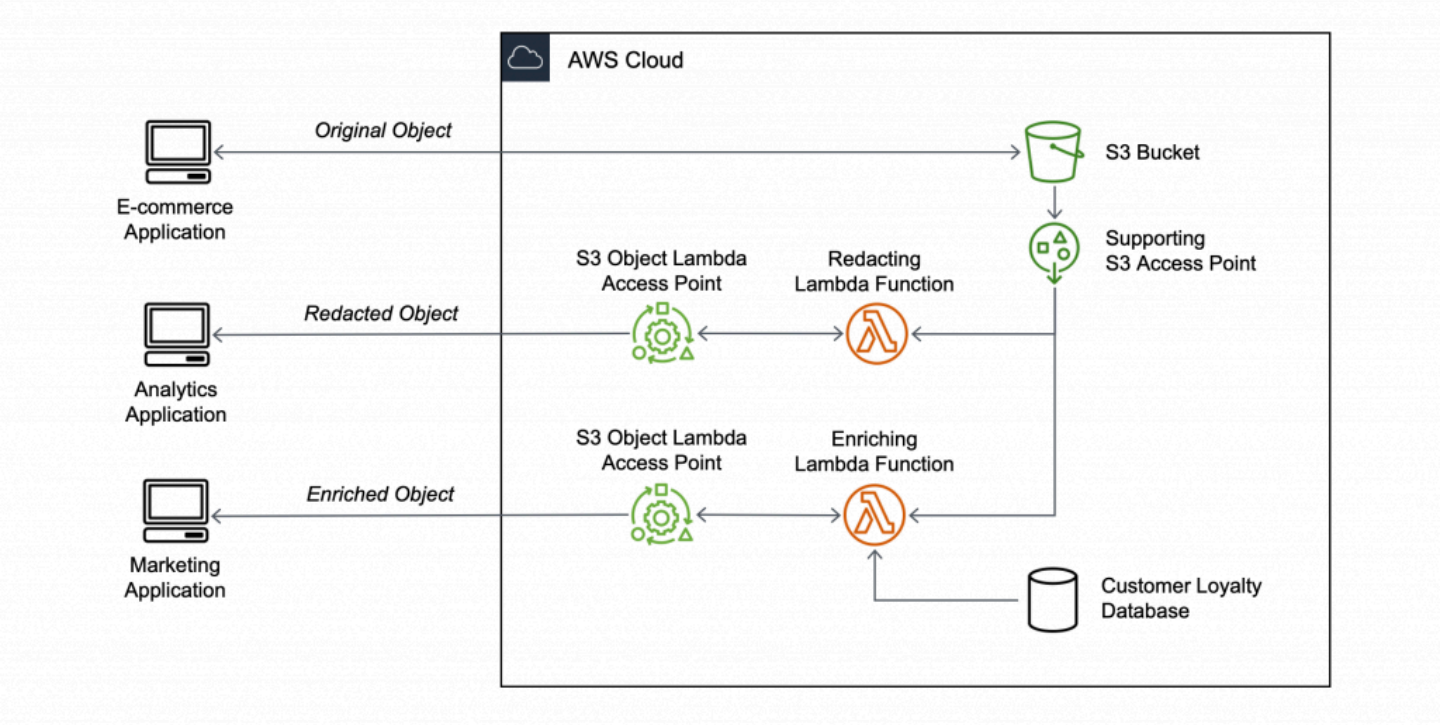
Mit S3 Object Lambda können Sie Ihren eigenen Code hinzufügen, um Daten zu verarbeiten, die von S3 abgerufen werden, bevor Sie sie an eine Anwendung zurückgeben. In jedem der folgenden Tutorials werden Daten geändert, sobald sie aus Amazon S3 abgerufen werden, ohne das bestehende Objekt zu ändern oder mehrere Kopien der Daten zu verwalten. Im ersten Tutorial wird erklärt, wie Sie einer GET S3-Anfrage eine AWS Lambda Funktion hinzufügen, um ein von S3 abgerufenes Objekt zu ändern. Das zweite Tutorial zeigt, wie Sie mit einer vordefinierten Lambda-Funktion, die von Amazon Comprehend unterstützt wird, personenbezogene Daten (PII) schützen, die aus S3 abgerufen wurden, bevor sie an eine Anwendung zurückgegeben werden. Das dritte Tutorial verwendet S3 Object Lambda, um einem Bild, das aus Amazon S3 abgerufen wird, ein Wasserzeichen hinzuzufügen.

- Tutorial: Transformieren von Daten für Ihre Anwendung mit S3 Object Lambda
- Tutorial: Erkennen und Redigieren von PII Daten mit S3 Object Lambda und Amazon Comprehend
- Tutorial: Verwenden von S3 Object Lambda, um Bilder beim Abrufen dynamisch mit Wasserzeichen zu versehen

Tutorial: Transformieren von Daten für Ihre Anwendung mit S3 Object Lambda

Wenn Sie Daten in Amazon S3 speichern, können Sie sie für mehrere Anwendungen verwenden. Jede Anwendung hat jedoch möglicherweise eindeutige Anforderungen an das Datenformat und benötigt möglicherweise eine Änderung oder Verarbeitung Ihrer Daten für einen bestimmten Anwendungsfall. Beispielsweise kann ein von einer E-Commerce-Anwendung erstellter Datensatz personenbezogene Daten (PII) enthalten. Wenn dieselben Daten für Analysen verarbeitet werden, PII ist dies nicht erforderlich und sollte redigiert werden. Wenn jedoch derselbe Datensatz für eine Marketingkampagne verwendet wird, müssen Sie die Daten möglicherweise mit zusätzlichen Details wie Informationen aus der Kundenbindungsdatenbank anreichern.

Mit S3 Object Lambda können Sie Ihren eigenen Code hinzufügen, um Daten zu verarbeiten, die von S3 abgerufen werden, bevor Sie sie an eine Anwendung zurückgeben. Insbesondere können Sie eine AWS Lambda Funktion konfigurieren und sie an einen S3 Object Lambda Access Point anhängen. Wenn eine Anwendung standardmäßige GET S3-Anfragen über den S3 Object Lambda Access Point sendet, wird die angegebene Lambda-Funktion aufgerufen, um alle Daten zu verarbeiten, die aus einem S3-Bucket über den unterstützenden S3-Zugriffspunkt abgerufen wurden. Dann gibt der S3 Object Lambda Access Point das transformierte Ergebnis zurück an die Anwendung aus. Sie können Ihre eigenen benutzerdefinierten Lambda-Funktionen erstellen und ausführen, indem Sie die S3 Object Lambda-Datentransformation an Ihren spezifischen Anwendungsfall anpassen, ohne dass Änderungen an Ihren Anwendungen erforderlich sind.



Ziel

In diesem Tutorial erfahren Sie, wie Sie standardmäßigen GET S3-Anfragen benutzerdefinierten Code hinzufügen, um das von S3 abgerufene angeforderte Objekt so zu ändern, dass das Objekt den Anforderungen des anfragenden Clients oder der anfragenden Anwendung entspricht. Insbesondere erfahren Sie, wie Sie den gesamten Text im ursprünglichen Objekt, das in S3 gespeichert ist, über S3 Object Lambda in Großbuchstaben umwandeln.

> **Note**
>
> Dieses Tutorial verwendet Python-Code, um die Daten zu transformieren. Beispiele für die Verwendung anderer AWS SDKs finden Sie unter Transformieren von Daten für Ihre Anwendung mit S3 Object Lambda in der AWS SDK Codebeispielbibliothek.

Themen

- Voraussetzungen
- Schritt 1: Einen S3-Bucket erstellen
- Schritt 2: Hochladen einer Datei zu einem S3-Bucket
- Schritt 3: Erstellen eines S3-Zugriffspunkts

- Schritt 4: Erstellen einer Lambda-Funktion
- Schritt 5: Konfigurieren Sie eine IAM Richtlinie für die Ausführungsrolle Ihrer Lambda-Funktion
- Schritt 6: Erstellen eines S3 Object Lambda Access Point
- Schritt 7: Anzeigen der transformierten Daten
- Schritt 8: Bereinigen
- Nächste Schritte

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, benötigen Sie einen AWS-Konto , bei dem Sie sich als AWS Identity and Access Management (IAM) -Benutzer mit den richtigen Berechtigungen anmelden können. Sie müssen außerdem Python Version 3.8 oder höher installieren.

Teilschritte

- Erstellen Sie einen IAM Benutzer mit Berechtigungen in Ihrer AWS-Konto (Konsole)
- Installieren Sie Python 3.8 oder höher auf Ihrem lokalen Computer

Erstellen Sie einen IAM Benutzer mit Berechtigungen in Ihrer AWS-Konto (Konsole)

Sie können einen IAM Benutzer für das Tutorial erstellen. Um dieses Tutorial abzuschließen, muss Ihr IAM Benutzer die folgenden IAM Richtlinien anhängen, um auf relevante AWS Ressourcen zuzugreifen und bestimmte Aktionen auszuführen. Weitere Informationen zum Erstellen eines IAM Benutzers finden Sie im Benutzerhandbuch unter IAMBenutzer erstellen (Konsole). IAM

Ihr IAM Benutzer benötigt die folgenden Richtlinien:

- AmazonS3 FullAccess — Erteilt Berechtigungen für alle Amazon S3-Aktionen, einschließlich der Berechtigungen zum Erstellen und Verwenden eines Object Lambda Access Points.
- AWSLambda_ FullAccess — Erteilt Berechtigungen für alle Lambda-Aktionen.
- IAMFullAccess— Erteilt Berechtigungen für alle IAM Aktionen.
- IAMAccessAnalyzerReadOnlyAccess— Erteilt Berechtigungen zum Lesen aller von Access Analyzer bereitgestellten IAM Zugriffsinformationen.
- CloudWatchLogsFullAccess— Gewährt vollen Zugriff auf CloudWatch Logs.

> **Note**
>
> Der Einfachheit halber erstellt und verwendet dieses Tutorial einen IAM Benutzer. Denken Sie nach Abschluss dieses Tutorials an Löschen Sie den IAM Benutzer. Für den produktiven Einsatz empfehlen wir, dass Sie die bewährten Sicherheitsmethoden IAM im IAM Benutzerhandbuch befolgen. Eine bewährte Methode ist, dass menschliche Benutzer den Verbund mit einem Identitätsanbieter verwenden, um mit temporären Anmeldeinformationen auf AWS zuzugreifen. Eine weitere bewährte Methode besteht darin, Workloads die Verwendung temporärer Anmeldeinformationen mit IAM Rollen für den Zugriff AWS vorzuschreiben. Informationen AWS IAM Identity Center zur Erstellung von Benutzern mit temporären Anmeldeinformationen finden Sie unter Erste Schritte im AWS IAM Identity Center Benutzerhandbuch.
> In diesem Tutorial werden auch AWS -verwaltete Richtlinien mit vollem Zugriff verwendet. Für die Verwendung in der Produktion empfehlen wir, dass Sie stattdessen nur die für Ihren Anwendungsfall erforderlichen Mindestberechtigungen gemäß Bewährte Methoden in Bezug auf die Sicherheit erteilen.

Installieren Sie Python 3.8 oder höher auf Ihrem lokalen Computer

Verwenden Sie das folgende Verfahren, um Python 3.8 oder höher auf Ihrem lokalen Computer zu installieren. Weitere Installationsanweisungen finden Sie auf der Seite Python herunterladen im Python-Anfängerhandbuch.

1. Öffnen Sie das lokale Terminal oder die Shell, und führen Sie den folgenden Befehl aus, um festzustellen, ob Python bereits installiert ist, und wenn ja, welche Version installiert ist.

   ```
   python --version
   ```

2. Wenn Sie nicht über Python 3.8 oder höher verfügen, laden Sie die Datei offizieller Installer von Python 3.8 oder höher herunter, die für Ihren lokalen Computer geeignet ist.
3. Führen Sie das Installationsprogramm aus, indem Sie auf die heruntergeladene Datei doppelklicken und die Schritte ausführen, um die Installation abzuschließen.

   Windows-Benutzer wählen Python 3.X hinzufügen PATH im Installationsassistenten, bevor Sie Jetzt installieren wählen.
4. Starten Sie das Terminal neu, indem Sie es schließen und erneut öffnen.

5. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob Python 3.8 oder höher ordnungsgemäß installiert ist.

   Führen Sie für macOS-Benutzer diesen Befehl aus:

   ```
   python3 --version
   ```

   Führen Sie für Windows-Benutzer diesen Befehl aus:

   ```
   python --version
   ```

6. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob der pip3-Paket-Manager installiert ist. Wenn Sie in der Befehlsantwort eine Pip-Versionsnummer und Python 3.8 oder höher sehen, bedeutet dies, dass der pip3-Paketmanager erfolgreich installiert wurde.

   ```
   pip --version
   ```

Schritt 1: Einen S3-Bucket erstellen

Erstellen Sie einen Bucket zum Speichern der ursprünglichen Daten, die Sie transformieren möchten.

So erstellen Sie einen Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie Create Bucket (Bucket erstellen) aus.

   Die Seite Create bucket (Bucket erstellen) wird geöffnet.
4. Geben Sie für Bucket-Name einen Namen für Ihren Bucket ein (z. B. **`tutorial-bucket`**).

   Weitere Informationen zur Benennung von Buckets in Amazon S3 finden Sie unter Allgemeine Benennungsregeln für Buckets.
5. Wählen Sie unter Region den AWS-Region Ort aus, an dem sich der Bucket befinden soll.

   Weitere Informationen zur Bucket-Region finden Sie unter Bucket-Übersicht.
6. Belassen Sie die Einstellungen für den öffentlichen Zugriff für diesen Bucket blockieren bei den Standardeinstellungen (Alle öffentlichen Zugriffe blockieren ist aktiviert).

Es wird empfohlen, alle Einstellungen für öffentlichen Zugriff blockieren aktiviert zu belassen, es sei denn, Sie müssen eine oder mehrere dieser Einstellungen für Ihren Anwendungsfall deaktivieren. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

7. Lassen Sie die restlichen Einstellungen auf die Standardwerte eingestellt.

   (Optional) Informationen zum Konfigurieren zusätzlicher Bucket-Einstellungen für Ihren spezifischen Anwendungsfall finden Sie unter Erstellen eines Buckets.

8. Wählen Sie Bucket erstellen aus.

Schritt 2: Hochladen einer Datei zu einem S3-Bucket

Laden Sie die Textdatei zu einem S3-Bucket hoch. Diese Textdatei enthält die Originaldaten, die Sie später in diesem Tutorial in Großbuchstaben umwandeln.

Sie können beispielsweise eine `tutorial.txt`-Datei, die den folgenden Text enthält, hochladen:

```
Amazon S3 Object Lambda Tutorial:
You can add your own code to process data retrieved from S3 before
returning it to an application.
```

Hochladen einer Datei in einen Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie in Schritt 1 (z. B. **`tutorial-bucket`**) erstellt haben, in den Sie die Datei hochladen möchten.
4. Wählen Sie im Tab Objects (Objekte) für Ihren Bucket die Option Upload (Hochladen) aus.
5. Wählen Sie auf der Seite Upload unter Dateien und Ordner die Option Dateien hinzufügen aus.
6. Wählen Sie eine hochzuladende Datei und dann Open (Öffnen) aus. Sie können z. B. das oben erwähnte `tutorial.txt`-Dateibeispiel hochladen.
7. Klicken Sie auf Hochladen.

Schritt 3: Erstellen eines S3-Zugriffspunkts

Um einen S3 Object Lambda Access Point für den Zugriff und die Umwandlung der ursprünglichen Daten zu verwenden, müssen Sie einen S3-Zugriffspunkt erstellen und dem S3-Bucket zuordnen, den Sie in Schritt 1 erstellt haben. Der Access Point muss sich in demselben Zustand befinden AWS-Region wie die Objekte, die Sie transformieren möchten.

Später in diesem Tutorial verwenden Sie diesen Zugriffspunkt als unterstützenden Zugriffspunkt für Ihren Object Lambda Access Point.

So erstellen Sie einen Zugangspunkt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Zugriffspunkte aus.
3. Wählen Sie auf der Seite Zugriffspunkte die Option Zugriffspunkt erstellen aus.
4. Geben Sie im Feld Name des Zugriffspunkts den gewünschten Namen für den Zugriffspunkt ein (z. B. **`tutorial-access-point`**).

   Weitere Hinweise zur Benennung von Zugangspunkten finden Sie unter Regeln zur Benennung von Amazon S3-Zugriffspunkten.
5. Wählen Sie im Feld Bucket-Name den Namen des Buckets aus, den Sie in Schritt 1 (z. B. **`tutorial-bucket`**) erstellt haben. S3 hängt den Zugriffspunkt an diesen Bucket an.

   (Optional) Sie können S3 durchsuchen auswählen, um die Buckets in Ihrem Konto zu browsen und zu durchsuchen. Wenn Sie Browse S3 (S3 durchsuchen) wählen, wählen Sie den gewünschten Bucket aus und dann Choose path (Pfad wählen), um das Feld Bucket name (Bucket-Name) mit dem Namen dieses Buckets zu füllen.
6. Wählen Sie für Netzwerkursprung Internetaus.

   Weitere Informationen zu Netzwerkursprüngen für Zugangspunkte finden Sie unter Erstellen von Zugriffspunkten, die auf eine Virtual Private Cloud beschränkt sind.
7. Alle Block-Public-Access-Einstellungen sind standardmäßig für Zugriffspunkte aktiviert. Sie sollten alle Einstellungen für das Blockieren aller öffentlichen Zugriffe aktiviert lassen.

   Weitere Informationen finden Sie unter Verwalten des öffentlichen Zugriffs auf Zugriffspunkte.
8. Behalten Sie für alle anderen Zugriffspunktseinstellungen die Standardeinstellungen bei.

(Optional) Sie können die Zugriffspunktseinstellungen ändern, um Ihren Anwendungsfall zu unterstützen. Für dieses Tutorial empfehlen wir, dass die Standardeinstellungen beibehalten werden.

(Optional) Wenn Sie den Zugriff auf Ihren Zugriffspunkt verwalten müssen, können Sie eine Zugriffspunkt-Richtlinie angeben. Weitere Informationen finden Sie unter Beispiele von - Zugriffspunktrichtlinien.

9. Wählen Sie Create access point (Zugriffspunkt erstellen) aus.

Schritt 4: Erstellen einer Lambda-Funktion

Erstellen Sie zum Transformieren der ursprünglichen Daten eine Lambda-Funktion zur Verwendung mit dem S3 Object Lambda Access Point.

Teilschritte

- Schreiben von Lambda-Funktionscode und Erstellen eines Bereitstellungspakets mit einer virtuellen Umgebung
- Erstellen einer Lambda-Funktion mit einer Ausführungsrolle (Konsole)
- Stellen Sie Ihren Lambda-Funktion-Code mit ZIP-Dateiarchiven bereit und konfigurieren Sie die Lambda-Funktion (Konsole)

Schreiben von Lambda-Funktionscode und Erstellen eines Bereitstellungspakets mit einer virtuellen Umgebung

1. Erstellen Sie auf dem lokalen Computer einen Ordner mit dem Ordnernamen `object-lambda` für die virtuelle Umgebung, die Sie zu einem späteren Zeitpunkt in diesem Tutorial verwenden können.
2. Erstellen Sie im `object-lambda`-Ordner eine Datei mit einer Lambda-Funktion, die den gesamten Text im ursprünglichen Objekt in Großbuchstaben ändert. Sie können z. B. die folgende Funktion verwenden, die in Python geschrieben wurde. Speichern Sie diese Funktion in einer Datei mit dem Name`transform.py`.

```
import boto3
import requests
from botocore.config import Config
```

```
# This function capitalizes all text in the original object
def lambda_handler(event, context):
    object_context = event["getObjectContext"]
    # Get the presigned URL to fetch the requested original object
    # from S3
    s3_url = object_context["inputS3Url"]
    # Extract the route and request token from the input context
    request_route = object_context["outputRoute"]
    request_token = object_context["outputToken"]

    # Get the original S3 object using the presigned URL
    response = requests.get(s3_url)
    original_object = response.content.decode("utf-8")

    # Transform all text in the original object to uppercase
    # You can replace it with your custom code based on your use case
    transformed_object = original_object.upper()

    # Write object back to S3 Object Lambda
    s3 = boto3.client('s3', config=Config(signature_version='s3v4'))
    # The WriteGetObjectResponse API sends the transformed data
    # back to S3 Object Lambda and then to the user
    s3.write_get_object_response(
        Body=transformed_object,
        RequestRoute=request_route,
        RequestToken=request_token)

    # Exit the Lambda function: return the status code
    return {'status_code': 200}
```

Note

Die vorangehende Lambda-Beispielfunktion lädt das gesamte angeforderte Objekt in den Speicher, bevor es transformiert und an den Client zurückgegeben wird. Alternativ können Sie das Objekt aus S3 streamen, um das Laden des gesamten Objekts in den Speicher zu vermeiden. Diese Methode ist nützlich, wenn Sie mit großen Objekten arbeiten. Weitere Informationen zum Streamen von Antworten mit Object Lambda Access Points finden Sie in den Streaming-Beispielen in Arbeiten mit `GetObject`-Anforderungen in Lambda.

Wenn Sie eine Lambda-Funktion für die Verwendung mit einem S3 Object Lambda Access Point schreiben, basiert die Funktion auf dem Eingabe-Ereigniskontext, den S3 Object Lambda für die Lambda-Funktion bereitstellt. Der Ereigniskontext stellt Informationen über die Anforderung bereit, die in dem von S3 Object Lambda an Lambda übergebenen Ereignis gestellt wird. Sie enthält die Parameter, die Sie verwenden, um die Lambda-Funktion zu erstellen.

Die Felder, die zum Erstellen der vorhergehenden Lambda-Funktion verwendet werden, lauten wie folgt:

Das Feld von `getObjectContext` bezeichnet die Eingabe- und Ausgabedetails für Verbindungen zu Amazon S3 und S3 Object Lambda. Es enthält die folgenden Felder:

- `inputS3Url`— Ein vorsigniertesURL, mit dem die Lambda-Funktion das Originalobjekt vom unterstützenden Access Point herunterladen kann. Durch die Verwendung eines vorsignierten URL Objekts benötigt die Lambda-Funktion keine Amazon S3 S3-Leseberechtigungen, um das ursprüngliche Objekt abzurufen, und sie kann nur auf das Objekt zugreifen, das bei jedem Aufruf verarbeitet wurde.
- `outputRoute`— Ein Routing-Token, das dem S3-Objekt-Lambda hinzugefügt wird, URL wenn die Lambda-Funktion aufruft`WriteGetObjectResponse`, um das transformierte Objekt zurückzusenden.
- `outputToken` – Ein Token, das von S3 Object Lambda verwendet wird, um den `WriteGetObjectResponse`-Aufruf mit dem ursprünglichen Anrufer abzugleichen, wenn das transformierte Objekt zurücksendet wird.

Weitere Informationen zu allen Feldern im Ereigniskontext finden Sie unter Format und Verwendung des Ereigniskontexts und Schreiben von Lambda-Funktionen für S3 Object Lambda Access Points.

3. Geben Sie in Ihrem lokalen Terminal den folgenden Befehl ein, um das `virtualenv`-Paket zu installieren:

```
python -m pip install virtualenv
```

4. Öffnen Sie in Ihrem lokalen Terminal den `object-lambda`-Ordner, den Sie zuvor erstellt haben und geben Sie dann den folgenden Befehl ein, um eine virtuelle Umgebung namens `venv` zu erstellen und zu initialisieren.

```
python -m virtualenv venv
```

5. Geben Sie zum Aktivieren der virtuellen Umgebung den folgenden Befehl ein, um die `activate`-Datei aus dem Ordner der Umgebung auszuführen:

   Führen Sie für macOS-Benutzer diesen Befehl aus:

   ```
   source venv/bin/activate
   ```

   Führen Sie für Windows-Benutzer diesen Befehl aus:

   ```
   .\venv\Scripts\activate
   ```

   Jetzt ändert sich die Eingabeaufforderung und zeigt (venv) an, was darauf hinweist, dass die virtuelle Umgebung aktiv ist.
6. Um die erforderlichen Bibliotheken zu installieren, führen Sie die folgenden Befehle Zeile für Zeile in der virtuellen `venv`-Umgebung aus.

   Mit diesen Befehlen werden aktualisierte Versionen der Abhängigkeiten Ihrer `lambda_handler`-Lambda -Funktion installiert. Bei diesen Abhängigkeiten handelt es sich AWS SDK um das Modul für Python (Boto3) und das Modul Anfragen.

   ```
   pip3 install boto3
   ```

   ```
   pip3 install requests
   ```
7. Sie können die virtuelle Umgebung deaktivieren, indem Sie den folgenden Befehl ausführen:

   ```
   deactivate
   ```
8. Um ein Bereitstellungspaket mit den installierten Bibliotheken als `.zip`-Datei mit dem Namen `lambda.zip` an der Wurzel des `object-lambda`-Verzeichnisses zu erstellen, führen Sie die folgenden Befehle Zeile für Zeile in Ihrem lokalen Terminal aus.

   > **Tip**
   >
   > Die folgenden Befehle müssen möglicherweise angepasst werden, um in Ihrer bestimmten Umgebung zu funktionieren. Eine Bibliothek kann beispielsweise in `site-`

`packages` oder `dist-packages` erscheinen, und der erste Ordner könnte `lib` oder `lib64` sein. Außerdem kann der Ordner `python` mit einer anderen Python-Version benannt werden. Sie können den Befehl `pip show` verwenden, um ein spezielles Paket zu finden.

Führen Sie für macOS-Benutzer diese Befehle aus:

```
cd venv/lib/python3.8/site-packages
```

```
zip -r ../../../../lambda.zip .
```

Führen Sie für Windows-Benutzer diese Befehle aus:

```
cd .\venv\Lib\site-packages\
```

```
powershell Compress-Archive * ../../../lambda.zip
```

Der letzte Befehl speichert das Bereitstellungspaket im Stamm des `object-lambda`-Verzeichnisses.

9. Fügen Sie dem Stamm Ihres Bereitstellungspakets die Funktionscode-Datei `transform.py` hinzu.

   Führen Sie für macOS-Benutzer diese Befehle aus:

   ```
   cd ../../../../
   ```

   ```
   zip -g lambda.zip transform.py
   ```

   Führen Sie für Windows-Benutzer diese Befehle aus:

   ```
   cd ..\..\..\
   ```

   ```
   powershell Compress-Archive -update transform.py lambda.zip
   ```

Nach diesem Schritt sollte Ihre Verzeichnisstruktur wie folgt aussehen:

```
lambda.zip$
  # transform.py
  # __pycache__
  | boto3/
  # certifi/
  # pip/
  # requests/
  ...
```

Erstellen einer Lambda-Funktion mit einer Ausführungsrolle (Konsole)

1. Melden Sie sich bei an AWS Management Console und öffnen Sie die AWS Lambda Konsole unter. https://console.aws.amazon.com/lambda/

2. Wählen Sie im linken Navigationsbereich die Option Functions (Funktionen) aus.
3. Wählen Sie Create function (Funktion erstellen).
4. Wählen Sie Author from scratch aus.
5. Führen Sie unter Basic information (Grundlegende Informationen) die folgenden Schritte aus:

   a. Geben Sie für Function name (Funktionsname) **`tutorial-object-lambda-function`** ein.

   b. Wählen Sie für Laufzeit Python 3.8 oder eine höhere Version aus.

6. Erweitern Sie den Abschnitt Ändern der standardmäßigen Ausführungsrolle. Wählen Sie in Execution role (Ausführungsrolle) die Option Create a new role with basic Lambda permissions (Neue Rolle mit Lambda-Basisberechtigungen erstellen) aus.

   In Schritt 5 weiter unten in diesem Tutorial fügen Sie AmazonS3 der Ausführungsrolle ObjectLambdaExecutionRolePolicy dieser Lambda-Funktion hinzu.

7. Lassen Sie die restlichen Einstellungen bei den Standardwerten.
8. Wählen Sie Funktion erstellen aus.

Stellen Sie Ihren Lambda-Funktion-Code mit ZIP-Dateiarchiven bereit und konfigurieren Sie die Lambda-Funktion (Konsole)

1. Wählen Sie in der AWS Lambda Konsole https://console.aws.amazon.com/lambda/unter im linken Navigationsbereich Funktionen aus.
2. Wählen Sie die Lambda-Funktion aus, die Sie zuvor erstellt haben (z. B **`tutorial-object-lambda-function`**).
3. Wählen Sie auf der Seite mit den Details der Lambda-Funktion den Code-Tab. Wählen Sie im Abschnitt Quellcode die Option Upload von und dann ZIP-Datei aus.
4. Wählen Sie Upload (Hochladen) aus, um Ihre lokale `.zip`-Datei auszuwählen.
5. Wählen Sie die zuvor erstellte `lambda.zip`-Datei und klicken Sie dann auf Öffnen.
6. Wählen Sie Save (Speichern) aus.
7. Wählen Sie im Abschnitt Laufzeiteinstellungen die OptionBearbeiten aus.
8. Bestätigen Sie auf der Seite Bearbeiten von Laufzeiteinstellungen, dass Laufzeit auf Python 3.8 oder eine höhere Version gesetzt wird.
9. Um der Lambda-Laufzeit mitzuteilen, welche Handler-Methode in Ihrem Lambda-Funktionscode aufgerufen werden soll, geben Sie **`transform.lambda_handler`** für Handler ein.

   Wenn Sie eine Funktion in Python konfigurieren, besteht der Wert der Handler-Einstellung aus dem Dateinamen und dem Namen des Handler-Moduls, getrennt durch einen Punkt. Beispielsweise ruft `transform.lambda_handler` die `lambda_handler`-Methode auf, die in der `transform.py`-Datei definiert ist.
10. Wählen Sie Save (Speichern) aus.
11. (Optional) Wählen Sie auf der Detailseite der Lambda-Funktion den Konfiguration-Tab. Wählen Sie im linken Navigationsbereich die OptionAllgemeine Konfiguration und anschließend Bearbeiten aus. Geben Sie im Timeout-Feld **`1`** m **`0`** s ein. Lassen Sie die restlichen Einstellungen auf die Standardwerte eingestellt und wählen Sie Speichern aus.

    Timeout ist die Zeitspanne, die Lambda einer Funktion für einen Aufruf zulässt, bevor diese gestoppt wird. Der Standardwert ist 3 Sekunden. Die maximale Dauer für eine Lambda-Funktion, die von S3 Object Lambda verwendet wird, beträgt 60 Sekunden. Die Preise basieren auf dem konfigurierten Arbeitsspeicher und der Zeit, für die der Code ausgeführt wird.

Schritt 5: Konfigurieren Sie eine IAM Richtlinie für die Ausführungsrolle Ihrer Lambda-Funktion

Damit Ihre Lambda-Funktion dem `GetObject` Aufrufer benutzerdefinierte Daten und Antwortheader bereitstellen kann, muss die Ausführungsrolle Ihrer Lambda-Funktion über IAM Berechtigungen zum Aufrufen von verfügen. `WriteGetObjectResponse` API

So fügen Sie Ihrer Lambda-Funktionsrolle eine IAM Richtlinie hinzu

1. Wählen Sie in der AWS Lambda Konsole https://console.aws.amazon.com/lambda/unter im linken Navigationsbereich Funktionen aus.
2. Wählen Sie die Funktion aus, die Sie in Schritt 4 erstellt haben (z. B. **`tutorial-object-lambda-function`**).
3. Wählen Sie auf der Detailseite Ihrer Lambda-Funktion den Tag Konfiguration und danach Berechtigungen im linken Navigationsbereich.
4. Wählen Sie unter Ausführungsrolle im Bereich den Link des Rollennamens aus. Die IAM Konsole wird geöffnet.
5. Wählen Sie auf der Übersichtsseite der IAM Konsole für die Ausführungsrolle Ihrer Lambda-Funktion den Tab Berechtigungen aus. Wählen Sie dann im Menü Add Permissions (Berechtigungen hinzufügen) die Option Attach policies (Richtlinien anfügen) aus.
6. Geben Sie auf der Seite Attach permissions (Berechtigungen anfügen) **`AmazonS3ObjectLambdaExecutionRolePolicy`** in das Suchfeld ein, um die Liste der Richtlinien zu filtern. Aktivieren Sie das Kontrollkästchen neben dem Namen der ObjectLambdaExecutionRolePolicyAmazonS3-Richtlinie.
7. Wählen Sie Richtlinien anfügen.

Schritt 6: Erstellen eines S3 Object Lambda Access Point

Ein S3 Object Lambda Access Point bietet die Flexibilität, eine Lambda-Funktion direkt von einer GET S3-Anfrage aus aufzurufen, sodass die Funktion Daten verarbeiten kann, die von einem S3-Zugangspunkt abgerufen wurden. Beim Erstellen und Konfigurieren eines S3 Object Lambda Access Points müssen Sie die Lambda-Funktion angeben, die aufgerufen werden soll, und den Ereigniskontext im JSON Format als benutzerdefinierte Parameter für Lambda zur Verfügung stellen.

So erstellen Sie einen S3 Object Lambda Access Point

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

2. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
3. Wählen Sie auf der Seite Object Lambda access points (Objekt-Lambda-Zugriffspunkte) die Option Create Object Lambda access point (Objekt-Lambda-Zugriffspunkt erstellen).
4. Geben Sie unter Name des Objekt-Lambda-Access Point den Namen ein, den Sie für den Object Lambda Access Point verwenden möchten (z. B. **`tutorial-object-lambda-accesspoint`**).
5. Geben Sie für Unterstützenden Zugangspunkt den Standardzugangspunkt ein, den Sie in Schritt 3 erstellt haben (z. B. **`tutorial-access-point`**), oder navigieren Sie zu diesem Zugangspunkt und wählen dann Unterstützenden Zugangspunkt auswählen aus.
6. Um für S3 Objekte aus dem S3-Bucket abzurufenAPIs, damit die Lambda-Funktion sie verarbeiten kann, wählen Sie GetObject.
7. Für die Funktion Lambda aufrufen können Sie eine der beiden folgenden Optionen für dieses Tutorial auswählen.
   - Wählen Sie Wählen Sie aus Funktionen in Ihrem Kontound wählen Sie die Lambda-Funktion aus, die Sie in Schritt 4 (z. B. **`tutorial-object-lambda-function`**) aus der Lambda-Funktion-Dropdown-Liste erstellt haben.
   - Wählen Sie Enter ARN und geben Sie dann den Amazon-Ressourcennamen (ARN) der Lambda-Funktion ein, die Sie in Schritt 4 erstellt haben.
8. Wählen Sie für Lambda-Funktionsversion $ LATEST (die neueste Version der Lambda-Funktion, die Sie in Schritt 4 erstellt haben).
9. (Optional) Wenn Sie Ihre Lambda-Funktion benötigen, um GET Anfragen mit Bereichs- und Artikelnummernüberschriften zu erkennen und zu verarbeiten, wählen Sie Lambda-Funktion unterstützt Anfragen mit Bereich und Lambda-Funktion unterstützt Anfragen mit Artikelnummern aus. Deaktivieren Sie andernfalls diese beiden Kontrollkästchen.

   Weitere Informationen zur Verwendung von Bereichs- oder Teilenummern mit S3 Object Lambda finden Sie unter Arbeitet mit Range and partNumber Header.
10. (Optional) Fügen Sie unter Payload — optional JSON Text hinzu, um Ihre Lambda-Funktion mit zusätzlichen Informationen zu versorgen.

    Eine Payload ist optionaler JSON Text, den Sie Ihrer Lambda-Funktion als Eingabe für alle Aufrufe bereitstellen können, die von einem bestimmten S3 Object Lambda Access Point kommen. Um das Verhalten für mehrere Object Lambda Access Points anzupassen, die dieselbe Lambda-Funktion aufrufen, können Sie Nutzlasten mit unterschiedlichen Parametern konfigurieren und so die Flexibilität Ihrer Lambda-Funktion erweitern.

Weitere Informationen zur Nutzlast finden Sie unter Format und Verwendung des Ereigniskontexts.

11. (Optional) Wählen Sie für Anforderungsmetriken – optional die Option Deaktivieren oder Aktivieren aus, um Ihrem Object Lambda Access Point Amazon-S3-Überwachung hinzuzufügen. Anforderungsmetriken werden zum CloudWatch Standardtarif von Amazon abgerechnet. Weitere Informationen finden Sie unter CloudWatchPreise.
12. Behalten Sie unter Objekt-Lambda-Zugriffspunkt-Richtlinie - optional die Standardeinstellung bei.

    (Optional) Sie können eine Ressourcenrichtlinie festlegen. Diese Ressourcenrichtlinie gewährt die `GetObject` API Erlaubnis, den angegebenen Object Lambda Access Point zu verwenden.
13. Lassen Sie die restlichen Einstellungen auf die Standardwerte eingestellt und klicken Sie auf Erstellen eines Objekt-Lambda-Zugriffspunkts.

Schritt 7: Anzeigen der transformierten Daten

Jetzt ist S3 Object Lambda bereit, Ihre Daten für Ihren Anwendungsfall zu transformieren. In diesem Tutorial wandelt S3 Object Lambda den gesamten Text in Ihrem Objekt in Großbuchstaben um.

Teilschritte

- Zeigen Sie die transformierten Daten in Ihrem S3 Object Lambda Access Point an
- Ausführen eines Python-Skripts zum Drucken der ursprünglichen und transformierten Daten

Zeigen Sie die transformierten Daten in Ihrem S3 Object Lambda Access Point an

Wenn Sie den Abruf einer Datei über Ihren S3 Object Lambda Access Point anfordern, `GetObject` API rufen Sie S3 Object Lambda auf. S3 Object Lambda ruft die Lambda-Funktion auf, um Ihre Daten zu transformieren, und gibt dann die transformierten Daten als Antwort auf den Standard-S3-Aufruf zurück. `GetObject` API

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
3. Wählen Sie auf der Seite Objekt-Lambda-Zugriffspunkte den S3 Object Lambda Access Point aus, den Sie in Schritt 6 erstellt haben (z. B. **`tutorial-object-lambda-accesspoint`**).

4. Wählen Sie auf der Registerkarte Objekte Ihres S3 Object Lambda Access Point die Datei mit dem gleichen Namen (z. B. `tutorial.txt`) wie diejenige aus, die Sie in Schritt 2 in den S3-Bucket hochgeladen haben.

   Diese Datei sollte alle transformierten Daten enthalten.
5. Um die transformierten Daten anzuzeigen, wählen Sie Öffnen oder Herunterladen aus.

Ausführen eines Python-Skripts zum Drucken der ursprünglichen und transformierten Daten

Sie können S3 Object Lambda mit Ihren vorhandenen Anwendungen verwenden. Aktualisieren Sie dazu Ihre Anwendungskonfiguration, sodass der neue S3 Object Lambda Access PointARN, den Sie in Schritt 6 erstellt haben, zum Abrufen von Daten aus S3 verwendet wird.

Das folgende Python-Beispielskript druckt sowohl die ursprünglichen Daten aus dem S3-Bucket als auch die transformierten Daten aus dem S3 Object Lambda Access Point.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
3. Wählen Sie auf der Seite Objekt-Lambda-Zugriffspunkte die Optionsschaltfläche links neben dem S3 Object Lambda Access Point aus, den Sie in Schritt 6 erstellt haben (z. B. **`tutorial-object-lambda-accesspoint`**).
4. Wählen Sie die Option CopyARN Kopieren aus.
5. Speichern Sie das ARN für eine spätere Verwendung.
6. Schreiben Sie ein Python-Skript auf Ihrem lokalen Computer, um sowohl die ursprünglichen Daten aus Ihrem S3-Bucket (z. B. `tutorial.txt`) als auch die transformierten Daten aus Ihrem S3 Object Lambda Access Point (z. B. `tutorial.txt`) zu drucken. Sie können folgendes Beispiel-Skript nutzen.

```
import boto3
from botocore.config import Config

s3 = boto3.client('s3', config=Config(signature_version='s3v4'))

def getObject(bucket, key):
    objectBody = s3.get_object(Bucket = bucket, Key = key)
    print(objectBody["Body"].read().decode("utf-8"))
    print("\n")
```

```
print('Original object from the S3 bucket:')
# Replace the two input parameters of getObject() below with
# the S3 bucket name that you created in Step 1 and
# the name of the file that you uploaded to the S3 bucket in Step 2
getObject("tutorial-bucket",
          "tutorial.txt")

print('Object transformed by S3 Object Lambda:')
# Replace the two input parameters of getObject() below with
# the ARN of your S3 Object Lambda Access Point that you saved earlier and
# the name of the file with the transformed data (which in this case is
# the same as the name of the file that you uploaded to the S3 bucket
# in Step 2)
getObject("arn:aws:s3-object-lambda:us-west-2:111122223333:accesspoint/tutorial-
object-lambda-accesspoint",
          "tutorial.txt")
```

7. Speichern Sie Ihr Python-Skript unter einem benutzerdefinierten Namen (z. B. `tutorial_print.py`) im Ordner (z. B. `object-lambda`), den Sie in Schritt 4 auf Ihrem lokalen Computer erstellt haben.

8. Führen Sie in Ihrem lokalen Terminal den folgenden Befehl vom Stamm des Verzeichnisses aus (z. B. `object-lambda`), das Sie inSchritt 4 erstellt haben.

   ```
   python3 tutorial_print.py
   ```

   Sie sollten sowohl die ursprünglichen Daten als auch die transformierten Daten (alle Texte in Großbuchstaben) über das Terminal sehen. z. B. sollten Sie etwas wie den folgenden Text sehen.

   ```
   Original object from the S3 bucket:
   Amazon S3 Object Lambda Tutorial:
   You can add your own code to process data retrieved from S3 before
   returning it to an application.

   Object transformed by S3 Object Lambda:
   AMAZON S3 OBJECT LAMBDA TUTORIAL:
   YOU CAN ADD YOUR OWN CODE TO PROCESS DATA RETRIEVED FROM S3 BEFORE
   RETURNING IT TO AN APPLICATION.
   ```

Schritt 8: Bereinigen

Wenn Sie Ihre Daten über S3 Object Lambda nur als Tutorial transformiert haben, löschen Sie die AWS -Ressourcen, die Sie zugewiesen haben, damit keine Gebühren mehr anfallen.

Teilschritte

- Löschen des Object Lambda Access Point
- Löschen des S3-Zugriffspunkts
- Löschen der Ausführungsrolle für Ihre Lambda-Funktion
- Löschen Sie die Lambda-Funktion
- Löschen Sie die CloudWatch Protokollgruppe
- Löschen Sie die Originaldatei im S3-Quell-Bucket
- Löschen des S3-Quell-Bucket
- Löschen Sie den IAM Benutzer

Löschen des Object Lambda Access Point

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
3. Wählen Sie auf der Seite Objekt-Lambda-Zugriffspunkte die Optionsschaltfläche links neben dem S3 Object Lambda Access Point aus, den Sie in Schritt 6 erstellt haben (z. B. **`tutorial-object-lambda-accesspoint`**).
4. Wählen Sie Delete (Löschen).
5. Bestätigen Sie, dass Sie Ihren Object Lambda Access Point löschen möchten, indem Sie den Namen in das angezeigte Textfeld eingeben und dann Löschen wählen.

Löschen des S3-Zugriffspunkts

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Zugriffspunkte aus.
3. Navigieren Sie zu dem Zugriffspunkt, den Sie in Schritt 3 (z. B. **`tutorial-access-point`**) erstellt haben, und wählen Sie die Schaltfläche neben dem Namen des Zugriffspunkts aus.

4. Wählen Sie Löschen aus.
5. Bestätigen Sie, dass Sie Ihren Zugriffspunkt löschen möchten, indem Sie den Namen in das angezeigte Textfeld eingeben und dann Delete (Löschen) wählen.

Löschen der Ausführungsrolle für Ihre Lambda-Funktion

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.
2. Wählen Sie im linken Navigationsbereich die Option Funktionen aus.
3. Wählen Sie die Funktion aus, die Sie in Schritt 4 erstellt haben (z. B. **`tutorial-object-lambda-function`**).
4. Wählen Sie auf der Detailseite Ihrer Lambda-Funktion den Tag Konfiguration und danach Berechtigungen im linken Navigationsbereich.
5. Wählen Sie unter Ausführungsrolle im Bereich den Link des Rollennamens aus. Die IAM Konsole wird geöffnet.
6. Wählen Sie auf der Übersichtsseite der IAM Konsole mit der Ausführungsrolle Ihrer Lambda-Funktion die Option Rolle löschen aus.
7. Klicken Sie im Dialogfeld auf Rolle löschen und anschließend auf Ja, löschen.

Löschen Sie die Lambda-Funktion

1. Wählen Sie in der AWS Lambda Konsole https://console.aws.amazon.com/lambda/unter im linken Navigationsbereich Funktionen aus.
2. Aktivieren Sie das Kontrollkästchen links neben dem Namen der Funktion, die Sie im Abschnitt Schritt 4 (z. B. **`tutorial-object-lambda-function`**) erstellt haben.
3. Wählen Sie Actions (Aktionen) und anschließend Delete (Löschen).
4. Wählen Sie im Bestätigungsdialogfeld Delete (Löschen) die Option Delete (Löschen) aus.

Löschen Sie die CloudWatch Protokollgruppe

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im linken Navigationsbereich Log groups (Protokollgruppen) aus.
3. Suchen Sie die Protokollgruppe, deren Name mit der Lambda-Funktion endet, die Sie in Schritt 4 (z. B. **`tutorial-object-lambda-function`**) erstellt haben.

4. Aktivieren Sie das Kontrollkästchen links neben dem Namen der Protokollgruppe.
5. Wählen Sie Actions (Aktionen) und dann Delete log group(s) (Protokollgruppe(n) löschen) aus.
6. Wählen Sie im Dialogfeld Delete log group(s) (Protokollgruppe(n) löschen) die Option Delete (Löschen) aus.

Löschen Sie die Originaldatei im S3-Quell-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie in der Liste Bucket-Name den Namen des Buckets aus, in den Sie die Originaldatei in Schritt 2 (z. B. **`tutorial-bucket`**) hochgeladen haben.
4. Markieren Sie das Kontrollkästchen links neben dem Namen des Objekts, das Sie löschen möchten (z. B. `tutorial.txt`).
5. Wählen Sie Löschen.
6. Bestätigen Sie auf der Seite Löschen von Objekten im Abschnitt Objekte endgültig löschen?, dass Sie dieses Objekt löschen möchten, indem Sie **`permanently delete`** im Textfeld eingeben.
7. Wählen Sie Delete objects (Objekte löschen).

Löschen des S3-Quell-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie in der Liste Buckets die Optionsschaltfläche neben dem Namen des Buckets aus, den Sie in Schritt 1 (z. B. **`tutorial-bucket`**) erstellt haben.
4. Wählen Sie Löschen.
5. Bestätigen Sie auf der Seite Delete bucket (Bucket löschen), dass Sie den Bucket löschen möchten. Geben Sie dazu den Bucket-Namen in das Textfeld ein und wählen Sie Delete bucket (Bucket löschen).

Löschen Sie den IAM Benutzer

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Benutzer, und aktivieren Sie dann das Kontrollkästchen neben dem Benutzer, den Sie löschen möchten.
3. Wählen Sie oben auf der Seite Delete (Löschen) aus.
4. Im Bereich Löschen ***`user name`***? Geben Sie im Dialogfeld den Benutzernamen in das Texteingabefeld ein, um das Löschen des Benutzers zu bestätigen. Wählen Sie Löschen.

Nächste Schritte

Nach Abschluss dieses Tutorials können Sie die Lambda-Funktion an Ihren Anwendungsfall anpassen, um die von GET Standard-S3-Anfragen zurückgegebenen Daten zu ändern.

Im Folgenden finden Sie eine Liste der häufigsten Anwendungsfälle für S3 Object Lambda:

- Maskierung sensibler Daten für Sicherheit und Compliance.

  Weitere Informationen finden Sie unter Tutorial: Erkennen und Redigieren von PII Daten mit S3 Object Lambda und Amazon Comprehend.
- Filtern bestimmter Datenzeilen, um bestimmte Informationen bereitzustellen.
- Erweitern von Daten mit Informationen aus anderen Diensten oder Datenbanken.
- Konvertierung zwischen Datenformaten, z. B. Konvertierung in XML aus JSON Gründen der Anwendungskompatibilität.
- Komprimieren oder Dekomprimieren von Dateien, während sie heruntergeladen werden.
- Größenänderung und Wasserzeichen für Bilder.

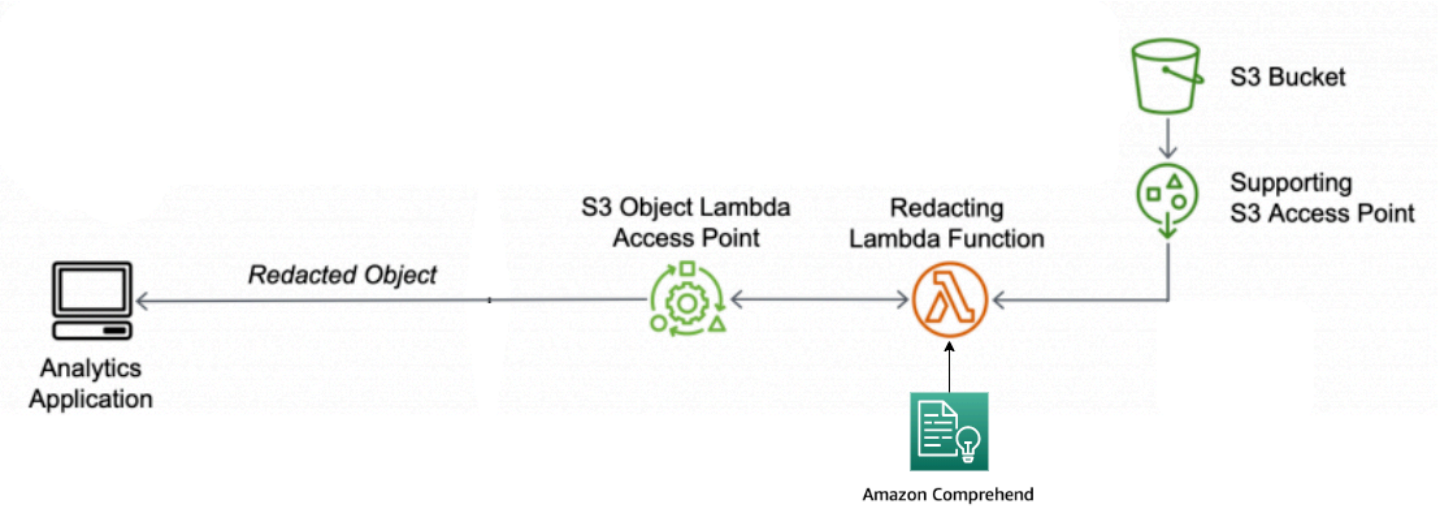
  Weitere Informationen finden Sie unter Tutorial: Verwenden von S3 Object Lambda, um Bilder beim Abrufen dynamisch mit Wasserzeichen zu versehen.
- Implementieren benutzerdefinierter Autorisierungsregeln für den Zugriff auf Daten.

Weitere Informationen zu S3 Object Lambda finden Sie unter Transformieren von Objekten mit S3 Object Lambda.

Tutorial: Erkennen und Redigieren von PII Daten mit S3 Object Lambda und Amazon Comprehend

Wenn Sie Amazon S3 für gemeinsam genutzte Datensätze verwenden, auf die mehrere Anwendungen und Benutzer zugreifen können, ist es wichtig, dass vertrauliche Informationen, wie z. B. personenbezogene Daten (PII), nur autorisierten Personen zur Verfügung stehen. Wenn eine Marketinganwendung beispielsweise Daten verwendet, die Daten enthaltenPII, muss sie möglicherweise zuerst PII Daten maskieren, um die Datenschutzanforderungen zu erfüllen. Wenn eine Analytikanwendung ein Bestandsdatensatz für Produktionsaufträge verwendet, muss sie möglicherweise zuerst die Kreditkarteninformationen des Kunden überarbeiten, um unbeabsichtigte Datenverluste zu verhindern.

Mit S3 Object Lambda und einer vorgefertigten AWS Lambda Funktion, die von Amazon Comprehend unterstützt wird, können Sie aus S3 abgerufene PII Daten schützen, bevor Sie sie an eine Anwendung zurückgeben. Insbesondere können Sie die vorgefertigte Lambda-Funktion als Redigierungsfunktion nutzen und sie an einen S3 Object Lambda Access Point anfügen. Wenn eine Anwendung (z. B. eine Analyseanwendung) standardmäßige GET S3-Anfragen sendet, rufen diese über den S3 Object Lambda Access Point gestellten Anfragen die vordefinierte Lambda-Funktion zum Schwärzen auf, um PII Daten zu erkennen und zu redigieren, die aus einem S3-Bucket über einen unterstützenden S3-Zugriffspunkt abgerufen wurden. Dann gibt der S3 Object Lambda Access Point das redigierte Ergebnis zurück an die Anwendung aus.



Dabei verwendet die vordefinierte Lambda-Funktion Amazon Comprehend, einen Service zur Verarbeitung natürlicher Sprache (NLP), um Variationen in der Darstellung zu erfassen, unabhängig davon, wie sie im Text PII vorhanden PII ist (z. B. numerisch oder als Kombination aus Wörtern und Zahlen). Amazon Comprehend kann sogar den Kontext im Text verwenden, um

zu verstehen, ob es sich bei einer 4-stelligen Zahl um einePIN, um die letzten vier Zahlen einer Sozialversicherungsnummer (SSN) oder um ein Jahr handelt. Amazon Comprehend verarbeitet jede Textdatei im UTF -8-Format und kann in großem Umfang schützen, ohne die PII Genauigkeit zu beeinträchtigen. Weitere Informationen finden Sie unter Was ist Amazon Comprehend? im Entwicklerhandbuch für Amazon Comprehend.

Ziel

In diesem Tutorial erfahren Sie, wie Sie S3 Object Lambda mit der vordefinierten Lambda-Funktion `ComprehendPiiRedactionS3ObjectLambda` verwenden. Diese Funktion verwendet Amazon Comprehend, um Entitäten zu erkennenPII. Diese Entitäten werden dann durch Sternchen ersetzt. Durch das PII Redigieren verbergen Sie sensible Daten, was zu Sicherheit und Compliance beitragen kann.

Sie lernen auch, wie Sie eine vorgefertigte AWS Lambda Funktion verwenden und konfigurieren, AWS Serverless Application Repositoryum mit S3 Object Lambda für eine einfache Bereitstellung zusammenzuarbeiten.

Themen

- Voraussetzungen: Erstellen Sie einen IAM Benutzer mit Berechtigungen
- Schritt 1: Einen S3-Bucket erstellen
- Schritt 2: Hochladen einer Datei zu einem S3-Bucket
- Schritt 3: Erstellen eines S3-Zugriffspunkts
- Schritt 4: Konfigurieren und Bereitstellen einer vordefinierten Lambda-Funktion
- Schritt 5: Erstellen eines S3 Object Lambda Access Point
- Schritt 6: Verwenden des S3 Object Lambda Access Point zum Abrufen der redigierten Daten
- Schritt 7: Bereinigen
- Nächste Schritte

Voraussetzungen: Erstellen Sie einen IAM Benutzer mit Berechtigungen

Bevor Sie mit diesem Tutorial beginnen, benötigen Sie ein AWS Konto, mit dem Sie sich als AWS Identity and Access Management Benutzer (IAMBenutzer) mit den richtigen Berechtigungen anmelden können.

Sie können einen IAM Benutzer für das Tutorial erstellen. Um dieses Tutorial abzuschließen, muss Ihr IAM Benutzer die folgenden IAM Richtlinien anhängen, um auf relevante AWS Ressourcen zuzugreifen und bestimmte Aktionen auszuführen.

> **Note**
>
> Der Einfachheit halber wird in diesem Tutorial ein IAM Benutzer erstellt und verwendet. Denken Sie nach Abschluss dieses Tutorials an Löschen Sie den IAM Benutzer. Für den produktiven Einsatz empfehlen wir, dass Sie die bewährten Sicherheitsmethoden IAM im IAM Benutzerhandbuch befolgen. Eine bewährte Methode ist, dass menschliche Benutzer den Verbund mit einem Identitätsanbieter verwenden, um mit temporären Anmeldeinformationen auf AWS zuzugreifen. Eine weitere bewährte Methode besteht darin, Workloads die Verwendung temporärer Anmeldeinformationen mit IAM Rollen für den Zugriff AWS vorzuschreiben. Informationen AWS IAM Identity Center zur Erstellung von Benutzern mit temporären Anmeldeinformationen finden Sie unter Erste Schritte im AWS IAM Identity Center Benutzerhandbuch.
>
> In diesem Tutorial werden auch Richtlinien mit vollem Zugriff verwendet. Für die Verwendung in der Produktion empfehlen wir, dass Sie stattdessen nur die für Ihren Anwendungsfall erforderlichen Mindestberechtigungen gemäß Bewährte Methoden in Bezug auf die Sicherheit erteilen.

Ihr IAM Benutzer benötigt die folgenden AWS verwalteten Richtlinien:

- AmazonS3 FullAccess — Erteilt Berechtigungen für alle Amazon S3-Aktionen, einschließlich der Berechtigungen zum Erstellen und Verwenden eines Object Lambda Access Points.
- AWSLambda_ FullAccess — Erteilt Berechtigungen für alle Lambda-Aktionen.
- AWSCloudFormationFullAccess— Erteilt Berechtigungen für alle AWS CloudFormation Aktionen.
- IAMFullAccess— Erteilt Berechtigungen für alle IAM Aktionen.
- IAMAccessAnalyzerReadOnlyAccess— Erteilt Berechtigungen zum Lesen aller von Access Analyzer bereitgestellten IAM Zugriffsinformationen.

Sie können diese vorhandenen Richtlinien direkt anhängen, wenn Sie einen IAM Benutzer erstellen. Weitere Informationen zum Erstellen eines IAM Benutzers finden Sie im Benutzerhandbuch unter IAMBenutzer erstellen (Konsole). IAM

Darüber hinaus benötigt Ihr IAM Benutzer eine vom Kunden verwaltete Richtlinie. Um dem IAM Benutzer Berechtigungen für alle AWS Serverless Application Repository Ressourcen und Aktionen zu gewähren, müssen Sie eine IAM Richtlinie erstellen und die Richtlinie an den IAM Benutzer anhängen.

Um eine IAM Richtlinie zu erstellen und an Ihren IAM Benutzer anzuhängen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich die Option Policies (Richtlinien) aus.
3. Wählen Sie Richtlinie erstellen aus.
4. Wählen Sie im Tab Visual editor (Visueller Editor) für Service Choose a service (Wählen Sie einen Service) aus. Wählen Sie dann Serverless Application Repository.
5. Wählen Sie für Aktionen unter Manuelle Aktionen Alle Aktionen des Serverless Application Repository (serverlessrepo: *) für dieses Tutorial.

   Als bewährte Methode für die Sicherheit sollten Sie basierend auf Ihrem Anwendungsfall nur für jene Aktionen und Ressourcen Berechtigungen zulassen, für die ein Benutzer Zugriff benötigt. Weitere Informationen finden Sie IAMim IAMBenutzerhandbuch unter Bewährte Sicherheitsmethoden.
6. Wählen Sie für Ressourcen Alle Ressourcen für dieses Tutorial.

   Als bewährte Methode sollten Sie Berechtigungen nur für bestimmte Ressourcen in bestimmten Konten definieren. Alternativ können Sie die geringste Berechtigung mit Bedingungsschlüsseln erteilen. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Geringste Zugriffsrechte gewähren.
7. Wählen Sie Next: Tags (Weiter: Tags) aus.
8. Klicken Sie auf Weiter: Prüfen.
9. Geben Sie auf der Seite Review policy (Richtlinie überprüfen) einen Namen (z. B. **`tutorial-serverless-application-repository`**) und eine Beschreibung (optional) für die Richtlinie ein, die Sie erstellen. Überprüfen Sie die Richtlinienzusammenfassung, um sicherzustellen, dass Sie die beabsichtigten Berechtigungen erteilt haben, und wählen Sie dann Richtlinie erstellen aus, um Ihre neue Richtlinie zu speichern.
10. Wählen Sie im linken Navigationsbereich Benutzer aus. Wählen Sie dann den IAM Benutzer für dieses Tutorial aus.

11. Wählen Sie auf der Seite Übersicht des ausgewählten Benutzers den Tab Berechtigungen und danach Hinzufügen von Berechtigungen.
12. Wählen Sie unter Berechtigungen gewähren die Option Vorhandene Richtlinien direkt anfügen aus.
13. Aktivieren Sie das Kontrollkästchen neben der gerade erstellten Richtlinie (z. B. **`tutorial-serverless-application-repository`**) und wählen Sie dann Nächstes: Überprüfung.
14. Prüfen Sie unter Richtlinienübersicht die Zusammenfassung, um sicherzustellen, dass Sie die beabsichtigte Richtlinie angefügt haben. Wählen Sie dann Add permissions (Berechtigungen hinzufügen) aus.

Schritt 1: Einen S3-Bucket erstellen

Erstellen Sie einen Bucket zum Speichern der ursprünglichen Daten, die Sie transformieren möchten.

So erstellen Sie einen Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie Create Bucket (Bucket erstellen) aus.

   Die Seite Create bucket (Bucket erstellen) wird geöffnet.
4. Geben Sie für Bucket-Name einen Namen für Ihren Bucket ein (z. B. **`tutorial-bucket`**).

   Weitere Informationen zur Benennung von Buckets in Amazon S3 finden Sie unter Allgemeine Benennungsregeln für Buckets.
5. Unter Region wählen Sie die AWS-Region aus, in der sich der Bucket befinden soll.

   Weitere Informationen zur Bucket-Region finden Sie unter Bucket-Übersicht.
6. Belassen Sie die Einstellungen für den öffentlichen Zugriff für diesen Bucket blockieren bei den Standardeinstellungen (Alle öffentlichen Zugriffe blockieren ist aktiviert).

   Es wird empfohlen, alle Einstellungen für öffentlichen Zugriff blockieren aktiviert zu belassen, es sei denn, Sie müssen eine oder mehrere dieser Einstellungen für Ihren Anwendungsfall deaktivieren. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.
7. Lassen Sie die restlichen Einstellungen auf die Standardwerte eingestellt.

(Optional) Informationen zum Konfigurieren zusätzlicher Bucket-Einstellungen für Ihren spezifischen Anwendungsfall finden Sie unter Erstellen eines Buckets.

8. Wählen Sie Bucket erstellen aus.

Schritt 2: Hochladen einer Datei zu einem S3-Bucket

Laden Sie eine Textdatei mit bekannten PII Daten verschiedener Typen, wie Namen, Bankinformationen, Telefonnummern usw., als Originaldaten in den S3-Bucket hoch, die Sie später in diesem Tutorial PII redigieren werden. SSNs

Sie können beispielsweise nach der `tutorial.txt`-Datei hochladen. Dies ist ein Beispiel für eine Eingabedatei von Amazon Comprehend.

```
Hello Zhang Wei, I am John. Your AnyCompany Financial Services,
LLC credit card account 1111-0000-1111-0008 has a minimum payment
of $24.53 that is due by July 31st. Based on your autopay settings,
we will withdraw your payment on the due date from your
bank account number XXXXXX1111 with the routing number XXXXX0000.

Your latest statement was mailed to 100 Main Street, Any City,
WA 98121.
After your payment is received, you will receive a confirmation
text message at 206-555-0100.
If you have questions about your bill, AnyCompany Customer Service
is available by phone at 206-555-0199 or
email at support@anycompany.com.
```

Hochladen einer Datei in einen Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie in Schritt 1 (z. B. **`tutorial-bucket`**) erstellt haben, in den Sie die Datei hochladen möchten.
4. Wählen Sie im Tab Objects (Objekte) für Ihren Bucket die Option Upload (Hochladen) aus.
5. Wählen Sie auf der Seite Upload unter Dateien und Ordner die Option Dateien hinzufügen aus.
6. Wählen Sie eine hochzuladende Datei und dann Open (Öffnen) aus. Sie können z. B. das oben erwähnte `tutorial.txt`-Dateibeispiel hochladen.

7. Klicken Sie auf Hochladen.

Schritt 3: Erstellen eines S3-Zugriffspunkts

Um einen S3 Object Lambda Access Point für den Zugriff und die Umwandlung der ursprünglichen Daten zu verwenden, müssen Sie einen S3-Zugriffspunkt erstellen und dem S3-Bucket zuordnen, den Sie in Schritt 1 erstellt haben. Der Access Point muss sich in demselben Zustand befinden AWS-Region wie die Objekte, die Sie transformieren möchten.

Später in diesem Tutorial verwenden Sie diesen Zugriffspunkt als unterstützenden Zugriffspunkt für Ihren Object Lambda Access Point.

So erstellen Sie einen Zugangspunkt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Zugriffspunkte aus.
3. Wählen Sie auf der Seite Zugriffspunkte die Option Zugriffspunkt erstellen aus.
4. Geben Sie im Feld Name des Zugriffspunkts den gewünschten Namen für den Zugriffspunkt ein (z. B. **`tutorial-pii-access-point`**).

   Weitere Hinweise zur Benennung von Zugangspunkten finden Sie unter Regeln zur Benennung von Amazon S3-Zugriffspunkten.
5. Wählen Sie im Feld Bucket-Name den Namen des Buckets aus, den Sie in Schritt 1 (z. B. **`tutorial-bucket`**) erstellt haben. S3 hängt den Zugriffspunkt an diesen Bucket an.

   (Optional) Sie können S3 durchsuchen auswählen, um die Buckets in Ihrem Konto zu browsen und zu durchsuchen. Wenn Sie Browse S3 (S3 durchsuchen) wählen, wählen Sie den gewünschten Bucket aus und dann Choose path (Pfad wählen), um das Feld Bucket name (Bucket-Name) mit dem Namen dieses Buckets zu füllen.
6. Wählen Sie für Netzwerkursprung Internetaus.

   Weitere Informationen zu Netzwerkursprüngen für Zugangspunkte finden Sie unter Erstellen von Zugriffspunkten, die auf eine Virtual Private Cloud beschränkt sind.
7. Alle Block-Public-Access-Einstellungen sind standardmäßig für Zugriffspunkte aktiviert. Sie sollten alle Einstellungen für das Blockieren aller öffentlichen Zugriffe aktiviert lassen. Weitere Informationen finden Sie unter Verwalten des öffentlichen Zugriffs auf Zugriffspunkte.

8. Behalten Sie für alle anderen Zugriffspunktseinstellungen die Standardeinstellungen bei.

   (Optional) Sie können die Zugriffspunktseinstellungen ändern, um Ihren Anwendungsfall zu unterstützen. Für dieses Tutorial empfehlen wir, dass die Standardeinstellungen beibehalten werden.

   (Optional) Wenn Sie den Zugriff auf Ihren Zugriffspunkt verwalten müssen, können Sie eine Zugriffspunkt-Richtlinie angeben. Weitere Informationen finden Sie unter Beispiele von - Zugriffspunktrichtlinien.
9. Wählen Sie Zugangspunkt erstellen aus.

Schritt 4: Konfigurieren und Bereitstellen einer vordefinierten Lambda-Funktion

Um PII Daten zu redigieren, konfigurieren und implementieren Sie die vorgefertigte AWS Lambda Funktion `ComprehendPiiRedactionS3ObjectLambda` für die Verwendung mit Ihrem S3 Object Lambda Access Point.

Konfigurieren und Bereitstellen der Lambda-Funktion

1. Melden Sie sich bei an AWS Management Console und sehen Sie sich die `ComprehendPiiRedactionS3ObjectLambda`Funktion in der an. AWS Serverless Application Repository
2. Behalten Sie für Anwendungseinstellungen unter Anwendungsname den Standardwert (`ComprehendPiiRedactionS3ObjectLambda`) für dieses Tutorial bei.

   (Optional) Sie können den Namen eingeben, den Sie dieser Anwendung geben möchten. Sie können dies tun, wenn Sie planen, mehrere Lambda-Funktionen für unterschiedliche Zugriffsanforderungen für denselben freigegebenen Datensatz zu konfigurieren.
3. Behalten Sie für MaskCharacterden Standardwert (*****) bei. Das Maskenzeichen ersetzt jedes Zeichen in der geschwärzten PII Entität.
4. Behalten Sie für MaskModeden Standardwert (MASK) bei. Der MaskModeWert gibt an, ob die PII Entität mit dem `MASK` Zeichen oder dem `PII_ENTITY_TYPE` Wert geschwärzt wird.
5. Um die angegebenen Datentypen zu schwärzen PiiEntityTypes, behalten Sie für den Standardwert bei. ALL Der PiiEntityTypesWert gibt die PII Entitätstypen an, die bei der Schwärzung berücksichtigt werden sollen.

   Weitere Informationen zur Liste der unterstützten PII Entitätstypen finden Sie unter Detect Personally Identifiable Information (PII) im Amazon Comprehend Developer Guide.

6. Lassen Sie die restlichen Einstellungen bei den Standardwerten.

   (Optional) Informationen zum Konfigurieren von zusätzlichen Einstellungen für Ihren speziellen Anwendungsfall finden Sie im Abschnitt Readme-Datei auf der linken Seite der Seite.
7. Aktivieren Sie das Kontrollkästchen neben Ich bestätige, dass diese App benutzerdefinierte IAM Rollen erstellt.
8. Wählen Sie Bereitstellen.
9. Wählen Sie auf der Seite der neuen Anwendung unter Ressourcen dieLogische ID der Lambda-Funktion, die Sie bereitgestellt haben, um die Funktion auf der Lambda-Funktionsseite zu überprüfen.

Schritt 5: Erstellen eines S3 Object Lambda Access Point

Ein S3 Object Lambda Access Point bietet die Flexibilität, eine Lambda-Funktion direkt von einer GET S3-Anfrage aus aufzurufen, sodass die Funktion PII Daten, die von einem S3-Zugangspunkt abgerufen wurden, unkenntlich machen kann. Beim Erstellen und Konfigurieren eines S3 Object Lambda Access Points müssen Sie die schwärzende Lambda-Funktion angeben, um den Ereigniskontext aufzurufen und im JSON Format als benutzerdefinierte Parameter für Lambda bereitzustellen.

Der Ereigniskontext stellt Informationen über die Anforderung bereit, die in dem von S3 Object Lambda an Lambda übergebenen Ereignis gestellt wird. Weitere Informationen zu allen Feldern im Ereigniskontext finden Sie unter Format und Verwendung des Ereigniskontexts.

So erstellen Sie einen S3 Object Lambda Access Point

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
3. Wählen Sie auf der Seite Object Lambda access points (Objekt-Lambda-Zugriffspunkte) die Option Create Object Lambda access point (Objekt-Lambda-Zugriffspunkt erstellen).
4. Geben Sie unter Name des Objekt-Lambda-Access Point den Namen ein, den Sie für den Object Lambda Access Point verwenden möchten (z. B. **`tutorial-pii-object-lambda-accesspoint`**).
5. Geben Sie für Unterstützenden Zugangspunkt den Standardzugangspunkt ein, den Sie in Schritt 3 erstellt haben (z. B. **`tutorial-pii-access-point`**), oder navigieren Sie zu diesem Zugangspunkt und wählen dann Unterstützenden Zugangspunkt auswählen aus.

6. Um für S3 Objekte aus dem S3-Bucket abzurufenAPIs, damit die Lambda-Funktion sie verarbeiten kann, wählen Sie GetObject.
7. Für die Funktion Lambda aufrufen können Sie eine der beiden folgenden Optionen für dieses Tutorial auswählen.
   - Wählen Sie Wählen Sie aus Funktionen in Ihrem Kontound wählen Sie die Lambda-Funktion aus, die Sie in Schritt 4 (z. B. **`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**) aus der Lambda-Funktion-Dropdown-Liste bereitgestellt haben.
   - Wählen Sie Enter ARN und geben Sie dann den Amazon-Ressourcennamen (ARN) der Lambda-Funktion ein, die Sie in Schritt 4 erstellt haben.
8. Wählen Sie für Lambda-Funktionsversion $ LATEST (die neueste Version der Lambda-Funktion, die Sie in Schritt 4 bereitgestellt haben).
9. (Optional) Wenn Sie Ihre Lambda-Funktion benötigen, um GET Anfragen mit Bereichs- und Artikelnummernüberschriften zu erkennen und zu verarbeiten, wählen Sie Lambda-Funktion unterstützt Anfragen mit Bereich und Lambda-Funktion unterstützt Anfragen mit Artikelnummern. Deaktivieren Sie andernfalls diese beiden Kontrollkästchen.

   Weitere Informationen zur Verwendung von Bereichs- oder Teilenummern mit S3 Object Lambda finden Sie unter Arbeitet mit Range and partNumber Header.
10. (Optional) Fügen Sie unter Payload — optional JSON Text hinzu, um Ihre Lambda-Funktion mit zusätzlichen Informationen zu versorgen.

    Eine Payload ist optionaler JSON Text, den Sie Ihrer Lambda-Funktion als Eingabe für alle Aufrufe bereitstellen können, die von einem bestimmten S3 Object Lambda Access Point kommen. Um das Verhalten für mehrere Object Lambda Access Points anzupassen, die dieselbe Lambda-Funktion aufrufen, können Sie Nutzlasten mit unterschiedlichen Parametern konfigurieren und so die Flexibilität Ihrer Lambda-Funktion erweitern.

    Weitere Informationen zur Nutzlast finden Sie unter Format und Verwendung des Ereigniskontexts.
11. (Optional) Wählen Sie für Anforderungsmetriken – optional die Option Deaktivieren oder Aktivieren aus, um Ihrem Object Lambda Access Point Amazon-S3-Überwachung hinzuzufügen. Anforderungsmetriken werden zum CloudWatch Standardtarif von Amazon abgerechnet. Weitere Informationen finden Sie unter CloudWatchPreise.
12. Behalten Sie unter Objekt-Lambda-Zugriffspunkt-Richtlinie - optional die Standardeinstellung bei.

(Optional) Sie können eine Ressourcenrichtlinie festlegen. Diese Ressourcenrichtlinie gewährt die `GetObject` API Erlaubnis, den angegebenen Object Lambda Access Point zu verwenden.

13. Lassen Sie die restlichen Einstellungen auf die Standardwerte eingestellt und klicken Sie auf Erstellen eines Objekt-Lambda-Zugriffspunkts.

Schritt 6: Verwenden des S3 Object Lambda Access Point zum Abrufen der redigierten Daten

Jetzt ist S3 Object Lambda bereit, PII Daten aus Ihrer Originaldatei zu redigieren.

So verwenden Sie den S3 Object Lambda Access Point zum Abrufen der redigierten Daten

Wenn Sie den Abruf einer Datei über Ihren S3 Object Lambda Access Point anfordern, `GetObject` API rufen Sie S3 Object Lambda auf. S3 Object Lambda ruft die Lambda-Funktion auf, um Ihre PII Daten zu redigieren, und gibt die transformierten Daten als Antwort auf den Standard-S3-Aufruf zurück. `GetObject` API

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
3. Wählen Sie auf der Seite Objekt-Lambda-Zugriffspunkte den S3 Object Lambda Access Point aus, den Sie in Schritt 5 erstellt haben (z. B. **`tutorial-pii-object-lambda-accesspoint`**).
4. Wählen Sie auf der Registerkarte Objekte Ihres S3 Object Lambda Access Point die Datei mit dem gleichen Namen (z. B. `tutorial.txt`) wie diejenige aus, die Sie in Schritt 2 in den S3-Bucket hochgeladen haben.

   Diese Datei sollte alle transformierten Daten enthalten.
5. Um die transformierten Daten anzuzeigen, wählen Sie Öffnen oder Herunterladen aus.

   Sie sollten in der Lage sein, die redigierte Datei zu sehen, wie im folgenden Beispiel gezeigt.

```
Hello *********. Your AnyCompany Financial Services,
LLC credit card account ******************* has a minimum payment
of $24.53 that is due by *********. Based on your autopay settings,
we will withdraw your payment on the due date from your
bank account ********** with the routing number *********.

Your latest statement was mailed to ********************************.
```

```
After your payment is received, you will receive a confirmation
text message at ************.
If you have questions about your bill, AnyCompany Customer Service
is available by phone at ************ or
email at **********************.
```

Schritt 7: Bereinigen

Wenn Sie Ihre Daten nur zu Lernzwecken mit S3 Object Lambda redigiert haben, löschen Sie die AWS Ressourcen, die Sie zugewiesen haben, sodass Ihnen keine Gebühren mehr anfallen.

Teilschritte

- Löschen des Object Lambda Access Point
- Löschen des S3-Zugriffspunkts
- Löschen Sie die Lambda-Funktion
- Löschen Sie die CloudWatch Protokollgruppe
- Löschen Sie die Originaldatei im S3-Quell-Bucket
- Löschen des S3-Quell-Bucket
- Löschen Sie die IAM Rolle für Ihre Lambda-Funktion
- Löschen Sie die vom Kunden verwaltete Richtlinie für Ihren IAM Benutzer
- Löschen Sie den IAM Benutzer

Löschen des Object Lambda Access Point

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Objekt-Lambda-Zugriffspunkte aus.
3. Wählen Sie auf der Seite Objekt-Lambda-Zugriffspunkte die Optionsschaltfläche links neben dem S3 Object Lambda Access Point aus, den Sie in Schritt 5 erstellt haben (z. B. **`tutorial-pii-object-lambda-accesspoint`**).
4. Wählen Sie Delete (Löschen).
5. Bestätigen Sie, dass Sie Ihren Object Lambda Access Point löschen möchten, indem Sie den Namen in das angezeigte Textfeld eingeben und dann Löschen wählen.

Löschen des S3-Zugriffspunkts

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Zugriffspunkte aus.
3. Navigieren Sie zu dem Zugriffspunkt, den Sie in Schritt 3 (z. B. **`tutorial-pii-access-point`**) erstellt haben, und wählen Sie die Optionsschaltfläche neben dem Namen des Zugriffspunkts aus.
4. Wählen Sie Löschen.
5. Bestätigen Sie, dass Sie Ihren Zugriffspunkt löschen möchten, indem Sie den Namen in das angezeigte Textfeld eingeben und dann Delete (Löschen) wählen.

Löschen Sie die Lambda-Funktion

1. Wählen Sie in der AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/im linken Navigationsbereich Funktionen aus.
2. Wählen Sie die Funktion aus, die Sie in Schritt 4 (z. B. **`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**) erstellt haben.
3. Wählen Sie Aktionen und anschließend Löschen aus.
4. Wählen Sie im Bestätigungsdialogfeld Delete (Löschen) die Option Delete (Löschen) aus.

Löschen Sie die CloudWatch Protokollgruppe

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im linken Navigationsbereich Log groups (Protokollgruppen) aus.
3. Suchen Sie die Protokollgruppe, deren Name mit der Lambda-Funktion endet, die Sie in Schritt 4 (z. B. **`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**) erstellt haben.
4. Wählen Sie Actions (Aktionen) und dann Delete log group(s) (Protokollgruppe(n) löschen) aus.
5. Wählen Sie im Dialogfeld Delete log group(s) (Protokollgruppe(n) löschen) die Option Delete (Löschen) aus.

Löschen Sie die Originaldatei im S3-Quell-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie in der Liste Bucket-Name den Namen des Buckets aus, in den Sie die Originaldatei in Schritt 2 (z. B. **`tutorial-bucket`**) hochgeladen haben.
4. Markieren Sie das Kontrollkästchen links neben dem Namen des Objekts, das Sie löschen möchten (z. B. `tutorial.txt`).
5. Wählen Sie Löschen.
6. Bestätigen Sie auf der Seite Löschen von Objekten im Abschnitt Objekte endgültig löschen?, dass Sie dieses Objekt löschen möchten, indem Sie **`permanently delete`** im Textfeld eingeben.
7. Wählen Sie Delete objects (Objekte löschen).

Löschen des S3-Quell-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie in der Liste Buckets das Optionsfeld neben dem Namen des Buckets aus, den Sie in Schritt 1 (z. B. **`tutorial-bucket`**) erstellt haben.
4. Wählen Sie Löschen.
5. Bestätigen Sie auf der Seite Delete bucket (Bucket löschen), dass Sie den Bucket löschen möchten. Geben Sie dazu den Bucket-Namen in das Textfeld ein und wählen Sie Delete bucket (Bucket löschen).

Löschen Sie die IAM Rolle für Ihre Lambda-Funktion

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Roles (Rollen), und aktivieren Sie dann das Kontrollkästchen neben der Rolle, die Sie löschen möchten. Der Rollenname beginnt mit dem Namen der Lambda-Funktion, die Sie in Schritt 4 (z. B. **`serverlessrepo-ComprehendPiiRedactionS3ObjectLambda`**) erstellt haben.
3. Wählen Sie Löschen.
4. Geben Sie im Dialogfeld Löschen den Rollennamen in das Texteingabefeld ein, um das Löschen zu bestätigen. Wählen Sie dann Löschen aus.

Löschen Sie die vom Kunden verwaltete Richtlinie für Ihren IAM Benutzer

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich die Option Richtlinien aus.
3. Geben Sie auf der Seite Richtlinien den Namen der vom Kunden verwalteten Richtlinie ein, die Sie in Voraussetzungen (z. B. **`tutorial-serverless-application-repository`**) erstellt haben, in das Suchfeld ein, um die Richtlinienliste zu filtern. Aktivieren Sie die Optionsschaltfläche neben dem Namen der Richtlinie, die Sie löschen möchten.
4. Wählen Sie Aktionen und anschließend Löschen aus.
5. Bestätigen Sie, dass Sie diese Richtlinie löschen möchten, indem Sie den Namen in das angezeigte Textfeld eingeben und dann Delete (Löschen) wählen.

Löschen Sie den IAM Benutzer

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Benutzer, und aktivieren Sie dann das Kontrollkästchen neben dem Benutzer, den Sie löschen möchten.
3. Wählen Sie oben auf der Seite Delete (Löschen) aus.
4. Im Bereich Löschen ***`user name`***? Geben Sie im Dialogfeld den Benutzernamen in das Texteingabefeld ein, um das Löschen des Benutzers zu bestätigen. Wählen Sie Löschen.

Nächste Schritte

Nachdem Sie dieses Tutorial abgeschlossen haben, können Sie die folgenden verwandten Anwendungsfälle näher untersuchen:

- Sie können mehrere S3 Object Lambda Access Points erstellen und sie mit vorgefertigten Lambda-Funktionen aktivieren, die PII je nach den Geschäftsanforderungen der Datenzugriffer unterschiedlich konfiguriert sind, um bestimmte Typen zu redigieren.

  Jeder Benutzertyp nimmt eine IAM Rolle an und hat nur Zugriff auf einen S3 Object Lambda Access Point (verwaltet durch IAM Richtlinien). Dann fügen Sie jede `ComprehendPiiRedactionS3ObjectLambda`-Lambda-Funktion, die für einen anderen Redigierungsanwendungsfall konfiguriert ist, an einen anderen S3 Object Lambda Access Point

an. Für jeden S3 Object Lambda Access Point können Sie über einen unterstützenden S3-Zugriffspunkt verfügen, um Daten aus einem S3-Bucket zu lesen, der den freigegebenen Datensatz speichert.

Weitere Informationen zum Erstellen einer S3-Bucket-Richtlinie, die es Benutzern ermöglicht, nur über S3-Zugriffspunkte aus dem Bucket zu lesen, finden Sie unter Konfiguration von IAM Richtlinien für die Verwendung von Access Points.

Weitere Informationen dazu, wie Sie einem Benutzer die Berechtigung für den Zugriff auf die Lambda-Funktion, den S3-Zugriffspunkt und den S3 Object Lambda Access Point erteilen, finden Sie unter Konfiguration von IAM Richtlinien für Object Lambda Access Points.

- Sie können Ihre eigene Lambda-Funktion erstellen und S3 Object Lambda mit Ihrer angepassten Lambda-Funktion verwenden, um Ihre spezifischen Datenanforderungen zu erfüllen.

  Um beispielsweise verschiedene Datenwerte zu untersuchen, können Sie S3 Object Lambda und Ihre eigene Lambda-Funktion verwenden, die zusätzliche Amazon-Comprehend-Funktionen verwenden, z. B. Entitätserkennung, Schlüsselphrasenerkennung, Stimmungsanalyse und Dokumentklassifizierung, um Daten zu verarbeiten. Sie können S3 Object Lambda auch zusammen mit Amazon Comprehend Medical, einem HIPAA geeigneten NLP Service, verwenden, um Daten kontextsensitiv zu analysieren und zu extrahieren.

  Weitere Informationen zum Transformieren von Daten mit S3 Object Lambda und Ihrer eigenen Lambda-Funktion finden Sie unter Tutorial: Transformieren von Daten für Ihre Anwendung mit S3 Object Lambda.

Debuggen und Problembehandlung bei S3 Object Lambda

Anforderungen an Zugriffspunkte von Amazon S3 Object Lambda können zu neuen Fehlerantworten führen, wenn mit dem Aufruf oder der Ausführung der Lambda-Funktion etwas schief geht. Diese Fehler folgen dem gleichen Format wie die Standardfehler von Amazon S3. Informationen zu S3 Object Lambda-Fehlern finden Sie in der Liste der S3 Object Lambda-Fehlercodes in der Amazon Simple Storage Service API Reference.

Weitere Informationen zum allgemeinen Debuggen von Lambda-Funktionen finden Sie unter Monitoring and troubleshooting Lambda applications (Überwachung und Fehlerbehebung bei Lambda-Anwendungen) im AWS Lambda -Entwicklerhandbuch.

Informationen zu standardmäßigen Amazon S3 S3-Fehlern finden Sie unter Error Responses in der Amazon Simple Storage Service API Reference.

Sie können in Amazon Anforderungsmetriken CloudWatch für Ihre Object Lambda Access Points aktivieren. Diese Metriken helfen Ihnen dabei, die Betriebsleistung Ihrer Zugriffspunkte zu überwachen. Sie können Anforderungsmetriken während oder nach der Erstellung Ihres Object Lambda Access Point aktivieren. Weitere Informationen finden Sie unter Lambda-Anforderungsmetriken für S3-Objekte in CloudWatch.

Um eine detailliertere Protokollierung der an Ihre Object Lambda Access Points gesendeten Anforderungen zu erhalten, können Sie AWS CloudTrail -Datenereignisse aktivieren. Weitere Informationen finden Sie unter Protokollieren von Datenereignissen für Trails im AWS CloudTrail -Benutzerhandbuch.

Informationen zu S3-Objekt-Lambda-Tutorials finden Sie im Folgenden:

- Tutorial: Transformieren von Daten für Ihre Anwendung mit S3 Object Lambda
- Tutorial: Erkennen und Redigieren von PII Daten mit S3 Object Lambda und Amazon Comprehend
- Tutorial: Verwenden von S3 Object Lambda, um Bilder beim Abrufen dynamisch mit Wasserzeichen zu versehen

Weitere Hinweise zu standardmäßigen Zugriffspunkten finden Sie unter Verwaltung des Zugriffs auf gemeinsam genutzte Datensätze mit Access Points.

Hinweise zum Arbeiten mit Buckets finden Sie unter Bucket-Übersicht. Weitere Informationen zur Arbeit mit Objekten finden Sie unter Übersicht über Amazon-S3-Objekte.

Massenausführung von Objektoperationen mit Batch Operations

Sie können S3-Batch-Vorgänge verwenden, um umfangreiche Batch-Vorgänge für Amazon S3-Objekte durchzuführen. S3-Batch-Vorgänge kann eine einzelne Operation für Listen von Amazon S3-Objekten durchführen, die Sie angeben. Ein einziger Auftrag kann eine festgelegte Operation auf Milliarden von Objekten mit mehreren Exabytes an Daten durchführen. Amazon S3 verfolgt den Fortschritt, versendet Benachrichtigungen und speichert einen detaillierten Abschlussbericht zu allen Aktionen. So profitieren Sie von einer vollständig verwalteten, prüfbaren und serverlosen Umgebung. Sie können S3 Batch Operations über die Amazon S3-Konsole AWS CLI, AWS SDKs, oder Amazon S3 verwenden RESTAPI.

Verwenden Sie S3 Batch Operations, um Objekte zu kopieren und Objekt-Tags oder Zugriffskontrolllisten festzulegen (ACLs). Sie können Objekt-Wiederherstellungen auch von S3 Glacier Flexible Retrieval aus initiieren oder eine AWS Lambda -Funktion aufrufen, um benutzerdefinierte Aktionen mit Ihren Objekten durchzuführen. Sie können diese Vorgänge für eine benutzerdefinierte Liste von Objekten durchführen oder mit einem Amazon S3 Inventory einfach Objektlisten erzeugen. Amazon S3 Batch Operations verwendet dieselben Amazon API S3-Operationen, die Sie bereits mit Amazon S3 verwenden.

> **Note**
>
> Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter S3 Express One Zone und Arbeiten mit Verzeichnis-Buckets. Weitere Informationen zur Verwendung von Batch Operations mit S3 Express One Zone und Verzeichnis-Buckets finden Sie unter Verwenden von Batch Operations mit Verzeichnis-Buckets..

Grundlagen von S3-BatchVorgänge

Sie können S3-Batch-Vorgänge verwenden, um umfangreiche Batch-Vorgänge für Amazon S3-Objekte durchzuführen. S3-Batch-Vorgänge kann eine einzelne Operation oder Aktion in Listen von Amazon S3-Objekten ausführen, die Sie angeben.

Terminologie

In diesem Abschnitt werden die Begriffe Manifeste, Jobs, Operationen und Aufgaben verwendet, die wie folgt definiert sind:

Manifest

Ein Manifest ist ein Amazon-S3-Objekt, das Objektschlüssel enthält, die Amazon S3 bearbeiten soll. Wenn Sie einen Batch Operations-Job erstellen möchten, müssen Sie ein Manifest angeben. Ihr benutzergeneriertes Manifest muss den Bucket-Namen, den Objektschlüssel und optional die Objektversion für jedes Objekt enthalten. Wenn Sie ein benutzergeneriertes Manifest bereitstellen, muss es in Form eines Amazon S3 S3-Lagerberichts oder einer CSV Datei vorliegen.

Sie können Amazon S3 auch automatisch ein Manifest auf der Grundlage von Objektfilterkriterien generieren lassen, die Sie bei der Erstellung Ihres Jobs angeben. Diese Option ist für S3-Batch-Replikationsaufträge verfügbar, die Sie in der Amazon S3-Konsole erstellen, oder für jeden

Auftragstyp, den Sie mit AWS Command Line Interface (AWS CLI) AWS SDKs, oder Amazon S3 erstellen RESTAPI.

Aufgabe

Ein Auftrag ist die grundlegende Arbeitseinheit für S3-BatchVorgänge. Eine Aufgabe enthält alle Informationen, die erforderlich sind, um die angegebene Operation für die im Manifest aufgeführten Objekte auszuführen. Sobald Sie diese Informationen bereitgestellt und den Beginn des Auftrags angefordert haben, führt die Aufgabe die Operation für alle Objekte im Manifest durch.

Operation

Bei dem Vorgang handelt es sich um die Art der API Aktion, z. B. das Kopieren von Objekten, die der Job Batch Operations ausführen soll. Jeder Auftrag führt einen einzelnen Typ von Operation für alle Objekte aus, die im Manifest angegeben sind.

Aufgabe

Eine Aufgabe ist die Ausführungseinheit für einen Auftrag. Eine Aufgabe stellt einen einzelnen Aufruf an einen Amazon S3 oder eine AWS Lambda API Operation dar, um den Vorgang des Auftrags für ein einzelnes Objekt auszuführen. Während der Lebensdauer eines Auftrag erstellt S3-Batch-Vorgänge eine Aufgabe für jedes Objekt, das im Manifest angegeben ist.

Funktionsweise eines S3-Batchoperations-Auftrags

Ein Auftrag ist die grundlegende Arbeitseinheit für S3-BatchVorgänge. Eine Aufgabe enthält alle Informationen, die erforderlich sind, um die angegebene Operation für eine Liste von Objekten auszuführen. Um einen Auftrag zu erstellen, übergeben Sie S3-Batch-Vorgänge eine Liste von Objekten und geben die Aktion an, die für diese Objekte ausgeführt werden soll.

Informationen zu den Vorgängen, die S3 Batch Operations unterstützt, finden Sie unter Von S3 Batch-Vorgänge unterstützte Vorgänge.

Ein Batch-Job führt einen bestimmten Vorgang für jedes Objekt aus, das in seinem Manifest enthalten ist. Ein Manifest listet die Objekte auf, die ein Batch-Auftrag verarbeiten soll, und es wird als Objekt in einem Bucket gespeichert. Sie können einen Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory Bericht im Format mit kommagetrennten Werten (CSV) als Manifest verwenden, wodurch es einfach ist, große Listen von Objekten zu erstellen, die sich in einem Bucket befinden. Sie können auch ein Manifest in einem einfachen CSV Format angeben, das es Ihnen ermöglicht, Batch-

Operationen für eine benutzerdefinierte Liste von Objekten durchzuführen, die in einem einzelnen Bucket enthalten sind.

Nachdem Sie einen Auftrag erstellt haben, verarbeitet Amazon S3 die Liste der Objekte in dem Manifest und führt die festgelegte Operation für jedes Objekt aus. Während ein Auftrag ausgeführt wird, können Sie den Fortschritt programmgesteuert oder über die Amazon S3-Konsole überwachen. Sie können einen Auftrag auch so konfigurieren, dass er nach Ende der Ausführung einen Abschlussbericht erzeugt. Der Abschlussbericht beschreibt die Ergebnisse jeder Aufgabe, die von dem Auftrag durchgeführt wurde. Weitere allgemeine Informationen zur Überwachung von Aufträgen finden Sie unter Verwalten von S3-Batch-Vorgangsaufträgen.

Tutorial zu S3-Batchvorgängen

Das folgende Tutorial enthält vollständige end-to-end Verfahren für einige Batch-Operationsaufgaben.

- Tutorial: Batch-Transcodierung von Videos mit S3 Batch Operations

Gewähren von Berechtigungen für Batchoperationen

Bevor Sie S3 Batch Operation-Aufträge erstellen und ausführen, müssen Sie die erforderlichen Berechtigungen erteilen. Um einen Amazon-S3-Batch-Vorgangsauftrag zu erstellen, ist die `s3:CreateJob`-Benutzerberechtigung erforderlich. Dieselbe Entität, die den Job erstellt, muss auch die `iam:PassRole` Berechtigung haben, die Rolle AWS Identity and Access Management (IAM), die für den Job angegeben ist, an Batch Operations zu übergeben.

Allgemeine Informationen zur Angabe von IAM Ressourcen finden Sie unter IAMJSONRichtlinie, Ressourcenelemente im IAMBenutzerhandbuch. Die folgenden Abschnitte enthalten Informationen zum Erstellen einer IAM Rolle und zum Anhängen von Richtlinien.

Themen

- Eine IAM S3-Rolle für Batch Operations erstellen
- Anfügen von Berechtigungsrichtlinien

Eine IAM S3-Rolle für Batch Operations erstellen

Amazon S3 benötigt Berechtigungen, um S3-Batch-Vorgänge für Sie auszuführen. Sie gewähren diese Berechtigungen über eine AWS Identity and Access Management (IAM) -Rolle. Dieser Abschnitt enthält Beispiele für die Vertrauens- und Berechtigungsrichtlinien, die Sie beim Erstellen

einer IAM Rolle verwenden. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter IAMRollen. Beispiele finden Sie unter Steuern von Berechtigungen für Batch-Operationen mithilfe von Job-Tags und Kopieren von Objekten mit S3 BatchVorgänge.

In Ihren IAM Richtlinien können Sie auch Bedingungsschlüssel verwenden, um Zugriffsberechtigungen für S3 Batch Operations-Jobs zu filtern. Weitere Informationen und eine vollständige Liste der Amazon S3-spezifischen Bedingungsschlüssel finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

Vertrauensrichtlinie

Damit der Dienstprinzipal von S3 Batch Operations die IAM Rolle übernehmen kann, fügen Sie der Rolle die folgende Vertrauensrichtlinie hinzu.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"batchoperations.s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

Anfügen von Berechtigungsrichtlinien

Je nach Art der Operation können Sie eine der folgenden Richtlinien anfügen.

Bevor Sie Berechtigungen konfigurieren, beachten Sie Folgendes:

- Unabhängig von der Operation benötigt Amazon S3 Berechtigungen, um das Manifestobjekt aus Ihrem S3-Bucket zu lesen und optional einen Bericht in Ihren Bucket zu schreiben. Daher enthalten alle der folgenden Richtlinien diese Berechtigungen.
- Für Amazon S3 S3-Inventarberichtsmanifeste benötigt S3 Batch Operations die Erlaubnis, das manifest.json-Objekt und alle zugehörigen CSV Datendateien zu lesen.

- Versionsspezifische Berechtigungen wie `s3:GetObjectVersion` sind nur erforderlich, wenn Sie die Versions-ID der Objekte festlegen.
- Wenn Sie S3-Batchoperationen für verschlüsselte Objekte ausführen, muss die IAM Rolle auch Zugriff auf die AWS KMS Schlüssel haben, mit denen sie verschlüsselt wurden.
- Wenn Sie ein mit verschlüsseltes Inventarberichtsmanifest einreichen AWS KMS, muss Ihre IAM Richtlinie die Berechtigungen `"kms:Decrypt"` `"kms:GenerateDataKey"` für das Objekt manifest.json und alle zugehörigen CSV Datendateien enthalten.
- Wenn der Batch Operations-Job ein Manifest in einem Bucket generiert, für den Zugriffskontrolllisten (ACLs) aktiviert sind und sich in einem anderen Bucket befindet AWS-Konto, müssen Sie die `s3:PutObjectAcl` Berechtigung in der IAM Richtlinie der IAM Rolle erteilen, die für den Batch-Job konfiguriert ist. Wenn Sie diese Berechtigung nicht angeben, schlägt der Batch-Job mit der folgenden Fehlermeldung fehl`Error occurred when preparing manifest: Failed to write manifest`.

Objekte kopieren: PutObject

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:PutObjectTagging"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
        },
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:GetObjectTagging",
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-source-bucket",
                "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
            ]
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
            ]
        }
    ]
}
```

Objekt-Tagging ersetzen: PutObjectTagging

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObjectTagging",
        "s3:PutObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
```

```
        "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObject"
      ],
      "Resource":[
        "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
      ]
    }
  ]
}
```

Objekt-Tagging löschen: DeleteObjectTagging

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
              "s3:DeleteObjectTagging",
              "s3:DeleteObjectVersionTagging"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
            ]
        }
    ]
}
```

Zugriffskontrollliste ersetzen: PutObjectAcl

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObjectAcl",
        "s3:PutObjectVersionAcl"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObject"
      ],
      "Resource":[
        "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
      ]
    }
  ]
}
```

Objekte wiederherstellen: RestoreObject

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
          "s3:RestoreObject"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObject"
      ],
      "Resource":[
        "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
      ]
    }
  ]
}
```

Aufbewahrung von Object Lock anwenden: PutObjectRetention

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetBucketObjectLockConfiguration",
            "Resource": [
```

```
                "arn:aws:s3:::amzn-s3-demo-destination-bucket"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObjectRetention",
                "s3:BypassGovernanceRetention"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
            ]
        }
    ]
}
```

Rechtliche Sperre für Objektsperre anwenden: PutObjectLegalHold

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetBucketObjectLockConfiguration",
```

```
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-destination-bucket"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "s3:PutObjectLegalHold",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
            ]
        }
    ]
}
```

Existierende Objekte replizieren: InitiateReplication mit einem von S3 generierten Manifest

Verwenden Sie diese Richtlinie, wenn Sie ein von S3 generiertes Manifest verwenden und speichern. Weitere Informationen zur Verwendung von Batch-Operationen zum Replizieren vorhandener Objekte finden Sie unterReplizieren vorhandener Objekte mit Batch Replication.

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

```
         "Action":[
            "s3:InitiateReplication"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
         ]
      },
      {
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:PutInventoryConfiguration"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
         ]
      },
      {
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*",
            "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
         ]
      }
   ]
}
```

Replizieren Sie vorhandene Objekte: InitiateReplication mit einem Benutzermanifest

Verwenden Sie diese Richtlinie, wenn Sie ein vom Benutzer bereitgestelltes Manifest verwenden. Weitere Informationen zur Verwendung von Batch-Operationen zum Replizieren vorhandener Objekte finden Sie unterReplizieren vorhandener Objekte mit Batch Replication.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action":[
            "s3:InitiateReplication"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
         ]
      },
            {
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
         ]
      }
   ]
}
```

Erstellen eines S3-Batch-Vorgangsauftrags

Mit S3 Batch Operations können Sie umfangreiche Stapelvorgänge für eine Liste bestimmter Amazon-S3-Objekte durchführen. In diesem Abschnitt werden die Informationen beschrieben, die Sie zum Erstellen eines S3-Batch-Vorgangsauftrags benötigen, sowie die Ergebnisse einer `CreateJob`-Anforderung. Es enthält auch Anweisungen zum Erstellen eines Batch Operations-Jobs mithilfe der Amazon S3 S3-Konsole, AWS Command Line Interface (AWS CLI) und AWS SDK for Java.

Wenn Sie einen S3-Batchvorgangs-Auftrag erstellen, können Sie einen Abschlussbericht für alle oder nur für die fehlgeschlagenen Aufträge anfordern. Sofern mindestens eine Aufgabe erfolgreich aufgerufen wurde, generiert S3 Batch Operations einen Bericht für abgeschlossene, fehlgeschlagene oder abgebrochene Aufträge. Weitere Informationen finden Sie unter Beispiele: Abschlussberichte zu S3-BatchVorgänge.

Themen

- Batch-Vorgangsauftrag-Anforderungselemente
- Angeben eines Manifests

Batch-Vorgangsauftrag-Anforderungselemente

Um einen S3-Batch-Vorgangsauftrag zu erstellen, müssen Sie die folgenden Informationen angeben:

Operation

Geben Sie die Operation an, die S3-Batch-Vorgänge für die Objekte im Manifest ausführen soll. Jeder Operationstyp akzeptiert Parameter, die für diesen Vorgang spezifisch sind. Mit Batch Operations können Sie eine Operation in großen Mengen ausführen, mit den gleichen Ergebnissen, als ob Sie diese Operation one-by-one für jedes Objekt ausgeführt hätten.

Manifest

Das Manifest ist eine Liste aller Objekte, für die S3 Batch Operations die festgelegte Aktion ausführen soll. Sie können die folgenden Verfahren verwenden, um ein Manifest für einen Batch-Operations-Auftrag anzugeben:

- Erstellen Sie manuell Ihre eigene benutzerdefinierte, CSV formatierte Objektliste.
- Wählen Sie einen vorhandenen CSV Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory -formatierten Bericht aus.

- Weisen Sie Batch Operations an, automatisch ein Manifest auf der Grundlage von Objektfilterkriterien zu generieren, die Sie bei der Erstellung Ihres Auftrags angeben. Diese Option ist für Batch-Replikationsaufträge verfügbar, die Sie in der Amazon S3-Konsole erstellen, oder für jeden Auftragstyp, den Sie mithilfe von AWS CLI AWS SDKs, oder Amazon S3 erstellen RESTAPI.

> **Note**
>
> - Unabhängig davon, wie Sie Ihr Manifest angeben, muss die Liste selbst in einem Allzweck-Bucket gespeichert werden. Batch Operations kann keine vorhandenen Manifeste aus Verzeichnis-Buckets importieren (oder generierte Manifeste in Verzeichnis-Buckets speichern). Im Manifest beschriebene Objekte können jedoch in Verzeichnis-Buckets gespeichert werden. Weitere Informationen finden Sie unter Verzeichnis-Buckets.
> - Wenn sich die Objekte in Ihrem Manifest in einem versionierten Bucket befinden, weist die Angabe der Version IDs für die Objekte Batch Operations an, den Vorgang für eine bestimmte Version auszuführen. Wenn keine Version IDs angegeben ist, führt Batch Operations den Vorgang mit der neuesten Version der Objekte aus. Wenn Ihr Manifest ein Versions-ID-Feld enthält, müssen Sie eine Versions-ID für alle Objekte im Manifest angeben.

Weitere Informationen finden Sie unter Angeben eines Manifests.

Priorität

Mit Auftragsprioritäten können Sie angeben, welche relative Priorität dieser Auftrag gegenüber den anderen in Ihrem Konto ausgeführten Aufträgen besitzt. Eine höhere Nummer bedeutet eine höhere Priorität.

Auftragsprioritäten sind nur in Beziehung zu den für andere Aufträge in demselben Konto und in der derselben Region festgelegten Prioritäten bedeutsam. Sie können wählen, welches Nummerierungssystem für Sie funktioniert. Beispielsweise können Sie allen Restore (`RestoreObject`) -Aufträgen die Priorität 1, allen Copy (`CopyObject`) -Aufträgen die Priorität 2 und allen Replace-Zugriffskontrolllisten (ACLs) (**`PutObjectAcl`**) -Aufträgen die Priorität 3 zuweisen.

S3 Batch Operations priorisiert Aufträge gemäß den Prioritätszahlen, garantiert aber keine strikte Sortierung. Daher sind Auftrags-Prioritäten nicht dazu geeignet, sicherzustellen, dass ein

Auftrag vor einem anderen Auftrag gestartet oder beendet wird. Wenn Sie eine strikte Sortierung gewährleisten möchten, müssen Sie den Abschluss eines Auftrags abwarten, bevor Sie den nächsten starten.

RoleArn

Geben Sie eine AWS Identity and Access Management (IAM) Rolle an, um den Job auszuführen. Die IAM Rolle, die Sie verwenden, muss über ausreichende Berechtigungen verfügen, um den im Job angegebenen Vorgang auszuführen. Um beispielsweise einen `CopyObject` Job auszuführen, muss die IAM Rolle über die `s3:GetObject` Berechtigung für den Quell-Bucket und die `s3:PutObject` Berechtigung für den Ziel-Bucket verfügen. Außerdem braucht die Rolle die Berechtigungen, das Manifest zu lesen und den Auftragsabschlussbericht zu schreiben.

Weitere Informationen zu IAM IAMRollen finden Sie im IAMBenutzerhandbuch unter Rollen.

Weitere Informationen zu Amazon-S3-Berechtigungen finden Sie unter Politische Maßnahmen für Amazon S3.

> **Note**
>
> Batch-Operations-Aufträge, die Aktionen für Verzeichnis-Buckets ausführen, erfordern bestimmte Berechtigungen. Weitere Informationen finden Sie unter AWS Identity and Access Management (IAM) für S3 Express One Zone.

Bericht

Geben Sie an, ob S3-Batch-Vorgänge einen Abschlussbericht erstellen soll. Wenn Sie einen Auftragsabschlussbericht anfordern, müssen Sie auch die Parameter für den Bericht in diesem Element angeben. Die notwendigen Informationen umfassen:

- Den Bucket, in dem der Bericht gespeichert werden soll

  > **Note**
  >
  > Der Bericht muss in einem Allzweck-Bucket gespeichert werden. Batch Operations kann Berichte nicht in Verzeichnis-Buckets speichern. Weitere Informationen finden Sie unter Verzeichnis-Buckets.

- Das Format des Berichts
- Ob der Bericht die Details aller oder nur fehlgeschlagener Aufgaben enthält

- Eine optionale Präfix-Zeichenfolge

> **Note**
>
> Abschlussberichte werden immer mit serverseitiger Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt.

Markierungen (optional)

Sie können den Zugriff auf Ihre S3-Batch-Vorgänge-Aufträge kennzeichnen und steuern, indem Sie Markierungen hinzufügen. Sie können Tags verwenden, um zu identifizieren, wer für einen Batch-Operations-Auftrag verantwortlich ist, oder um zu steuern, wie Benutzer mit einzelnen Batch-Operations-Aufträgen interagieren. Das Vorhandensein von Aufgabe-Markierungen kann einem Benutzer die Möglichkeit gewähren oder einschränken, eine Aufgabe abzubrechen, eine Aufgabe im Bestätigungsstatus zu aktivieren oder die Prioritätsstufe einer Aufgabe zu ändern. Beispielsweise können Sie einem Benutzer die Berechtigung zum Aufrufen des `CreateJob`-Vorgangs erteilen, vorausgesetzt, der Auftrag wird mit dem Tag `"Department=Finance"` erstellt.

Sie können Aufträge mit bereits zugeordneten Markierungen erstellen und Sie können bereits erstellten Aufträgen nachträglich Markierungen hinzufügen.

Weitere Informationen finden Sie unter the section called “Verwenden von Markierungen”.

Description (optional)

Um Ihren Auftrag zu verfolgen und zu überwachen, können Sie auch eine Beschreibung von bis zu 256 Zeichen angeben. Amazon S3 enthält diese Beschreibung immer dann, wenn Informationen über einen Auftrag zurückgegeben oder Auftragsdetails auf der Amazon-S3-Konsole angezeigt werden. So können Sie Aufträge ganz einfach nach den ihnen zugewiesenen Beschreibungen sortieren und filtern. Beschreibungen müssen nicht eindeutig sein, sodass Sie Beschreibungen auch als Kategorien (z. B. "Wöchentliche Aufträge zum Kopieren von Protokollen") verwenden können, um Gruppen ähnlicher Aufträge nachzuverfolgen.

Angeben eines Manifests

Ein Manifest ist ein Amazon-S3-Objekt, das Objektschlüssel enthält, die Amazon S3 bearbeiten soll. Sie können ein Manifest mit einer der folgenden Methoden bereitstellen:

- Erstellen Sie eine neue Manifestdatei manuell.
- Verwenden Sie ein vorhandenes Manifest.
- Weisen Sie Batch Operations an, automatisch ein Manifest auf der Grundlage von Objektfilterkriterien zu generieren, die Sie bei der Erstellung Ihres Auftrags angeben. Diese Option ist für Batch-Replikationsaufträge verfügbar, die Sie in der Amazon S3-Konsole erstellen, oder für jeden Auftragstyp, den Sie mithilfe von AWS CLI AWS SDKs, oder Amazon S3 erstellen RESTAPI.

> **Note**
>
> - Amazon S3 Batch Operations unterstützt keine regionsübergreifende Manifestgenerierung.
> - Unabhängig davon, wie Sie Ihr Manifest angeben, muss die Liste selbst in einem Allzweck-Bucket gespeichert werden. Batch Operations kann keine vorhandenen Manifeste aus Verzeichnis-Buckets importieren (oder generierte Manifeste in Verzeichnis-Buckets speichern). Im Manifest beschriebene Objekte können jedoch in Verzeichnis-Buckets gespeichert werden. Weitere Informationen finden Sie unter Verzeichnis-Buckets.

Erstellen einer Manifestdatei

Um eine Manifestdatei manuell zu erstellen, geben Sie den Manifest-Objektschlüssel ETag (Entity-Tag) und die optionale Versions-ID in einer CSV -formatierten Liste an. Der Inhalt des Manifests muss -codiert sein. URL

Standardmäßig verwendet Amazon S3 automatisch serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3), um ein Manifest zu verschlüsseln, das in einen Amazon S3 S3-Bucket hochgeladen wird. Manifeste, die serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) verwenden, werden nicht unterstützt. Manifeste, die serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-KMS) verwenden, werden nur unterstützt, wenn Sie Inventarberichte im -Format verwendenCSV. Die Verwendung eines manuell erstellten Manifests mit AWS KMS wird nicht unterstützt.

Ihr Manifest muss den Bucket-Namen, den Objektschlüssel und (optional) die Objektversion enthalten. Sämtliche anderen Felder in dem Manifest werden nicht von S3-Batch-Vorgänge verwendet.

> **Note**
>
> Wenn sich die Objekte in Ihrem Manifest in einem versionierten Bucket befinden, weist die Angabe der Version IDs für die Objekte Batch Operations an, den Vorgang für eine bestimmte Version auszuführen. Wenn keine Version IDs angegeben ist, führt Batch Operations den Vorgang mit der neuesten Version der Objekte aus. Wenn Ihr Manifest ein Versions-ID-Feld enthält, müssen Sie eine Versions-ID für alle Objekte im Manifest angeben.

Im Folgenden finden Sie ein Beispiel für ein Manifest im CSV Format ohne VersionIDs.

```
amzn-s3-demo-bucket1,objectkey1
amzn-s3-demo-bucket1,objectkey2
amzn-s3-demo-bucket1,objectkey3
amzn-s3-demo-bucket1,photos/jpgs/objectkey4
amzn-s3-demo-bucket1,photos/jpgs/newjersey/objectkey5
amzn-s3-demo-bucket1,object%20key%20with%20spaces
```

Das Folgende ist ein Beispiel für ein Manifest in einem CSV Format, das die Version enthältIDs.

```
amzn-s3-demo-bucket1,objectkey1,PZ9ibn9D5lP6p298B7S9_ceqx1n5EJ0p
amzn-s3-demo-bucket1,objectkey2,YY_ouuAJByNW1LRBfFMfxMge7XQWxMBF
amzn-s3-demo-bucket1,objectkey3,jbo9_jhdPEyB4RrmOxWS0kU0EoNrU_oI
amzn-s3-demo-bucket1,photos/jpgs/objectkey4,6EqlikJJxLTsHsnbZbSRffn24_eh5Ny4
amzn-s3-demo-bucket1,photos/jpgs/newjersey/objectkey5,imHf3FAiRsvBW_EHB8GOu.NHunHO1gVs
amzn-s3-demo-bucket1,object%20key%20with%20spaces,9HkPvDaZY5MVbMhn6TMn1YTb5ArQAo3w
```

Angeben einer vorhandenen Manifestdatei

Sie können ein Manifest in einer Auftragserstellungsanforderung in einem der beiden folgenden Formate erstellen:

- Amazon S3 S3-Inventarbericht — Muss ein Amazon S3 S3-Inventarbericht CSV im -Format sein. Sie müssen die `manifest.json`-Datei angeben, die mit dem Bestandsbericht verknüpft ist. Weitere Informationen zu Bestandsberichten finden Sie unter Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory. Wenn der Inventarbericht eine Version enthältIDs, arbeitet S3 Batch Operations mit den spezifischen Objektversionen.

> **Note**
>
> - S3 Batch Operations unterstützt CSV Inventarberichte, die mit SSE - verschlüsselt sindKMS.
> - Wenn Sie ein mit SSE — verschlüsseltes Inventarberichtsmanifest einreichenKMS, muss Ihre IAM Richtlinie die Berechtigungen `"kms:Decrypt"` und `"kms:GenerateDataKey"` für das `manifest.json` Objekt und alle zugehörigen CSV Datendateien enthalten.

- CSVDatei — Jede Zeile in der Datei muss den Bucket-Namen, den Objektschlüssel und optional die Objektversion enthalten. Objektschlüssel müssen URL -kodiert sein, wie in den folgenden Beispielen gezeigt. Das Manifest muss entweder die Version IDs für alle Objekte enthalten oder die Version für alle Objekte weglassen. IDs Weitere Hinweise zum CSV Manifestformat finden Sie unter JobManifestSpecin der Amazon Simple Storage Service API Reference.

> **Note**
>
> S3 Batch Operations unterstützt keine CSV Manifestdateien, die mit SSE - verschlüsselt sindKMS.

> **Important**
>
> Wenn Sie ein manuell erstelltes Manifest und einen versionierten Bucket verwenden, empfehlen wir Ihnen, die Version IDs für die Objekte anzugeben. Wenn Sie einen Auftrag erstellen, analysiert S3-Batch-Vorgänge das gesamte Manifest, bevor Sie den Auftrag ausführen. Es wird jedoch kein „Snapshot“ des Bucket-Zustands aufgenommen.
> Da Manifeste Milliarden von Objekten enthalten können, kann die Ausführung von Aufträgen längere Zeit dauern, was sich darauf auswirken kann, für welche Version eines Objekts der Auftrag durchgeführt wird. Angenommen, Sie überschreiben ein Objekt mit einer neuen Version, während ein Auftrag ausgeführt wird, und Sie haben keine Versions-ID für dieses Objekt angegeben. In diesem Fall führt Amazon S3 den Vorgang auf der neuesten Version des Objekts durch, und nicht auf der Version, die galt, als Sie den Auftrag erstellten. Die einzige Möglichkeit, dieses Verhalten zu vermeiden, besteht darin, die Version IDs für die Objekte anzugeben, die im Manifest aufgeführt sind.

Automatisches Generieren eines Manifests

Sie können Amazon S3 Batch Operations anweisen, automatisch ein Manifest auf der Grundlage von Objektfilterkriterien zu generieren, die Sie bei der Erstellung Ihres Auftrags angeben. Diese Option ist für Batch-Replikationsaufträge verfügbar, die Sie in der Amazon S3-Konsole erstellen, oder für jeden Auftragstyp, den Sie mithilfe von AWS CLI AWS SDKs, oder Amazon S3 erstellen RESTAPI. Weitere Informationen zur Batch-Replikationen finden Sie unter Replizieren vorhandener Objekte mit Batch Replication.

Um ein Manifest automatisch zu generieren, geben Sie im Rahmen Ihrer Anforderung zur Auftragserstellung die folgenden Elemente an:

- Informationen über den Bucket, der Ihre Quellobjekte enthält, einschließlich des Bucket-Besitzers und des Amazon-Ressourcennamens (ARN)
- Informationen zur Manifestausgabe, einschließlich eines Flags zum Erstellen einer Manifestdatei, des Besitzers des Ausgabe-BucketsARN, des Präfixes, des Dateiformats und des Verschlüsselungstyps
- Optionale Kriterien zum Filtern von Objekten nach Erstellungsdatum, Schlüsselnamen, Größe und Speicherklasse. Bei Replikationsaufträgen können Sie auch Tags verwenden, um Objekte zu filtern.

Objektfilterkriterien

Um die Liste der Objekte zu filtern, die in ein automatisch generiertes Manifest aufgenommen werden sollen, können Sie die folgenden Kriterien angeben. Weitere Informationen finden Sie unter JobManifestGeneratorFilterin der Amazon S3 API S3-Referenz.

CreatedAfter

Falls angegeben, enthält das generierte Manifest nur Quell-Bucket-Objekte, die nach diesem Zeitpunkt erstellt wurden.

CreatedBefore

Falls angegeben, enthält das generierte Manifest nur Quell-Bucket-Objekte, die vor diesem Zeitpunkt erstellt wurden.

EligibleForReplication

Falls angegeben, enthält das generierte Manifest nur Objekte, wenn sie gemäß der Replikationskonfiguration im Quell-Bucket für die Replikation in Frage kommen.

KeyNameConstraint

Falls angegeben, enthält das generierte Manifest nur Quell-Bucket-Objekte, deren Objektschlüssel den für MatchAnySubstringMatchAnyPrefix, und angegebenen Zeichenketteneinschränkungen entsprechen MatchAnySuffix.

MatchAnySubstring— Falls angegeben, enthält das generierte Manifest Objekte, wenn die angegebene Zeichenfolge an einer beliebigen Stelle in der Objektschlüsselzeichenfolge vorkommt.

MatchAnyPrefix— Falls angegeben, enthält das generierte Manifest Objekte, wenn die angegebene Zeichenfolge am Anfang der Objektschlüsselzeichenfolge steht.

MatchAnySuffix— Falls angegeben, enthält das generierte Manifest Objekte, wenn die angegebene Zeichenfolge am Ende der Objektschlüsselzeichenfolge steht.

MatchAnyStorageClass

Falls angegeben, enthält das generierte Manifest nur Quell-Bucket-Objekte, die mit der angegebenen Speicherklasse gespeichert sind.

ObjectReplicationStatuses

Falls angegeben, enthält das generierte Manifest nur Quell-Bucket-Objekte, die einen der angegebenen Replikationsstatus haben.

ObjectSizeGreaterThanBytes

Falls angegeben, enthält das generierte Manifest nur Quell-Bucket-Objekte, deren Dateigröße die angegebene Byteanzahl überschreitet.

ObjectSizeLessThanBytes

Falls angegeben, enthält das generierte Manifest nur Quell-Bucket-Objekte, deren Dateigröße die angegebene Byteanzahl unterschreitet.

> **Note**
>
> Die meisten Aufträge mit automatisch generierten Manifesten können nicht geklont werden. Stapelreplikationsaufträge können geklont werden, es sei denn, sie verwenden die Manifest-Filterkriterien `KeyNameConstraint`, `MatchAnyStorageClass`, `ObjectSizeGreaterThanBytes` oder `ObjectSizeLessThanBytes`.

Die Syntax für die Angabe von Manifestkriterien hängt von der Methode ab, mit der Sie Ihren Job erstellen. Beispiele finden Sie unter Erstellen eines-Auftrags.

Erstellen eines-Auftrags

Sie können S3-Batch Operations-Jobs mithilfe der Amazon S3-Konsole AWS CLI, AWS SDKs, oder Amazon S3 erstellen RESTAPI.

Weitere Informationen zum Erstellen einer Auftragsanfrage finden Sie unter Batch-Vorgangsauftrag-Anforderungselemente.

Voraussetzungen

Bevor Sie einen Batch-Operations-Auftrag erstellen, bestätigen Sie, dass Sie die entsprechenden Berechtigungen konfiguriert haben. Weitere Informationen finden Sie unter Gewähren von Berechtigungen für Batchoperationen.

Verwenden der S3-Konsole

So erstellen Sie einen Batch-Auftrag:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen der aktuell angezeigten Datei aus AWS-Region. Wählen Sie als Nächstes die Region aus, in der Sie Ihren Job erstellen möchten.

   > **Note**
   >
   > Für Kopiervorgänge müssen Sie den Job in derselben Region wie den Ziel-Bucket erstellen. Für alle anderen Operationen müssen Sie den Job in derselben Region wie die Objekte im Manifest erstellen.

3. Wählen Sie im linken Navigationsbereich der Amazon S3 S3-Konsole Batch Operations aus.
4. Wählen Sie Job erstellen aus.
5. Sehen Sie sich AWS-Regionan, wo Sie Ihren Job erstellen möchten.
6. Wählen Sie unter Manifest format (Manifestformat) das zu verwendende Manifestobjekt aus.

   - Wenn Sie den S3-Inventarbericht wählen, geben Sie den Pfad zum manifest.json-Objekt ein, das Amazon S3 als Teil des CSV -formatierten Inventarberichts generiert hat, und optional die

Versions-ID für das Manifest-Objekt, wenn Sie eine andere Version als die neueste verwenden möchten.

- Wenn Sie möchten CSV, geben Sie den Pfad zu einem Manifest-Objekt im -Format einCSV. Das Manifestobjekt muss das in der Konsole beschriebene Format befolgen. Sie können zudem die Versions-ID des Manifestobjekts angeben, wenn Sie eine andere Version als die aktuelle verwenden möchten.

> **Note**
>
> Die Amazon-S3-Konsole unterstützt die automatische Manifestgenerierung nur für Stapelreplikationsaufträge. Wenn Sie möchten, dass Amazon S3 für alle anderen Auftragstypen automatisch ein Manifest auf der Grundlage der von Ihnen angegebenen Filterkriterien generiert, müssen Sie Ihren Job mit AWS CLI AWS SDKs, oder Amazon S3 konfigurieren RESTAPI.

7. Wählen Sie Weiter.
8. Wählen Sie unter Operation (Vorgang) den Vorgang aus, den Sie für alle Objekte in dem Manifest ausführen möchten. Geben Sie die Informationen für die ausgewählte Operation ein und wählen Sie anschließend Next (Weiter) aus.
9. Geben Sie die Informationen für Configure additional options (Zusätzliche Optionen konfigurieren) ein und wählen Sie anschließend Next (Weiter) aus.
10. Überprüfen Sie die Einstellungen unter Review (Überprüfen). Wenn Sie Änderungen vornehmen müssen, wählen Sie Previous. Wählen Sie andernfalls Auftrag erstellen.

Mit dem AWS CLI

Um Ihren Batch Operations-Job mit dem zu erstellen AWS CLI, wählen Sie eines der folgenden Beispiele aus, je nachdem, ob Sie ein vorhandenes Manifest angeben oder ein Manifest automatisch generieren.

Specify manifest

Das folgende Beispiel zeigt, wie Sie mit dem AWS CLI einen `S3PutObjectTagging` S3-Auftrag für Batch Operations erstellen, der auf Objekte reagiert, die in einer vorhandenen Manifestdatei aufgeführt sind.

So erstellen Sie einen Batch **`S3PutObjectTagging`** Operations-Job durch Angabe eines Manifests

1. Verwenden Sie die folgenden Befehle, um eine Rolle AWS Identity and Access Management (IAM) zu erstellen, und erstellen Sie anschließend eine IAM Richtlinie, um die entsprechenden Berechtigungen zuzuweisen. Die folgende Rolle und die folgende Richtlinie gewähren Amazon S3 die Berechtigung zum Hinzufügen von Objekt-Tags, was Sie benötigen, wenn Sie in einem nachfolgenden Schritt den Auftrag erstellen.

   a. Verwenden Sie den folgenden Beispielbefehl, um eine IAM Rolle zu erstellen, die Batch Operations verwenden kann. Ersetzen Sie *`S3BatchJobRole`* durch den Namen, den Sie der Rolle geben möchten, um diesen Beispielbefehl zu verwenden.

```
aws iam create-role \
 --role-name S3BatchJobRole \
 --assume-role-policy-document '{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"batchoperations.s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}'
```

   Notieren Sie sich den Amazon-Ressourcennamen (ARN) der Rolle. Sie benötigen denARN, wenn Sie einen Job erstellen.

   b. Verwenden Sie den folgenden Beispielbefehl, um eine IAM Richtlinie mit den erforderlichen Berechtigungen zu erstellen und sie der IAM Rolle zuzuordnen, die Sie im vorherigen Schritt erstellt haben. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter Gewähren von Berechtigungen für Batchoperationen.

> **Note**
>
> Batch-Operations-Aufträge, die Aktionen für Verzeichnis-Buckets ausführen, erfordern bestimmte Berechtigungen. Weitere Informationen finden Sie unter AWS Identity and Access Management (IAM) für S3 Express One Zone.

Um diesen Beispielbefehl zu verwenden, ersetzen Sie die *`user input placeholders`* wie folgt:

- *`S3BatchJobRole`*Ersetzen Sie es durch den Namen Ihrer IAM Rolle. Stellen Sie sicher, dass dieser Name mit dem Namen übereinstimmt, den Sie zuvor verwendet haben.
- *`PutObjectTaggingBatchJobPolicy`*Ersetzen Sie es durch den Namen, den Sie Ihrer IAM Richtlinie geben möchten.
- Ersetzen Sie *`amzn-s3-demo-destination-bucket`* durch den Namen des Buckets, der die Objekte enthält, auf die Sie Tags anwenden möchten.
- Ersetzen *`amzn-s-3 demo-manifest-bucket`* mit dem Namen des Buckets, der das Manifest enthält.
- Ersetzen *`amzn-s3- demo-completion-report-bucket`* mit dem Namen des Buckets, an den der Abschlussbericht gesendet werden soll.

```
aws iam put-role-policy \
  --role-name S3BatchJobRole \
  --policy-name PutObjectTaggingBatchJobPolicy \
  --policy-document '{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObjectTagging",
        "s3:PutObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
    },
    {
```

```
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-manifest-bucket",
        "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
      ]
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutObject",
        "s3:GetBucketLocation"
      ],
      "Resource":[
        "arn:aws:s3:::amzn-s3-demo-completion-report-bucket",
        "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
      ]
    }
  ]
}'
```

2. Verwenden Sie den folgenden Beispielbefehl, um einen `S3PutObjectTagging`-Auftrag zu erstellen.

   Die `manifest.csv`-Datei stellt eine Liste mit Bucket- und Objektschlüsselwerten bereit. Der Auftrag wendet die angegebenen Tags auf Objekte an, die im Manifest aufgelistet sind. Das `ETag` ist das ETag `manifest.csv` Objekt, das Sie von der Amazon S3 S3-Konsole abrufen können. Diese Anforderung enthält den `no-confirmation-required`-Parameter, damit Sie den Job ausführen können, ohne ihn mit dem `update-job-status`-Befehl bestätigen zu müssen. Weitere Informationen finden Sie unter create-jobin der AWS CLI Befehlsreferenz.

   Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen. Ersetzen *`IAM-Rolle`* mit ARN der IAM Rolle, die Sie zuvor erstellt haben.

```
aws s3control create-job \
    --region us-west-2 \
```

```
    --account-id acct-id \
    --operation '{"S3PutObjectTagging": { "TagSet": [{"Key":"keyOne",
 "Value":"ValueOne"}] }}' \
    --manifest '{"Spec":{"Format":"S3BatchOperations_CSV_20180820","Fields":
["Bucket","Key"]},"Location":{"ObjectArn":"arn:aws:s3:::amzn-s3-demo-manifest-
bucket/manifest.csv","ETag":"60e460c9d1046e73f7dde5043ac3ae85"}}' \
    --report '{"Bucket":"arn:aws:s3:::amzn-s3-demo-
completion-report-bucket","Prefix":"final-reports",
 "Format":"Report_CSV_20180820","Enabled":true,"ReportScope":"AllTasks"}' \
    --priority 42 \
    --role-arn IAM-role \
    --client-request-token $(uuidgen) \
    --description "job description" \
    --no-confirmation-required
```

Als Antwort gibt Amazon S3 eine Auftrags-ID zurück (zum Beispie, `00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c`). Sie benötigen die Auftrags-ID, um den Auftrag zu identifizieren, zu überwachen und zu ändern.

Generate manifest

Das folgende Beispiel zeigt, wie Sie einen S3-Batch-Operations–`S3DeleteObjectTagging`-Auftrag erstellen, der automatisch ein Manifest auf der Grundlage Ihrer Objektfilterkriterien generiert. Zu diesen Kriterien gehören das Erstellungsdatum, der Schlüsselname, die Größe, die Speicherklasse und die Tags.

So erstellen Sie einen Batch **`S3DeleteObjectTagging`** Operations-Job durch Generieren eines Manifests

1. Verwenden Sie die folgenden Befehle, um eine Rolle AWS Identity and Access Management (IAM) und anschließend eine IAM Richtlinie zum Zuweisen von Berechtigungen zu erstellen. Die folgende Rolle und die folgende Richtlinie gewähren Amazon S3 die Berechtigung zum Löschen von Objekt-Tags, was Sie benötigen, wenn Sie in einem nachfolgenden Schritt den Auftrag erstellen.

   a. Verwenden Sie den folgenden Beispielbefehl, um eine IAM Rolle zu erstellen, die Batch Operations verwenden kann. Ersetzen Sie *`S3BatchJobRole`* durch den Namen, den Sie der Rolle geben möchten, um diesen Beispielbefehl zu verwenden.

```
aws iam create-role \
 --role-name S3BatchJobRole \
 --assume-role-policy-document '{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"batchoperations.s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}'
```

Notieren Sie sich den Amazon-Ressourcennamen (ARN) der Rolle. Sie benötigen denARN, wenn Sie einen Job erstellen.

b. Verwenden Sie den folgenden Beispielbefehl, um eine IAM Richtlinie mit den erforderlichen Berechtigungen zu erstellen und sie der IAM Rolle zuzuordnen, die Sie im vorherigen Schritt erstellt haben. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter Gewähren von Berechtigungen für Batchoperationen.

> **Note**
>
> Batch-Operations-Aufträge, die Aktionen für Verzeichnis-Buckets ausführen, erfordern bestimmte Berechtigungen. Weitere Informationen finden Sie unter AWS Identity and Access Management (IAM) für S3 Express One Zone.

Um diesen Beispielbefehl zu verwenden, ersetzen Sie die *user input placeholders* wie folgt:

- *S3BatchJobRole*Ersetzen Sie es durch den Namen Ihrer IAM Rolle. Stellen Sie sicher, dass dieser Name mit dem Namen übereinstimmt, den Sie zuvor verwendet haben.
- *DeleteObjectTaggingBatchJobPolicy*Ersetzen Sie es durch den Namen, den Sie Ihrer IAM Richtlinie geben möchten.

- Ersetzen Sie *amzn-s3-demo-destination-bucket* durch den Namen des Buckets, der die Objekte enthält, auf die Sie Tags anwenden möchten.
- Ersetzen Sie *amzn-s3-demo-manifest-bucket* durch den Namen des Buckets, in dem das Manifest gespeichert werden soll.
- Ersetzen Sie *amzn-s3-demo-completion-report-bucket* durch den Namen des Buckets, an den der Abschlussbericht übermittelt werden soll.

```
aws iam put-role-policy \
  --role-name S3BatchJobRole \
  --policy-name DeleteObjectTaggingBatchJobPolicy \
  --policy-document '{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "s3:DeleteObjectTagging",
        "s3:DeleteObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
    },
    {
      "Effect":"Allow",
      "Action":[
        "s3:PutInventoryConfiguration"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-manifest-bucket",
        "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
      ]
    },
    {
```

```
      "Effect":"Allow",
      "Action":[
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Resource":[
        "arn:aws:s3:::amzn-s3-demo-completion-report-bucket",
        "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*",
        "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
      ]
    }
  ]
}'
```

2. Verwenden Sie den folgenden Beispielbefehl, um den `S3DeleteObjectTagging`-Auftrag zu erstellen.

   In diesem Beispiel geben die Werte im `--report`-Abschnitt den Bucket, das Präfix, das Format und den Umfang des Auftragsberichts an, der generiert wird. Der `--manifest-generator`-Abschnitt enthält Informationen über den Quell-Bucket, der die Objekte enthält, auf die der Auftrag wirkt, Informationen über die Manifest-Ausgabeliste, die für den Auftrag generiert wird, und Filterkriterien, um den Bereich der Objekte, die in das Manifest aufgenommen werden sollen, nach Erstellungsdatum, Namenseinschränkungen, Größe und Speicherklasse einzugrenzen. Der Befehl gibt auch die Priorität, IAM Rolle und des Jobs an AWS-Region.

   Weitere Informationen finden Sie unter create-jobin der AWS CLI Befehlsreferenz.

   Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen. Ersetzen *`IAM-Rolle`* mit ARN der IAM Rolle, die Sie zuvor erstellt haben.

```
aws s3control create-job \
    --account-id 012345678901 \
    --operation '{
        "S3DeleteObjectTagging": {}
    }' \
    --report '{
        "Bucket":"arn:aws:s3:::amzn-s3-demo-completion-report-bucket",
        "Prefix":"reports",
        "Format":"Report_CSV_20180820",
```

```
        "Enabled":true,
        "ReportScope":"AllTasks"
    }' \
    --manifest-generator '{
        "S3JobManifestGenerator": {
          "ExpectedBucketOwner": "012345678901",
          "SourceBucket": "arn:aws:s3:::amzn-s3-demo-source-bucket",
          "EnableManifestOutput": true,
          "ManifestOutputLocation": {
            "ExpectedManifestBucketOwner": "012345678901",
            "Bucket": "arn:aws:s3:::amzn-s3-demo-manifest-bucket",
            "ManifestPrefix": "prefix",
            "ManifestFormat": "S3InventoryReport_CSV_20211130"
          },
          "Filter": {
            "CreatedAfter": "2023-09-01",
            "CreatedBefore": "2023-10-01",
            "KeyNameConstraint": {
              "MatchAnyPrefix": [
                "prefix"
              ],
              "MatchAnySuffix": [
                "suffix"
              ]
            },
            "ObjectSizeGreaterThanBytes": 100,
            "ObjectSizeLessThanBytes": 200,
            "MatchAnyStorageClass": [
              "STANDARD",
              "STANDARD_IA"
            ]
          }
        }
      }' \
     --priority 2 \
     --role-arn IAM-role \
     --region us-east-1
```

Als Antwort gibt Amazon S3 eine Auftrags-ID zurück (zum Beispie, `00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c`). Sie benötigen diese Auftrags-ID, um den Auftrag zu identifizieren, zu überwachen oder zu ändern.

Mit dem AWS SDK for Java

Um Ihren Batch Operations-Job mit dem zu erstellen AWS SDK for Java, wählen Sie eines der folgenden Beispiele aus, je nachdem, ob Sie ein vorhandenes Manifest angeben oder ein Manifest automatisch generieren.

Specify manifest

Das folgende Beispiel zeigt, wie Sie einen S3-Batch-Operations–`S3PutObjectTagging`-Auftrag für Objekte erstellen, die in einer vorhandenen Manifestdatei aufgelistet sind. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

Example

```
package aws.example.s3control;



import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.*;

import java.util.UUID;
import java.util.ArrayList;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateJob {
    public static void main(String[] args) {
        String accountId = "Account ID";
        String iamRoleArn = "IAM Role ARN";
        String reportBucketName = "arn:aws:s3:::amzn-s3-demo-completion-report-
bucket";
        String uuid = UUID.randomUUID().toString();

        ArrayList tagSet = new ArrayList<S3Tag>();
        tagSet.add(new S3Tag().withKey("keyOne").withValue("ValueOne"));
```

```
        try {
            JobOperation jobOperation = new JobOperation()
                    .withS3PutObjectTagging(new S3SetObjectTaggingOperation()
                            .withTagSet(tagSet)
                    );

            JobManifest manifest = new JobManifest()
                    .withSpec(new JobManifestSpec()
                            .withFormat("S3BatchOperations_CSV_20180820")
                            .withFields(new String[]{
                                    "Bucket", "Key"
                            }))
                    .withLocation(new JobManifestLocation()
                            .withObjectArn("arn:aws:s3:::my_manifests/manifest.csv")
                            .withETag("60e460c9d1046e73f7dde5043ac3ae85"));
            JobReport jobReport = new JobReport()
                    .withBucket(reportBucketName)
                    .withPrefix("reports")
                    .withFormat("Report_CSV_20180820")
                    .withEnabled(true)
                    .withReportScope("AllTasks");

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.createJob(new CreateJobRequest()
                    .withAccountId(accountId)
                    .withOperation(jobOperation)
                    .withManifest(manifest)
                    .withReport(jobReport)
                    .withPriority(42)
                    .withRoleArn(iamRoleArn)
                    .withClientRequestToken(uuid)
                    .withDescription("job description")
                    .withConfirmationRequired(false)
            );

        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
```

```
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Generate manifest

Das folgende Beispiel zeigt, wie Sie einen S3-Batch-Operations–`s3PutObjectCopy`-Auftrag erstellen, der automatisch ein Manifest auf der Grundlage Ihrer Objektfilterkriterien, einschließlich des Erstellungsdatums, des Schlüsselnamens und der Größe, generiert. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

Example

```
        package aws.example.s3control;

    import com.amazonaws.AmazonServiceException;
    import com.amazonaws.SdkClientException;
    import com.amazonaws.auth.profile.ProfileCredentialsProvider;
    import com.amazonaws.services.s3control.AWSS3Control;
    import com.amazonaws.services.s3control.AWSS3ControlClient;
    import com.amazonaws.services.s3control.model.CreateJobRequest;
    import com.amazonaws.services.s3control.model.CreateJobResult;
    import com.amazonaws.services.s3control.model.JobManifestGenerator;
    import com.amazonaws.services.s3control.model.JobManifestGeneratorFilter;
    import com.amazonaws.services.s3control.model.JobOperation;
    import com.amazonaws.services.s3control.model.JobReport;
    import com.amazonaws.services.s3control.model.KeyNameConstraint;
    import com.amazonaws.services.s3control.model.S3JobManifestGenerator;
    import com.amazonaws.services.s3control.model.S3ManifestOutputLocation;
    import com.amazonaws.services.s3control.model.S3SetObjectTaggingOperation;
    import com.amazonaws.services.s3control.model.S3Tag;

    import java.time.Instant;
    import java.util.Date;
    import java.util.UUID;
    import java.util.ArrayList;

    import static com.amazonaws.regions.Regions.US_WEST_2;

    public class test {
```

```
        public static void main(String[] args) {
            String accountId = "012345678901";
            String iamRoleArn = "arn:aws:iam::012345678901:role/ROLE";
            String sourceBucketName = "arn:aws:s3:::amzn-s3-demo-source-bucket";
            String reportBucketName = "arn:aws:s3:::amzn-s3-demo-completion-report-
bucket";
            String manifestOutputBucketName = "arn:aws:s3:::amzn-s3-demo-manifest-
bucket";
            String uuid = UUID.randomUUID().toString();
            long minimumObjectSize = 100L;

            ArrayList<S3Tag> tagSet = new ArrayList<>();
            tagSet.add(new S3Tag().withKey("keyOne").withValue("ValueOne"));

            ArrayList<String> prefixes = new ArrayList<>();
            prefixes.add("s3KeyStartsWith");

            try {
                JobOperation jobOperation = new JobOperation()
                        .withS3PutObjectTagging(new S3SetObjectTaggingOperation()
                                .withTagSet(tagSet)
                        );
                S3ManifestOutputLocation manifestOutputLocation = new
 S3ManifestOutputLocation()
                        .withBucket(manifestOutputBucketName)
                        .withManifestPrefix("manifests")
                        .withExpectedManifestBucketOwner(accountId)
                        .withManifestFormat("S3InventoryReport_CSV_20211130");

                JobManifestGeneratorFilter jobManifestGeneratorFilter = new
 JobManifestGeneratorFilter()
                        .withEligibleForReplication(true)
                        .withKeyNameConstraint(
                                new KeyNameConstraint()
                                        .withMatchAnyPrefix(prefixes))
                        .withCreatedBefore(Date.from(Instant.now()))
                        .withObjectSizeGreaterThanBytes(minimumObjectSize);

                S3JobManifestGenerator s3JobManifestGenerator = new
 S3JobManifestGenerator()
                        .withEnableManifestOutput(true)
                        .withManifestOutputLocation(manifestOutputLocation)
                        .withFilter(jobManifestGeneratorFilter)
                        .withSourceBucket(sourceBucketName);
```

```
                JobManifestGenerator jobManifestGenerator = new
JobManifestGenerator()
                        .withS3JobManifestGenerator(s3JobManifestGenerator);

                JobReport jobReport = new JobReport()
                        .withBucket(reportBucketName)
                        .withPrefix("reports")
                        .withFormat("Report_CSV_20180820")
                        .withEnabled(true)
                        .withReportScope("AllTasks");

                AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                        .withCredentials(new ProfileCredentialsProvider())
                        .withRegion(US_WEST_2)
                        .build();

                CreateJobResult createJobResult = s3ControlClient.createJob(new
CreateJobRequest()
                        .withAccountId(accountId)
                        .withOperation(jobOperation)
                        .withManifestGenerator(jobManifestGenerator)
                        .withReport(jobReport)
                        .withPriority(42)
                        .withRoleArn(iamRoleArn)
                        .withClientRequestToken(uuid)
                        .withDescription("job description")
                        .withConfirmationRequired(true)
                );

                System.out.println("Created job " + createJobResult.getJobId());

            } catch (AmazonServiceException e) {
                // The call was transmitted successfully, but Amazon S3 couldn't
process
                // it and returned an error response.
                e.printStackTrace();
            } catch (SdkClientException e) {
                // Amazon S3 couldn't be contacted for a response, or the client
                // couldn't parse the response from Amazon S3.
                e.printStackTrace();
            }
        }
```

```
}
```

Verwenden Sie den REST API

Sie können den verwenden RESTAPI, um einen Batch Operations-Job zu erstellen. Weitere Informationen finden Sie unter CreateJobin der Amazon Simple Storage Service API Reference.

Auftragsantworten

Wenn die `CreateJob`-Anforderung erfolgreich ist, gibt Amazon S3 eine Auftrags-ID zurück. Die Auftrags-ID ist eine eindeutige Kennung, die Amazon S3 automatisch erstellt. Anhand dieser ID können Sie Ihren Batch-Vorgangsauftrag identifizieren und dessen Status überwachen.

Wenn Sie einen Job über, oder erstellen AWS CLI AWS SDKs, können Sie S3 Batch Operations so einrichten RESTAPI, dass die Verarbeitung des Jobs automatisch beginnt. Der Auftrag wird ausgeführt, sobald er bereit ist, und muss nicht hinter Aufträgen mit höherer Priorität warten.

Wenn Sie einen Auftrag über die Amazon-S3-Konsole erstellen, müssen Sie die Auftragsdetails überprüfen und bestätigen, dass Sie ihn ausführen möchten, bevor Batch Operations mit seiner Verarbeitung beginnen kann. Wenn ein Auftrag den Status „Ausgesetzt“ mehr als 30 Tage behält, schlägt er fehl.

Von S3 Batch-Vorgänge unterstützte Vorgänge

Sie können S3-Batch-Vorgänge verwenden, um umfangreiche Batch-Vorgänge für Amazon S3-Objekte durchzuführen. S3-Batch-Vorgänge kann eine einzelne Operation für Listen von Amazon S3-Objekten durchführen, die Sie angeben. Ein einziger Auftrag kann eine festgelegte Operation auf Milliarden von Objekten mit mehreren Exabytes an Daten durchführen. Amazon S3 verfolgt den Fortschritt, versendet Benachrichtigungen und speichert einen detaillierten Abschlussbericht zu allen Aktionen. So profitieren Sie von einer vollständig verwalteten, prüfbaren und serverlosen Umgebung. Sie können S3 Batch Operations über die Amazon S3-Konsole AWS CLI, AWS SDKs, oder Amazon S3 verwenden RESTAPI.

S3-Batchoperationen unterstützt die folgenden Operationen:

Kopieren von Objekten

Sie können Amazon S3 Batch Operations verwenden, um umfangreiche Batch-Operationen an Amazon S3 S3-Objekten durchzuführen. Der Kopiervorgang Batch Operations kopiert jedes Objekt,

das im Manifest angegeben ist. Sie können Objekte in einen Bucket in derselben AWS-Region oder in einen Bucket in einer anderen Region kopieren. S3-Batch-Vorgänge unterstützt die meisten über Amazon S3 verfügbaren Optionen zum Kopieren von Objekten. Zu diesen Optionen gehören das Festlegen von Objekt-Metadaten, das Festlegen von Berechtigungen sowie das Ändern der Speicherklasse eines Objekts.

Sie können den Vorgang Kopieren auch verwenden, um vorhandene unverschlüsselte Objekte zu kopieren und sie wie verschlüsselte Objekte in denselben Bucket zurückzuschreiben. Weitere Informationen finden Sie unter Encrypting Objects with Amazon S3 Batch Operations (Verschlüsseln von Objekten mit Amazon S3 Batch Operations).

Wenn Sie Objekte kopieren, können Sie den Prüfsummenalgorithmus zur Berechnung der Prüfsumme des Objekts ändern. Wenn Objekte keine zusätzliche berechnete Prüfsumme haben, können Sie eine hinzufügen, indem Sie den von Amazon S3 zu verwendenden Prüfsummenalgorithmus angeben. Weitere Informationen finden Sie unter Überprüfung der Objektintegrität.

Weitere Informationen zum Kopieren von Objekten in Amazon S3 und zu den erforderlichen und optionalen Parametern finden Sie Objekte kopieren, verschieben und umbenennen in diesem Handbuch und CopyObjectin der Amazon Simple Storage Service API Reference.

Beschränkungen und Einschränkungen

Wenn Sie den Vorgang Batch Operations Copy verwenden, gelten die folgenden Einschränkungen und Beschränkungen:

- Alle Quellobjekte müssen sich in einem Bucket befinden.
- Alle Zielobjekte müssen sich in einem Bucket befinden.
- Sie benötigen Lese-Berechtigungen für den Quell-Bucket und Schreib-Berechtigungen für den Ziel-Bucket.
- Die maximale Größe eines Objekts, das kopiert werden soll, beträgt 5 GB.
- Wenn Sie versuchen, Objekte aus den Klassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive in die Speicherklasse S3 Standard zu kopieren, müssen Sie diese Objekte zuerst wiederherstellen. Weitere Informationen finden Sie unter Wiederherstellen eines archivierten Objekts.
- Sie müssen Ihre Batch Operations Copy-Jobs in der Zielregion erstellen, also der Region, in die Sie die Objekte kopieren möchten.

- Alle `CopyObject` Optionen werden unterstützt, mit Ausnahme der bedingten Prüfungen von Entitäts-Tags (ETags) und der serverseitigen Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C).
- Wenn der Ziel-Bucket keine Version hat, überschreiben Sie alle Objekte, die dieselben Schlüsselnamen haben.
- Objekte werden nicht unbedingt in derselben Reihenfolge kopiert, in der sie im Manifest erscheinen. Wenn es bei versionierten Buckets wichtig ist, die Reihenfolge der aktuellen oder nicht aktuellen Versionen beizubehalten, kopieren Sie zuerst alle Versionen, die nicht aktuell sind. Kopieren Sie dann, nachdem der erste Auftrag abgeschlossen ist, die aktuellen Versionen in einen nachfolgenden Auftrag.
- Das Kopieren von Objekten in die Klasse Reduced Redundancy Storage (RRS) wird nicht unterstützt.

Kopieren von Objekten mit S3 BatchVorgänge

Sie können Amazon S3 Batch Operations verwenden, um umfangreiche Batch-Operationen an Amazon S3 S3-Objekten durchzuführen. Sie können S3 Batch Operations verwenden, um einen Copy (`CopyObject`) -Job zu erstellen, um Objekte innerhalb desselben Kontos oder in ein anderes Zielkonto zu kopieren.

Die folgenden Beispiele zeigen, wie Sie ein Manifest speichern und verwenden, das sich in einem anderen Konto befindet. Das erste Beispiel zeigt, wie Sie Amazon S3 Inventory verwenden können, um den Inventarbericht zur Verwendung bei der Auftragserstellung an das Zielkonto zu senden. Das zweite Beispiel zeigt, wie ein Manifest mit kommagetrennten Werten (CSV) im Quell- oder Zielkonto verwendet wird. Das dritte Beispiel zeigt, wie der Copy-Vorgang verwendet wird, um S3-Bucket-Keys für vorhandene Objekte zu aktivieren, die mithilfe serverseitiger Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-) verschlüsselt wurden. KMS

Beispiele für Kopieroperation

- Verwenden eines Inventarberichts zum Kopieren von Objekten AWS-Konten
- Verwenden eines CSV Manifests zum Kopieren von Objekten AWS-Konten
- Verwenden von Batch-Operationen zur Aktivierung von S3-Bucket-Keys für SSE - KMS

Verwenden eines Inventarberichts zum Kopieren von Objekten AWS-Konten

Sie können Amazon S3 Batch Operations verwenden, um umfangreiche Batch-Operationen an Amazon S3 S3-Objekten durchzuführen. Sie können S3 Batch Operations verwenden, um einen Copy (`CopyObject`) -Job zu erstellen, um Objekte innerhalb desselben Kontos oder in ein anderes Zielkonto zu kopieren.

Sie können Amazon S3 Inventory verwenden, um einen Inventarbericht zu erstellen und den Bericht verwenden, um eine Liste (Manifest) von Objekten zu erstellen, die mit S3 Batch Operations kopiert werden sollen. Weitere Informationen zur Verwendung eines CSV Manifests im Quell- oder Zielkonto finden Sie unterthe section called “Verwenden eines CSV Manifests zum Kopieren von Objekten AWS-Konten”.

Der Amazon S3 Inventory generiert Bestandslisten der Objekte in einem Bucket. Die resultierende Liste wird in einer Ausgabedatei veröffentlicht. Der Bucket, für den die Bestandsliste erstellt wird, wird als Quell-Bucket bezeichnet, der Bucket, in dem die Bestandsberichtsdatei gespeichert wird, dagegen als Ziel-Bucket.

Der Amazon S3 S3-Inventarbericht kann so konfiguriert werden, dass er an einen anderen gesendet wird AWS-Konto. Dadurch kann S3 Batch Operations den Inventarbericht lesen, wenn der Job im Zielkonto erstellt wird.

Weitere Informationen zu Quell- und Ziel-Buckets für Amazon S3 Inventory finden Sie unter Quell- und Ziel-Buckets.

Der einfachste Weg, ein Inventar einzurichten, ist die Verwendung der Amazon S3-Konsole, aber Sie können auch Amazon S3 RESTAPI, AWS Command Line Interface (AWS CLI) oder verwenden AWS SDKs.

Die folgende Konsolenprozedur enthält die allgemeinen Schritte zum Einrichten von Berechtigungen für einen S3-Batchoperations-Auftrag. Bei diesem Verfahren kopieren Sie Objekte aus einem Quellkonto in ein Zielkonto, wobei der Bestandsbericht im Zielkonto gespeichert wird.

So richten Sie einen Amazon S3 Inventory für Quell- und Ziel-Buckets ein, die zu unterschiedlichen Konten gehören

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.

3. Entscheiden Sie sich für einen Ziel-Manifest-Bucket (oder erstellen Sie ihn), in dem der Inventarbericht gespeichert werden soll. In diesem Verfahren ist das Zielkonto das Konto, zu dem der Ziel-Manifest-Bucket sowie der Bucket gehören, in den die Objekte kopiert werden.
4. Konfigurieren Sie einen Inventarbericht für einen Quell-Bucket. Informationen zur Verwendung der Konsole zum Konfigurieren einer Bestandsliste oder zum Verschlüsseln einer Bestandsdatei finden Sie unter Konfigurieren von Amazon S3 Inventory.

   Wenn Sie den Inventarbericht konfigurieren, geben Sie den Ziel-Bucket an, in dem die Liste gespeichert werden soll. Der Bestandsbericht für den Quell-Bucket wird im Ziel-Bucket veröffentlicht. In diesem Verfahren ist das Quellkonto das Konto, zu dem der Quell-Bucket gehört.

   Wählen Sie CSVdas Ausgabeformat aus.

   Wenn Sie Informationen für den Ziel-Bucket eingeben, wählen Sie Buckets in another account (Buckets in einem anderen Konto). Geben Sie dann den Namen des Ziel-Manifest-Buckets ein. Optional können Sie die Konto-ID des Zielkontos eingeben.

   Nachdem die Bestands-Konfiguration gespeichert wurde, zeigt die Konsole eine Meldung wie die folgende an:

   Amazon S3 could not create a bucket policy on the destination bucket. Ask the destination bucket owner to add the following bucket policy to allow Amazon S3 to place data in that bucket. (Amazon S3 konnte keine Bucket-Richtlinie für den Ziel-Bucket erstellen.Bitten Sie den Eigentümer des Ziel-Buckets, die folgende Bucket-Richtlinie hinzuzufügen, damit Amazon S3 Daten in diesen Bucket einfügen kann.

   Die Konsole zeigt dann eine Bucket-Richtlinie an, die Sie für den Ziel-Bucket verwenden können.
5. Kopieren Sie die in der Konsole angezeigte Ziel-Bucket-Richtlinie.
6. Fügen Sie die kopierte Bucket-Richtlinie im Zielkonto dem Ziel-Manifest-Bucket hinzu, in dem der Bestandsbericht gespeichert wird.
7. Erstellen Sie eine Rolle im Zielkonto, die auf der S3-BatchVorgängevertrauensrichtlinie basiert. Weitere Informationen zu dieser Vertrauensrichtlinie finden Sie unterVertrauensrichtlinie.

   Weitere Informationen zum Erstellen einer Rolle finden Sie im IAMBenutzerhandbuch unter Erstellen einer Rolle zum Delegieren von Berechtigungen AWS-Service an eine.

Geben Sie einen Namen für die Rolle ein (in der folgenden Beispielrolle wird der Name verwendet) *BatchOperationsDestinationRoleCOPY*). Wählen Sie den S3-Dienst und dann den Anwendungsfall S3 Batch Operations aus, der die Vertrauensrichtlinie auf die Rolle anwendet.

Wählen Sie dann Create policy (Richtlinie erstellen) aus, um der Rolle die folgende Richtlinie anzufügen. Um diese Richtlinie zu verwenden, ersetzen Sie den *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperationsDestinationObjectCOPY",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:PutObjectVersionAcl",
        "s3:PutObjectAcl",
        "s3:PutObjectVersionTagging",
        "s3:PutObjectTagging",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-destination-bucket/*",
        "arn:aws:s3:::amzn-s3-demo-source-bucket/*",
        "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
      ]
    }
  ]
}
```

Die Rolle verwendet die Richtlinie, um die Berechtigung `batchoperations.s3.amazonaws.com` zum Lesen des Manifests im Ziel-Bucket zu gewähren. Es gewährt auch Berechtigungen für `GET` Objekte, Zugriffskontrolllisten (ACLs),

Tags und Versionen im Quellobjekt-Bucket. Außerdem gewährt es Berechtigungen für PUT ObjekteACLs, Tags und Versionen für den Zielobjekt-Bucket.

8. Erstellen Sie im Quellkonto eine Bucket-Richtlinie für den Quell-Bucket, die der Rolle, die Sie im vorherigen Schritt erstellt haben, Berechtigungen für GET ObjekteACLs, Tags und Versionen im Quell-Bucket gewährt. Dieser Schritt ermöglicht es S3-BatchVorgänge, Objekte über die vertrauenswürdige Rolle aus dem Quell-Bucket abzurufen.

   Das folgende Beispiel zeigt eine Bucket-Richtlinie für das Quellkonto. Um diese Richtlinie zu verwenden, ersetzen Sie die *`Platzhalter für Benutzereingaben`* mit Ihren eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperationsSourceObjectCOPY",
      "Effect": "Allow",
      "Principal": {
        "AWS":
 "arn:aws:iam::DestinationAccountNumber:role/BatchOperationsDestinationRoleCOPY"
      },
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
    }
  ]
}
```

9. Sobald der Inventarbericht verfügbar ist, erstellen Sie einen S3 Batch Operations Copy (`CopyObject`) -Job im Zielkonto und wählen Sie den Inventarbericht aus dem Zielmanifest-Bucket aus. Sie benötigen den ARN für die IAM Rolle, die Sie im Zielkonto erstellt haben.

   Allgemeine Informationen zum Erstellen eines Auftrags finden Sie unter Erstellen eines S3-Batch-Vorgangsauftrags.

Informationen zum Erstellen eines Jobs mithilfe der Konsole finden Sie unterErstellen eines S3-Batch-Vorgangsauftrags.

Verwenden eines CSV Manifests zum Kopieren von Objekten AWS-Konten

Sie können Amazon S3 Batch Operations verwenden, um umfangreiche Batch-Operationen an Amazon S3 S3-Objekten durchzuführen. Sie können S3 Batch Operations verwenden, um einen Copy (`CopyObject`) -Job zu erstellen, um Objekte innerhalb desselben Kontos oder in ein anderes Zielkonto zu kopieren.

Sie können ein CSV Manifest verwenden, das im Quellkonto gespeichert ist, um Objekte AWS-Konten mit S3 Batch Operations zu kopieren. Informationen zur Verwendung eines S3-Inventarberichts als Manifest finden Sie unterthe section called “Verwenden eines Inventarberichts zum Kopieren von Objekten AWS-Konten”.

Ein Beispiel für das CSV Format für Manifestdateien finden Sie unterthe section called “Erstellen einer Manifestdatei”.

Das folgende Verfahren zeigt, wie Sie Berechtigungen einrichten, wenn Sie einen S3 Batch Operations-Job verwenden, um Objekte von einem Quellkonto in ein Zielkonto mit einer CSV Manifestdatei zu kopieren, die im Quellkonto gespeichert ist.

So verwenden Sie ein CSV Manifest zum Kopieren von Objekten AWS-Konten

1. Erstellen Sie eine AWS Identity and Access Management (IAM) -Rolle im Zielkonto, die auf der Vertrauensrichtlinie von S3 Batch Operations basiert. In diesem Verfahren ist das Zielkonto das Konto, in das die Objekte kopiert werden.

   Weitere Informationen zur Vertrauensrichtlinie finden Sie unter Vertrauensrichtlinie.

   Weitere Informationen zum Erstellen einer Rolle finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an eine AWS-Service im IAMBenutzerhandbuch.

   Wenn Sie die Rolle mithilfe der Konsole erstellen, geben Sie einen Namen für die Rolle ein (die folgende Beispielrolle verwendet den Namen*`BatchOperationsDestinationRoleCOPY`*). Wählen Sie den S3-Dienst und dann den Anwendungsfall S3 Batch Operations aus, der die Vertrauensrichtlinie auf die Rolle anwendet.

Wählen Sie dann Create policy (Richtlinie erstellen) aus, um der Rolle die folgende Richtlinie anzufügen. Um diese Richtlinie zu verwenden, ersetzen Sie den *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperationsDestinationObjectCOPY",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:PutObjectVersionAcl",
        "s3:PutObjectAcl",
        "s3:PutObjectVersionTagging",
        "s3:PutObjectTagging",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": [
        "arn:aws: s3:::amzn-s3-demo-destination-bucket/*",
        "arn:aws: s3:::amzn-s3-demo-source-bucket/*",
        "arn:aws: s3:::amzn-s3-demo-manifest-bucket/*"
      ]
    }
  ]
}
```

Mit der Richtlinie gewährt die Rolle die Berechtigung `batchoperations.s3.amazonaws.com` zum Lesen des Manifests im Quell-Manifest-Bucket. Es gewährt Berechtigungen für `GET` Objekte, Zugriffskontrolllisten (ACLs), Tags und Versionen im Quellobjekt-Bucket. Es gewährt auch Berechtigungen für `PUT` ObjekteACLs, Tags und Versionen für den Zielobjekt-Bucket.

2. Erstellen Sie im Quellkonto eine Bucket-Richtlinie für den Bucket, der das Manifest enthält, um der Rolle, die Sie im vorherigen Schritt erstellt haben, Berechtigungen für `GET` Objekte und Versionen im Quellmanifest-Bucket zu gewähren.

In diesem Schritt kann S3 Batch Operations das Manifest mithilfe der vertrauenswürdigen Rolle lesen. Weisen Sie die Bucket-Richtlinie dem Bucket zu, in dem sich das Manifest befindet.

Das folgende Beispiel zeigt eine Bucket-Richtlinie, die dem Quell-Manifest-Bucket zugewiesen werden soll. Um diese Richtlinie zu verwenden, ersetzen Sie *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperationsSourceManfiestRead",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws: iam::DestinationAccountNumber:user/ConsoleUserCreatingJob",
          "arn:aws:
 iam::DestinationAccountNumber:role/BatchOperationsDestinationRoleCOPY"
        ]
      },
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "arn:aws: s3:::amzn-s3-demo-manifest-bucket/*"
    }
  ]
}
```

Diese Richtlinie gewährt außerdem die erforderlichen Berechtigungen, damit ein Konsolenbenutzer, der einen Auftrag im Zielkonto erstellt, diese Berechtigungen über dieselbe Bucket-Richtlinie auch für den Ziel-Manifest-Bucket erhält.

3. Erstellen Sie im Quellkonto eine Bucket-Richtlinie für den Quell-Bucket, die der Rolle, die Sie erstellt haben, Berechtigungen für GET ObjekteACLs, Tags und Versionen im Quellobjekt-Bucket gewährt. S3-Batch-Vorgänge kann dann Objekte über die vertrauenswürdige Rolle aus dem Quell-Bucket abrufen.

   Es folgt ein Beispiel für die Bucket-Richtlinie für den Bucket, der die Quellobjekte enthält. Um diese Richtlinie zu verwenden, ersetzen Sie die *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowBatchOperationsSourceObjectCOPY",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:
 iam::DestinationAccountNumber:role/BatchOperationsDestinationRoleCOPY"
      },
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": "arn:aws: s3:::amzn-s3-demo-source-bucket/*"
    }
  ]
}
```

4. Erstellen Sie einen S3-Batchoperations-Auftrag im Zielkonto. Sie benötigen den Amazon-Ressourcennamen (ARN) für die Rolle, die Sie im Zielkonto erstellt haben. Weitere Informationen zum Erstellen eines Auftrags finden Sie unter Erstellen eines S3-Batch-Vorgangsauftrags.

Verwenden von Batch-Operationen zur Aktivierung von S3-Bucket-Keys für SSE - KMS

S3-Bucket-Keys reduzieren die Kosten der serverseitigen Verschlüsselung mit AWS Key Management Service (AWS KMS) (SSE-KMS), indem sie den Anforderungsverkehr von Amazon S3 zu AWS KMS verringern. Weitere Informationen erhalten Sie unter Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys und Konfigurieren Sie Ihren Bucket so, dass er einen S3-Bucket-Key mit SSE - KMS für neue Objekte verwendet. Wenn Sie einen `CopyObject` Vorgang mithilfe von, oder AWS CLI ausführen REST API AWS SDKs, können Sie einen S3-Bucket-Key auf Objektebene aktivieren oder deaktivieren, indem Sie den `x-amz-server-side-encryption-bucket-key-enabled` Anforderungsheader mit dem `false` Wert `true` oder hinzufügen.

Wenn Sie einen S3-Bucket Key für ein Objekt mithilfe eines `CopyObject` Vorgangs konfigurieren, aktualisiert Amazon S3 nur die Einstellungen für dieses Objekt. Die S3-Bucket Key-Einstellungen

für den Ziel-Bucket ändern sich nicht. Wenn Sie eine `CopyObject` Anfrage für ein AWS KMS verschlüsseltes Objekt an einen Bucket senden, für den S3 Bucket Keys aktiviert sind, verwendet Ihr Vorgang auf Objektebene automatisch S3 Bucket Keys, sofern Sie die Schlüssel nicht im Anforderungsheader deaktivieren. Wenn Sie keinen S3-Bucket-Schlüssel für Ihr Objekt angeben, wendet Amazon S3 die S3-Bucket-Schlüssel-Einstellungen für den Ziel-Bucket auf das Objekt an.

Um Ihre vorhandenen Amazon S3 S3-Objekte zu verschlüsseln, können Sie S3 Batch Operations verwenden. Mit der Operation Batch Operations Copy können Sie vorhandenen nicht verschlüsselte Objekte kopieren und die neuen verschlüsselten Objekte in denselben Bucket schreiben. Weitere Informationen finden Sie im AWS Storage-Blog unter Objekte mit Amazon S3 Batch Operations verschlüsseln.

Im folgenden Beispiel verwenden Sie den Vorgang Batch Operations Copy, um S3 Bucket Keys für bestehende Objekte zu aktivieren. Weitere Informationen finden Sie unter the section called “Konfigurieren eines S3-Bucket-Schlüssels für ein Objekt”.

Themen

- Überlegungen zur Verwendung von S3 Batch Operations zur Verschlüsselung von Objekten mit aktivierten S3-Bucket-Keys
- Voraussetzungen
- Schritt 1: Abrufen der Liste von Objekten mithilfe von Amazon S3 Inventory
- Schritt 2: Filtern Sie Ihre Objektliste mit S3 Select
- Schritt 3: Einrichten und Ausführen des Auftrags für S3-BatchVorgänge

Überlegungen zur Verwendung von S3 Batch Operations zur Verschlüsselung von Objekten mit aktivierten S3-Bucket-Keys

Beachten Sie die folgenden Probleme, wenn Sie S3 Batch Operations verwenden, um Objekte mit aktivierten S3-Bucket-Keys zu verschlüsseln:

- Die Kosten für Aufträge, Objekte und Anforderungen in S3 Batch Operations werden Ihnen zusätzlich zu allen mit der Operation, die S3 Batch Operations in Ihrem Namen ausführt, verbundenen Kosten berechnet, einschließlich Datenübertragungen, Anforderungen und anderen Gebühren. Weitere Informationen finden Sie unter Amazon S3 – Preise.
- Wenn Sie einen versionierten Bucket verwenden, erstellt jeder ausgeführte S3-Batchoperations-Auftrag neue verschlüsselte Versionen Ihrer Objekte. Darüber hinaus werden die vorherigen Versionen ohne konfigurierten S3-Bucket-Schlüssel beibehalten. Richten Sie zum Löschen der

alten Versionen eine S3-Lebenszyklus-Ablauf-Richtlinie für nicht aktuelle Versionen ein, wie unter Elemente der Lebenszyklus-Konfiguration beschrieben.
- Mit dem Kopiervorgang werden neue Objekte mit neuen Erstellungsdaten erstellt, die sich auf Lebenszyklus-Aktionen wie die Archivierung auswirken können. Wenn Sie alle Objekte in Ihrem Bucket kopieren, weisen alle neuen Kopien identische oder ähnliche Erstellungsdaten auf. Um diese Objekte weiter zu identifizieren und unterschiedliche Lebenszyklus-Regeln für verschiedene Daten-Teilmengen zu erstellen, sollten Sie Objekt-Markierungen verwenden.

Voraussetzungen

Bevor Sie Ihre Objekte für die Verwendung eines S3-Bucket Keys konfigurieren, sollten Sie Folgendes überprüfenÄnderungen, die Sie vor dem Aktivieren eines S3-Bucket-Schlüssels beachten sollten.

Um dieses Beispiel verwenden zu können, benötigen Sie einen AWS-Konto und mindestens einen S3-Bucket für Ihre Arbeitsdateien und verschlüsselten Ergebnisse. Möglicherweise finden Sie auch vieles aus der vorhandenen Dokumentation zu S3 Batch Operations nützlich, einschließlich der folgenden Themen:

- Grundlagen von S3-BatchVorgänge
- Erstellen eines S3-Batch-Vorgangsauftrags
- Von S3 Batch-Vorgänge unterstützte Vorgänge
- Verwalten von S3-Batch-Vorgangsaufträgen

Schritt 1: Abrufen der Liste von Objekten mithilfe von Amazon S3 Inventory

Geben Sie zunächst den S3-Bucket an, der die zu verschlüsselnden Objekte enthält, und rufen Sie eine Liste des Inhalts ab. Ein Amazon S3-Bestandsbericht ist die bequemste und kostengünstigste Methode, dies zu tun. Der Bericht enthält die Liste der Objekte in einem Bucket zusammen mit den zugehörigen Metadaten. In diesem Schritt ist der Quell-Bucket der inventarisierte Bucket und der Ziel-Bucket der Bucket, in dem Sie die Inventarberichtsdatei speichern. Weitere Informationen zu Quell- und Ziel-Buckets für Amazon S3 Inventory finden Sie unter Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory.

Die einfachste Möglichkeit zum Einrichten eines Bestands ist mit der AWS Management Console. Sie können aber auch RESTAPI, AWS Command Line Interface (AWS CLI) oder verwenden. AWS SDKs Bevor Sie diese Schritte ausführen, müssen Sie sich bei der Konsole anmelden und

die Amazon S3 S3-Konsole unter öffnen https://console.aws.amazon.com/s3/. Wenn Fehler mit Berechtigung verweigert auftreten, fügen Sie eine Bucket-Richtlinie zu Ihrem Ziel-Bucket hinzu. Weitere Informationen finden Sie unter Gewähren von Berechtigungen für S3 Inventory und S3 Analytics.

Abrufen der Liste von Objekten mithilfe von S3 Inventory

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets und wählen Sie einen Bucket aus, der zu verschlüsselnde Objekte enthält.
3. Navigieren Sie im Tab Verwaltung zu dem Abschnitt Inventory configurationes (Bestands-Konfigurationen), und wählen Sie Create inventory configuration (Bestands-Konfiguration erstellen).
4. Geben Sie Ihrem neuen Bestand einen Namen, geben Sie den Namen des Ziel-S3-Buckets ein und erstellen Sie optional ein Zielpräfix für Amazon S3, um Objekte in diesem Bucket zuzuweisen.
5. Wählen Sie als Ausgabeformat. CSV
6. (Optional) Wählen Sie im Abschnitt Zusätzliche Felder – optional die Option Verschlüsselung und alle anderen für Sie interessanten Listenfelder aus. Legen Sie die Häufigkeit für die Berichterstattung auf Daily (Täglich), damit der erste Bericht früher an Ihren Bucket geliefert wird.
7. Wählen Sie Save (Speichern) aus, um die Konfiguration zu speichern.

Amazon S3 benötigt bis zu 48 Stunden, den ersten Bericht bereitzustellen. Kehren Sie daher zurück, wenn der erste Bericht eingeht. Nachdem Sie Ihren ersten Bericht erhalten haben, fahren Sie mit dem nächsten Schritt fort, um den Inhalt Ihres S3-Inventarberichts zu filtern. Wenn Sie keine Bestandsberichte für diesen Bucket mehr erhalten möchten, löschen Sie Ihre S3-Bestands-Konfiguration. Andernfalls liefert Amazon S3 weiterhin Berichte nach einem täglichen oder wöchentlichen Zeitplan.

Eine Inventarliste ist keine einheitliche point-in-time Ansicht aller Objekte. Bestandslisten sind fortlaufende Snapshots von Bucket-Elementen, die letztendlich konsistent sind (d. h. die Liste enthält möglicherweise keine vor kurzem hinzugefügten oder gelöschten Objekte). Die Kombination von S3 Inventory und S3-Batch-Vorgänge funktioniert am besten, wenn Sie mit statischen Objekten oder mit einem Objektsatz arbeiten, den Sie vor zwei oder mehr Tagen erstellt haben. Um mit neueren Daten zu arbeiten, verwenden Sie den ListObjectsV2(GETBucket-) API Operation, um Ihre Objektliste

manuell zu erstellen. Falls erforderlich, wiederholen Sie den Vorgang für die nächsten Tage oder bis Ihr Inventarbericht den gewünschten Status für alle Objekte anzeigt.

Schritt 2: Filtern Sie Ihre Objektliste mit S3 Select

Nachdem Sie Ihren S3-Inventarbericht erhalten haben, können Sie den Inhalt des Berichts so filtern, dass nur die Objekte aufgelistet werden, die bei aktivierten S3-Bucket-Keys nicht verschlüsselt sind. Wenn Sie möchten, dass alle Objekte Ihres Buckets mit aktivierten S3-Bucket-Keys verschlüsselt werden, können Sie diesen Schritt ignorieren. Wenn Sie Ihren S3-Inventarbericht zu diesem Zeitpunkt filtern, sparen Sie jedoch Zeit und Kosten für die erneute Verschlüsselung von Objekten, die Sie zuvor mit aktivierten S3-Bucket-Keys verschlüsselt haben.

In den folgenden Schritten wird zwar gezeigt, wie Sie mithilfe von Amazon S3 Select filtern, Sie können jedoch auch Amazon Athena verwenden. Um zu entscheiden, welches Tool verwendet werden soll, schauen Sie sich die `manifest.json`-Datei Ihres S3-Bestandsberichts an. Diese Datei listet die Anzahl der Datendateien auf, die diesem Bericht zugeordnet sind. Wenn die Zahl groß ist, verwenden Sie Amazon Athena, da es über mehrere S3-Objekte läuft, während S3 Select für jeweils ein Objekt funktioniert. Weitere Informationen zur gemeinsamen Verwendung von Amazon S3 und Athena finden Sie unter Abfragen von Amazon S3 Inventory mit Amazon Athena und „Verwenden von Athena“ im AWS Speicher-Blogbeitrag Verschlüsseln von Objekten mit Amazon S3 Batch Operations.

So filtern Sie Ihren S3-Inventarbericht mithilfe von S3 Select

1. Öffnen Sie die `manifest.json` Datei aus Ihrem Inventarbericht und sehen Sie sich den `fileSchema` Abschnitt der anJSON. Dadurch wird die Abfrage informiert, die Sie für die Daten ausführen.

   Im Folgenden finden Sie JSON eine `manifest.json` Beispieldatei für ein Inventar im CSV - Format in einem Bucket mit aktivierter Versionierung. Je nachdem, wie Sie Ihren Bestandsbericht konfiguriert haben, sieht Ihr Manifest möglicherweise anders aus.

```
 {
   "sourceBucket": "batchoperationsdemo",
   "destinationBucket": "arn:aws:s3:::amzn-s3-demo-destination-bucket",
   "version": "2021-05-22",
   "creationTimestamp": "1558656000000",
   "fileFormat": "CSV",
   "fileSchema": "Bucket, Key, VersionId, IsLatest, IsDeleteMarker,
BucketKeyStatus",
   "files": [
```

```
      {
        "key": "demoinv/batchoperationsdemo/DemoInventory/data/009a40e4-
f053-4c16-8c75-6100f8892202.csv.gz",
        "size": 72691,
        "MD5checksum": "c24c831717a099f0ebe4a9d1c5d3935c"
      }
    ]
  }
```

Wenn die Versionierung für den Bucket nicht aktiviert ist oder wenn Sie den Bericht für die neuesten Versionen ausführen möchten, ist die `fileSchema` `Bucket`, `Key`, und `BucketKeyStatus`.

Wenn die Versionsverwaltung aktiviert ist, beinhaltet das `fileSchema` je nach der Einrichtung des Bestandsberichts möglicherweise Folgendes: `Bucket`, `Key`, `VersionId`, `IsLatest`, `IsDeleteMarker`, `BucketKeyStatus`. Achten Sie also auf die Spalten 1, 2, 3 und 6, wenn Sie Ihre Abfrage ausführen.

S3 Batch Operations benötigt den Bucket, den Schlüssel und die Versions-ID als Eingabe, um den Auftrag auszuführen, zusätzlich zu dem Feld, nach dem gesucht werden soll, nämlich `BucketKeyStatus`. Sie benötigen das `VersionID` Feld nicht, aber es hilft, das Feld zu spezifizieren, wenn Sie mit einem versionierten Bucket arbeiten. `VersionID` Weitere Informationen finden Sie unter Arbeiten mit Objekten in einem versioning-fähigen Bucket.

2. Suchen Sie die Datendateien für den Bestandsbericht. Das `manifest.json`-Objekt listet die Datendateien unter files (Dateien) auf.
3. Klicken Sie nach dem Suchen und Auswählen der Datendatei in der S3-Konsole auf Actions (Aktionen) und wählen Sie dann Query with S3 Select (Abfragen mit S3 Select).
4. Lassen Sie die Voreinstellung CSV, das Komma und die GZIPFelder ausgewählt und wählen Sie Weiter.
5. Um das Format Ihres Bestands zu überprüfen, bevor Sie fortfahren, wählen Sie Show file preview (Dateivorschau anzeigen).
6. Geben Sie die Spalten, auf die verwiesen werden soll, in das SQL Ausdrucksfeld ein und wählen Sie „Ausführen SQL“. Der folgende Ausdruck gibt die Spalten 1 bis 3 für alle Objekte zurück, die keinen S3-Bucket-Schlüssel konfiguriert haben.

   ```
   select s._1, s._2, s._3 from s3object s where s._6 = 'DISABLED'
   ```

   Nachfolgend sehen Sie einige Beispielergebnisse.

```
      batchoperationsdemo,0100059%7Ethumb.jpg,lsrtIxksLu0R0ZkYPL.LhgD5caTYn6vu
      batchoperationsdemo,0100074%7Ethumb.jpg,sd2M60g6Fdazoi6D5kNARIE7KzUibmHR
      batchoperationsdemo,0100075%7Ethumb.jpg,TLYESLnl1mXD5c4BwiOIinqFrktddkoL
      batchoperationsdemo,0200147%7Ethumb.jpg,amufzfMi_fEw0Rs99rxR_HrDFlE.l3Y0
      batchoperationsdemo,0301420%7Ethumb.jpg,9qGU2SEscL.C.c_sK89trmXYIwooABSh
      batchoperationsdemo,0401524%7Ethumb.jpg,ORnEWNuB1QhHrrYAGFsZhbyvEYJ3DUor
      batchoperationsdemo,200907200065HQ
%7Ethumb.jpg,d8LgvIVjbDR5mUVwW6pu9ahTfReyn5V4
      batchoperationsdemo,200907200076HQ
%7Ethumb.jpg,XUT25d7.gK40u_GmnupdaZg3BVx2jN40
      batchoperationsdemo,201103190002HQ
%7Ethumb.jpg,z.2sVRh0myqVi0BuIrngWlsRPQdb7qOS
```

7. Laden Sie die Ergebnisse herunter, speichern Sie sie in einem CSV Format und laden Sie sie als Objektliste für den S3 Batch Operations-Job auf Amazon S3 hoch.
8. Wenn Sie mehrere Manifestdateien haben, führen Sie auch auf diesen Query with S3 Select (Abfragen mit S3 Select) aus. Abhängig von der Größe der Ergebnisse können Sie die Listen kombinieren und einen einzelnen Auftrag für S3-Batch-Vorgänge ausführen oder jede Liste als separater Auftrag ausführen. Um die Anzahl der auszuführenden Jobs zu bestimmen, sollten Sie den Preis für die Ausführung der einzelnen S3 Batch Operations-Jobs berücksichtigen.

Schritt 3: Einrichten und Ausführen des Auftrags für S3-BatchVorgänge

Nachdem Sie Ihre gefilterten CSV Listen mit S3-Objekten erstellt haben, können Sie mit dem S3-Batch-Operations-Job beginnen, um die Objekte mit aktivierten S3-Bucket-Keys zu verschlüsseln.

Ein Auftrag bezieht sich kollektiv auf die Liste (Manifeste) der bereitgestellten Objekte, die durchgeführte Operation und die angegebenen Parameter. Die einfachste Methode, diese Gruppe von Objekten mit aktivierten S3-Bucket Keys zu verschlüsseln, besteht darin, den Kopiervorgang zu verwenden und dasselbe Zielpräfix wie die im Manifest aufgelisteten Objekte anzugeben. In einem Bucket ohne Version überschreibt dieser Vorgang die vorhandenen Objekte. In einem Bucket mit aktivierter Versionierung erstellt dieser Vorgang eine neuere, verschlüsselte Version der Objekte.

Geben Sie beim Kopieren der Objekte an, dass Amazon S3 die Objekte mit SSE - KMS Verschlüsselung verschlüsseln soll. Bei diesem Job werden die Objekte kopiert, sodass für alle Ihre Objekte nach Abschluss ein aktualisiertes Erstellungsdatum angezeigt wird, unabhängig davon, wann Sie sie ursprünglich zu Amazon S3 hinzugefügt haben. Geben Sie auch die anderen Eigenschaften für Ihre Gruppe von Objekten als Teil des S3 Batchoperations-Auftrags an, einschließlich Objekt-Markierungen und Speicherklasse.

Teilschritte

- Richten Sie Ihre IAM Richtlinie ein
- Richten Sie Ihre Batch IAM Operations-Rolle ein
- Aktivieren Sie S3 Bucket Keys für einen vorhandenen Bucket
- Erstellen eines Auftrags für S3-BatchVorgänge
- Ausführen eines Auftrags für S3-BatchVorgänge

Richten Sie Ihre IAM Richtlinie ein

1. Öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Richtlinie und dann Richtlinie erstellen aus.
3. Wählen Sie die JSONRegisterkarte aus. Wählen Sie Richtlinie bearbeiten und fügen Sie die IAM Beispielrichtlinie hinzu, die im folgenden Codeblock angezeigt wird.

   Nachdem Sie das Richtlinienbeispiel in Ihre IAMKonsole kopiert haben, ersetzen Sie Folgendes:

   a. *amzn-s3-demo-source-bucket*Ersetzen Sie es durch den Namen Ihres Quell-Buckets, aus dem Objekte kopiert werden sollen.

   b. *amzn-s3-demo-destination-bucket*Ersetzen Sie ihn durch den Namen Ihres Ziel-Buckets, in den Objekte kopiert werden sollen.

   c. Ersetzen Sie *amzn-s3-demo-manifest-bucket*/*manifest-key* durch den Namen Ihres Manifest-Objekts.

   d. *amzn-s3-demo-completion-report-bucket*Ersetzen Sie ihn durch den Namen des Buckets, in dem Sie Ihre Abschlussberichte speichern möchten.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CopyObjectsToEncrypt",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:PutObjectTagging",
        "s3:PutObjectAcl",
        "s3:PutObjectVersionTagging",
```

```
        "s3:PutObjectVersionAcl",
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:GetObjectTagging",
        "s3:GetObjectVersion",
        "s3:GetObjectVersionAcl",
        "s3:GetObjectVersionTagging"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-source-bucket/*",
        "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
      ]
    },
    {
      "Sid": "ReadManifest",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-manifest-bucket/manifest-key"
    },
    {
      "Sid": "WriteReport",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
    }
  ]
}
```

4. Wählen Sie Next: Markierungen (Weiter: Markierungen) aus.
5. Wählen Sie alle gewünschten Markierungen aus (optional) und wählen Sie Next: Review (Weiter: Überprüfung).
6. Geben Sie einen Namen und optional eine Beschreibung für die Richtlinie an und wählen Sie dann Create policy (Richtlinie erstellen) aus.
7. Wählen Sie Review policy (Richtlinie überprüfen) aus und klicken Sie anschließend auf Save changes (Änderungen speichern).

8. Nachdem Ihre S3-Richtlinie für Batch Operations nun abgeschlossen ist, kehren Sie über die Konsole zur Seite IAM Richtlinien zurück. Markieren Sie den Richtliniennamen, klicken Sie auf die Schaltfläche links neben dem Richtliniennamen und wählen Sie Policy actions (Richtlinienaktionen), und dann Attach (Hinzufügen).

   Um die neu erstellte Richtlinie an eine IAM Rolle anzuhängen, wählen Sie die entsprechenden Benutzer, Gruppen oder Rollen in Ihrem Konto aus und wählen Sie Richtlinie anhängen. Dadurch kehren Sie zur IAM Konsole zurück.

Richten Sie Ihre Batch IAM Operations-Rolle ein

1. Wählen Sie in der IAMKonsole im Navigationsbereich Rollen und dann Rolle erstellen aus.
2. Wählen Sie AWS-Service, S3 und S3 Batch Operations. Wählen Sie dann Next: Permissions.
3. Beginnen Sie mit der Eingabe des Namens der IAM Richtlinie, die Sie gerade erstellt haben. Aktivieren Sie das Kontrollkästchen nach dem Richtliniennamen, wenn es angezeigt wird, und wählen Sie Next: Markierungen (Weiter: Markierungen).
4. (Optional) Fügen Sie Markierungen hinzu, oder lassen Sie die Schlüssel- und Wertefelder für diese Übung leer. Klicken Sie auf Next: Review (Weiter: Prüfen).
5. Geben Sie einen Rollennamen ein, akzeptieren Sie die Standard-Beschreibung oder fügen Sie eine eigene hinzu. Wählen Sie Create role aus.
6. Stellen Sie sicher, dass der Benutzer, der den Auftrag erstellt, über die Berechtigungen im folgenden Beispiel verfügt.

   *account-id*Ersetzen Sie es durch Ihre AWS-Konto ID und *IAM-role-name* durch den Namen, den Sie für die IAM Rolle verwenden möchten, die Sie später im Schritt zur Auftragserstellung für Batch Operations erstellen werden. Weitere Informationen finden Sie unter Gewähren von Berechtigungen für Batchoperationen.

```
{
"Sid": "AddIamPermissions",
"Effect": "Allow",
"Action": [
"iam:GetRole",
"iam:PassRole"
],
"Resource": "arn:aws:iam::account-id:role/IAM-role-name"
}
```

Aktivieren Sie S3 Bucket Keys für einen vorhandenen Bucket

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Bucket aus, für den Sie einen S3-Bucket-Schlüssel aktivieren möchten.
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie unter Default encryption (Standard-Verschlüsselung) Edit (Bearbeiten) aus.
5. Unter Verschlüsselungstyp können Sie zwischen von Amazon S3 verwalteten Schlüsseln (SSE-S3) und AWS Key Management Service Schlüssel (SSE-KMS) wählen.
6. Wenn Sie unter AWS Key Management Service Schlüssel (SSE-KMS) ausgewählt haben AWS KMS key, können Sie den AWS KMS Schlüssel über eine der folgenden Optionen angeben.

   - Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Aus Ihren AWS KMS Schlüsseln auswählen. Wählen Sie aus der Liste der verfügbaren Schlüssel einen symmetrischen KMS Verschlüsselungsschlüssel aus, der sich in derselben Region wie Ihr Bucket befindet. Sowohl der AWS verwaltete Schlüssel (`aws/s3`) als auch Ihre vom Kunden verwalteten Schlüssel werden in der Liste angezeigt.
   - Um den KMS Schlüssel einzugebenARN, wählen Sie AWS KMS Schlüssel ARN eingeben und geben Sie dann Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
   - Um einen neuen, vom Kunden verwalteten Schlüssel in der AWS KMS Konsole zu erstellen, wählen Sie KMSSchlüssel erstellen aus.
7. Unter Bucket Key (Bucket-Schlüssel), wählen Sie Enable (Aktivieren) und dann Save changes (Änderungen speichern) aus.

Da nun ein S3-Bucket-Key auf Bucket-Ebene aktiviert ist, übernehmen Objekte, die in diesen Bucket hochgeladen, geändert oder kopiert werden, standardmäßig diese Verschlüsselungskonfiguration. Dies umfasst Objekte, die mit Amazon S3 Batch Operations kopiert wurden.

Erstellen eines Auftrags für S3-BatchVorgänge

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im Navigationsbereich die Option Batch Operations (Batch-Vorgänge) und dann Create Job (Auftrag erstellen).
3. Wählen Sie die Region aus, in der Sie Ihre Objekte speichern, und wählen Sie CSVden Manifesttyp aus.

4. Geben Sie den Pfad ein oder navigieren Sie zu der CSV Manifestdatei, die Sie zuvor aus den Ergebnissen von S3 Select (oder Athena) erstellt haben. Wenn Ihr Manifest eine Version enthältIDs, wählen Sie dieses Feld aus. Wählen Sie Next.
5. Wählen Sie die Copy-Operation und wählen Sie den Ziel-Bucket für den Kopiervorgang aus. Sie können serverseitige Verschlüsselung deaktiviert lassen. Solange S3 Bucket-Schlüssel für den Ziel-Bucket aktiviert ist, wendet der Kopiervorgang S3 Bucket-Schlüssel auf den Ziel-Bucket an.
6. (Optional) Wählen Sie nach Bedarf eine Speicherklasse und die anderen Parameter aus. Die in diesem Schritt angegebenen Parameter gelten für alle Vorgänge, die an den im Manifest aufgeführten Objekten ausgeführt werden. Wählen Sie Weiter.
7. Gehen Sie zum Konfigurieren der serverseitigen Verschlüsselung wie folgt vor:

   a. Wählen Sie unter Serverseitige Verschlüsselung eine der folgenden Optionen aus:

      - Um die Bucket-Einstellungen für die serverseitige Standardverschlüsselung von Objekten beizubehalten, wenn sie in Amazon S3 gespeichert werden, wählen Sie Keinen Verschlüsselungsschlüssel angeben aus. Solange S3-Bucket-Schlüssel für den Ziel-Bucket aktiviert sind, wendet der Kopiervorgang S3-Bucket-Schlüssel auf den Ziel-Bucket an.

        > **Note**
        >
        > Wenn die Bucket-Richtlinie für das angegebene Ziel vorschreibt, dass Objekte verschlüsselt werden müssen, bevor sie in Amazon S3 gespeichert werden, müssen Sie einen Verschlüsselungsschlüssel angeben. Andernfalls schlägt das Kopieren von Objekten in das Ziel fehl.

      - Um Objekte zu verschlüsseln, bevor sie in Amazon S3 gespeichert werden, wählen Sie Verschlüsselungsschlüssel angeben aus.

   b. Wenn Sie unter Verschlüsselungseinstellungen die Option Verschlüsselungsschlüssel angeben auswählen, müssen Sie entweder Verwenden von Ziel-Bucket-Einstellungen für die Standardverschlüsselung oder Überschreiben der Ziel-Bucket-Einstellungen für die Standardverschlüsselung auswählen.

   c. Wenn Sie Überschreiben der Ziel-Bucket-Einstellungen für die Standardverschlüsselung auswählen, müssen Sie die folgenden Verschlüsselungseinstellungen konfigurieren.

      i. Unter Verschlüsselungstyp müssen Sie entweder von Amazon S3 verwaltete Schlüssel (SSE-S3) oder AWS Key Management Service Schlüssel (SSE-KMS) auswählen. SSE-

S3 verwendet eine der stärksten Blockchiffren — den 256-Bit-Advanced Encryption Standard (-256), um jedes Objekt zu verschlüsseln. AES SSE- bietet Ihnen mehr Kontrolle KMS über Ihren Schlüssel. Weitere Informationen erhalten Sie unter Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden (SSE-S3) und Serverseitige Verschlüsselung mit AWS KMS Schlüsseln verwenden (SSE-) KMS.

ii. Wenn Sie den AWS Key Management Service Schlüssel (SSE-KMS) unter wählen AWS KMS key, können Sie Ihren AWS KMS key mithilfe einer der folgenden Optionen angeben.

- Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Wählen Sie aus Ihrem und wählen Sie dann einen symmetrischen KMS Verschlüsselungsschlüssel aus AWS KMS keys, der sich in derselben Region wie Ihr Bucket befindet. Sowohl der AWS verwaltete Schlüssel (`aws/s3`) als auch Ihre vom Kunden verwalteten Schlüssel werden in der Liste angezeigt.
- Um den KMS Schlüssel einzugebenARN, wählen Sie AWS KMS Schlüssel ARN eingeben und geben Sie Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
- Um einen neuen, vom Kunden verwalteten Schlüssel in der AWS KMS Konsole zu erstellen, wählen Sie KMS Schlüssel erstellen.

iii. Wählen Sie unter Bucket Key (Bucket-Schlüssel) die Option Enable (Aktivieren). Die Copy-Operation wendet einen S3-Bucket-Schlüssel auf den Ziel-Bucket an.

8. Geben Sie Ihrem Job eine Beschreibung (oder behalten Sie die Standardeinstellung bei), legen Sie die Prioritätsstufe fest, wählen Sie einen Berichtstyp aus und geben Sie den Pfad zum Ziel des Fertigstellungsberichts an.
9. Stellen Sie sicher, dass Sie im Abschnitt Berechtigungen die IAM Rolle Batch Operations auswählen, die Sie zuvor definiert haben. Wählen Sie Next.
10. Überprüfen Sie die Einstellungen unter Review (Überprüfen). Wenn Sie Änderungen vornehmen müssen, wählen Sie Previous (Vorherige) aus. Nachdem Sie die Einstellungen für Batch-Vorgänge bestätigt haben, wählen Sie Create Job (Auftrag erstellen).

    Weitere Informationen finden Sie unter Erstellen eines S3-Batch-Vorgangsauftrags.

Ausführen eines Auftrags für S3-BatchVorgänge

Der Setup-Assistent kehrt Sie automatisch zum Abschnitt S3-Batch-Vorgänge der Amazon S3-Konsole zurück. Ihr neuer Auftrag wechselt vom New-Zustand in den Preparing-Zustand, da S3 den

Prozess beginnt. Während des Preparing-Zustands liest S3 das Manifest des Auftrags, prüft es auf Fehler und berechnet die Anzahl der Objekte.

1. Klicken Sie auf die Aktualisierungs-Schaltfläche in der Amazon S3-Konsole, um den Fortschritt zu überprüfen. Abhängig von der Größe des Manifests kann das Lesen Minuten oder Stunden dauern.
2. Nachdem S3 das Manifest des Auftrags gelesen hat, wechselt der Auftrag in den Zustand Awaiting your confirmation (Wartet auf Ihre Bestätigung). Klicken Sie auf die Options-Schaltfläche links neben der Auftrags-ID und wählen Sie Run job (Auftrag ausführen).
3. Wählen Sie die Einstellungen für den Auftrag aus und wählen Sie in der unteren rechten Ecke des Fensters Run job (Auftrag ausführen) aus.

   Nachdem der Auftrag ausgeführt wird, können Sie die Schaltfläche Aktualisieren auswählen, um den Fortschritt über die Dashboard-Ansicht der Konsole zu überprüfen oder den bestimmten Auftrag auszuwählen.
4. Wenn der Auftrag abgeschlossen wurde, können Sie die Successful (erfolgreichen) und Failed (fehlgeschlagenen) Objektzählungen anzeigen, um zu bestätigen, dass alles wie erwartet ausgeführt wurde. Wenn Sie Auftragsberichte aktiviert haben, überprüfen Sie Ihren Auftragsbericht auf die genaue Ursache für fehlgeschlagene Vorgänge.

   Sie können diese Schritte auch mithilfe von AWS CLI AWS SDKs, oder Amazon S3 ausführen RESTAPI. Weitere Informationen zum Nachverfolgen von Auftragsstatus- und Abschlussberichten finden Sie unter Verfolgen von Auftragsstatus- und Abschluss.

Beispiele, die den Kopiervorgang mit Tags unter Verwendung von AWS CLI und zeigen AWS SDK for Java, finden Sie unterErstellen eines Batch-Vorgangsauftrags mit Auftrags-Markierungen zur Kennzeichnung.

Funktion aufrufen AWS Lambda

Sie können Amazon S3 Batch Operations verwenden, um umfangreiche Batch-Operationen an Amazon S3 S3-Objekten durchzuführen. Der Vorgang Batch Operations der AWS Lambda Funktion aufrufen initiiert AWS Lambda Funktionen zur Ausführung benutzerdefinierter Aktionen für Objekte, die in einem Manifest aufgeführt sind. In diesem Abschnitt wird beschrieben, wie Sie eine Lambda-Funktion zur Verwendung mit S3-Batch-Vorgänge erstellen und wie Sie einen Auftrag zum Aufrufen der Funktion erstellen. Die S3-Batch-Vorgängeaufgabe verwendet die `LambdaInvoke`-Operation, um eine Lambda-Funktion für jedes in einem Manifest aufgeführte Objekt auszuführen.

Sie können mit S3 Batch Operations arbeiten, indem Sie die Amazon S3-Konsole, AWS Command Line Interface (AWS CLI) AWS SDKs, oder Amazon S3 verwenden RESTAPI. Weitere Informationen zur Verwendung von Lambda finden Sie unter Getting started with AWS Lambda(Erste Schritte mit Lambda) im AWS Lambda -Entwicklerhandbuch.

In den folgenden Abschnitten werden die ersten Schritte mit S3-Batch-Vorgänge mit Lambda erläutert.

Themen

- Lambda mit Batch-Operationen verwenden
- Erstellen einer Lambda-Funktion zur Verwendung mit S3-BatchVorgänge
- Erstellen eines S3-Batchoperations-Auftrags, der eine Lambda-Funktion aufruft
- Bereitstellen von Informationen auf Aufgabenebene in Lambda-Manifesten
- Tutorial zu S3-Batchvorgängen

Lambda mit Batch-Operationen verwenden

Wenn Sie S3 Batch Operations mit verwenden AWS Lambda, müssen Sie neue Lambda-Funktionen speziell für die Verwendung mit S3 Batch Operations erstellen. Sie können vorhandene ereignisbasierte Funktionen von Amazon S3 nicht mit S3-Batch-Vorgänge wiederverwenden. Ereignisfunktionen können Nachrichten lediglich erhalten und nicht zurückgeben. Die Lambda-Funktionen, die mit S3-Batch-Vorgänge verwendet werden, müssen Nachrichten akzeptieren und zurückgeben. Weitere Informationen zur Verwendung von Lambda mit Amazon S3-Ereignissen finden Sie unter Using AWS Lambda with Amazon S3 im AWS Lambda Developer Guide.

Sie erstellen einen S3-Batchoperations-Auftrag, der Ihre Lambda-Funktion aufruft. Der Auftrag führt dieselbe Lambda-Funktion für alle in Ihrem Manifest aufgeführten Objekte aus. Sie können steuern, welche Versionen Ihrer Lambda-Funktion während der Verarbeitung der Objekte in Ihrem Manifest verwendet werden sollen. S3 Batch Operations unterstützt unqualifizierte Amazon-Ressourcennamen (ARNs), Aliase und bestimmte Versionen. Weitere Informationen finden Sie unter Einführung in die AWS Lambda Versionierung im Entwicklerhandbuch.AWS Lambda

Wenn Sie dem S3 Batch Operations-Job eine Funktion zur Verfügung stellenARN, die einen Alias oder den `$LATEST` Qualifier verwendet, und Sie die Version aktualisieren, auf die einer dieser beiden verweist, beginnt S3 Batch Operations, die neue Version Ihrer Lambda-Funktion aufzurufen. Dies kann nützlich sein, wenn Sie mitten in der Ausführung einer großen Aufgabe Funktionen aktualisieren

möchten. Wenn Sie nicht möchten, dass S3 Batch Operations die verwendete Version ändert, geben Sie die spezifische Version im `FunctionARN` Parameter an, wenn Sie Ihren Job erstellen.

Verwenden von Lambda- und Batch-Operationen mit Verzeichnis-Buckets

Verzeichnis-Buckets sind eine Art von Amazon S3 S3-Bucket, die für Workloads oder leistungskritische Anwendungen konzipiert sind, die eine konsistente Latenz im einstelligen Millisekundenbereich erfordern. Weitere Informationen finden Sie unter Verzeichnis-Buckets.

Für die Verwendung von Batch Operations zum Aufrufen von Lambda-Funktionen, die auf Verzeichnis-Buckets wirken, gelten besondere Anforderungen. Beispielsweise müssen Sie Ihre Lambda-Anfrage mithilfe eines aktualisierten JSON Schemas strukturieren und Folgendes angeben InvocationSchemaVersion2.0 (nicht 1.0), wenn Sie den Job erstellen. Dieses aktualisierte Schema ermöglicht Ihnen die Angabe optionaler Schlüssel-Wert-Paare für UserArguments, mit dem Sie bestimmte Parameter vorhandener Lambda-Funktionen ändern können. Weitere Informationen finden Sie unter Automatisieren der Objektverarbeitung in Amazon S3 S3-Verzeichnis-Buckets mit S3 Batch Operations und AWS Lambda im AWS Storage-Blog.

Antwort- und Ergebniscodes

S3 Batch Operations ruft die Lambda-Funktion mit einem oder mehreren Schlüsseln auf, denen jeweils ein Schlüssel `TaskID` zugeordnet ist. S3 Batch Operations erwartet von Lambda-Funktionen einen Ergebniscode pro Schlüssel. Jede in der Anfrage IDs gesendete Aufgabe, die nicht mit einem Ergebniscode pro Schlüssel zurückgegeben wird, erhält den Ergebniscode aus dem Feld. `treatMissingKeysAs` `treatMissingKeysAsist` ein optionales Anforderungsfeld und hat standardmäßig den Wert. `TemporaryFailure` Die folgende Tabelle enthält die anderen möglichen Ergebniscodes und Werte für das `treatMissingKeysAs` Feld.

Antwortcode	Beschreibung
`Succeeded`	Die Aufgabe wurde normal abgeschlossen. Wenn Sie einen Aufgabenabschlussbericht angefordert haben, ist die Ergebniszeichenfolge der Aufgabe in dem Bericht enthalten.
`TemporaryFailure`	Die Aufgabe unterlag einem vorübergehenden Fehler und wird erneut ausgeführt, bevor der Job abgeschlossen ist. Die Ergebniszeichenfol

Antwortcode	Beschreibung
	ge wird ignoriert. Wenn dies der letzte Versuch ist, ist die Fehlermeldung im abschließenden Bericht enthalten.
`PermanentFailure`	Die Aufgabe unterlag einem dauerhaften Fehler. Wenn Sie einen Aufgabenabschlussb ericht angefordert haben, ist die Aufgabe als `Failed` markiert und enthält die Fehlermel dungszeichenfolge. Ergebniszeichenfolge aus fehlgeschlagenen Aufgaben werden ignoriert.

Erstellen einer Lambda-Funktion zur Verwendung mit S3-BatchVorgänge

Dieser Abschnitt enthält Beispielberechtigungen AWS Identity and Access Management (IAM), die Sie mit Ihrer Lambda-Funktion verwenden müssen. Er enthält auch eine Beispiel-Lambda-Funktion zur Verwendung mit S3-BatchVorgänge. Wenn Sie noch nie eine Lambda-Funktion erstellt haben, finden Sie weitere Informationen unter Tutorial: Using AWS Lambda with Amazon S3 im AWS Lambda Developer Guide.

Sie müssen Lambda-Funktionen speziell für die Verwendung mit S3-Batch-Vorgänge erstellen. Sie können bestehende ereignisbasierte Lambda-Funktionen von Amazon S3 nicht wiederverwenden, da Lambda-Funktionen, die für S3 Batch Operations verwendet werden, spezielle Datenfelder akzeptieren und zurückgeben müssen.

Important

AWS Lambda In Java geschriebene Funktionen akzeptieren entweder RequestHandler oder RequestStreamHandlerHandler-Schnittstellen. Um das Anforderungs- und Antwortformat von S3 Batch Operations zu unterstützen, AWS Lambda ist jedoch die `RequestStreamHandler` Schnittstelle für die benutzerdefinierte Serialisierung und Deserialisierung einer Anfrage und Antwort erforderlich. Diese Schnittstelle ermöglicht es Lambda, ein InputStream und an die `handleRequest` Java-Methode OutputStream zu übergeben.
Achten Sie darauf, die `RequestStreamHandler`-Schnittstelle zu verwenden, wenn Sie Lambda-Funktionen mit S3-Batch-Vorgänge verwenden. Wenn Sie eine `RequestHandler` Schnittstelle verwenden, schlägt der Batch-Job fehl und im Abschlussbericht wird „Ungültig JSON zurückgegeben in Lambda payload“ angezeigt.

Weitere Informationen finden Sie unter Handler interfaces (Handler-Schnittstellen) im AWS Lambda -Benutzerhandbuch.

Beispiele von IAM-Berechtigungen

Im Folgenden finden Sie Beispiele für die IAM Berechtigungen, die für die Verwendung einer Lambda-Funktion mit S3 Batch Operations erforderlich sind.

Example – Vertrauensrichtlinie für S3-BatchVorgänge

Im Folgenden finden Sie ein Beispiel für die Vertrauensrichtlinie, die Sie für die IAM Rolle Batch Operations verwenden können. Diese IAM Rolle wird bei der Erstellung des Jobs angegeben und erteilt Batch Operations die Berechtigung, die IAM Rolle zu übernehmen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "batchoperations.s3.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Example — Lambda-Richtlinie IAM

Im Folgenden finden Sie ein Beispiel IAM für eine Richtlinie, die S3 Batch Operations die Erlaubnis erteilt, die Lambda-Funktion aufzurufen und das Eingabemanifest zu lesen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "BatchOperationsLambdaPolicy",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
```

```
                "s3:PutObject",
                "lambda:InvokeFunction"
            ],
            "Resource": "*"
        }
    ]
}
```

Beispiel: Anforderung und Antwort

Dieser Abschnitt bietet Beispiele für Anforderungen und Antworten für die Lambda-Funktion.

Example Anforderung

Das Folgende ist ein JSON Beispiel für eine Anfrage für die Lambda-Funktion.

```
{
    "invocationSchemaVersion": "1.0",
    "invocationId": "YXNkbGZqYWRmaiBhc2RmdW9hZHNmZGpmaGFzbGtkaGZza2RmaAo",
    "job": {
        "id": "f3cc4f60-61f6-4a2b-8a21-d07600c373ce"
    },
    "tasks": [
        {
            "taskId": "dGFza2lkZ29lc2hlcmUK",
            "s3Key": "customerImage1.jpg",
            "s3VersionId": "1",
            "s3BucketArn": "arn:aws:s3:us-east-1:0123456788:amzn-s3-demo-bucket1"
        }
    ]
}
```

Example Antwort

Das Folgende ist ein JSON Beispiel für eine Antwort für die Lambda-Funktion.

```
{
  "invocationSchemaVersion": "1.0",
  "treatMissingKeysAs" : "PermanentFailure",
  "invocationId" : "YXNkbGZqYWRmaiBhc2RmdW9hZHNmZGpmaGFzbGtkaGZza2RmaAo",
  "results": [
    {
      "taskId": "dGFza2lkZ29lc2hlcmUK",
      "resultCode": "Succeeded",
```

```
      "resultString": "[\"Mary Major", \"John Stiles\"]"
    }
  ]
}
```

Beispiel für eine Lambda-Funktion für S3-BatchVorgänge

Im folgenden Beispiel entfernt Python Lambda eine Löschmarkierung von einem versionierten Objekt.

Wie das Beispiel zeigt, sind Schlüssel aus S3 Batch Operations URL codiert. Um Amazon S3 mit anderen AWS Diensten zu verwenden, ist es wichtig, dass Sie den Schlüssel URL dekodieren, der von S3 Batch Operations übergeben wird.

```
import logging
from urllib import parse
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
logger.setLevel("INFO")

s3 = boto3.client("s3")


def lambda_handler(event, context):
    """
    Removes a delete marker from the specified versioned object.

    :param event: The S3 batch event that contains the ID of the delete marker
                  to remove.
    :param context: Context about the event.
    :return: A result structure that Amazon S3 uses to interpret the result of the
             operation. When the result code is TemporaryFailure, S3 retries the
             operation.
    """
    # Parse job parameters from Amazon S3 batch operations
    invocation_id = event["invocationId"]
    invocation_schema_version = event["invocationSchemaVersion"]

    results = []
    result_code = None
    result_string = None
```

```
    task = event["tasks"][0]
    task_id = task["taskId"]

    try:
        obj_key = parse.unquote(task["s3Key"], encoding="utf-8")
        obj_version_id = task["s3VersionId"]
        bucket_name = task["s3BucketArn"].split(":")[-1]

        logger.info(
            "Got task: remove delete marker %s from object %s.", obj_version_id,
obj_key
        )

        try:
            # If this call does not raise an error, the object version is not a delete
            # marker and should not be deleted.
            response = s3.head_object(
                Bucket=bucket_name, Key=obj_key, VersionId=obj_version_id
            )
            result_code = "PermanentFailure"
            result_string = (
                f"Object {obj_key}, ID {obj_version_id} is not " f"a delete marker."
            )

            logger.debug(response)
            logger.warning(result_string)
        except ClientError as error:
            delete_marker = error.response["ResponseMetadata"]["HTTPHeaders"].get(
                "x-amz-delete-marker", "false"
            )
            if delete_marker == "true":
                logger.info(
                    "Object %s, version %s is a delete marker.", obj_key,
obj_version_id
                )
                try:
                    s3.delete_object(
                        Bucket=bucket_name, Key=obj_key, VersionId=obj_version_id
                    )
                    result_code = "Succeeded"
                    result_string = (
                        f"Successfully removed delete marker "
                        f"{obj_version_id} from object {obj_key}."
                    )
```

```
                    logger.info(result_string)
                except ClientError as error:
                    # Mark request timeout as a temporary failure so it will be
retried.
                    if error.response["Error"]["Code"] == "RequestTimeout":
                        result_code = "TemporaryFailure"
                        result_string = (
                            f"Attempt to remove delete marker from  "
                            f"object {obj_key} timed out."
                        )
                        logger.info(result_string)
                    else:
                        raise
            else:
                raise ValueError(
                    f"The x-amz-delete-marker header is either not "
                    f"present or is not 'true'."
                )
    except Exception as error:
        # Mark all other exceptions as permanent failures.
        result_code = "PermanentFailure"
        result_string = str(error)
        logger.exception(error)
    finally:
        results.append(
            {
                "taskId": task_id,
                "resultCode": result_code,
                "resultString": result_string,
            }
        )
    return {
        "invocationSchemaVersion": invocation_schema_version,
        "treatMissingKeysAs": "PermanentFailure",
        "invocationId": invocation_id,
        "results": results,
    }
```

Erstellen eines S3-Batchoperations-Auftrags, der eine Lambda-Funktion aufruft

Wenn Sie einen S3-Batchoperations-Auftrag erstellen, um eine Lambda-Funktion aufzurufen, müssen Sie Folgendes angeben:

- Die ARN Ihrer Lambda-Funktion (die den Funktionsalias oder eine bestimmte Versionsnummer enthalten kann)
- Eine IAM Rolle mit der Berechtigung, die Funktion aufzurufen
- Den Aktionsparamete `LambdaInvokeFunction`

Weitere Informationen zum Erstellen eines S3-Batchoperations-Auftrags finden Sie unter Erstellen eines S3-Batch-Vorgangsauftrags und Von S3 Batch-Vorgänge unterstützte Vorgänge.

Im folgenden Beispiel wird ein S3-Auftrag für Batch Operations erstellt, der mithilfe von eine Lambda-Funktion aufruft. AWS CLI Um dieses Beispiel zu verwenden, ersetzen Sie *`Platzhalter für Benutzereingaben`* mit Ihren eigenen Informationen.

```
aws s3control create-job
    --account-id account-id
    --operation  '{"LambdaInvoke": { "FunctionArn": "arn:aws:lambda:region:account-
id:function:LambdaFunctionName" } }'
    --manifest '{"Spec":{"Format":"S3BatchOperations_CSV_20180820","Fields":
["Bucket","Key"]},"Location":{"ObjectArn":"arn:aws:s3:::amzn-s3-demo-manifest-
bucket","ETag":"ManifestETag"}}'
    --report '{"Bucket":"arn:aws:s3:::amzn-s3-demo-
bucket","Format":"Report_CSV_20180820","Enabled":true,"Prefix":"ReportPrefix","ReportScope":"Al
    --priority 2
    --role-arn arn:aws:iam::account-id:role/BatchOperationsRole
    --region region
    --description "Lambda Function"
```

Bereitstellen von Informationen auf Aufgabenebene in Lambda-Manifesten

Wenn Sie AWS Lambda Funktionen mit S3 Batch Operations verwenden, benötigen Sie möglicherweise zusätzliche Daten für jede Aufgabe oder Taste, an der gearbeitet wird. Beispielsweise möchten Sie möglicherweise, dass sowohl ein Quellobjektschlüssel als auch ein neuer Objektschlüssel bereitgestellt werden. Ihre Lambda-Funktion kann in einem solchen Fall den Quellschlüssel in einen neuen S3-Bucket unter einem neuen Namen speichern. Standardmäßig können Sie mit Batch Operations nur den Ziel-Bucket und eine Liste von Quellschlüsseln im Eingabemanifest für Ihren Job angeben. In den folgenden Beispielen wird beschrieben, wie

Sie zusätzliche Daten in Ihr Manifest aufnehmen können, um komplexere Lambda-Funktionen auszuführen.

Verwenden Sie das folgende URL -kodierte Format, um Schlüsselparameter in Ihrem S3-Batch-Operations-Manifest anzugeben, die im Code Ihrer Lambda-Funktion verwendet werden sollen. JSON Das `key`-Feld wird an Ihre Lambda-Funktion übergeben, als wäre es ein Amazon S3-Objektschlüssel. Es kann jedoch von der Lambda-Funktion so interpretiert werden, dass es andere Werte oder mehrere Schlüssel enthält, wie in den folgenden Beispielen gezeigt.

> **Note**
>
> Die maximale Anzahl von Zeichen für das Feld `key` im Manifest beträgt 1.024.

Example — Manifest, das die „Amazon S3 S3-Schlüssel“ durch Zeichenketten ersetzt JSON

Die URL -kodierte Version muss für S3 Batch Operations bereitgestellt werden.

```
amzn-s3-demo-bucket,{"origKey": "object1key", "newKey": "newObject1Key"}
amzn-s3-demo-bucket,{"origKey": "object2key", "newKey": "newObject2Key"}
amzn-s3-demo-bucket,{"origKey": "object3key", "newKey": "newObject3Key"}
```

Example — Manifest -kodiert URL

Diese URL -kodierte Version muss für S3 Batch Operations bereitgestellt werden. Die non-URL-encoded Version funktioniert nicht.

```
amzn-s3-demo-bucket,%7B%22origKey%22%3A%20%22object1key%22%2C%20%22newKey%22%3A
%20%22newObject1Key%22%7D
amzn-s3-demo-bucket,%7B%22origKey%22%3A%20%22object2key%22%2C%20%22newKey%22%3A
%20%22newObject2Key%22%7D
amzn-s3-demo-bucket,%7B%22origKey%22%3A%20%22object3key%22%2C%20%22newKey%22%3A
%20%22newObject3Key%22%7D
```

Example – Lambda-Funktion im Manifest-Format, die Ergebnisse in den Auftragsbericht schreibt.

Dieses URL -kodierte Manifestbeispiel enthält durch Pipen getrennte Objektschlüssel für die folgende Lambda-Funktion zum Analysieren.

```
amzn-s3-demo-bucket,object1key%7Clower
amzn-s3-demo-bucket,object2key%7Cupper
```

```
amzn-s3-demo-bucket,object3key%7Creverse
amzn-s3-demo-bucket,object4key%7Cdelete
```

Diese Lambda-Funktion zeigt, wie eine durch Pipe getrennte Aufgabe analysiert wird, die im S3-Batch-Operations-Manifest kodiert ist. Die Aufgabe gibt an, welcher Revisionsvorgang auf das angegebene Objekt angewendet wird.

```
import logging
from urllib import parse
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
logger.setLevel("INFO")

s3 = boto3.resource("s3")


def lambda_handler(event, context):
    """
    Applies the specified revision to the specified object.

    :param event: The Amazon S3 batch event that contains the ID of the object to
                  revise and the revision type to apply.
    :param context: Context about the event.
    :return: A result structure that Amazon S3 uses to interpret the result of the
             operation.
    """
    # Parse job parameters from Amazon S3 batch operations
    invocation_id = event["invocationId"]
    invocation_schema_version = event["invocationSchemaVersion"]

    results = []
    result_code = None
    result_string = None

    task = event["tasks"][0]
    task_id = task["taskId"]
    # The revision type is packed with the object key as a pipe-delimited string.
    obj_key, revision = parse.unquote(task["s3Key"], encoding="utf-8").split("|")
    bucket_name = task["s3BucketArn"].split(":")[-1]

    logger.info("Got task: apply revision %s to %s.", revision, obj_key)
```

```
    try:
        stanza_obj = s3.Bucket(bucket_name).Object(obj_key)
        stanza = stanza_obj.get()["Body"].read().decode("utf-8")
        if revision == "lower":
            stanza = stanza.lower()
        elif revision == "upper":
            stanza = stanza.upper()
        elif revision == "reverse":
            stanza = stanza[::-1]
        elif revision == "delete":
            pass
        else:
            raise TypeError(f"Can't handle revision type '{revision}'.")

        if revision == "delete":
            stanza_obj.delete()
            result_string = f"Deleted stanza {stanza_obj.key}."
        else:
            stanza_obj.put(Body=bytes(stanza, "utf-8"))
            result_string = (
                f"Applied revision type '{revision}' to " f"stanza {stanza_obj.key}."
            )

        logger.info(result_string)
        result_code = "Succeeded"
    except ClientError as error:
        if error.response["Error"]["Code"] == "NoSuchKey":
            result_code = "Succeeded"
            result_string = (
                f"Stanza {obj_key} not found, assuming it was deleted "
                f"in an earlier revision."
            )
            logger.info(result_string)
        else:
            result_code = "PermanentFailure"
            result_string = (
                f"Got exception when applying revision type '{revision}' "
                f"to {obj_key}: {error}."
            )
            logger.exception(result_string)
    finally:
        results.append(
            {
```

```
                "taskId": task_id,
                "resultCode": result_code,
                "resultString": result_string,
            }
        )
    return {
        "invocationSchemaVersion": invocation_schema_version,
        "treatMissingKeysAs": "PermanentFailure",
        "invocationId": invocation_id,
        "results": results,
    }
```

Tutorial zu S3-Batchvorgängen

Das folgende Tutorial enthält vollständige end-to-end Verfahren für einige Batch-Operationsaufgaben mit Lambda. In diesem Tutorial erfahren Sie, wie Sie Batch-Operationen einrichten, um eine Lambda-Funktion für die Batch-Transcodierung von Videos aufzurufen, die in einem S3-Quell-Bucket gespeichert sind. Die Lambda-Funktion ruft AWS Elemental MediaConvert auf, um die Videos zu transkodieren.

- Tutorial: Batch-Transcodierung von Videos mit S3 Batch Operations

Alle Objektmarkierungen ersetzen

Sie können Amazon S3 Batch Operations verwenden, um umfangreiche Batch-Operationen an Amazon S3 S3-Objekten durchzuführen. Der Vorgang Alle Objekt-Tags ersetzen ersetzt die Objekt-Tags für jedes Objekt, das im Manifest aufgeführt ist. Ein -Objekt-Tag ist ein Schlüssel-Wert-Paar, mit dem Sie Metadaten über ein Objekt speichern können.

Um einen Job „Alle Objekt-Tags ersetzen“ zu erstellen, geben Sie eine Reihe von Tags an, die Sie anwenden möchten. S3 Batch-Vorgänge wendet auf jedes Objekt den gleichen Markierungssatz an. Der von Ihnen bereitgestellte Tag-Satz ersetzt alle Tag-Sätze, die bereits mit den Objekten im Manifest verknüpft sind. S3 Batch Operations unterstützt das Hinzufügen von Tags zu Objekten nicht, während die vorhandenen Tags beibehalten werden.

Wenn die Objekte in Ihrem Manifest zu einem versionierten Bucket gehören, können Sie die Tag-Gruppen auf bestimmte Versionen der einzelnen Objekte anwenden. Geben Sie dazu für jedes

Objekt im Manifest eine Versions-ID an. Wenn Sie keine Versions-ID für Objekte angeben, wendet S3 Batch Operations den Tagsatz auf die neueste Version jedes Objekts an. Weitere Informationen über Batchoperationsmanifeste finden Sie unter Angeben eines Manifests.

Weitere Informationen zum Tagging von Objekten finden Sie Kategorisieren des Speichers mithilfe von Markierungen in diesem Handbuch und unter PutObjectTagging, GetObjectTagging, und DeleteObjectTaggingin der Amazon Simple Storage Service API Reference.

Informationen zur Verwendung der Konsole zum Erstellen eines Auftrags „Alle Objekt-Tags ersetzen“ finden Sie unterErstellen eines S3-Batch-Vorgangsauftrags.

Beschränkungen und Einschränkungen

Wenn Sie Batch-Operationen verwenden, um Objekt-Tags zu ersetzen, gelten die folgenden Einschränkungen und Beschränkungen:

- Die Rolle AWS Identity and Access Management (IAM), die Sie für die Ausführung des Batch Operations-Jobs angeben, muss über Berechtigungen zum Ausführen des zugrunde liegenden `PutObjectTagging` Vorgangs verfügen. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter PutObjectTaggingin der Amazon Simple Storage Service API Reference.
- S3 Batch Operations verwendet Amazon S3 PutObjectTaggingOperation zum Anwenden von Tags auf jedes Objekt im Manifest. Alle Einschränkungen und Begrenzungen, die auf den zugrunde liegenden Vorgang zutreffen, gelten auch für S3-Batchoperations-Aufträge.

Alle Objektmrkierungen löschen

Sie können Amazon S3 Batch Operations verwenden, um umfangreiche Batch-Operationen an Amazon S3 S3-Objekten durchzuführen. Der Vorgang Alle Objekt-Tags löschen entfernt alle Amazon S3-Objekt-Tag-Sätze, die derzeit mit den im Manifest aufgeführten Objekten verknüpft sind. S3 Batch Operations unterstützt nicht das Löschen von Tags aus Objekten, während andere Tags beibehalten werden.

Wenn sich die Objekte in Ihrem Manifest in einem versionierten Bucket befinden, können Sie die Tag-Sätze aus einer bestimmten Version eines Objekts entfernen. Dazu müssen Sie für jedes Objekt im Manifest eine Versions-ID angeben. Wenn Sie keine Versions-ID für ein Objekt angeben, entfernt S3 Batch-Vorgänge den Tag-Satz aus der neuesten Version jedes Objekts. Weitere Informationen über Batchoperationsmanifeste finden Sie unter Angeben eines Manifests.

Weitere Informationen zum Objekt-Tagging finden Sie Kategorisieren des Speichers mithilfe von Markierungen in diesem Handbuch und PutObjectTagging, GetObjectTagging, und DeleteObjectTaggingin der Amazon Simple Storage Service API Reference.

> **Warning**
>
> Durch die Ausführung dieser Aufgabe werden alle Objekt-Tag-Sätze für jedes im Manifest aufgelistete Objekt entfernt.

Informationen zur Verwendung der Konsole zum Erstellen eines Auftrags „Alle Objekt-Tags löschen“ finden Sie unterErstellen eines S3-Batch-Vorgangsauftrags.

Beschränkungen und Einschränkungen

Wenn Sie Batch Operations verwenden, um Objekt-Tags zu löschen, gelten die folgenden Einschränkungen und Beschränkungen:

- Die Rolle AWS Identity and Access Management (IAM), die Sie für die Ausführung des Jobs angeben, muss über Berechtigungen zur Ausführung des zugrunde liegenden Amazon S3 `DeleteObjectTagging` S3-Vorgangs verfügen. Weitere Informationen finden Sie unter DeleteObjectTaggingin der Amazon Simple Storage Service API Reference.
- S3 Batch Operations verwendet Amazon S3 DeleteObjectTaggingOperation zum Entfernen der Tag-Sets von jedem Objekt im Manifest. Alle Einschränkungen und Begrenzungen, die auf den zugrunde liegenden Vorgang zutreffen, gelten auch für S3-Batchoperations-Aufträge.

Ersetzt die Zugriffskontrollliste (ACL)

Sie können Amazon S3 Batch Operations verwenden, um umfangreiche Batch-Operationen an Amazon S3 S3-Objekten durchzuführen. Die Operation Replace access control list (ACL) ersetzt die Zugriffskontrolllisten (ACLs) für jedes Objekt, das im Manifest aufgeführt ist. Mithilfe von können Sie definierenACLs, wer auf ein Objekt zugreifen kann und welche Aktionen sie ausführen können.

> **Note**
>
> Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Wenn

diese ACLs Option deaktiviert ist, können Sie mithilfe von Richtlinien den Zugriff auf alle Objekte in Ihrem Bucket steuern, unabhängig davon, wer die Objekte in Ihren Bucket hochgeladen hat. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

S3 Batch Operations unterstützt benutzerdefinierteACLs, die Sie definieren, und vorgefertigte ACLs Dateien, die Amazon S3 mit einem vordefinierten Satz von Zugriffsberechtigungen bereitstellt.

Wenn sich die Objekte in Ihrem Manifest in einem versionierten Bucket befinden, können Sie ACLs sie auf bestimmte Versionen jedes Objekts anwenden. Geben Sie dazu eine Versions-ID für jedes Objekt im Manifest an. Wenn Sie für kein Objekt eine Versions-ID angeben, wendet S3 Batch Operations die ACL auf die neueste Version des Objekts an.

Weitere Informationen zu ACLs Amazon S3 finden Sie unterÜbersicht über die Zugriffskontrollliste (ACL).

S3 Block Public Access

Wenn Sie den öffentlichen Zugriff auf alle Objekte in einem Bucket einschränken möchten, empfehlen wir, Amazon S3 Block Public Access anstelle von S3 Batch Operations für die Anwendung zu verwendenACLs. Block Public Access kann den öffentlichen Zugriff auf einen Bucket oder ein Konto mithilfe einer einzelnen, einfachen Operation einschränken, die sich zudem auch noch schnell umsetzen lässt. Dieses Verhalten macht Amazon S3 Block Public Access zu einer besseren Wahl, wenn Sie den öffentlichen Zugriff auf alle Objekte in einem Bucket oder Konto kontrollieren möchten. Verwenden Sie S3 Batch Operations nur, wenn Sie auf jedes Objekt im Manifest eine benutzerdefinierte ACL Option anwenden müssen. Weitere Informationen über die S3 Block Public Access finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

S3 Object Ownership

Wenn sich die Objekte im Manifest in einem Bucket befinden, der die vom Bucket-Besitzer erzwungene Einstellung für Objekteigentum verwendet, kann der Vorgang Replace access control list (ACL) nur Objekte angebenACLs, die dem Bucket-Besitzer Vollzugriff gewähren. In diesem Fall kann der Vorgang Replace access control list (ACL) anderen Personen AWS-Konten oder Gruppen keine ACL Objektberechtigungen gewähren. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

Beschränkungen und Einschränkungen

Wenn Sie Batch Operations zum Ersetzen verwendenACLs, gelten die folgenden Einschränkungen und Beschränkungen:

- Die Rolle AWS Identity and Access Management (IAM), die Sie für die Ausführung des Auftrags Replace access control list (ACL) angeben, muss über Berechtigungen verfügen, um den zugrunde liegenden Amazon S3 `PutObjectAcl` S3-Vorgang auszuführen. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter PutObjectAclin der Amazon Simple Storage Service API Reference.
- S3 Batch Operations verwendet den Amazon S3 `PutObjectAcl` S3-Vorgang, um die angegebenen ACL Werte auf jedes Objekt im Manifest anzuwenden. Daher gelten alle Einschränkungen und Beschränkungen, die für den zugrunde liegenden `PutObjectAcl` Vorgang gelten, auch für S3-Aufträge mit der Zugriffssteuerungsliste (ACL) durch Batch Operations Replace.

Objekte mit Batch Operations wiederherstellen

Sie können Amazon S3 Batch Operations verwenden, um umfangreiche Batch-Operationen an Amazon S3 S3-Objekten durchzuführen. Der Wiederherstellungsvorgang initiiert Wiederherstellungsanforderungen für die archivierten Amazon-S3-Objekte, die in Ihrem Manifest aufgeführt sind. Die folgenden archivierten Objekte müssen wiederhergestellt werden, bevor auf sie in Echtzeit zugegriffen werden kann:

- Objekte, die in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive archiviert werden
- Objekte, die über die Speicherklasse S3 Intelligent-Tiering in den Stufen Archive Access oder Deep Archive Access archiviert sind

Verwenden einer Wiederherstellung (S3InitiateRestoreObjectOperation) Die Operation in Ihrem S3 Batch Operations-Job führt zu einer `RestoreObject` Anfrage für jedes Objekt, das im Manifest angegeben ist.

> **Important**
>
> Der Wiederherstellungsauftrag initiiert nur die Anforderung zur Wiederherstellung von Objekten. S3-Batch-Vorgänge melden die Aufgabe als abgeschlossen für jedes Objekt,

nachdem die Anfrage für dieses Objekt initiiert wurde. Amazon S3 aktualisiert den Auftrag nicht und benachrichtigt Sie nicht weiter, wenn die Objekte wiederhergestellt wurden. Sie können jedoch S3-Ereignis-Benachrichtigungen verwenden, um Benachrichtigungen zu erhalten, wenn die Objekte in Amazon S3 verfügbar sind. Weitere Informationen finden Sie unter Amazon-S3-Ereignis-Benachrichtigungen.

Wenn Sie einen Wiederherstellungsauftrag erstellen, sind die folgenden Argumente verfügbar:

ExpirationInDays

Dieses Argument gibt an, wie lange das Objekt S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive in Amazon S3 verfügbar bleibt. Für Wiederherstellungsaufträge, die auf Objekte von S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive abzielen, müssen Sie einen Wert `ExpirationInDays` auf 1 oder einen höheren Wert festlegen.

> **Important**
>
> Geben Sie diese Option nicht an`ExpirationInDays`, wenn Sie Aufträge für Wiederherstellungsvorgänge erstellen, die auf Objekte der Stufen S3 Intelligent-Tiering Archive Access und Deep Archive Access abzielen. Objekte in den Archivzugriffsebenen von S3 Intelligent-Tiering unterliegen nicht dem Ablauf der Wiederherstellung, sodass die Angabe `ExpirationInDays` zu einem Anforderungsfehler führt. `RestoreObject`

GlacierJobTier

Amazon S3 kann Objekte mithilfe einer von drei verschiedenen Abrufstufen wiederherstellen: `EXPEDITED`, `STANDARD` und `BULK`. Die S3-Batchoperations-Funktion unterstützt jedoch nur die Abrufstufen `STANDARD` und `BULK`. Weitere Informationen zu den Unterschieden zwischen den Wiederherstellungsstufen finden Sie unter Die Optionen zum Abrufen von Archiven verstehen.

Weitere Informationen zu den Preisen für jede Stufe finden Sie im Abschnitt Anfragen und Datenabrufe auf der Seite Amazon S3-Preise.

Unterschiede bei der Wiederherstellung aus S3 Glacier und S3 Intelligent-Tiering

Das Wiederherstellen archivierter Dateien aus den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive unterscheidet sich von der Wiederherstellung von Dateien aus der Speicherklasse S3 Intelligent-Tiering in den Stufen Archive Access oder Deep Archive Access.

- Wenn Sie von S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive wiederherstellen, wird eine temporäre Kopie des Objekts erstellt. Amazon S3 löscht diese Kopie nachdem der Wert, den Sie im `ExpirationInDays`-Argument angegeben haben, abgelaufen ist. Nachdem diese temporäre Kopie gelöscht wurde, müssen Sie eine zusätzliche Wiederherstellungsanfrage einreichen, um auf das Objekt zugreifen zu können.
- Geben Sie bei der Wiederherstellung archivierter S3 Intelligent-Tiering-Objekte das `ExpirationInDays`-Argument nicht an. Wenn Sie ein Objekt aus den Stufen S3 Intelligent-Tiering Archive Access oder Deep Archive Access wiederherstellen, wird das Objekt zurück in die Speicherklasse S3 Intelligent-Tiering Frequent Access übergehen. Nach mindestens 90 aufeinanderfolgenden Tagen ohne Zugriff wechselt das Objekt automatisch in die Stufe „Archive Access“. Das Objekt wechselt nach mindestens 180 aufeinanderfolgenden Tagen ohne Zugriff automatisch in die Stufe „Deep Archive Access“.
- Batch-Operations-Aufträge können entweder mit Objekte der Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive oder mit Objekten der Speicherstufen S3 Intelligent-Tiering Archive Access und Deep Archive Access arbeiten. Batch Operations kann im selben Auftrag nicht für beide Typen von archivierten Objekten ausgeführt werden. Um Objekte beider Typen wiederherzustellen, müssen Sie separate Batchoperations-Aufgaben erstellen.

Überlappende Wiederherstellungen

Wenn Ihr S3InitiateRestoreObjectOperationJob versucht, ein Objekt wiederherzustellen, das bereits wiederhergestellt wird. S3 Batch Operations geht wie folgt vor.

Der Wiederherstellungsvorgang für das Objekt ist erfolgreich, wenn eine der folgenden Bedingungen erfüllt ist:

- Im Vergleich zu der bereits ausgeführten Wiederherstellungsanforderung ist der `ExpirationInDays`-Wert für diesen Auftrag identisch und sein `GlacierJobTier`-Wert ist schneller.
- Die vorherige Wiederherstellungsanforderung wurde bereits abgeschlossen und das Objekt ist derzeit verfügbar. In diesem Fall aktualisiert Batch-Vorgänge das Ablaufdatum des

wiederhergestellten Objekts so, dass es mit dem in der laufenden Wiederherstellungsanforderung angegebenen `ExpirationInDays`-Wert übereinstimmt.

Der Wiederherstellungsvorgang für das Objekt schlägt fehl, wenn eine der folgenden Bedingungen erfüllt sind:

- Die bereits ausgeführte Wiederherstellungsanforderung wurde noch nicht abgeschlossen, und die Wiederherstellungsdauer für diesen Job (angegeben durch den `ExpirationInDays` Wert) unterscheidet sich von der Wiederherstellungsdauer, die in der laufenden Wiederherstellungsanforderung angegeben ist.
- Die Wiederherstellungsstufe für diesen Auftrag (angegeben durch den `GlacierJobTier` Wert) ist dieselbe oder langsamer als die Wiederherstellungsstufe, die in der in Bearbeitung befindlichen Wiederherstellungsanforderung angegeben ist.

Einschränkungen

`S3InitiateRestoreObjectOperation`Aufträge haben die folgenden Einschränkungen:

- Sie müssen die Aufgabe in derselben Region wie die archivierten Objekte erstellen.
- S3 Batch Operations unterstützt die `EXPEDITED` Abrufstufe nicht.

Weitere Informationen zum Wiederherstellen von Objekten finden Sie unter Wiederherstellen eines archivierten Objekts.

Aufrechterhaltung der S3-Objektsperre

Sie können Amazon S3 Batch Operations verwenden, um umfangreiche Batch-Operationen an Amazon S3 S3-Objekten durchzuführen. Sie können den Aufbewahrungsvorgang Object Lock verwenden, um Aufbewahrungsdaten für Ihre Objekte festzulegen, indem Sie entweder den Governance-Modus oder den Compliance-Modus verwenden. Diese Aufbewahrungsmodi wenden unterschiedliche Schutzgrade an. Sie können einen der beiden Aufbewahrungsmodi auf jede Objektversion anwenden. Aufbewahrungsdaten verhindern wie gesetzliche Aufbewahrungsfristen, dass ein Objekt überschrieben oder gelöscht wird. Amazon S3 speichert das in den Metadaten des Objekts angegebene Aufbewahrungsdatum und schützt die angegebene Version der Objektversion, bis die Aufbewahrungsfrist abläuft.

Sie können S3 Batch Operations mit Object Lock verwenden, um die Aufbewahrungsdaten vieler Amazon S3 S3-Objekte gleichzeitig zu verwalten. Sie geben die Liste der Zielobjekte in Ihrem Manifest an und senden das Manifest zur Fertigstellung an Batch Operations. Weitere Informationen finden Sie unter S3-Objektsperre the section called “Aufbewahrungszeiträume”.

Ihr S3-Batchoperations-Auftrag mit Aufbewahrungsdaten läuft bis zum Abschluss, bis zum Abbruch oder bis ein Fehlerstatus erreicht ist. Wir empfehlen die Verwendung von S3 Batch Operations und S3 Object Lock Aufbewahrung, wenn Sie das Aufbewahrungsdatum für viele Objekte mit einer einzigen Anfrage hinzufügen, ändern oder entfernen möchten.

Batch-Vorgänge überprüft, ob die Objektsperre in Ihrem Bucket aktiviert ist, bevor Schlüssel im Manifest verarbeitet werden. Um die Operationen und die Validierung durchzuführen, benötigt Batch Operations die `s3:PutObjectRetention` Berechtigungen `s3:GetBucketObjectLockConfiguration` und in einer AWS Identity and Access Management (IAM) -Rolle, damit Batch Operations Object Lock in Ihrem Namen aufrufen kann. Weitere Informationen finden Sie unter the section called “Überlegungen zu Object Lock”.

Informationen zur Verwendung dieses Vorgangs mit der REST API finden Sie `S3PutObjectRetention` in der CreateJobBetrieb in der Amazon Simple Storage Service API Reference.

Ein AWS Command Line Interface (AWS CLI) Beispiel für die Verwendung dieses Vorgangs finden Sie unterthe section called “Batch-Vorgänge mit Objektsperrenaufbewahrung verwenden”. Ein AWS SDK for Java Beispiel finden Sie unterthe section called “Batch-Vorgänge mit Objektsperrenaufbewahrung verwenden”.

Beschränkungen und Einschränkungen

Wenn Sie Batch Operations verwenden, um Aufbewahrungsfristen für Object Lock anzuwenden, gelten die folgenden Einschränkungen und Beschränkungen:

- S3 Batch Operations nimmt keine Änderungen auf Bucket-Ebene vor.
- Versionierung und die S3-Objektsperre müssen auf dem Bucket konfiguriert werden, in dem der Auftrag ausgeführt wird.
- Alle im Manifest aufgelisteten Objekte müssen sich im selben Bucket befinden.
- Die Operation funktioniert mit der neuesten Version des Objekts, es sei denn, eine Version ist explizit im Manifest angegeben.
- Sie benötigen die `s3:PutObjectRetention` Erlaubnis in Ihrer IAM Rolle, um einen Object Lock-Aufbewahrungsjob verwenden zu können.

- Die `s3:GetBucketObjectLockConfiguration` IAM Berechtigung ist erforderlich, um zu bestätigen, dass Object Lock für den S3-Bucket aktiviert ist, für den Sie den Job ausführen.
- Sie können den Aufbewahrungszeitraum von Objekten nur verlängern, wenn `COMPLIANCE` Modus-Aufbewahrungsdaten angewendet werden. Dieser Aufbewahrungszeitraum kann nicht verkürzt werden.

S3-Objektsperre aufgrund gesetzlicher Aufbewahrungsfristen

Sie können Amazon S3 Batch Operations verwenden, um umfangreiche Batch-Operationen an Amazon S3 S3-Objekten durchzuführen. Sie können den Vorgang „Object Lock Legal Hold“ verwenden, um einer Objektversion einen Legal Hold zuzuweisen. Wie das Festlegen von Aufbewahrungszeiträumen verhindern auch rechtliche Aufbewahrungsfristen das Überschreiben oder Löschen von Objektversionen. Einer gesetzlichen Sperre ist jedoch keine Aufbewahrungsfrist zugeordnet und sie bleibt so lange wirksam, bis sie aufgehoben wird.

Sie können S3-Batch-Vorgänge mit Objektsperre verwenden, um vielen Amazon S3-Objekten gleichzeitig gesetzliche Aufbewahrungsfristen hinzuzufügen. Geben Sie dazu eine Liste der Zielobjekte in Ihrem Manifest an und senden Sie diese Liste an Batch Operations. Ihr gesetzlicher Aufbewahrungsauftrag für S3 Batch Operations Object Lock wird bis zum Abschluss, bis er abgebrochen wird oder bis ein Fehlerstatus erreicht ist, ausgeführt.

S3 Batch Operations überprüft, ob Object Lock in Ihrem S3-Bucket aktiviert ist, bevor Objekte im Manifest verarbeitet werden. Um die Objektoperationen und die Validierung auf Bucket-Ebene durchzuführen, benötigt S3 Batch Operations die Rolle `s3:PutObjectLegalHold` and `s3:GetBucketObjectLockConfiguration` in an AWS Identity and Access Management (IAM). Diese Berechtigungen ermöglichen es S3 Batch Operations, S3 Object Lock in Ihrem Namen aufzurufen.

Wenn Sie einen S3-Auftrag für Batch Operations erstellen, um einen gesetzlichen Aufbewahrungsstatus zu entfernen, müssen Sie nur den Status „Legal Hold“ angeben`Off`. Weitere Informationen finden Sie unter the section called “Überlegungen zu Object Lock”.

Informationen zur Verwendung dieses Vorgangs mit Amazon S3 REST API finden Sie `S3PutObjectLegalHold` in der CreateJobBetrieb in der Amazon Simple Storage Service API Reference.

Ein Beispiel für die Verwendung dieses Vorgangs finden Sie unterVerwendung des AWS SDK for Java.

Beschränkungen und Einschränkungen

Wenn Sie Batch Operations verwenden, um eine gesetzliche Sperre für eine Objektsperre anzuwenden oder zu entfernen, gelten die folgenden Einschränkungen und Beschränkungen:

- S3 Batch Operations nimmt keine Änderungen auf Bucket-Ebene vor.
- Alle im Manifest aufgelisteten Objekte müssen sich im selben Bucket befinden.
- Versionierung und die S3-Objektsperre müssen auf dem Bucket konfiguriert werden, in dem der Auftrag ausgeführt wird.
- Der Vorgang „Object Lock Legal Hold“ funktioniert mit der neuesten Version des Objekts, sofern nicht ausdrücklich eine Version im Manifest angegeben ist.
- Die `s3:PutObjectLegalHold` Berechtigung ist in Ihrer IAM Rolle erforderlich, um Objekten eine gesetzliche Sperre hinzuzufügen oder sie zu entfernen.
- Die `s3:GetBucketObjectLockConfiguration` IAM Berechtigung ist erforderlich, um zu bestätigen, dass S3 Object Lock für den S3-Bucket aktiviert ist, in dem der Job ausgeführt wird.

- Kopieren von Objekten
- Funktion aufrufen AWS Lambda
- Alle Objektmarkierungen ersetzen
- Alle Objektmrkierungen löschen
- Ersetzt die Zugriffskontrollliste (ACL)
- Objekte mit Batch Operations wiederherstellen
- Aufrechterhaltung der S3-Objektsperre
- S3-Objektsperre aufgrund gesetzlicher Aufbewahrungsfristen
- Replizieren vorhandener Objekte mit Batch Replication

Verwalten von S3-Batch-Vorgangsaufträgen

Amazon S3 bietet robuste Tools, die Sie bei der Verwaltung Ihrer Aufträge für die S3-Batch-Vorgänge unterstützen, nachdem Sie diese erstellt haben. In diesem Abschnitt werden die Vorgänge beschrieben, mit denen Sie Ihre Jobs mithilfe der Amazon S3-Konsole AWS Command Line Interface (AWS CLI) oder Amazon S3 verwalten und verfolgen können RESTAPI. AWS SDKs

Themen

- Verwenden der Amazon-S3-Konsole zum Verwalten Ihrer S3-Batch-Vorgangsaufträge
- Auflisten von Aufträgen
- Anzeigen von Auftragsdetails
- Zuweisen der Auftragspriorität

Verwenden der Amazon-S3-Konsole zum Verwalten Ihrer S3-Batch-Vorgangsaufträge

Verwenden der Konsole zum Verwalten Ihrer S3-Batch-Vorgangsaufträge Beispielsweise ist Folgendes möglich:

- Aktive und in Warteschlange gewartete Aufträge anzeigen
- Ändern der Priorität eines Auftrags
- Bestätigen und Ausführen eines Auftrags
- Einen Auftrag klonen
- Abbrechen eines Auftrags

Verwenden der Konsole zum Verwalten Ihrer Batchvorgänge

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Klicken Sie im linken Navigationsbereich auf Batchvorgänge.
3. Wählen Sie den spezifischen Auftrag aus, den Sie verwalten möchten.

Auflisten von Aufträgen

Sie können eine Liste Ihrer S3-Batch-Vorgangsaufträge abrufen. Die Liste umfasst Aufträge, die noch nicht abgeschlossen sind, sowie Aufträge, die in den letzten 90 Tagen abgeschlossen wurden. Die Auftragsliste enthält Informationen zu jedem Auftrag, z. B. die ID, die Beschreibung, die Priorität, den aktuellen Status und die Anzahl an Aufgaben, die erfolgreich durchgeführt wurden oder fehlgeschlagen sind. Sie können die Auftragsliste nach dem Status filtern. Wenn Sie eine Auftragsliste über die Konsole abrufen, können Sie Ihre Aufträge auch nach Beschreibung oder ID durchsuchen und nach AWS-Region filtern.

Holen Sie sich eine Liste von **`Active`** und **`Complete`** Jobs

Im folgenden AWS CLI Beispiel wird eine Liste von `Active` `Complete` Und-Jobs abgerufen. Um dieses Beispiel zu verwenden, ersetzen Sie das *`Platzhalter für Benutzereingaben`* mit Ihren eigenen Informationen.

```
aws s3control list-jobs \
    --region us-west-2 \
    --account-id account-id \
    --job-statuses '["Active","Complete"]' \
    --max-results 20
```

Weitere Informationen und Beispiele finden Sie unter list-jobsin der AWS CLI Befehlsreferenz.

Anzeigen von Auftragsdetails

Wenn Sie mehr Informationen zu einem Amazon S3 Batch Operations-Job benötigen, als Sie durch das Auflisten von Aufträgen abrufen können, können Sie alle Details für einen einzelnen Job anzeigen. Sie können Details zu Aufträgen anzeigen, die noch nicht abgeschlossen sind, oder zu Aufträgen, die in den letzten 90 Tagen abgeschlossen wurden. Neben den Angaben, die in einer Auftragsliste enthalten sind, umfassen die Details zu einem einzelnen Auftrag auch noch andere Informationen, wie z. B:

- Die Betriebsparameter
- Details zum Manifest
- Informationen zum Abschlussbericht (falls Sie einen konfiguriert haben, als Sie den Auftrag erstellt haben)
- Der Amazon-Ressourcenname (ARN) der Benutzerrolle AWS Identity and Access Management (IAM), die Sie zur Ausführung des Jobs zugewiesen haben

Durch das Anzeigen der Details eines einzelnen Auftrags erhalten Sie Zugriff auf die gesamte Konfiguration eines Auftrags. Um die Details eines Jobs anzuzeigen, können Sie die Amazon S3 S3-Konsole oder die AWS Command Line Interface (AWS CLI) verwenden.

Die Beschreibung eines Auftrags in S3 Batch Operations in der Amazon-S3-Konsole abrufen

So zeigen Sie die Beschreibung eines Batch-Operations-Auftrags über die Konsole an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Klicken Sie im linken Navigationsbereich auf Batchvorgänge.
3. Wählen Sie die Auftrags-ID des spezifischen Auftrags aus, um die zugehörigen Details anzuzeigen.

Holen Sie sich eine Stellenbeschreibung für S3 Batch Operations in der AWS CLI

Im folgenden Beispiel wird die Beschreibung eines Auftrags in S3 Batch Operations unter Verwendung der AWS CLI abgerufen. Um den folgenden Beispielbefehl zu verwenden, ersetzen Sie *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
aws s3control describe-job \
--region us-west-2 \
--account-id account-id \
--job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c
```

Weitere Informationen und Beispiele finden Sie unter describe-jobin der AWS CLI Befehlsreferenz.

Zuweisen der Auftragspriorität

Sie können jedem Amazon S3 Batch Operations-Job eine numerische Priorität zuweisen, bei der es sich um eine beliebige positive Ganzzahl handeln kann. S3 Batch Operations priorisiert Jobs entsprechend der zugewiesenen Priorität. Aufträge mit höherer Priorität (oder einem höheren Wert für den Prioritätsparameter) werden zuerst bewertet. Die Priorität wird in absteigender Reihenfolge ermittelt. So erhält beispielsweise eine Auftragswarteschlange mit einem Prioritätswert von 10 Vorrang vor einer Auftragswarteschlange mit dem Wert 1.

Sie können die Priorität eines Jobs ändern, während der Job ausgeführt wird. Wenn Sie einen neuen Auftrag übermitteln, während ein Auftrag ausgeführt wird, kann der Auftrag mit niedrigerer Priorität pausieren und der Auftrag mit höherer Priorität ausgeführt werden.

Das Ändern der Priorität eines Jobs hat keinen Einfluss auf die Verarbeitungsgeschwindigkeit des Jobs.

> **Note**
>
> S3 Batch Operations berücksichtigt Jobprioritäten nach bestem Wissen und Gewissen. Obwohl Jobs mit höheren Prioritäten generell Vorrang vor Aufträgen mit niedrigerer Priorität haben, garantiert Amazon S3 keine strikte Reihenfolge der Jobs.

Verwenden der S3-Konsole

So aktualisieren Sie die Auftragspriorität in der Amazon S3 S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Klicken Sie im linken Navigationsbereich auf Batchvorgänge.
3. Wählen Sie den spezifischen Auftrag aus, den Sie verwalten möchten.
4. Wählen Sie Action (Aktion). Wählen Sie in der Dropdown-Liste Update priority (Priorität aktualisieren).

Verwenden Sie den AWS CLI

Im folgenden Beispiel wird die Auftragspriorität mithilfe von aktualisiert AWS CLI. Eine höhere Zahl weist auf eine höhere Ausführungspriorität hin. Um den folgenden Beispielbefehl zu verwenden, ersetzen Sie den *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
aws s3control update-job-priority \
    --region us-west-2 \
    --account-id account-id \
    --priority 98 \
    --job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c
```

Mit dem AWS SDK for Java

Im folgenden Beispiel wird die Priorität eines S3-Batch-Vorgangsauftrags unter Verwendung des AWS SDK for Java aktualisiert.

Weitere Informationen zur Auftragspriorität finden Sie unter Zuweisen der Auftragspriorität.

Example

```
package aws.example.s3control;


import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.UpdateJobPriorityRequest;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class UpdateJobPriority {
    public static void main(String[] args) {
        String accountId = "Account ID";
        String jobId = "00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c";

        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.updateJobPriority(new UpdateJobPriorityRequest()
                    .withAccountId(accountId)
                    .withJobId(jobId)
                    .withPriority(98));

        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Verfolgen von Auftragsstatus- und Abschluss

Mit S3 Batch Operations können Sie den Auftragsstatus anzeigen und aktualisieren, Benachrichtigungen und Protokollierung hinzufügen, Auftragsfehler verfolgen und Abschlussberichte erstellen.

Themen

- Auftragsstatuses
- Auftragsstatus wird aktualisiert
- Benachrichtigungen und Protokollierung
- Nachverfolgung von Auftragsfehlern
- Abschlussberichte
- Beispiele: Verfolgen eines S3 Batch Operations-Jobs in Amazon EventBridge über AWS CloudTrail
- Beispiele: Abschlussberichte zu S3-BatchVorgänge

Auftragsstatuses

Nachdem Sie einen Auftrag erstellt und ausgeführt haben, durchläuft dieser eine Reihe von Status. In der folgenden Tabelle werden der Status und die möglichen Übergänge zwischen ihnen beschrieben.

Status	Beschreibung	Übergänge
`New`	Ein Auftrag beginnt nach der Erstellung mit dem Status `New`.	Ein Auftrag wechselt automatisch in den Status `Preparing` , sobald Amazon S3 mit der Verarbeitung des Manifestobjekts beginnt.
`Preparing`	Amazon S3 verarbeitet das Manifestobjekt und andere Auftragsparameter, um den Auftrag einzurichten und auszuführen.	Ein Auftrag wechselt automatisch in den Status `Ready`, sobald Amazon S3 die Verarbeitung des Manifests und anderer Parameter abschließt. Der Job ist dann bereit, mit der Ausführung des angegebenen Vorgangs für

Status	Beschreibung	Übergänge
		die im Manifest aufgelisteten Objekte zu beginnen. Wenn der Auftrag vor der Ausführung bestätigt werden muss, wie dies bei der Erstellung eines Auftrags mit der Amazon-S3-Konsole der Fall ist, wechselt der Auftrag von `Preparing` zu `Suspended` . Der Auftrag verbleibt im Status `Suspended` , bis Sie bestätigen, dass Sie ihn ausführen möchten.
`Suspended`	Der Job muss bestätigt werden, aber Sie haben noch nicht bestätigt, dass Sie ihn ausführen möchten. Nur Aufträge, die mit der Amazon-S3-Konsole erstellt wurden, benötigen eine Bestätigung. Ein Job, der mit der Konsole erstellt wurde, wechselt unmittelbar danach in den `Suspended` Status`Preparing` . Nachdem Sie die Ausführung des Auftrags bestätigen und der Auftrag in den Status `Ready` wechselt, kehrt er nicht wieder in den Status `Suspended` zurück.	Nachdem Sie die Ausführung des Auftrags bestätigen, wechselt der Status zu `Ready`.

Status	Beschreibung	Übergänge
`Ready`	Amazon S3 ist bereit, die angeforderten Objekt-Vorgänge auszuführen.	Ein Auftrag wechselt automatisch in den Status `Active`, sobald Amazon S3 mit dessen Ausführung beginnt. Der Zeitraum, den ein Auftrag im Status `Ready` verbringt, hängt davon ab, ob bereits Aufträge mit einer höheren Priorität ausgeführt werden und wie lange es dauert, diese Aufträge abzuschließen.
`Active`	Amazon S3 führt die angeforderte Operation an den im Manifest aufgeführten Objekten aus. Während ein Auftrag ausgeführt wird`Active`, können Sie seinen Fortschritt mit der Amazon S3 S3-Konsole oder den `DescribeJob` Vorgang über REST API AWS CLI, oder überwachen AWS SDKs.	Ein Job verlässt den `Active` Status, in dem für den Job keine Operationen mehr an Objekten ausgeführt werden. Dieses Verhalten kann automatisch auftreten, z. B. wenn ein Job erfolgreich abgeschlossen wird oder fehlschlägt. Dieses Verhalten kann auch als Folge von Benutzeraktionen auftreten, z. B. durch das Abbrechen eines Jobs. Der neue Status des Auftrags hängt vom Grund für den Übergang ab.
`Pausing`	Der Auftrag wechselt von `Paused` zu einem anderen Status über.	Ein Auftrag wechselt automatisch in den Status `Paused`, wenn die Stufe `Pausing` abgeschlossen ist.

Status	Beschreibung	Übergänge
`Paused`	Ein Auftrag wechselt in den Status `Paused`, wenn ein anderer Auftrag mit einer höheren Priorität abgesendet wird, während der aktuelle Auftrag ausgeführt wird.	Ein Auftrag mit dem Status `Paused` kehrt automatisch in den Status `Active` zurück, nachdem alle Aufträge mit einer höheren Priorität, die die Ausführung des Auftrag blockierten, abgeschlossen, fehlgeschlagen oder gesperrt sind.
`Complete`	Der Auftrag hat die Ausführung der angeforderten Operation an allen Objekten abgeschlossen. Die Operation kann bei den einzelnen Objekten entweder erfolgreich abgeschlossen oder fehlgeschlagen sein. Wenn Sie den Auftrag so konfiguriert haben, dass ein Abschlussbericht generiert werden soll, dann ist der Bericht sofort verfügbar, nachdem der Auftrag in den Status gewechselt is `Complete`.	`Complete` ist ein Beendigungsstatus. Sobald ein Job erreicht ist`Complete`, wechselt er nicht mehr in einen anderen Status.
`Cancelling`	Der Auftrag wechselt in den Status `Cancelled` .	Ein Auftrag wechselt automatisch in den Status `Cancelled` , wenn die Stufe `Cancelling` abgeschlossen ist.

Status	Beschreibung	Übergänge
`Cancelled`	Sie haben beantragt, dass der Job storniert wird, und S3 Batch Operations hat den Job erfolgreich storniert. Der Job sendet keine neuen Anfragen an Amazon S3.	`Cancelled` ist ein Beendigungsstatus. Sobald ein Job erreicht ist`Cancelled`, wird er nicht in einen anderen Status übergehen.
`Failing`	Der Auftrag wechselt in den Status `Failed`.	Ein Auftrag wechselt automatisch in den Status `Failed`, nachdem die Stufe `Failing` abgeschlossen ist.
`Failed`	Der Auftrag ist fehlgesch lagen und wird nicht länger ausgeführt. Weitere Informati onen zu Auftragsfehlern finden Sie unter Nachverfolgung von Auftragsfehlern.	`Failed` ist ein Beendigun gsstatus. Sobald ein Job das Land erreicht hat`Failed`, wird er nicht mehr in einen anderen Status übergehen.

Auftragsstatus wird aktualisiert

Im Folgenden AWS CLI und in den AWS SDK for Java Beispielen wird der Status eines Batch Operations-Jobs aktualisiert. Weitere Informationen zur Verwendung der Amazon S3 S3-Konsole zur Verwaltung von Batch Operationsaufträgen finden Sie unterVerwenden der Amazon-S3-Konsole zum Verwalten Ihrer S3-Batch-Vorgangsaufträge.

Verwenden von AWS CLI

Um die folgenden Beispielbefehle zu verwenden, ersetzen Sie den *`Platzhalter für Benutzereingaben`* mit Ihren eigenen Informationen.

- Wenn Sie den `--no-confirmation-required` Parameter in Ihrem `create-job` Befehl nicht angegeben haben, bleibt der Job so lange unterbrochen, bis Sie den Job bestätigen, indem Sie seinen Status auf setzen`Ready`. Amazon S3 erlaubt dann die Ausführung des Auftrags.

```
aws s3control update-job-status \
```

```
    --region us-west-2 \
    --account-id 123456789012 \
    --job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c \
    --requested-job-status 'Ready'
```

- Brechen Sie den Auftrag ab, indem Sie den Auftragsstatus auf einstelle `Cancelled`.

```
aws s3control update-job-status \
     --region us-west-2 \
     --account-id 123456789012 \
     --job-id 00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c \
     --status-update-reason "No longer needed" \
     --requested-job-status Cancelled
```

Verwendung des AWS SDK for Java

Das folgende Beispiel aktualisiert den Status eines S3 Batch Operations-Jobs mithilfe von AWS SDK for Java.

Weitere Informationen zum Auftragsstatus finden Sie unter Verfolgen von Auftragsstatus- und Abschluss.

Example

```
package aws.example.s3control;


import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.UpdateJobStatusRequest;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class UpdateJobStatus {
    public static void main(String[] args) {
        String accountId = "Account ID";
        String jobId = "00e123a4-c0d8-41f4-a0eb-b46f9ba5b07c";
```

```
        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.updateJobStatus(new UpdateJobStatusRequest()
                    .withAccountId(accountId)
                    .withJobId(jobId)
                    .withRequestedJobStatus("Ready"));

        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Benachrichtigungen und Protokollierung

Sie können nicht nur Abschlussberichte anfordern, sondern auch die Aktivitäten von Batch Operations erfassen, überprüfen und prüfen, indem Sie AWS CloudTrail Da Batch Operations bestehende Amazon S3 API S3-Operationen zur Ausführung von Aufgaben verwendet, geben diese Aufgaben auch dieselben Ereignisse aus, als würden Sie sie direkt aufrufen. Daher können Sie den Fortschritt Ihres Auftrags und all seiner Aufgaben verfolgen und aufzeichnen, indem Sie dieselben Tools und Prozesse für Benachrichtigungen, Protokollierung und Prüfung verwenden, die Sie bereits mit Amazon S3 verwenden. Weitere Informationen finden Sie in den Beispielen in den folgenden Abschnitten.

> **Note**
>
> Batch Operations generiert CloudTrail während der Auftragsausführung sowohl Verwaltungs- als auch Datenereignisse. Das Volumen dieser Ereignisse hängt von der Anzahl der Schlüssel im Manifest der einzelnen Jobs ab. Weitere Informationen finden Sie auf der Seite mit den CloudTrail Preisen. Dort finden Sie Beispiele dafür, wie sich die Preise je nach der Anzahl der Trails, die Sie in Ihrem Konto konfiguriert haben, ändern. Informationen

zum Konfigurieren und Protokollieren von Ereignissen, damit sie Ihren Anforderungen entsprechen, finden Sie unter Create your first trail (Erstellen Ihres ersten Trails) im AWS CloudTrail -Benutzerhandbuch.

Weitere Informationen über Amazon-S3-Ereignisse finden Sie unter Amazon-S3-Ereignis-Benachrichtigungen.

Nachverfolgung von Auftragsfehlern

Wenn bei einem S3-Batch-Vorgangsauftrag ein Problem auftritt, das die erfolgreiche Ausführung verhindert, schlägt der Auftrag fehl. Dies ist beispielsweise der Fall, wenn das angegebene Manifest nicht gelesen werden kann. Wenn ein Auftrag fehlschlägt, generiert er einen oder mehrere Fehlercodes oder Fehlergründe. S3 Batch Operations speichert die Fehlercodes und Gründe zusammen mit dem Job, sodass Sie sie einsehen können, indem Sie die Auftragsdetails anfordern. Wenn Sie einen Abschlussbericht für den Auftrag angefordert haben, sind die Fehlercodes und -ursachen auch darin enthalten.

Um zu verhindern, dass zu viele Vorgänge fehlschlagen, wendet Amazon S3 einen Schwellenwert für fehlgeschlagene Aufträge auf jeden Batch-Vorgangsauftrag an. Wenn ein Job mindestens 1.000 Aufgaben ausgeführt hat, überwacht Amazon S3 die Rate der Aufgabenausfälle. Wenn die Fehlerrate (also die Anzahl fehlgeschlagener Aufgaben im Verhältnis zur Gesamtzahl ausgeführter Aufgaben) zu einem beliebigen Zeitpunkt 50 Prozent überschreitet, schlägt der Auftrag fehl. Wenn Ihr Auftrag fehlschlägt, weil er den Schwellenwert für fehlgeschlagene Aufträge überschritten hat, können Sie die Ursache der Fehler identifizieren. Beispielsweise könnte der Fehler darin liegen, dass Sie versehentlich Objekte in das Manifest aufgenommen haben, die im angegebenen Bucket nicht vorhanden sind. Nach dem Beheben der Fehler können Sie den Auftrag erneut übermitteln.

Note

S3 Batch Operations arbeitet asynchron und die Aufgaben werden nicht unbedingt in der Reihenfolge ausgeführt, in der die Objekte im Manifest aufgeführt sind. Daher können Sie anhand der Reihenfolge im Manifest nicht feststellen, welche Objektaufgaben erfolgreich ausgeführt wurden und welche fehlgeschlagen sind. Stattdessen können Sie den Abschlussbericht des Auftrags (falls Sie einen angefordert haben) oder Ihre AWS CloudTrail Ereignisprotokolle einsehen, um die Ursache der Fehler zu ermitteln.

Abschlussberichte

Wenn Sie einen Auftrag erstellen, können Sie einen Abschlussbericht anfordern. Solange S3 Batch Operations erfolgreich mindestens eine Aufgabe aufruft, generiert Amazon S3 einen Abschlussbericht, nachdem der Job laufende Aufgaben abgeschlossen hat, fehlschlägt oder abgebrochen wurde. Sie können den Abschlussbericht so konfigurieren, dass er alle oder nur fehlgeschlagene Aufgaben enthält.

Der Abschlussbericht enthält die Auftragskonfiguration, den Status und Informationen für jede Aufgabe, einschließlich Objektschlüssel und Version, Status, Fehlercodes und Beschreibungen aller Fehler. Abschlussberichte bieten eine einfache Möglichkeit, die Ergebnisse von Aufgaben in einem konsolidierten Format ohne zusätzliche Einrichtung anzuzeigen. Abschlussberichte werden mithilfe serverseitiger Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt. Ein Beispiel für einen Abschlussbericht finden Sie unter Beispiele: Abschlussberichte zu S3-BatchVorgänge.

Wenn Sie keinen Abschlussbericht konfigurieren, können Sie Ihren Job und seine Aufgaben trotzdem mithilfe von Amazon überwachen CloudTrail und prüfen CloudWatch. Weitere Informationen finden Sie unter den folgenden Themen:

Themen

- Beispiele: Verfolgen eines S3 Batch Operations-Jobs in Amazon EventBridge über AWS CloudTrail
- Beispiele: Abschlussberichte zu S3-BatchVorgänge

Beispiele: Verfolgen eines S3 Batch Operations-Jobs in Amazon EventBridge über AWS CloudTrail

Die Auftragsaktivität von Amazon S3-Batch-Vorgänge wird in Form von Ereignissen in aufgezeichne AWS CloudTrail. Sie können eine benutzerdefinierte Regel in Amazon erstellen EventBridge und diese Ereignisse an die Zielbenachrichtigungsressource Ihrer Wahl senden, z. B. Amazon Simple Notification Service (AmazonSNS).

> **Note**
>
> Amazon EventBridge ist die bevorzugte Methode, um Ihre Veranstaltungen zu verwalten. Amazon CloudWatch Events und EventBridge sind derselbe zugrunde liegende Service und EventBridge bieten API jedoch mehr Funktionen. Änderungen, die Sie in einer der

beiden CloudWatch oder in jeder Konsole vornehmen, EventBridge werden in jeder Konsole angezeigt. Weitere Informationen finden Sie im EventBridge Amazon-Benutzerhandbuch.

Tracking-Beispiele

- S3 Batch Operations-Ereignisse, aufgezeichnet in CloudTrail
- EventBridge Regel für die Nachverfolgung von S3 Batch Operations-Auftragsereignissen

S3 Batch Operations-Ereignisse, aufgezeichnet in CloudTrail

Wenn ein Batch Operations-Job erstellt wird, wird er als `JobCreated` Ereignis in aufgezeichnet CloudTrail. Wenn der Job ausgeführt wird, ändert sich sein Status während der Verarbeitung, und andere `JobStatusChanged` Ereignisse werden in aufgezeichnet CloudTrail. Sie können diese Ereignisse auf der CloudTrail Konsole anzeigen. Weitere Informationen zu CloudTrail finden Sie im AWS CloudTrail Benutzerhandbuch.

> **Note**
>
> Es werden nur `status-change` Auftragsereignisse von S3 Batch Operations aufgezeichnet CloudTrail.

Example — Ereignis zum Abschluss eines S3 Batch Operations-Auftrags, aufgezeichnet von CloudTrail

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "accountId": "123456789012",
        "invokedBy": "s3.amazonaws.com"
    },
    "eventTime": "2020-02-05T18:25:30Z",
    "eventSource": "s3.amazonaws.com",
    "eventName": "JobStatusChanged",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "s3.amazonaws.com",
    "userAgent": "s3.amazonaws.com",
    "requestParameters": null,
```

```
    "responseElements": null,
    "eventID": "f907577b-bf3d-4c53-b9ed-8a83a118a554",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "123412341234",
    "serviceEventDetails": {
        "jobId": "d6e58ec4-897a-4b6d-975f-10d7f0fb63ce",
        "jobArn": "arn:aws:s3:us-west-2:181572960644:job/
d6e58ec4-897a-4b6d-975f-10d7f0fb63ce",
        "status": "Complete",
        "jobEventId": "b268784cf0a66749f1a05bce259804f5",
        "failureCodes": [],
        "statusChangeReason": []
    }
}
```

EventBridge Regel für die Nachverfolgung von S3 Batch Operations-Auftragsereignissen

Das folgende Beispiel zeigt, wie Sie in Amazon eine Regel erstellen EventBridge , um S3 Batch Operations-Ereignisse AWS CloudTrail zu erfassen, die von einem Ziel Ihrer Wahl aufgezeichnet wurden.

Dazu erstellen Sie eine Regel, indem Sie alle Schritte unter Erstellen von EventBridge Regeln, die auf Ereignisse reagieren, befolgen. Fügen Sie gegebenenfalls die folgende benutzerdefinierte S3-Batchoperations-Ereignismuster-Richtlinie ein und wählen Sie einen Zielservice aus.

Benutzerdefinierte S3-Batchoperationen-Ereignismusterrichtlinie

```
{
    "source": [
        "aws.s3"
    ],
    "detail-type": [
        "AWS Service Event via CloudTrail"
    ],
    "detail": {
        "eventSource": [
            "s3.amazonaws.com"
        ],
        "eventName": [
            "JobCreated",
            "JobStatusChanged"
        ]
```

```
    }
}
```

Die folgenden Beispiele sind zwei Batch Operations-Ereignisse, die von einer EventBridge Ereignisregel an Amazon Simple Queue Service (AmazonSQS) gesendet wurden. Ein Batchoperations-Auftrag durchläuft viele verschiedene Zustände während der Verarbeitung (`New`, `Preparing`, `Active` usw.), sodass Sie davon ausgehen können, dass Sie mehrere Nachrichten für jeden Auftrag erhalten.

Example — JobCreated Beispiel für ein Ereignis

```
{
    "version": "0",
    "id": "51dc8145-541c-5518-2349-56d7dffdf2d8",
    "detail-type": "AWS Service Event via CloudTrail",
    "source": "aws.s3",
    "account": "123456789012",
    "time": "2020-02-27T15:25:49Z",
    "region": "us-east-1",
    "resources": [],
    "detail": {
        "eventVersion": "1.05",
        "userIdentity": {
            "accountId": "11112223334444",
            "invokedBy": "s3.amazonaws.com"
        },
        "eventTime": "2020-02-27T15:25:49Z",
        "eventSource": "s3.amazonaws.com",
        "eventName": "JobCreated",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "s3.amazonaws.com",
        "userAgent": "s3.amazonaws.com",
        "eventID": "7c38220f-f80b-4239-8b78-2ed867b7d3fa",
        "readOnly": false,
        "eventType": "AwsServiceEvent",
        "serviceEventDetails": {
            "jobId": "e849b567-5232-44be-9a0c-40988f14e80c",
            "jobArn": "arn:aws:s3:us-east-1:181572960644:job/
e849b567-5232-44be-9a0c-40988f14e80c",
            "status": "New",
            "jobEventId": "f177ff24f1f097b69768e327038f30ac",
            "failureCodes": [],
            "statusChangeReason": []
```

```
        }
    }
}
```

Example — JobStatusChanged Ereignis zum Abschluss des Auftrags

```
{
  "version": "0",
  "id": "c8791abf-2af8-c754-0435-fd869ce25233",
  "detail-type": "AWS Service Event via CloudTrail",
  "source": "aws.s3",
  "account": "123456789012",
  "time": "2020-02-27T15:26:42Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "eventVersion": "1.05",
    "userIdentity": {
      "accountId": "1111222233334444",
      "invokedBy": "s3.amazonaws.com"
    },
    "eventTime": "2020-02-27T15:26:42Z",
    "eventSource": "s3.amazonaws.com",
    "eventName": "JobStatusChanged",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "s3.amazonaws.com",
    "userAgent": "s3.amazonaws.com",
    "eventID": "0238c1f7-c2b0-440b-8dbd-1ed5e5833afb",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "serviceEventDetails": {
      "jobId": "e849b567-5232-44be-9a0c-40988f14e80c",
      "jobArn": "arn:aws:s3:us-east-1:181572960644:job/
e849b567-5232-44be-9a0c-40988f14e80c",
      "status": "Complete",
      "jobEventId": "51f5ac17dba408301d56cd1b2c8d1e9e",
      "failureCodes": [],
      "statusChangeReason": []
    }
  }
}
```

Beispiele: Abschlussberichte zu S3-BatchVorgänge

Wenn Sie einen S3-Batchoperations-Auftrag erstellen, können Sie einen Abschlussbericht für alle oder nur für die fehlgeschlagenen Aufträge anfordern. Sofern mindestens eine Aufgabe erfolgreich aufgerufen wurde, generiert S3-Batch-Vorgänge einen Bericht für abgeschlossene, fehlgeschlagene oder abgebrochene Aufträge.

Der Abschlussbericht enthält zusätzliche Informationen zu jeder Aufgabe, darunter den Namen und die Version des Objektschlüssels, Status, Fehlercodes sowie Beschreibungen zu etwaigen Fehlern. Die Beschreibung der Fehler für jede fehlgeschlagene Aufgabe kann herangezogen werden, um Probleme im Rahmen der Auftragserstellung zu diagnostizieren, beispielsweise mit Berechtigungen.

> **Note**
>
> Abschlussberichte werden immer mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt.

Example — Manifest-Ergebnisdatei der obersten Ebene

Die `manifest.json`-Datei der höchsten Ebene enthält die Position jedes Erfolgsberichts und (sofern Fehler in der Auftragsabwicklung aufgetreten sind) jedes Fehlerberichts (siehe folgendes Beispiel).

```
{
    "Format": "Report_CSV_20180820",
    "ReportCreationDate": "2019-04-05T17:48:39.725Z",
    "Results": [
        {
            "TaskExecutionStatus": "succeeded",
            "Bucket": "my-job-reports",
            "MD5Checksum": "83b1c4cbe93fc893f54053697e10fd6e",
            "Key": "job-f8fb9d89-a3aa-461d-bddc-ea6a1b131955/
results/6217b0fab0de85c408b4be96aeaca9b195a7daa5.csv"
        },
        {
            "TaskExecutionStatus": "failed",
            "Bucket": "my-job-reports",
            "MD5Checksum": "22ee037f3515975f7719699e5c416eaa",
            "Key": "job-f8fb9d89-a3aa-461d-bddc-ea6a1b131955/results/
b2ddad417e94331e9f37b44f1faf8c7ed5873f2e.csv"
```

```
        }
    ],
    "ReportSchema": "Bucket, Key, VersionId, TaskStatus, ErrorCode, HTTPStatusCode,
 ResultMessage"
}
```

Example — Berichte über fehlgeschlagene Aufgaben

Berichte zu fehlgeschlagenen Aufgaben enthalten die folgenden Informationen für alle fehlgeschlagenen Aufgaben:

- `Bucket`
- `Key`
- `VersionId`
- `TaskStatus`
- `ErrorCode`
- `HTTPStatusCode`
- `ResultMessage`

Der folgende Beispielbericht zeigt einen Fall, in dem bei der AWS Lambda Funktion ein Timeout aufgetreten ist, wodurch Fehler den Schwellenwert für Fehler überschritten haben. Deshalb wurde die Kennzeichnung `PermanentFailure` zugewiesen.

```
amzn-s3-demo-bucket1,image_14975,,failed,200,PermanentFailure,"Lambda returned
 function error: {""errorMessage"":""2019-04-05T17:35:21.155Z 2845ca0d-38d9-4c4b-
abcf-379dc749c452 Task timed out after 3.00 seconds""}"
amzn-s3-demo-bucket1,image_15897,,failed,200,PermanentFailure,"Lambda returned
 function error: {""errorMessage"":""2019-04-05T17:35:29.610Z 2d0a330b-de9b-425f-
b511-29232fde5fe4 Task timed out after 3.00 seconds""}"
amzn-s3-demo-bucket1,image_14819,,failed,200,PermanentFailure,"Lambda returned
 function error: {""errorMessage"":""2019-04-05T17:35:22.362Z fcf5efde-74d4-4e6d-b37a-
c7f18827f551 Task timed out after 3.00 seconds""}"
amzn-s3-demo-bucket1,image_15930,,failed,200,PermanentFailure,"Lambda returned
 function error: {""errorMessage"":""2019-04-05T17:35:29.809Z 3dd5b57c-4a4a-48aa-8a35-
cbf027b7957e Task timed out after 3.00 seconds""}"
amzn-s3-demo-bucket1,image_17644,,failed,200,PermanentFailure,"Lambda
 returned function error: {""errorMessage"":""2019-04-05T17:35:46.025Z
 10a764e4-2b26-4d8c-9056-1e1072b4723f Task timed out after 3.00 seconds""}"
```

```
amzn-s3-demo-bucket1,image_17398,,failed,200,PermanentFailure,"Lambda returned
 function error: {""errorMessage"":""2019-04-05T17:35:44.661Z 1e306352-4c54-4eba-
aee8-4d02f8c0235c Task timed out after 3.00 seconds""}"
```

Example — Bericht über erfolgreiche Aufgaben

Berichte über erfolgreiche Aufgaben enthalten die folgenden Informationen zu den erfolgreichen Aufgaben:

- `Bucket`
- `Key`
- `VersionId`
- `TaskStatus`
- `ErrorCode`
- `HTTPStatusCode`
- `ResultMessage`

Im folgenden Beispiel hat die Lambda-Funktion das Amazon S3-Objekt in einen anderen Bucket kopiert. Die zurückgegebene Amazon S3-Antwort wird an S3-Batch-Vorgänge zurückgegeben und dann in den endgültigen Abschlussbericht geschrieben.

```
amzn-s3-demo-bucket1,image_17775,,succeeded,200,,"{u'CopySourceVersionId':
 'xVR78haVKlRnurYofbTfYr3ufYbktF8h', u'CopyObjectResult': {u'LastModified':
 datetime.datetime(2019, 4, 5, 17, 35, 39, tzinfo=tzlocal()), u'ETag':
 '""fe66f4390c50f29798f040d7aae72784""'}, 'ResponseMetadata': {'HTTPStatusCode':
 200, 'RetryAttempts': 0, 'HostId': 'nXNaClIMxEJzWNmeMNQV2KpjbaCJLn0OGoXWZpuVOFS/
iQYWxb3QtTvzX9SVfx2lA3oTKLwImKw=', 'RequestId': '3ED5852152014362', 'HTTPHeaders':
 {'content-length': '234', 'x-amz-id-2': 'nXNaClIMxEJzWNmeMNQV2KpjbaCJLn0OGoXWZpuVOFS/
iQYWxb3QtTvzX9SVfx2lA3oTKLwImKw=', 'x-amz-copy-source-version-id':
 'xVR78haVKlRnurYofbTfYr3ufYbktF8h', 'server': 'AmazonS3', 'x-amz-request-id':
 '3ED5852152014362', 'date': 'Fri, 05 Apr 2019 17:35:39 GMT', 'content-type':
 'application/xml'}}}"
amzn-s3-demo-bucket1,image_17763,,succeeded,200,,"{u'CopySourceVersionId':
 '6HjOUSim4Wj6BTcbxToXW44pSZ.40pwq', u'CopyObjectResult': {u'LastModified':
 datetime.datetime(2019, 4, 5, 17, 35, 39, tzinfo=tzlocal()),
 u'ETag': '""fe66f4390c50f29798f040d7aae72784""'}, 'ResponseMetadata':
 {'HTTPStatusCode': 200, 'RetryAttempts': 0, 'HostId': 'GiCZNYr8LHd/
Thyk6beTRP96IGZk2sYxujLe13TuuLpq6U2RD3we0YoluuIdm1PRvkMwnEW1aFc=', 'RequestId':
 '1BC9F5B1B95D7000', 'HTTPHeaders': {'content-length': '234', 'x-amz-id-2':
```

```
 'GiCZNYr8LHd/Thyk6beTRP96IGZk2sYxujLe13TuuLpq6U2RD3we0YoluuIdm1PRvkMwnEW1aFc=', 'x-
amz-copy-source-version-id': '6HjOUSim4Wj6BTcbxToXW44pSZ.40pwq', 'server': 'AmazonS3',
 'x-amz-request-id': '1BC9F5B1B95D7000', 'date': 'Fri, 05 Apr 2019 17:35:39 GMT',
 'content-type': 'application/xml'}}}"
amzn-s3-demo-bucket1,image_17860,,succeeded,200,,"{u'CopySourceVersionId':
 'm.MDD0g_QsUnYZ8TBzVFrp.TmjN8PJyX', u'CopyObjectResult': {u'LastModified':
 datetime.datetime(2019, 4, 5, 17, 35, 40, tzinfo=tzlocal()), u'ETag':
 '""fe66f4390c50f29798f040d7aae72784""'}, 'ResponseMetadata': {'HTTPStatusCode':
 200, 'RetryAttempts': 0, 'HostId': 'F9ooZOgpE5g9sNgBZxjdiPHqB4+0DNWgj3qbsir
+sKai4fv7rQEcF2fBN1VeeFc2WH45a9ygb2g=', 'RequestId': '8D9CA56A56813DF3', 'HTTPHeaders':
 {'content-length': '234', 'x-amz-id-2': 'F9ooZOgpE5g9sNgBZxjdiPHqB4+0DNWgj3qbsir
+sKai4fv7rQEcF2fBN1VeeFc2WH45a9ygb2g=', 'x-amz-copy-source-version-id':
 'm.MDD0g_QsUnYZ8TBzVFrp.TmjN8PJyX', 'server': 'AmazonS3', 'x-amz-request-id':
 '8D9CA56A56813DF3', 'date': 'Fri, 05 Apr 2019 17:35:40 GMT', 'content-type':
 'application/xml'}}}"
```

Steuern von Zugriffs- und Labeling-Aufträgen mithilfe von Markierungen

Sie können den Zugriff auf Ihre S3-Batch-Vorgänge-Aufträge kennzeichnen und steuern, indem Sie Markierungen hinzufügen. Markierungen können verwendet werden, um zu ermitteln, wer für einen Batchoperations-Auftrag verantwortlich ist. Das Vorhandensein von Aufgabe-Markierungen kann einem Benutzer die Möglichkeit gewähren oder einschränken, eine Aufgabe abzubrechen, eine Aufgabe im Bestätigungsstatus zu aktivieren oder die Prioritätsstufe einer Aufgabe zu ändern. Sie können Jobs mit angehängten Tags erstellen, und Sie können Jobs nach deren Erstellung Tags hinzufügen. Jedes Tag ist ein Schlüssel-Wert-Paar, das beim Erstellen des Auftrags oder später aktualisiert werden kann.

> **Warning**
>
> Stellen Sie sicher, dass Ihre Job-Tags keine vertraulichen Informationen oder personenbezogenen Daten enthalten.

Betrachten Sie das folgende Tagging-Beispiel: Angenommen, Ihre Finanzabteilung soll einen Batchoperations-Auftrag erstellen. Sie könnten eine AWS Identity and Access Management (IAM) -Richtlinie schreiben, die es einem Benutzer ermöglicht, sie aufzurufen`CreateJob`, vorausgesetzt, der Job wird mit `Department` dem Tag erstellt, dem der Wert `Finance` zugewiesen wurde. Außerdem können Sie diese Richtlinie an alle Benutzer anfügen, die Mitglieder der Finanzabteilung sind.

Wenn Sie mit diesem Beispiel fortfahren, können Sie eine Richtlinie schreiben, die es einem Benutzer ermöglicht, die Priorität aller Aufträge, die über die gewünschten Markierungen verfügen, zu aktualisieren oder alle Aufträge, die diese Markierungen besitzen, abzubrechen. Weitere Informationen finden Sie unter the section called “Steuern von Berechtigungen”.

Sie können Markierungen beim Erstellen neuen S3-Batchoperations-Aufträgen oder bereits vorhandenen Aufträgen hinzufügen.

Beachten Sie die folgenden Tag-Einschränkungen:

- Sie können einem Auftrag bis zu 50 Markierungen zuordnen, solange diese über eindeutige Tag-Schlüssel verfügen.
- Ein Tag-Schlüssel kann maximal 128 Unicode-Zeichen lang sein, und die Tag-Werte können bis zu 256 Unicode-Zeichen lang sein.
- Bei Schlüsseln und Werten wird die Groß-/Kleinschreibung berücksichtigt.

Weitere Informationen zu Tag-Einschränkungen finden Sie unter Einschränkungen benutzerdefinierter Markierungen im AWS Billing and Cost Management Benutzerhandbuch.

APIOperationen im Zusammenhang mit dem Tagging von Jobs in S3 Batch Operations

Amazon S3 unterstützt die folgenden API Operationen, die spezifisch für das Tagging von S3 Batch Operations sind:

- GetJobTagging— Gibt den Tagsatz zurück, der einem Batch Operations-Job zugeordnet ist.
- PutJobTagging— Ersetzt den Tagsatz, der einem Job zugeordnet ist. Es gibt zwei unterschiedliche Szenarien für die Verwaltung von Job-Tags in S3 Batch Operations, die diese API Aktion verwenden:
  - Job hat keine Tags — Sie können einem Job eine Reihe von Tags hinzufügen (der Job hat keine vorherigen Tags).
  - Job hat einen Satz vorhandener Tags — Um den vorhandenen Tag-Satz zu ändern, können Sie entweder den vorhandenen Tag-Satz vollständig ersetzen oder Änderungen innerhalb des vorhandenen Tag-Sets vornehmen, indem Sie den vorhandenen Tag-Satz abrufen mit GetJobTagging, ändern Sie das Tag-Set und verwenden Sie diese API Aktion, um das Tag-Set durch das zu ersetzen, das Sie geändert haben.

> **Note**
>
> Wenn Sie diese Anforderung mit einem leeren Tag-Satz senden, löscht S3-Batch-Vorgänge den vorhandenen Tag-Satz für das Objekt. Wenn Sie diese Methode verwenden, wird Ihnen eine Tier-1-Anfrage (PUT) in Rechnung gestellt. Weitere Informationen finden Sie unter Amazon S3 – Preise.
> Wenn Sie vorhandene Markierungen für Ihren Batch-Vorgangsauftrag löschen möchten, wird die Aktion `DeleteJobTagging` bevorzugt, da sie dasselbe Ergebnis ohne anfallende Gebühren erzielt.

- DeleteJobTagging— Löscht den Tagsatz, der einem Batch Operations-Job zugeordnet ist.

Erstellen eines Batch-Vorgangsauftrags mit Auftrags-Markierungen zur Kennzeichnung

Sie können den Zugriff auf Ihre Amazon S3 Batch Operations-Jobs kennzeichnen und kontrollieren, indem Sie Tags hinzufügen. Markierungen können verwendet werden, um zu ermitteln, wer für einen Batch-Vorgangsauftrag verantwortlich ist. Sie können Aufträge mit ihnen zugeordneten Markierungen erstellen oder Sie können Aufträgen nach ihrer Erstellung Markierungen hinzufügen. Weitere Informationen finden Sie unter the section called “Verwenden von Markierungen”.

Mit dem AWS CLI

Im folgenden AWS CLI Beispiel wird ein `S3PutObjectCopy` S3-Auftrag für Batch Operations erstellt, der Job-Tags als Beschriftungen für den Job verwendet.

1. Wählen Sie die Aktion oder `OPERATION` aus, die der Batch-Vorgangsauftrag ausführen soll, und wählen Sie Ihre `TargetResource`.

```
read -d '' OPERATION <<EOF
{
  "S3PutObjectCopy": {
    "TargetResource": "arn:aws:s3:::amzn-s3-demo-destination-bucket"
  }
}
EOF
```

2. Identifizieren Sie die Auftrags-`TAGS`, die Sie für den Auftrag wünschen. In diesem Fall wenden Sie beiden Markierungen `department` und `FiscalYear` mit den Werten `Marketing` bzw. `2020` an.

```
read -d '' TAGS <<EOF
[
  {
    "Key": "department",
    "Value": "Marketing"
  },
  {
    "Key": "FiscalYear",
    "Value": "2020"
  }
]
EOF
```

3. Geben Sie das `MANIFEST` für den Batch-Vorgangsauftrag an.

```
read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "EXAMPLE_S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::amzn-s3-demo-manifest-bucket/example_manifest.csv",
    "ETag": "example-5dc7a8bfb90808fc5d546218"
  }
}
EOF
```

4. Konfigurieren Sie den `REPORT` für den Batch-Vorgangsauftrag.

```
read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::amzn-s3-demo-completion-report-bucket",
  "Format": "Example_Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/copy-with-replace-metadata",
  "ReportScope": "AllTasks"
}
EOF
```

5. Führen Sie die Aktion `create-job` aus, um Ihren Batch-Vorgangsauftrag mit Eingaben zu erstellen, die in den vorherigen Schritten festgelegt wurden.

```
aws \
    s3control create-job \
    --account-id 123456789012 \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/}" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn arn:aws:iam::123456789012:role/batch-operations-role \
    --tags "${TAGS//$'\n'/}" \
    --client-request-token "$(uuidgen)" \
    --region us-west-2 \
    --description "Copy with Replace Metadata";
```

Verwendung des AWS SDK for Java

Example

Im folgenden Beispiel wird ein S3-Batchoperations-Auftrag mit Markierungen unter Verwendung des AWS SDK for Java erstellt.

```
public String createJob(final AWSS3ControlClient awss3ControlClient) {
    final String manifestObjectArn = "arn:aws:s3:::amzn-s3-demo-manifest-bucket/
manifests/10_manifest.csv";
    final String manifestObjectVersionId = "example-5dc7a8bfb90808fc5d546218";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
            .withObjectArn(manifestObjectArn)
            .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
            new
 JobManifestSpec().withFormat(JobManifestFormat.S3InventoryReport_CSV_20161130);

    final JobManifest manifestToPublicApi = new JobManifest()
            .withLocation(manifestLocation)
            .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::amzn-s3-demo-completion-report-
bucket";
```

```
    final String jobReportPrefix = "example-job-reports";

    final JobReport jobReport = new JobReport()
            .withEnabled(true)
            .withReportScope(JobReportScope.AllTasks)
            .withBucket(jobReportBucketArn)
            .withPrefix(jobReportPrefix)
            .withFormat(JobReportFormat.Report_CSV_20180820);

    final String lambdaFunctionArn = "arn:aws:lambda:us-
west-2:123456789012:function:example-function";

    final JobOperation jobOperation = new JobOperation()
            .withLambdaInvoke(new
 LambdaInvokeOperation().withFunctionArn(lambdaFunctionArn));

    final S3Tag departmentTag = new
 S3Tag().withKey("department").withValue("Marketing");
    final S3Tag fiscalYearTag = new S3Tag().withKey("FiscalYear").withValue("2020");

    final String roleArn = "arn:aws:iam::123456789012:role/example-batch-operations-
role";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
            .withAccountId("123456789012")
            .withDescription("Test lambda job")
            .withManifest(manifestToPublicApi)
            .withOperation(jobOperation)
            .withPriority(priority)
            .withRoleArn(roleArn)
            .withReport(jobReport)
            .withTags(departmentTag, fiscalYearTag)
            .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
}
```

Löschen der Markierungen aus einem S3-Batchoperations-Auftrag

Sie können diese Beispiele verwenden, um die Tags aus einem Amazon S3 Batch Operations-Job zu löschen.

Mit dem AWS CLI

Im folgenden Beispiel werden die Markierungen aus einem Batchoperations-Auftrag unter Verwendung der AWS CLI gelöscht.

```
aws \
    s3control delete-job-tagging \
    --account-id 123456789012 \
    --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
    --region us-east-1
```

Löschen der Auftrags-Markierungen eines Batchoperations-Auftrags

Example

Im folgenden Beispiel werden die Markierungen eines S3-Batchoperations-Auftrags unter Verwendung des AWS SDK for Java-SDK für Java gelöscht.

```
public void deleteJobTagging(final AWSS3ControlClient awss3ControlClient,
                             final String jobId) {
    final DeleteJobTaggingRequest deleteJobTaggingRequest = new
 DeleteJobTaggingRequest()
            .withJobId(jobId);

    final DeleteJobTaggingResult deleteJobTaggingResult =
                awss3ControlClient.deleteJobTagging(deleteJobTaggingRequest);
}
```

Hinzufügen von Job-Tags zu einem vorhandenen Batch Operations-Job

Sie können das verwenden PutJobTaggingAPIVorgang zum Hinzufügen von Job-Tags zu Ihren bestehenden Amazon S3 Batch Operations-Jobs. Weitere Informationen finden Sie in den folgenden Beispielen.

Mit dem AWS CLI

Im Folgenden finden Sie ein Beispiel für das Hinzufügen von Job-Tags zu Ihrem S3 Batch Operations-Job mithilfe von AWS CLI. `s3control put-job-tagging` Um die Beispiele zu verwenden, ersetzen Sie das *`Platzhalter für Benutzereingaben`* mit Ihren eigenen Informationen.

> **Note**
>
> Wenn Sie diese Anfrage mit einem leeren Tagsatz senden, löscht Batch Operations den vorhandenen Tagsatz für das Objekt. Wenn Sie jedoch diesen Ansatz verwenden, wird Ihnen eine Tier-1-Anfrage (PUT) in Rechnung gestellt. Weitere Informationen finden Sie unter Amazon S3 – Preise.
>
> Um vorhandene Tags für Ihren Batch Operations-Job zu löschen, empfehlen wir stattdessen, den `DeleteJobTagging` Vorgang zu verwenden, da damit dasselbe Ergebnis erzielt wird, ohne dass Gebühren anfallen.

1. Identifizieren Sie die Auftrags-TAGS, die Sie für den Auftrag wünschen. In diesem Fall wenden Sie beiden Markierungen *`department`* und *`FiscalYear`* mit den Werten *`Marketing`* bzw. *`2020`* an.

```
read -d '' TAGS <<EOF
[
  {
    "Key": "department",
    "Value": "Marketing"
  },
  {
    "Key": "FiscalYear",
    "Value": "2020"
  }
]
EOF
```

2. Führen Sie den folgenden `put-job-tagging` Befehl mit den erforderlichen Parametern aus:

```
aws \
    s3control put-job-tagging \
    --account-id 123456789012 \
```

```
    --tags "${TAGS//$'\n'/}" \
    --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
    --region us-east-1
```

Verwendung des AWS SDK for Java

Example

Im folgenden Beispiel werden die Tags eines S3 Batch Operations-Jobs mithilfe von eingefügt AWS SDK for Java.

```
public void putJobTagging(final AWSS3ControlClient awss3ControlClient,
                          final String jobId) {
    final S3Tag departmentTag = new
 S3Tag().withKey("department").withValue("Marketing");
    final S3Tag fiscalYearTag = new S3Tag().withKey("FiscalYear").withValue("2020");

    final PutJobTaggingRequest putJobTaggingRequest = new PutJobTaggingRequest()
            .withJobId(jobId)
            .withTags(departmentTag, fiscalYearTag);

    final PutJobTaggingResult putJobTaggingResult =
 awss3ControlClient.putJobTagging(putJobTaggingRequest);
}
```

Abrufen der Markierungen eines S3-Batch-Vorgangsauftrags

Um die Tags eines Amazon S3 Batch Operations-Jobs abzurufen, können Sie den `GetJobTagging` API Vorgang verwenden. Weitere Informationen finden Sie in den folgenden Beispielen.

Mit dem AWS CLI

Im folgenden Beispiel werden die Markierungen eines Batch-Vorgangsauftrags unter Verwendung der CLI abgerufen AWS CLI. Um dieses Beispiel zu verwenden, ersetzen Sie das *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
aws \
    s3control get-job-tagging \
    --account-id 123456789012 \
    --job-id Example-e25a-4ed2-8bee-7f8ed7fc2f1c \
```

```
    --region us-east-1
```

Verwendung des AWS SDK for Java

Example

Im folgenden Beispiel werden die Markierungen eines S3 Batch-Vorgangsauftrags unter Verwendung der CLI abgerufen AWS SDK for Java.

```
public List<S3Tag> getJobTagging(final AWSS3ControlClient awss3ControlClient,
                                 final String jobId) {
    final GetJobTaggingRequest getJobTaggingRequest = new GetJobTaggingRequest()
            .withJobId(jobId);

    final GetJobTaggingResult getJobTaggingResult =
            awss3ControlClient.getJobTagging(getJobTaggingRequest);

    final List<S3Tag> tags = getJobTaggingResult.getTags();

    return tags;
}
```

Steuern von Berechtigungen für Batch-Operationen mithilfe von Job-Tags

Um Ihnen bei der Verwaltung Ihrer Amazon S3 Batch Operations-Jobs zu helfen, können Sie Job-Tags hinzufügen. Mit Auftrags-Markierungen können Sie den Zugriff auf Ihre BatchVorgängeaufträge steuern und erzwingen, dass Markierungen angewendet werden, wenn ein Auftrag erstellt wird.

Sie können bis zu 50 Auftrag-Markierungen auf jeden Batchoperations-Auftrag anwenden. Mithilfe von Tags können Sie detaillierte Richtlinien festlegen, um die Anzahl der Benutzer einzuschränken, die den Job bearbeiten können. Aufgaben-Markierungen können einem Benutzer die Möglichkeit gewähren oder einschränken, eine Aufgabe abzubrechen, eine Aufgabe im Bestätigungsstatus zu aktivieren oder die Prioritätsstufe einer Aufgabe zu ändern. Darüber hinaus können Sie erzwingen, dass Markierungen auf alle neuen Aufträge angewendet werden, und die zulässigen Schlüssel-Wert-Paare für die Markierungen angeben. Sie können all diese Bedingungen mithilfe der Richtliniensprache AWS Identity and Access Management (IAM) ausdrücken. Weitere Informationen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

Das folgende Beispiel zeigt, wie Sie S3-Batchoperations-Auftrags-Markierungen verwenden können, um Benutzern die Berechtigung zu erteilen, nur die Aufträge zu erstellen und zu bearbeiten, die in einer bestimmten Abteilung (z. B. der Abteilung Finanzen oder Compliance) ausgeführt werden. Sie können Aufträge auch basierend auf der Entwicklungsphase zuweisen, auf die sie sich beziehen, z. B QS oder Produktion.

In diesem Beispiel verwenden Sie S3 Batch Operations-Job-Tags in IAM Richtlinien, um Benutzern die Erlaubnis zu erteilen, nur die Jobs zu erstellen und zu bearbeiten, die in ihrer Abteilung ausgeführt werden. Sie weisen Aufträge basierend auf der Entwicklungsphase zu, mit der sie verbunden sind, wie QS oder Produktion.

In den folgenden Beispielen werden die folgenden Abteilungen verwendet, wobei jede Abteilung Batch-Operationen auf unterschiedliche Weise verwendet:

- Finanzen
- Compliance
- Business Intelligence
- Entwicklung

Themen

- Steuern des Zugriffs durch Zuweisen von Markierungen zu Benutzern und Ressourcen
- Markieren von BatchVorgängeaufträgen nach Stufe und Durchsetzen von Limits für die Auftragspriorität

Steuern des Zugriffs durch Zuweisen von Markierungen zu Benutzern und Ressourcen

In diesem Szenario verwenden die Administratoren die attributebasierte Zugriffskontrolle (). ABAC ABACist eine IAM Autorisierungsstrategie, bei der Berechtigungen definiert werden, indem sowohl Benutzern als auch Ressourcen Tags zugewiesen werden. AWS

Benutzern und Jobs wird eines der folgenden Abteilungstags zugewiesen:

Schlüssel : Wert

- `department : Finance`
- `department : Compliance`
- `department : BusinessIntelligence`

- `department : Engineering`

> **Note**
>
> Bei Tag-Schlüsseln und -Werten muss die Groß-/Kleinschreibung beachtet werden.

Mithilfe der ABAC Zugriffskontrollstrategie gewähren Sie einem Benutzer in der Finanzabteilung die Berechtigung, S3 Batch Operations-Jobs in seiner Abteilung zu erstellen und zu verwalten, indem Sie das Tag `department=Finance` seinem Benutzer zuordnen.

Darüber hinaus können Sie dem IAM Benutzer eine verwaltete Richtlinie zuordnen, die es jedem Benutzer in seinem Unternehmen ermöglicht, S3 Batch Operations-Jobs in seinen jeweiligen Abteilungen zu erstellen oder zu ändern.

Die Richtlinie in diesem Beispiel enthält drei Richtlinienanweisungen:

- Die erste Anweisung in der Richtlinie ermöglicht es dem Benutzer, einen Batchoperations-Auftrag zu erstellen, vorausgesetzt, die Auftragserstellungsanforderung enthält ein Auftrags-Tag, das der jeweiligen Abteilung entspricht. Dies wird mithilfe der Syntax `"${aws:PrincipalTag/department}"` ausgedrückt, die zum Zeitpunkt der Richtlinienauswertung durch das Abteilungs-Tag des Benutzers ersetzt wird. Die Bedingung ist erfüllt, wenn der für das Abteilungs-Tag in der Anforderung (`"aws:RequestTag/department"`) angegebene Wert mit dem der Abteilung des Benutzers übereinstimmt.
- Die zweite Anweisung in der Richtlinie ermöglicht es Benutzern, die Priorität von Aufträgen zu ändern oder den Status eines Auftrags zu aktualisieren, sofern der Auftrag, den der Benutzer aktualisiert, mit der Abteilung des Benutzers übereinstimmt.
- Die dritte Anweisung ermöglicht es einem Benutzer, die Markierungen eines Batchoperations-Auftrags jederzeit über eine `PutJobTagging`-Anforderung zu aktualisieren, solange (1) das Abteilung-Tag erhalten bleibt und (2) sich der Auftrag, der aktualisiert wird, innerhalb der Abteilung des Benutzers befindet.

```
{
      "Version": "2012-10-17",
      "Statement": [
            {
                  "Effect": "Allow",
                  "Action": "s3:CreateJob",
```

```
                "Resource": "*",
                "Condition": {
                      "StringEquals": {
                            "aws:RequestTag/department": "${aws:PrincipalTag/
department}"
                }
            }
        },
            {
                "Effect": "Allow",
                "Action": [
                      "s3:UpdateJobPriority",
                      "s3:UpdateJobStatus"
            ],
                "Resource": "*",
                "Condition": {
                      "StringEquals": {
                            "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
                }
            }
        },
            {
                "Effect": "Allow",
                "Action": "s3:PutJobTagging",
                "Resource": "*",
                "Condition": {
                      "StringEquals": {
                            "aws:RequestTag/department": "${aws:PrincipalTag/
department}",
                            "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
                }
            }
        }
    ]
}
```

Markieren von BatchVorgängeaufträgen nach Stufe und Durchsetzen von Limits für die Auftragspriorität

Alle S3-BatchVorgängeaufträge haben eine numerische Priorität, anhand derer Amazon S3 entscheidet, in welcher Reihenfolge die Aufträge ausgeführt werden sollen. In diesem Beispiel

beschränken Sie die maximale Priorität, die die meisten Benutzer Aufträgen zuweisen können, wobei höhere Prioritätsbereiche für eine begrenzte Gruppe von berechtigten Benutzern reserviert sind:

- Prioritätsbereich der QA-Stufe (niedrig): 1-100
- Prioritätsbereich der Produktionsstufe (hoch): 1-300

Dazu führen Sie einen neuen Tag-Satz ein, der die Phase des Auftrags repräsentiert:

Schlüssel : Wert

- `stage : QA`
- `stage : Production`

Erstellen und Aktualisieren von Aufträgen mit geringer Priorität in einer Abteilung

Mit dieser Richtlinie werden neben der abteilungsbasierten Einschränkung zwei neue Einschränkungen für die Erstellung und Aktualisierung von S3-BatchVorgängeaufträgen eingeführt:

- Es ermöglicht Benutzern, Aufträge in ihrer Abteilung mit einer neuen Bedingung zu erstellen oder zu aktualisieren, die erfordert, dass der Auftrag das Tag enthäl `stage=QA`.
- Es ermöglicht Benutzern, Aufträge mit einer maximalen Priorität von bis 100 zu erstellen oder zu aktualisieren.

```
{
        "Version": "2012-10-17",
        "Statement": [
        {
        "Effect": "Allow",
        "Action": "s3:CreateJob",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:RequestTag/department": "${aws:PrincipalTag/department}",
                "aws:RequestTag/stage": "QA"
            },
            "NumericLessThanEquals": {
                "s3:RequestJobPriority": 100
            }
        }
```

```
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:UpdateJobStatus"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/department": "${aws:PrincipalTag/department}"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": "s3:UpdateJobPriority",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:ResourceTag/department": "${aws:PrincipalTag/department}",
                "aws:ResourceTag/stage": "QA"
            },
            "NumericLessThanEquals": {
                "s3:RequestJobPriority": 100
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": "s3:PutJobTagging",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:RequestTag/department" : "${aws:PrincipalTag/department}",
                "aws:ResourceTag/department": "${aws:PrincipalTag/department}",
                "aws:RequestTag/stage": "QA",
                "aws:ResourceTag/stage": "QA"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": "s3:GetJobTagging",
        "Resource": "*"
```

```
    }
    ]
}
```

Erstellen und Aktualisieren von Aufträgen mit hoher Priorität in einer Abteilung

Einer kleinen Anzahl von Benutzern muss u. U. möglich sein, Aufträge hoher Priorität in QS oder Produktion zu erstellen. Um diesen Bedarf zu unterstützen, erstellen Sie eine verwaltete Richtlinie durch Abwandlung einer Richtlinie mit niedriger Priorität im vorherigen Abschnitt.

Diese Richtlinie gewährt die folgenden Aktionen:

- Ermöglicht Benutzern, Aufträge in ihrer Abteilung entweder mit dem Tag `stage=QA` oder `stage=Production` zu erstellen oder zu aktualisieren.
- Ermöglicht Benutzern, bei der Erstellung oder Aktualisierung eines Auftrags als Priorität bis zu 300 anzugeben.

```
{
      "Version": "2012-10-17",
      "Statement": [
          {
                "Effect": "Allow",
                "Action": "s3:CreateJob",
                "Resource": "*",
                "Condition": {
                      "ForAnyValue:StringEquals": {
                            "aws:RequestTag/stage": [
                                  "QA",
                                  "Production"
                    ]
                },
                      "StringEquals": {
                            "aws:RequestTag/department": "${aws:PrincipalTag/
department}"
                },
                      "NumericLessThanEquals": {
                            "s3:RequestJobPriority": 300
                }
            }
        },
          {
```

```
                "Effect": "Allow",
                "Action": [
                      "s3:UpdateJobStatus"
            ],
                "Resource": "*",
                "Condition": {
                      "StringEquals": {
                            "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
                }
            }
        },
          {
                "Effect": "Allow",
                "Action": "s3:UpdateJobPriority",
                "Resource": "*",
                "Condition": {
                      "ForAnyValue:StringEquals": {
                            "aws:ResourceTag/stage": [
                                  "QA",
                                  "Production"
                    ]
                },
                      "StringEquals": {
                            "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
                },
                      "NumericLessThanEquals": {
                            "s3:RequestJobPriority": 300
                }
            }
        },
          {
                "Effect": "Allow",
                "Action": "s3:PutJobTagging",
                "Resource": "*",
                "Condition": {
                      "StringEquals": {
                            "aws:RequestTag/department": "${aws:PrincipalTag/
department}",
                            "aws:ResourceTag/department": "${aws:PrincipalTag/
department}"
                },
                      "ForAnyValue:StringEquals": {
```

```
                        "aws:RequestTag/stage": [
                                "QA",
                                "Production"
                    ],
                        "aws:ResourceTag/stage": [
                                "QA",
                                "Production"
                    ]
                }
            }
        }
    ]
}
```

Verwalten der S3-Objektsperre mit S3-BatchVorgänge

Sie können S3-Batch-Vorgänge verwenden, um umfangreiche Batch-Vorgänge für Amazon S3-Objekte durchzuführen. S3-Batch-Vorgänge kann eine einzelne Operation für Listen von Amazon S3-Objekten durchführen, die Sie angeben. Ein einziger Auftrag kann eine festgelegte Operation auf Milliarden von Objekten mit mehreren Exabytes an Daten durchführen. Amazon S3 verfolgt den Fortschritt, versendet Benachrichtigungen und speichert einen detaillierten Abschlussbericht zu allen Aktionen. So profitieren Sie von einer vollständig verwalteten, prüfbaren und serverlosen Umgebung. Sie können S3 Batch Operations über die Amazon S3-Konsole AWS CLI, AWS SDKs, oder Amazon S3 verwenden RESTAPI.

Mit der S3-Objektsperre können Sie eine rechtliche Aufbewahrungsfrist für eine Objektversion festlegen. Wie das Festlegen von Aufbewahrungszeiträumen verhindern auch rechtliche Aufbewahrungsfristen das Überschreiben oder Löschen von Objektversionen. Einer gesetzlichen Sperre ist jedoch keine Aufbewahrungsfrist zugeordnet und sie bleibt in Kraft, bis die gesetzliche Sperre aufgehoben wird. Weitere Informationen finden Sie unter S3-Objektsperre aufgrund gesetzlicher Aufbewahrungsfristen.

Informationen zur Verwendung von S3 Batch Operations with Object Lock, um vielen Amazon S3 S3-Objekten gleichzeitig gesetzliche Aufbewahrungsfristen hinzuzufügen, finden Sie in den folgenden Themen.

Themen

- Aktivieren der S3-Objektsperre mit S3-BatchVorgänge
- Festlegen der Objektsperrenaufbewahrung mit BatchVorgänge

- Verwendung von S3-Batch-Vorgänge mit dem Compliance-Modus der S3-Objektsperrenaufbewahrung
- Verwenden von S3-Batch-Vorgänge mit dem Governance-Modus der S3-Objektsperrenaufbewahrung
- Verwendung von S3 Batch Operations zur Deaktivierung von S3 Object Lock Legal Holds

Aktivieren der S3-Objektsperre mit S3-BatchVorgänge

Sie können Amazon S3 Batch Operations mit S3 Object Lock verwenden, um die Aufbewahrung zu verwalten oder eine gesetzliche Aufbewahrungsfrist für viele Amazon S3 S3-Objekte gleichzeitig zu aktivieren. Sie geben die Liste der Zielobjekte in Ihrem Manifest an und senden sie zur Fertigstellung an BatchVorgänge. Weitere Informationen erhalten Sie unter the section called “Aufrechterhaltung der Objektsperre” und the section called “Objektsperre aufgrund gesetzlicher Aufbewahrungsfristen”.

Die folgenden Beispiele zeigen, wie Sie eine Rolle AWS Identity and Access Management (IAM) mit S3 Batch Operations-Berechtigungen erstellen und die Rollenberechtigungen aktualisieren, um Jobs zu erstellen, die Object Lock aktivieren. Sie benötigen außerdem ein CSV Manifest, das die Objekte für Ihren S3 Batch Operations-Job identifiziert. Weitere Informationen finden Sie unter the section called “Angeben eines Manifests”.

Um die folgenden Beispiele zu verwenden, ersetzen Sie das *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

Mit dem AWS CLI

1. Erstellen Sie eine IAM Rolle und weisen Sie S3 Batch Operationsberechtigungen zur Ausführung zu.

   Dieser Schritt ist für alle S3-BatchVorgängeaufträge erforderlich.

```
export AWS_PROFILE='aws-user'

read -d '' batch_operations_trust_policy <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
```

```
          "batchoperations.s3.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
aws iam create-role --role-name batch_operations-objectlock \
--assume-role-policy-document "${batch_operations_trust_policy}"
```

2. Richten Sie S3-Batch-Vorgänge zur Ausführung mit S3-Objektsperre ein.

   In diesem Schritt lassen Sie der Rolle Folgendes zu:

   a. Führen Sie die Objektsperre in dem S3-Bucket aus, der die Zielobjekte enthält, für die Batch-Vorgänge ausgeführt werden soll.

   b. Lesen Sie den S3-Bucket, in dem sich die CSV Manifestdatei und die Objekte befinden.

   c. Schreiben Sie die Ergebnisse des S3-Batch-Vorgangsauftrags in den Berichts-Bucket.

```
read -d '' batch_operations_permissions <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetBucketObjectLockConfiguration",
            "Resource": [
                "arn:aws:s3:::{{amzn-s3-demo-manifest-bucket}}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::{{amzn-s3-demo-manifest-bucket}}/*"
            ]
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::{{amzn-s3-demo-completion-report-bucket}}/*"
            ]
        }
    ]
}
EOF

aws iam put-role-policy --role-name batch_operations-objectlock \
--policy-name object-lock-permissions \
--policy-document "${batch_operations_permissions}"
```

Verwendung des AWS SDK for Java

Die folgenden Beispiele zeigen, wie Sie eine IAM Rolle mit S3 Batch Operations-Berechtigungen erstellen und die Rollenberechtigungen aktualisieren, um Jobs zu erstellen, die Object Lock aktivieren, mithilfe von AWS SDK for Java. Sie müssen auch über ein CSV-Manifest verfügen, das die Objekte für Ihren S3-Batch-Vorgangsauftrag identifiziert. Weitere Informationen finden Sie unter the section called “Angeben eines Manifests”.

Führen Sie die folgenden Schritte aus:

1. Erstellen Sie eine IAM Rolle und weisen Sie S3 Batch Operationsberechtigungen zur Ausführung zu. Dieser Schritt ist für alle S3-BatchVorgängeaufträge erforderlich.
2. Richten Sie S3-Batch-Vorgänge zur Ausführung mit S3-Objektsperre ein.

   Sie lassen der Rolle Folgendes zu:

   1. Führen Sie die Objektsperre in dem S3-Bucket aus, der die Zielobjekte enthält, für die Batch-Vorgänge ausgeführt werden soll.
   2. Lesen Sie den S3-Bucket, in dem sich die CSV Manifestdatei und die Objekte befinden.
   3. Schreiben Sie die Ergebnisse des S3-Batch-Vorgangsauftrags in den Berichts-Bucket.

```
public void createObjectLockRole() {
    final String roleName = "batch_operations-object-lock";

    final String trustPolicy = "{" +
            "  \"Version\": \"2012-10-17\", " +
            "  \"Statement\": [ " +
            "    { " +
            "      \"Effect\": \"Allow\", " +
            "      \"Principal\": { " +
            "        \"Service\": [" +
            "          \"batchoperations.s3.amazonaws.com\"" +
            "        ]" +
            "      }, " +
            "      \"Action\": \"sts:AssumeRole\" " +
            "    } " +
            "  ]" +
            "}";

    final String bopsPermissions = "{" +
            "    \"Version\": \"2012-10-17\"," +
            "    \"Statement\": [" +
            "        {" +
            "            \"Effect\": \"Allow\"," +
            "            \"Action\": \"s3:GetBucketObjectLockConfiguration\"," +
            "            \"Resource\": [" +
            "                \"arn:aws:s3:::amzn-s3-demo-manifest-bucket\"" +
            "            ]" +
            "        }," +
            "        {" +
            "            \"Effect\": \"Allow\"," +
            "            \"Action\": [" +
            "                \"s3:GetObject\"," +
            "                \"s3:GetObjectVersion\"," +
            "                \"s3:GetBucketLocation\"" +
            "            ]," +
            "            \"Resource\": [" +
            "                \"arn:aws:s3:::amzn-s3-demo-manifest-bucket/*\"" +
            "            ]" +
            "        }," +
            "        {" +
            "            \"Effect\": \"Allow\"," +
            "            \"Action\": [" +
            "                \"s3:PutObject\"," +
```

```
            "                \"s3:GetBucketLocation\"" +
            "            ]," +
            "            \"Resource\": [" +
            "                \"arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*\""
 +
            "            ]" +
            "        }" +
            "    ]" +
            "}";

    final AmazonIdentityManagement iam =
            AmazonIdentityManagementClientBuilder.defaultClient();

    final CreateRoleRequest createRoleRequest = new CreateRoleRequest()
            .withAssumeRolePolicyDocument(bopsPermissions)
            .withRoleName(roleName);

    final CreateRoleResult createRoleResult = iam.createRole(createRoleRequest);

    final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
            .withPolicyDocument(bopsPermissions)
            .withPolicyName("batch_operations-permissions")
            .withRoleName(roleName);

    final PutRolePolicyResult putRolePolicyResult =
 iam.putRolePolicy(putRolePolicyRequest);
}
```

Festlegen der Objektsperrenaufbewahrung mit BatchVorgänge

Sie können Amazon S3 Batch Operations mit S3 Object Lock verwenden, um die Aufbewahrung für viele Amazon S3 S3-Objekte gleichzeitig zu verwalten. Sie geben die Liste der Zielobjekte in Ihrem Manifest an und senden sie zur Fertigstellung an BatchVorgänge. Weitere Informationen erhalten Sie unter the section called “Aufrechterhaltung der Objektsperre” und the section called “Objektsperre aufgrund gesetzlicher Aufbewahrungsfristen”.

Die folgenden Beispiele zeigen, wie Sie eine Rolle AWS Identity and Access Management (IAM) mit S3 Batch Operations-Berechtigungen erstellen und die Rollenberechtigungen so aktualisieren, dass sie die `s3:PutObjectRetention` Berechtigungen enthalten, sodass Sie die S3 Object Lock-Aufbewahrung für die Objekte in Ihrem Manifest-Bucket ausführen können. Sie benötigen außerdem ein `CSV` Manifest, das die Objekte für Ihren S3 Batch Operations-Job identifiziert. Weitere Informationen finden Sie unter the section called “Angeben eines Manifests”.

Um die folgenden Beispiele zu verwenden, ersetzen Sie das *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

Mit dem AWS CLI

Das folgende AWS CLI Beispiel zeigt, wie Batch Operations verwendet werden kann, um die Aufbewahrung von S3 Object Lock auf mehrere Objekte anzuwenden.

```
export AWS_PROFILE='aws-user'

read -d '' retention_permissions <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObjectRetention"
            ],
            "Resource": [
                "arn:aws:s3:::{{amzn-s3-demo-manifest-bucket}}/*"
            ]
        }
    ]
}
EOF

aws iam put-role-policy --role-name batch_operations-objectlock --policy-
name retention-permissions --policy-document "${retention_permissions}"
```

Verwendung des AWS SDK for Java

Das folgende AWS SDK for Java Beispiel zeigt, wie Batch Operations verwendet werden kann, um die Aufbewahrung von S3 Object Lock auf mehrere Objekte anzuwenden.

```
public void allowPutObjectRetention() {
    final String roleName = "batch_operations-object-lock";

    final String retentionPermissions = "{" +
            "    \"Version\": \"2012-10-17\"," +
            "    \"Statement\": [" +
            "        {" +
```

```
            "            \"Effect\": \"Allow\"," +
            "            \"Action\": [" +
            "                \"s3:PutObjectRetention\"" +
            "            ]," +
            "            \"Resource\": [" +
            "                \"arn:aws:s3:::amzn-s3-demo-manifest-bucket*\"" +
            "            ]" +
            "        }" +
            "    ]" +
            "}";

    final AmazonIdentityManagement iam =
            AmazonIdentityManagementClientBuilder.defaultClient();

    final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
            .withPolicyDocument(retentionPermissions)
            .withPolicyName("retention-permissions")
            .withRoleName(roleName);

    final PutRolePolicyResult putRolePolicyResult =
 iam.putRolePolicy(putRolePolicyRequest);
}
```

Verwendung von S3-Batch-Vorgänge mit dem Compliance-Modus der S3-Objektsperrenaufbewahrung

Das folgende Beispiel baut auf den vorherigen Beispielen zum Erstellen einer Vertrauensrichtlinie sowie zum Festlegen von S3-Batch-Vorgangs und S3-Objektsperren-Konfigurations-Berechtigungen für Ihre Objekte auf. In diesem Beispiel wird der Aufbewahrungsmodus auf `COMPLIANCE` und dann `retain until date` auf den 1. Januar 2025 festgelegt. In diesem Beispiel wird ein Job erstellt, der auf Objekte im Manifest-Bucket abzielt und die Ergebnisse in dem von Ihnen identifizierten Berichts-Bucket meldet.

Um die folgenden Beispiele zu verwenden, ersetzen Sie den *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

Mit dem AWS CLI

Die folgenden AWS CLI Beispiele zeigen, wie Batch Operations verwendet werden kann, um den Aufbewahrungskonformitätsmodus von S3 Object Lock auf mehrere Objekte anzuwenden.

Example — Legen Sie den Compliance-Compliance-Modus für die Aufbewahrung von S3 Object Lock auf mehrere Objekte fest

```
export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/batch_operations-objectlock'

read -d '' OPERATION <<EOF
{
  "S3PutObjectRetention": {
    "Retention": {
      "RetainUntilDate":"2025-01-01T00:00:00",
      "Mode":"COMPLIANCE"
    }
  }
}
EOF

read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::amzn-s3-demo-manifest-bucket/compliance-objects-
manifest.csv",
    "ETag": "Your-manifest-ETag"
  }
}
EOF

read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::ReportBucket",
  "Format": "Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "amzn-s3-demo-completion-report-bucket/compliance-objects-batch-
operations",
```

```
  "ReportScope": "AllTasks"
}
EOF

aws \
    s3control create-job \
    --account-id "${ACCOUNT_ID}" \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/}" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn "${ROLE_ARN}" \
    --client-request-token "$(uuidgen)" \
    --region "${AWS_DEFAULT_REGION}" \
    --description "Set compliance retain-until to 1 Jul 2030";
```

Example — Verlängern Sie die **`COMPLIANCE`** Modi **`retain until date`** bis zum 15. Januar 2025

Im folgenden Beispiel wird das `COMPLIANCE` des `retain until date`-Modus bis zum 15. Januar 2025 erweitert.

```
export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/batch_operations-objectlock'

read -d '' OPERATION <<EOF
{
  "S3PutObjectRetention": {
    "Retention": {
      "RetainUntilDate":"2025-01-15T00:00:00",
      "Mode":"COMPLIANCE"
    }
  }
}
EOF

read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
```

```
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::amzn-s3-demo-manifest-bucket/compliance-objects-
manifest.csv",
    "ETag": "Your-manifest-ETag"
  }
}
EOF

read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::amzn-s3-demo-completion-report-bucket",
  "Format": "Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/compliance-objects-batch_operations",
  "ReportScope": "AllTasks"
}
EOF

aws \
    s3control create-job \
    --account-id "${ACCOUNT_ID}" \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/}" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn "${ROLE_ARN}" \
    --client-request-token "$(uuidgen)" \
    --region "${AWS_DEFAULT_REGION}" \
    --description "Extend compliance retention to 15 Jan 2025";
```

Verwendung des AWS SDK for Java

Die folgenden AWS SDK for Java Beispiele zeigen, wie Batch Operations verwendet werden kann, um den Aufbewahrungskonformitätsmodus von S3 Object Lock auf mehrere Objekte anzuwenden.

Example — Stellen Sie den Aufbewahrungsmodus auf COMPLIANCE und das Aufbewahrungsdatum bis zum 1. Januar 2025

```
public String createComplianceRetentionJob(final AWSS3ControlClient awss3ControlClient)
 throws ParseException {
```

```
    final String manifestObjectArn = "arn:aws:s3:::amzn-s3-demo-manifest-bucket/
compliance-objects-manifest.csv";
    final String manifestObjectVersionId = "your-object-version-Id";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
            .withObjectArn(manifestObjectArn)
            .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
            new JobManifestSpec()
                    .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
                    .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
            .withLocation(manifestLocation)
            .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::amzn-s3-demo-completion-report-
bucket";
    final String jobReportPrefix = "reports/compliance-objects-bops";

    final JobReport jobReport = new JobReport()
            .withEnabled(true)
            .withReportScope(JobReportScope.AllTasks)
            .withBucket(jobReportBucketArn)
            .withPrefix(jobReportPrefix)
            .withFormat(JobReportFormat.Report_CSV_20180820);

    final SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
    final Date janFirst = format.parse("01/01/2025");

    final JobOperation jobOperation = new JobOperation()
            .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
                    .withRetention(new S3Retention()
                            .withMode(S3ObjectLockRetentionMode.COMPLIANCE)
                            .withRetainUntilDate(janFirst)));

    final String roleArn = "arn:aws:iam::123456789012:role/batch_operations-object-
lock";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
            .withAccountId("123456789012")
```

```
            .withDescription("Set compliance retain-until to 1 Jan 2025")
            .withManifest(manifestToPublicApi)
            .withOperation(jobOperation)
            .withPriority(priority)
            .withRoleArn(roleArn)
            .withReport(jobReport)
            .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
}
```

Example — Den **`COMPLIANCE`** Modus verlängern **`retain until date`**

Das folgende Beispiel erweitert den `COMPLIANCE` Modus `retain until date` bis zum 15. Januar 2025.

```
public String createExtendComplianceRetentionJob(final AWSS3ControlClient
 awss3ControlClient) throws ParseException {
    final String manifestObjectArn = "arn:aws:s3:::amzn-s3-demo-manifest-bucket/
compliance-objects-manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
            .withObjectArn(manifestObjectArn)
            .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
            new JobManifestSpec()
                    .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
                    .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
            .withLocation(manifestLocation)
            .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::amzn-s3-demo-completion-report-
bucket";
    final String jobReportPrefix = "reports/compliance-objects-batch_operations";

    final JobReport jobReport = new JobReport()
            .withEnabled(true)
```

```
            .withReportScope(JobReportScope.AllTasks)
            .withBucket(jobReportBucketArn)
            .withPrefix(jobReportPrefix)
            .withFormat(JobReportFormat.Report_CSV_20180820);

    final SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
    final Date jan15th = format.parse("15/01/2025");

    final JobOperation jobOperation = new JobOperation()
            .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
                    .withRetention(new S3Retention()
                            .withMode(S3ObjectLockRetentionMode.COMPLIANCE)
                            .withRetainUntilDate(jan15th)));

    final String roleArn = "arn:aws:iam::123456789012:role/batch_operations-object-
lock";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
            .withAccountId("123456789012")
            .withDescription("Extend compliance retention to 15 Jan 2025")
            .withManifest(manifestToPublicApi)
            .withOperation(jobOperation)
            .withPriority(priority)
            .withRoleArn(roleArn)
            .withReport(jobReport)
            .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
}
```

Verwenden von S3-Batch-Vorgänge mit dem Governance-Modus der S3-Objektsperrenaufbewahrung

Das folgende Beispiel baut auf dem vorherigen Beispiel zum Erstellen einer Vertrauensrichtlinie sowie zum Festlegen von S3-Batch-Vorgangs- und S3-Objektsperren-Konfigurations-Berechtigungen auf. Dieses Beispiel zeigt, wie die Aufbewahrungs-Governance `retain until date` von S3 Object Lock mit dem Stand vom 30. Januar 2025 auf mehrere Objekte angewendet wird. Es erstellt einen

Batch-Vorgangsauftrag, der den Manifest-Bucket verwendet und die Ergebnisse im Berichts-Bucket meldet.

Um die folgenden Beispiele zu verwenden, ersetzen Sie das *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

Mit dem AWS CLI

Die folgenden AWS CLI Beispiele zeigen, wie Batch Operations verwendet werden kann, um den Aufbewahrungs-Governance-Modus von S3 Object Lock auf mehrere Objekte anzuwenden.

Example — Wenden Sie die Aufbewahrungs-Governance von S3 Object Lock auf mehrere Objekte an, wobei die Aufbewahrungsfrist bis zum 30. Januar 2025 gilt

```
export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/batch_operations-objectlock'

read -d '' OPERATION <<EOF
{
  "S3PutObjectRetention": {
    "Retention": {
      "RetainUntilDate":"2025-01-30T00:00:00",
      "Mode":"GOVERNANCE"
    }
  }
}
EOF

read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::amzn-s3-demo-manifest-bucket/governance-objects-
manifest.csv",
    "ETag": "Your-manifest-ETag"
```

```
  }
}
EOF

read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::amzn-s3-demo-completion-report-bucketT",
  "Format": "Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/governance-objects",
  "ReportScope": "AllTasks"
}
EOF

aws \
    s3control create-job \
    --account-id "${ACCOUNT_ID}" \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/}" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn "${ROLE_ARN}" \
    --client-request-token "$(uuidgen)" \
    --region "${AWS_DEFAULT_REGION}" \
    --description "Put governance retention";
```

Example — Umgehen Sie die Aufbewahrungs-Governance für mehrere Objekte

Das folgende Beispiel baut auf dem vorherigen Beispiel zum Erstellen einer Vertrauensrichtlinie sowie zum Festlegen von S3-Batch-Vorgangs- und S3-Objektsperren-Konfigurations-Berechtigungen auf. Es zeigt, wie Sie die Aufbewahrungs-Governance über mehrere Objekte hinweg umgehen und erstellt einen Batch-Vorgangsauftrag, der den Manifest-Bucket verwendet und die Ergebnisse im Berichts-Bucket meldet.

```
export AWS_PROFILE='aws-user'

read -d '' bypass_governance_permissions <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
```

```
                "s3:BypassGovernanceRetention"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
            ]
        }
    ]
}
EOF

aws iam put-role-policy --role-name batch-operations-objectlock --policy-name bypass-
governance-permissions --policy-document "${bypass_governance_permissions}"

export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/batch_operations-objectlock'

read -d '' OPERATION <<EOF
{
  "S3PutObjectRetention": {
    "BypassGovernanceRetention": true,
    "Retention": {
    }
  }
}
EOF

read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
    "ObjectArn": "arn:aws:s3:::amzn-s3-demo-manifest-bucket/governance-objects-
manifest.csv",
    "ETag": "Your-manifest-ETag"
  }
}
EOF
```

```
read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::amzn-s3-demo-completion-report-bucket",
  "Format": "Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/batch_operations-governance",
  "ReportScope": "AllTasks"
}
EOF

aws \
    s3control create-job \
    --account-id "${ACCOUNT_ID}" \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/}" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn "${ROLE_ARN}" \
    --client-request-token "$(uuidgen)" \
    --region "${AWS_DEFAULT_REGION}" \
    --description "Remove governance retention";
```

Verwendung des AWS SDK for Java

Das folgende Beispiel baut auf dem vorherigen Beispiel zum Erstellen einer Vertrauensrichtlinie sowie zum Festlegen von S3-Batch-Vorgangs- und S3-Objektsperren-Konfigurations-Berechtigungen auf. Dieses Beispiel zeigt, wie die Aufbewahrungs-Governance von S3 Object Lock mit der `retain until date` Einstellung auf den 30. Januar 2025 auf mehrere Objekte angewendet wird. In diesem Beispiel wird ein Batch Operations-Job erstellt, der den Manifest-Bucket verwendet und die Ergebnisse im Berichts-Bucket meldet.

Example — Wenden Sie die Aufbewahrungs-Governance von S3 Object Lock auf mehrere Objekte an, wobei die Aufbewahrungsfrist bis zum 30. Januar 2025 gilt

```
public String createGovernanceRetentionJob(final AWSS3ControlClient awss3ControlClient)
 throws ParseException {
    final String manifestObjectArn = "arn:aws:s3:::amzn-s3-demo-manifest-bucket/
governance-objects-manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
```

```
            .withObjectArn(manifestObjectArn)
            .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
            new JobManifestSpec()
                    .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
                    .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
            .withLocation(manifestLocation)
            .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::amzn-s3-demo-completion-report-
bucket";
    final String jobReportPrefix = "reports/governance-objects";

    final JobReport jobReport = new JobReport()
            .withEnabled(true)
            .withReportScope(JobReportScope.AllTasks)
            .withBucket(jobReportBucketArn)
            .withPrefix(jobReportPrefix)
            .withFormat(JobReportFormat.Report_CSV_20180820);

    final SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
    final Date jan30th = format.parse("30/01/2025");

    final JobOperation jobOperation = new JobOperation()
            .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
                    .withRetention(new S3Retention()
                            .withMode(S3ObjectLockRetentionMode.GOVERNANCE)
                            .withRetainUntilDate(jan30th)));

    final String roleArn = "arn:aws:iam::123456789012:role/batch_operations-object-
lock";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
            .withAccountId("123456789012")
            .withDescription("Put governance retention")
            .withManifest(manifestToPublicApi)
            .withOperation(jobOperation)
            .withPriority(priority)
            .withRoleArn(roleArn)
```

```
            .withReport(jobReport)
            .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
}
```

Example — Umgehen Sie die Aufbewahrungs-Governance für mehrere Objekte

Das folgende Beispiel baut auf dem vorherigen Beispiel zum Erstellen einer Vertrauensrichtlinie sowie zum Festlegen von S3-Batch-Vorgangs- und S3-Objektsperren-Konfigurations-Berechtigungen auf. Dieses Beispiel zeigt, wie Sie die Aufbewahrungs-Governance für mehrere Objekte umgehen und einen Batch Operations-Job erstellen, der den Manifest-Bucket verwendet und die Ergebnisse im Berichts-Bucket meldet.

```
public void allowBypassGovernance() {
    final String roleName = "batch_operations-object-lock";

    final String bypassGovernancePermissions = "{" +
            "    \"Version\": \"2012-10-17\"," +
            "    \"Statement\": [" +
            "        {" +
            "            \"Effect\": \"Allow\"," +
            "            \"Action\": [" +
            "                \"s3:BypassGovernanceRetention\"" +
            "            ]," +
            "            \"Resource\": [" +
            "                \"arn:aws:s3:::amzn-s3-demo-manifest-bucket/*\"" +
            "            ]" +
            "        }" +
            "    ]" +
            "}";

    final AmazonIdentityManagement iam =
            AmazonIdentityManagementClientBuilder.defaultClient();

    final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
            .withPolicyDocument(bypassGovernancePermissions)
            .withPolicyName("bypass-governance-permissions")
            .withRoleName(roleName);
```

```
    final PutRolePolicyResult putRolePolicyResult =
 iam.putRolePolicy(putRolePolicyRequest);
}
public String createRemoveGovernanceRetentionJob(final AWSS3ControlClient
 awss3ControlClient) {
    final String manifestObjectArn = "arn:aws:s3:::amzn-s3-demo-manifest-bucket/
governance-objects-manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
            .withObjectArn(manifestObjectArn)
            .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
            new JobManifestSpec()
                    .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
                    .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
            .withLocation(manifestLocation)
            .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::amzn-s3-demo-completion-report-
bucket";
    final String jobReportPrefix = "reports/batch_operations-governance";

    final JobReport jobReport = new JobReport()
            .withEnabled(true)
            .withReportScope(JobReportScope.AllTasks)
            .withBucket(jobReportBucketArn)
            .withPrefix(jobReportPrefix)
            .withFormat(JobReportFormat.Report_CSV_20180820);

    final JobOperation jobOperation = new JobOperation()
            .withS3PutObjectRetention(new S3SetObjectRetentionOperation()
                    .withRetention(new S3Retention()));

    final String roleArn = "arn:aws:iam::123456789012:role/batch_operations-object-
lock";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
            .withAccountId("123456789012")
```

```
            .withDescription("Remove governance retention")
            .withManifest(manifestToPublicApi)
            .withOperation(jobOperation)
            .withPriority(priority)
            .withRoleArn(roleArn)
            .withReport(jobReport)
            .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
}
```

Verwendung von S3 Batch Operations zur Deaktivierung von S3 Object Lock Legal Holds

Das folgende Beispiel baut auf den vorherigen Beispielen zum Erstellen einer Vertrauensrichtlinie sowie zum Festlegen von S3-Batch-Vorgangs- und S3-Objektsperren-Konfigurations-Berechtigungen auf. Dieses Beispiel zeigt, wie Object Lock Legal Hold für Objekte mithilfe von Batch-Operationen deaktiviert wird.

Im Beispiel wird zuerst die Rolle aktualisiert, um `s3:PutObjectLegalHold`-Berechtigungen zu erteilen, ein Batch-Vorgangsauftrag erstellt, der die rechtliche Aufbewahrungsfrist von den im Manifest identifizierten Objekten deaktiviert (entfernt) und anschließend Berichte darüber erstellt.

Um die folgenden Beispiele zu verwenden, ersetzen Sie den *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

Mit dem AWS CLI

Die folgenden AWS CLI Beispiele zeigen, wie Batch-Operationen verwendet werden, um die gesetzlichen Aufbewahrungsfristen von S3 Object Lock für mehrere Objekte zu deaktivieren.

Example — Aktualisiert die Rolle, um **`s3:PutObjectLegalHold`** Berechtigungen zu gewähren

```
export AWS_PROFILE='aws-user'

read -d '' legal_hold_permissions <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": [
                "s3:PutObjectLegalHold"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
            ]
        }
    ]

EOF

aws iam put-role-policy --role-name batch_operations-objectlock --policy-name legal-
hold-permissions --policy-document "${legal_hold_permissions}"
```

Example — Schaltet die gesetzliche Sperre aus

Im folgenden Beispiel wird die rechtliche Aufbewahrungsfrist deaktiviert.

```
export AWS_PROFILE='aws-user'
export AWS_DEFAULT_REGION='us-west-2'
export ACCOUNT_ID=123456789012
export ROLE_ARN='arn:aws:iam::123456789012:role/batch_operations-objectlock'

read -d '' OPERATION <<EOF
{
  "S3PutObjectLegalHold": {
    "LegalHold": {
      "Status":"OFF"
    }
  }
}
EOF

read -d '' MANIFEST <<EOF
{
  "Spec": {
    "Format": "S3BatchOperations_CSV_20180820",
    "Fields": [
      "Bucket",
      "Key"
    ]
  },
  "Location": {
```

```
    "ObjectArn": "arn:aws:s3:::amzn-s3-demo-manifest-bucket/legalhold-object-
manifest.csv",
    "ETag": "Your-manifest-ETag"
  }
}
EOF

read -d '' REPORT <<EOF
{
  "Bucket": "arn:aws:s3:::amzn-s3-demo-completion-report-bucket",
  "Format": "Report_CSV_20180820",
  "Enabled": true,
  "Prefix": "reports/legalhold-objects-batch_operations",
  "ReportScope": "AllTasks"
}
EOF

aws \
    s3control create-job \
    --account-id "${ACCOUNT_ID}" \
    --manifest "${MANIFEST//$'\n'}" \
    --operation "${OPERATION//$'\n'/}" \
    --report "${REPORT//$'\n'}" \
    --priority 10 \
    --role-arn "${ROLE_ARN}" \
    --client-request-token "$(uuidgen)" \
    --region "${AWS_DEFAULT_REGION}" \
    --description "Turn off legal hold";
```

Verwendung des AWS SDK for Java

Die folgenden AWS SDK for Java Beispiele zeigen, wie Batch-Operationen verwendet werden, um die gesetzlichen Aufbewahrungsfristen von S3 Object Lock für mehrere Objekte zu deaktivieren.

Example — Aktualisiert die Rolle, um **`s3:PutObjectLegalHold`** Berechtigungen zu gewähren

```
public void allowPutObjectLegalHold() {
    final String roleName = "batch_operations-object-lock";

    final String legalHoldPermissions = "{" +
            "    \"Version\": \"2012-10-17\"," +
            "    \"Statement\": [" +
            "        {" +
```

```
            "            \"Effect\": \"Allow\"," +
            "            \"Action\": [" +
            "                \"s3:PutObjectLegalHold\"" +
            "            ]," +
            "            \"Resource\": [" +
            "                \"arn:aws:s3:::amzn-s3-demo-manifest-bucket/*\"" +
            "            ]" +
            "        }" +
            "    ]" +
            "}";

    final AmazonIdentityManagement iam =
            AmazonIdentityManagementClientBuilder.defaultClient();

    final PutRolePolicyRequest putRolePolicyRequest = new PutRolePolicyRequest()
            .withPolicyDocument(legalHoldPermissions)
            .withPolicyName("legal-hold-permissions")
            .withRoleName(roleName);

    final PutRolePolicyResult putRolePolicyResult =
 iam.putRolePolicy(putRolePolicyRequest);
}
```

Example — Schaltet die gesetzliche Sperre aus

Verwenden Sie das folgende Beispiel, wenn Sie die rechtliche Aufbewahrungsfrist deaktivieren möchten.

```
public String createLegalHoldOffJob(final AWSS3ControlClient awss3ControlClient) {
    final String manifestObjectArn = "arn:aws:s3:::amzn-s3-demo-manifest-bucket/
legalhold-object-manifest.csv";
    final String manifestObjectVersionId = "15ad5ba069e6bbc465c77bf83d541385";

    final JobManifestLocation manifestLocation = new JobManifestLocation()
            .withObjectArn(manifestObjectArn)
            .withETag(manifestObjectVersionId);

    final JobManifestSpec manifestSpec =
            new JobManifestSpec()
                    .withFormat(JobManifestFormat.S3BatchOperations_CSV_20180820)
                    .withFields("Bucket", "Key");

    final JobManifest manifestToPublicApi = new JobManifest()
```

```
            .withLocation(manifestLocation)
            .withSpec(manifestSpec);

    final String jobReportBucketArn = "arn:aws:s3:::amzn-s3-demo-completion-report-
bucket";
    final String jobReportPrefix = "reports/legalhold-objects-batch_operations";

    final JobReport jobReport = new JobReport()
            .withEnabled(true)
            .withReportScope(JobReportScope.AllTasks)
            .withBucket(jobReportBucketArn)
            .withPrefix(jobReportPrefix)
            .withFormat(JobReportFormat.Report_CSV_20180820);

    final JobOperation jobOperation = new JobOperation()
            .withS3PutObjectLegalHold(new S3SetObjectLegalHoldOperation()
                    .withLegalHold(new S3ObjectLockLegalHold()
                            .withStatus(S3ObjectLockLegalHoldStatus.OFF)));

    final String roleArn = "arn:aws:iam::123456789012:role/batch_operations-object-
lock";
    final Boolean requiresConfirmation = true;
    final int priority = 10;

    final CreateJobRequest request = new CreateJobRequest()
            .withAccountId("123456789012")
            .withDescription("Turn off legal hold")
            .withManifest(manifestToPublicApi)
            .withOperation(jobOperation)
            .withPriority(priority)
            .withRoleArn(roleArn)
            .withReport(jobReport)
            .withConfirmationRequired(requiresConfirmation);

    final CreateJobResult result = awss3ControlClient.createJob(request);

    return result.getJobId();
}
```

Tutorial: Batch-Transcodierung von Videos mit S3 Batch Operations

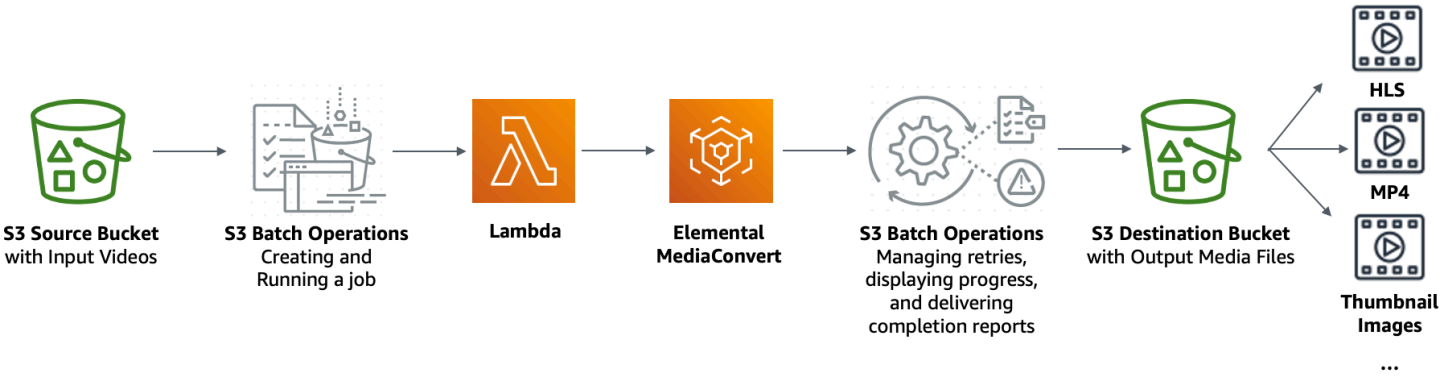
Videokonsumenten nutzen Geräte aller Formen, Größen und Jahrgänge, um Medieninhalte zu genießen. Diese breite Palette an Geräten stellt eine Herausforderung für Content-Ersteller und -

Distributoren dar. Videos müssen nicht in einem one-size-fits-all Format vorliegen, sondern müssen so konvertiert werden, dass sie ein breites Spektrum an Größen, Formaten und Bitraten abdecken können. Diese Konvertierungsaufgabe ist noch schwieriger, wenn Sie eine große Anzahl von Videos haben, die konvertiert werden müssen.

AWS bietet Ihnen eine Methode zum Aufbau einer skalierbaren, verteilten Architektur, die Folgendes bietet:

- Greift Eingabevideos auf
- Verarbeitet die Videos zur Wiedergabe auf einer Vielzahl von Geräten
- Speichert die transcodierten Mediendateien
- Liefert die Ausgabemediendateien und befriedigt Ansprüche

Wenn Sie umfangreiche Video-Repositories in Amazon S3 gespeichert haben, können Sie diese Videos aus ihren Quellformaten in mehrere Dateitypen in der Größe, Auflösung und Format transkodieren, die von einem bestimmten Videoplayer oder Gerät benötigt werden. Insbesondere bietet Ihnen S3 Batch Operations eine Lösung zum Aufrufen von AWS Lambda Funktionen für vorhandene Eingabevideos in einem S3-Quell-Bucket. Dann rufen die Lambda-Funktionen AWS Elemental MediaConvert auf, um umfangreiche Videotranskodierungsaufgaben auszuführen. Die konvertierten Ausgabe-Mediendateien werden in einem S3-Ziel-Bucket gespeichert.



Ziel

In diesem Lernprogramm erfahren Sie, wie Sie S3-Batch-Vorgänge einrichten, um eine Lambda-Funktion zum Batch-Transcodieren von Videos aufzurufen, die in einem S3-Quell-Bucket gespeichert sind. Die Lambda-Funktion ruft MediaConvert auf, um die Videos zu transkodieren. Die Ausgänge für jedes Video im S3-Quell-Bucket sind wie folgt:

- Ein HTTPLive-Streaming-Stream (HLS) mit adaptiver Bitrate für die Wiedergabe auf Geräten verschiedener Größen und unterschiedlicher Bandbreiten
- Eine Videodatei MP4
- In Intervallen gesammelte Miniaturbilder

Themen

- Voraussetzungen
- Schritt 1: Erstellen Sie einen S3-Bucket für Ausgabe-Mediendateien
- Schritt 2: Erstelle eine IAM Rolle für MediaConvert
- Schritt 3: Erstellen Sie eine IAM Rolle für Ihre Lambda-Funktion
- Schritt 4: Erstellen einer Lambda-Funktion für die Videotranscodierung
- Schritt 5: Konfigurieren des Amazon S3-Bestands für Ihren S3-Quell-Bucket
- Schritt 6: Eine IAM Rolle für S3 Batch Operations erstellen
- Schritt 7: Einrichten und Ausführen eines Auftrags für S3-Batchvorgänge
- Schritt 8: Überprüfen Sie die Ausgabe-Mediendateien aus Ihrem S3-Ziel-Bucket
- Schritt 9: Bereinigen
- Nächste Schritte

Voraussetzungen

Bevor Sie mit diesem Lernprogramm beginnen können, benötigen Sie einen Amazon S3-Quell-Bucket (z. B. *amzn-s3-demo-source-bucket*) mit Videos, die bereits darin transcodiert werden sollen.

Sie können dem Bucket einen anderen Namen geben, wenn Sie möchten. Weitere Informationen zu Amazon S3 Bucket-Namen finden Sie unter Allgemeine Benennungsregeln für Buckets.

Lassen Sie für den S3-Quell-Bucket die Einstellungen für öffentlichen Zugriff für diesen Bucket blockieren auf die Standardeinstellungen festlegen (Alle öffentlichen Zugriffe blockieren ist aktiviert). Weitere Informationen finden Sie unter Erstellen eines Buckets.

Weitere Informationen über das Hochladen von Videos in den S3-Quell-Bucket finden Sie unter Objekte hochladen. Wenn Sie ein Video in S3 hochladen, können Sie auch Amazon S3 Transfer

Acceleration nutzen, um schnelle und sichere Dateiübertragungen zu konfigurieren. Transfer Acceleration kann das Hochladen von Videos zu Ihrem S3-Bucket zur Übertragung größerer Videos über große Entfernungen beschleunigen. Weitere Informationen finden Sie unter Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration.

Schritt 1: Erstellen Sie einen S3-Bucket für Ausgabe-Mediendateien

In diesem Schritt erstellen Sie einen S3-Ziel-Bucket, um die konvertierten Ausgabe-Mediendateien zu speichern. Sie erstellen auch eine Cross-Origin-Konfiguration für die gemeinsame Nutzung von Ressourcen (CORS), um den ursprungsübergreifenden Zugriff auf die transkodierten Mediendateien zu ermöglichen, die in Ihrem S3-Ziel-Bucket gespeichert sind.

Teilschritte

- Erstellen Sie einen Bucket für die Ausgabe-Mediendateien
- Fügen Sie dem S3-Ausgabe-Bucket eine CORS Konfiguration hinzu

Erstellen Sie einen Bucket für die Ausgabe-Mediendateien

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie Create Bucket (Bucket erstellen) aus.
4. Geben Sie im Feld Bucket Name einen Namen für Ihren Bucket ein (z. B. *amzn-s3-demo-destination-bucket1*).
5. Wählen Sie unter Region den AWS-Region Ort aus, an dem sich der Bucket befinden soll.
6. Um den öffentlichen Zugriff auf Ihre Ausgabemediendateien zu gewährleisten, deaktivieren Sie unter Einstellungen für den öffentlichen Zugriff für diesen Bucket blockieren Alle öffentlichen Zugriffe blockieren.

> **Warning**
>
> Bevor Sie diesen Schritt ausführen, lesen Sie den Abschnitt Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher, um sicherzustellen, dass Sie die mit dem Zulassen eines öffentlichen Zugriffs verbundenen Risiken kennen und akzeptieren. Wenn Sie die Einstellungen für Block Public Access deaktivieren, um Ihren Bucket öffentlich

zu machen, kann jeder im Internet auf Ihren Bucket zugreifen. Wir empfehlen Ihnen, den gesamten öffentlichen Zugriff auf Ihre Buckets zu blockieren.
Wenn Sie die Einstellungen zum Blockieren des öffentlichen Zugriffs nicht löschen möchten, können Sie Amazon verwenden, CloudFront um die transkodierten Mediendateien an Zuschauer (Endbenutzer) zu senden. Weitere Informationen finden Sie unter Tutorial: Hosten von On-Demand-Streaming-Videos mit Amazon S3 CloudFront, Amazon und Amazon Route 53.

7. Aktivieren Sie das Kontrollkästchen neben I acknowledge that the current settings may result in this bucket and the objects within becoming public (Ich bestätige, dass die aktuellen Einstellungen dazu führen können, dass dieser Bucket und die darin enthaltenen Objekte öffentlich werden).
8. Lassen Sie die restlichen Einstellungen bei den Standardwerten.
9. Wählen Sie Bucket erstellen aus.

Fügen Sie dem S3-Ausgabe-Bucket eine CORS Konfiguration hinzu

Eine JSON CORS Konfiguration definiert eine Methode für Client-Webanwendungen (in diesem Kontext Videoplayer), die in eine Domain geladen werden, transkodierte Ausgabemediendateien in einer anderen Domain abzuspielen.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste der Buckets den Namen des Buckets, den Sie zuvor erstellt haben (z. B. *amzn-s3-demo-destination-bucket1*).
4. Wählen Sie die Registerkarte Berechtigungen.
5. Wählen Sie im Abschnitt Cross-Origin Resource Sharing (CORS) die Option Bearbeiten aus.
6. Kopieren Sie die folgende CORS Konfiguration und fügen Sie sie in das Textfeld für die CORS Konfiguration ein.

   Die CORS Konfiguration muss im JSON Format vorliegen. In diesem Beispiel verwendet das `AllowedOrigins`-Attribut das Platzhalterzeichen (`*`) um alle Ursprünge anzugeben. Wenn du deine spezifische Herkunft kennst, kannst du das `AllowedOrigins` Attribut auf deinen spezifischen Spieler beschränkenURL. Weitere Informationen zum Konfigurieren von diesem und anderen Attributen finden Sie unter Elemente einer CORS Konfiguration.

```
[
    {
        "AllowedOrigins": [
            "*"
        ],
        "AllowedMethods": [
            "GET"
        ],
        "AllowedHeaders": [
            "*"
        ],
        "ExposeHeaders": []

    }
]
```

7. Wählen Sie Änderungen speichern.

Schritt 2: Erstelle eine IAM Rolle für MediaConvert

Um die in Ihrem S3-Bucket gespeicherten Eingabevideos AWS Elemental MediaConvert zu transkodieren, benötigen Sie eine AWS Identity and Access Management (IAM) -Servicerolle, um MediaConvert Berechtigungen zum Lesen und Schreiben von Videodateien aus und in Ihre S3-Quell- und Ziel-Buckets zu erteilen. Wenn Sie Transcodierungsaufträge ausführen, verwendet die MediaConvert Konsole diese Rolle.

Um eine IAM Rolle zu erstellen für MediaConvert

1. Erstellen Sie eine IAM Rolle mit einem Rollennamen, den Sie wählen (z. B.**`tutorial-mediaconvert-role`**). Um diese Rolle zu erstellen, folgen Sie den Schritten unter Erstellen Sie Ihre MediaConvert Rolle in IAM (Konsole) im AWS Elemental MediaConvert Benutzerhandbuch.
2. Nachdem Sie die IAM Rolle für erstellt haben MediaConvert, wählen Sie in der Rollenliste den Namen der Rolle aus MediaConvert , für die Sie erstellt haben (z. B.**`tutorial-mediaconvert-role`**).
3. Kopieren Sie auf der Übersichtsseite die Rolle ARN (die mit beginnt`arn:aws:iam::`) und speichern Sie sie ARN zur späteren Verwendung.

   Weitere Informationen zu ARNs finden Sie unter Amazon Resource Names (ARNs) in der AWS Allgemeinen Referenz.

Schritt 3: Erstellen Sie eine IAM Rolle für Ihre Lambda-Funktion

Um Videos mit MediaConvert und S3 Batch Operations stapelweise zu transkodieren, verwenden Sie eine Lambda-Funktion, um diese beiden Dienste zu verbinden, um Videos zu konvertieren. Diese Lambda-Funktion muss über eine IAM Rolle verfügen, die der Lambda-Funktion Zugriffsberechtigungen MediaConvert und S3-Stapeloperationen gewährt.

Teilschritte

- Erstellen Sie eine IAM Rolle für Ihre Lambda-Funktion
- Betten Sie eine Inline-Richtlinie für die IAM Rolle Ihrer Lambda-Funktion ein

Erstellen Sie eine IAM Rolle für Ihre Lambda-Funktion

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Roles (Rollen) und dann Create Role (Rolle erstellen) aus.
3. Wählen Sie den AWS -Service-Rollentyp und wählen Sie dann unter Häufige Anwendungsfälle Lambda aus.
4. Wählen Sie Weiter: Berechtigungen aus.
5. Geben Sie auf der Seite Attach permissions (Berechtigungen anfügen) **`AWSLambdaBasicExecutionRole`** in das Suchfeld ein, um die Liste der Richtlinien zu filtern. Um die verwaltete Richtlinie AWSLambdaBasicExecutionRolean diese Rolle anzuhängen, um Schreibberechtigungen für Amazon CloudWatch Logs zu gewähren, aktivieren Sie das Kontrollkästchen neben AWSLambdaBasicExecutionRole.
6. Wählen Sie Weiter: Markierungen.
7. (Optional) Fügen Sie der verwalteten Richtlinie Tags hinzu.
8. Wählen Sie Weiter: Prüfen aus.
9. Geben Sie für Role name (Rollenname) den Namen **`tutorial-lambda-transcode-role`** ein.
10. Wählen Sie Create role (Rolle erstellen) aus.

Betten Sie eine Inline-Richtlinie für die IAM Rolle Ihrer Lambda-Funktion ein

Um der MediaConvert Ressource, die für die Ausführung der Lambda-Funktion benötigt wird, Berechtigungen zu gewähren, müssen Sie eine Inline-Richtlinie verwenden.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Roles aus.
3. Wählen Sie in der Liste Rollen den Namen der IAM Rolle aus, die Sie zuvor für Ihre Lambda-Funktion erstellt haben (z. B.**`tutorial-lambda-transcode-role`**).
4. Wählen Sie den Tab Permissions (Berechtigungen).
5. Wählen Sie Add inline Policy (Inline-Richtlinie auswählen).
6. Wählen Sie die JSONRegisterkarte aus, kopieren Sie dann die folgende JSON Richtlinie und fügen Sie sie ein.

   Ersetzen Sie in der JSON Richtlinie den ARN Beispielwert von `Resource` durch die Rolle ARN der IAM Rolle MediaConvert , für die Sie in Schritt 2 erstellt haben (z. B.**`tutorial-mediaconvert-role`**).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Sid": "Logging"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::111122223333:role/tutorial-mediaconvert-role"
            ],
            "Effect": "Allow",
```

```
            "Sid": "PassRole"
        },
        {
            "Action": [
                "mediaconvert:*"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "MediaConvertService"
        },
        {
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "*"
            ],
            "Effect": "Allow",
            "Sid": "S3Service"
        }
    ]
}
```

7. Wählen Sie Review policy (Richtlinie überprüfen) aus.
8. Geben Sie unter Name **`tutorial-lambda-policy`** ein.
9. Wählen Sie Create Policy (Richtlinie erstellen) aus.

   Nachdem Sie eine Inline-Richtlinie erstellt haben, wird sie automatisch in die IAM Rolle Ihrer Lambda-Funktion eingebettet.

Schritt 4: Erstellen einer Lambda-Funktion für die Videotranscodierung

In diesem Abschnitt des Tutorials erstellen Sie eine Lambda-Funktion mithilfe von SDK for Python zur Integration mit S3 Batch Operations und MediaConvert. Um mit der Transcodierung der Videos zu beginnen, die bereits in Ihrem S3-Quell-Bucket gespeichert sind, führen Sie einen S3-Batchvorgang-Auftrag aus, der direkt die Lambda-Funktion für jedes Video im S3-Quell-Bucket aufruft. Anschließend sendet die Lambda-Funktion einen Transcodierungsauftrag für jedes Video an. MediaConvert

Teilschritte

- Schreiben von Lambda-Funktionscode und Erstellen eines Bereitstellungspakets
- Erstellen einer Lambda-Funktion mit einer Ausführungsrolle (Konsole)
- Stellen Sie Ihre Lambda-Funktion mit ZIP-Dateiarchiven bereit und konfigurieren Sie die Lambda-Funktion (Konsole)

Schreiben von Lambda-Funktionscode und Erstellen eines Bereitstellungspakets

1. Erstellen Sie auf Ihrem lokalen Computer einen Ordner mit dem Namen `batch-transcode`.
2. Erstellen Sie im `batch-transcode` Ordner eine Datei mit JSON den Auftragseinstellungen. Sie können z. B. die Einstellungen in diesem Abschnitt verwenden und die Datei `job.json` benennen.

   Eine `job.json`-Datei gibt Folgendes an:

   - Welche Dateien transcodiert werden sollen
   - Wie Sie Ihre Eingabevideos transcodieren möchten
   - Welche Ausgabemediendateien Sie erstellen möchten
   - Wie die transcodierten Dateien benannt werden sollen
   - Wo die transcodierten Dateien gespeichert werden sollen
   - Welche fortschrittlichen Funktionen angewendet werden sollen und so weiter

   In diesem Tutorial wird die folgende `job.json`-Datei verwendet, um die folgenden Ausgaben für jedes Video im S3-Quell-Bucket zu erstellen:

   - Ein HTTP Live-Streaming-Stream (HLS) mit adaptiver Bitrate für die Wiedergabe auf mehreren Geräten unterschiedlicher Größe und Bandbreite
   - Eine Videodatei MP4
   - In Intervallen gesammelte Miniaturbilder

   Diese `job.json` Beispieldatei verwendet die qualitätsdefinierte variable Bitrate (QVBR), um die Videoqualität zu optimieren. Die HLS Ausgabe ist Apple-kompatibel (Audio ohne Mischung aus Video, Segmentdauer von 6 Sekunden und optimierte Videoqualität durch Auto). QVBR

Wenn Sie die hier bereitgestellten Beispieleinstellungen nicht verwenden möchten, können Sie eine `job.json`-Spezifikation basierend auf Ihrem Anwendungsfall erstellen. Stellen Sie sicher, dass die Eingabedateien ähnliche Video- und Audiokonfigurationen aufweisen, um die Konsistenz Ihrer Ausgänge zu gewährleisten. Für beliebige Eingabedateien mit unterschiedlichen Video- und Audiokonfigurationen erstellen Sie getrennte Automatisierungen (eindeutige `job.json`-Einstellungen). Weitere Informationen finden Sie im JSON Benutzerhandbuch unter Beispiele für AWS Elemental MediaConvert Jobeinstellungen.AWS Elemental MediaConvert

```
{
  "OutputGroups": [
    {
      "CustomName": "HLS",
      "Name": "Apple HLS",
      "Outputs": [
        {
          "ContainerSettings": {
            "Container": "M3U8",
            "M3u8Settings": {
              "AudioFramesPerPes": 4,
              "PcrControl": "PCR_EVERY_PES_PACKET",
              "PmtPid": 480,
              "PrivateMetadataPid": 503,
              "ProgramNumber": 1,
              "PatInterval": 0,
              "PmtInterval": 0,
              "TimedMetadata": "NONE",
              "VideoPid": 481,
              "AudioPids": [
                482,
                483,
                484,
                485,
                486,
                487,
                488,
                489,
                490,
                491,
                492
              ]
            }
```

```
            },
            "VideoDescription": {
              "Width": 640,
              "ScalingBehavior": "DEFAULT",
              "Height": 360,
              "TimecodeInsertion": "DISABLED",
              "AntiAlias": "ENABLED",
              "Sharpness": 50,
              "CodecSettings": {
                "Codec": "H_264",
                "H264Settings": {
                  "InterlaceMode": "PROGRESSIVE",
                  "NumberReferenceFrames": 3,
                  "Syntax": "DEFAULT",
                  "Softness": 0,
                  "GopClosedCadence": 1,
                  "GopSize": 2,
                  "Slices": 1,
                  "GopBReference": "DISABLED",
                  "MaxBitrate": 1200000,
                  "SlowPal": "DISABLED",
                  "SpatialAdaptiveQuantization": "ENABLED",
                  "TemporalAdaptiveQuantization": "ENABLED",
                  "FlickerAdaptiveQuantization": "DISABLED",
                  "EntropyEncoding": "CABAC",
                  "FramerateControl": "INITIALIZE_FROM_SOURCE",
                  "RateControlMode": "QVBR",
                  "CodecProfile": "MAIN",
                  "Telecine": "NONE",
                  "MinIInterval": 0,
                  "AdaptiveQuantization": "HIGH",
                  "CodecLevel": "AUTO",
                  "FieldEncoding": "PAFF",
                  "SceneChangeDetect": "TRANSITION_DETECTION",
                  "QualityTuningLevel": "SINGLE_PASS_HQ",
                  "FramerateConversionAlgorithm": "DUPLICATE_DROP",
                  "UnregisteredSeiTimecode": "DISABLED",
                  "GopSizeUnits": "SECONDS",
                  "ParControl": "INITIALIZE_FROM_SOURCE",
                  "NumberBFramesBetweenReferenceFrames": 2,
                  "RepeatPps": "DISABLED"
                }
              },
              "AfdSignaling": "NONE",
```

```
            "DropFrameTimecode": "ENABLED",
            "RespondToAfd": "NONE",
            "ColorMetadata": "INSERT"
          },
          "OutputSettings": {
            "HlsSettings": {
              "AudioGroupId": "program_audio",
              "AudioRenditionSets": "program_audio",
              "SegmentModifier": "$dt$",
              "IFrameOnlyManifest": "EXCLUDE"
            }
          },
          "NameModifier": "_360"
        },
        {
          "ContainerSettings": {
            "Container": "M3U8",
            "M3u8Settings": {
              "AudioFramesPerPes": 4,
              "PcrControl": "PCR_EVERY_PES_PACKET",
              "PmtPid": 480,
              "PrivateMetadataPid": 503,
              "ProgramNumber": 1,
              "PatInterval": 0,
              "PmtInterval": 0,
              "TimedMetadata": "NONE",
              "TimedMetadataPid": 502,
              "VideoPid": 481,
              "AudioPids": [
                482,
                483,
                484,
                485,
                486,
                487,
                488,
                489,
                490,
                491,
                492
              ]
            }
          },
          "VideoDescription": {
```

```
            "Width": 960,
            "ScalingBehavior": "DEFAULT",
            "Height": 540,
            "TimecodeInsertion": "DISABLED",
            "AntiAlias": "ENABLED",
            "Sharpness": 50,
            "CodecSettings": {
              "Codec": "H_264",
              "H264Settings": {
                "InterlaceMode": "PROGRESSIVE",
                "NumberReferenceFrames": 3,
                "Syntax": "DEFAULT",
                "Softness": 0,
                "GopClosedCadence": 1,
                "GopSize": 2,
                "Slices": 1,
                "GopBReference": "DISABLED",
                "MaxBitrate": 3500000,
                "SlowPal": "DISABLED",
                "SpatialAdaptiveQuantization": "ENABLED",
                "TemporalAdaptiveQuantization": "ENABLED",
                "FlickerAdaptiveQuantization": "DISABLED",
                "EntropyEncoding": "CABAC",
                "FramerateControl": "INITIALIZE_FROM_SOURCE",
                "RateControlMode": "QVBR",
                "CodecProfile": "MAIN",
                "Telecine": "NONE",
                "MinIInterval": 0,
                "AdaptiveQuantization": "HIGH",
                "CodecLevel": "AUTO",
                "FieldEncoding": "PAFF",
                "SceneChangeDetect": "TRANSITION_DETECTION",
                "QualityTuningLevel": "SINGLE_PASS_HQ",
                "FramerateConversionAlgorithm": "DUPLICATE_DROP",
                "UnregisteredSeiTimecode": "DISABLED",
                "GopSizeUnits": "SECONDS",
                "ParControl": "INITIALIZE_FROM_SOURCE",
                "NumberBFramesBetweenReferenceFrames": 2,
                "RepeatPps": "DISABLED"
              }
            },
            "AfdSignaling": "NONE",
            "DropFrameTimecode": "ENABLED",
            "RespondToAfd": "NONE",
```

```
            "ColorMetadata": "INSERT"
          },
          "OutputSettings": {
            "HlsSettings": {
              "AudioGroupId": "program_audio",
              "AudioRenditionSets": "program_audio",
              "SegmentModifier": "$dt$",
              "IFrameOnlyManifest": "EXCLUDE"
            }
          },
          "NameModifier": "_540"
        },
        {
          "ContainerSettings": {
            "Container": "M3U8",
            "M3u8Settings": {
              "AudioFramesPerPes": 4,
              "PcrControl": "PCR_EVERY_PES_PACKET",
              "PmtPid": 480,
              "PrivateMetadataPid": 503,
              "ProgramNumber": 1,
              "PatInterval": 0,
              "PmtInterval": 0,
              "TimedMetadata": "NONE",
              "VideoPid": 481,
              "AudioPids": [
                482,
                483,
                484,
                485,
                486,
                487,
                488,
                489,
                490,
                491,
                492
              ]
            }
          },
          "VideoDescription": {
            "Width": 1280,
            "ScalingBehavior": "DEFAULT",
            "Height": 720,
```

```
            "TimecodeInsertion": "DISABLED",
            "AntiAlias": "ENABLED",
            "Sharpness": 50,
            "CodecSettings": {
              "Codec": "H_264",
              "H264Settings": {
                "InterlaceMode": "PROGRESSIVE",
                "NumberReferenceFrames": 3,
                "Syntax": "DEFAULT",
                "Softness": 0,
                "GopClosedCadence": 1,
                "GopSize": 2,
                "Slices": 1,
                "GopBReference": "DISABLED",
                "MaxBitrate": 5000000,
                "SlowPal": "DISABLED",
                "SpatialAdaptiveQuantization": "ENABLED",
                "TemporalAdaptiveQuantization": "ENABLED",
                "FlickerAdaptiveQuantization": "DISABLED",
                "EntropyEncoding": "CABAC",
                "FramerateControl": "INITIALIZE_FROM_SOURCE",
                "RateControlMode": "QVBR",
                "CodecProfile": "MAIN",
                "Telecine": "NONE",
                "MinIInterval": 0,
                "AdaptiveQuantization": "HIGH",
                "CodecLevel": "AUTO",
                "FieldEncoding": "PAFF",
                "SceneChangeDetect": "TRANSITION_DETECTION",
                "QualityTuningLevel": "SINGLE_PASS_HQ",
                "FramerateConversionAlgorithm": "DUPLICATE_DROP",
                "UnregisteredSeiTimecode": "DISABLED",
                "GopSizeUnits": "SECONDS",
                "ParControl": "INITIALIZE_FROM_SOURCE",
                "NumberBFramesBetweenReferenceFrames": 2,
                "RepeatPps": "DISABLED"
              }
            },
            "AfdSignaling": "NONE",
            "DropFrameTimecode": "ENABLED",
            "RespondToAfd": "NONE",
            "ColorMetadata": "INSERT"
          },
          "OutputSettings": {
```

```
            "HlsSettings": {
              "AudioGroupId": "program_audio",
              "AudioRenditionSets": "program_audio",
              "SegmentModifier": "$dt$",
              "IFrameOnlyManifest": "EXCLUDE"
            }
          },
          "NameModifier": "_720"
        },
        {
          "ContainerSettings": {
            "Container": "M3U8",
            "M3u8Settings": {}
          },
          "AudioDescriptions": [
            {
              "AudioSourceName": "Audio Selector 1",
              "CodecSettings": {
                "Codec": "AAC",
                "AacSettings": {
                  "Bitrate": 96000,
                  "CodingMode": "CODING_MODE_2_0",
                  "SampleRate": 48000
                }
              }
            }
          ],
          "OutputSettings": {
            "HlsSettings": {
              "AudioGroupId": "program_audio",
              "AudioTrackType": "ALTERNATE_AUDIO_AUTO_SELECT_DEFAULT"
            }
          },
          "NameModifier": "_audio"
        }
      ],
      "OutputGroupSettings": {
        "Type": "HLS_GROUP_SETTINGS",
        "HlsGroupSettings": {
          "ManifestDurationFormat": "INTEGER",
          "SegmentLength": 6,
          "TimedMetadataId3Period": 10,
          "CaptionLanguageSetting": "OMIT",
          "Destination": "s3://EXAMPLE-BUCKET/HLS/",
```

```
          "DestinationSettings": {
            "S3Settings": {
              "AccessControl": {
                "CannedAcl": "PUBLIC_READ"
              }
            }
          },
          "TimedMetadataId3Frame": "PRIV",
          "CodecSpecification": "RFC_4281",
          "OutputSelection": "MANIFESTS_AND_SEGMENTS",
          "ProgramDateTimePeriod": 600,
          "MinSegmentLength": 0,
          "DirectoryStructure": "SINGLE_DIRECTORY",
          "ProgramDateTime": "EXCLUDE",
          "SegmentControl": "SEGMENTED_FILES",
          "ManifestCompression": "NONE",
          "ClientCache": "ENABLED",
          "StreamInfResolution": "INCLUDE"
        }
      }
    },
    {
      "CustomName": "MP4",
      "Name": "File Group",
      "Outputs": [
        {
          "ContainerSettings": {
            "Container": "MP4",
            "Mp4Settings": {
              "CslgAtom": "INCLUDE",
              "FreeSpaceBox": "EXCLUDE",
              "MoovPlacement": "PROGRESSIVE_DOWNLOAD"
            }
          },
          "VideoDescription": {
            "Width": 1280,
            "ScalingBehavior": "DEFAULT",
            "Height": 720,
            "TimecodeInsertion": "DISABLED",
            "AntiAlias": "ENABLED",
            "Sharpness": 100,
            "CodecSettings": {
              "Codec": "H_264",
              "H264Settings": {
```

```
                "InterlaceMode": "PROGRESSIVE",
                "ParNumerator": 1,
                "NumberReferenceFrames": 3,
                "Syntax": "DEFAULT",
                "Softness": 0,
                "GopClosedCadence": 1,
                "HrdBufferInitialFillPercentage": 90,
                "GopSize": 2,
                "Slices": 2,
                "GopBReference": "ENABLED",
                "HrdBufferSize": 10000000,
                "MaxBitrate": 5000000,
                "ParDenominator": 1,
                "EntropyEncoding": "CABAC",
                "RateControlMode": "QVBR",
                "CodecProfile": "HIGH",
                "MinIInterval": 0,
                "AdaptiveQuantization": "AUTO",
                "CodecLevel": "AUTO",
                "FieldEncoding": "PAFF",
                "SceneChangeDetect": "ENABLED",
                "QualityTuningLevel": "SINGLE_PASS_HQ",
                "UnregisteredSeiTimecode": "DISABLED",
                "GopSizeUnits": "SECONDS",
                "ParControl": "SPECIFIED",
                "NumberBFramesBetweenReferenceFrames": 3,
                "RepeatPps": "DISABLED",
                "DynamicSubGop": "ADAPTIVE"
              }
            },
            "AfdSignaling": "NONE",
            "DropFrameTimecode": "ENABLED",
            "RespondToAfd": "NONE",
            "ColorMetadata": "INSERT"
          },
          "AudioDescriptions": [
            {
              "AudioTypeControl": "FOLLOW_INPUT",
              "AudioSourceName": "Audio Selector 1",
              "CodecSettings": {
                "Codec": "AAC",
                "AacSettings": {
                  "AudioDescriptionBroadcasterMix": "NORMAL",
                  "Bitrate": 160000,
```

```
                "RateControlMode": "CBR",
                "CodecProfile": "LC",
                "CodingMode": "CODING_MODE_2_0",
                "RawFormat": "NONE",
                "SampleRate": 48000,
                "Specification": "MPEG4"
              }
            },
            "LanguageCodeControl": "FOLLOW_INPUT",
            "AudioType": 0
          }
        ]
      }
    ],
    "OutputGroupSettings": {
      "Type": "FILE_GROUP_SETTINGS",
      "FileGroupSettings": {
        "Destination": "s3://EXAMPLE-BUCKET/MP4/",
        "DestinationSettings": {
          "S3Settings": {
            "AccessControl": {
              "CannedAcl": "PUBLIC_READ"
            }
          }
        }
      }
    }
  },
  {
    "CustomName": "Thumbnails",
    "Name": "File Group",
    "Outputs": [
      {
        "ContainerSettings": {
          "Container": "RAW"
        },
        "VideoDescription": {
          "Width": 1280,
          "ScalingBehavior": "DEFAULT",
          "Height": 720,
          "TimecodeInsertion": "DISABLED",
          "AntiAlias": "ENABLED",
          "Sharpness": 50,
          "CodecSettings": {
```

```
            "Codec": "FRAME_CAPTURE",
            "FrameCaptureSettings": {
              "FramerateNumerator": 1,
              "FramerateDenominator": 5,
              "MaxCaptures": 500,
              "Quality": 80
            }
          },
          "AfdSignaling": "NONE",
          "DropFrameTimecode": "ENABLED",
          "RespondToAfd": "NONE",
          "ColorMetadata": "INSERT"
        }
      }
    ],
    "OutputGroupSettings": {
      "Type": "FILE_GROUP_SETTINGS",
      "FileGroupSettings": {
        "Destination": "s3://EXAMPLE-BUCKET/Thumbnails/",
        "DestinationSettings": {
          "S3Settings": {
            "AccessControl": {
              "CannedAcl": "PUBLIC_READ"
            }
          }
        }
      }
    }
  }
],
"AdAvailOffset": 0,
"Inputs": [
  {
    "AudioSelectors": {
      "Audio Selector 1": {
        "Offset": 0,
        "DefaultSelection": "DEFAULT",
        "ProgramSelection": 1
      }
    },
    "VideoSelector": {
      "ColorSpace": "FOLLOW"
    },
    "FilterEnable": "AUTO",
```

```
      "PsiControl": "USE_PSI",
      "FilterStrength": 0,
      "DeblockFilter": "DISABLED",
      "DenoiseFilter": "DISABLED",
      "TimecodeSource": "EMBEDDED",
      "FileInput": "s3://EXAMPLE-INPUT-BUCKET/input.mp4"
    }
  ]
}
```

3. Erstellen Sie im `batch-transcode`-Ordner eine Datei mit einer Lambda-Funktion. Sie können das folgende Python-Beispiel verwenden und die Datei `convert.py` nennen.

   S3-Batchvorgänge sendet bestimmte Aufgabendaten an eine Lambda-Funktion und erfordert Ergebnisdaten zurück. Anforderungs- und Antwortbeispielen für die Lambda-Funktion, Informationen über Antwort- und Ergebniscodes sowie Beispiel-Lambda-Funktionen für S3-Batchvorgänge finden Sie unter Funktion aufrufen AWS Lambda.

```
import json
import os
from urllib.parse import urlparse
import uuid
import boto3

"""
When you run an S3 Batch Operations job, your job
invokes this Lambda function. Specifically, the Lambda function is
invoked on each video object listed in the manifest that you specify
for the S3 Batch Operations job in Step 5.

Input parameter "event": The S3 Batch Operations event as a request
                         for the Lambda function.

Input parameter "context": Context about the event.

Output: A result structure that Amazon S3 uses to interpret the result
        of the operation. It is a job response returned back to S3 Batch
 Operations.
"""
def handler(event, context):

    invocation_schema_version = event['invocationSchemaVersion']
    invocation_id = event['invocationId']
```

```
    task_id = event['tasks'][0]['taskId']

    source_s3_key = event['tasks'][0]['s3Key']
    source_s3_bucket = event['tasks'][0]['s3BucketArn'].split(':::')[-1]
    source_s3 = 's3://' + source_s3_bucket + '/' + source_s3_key

    result_list = []
    result_code = 'Succeeded'
    result_string = 'The input video object was converted successfully.'

    # The type of output group determines which media players can play
    # the files transcoded by MediaConvert.
    # For more information, see Creating outputs with AWS Elemental MediaConvert.
    output_group_type_dict = {
        'HLS_GROUP_SETTINGS': 'HlsGroupSettings',
        'FILE_GROUP_SETTINGS': 'FileGroupSettings',
        'CMAF_GROUP_SETTINGS': 'CmafGroupSettings',
        'DASH_ISO_GROUP_SETTINGS': 'DashIsoGroupSettings',
        'MS_SMOOTH_GROUP_SETTINGS': 'MsSmoothGroupSettings'
    }

    try:
        job_name = 'Default'
        with open('job.json') as file:
            job_settings = json.load(file)

        job_settings['Inputs'][0]['FileInput'] = source_s3

        # The path of each output video is constructed based on the values of
        # the attributes in each object of OutputGroups in the job.json file.
        destination_s3 = 's3://{0}/{1}/{2}' \
            .format(os.environ['amzn-s3-demo-destination-bucket'],
                    os.path.splitext(os.path.basename(source_s3_key))[0],
                    os.path.splitext(os.path.basename(job_name))[0])

        for output_group in job_settings['OutputGroups']:
            output_group_type = output_group['OutputGroupSettings']['Type']
            if output_group_type in output_group_type_dict.keys():
                output_group_type = output_group_type_dict[output_group_type]
                output_group['OutputGroupSettings'][output_group_type]
['Destination'] = \
                    "{0}{1}".format(destination_s3,
                                    urlparse(output_group['OutputGroupSettings']
[output_group_type]['Destination']).path)
```

```
            else:
                raise ValueError("Exception: Unknown Output Group Type {}."
                                 .format(output_group_type))

        job_metadata_dict = {
            'assetID': str(uuid.uuid4()),
            'application': os.environ['Application'],
            'input': source_s3,
            'settings': job_name
        }

        region = os.environ['AWS_DEFAULT_REGION']
        endpoints = boto3.client('mediaconvert', region_name=region) \
            .describe_endpoints()
        client = boto3.client('mediaconvert', region_name=region,
                              endpoint_url=endpoints['Endpoints'][0]['Url'],
                              verify=False)

        try:
            client.create_job(Role=os.environ['MediaConvertRole'],
                              UserMetadata=job_metadata_dict,
                              Settings=job_settings)
        # You can customize error handling based on different error codes that
        # MediaConvert can return.
        # For more information, see MediaConvert error codes.
        # When the result_code is TemporaryFailure, S3 Batch Operations retries
        # the task before the job is completed. If this is the final retry,
        # the error message is included in the final report.
        except Exception as error:
            result_code = 'TemporaryFailure'
            raise

    except Exception as error:
        if result_code != 'TemporaryFailure':
            result_code = 'PermanentFailure'
        result_string = str(error)

    finally:
        result_list.append({
            'taskId': task_id,
            'resultCode': result_code,
            'resultString': result_string,
        })
```

```
    return {
        'invocationSchemaVersion': invocation_schema_version,
        'treatMissingKeyAs': 'PermanentFailure',
        'invocationId': invocation_id,
        'results': result_list
    }
```

4. Um ein Bereitstellungspaket mit `convert.py` und `job.json` als `.zip`-Datei mit dem Namen `lambda.zip` zu erstellen, öffnen Sie in Ihrem lokalen Terminal den `batch-transcode`-Ordner, den Sie bereits erstellt haben, und führen Sie den folgenden Befehl aus.

   Führen Sie für macOS-Benutzer den folgenden Befehl aus:

   ```
   zip -r lambda.zip convert.py job.json
   ```

   Führen Sie für Windows-Benutzer die folgenden Befehle aus:

   ```
   powershell Compress-Archive convert.py lambda.zip
   ```

   ```
   powershell Compress-Archive -update job.json lambda.zip
   ```

Erstellen einer Lambda-Funktion mit einer Ausführungsrolle (Konsole)

1. Öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.
2. Wählen Sie im linken Navigationsbereich die Option Functions (Funktionen) aus.
3. Wählen Sie Create function (Funktion erstellen).
4. Wählen Sie Author from scratch aus.
5. Führen Sie unter Basic information (Grundlegende Informationen) die folgenden Schritte aus:
   a. Geben Sie für Function name (Funktionsname) **`tutorial-lambda-convert`** ein.
   b. Wählen Sie für Laufzeit Python 3.8 oder eine höhere Version von Python aus.
6. Wählen Sie Ändern der standardmäßigen Ausführungsrolle und wählen Sie unter Ausführungsrolle Verwenden einer vorhandenen Rolle.
7. Wählen Sie unter Existierende Rolle den Namen der IAM Rolle aus, die Sie in Schritt 3 für Ihre Lambda-Funktion erstellt haben (z. B.**`tutorial-lambda-transcode-role`**).
8. Lassen Sie die restlichen Einstellungen auf die Standardwerte eingestellt.

9. Wählen Sie Funktion erstellen aus.

Stellen Sie Ihre Lambda-Funktion mit ZIP-Dateiarchiven bereit und konfigurieren Sie die Lambda-Funktion (Konsole)

1. Wählen Sie im Abschnitt Code-Quelle der Seite für die Lambda-Funktion, die Sie erstellt haben (z. B. **`tutorial-lambda-convert`**), Hochladen von und dann ZIP-Datei aus.
2. Wählen Sie Upload (Hochladen) aus, um Ihre lokale `.zip`-Datei auszuwählen.
3. Wählen Sie die zuvor erstellte `lambda.zip`-Datei und klicken Sie auf Öffnen.
4. Wählen Sie Speichern aus.
5. Wählen Sie im Abschnitt Runtime settings (Laufzeiteinstellungen) die OptionBearbeiten aus.
6. Um der Lambda-Laufzeit mitzuteilen, welche Handler-Methode in Ihrem Lambda-Funktionscode aufgerufen werden soll, geben Sie **`convert.handler`** in das Feld Handler ein.

   Wenn Sie eine Funktion in Python konfigurieren, besteht der Wert der Handler-Einstellung aus dem Dateinamen und dem Namen des Handler-Moduls, getrennt durch einen Punkt (`.`). Beispielsweise ruft `convert.handler` die Methode `handler` auf, die in der Datei `convert.py` definiert ist.
7. Wählen Sie Save (Speichern) aus.
8. Wählen Sie auf der Seite der Lambda-Funktion den Tab Konfiguration. Im linken Navigationsbereich auf der Registerkarte Konfiguration wählen Sie Umgebungsvariablen und wählen dann Bearbeiten aus.
9. Wählen Sie Umgebungsvariablen hinzufügen aus. Geben Sie dann den angegebenen Schlüssel und Wert für jede der folgenden Umgebungsvariablen ein:

   - Schlüssel: **`DestinationBucket`** Wert: **`amzn-s3-demo-destination-bucket1`**

     Dieser Wert ist der S3 Bucket für Ausgabe-Mediendateien, die Sie in Schritt 1 erstellt haben.
   - Schlüssel: **`MediaConvertRole`** Wert: **`arn:aws:iam::`*`111122223333`*`:role/tutorial-mediaconvert-role`**

     Dieser Wert entspricht ARN der IAM Rolle MediaConvert , für die Sie in Schritt 2 erstellt haben. Stellen Sie sicher, dass Sie diesen ARN Wert durch den tatsächlichen ARN Wert Ihrer IAM Rolle ersetzen.
   - Schlüssel: **`Application`** Wert: **`Batch-Transcoding`**

Dieser Wert ist der Name der Anwendung.

10. Wählen Sie Save (Speichern) aus.
11. (Optional) Wählen Sie im Tab Konfigurationim Abschnitt Allgemeine Konfiguration des linken Navigationsbereichs Bearbeiten aus. Geben Sie im Timeout-Feld **2** m **0** s ein. Wählen Sie dann Save (Speichern) aus.

    Timeout ist die Zeitspanne, die Lambda einer Funktion für einen Aufruf zulässt, bevor diese gestoppt wird. Der Standardwert ist 3 Sekunden. Die Preise basieren auf dem konfigurierten Arbeitsspeicher und der Zeit, für die der Code ausgeführt wird. Weitere Informationen finden Sie unter AWS Lambda Preise.

Schritt 5: Konfigurieren des Amazon S3-Bestands für Ihren S3-Quell-Bucket

Nachdem Sie die Transcodierungs-Lambda-Funktion eingerichtet haben, erstellen Sie einen S3-Batchvorgangsauftrag, um eine Reihe von Videos zu transcodieren. Zuerst benötigen Sie eine Liste von Eingabe-Videoobjekten, für die S3-Batch-Vorgänge die angegebene Transcodierungsaktion ausführen soll. Um eine Liste von Eingabe-Videoobjekten abzurufen, können Sie einen S3-Inventarbericht für Ihren S3-Quell-Bucket erstellen (z. B. *amzn-s3-demo-source-bucket*).

Teilschritte

- Erstellen und Konfigurieren eines Buckets für S3-Inventarberichte für Eingabevideos
- Konfigurieren des Amazon S3-Bestands für Ihren S3-Quell-Bucket
- Überprüfen Sie den Lagerbestandsbericht für Ihren S3-Videoquellen-Bucket

Erstellen und Konfigurieren eines Buckets für S3-Inventarberichte für Eingabevideos

Um einen S3-Bestandslistenbericht zu speichern, der die Objekte des S3-Quell-Buckets auflistet, erstellen Sie einen S3-Inventarziel-Bucket und konfigurieren Sie dann eine Bucket-Richtlinie, damit der Bucket Inventardateien in den S3-Quell-Bucket schreibt.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie Create Bucket (Bucket erstellen) aus.

4. Geben Sie im Feld Bucket Name einen Namen für Ihren Bucket ein (z. B. *amzn-s3-demo-destination-bucket2*).
5. Wählen Sie für den AWS-Region Ort aus AWS-Region, an dem sich der Bucket befinden soll.

   Der Ziel-Bucket für das Inventar muss sich im selben Bucket befinden AWS-Region wie der Quell-Bucket, in dem Sie S3 Inventory einrichten. Der Bestands-Ziel-Bucket kann sich in einem anderem AWS-Konto befinden.
6. Behalten Sie die Einstellungen für öffentlichen Zugriff für diesen Bucket blockieren auf die Standardeinstellungen gestellt (Alle öffentlichen Zugriffe blockieren ist aktiviert).
7. Lassen Sie die restlichen Einstellungen auf die Standardwerte eingestellt.
8. Wählen Sie Bucket erstellen aus.
9. Wählen Sie in der Liste der Buckets den Namen des Buckets, den Sie gearde erstellt haben (z. B. *amzn-s3-demo-destination-bucket2*).
10. Um Amazon S3 die Berechtigung zu erteilen, Daten für die Bestandsberichte in den S3-Basisziel-Bucket zu schreiben, wählen Sie das Tab Berechtigungen aus.
11. Scrollen Sie nach unten zum Abschnitt Bucket-Richtlinie und wählen Sie Bearbeiten aus. Die Seite Bucket-Richtlinie wird geöffnet.
12. Um Berechtigungen für S3-Inventar zu erteilen, fügen Sie im Feld Richtlinie die folgende Bucket-Richtlinie ein.

    Ersetzen Sie die drei Beispielwerte durch die folgenden Werte:

    - Der Name des Buckets, den Sie erstellt haben, um die Bestandsberichte zu speichern (z. B. *amzn-s3-demo-destination-bucket2*).
    - Der Name des Quell-Buckets, in dem die Eingabe-Videos gespeichert werden (z. B. *amzn-s3-demo-source-bucket*).
    - Die AWS-Konto ID, mit der Sie den S3-Videoquellen-Bucket erstellt haben (z. *111122223333* B.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"InventoryAndAnalyticsExamplePolicy",
      "Effect":"Allow",
      "Principal": {"Service": "s3.amazonaws.com"},
```

```
      "Action":"s3:PutObject",
      "Resource":["arn:aws:s3:::amzn-s3-demo-destination-bucket2/*"],
      "Condition": {
          "ArnLike": {
              "aws:SourceArn": "arn:aws:s3:::amzn-s3-demo-source-bucket"
           },
         "StringEquals": {
             "aws:SourceAccount": "111122223333",
             "s3:x-amz-acl": "bucket-owner-full-control"
          }
       }
    }
  ]
}
```

13. Wählen Sie Änderungen speichern.

Konfigurieren des Amazon S3-Bestands für Ihren S3-Quell-Bucket

Sie müssen S3-Inventar für Ihren S3-Videoquellen-Bucket konfigurieren, um eine flache Dateiliste mit Videoobjekten und Metadaten zu generieren. Diese geplanten Bestandsberichte können alle Objekte im Bucket oder Objekte enthalten, die nach einem gemeinsamen Präfix gruppiert sind. In diesem Tutorial enthält der S3-Bestandsbericht alle Video-Objekte in Ihrem S3-Quell-Bucket.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Um einen S3-Inventarbericht der Eingabe-Videos in Ihrem S3-Quell-Bucket zu konfigurieren, wählen Sie in der Liste Buckets zum Beispiel den Namen des S3-Quell-Buckets aus (z. B. *amzn-s3-demo-source-bucket*).
4. Wählen Sie den Tab Management.
5. Blättern Sie nach unten bis zur Konfigurationen für Bestandserfassung und wählen Sie Erstellen der Bestandskonfiguration aus.
6. Geben Sie für Name der Bestandserfassung z. B. einen Namen ein (z. B. **tutorial-inventory-config**).
7. Wählen Sie unter Bestandserfassung Nur aktuelle Version für Objekt-Versionen und lassen Sie andere Einstellungen der Bestandserfassung auf die Standardwerte für dieses Tutorial eingestellt.

8. Wählen Sie im Abschnitt Berichtsdetails für Ziel-Bucket Dieses Konto aus.
9. Wählen Sie für Ziel S3 durchsuchen und wählen Sie den Ziel-Bucket aus, den Sie bereits erstellt haben, um die Lagerbestandsberichte zu speichern (z. B .*amzn-s3-demo-destination-bucket2*). Wählen Sie dann Pfad wählen aus.

   Der Ziel-Bucket für das Inventar muss sich im selben Bucket befinden AWS-Region wie der Quell-Bucket, in dem Sie S3 Inventory einrichten. Der Bestands-Ziel-Bucket kann sich in einem anderem AWS-Konto befinden.

   Unter dem Bucket-Feld Destination (Ziel) wird die Destination bucket permission (Ziel-Bucket-Berechtigung) der Bestands-Ziel-Bucket-Richtlinie hinzugefügt, damit Amazon S3 Daten in diesen Bestands-Ziel-Bucket platzieren kann. Weitere Informationen finden Sie unter Erstellen einer Ziel-Bucket-Richtlinie.
10. Wählen Sie für Häufigkeit Täglich aus.
11. Wählen Sie als Ausgabeformat die Option CSV.
12. Wählen Sie für Status die Option Aktiviert.
13. Wählen Sie im Abschnitt Serverseitige Verschlüsselung Deaktivieren für dieses Tutorial.

    Weitere Informationen erhalten Sie unter Konfigurieren des Bestands mit der S3-Konsole und Erteilen der Berechtigung an Amazon S3 zur Verwendung Ihres vom Kunden verwalteten Schlüssels für die Verschlüsselung.
14. Wählen Sie unter Zusätzliche Felder – optional Größe, Letzte Änderung, und Speicherklasse aus.
15. Wählen Sie Create (Erstellen) aus.

Weitere Informationen finden Sie unter Konfigurieren des Bestands mit der S3-Konsole.

Überprüfen Sie den Lagerbestandsbericht für Ihren S3-Videoquellen-Bucket

Wenn ein Bestandsbericht veröffentlicht wird, werden die Manifestdateien an den S3-Bestands-Ziel-Bucket gesendet.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Video-Quell-Buckets (z. B. *amzn-s3-demo-source-bucket*).

4. Wählen Sie Management (Verwaltung) aus.
5. Um zu sehen, ob Ihr S3-Bestandsbericht für das Erstellen eines S3-Batchvorgänge-Auftrags in Schritt 7 bereit ist, überprüfen Sie unter Bestandskonfigurationen, ob die Schaltfläche Auftrag aus Manifest erstellen aktiviert ist.

   > **Note**
   >
   > Es kann bis zu 48 Stunden dauern, bis der erste Bestandsbericht bereitgestellt wird. Wenn die Schaltfläche Auftrag aus Manifest erstellendeaktiviert ist, wurde der erste Bestandsbericht nicht zugestellt. Warten Sie, bis der erste Bestandsbericht zugestellt wurde und die Schaltfläche Auftrag aus Manifest erstellen aktiviert ist, bevor Sie einen S3-Batchvorgänge-Auftrag in Schritt 7 erstellen.

6. Um einen S3-Bestandsbericht (`manifest.json`) in der die Spalte Ziel zu überprüfen, wählen Sie den Namen des Bestandsziels aus, den Sie bereits zum Speichern von Bestandsberichten erstellt haben (z. B. *`amzn-s3-demo-destination-bucket2`*).
7. Wählen Sie auf der Registerkarte Objekte den vorhandenen Ordner mit dem Namen Ihres S3-Quell-Buckets (z. B. *`amzn-s3-demo-source-bucket`*). Wählen Sie dann den Namen aus, den Sie in Name der Bestandserfassung eingegeben haben, als Sie die Inventarkonfiguration früher erstellt haben (z. B. **`tutorial-inventory-config`**).

   Sie können eine Liste von Ordnern mit den Generierungsdaten der Berichte als Namen sehen.
8. Um den täglichen S3-Bestandsbericht an einem bestimmten Datum zu überprüfen, wählen Sie den Ordner mit dem entsprechenden Namen des Generierungsdatums aus und wählen Sie dann `manifest.json` aus.
9. Um die Details des Bestandsberichts zu einem bestimmten Datum zu überprüfen, wählen Sie auf der Seite manifest.json Herunterladen oder Öffnen aus.

Schritt 6: Eine IAM Rolle für S3 Batch Operations erstellen

Um S3 Batch Operations für die Batch-Transcodierung zu verwenden, müssen Sie zunächst eine IAM Rolle erstellen, um Amazon S3 Berechtigungen zur Durchführung von S3-Batch-Vorgängen zu erteilen.

Teilschritte

- Erstellen Sie eine IAM Richtlinie für S3 Batch Operations

- Erstellen Sie eine IAM S3-Rolle für Batch Operations und fügen Sie Berechtigungsrichtlinien hinzu

Erstellen Sie eine IAM Richtlinie für S3 Batch Operations

Sie müssen eine IAM Richtlinie erstellen, die S3 Batch Operations die Erlaubnis erteilt, das Eingabemanifest zu lesen, die Lambda-Funktion aufzurufen und den Bericht zum Abschluss des S3 Batch Operations-Jobs zu schreiben.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter. https://console.aws.amazon.com/iam/
2. Wählen Sie im linken Navigationsbereich die Option Policies (Richtlinien) aus.
3. Wählen Sie Create Policy (Richtlinie erstellen) aus.
4. Wählen Sie die JSONRegisterkarte.
5. Fügen Sie die folgende JSON Richtlinie in das JSONTextfeld ein.

   Ersetzen Sie in der JSON Richtlinie die vier Beispielwerte durch die folgenden Werte:

   - Der Name des Quell-Buckets, in dem Ihre Eingabe-Videos gespeichert werden (z. B. *amzn-s3-demo-source-bucket*).
   - Der Name des Bestands-Ziel-Buckets, den Sie in Schritt 5 erstellt haben, um `manifest.json`-Dateien (z. B. *amzn-s3-demo-destination-bucket2*) zu speichern.
   - Der Name des Buckets, den Sie in Schritt 1 erstellt haben, um Ausgabe-Mediendateien zu speichern (z. B. *amzn-s3-demo-destination-bucket1*). In diesem Tutorial legen wir Auftragsvervollständigungsberichte in den Ziel-Bucket für Ausgabe-Mediendateien.
   - Die Rolle ARN der Lambda-Funktion, die Sie in Schritt 4 erstellt haben. Gehen Sie wie folgt vor, um die Rolle ARN der Lambda-Funktion zu finden und zu kopieren:
     - Öffnen Sie in einem neuen Browser-Tab die Seite Funktionen der Lambda-Konsole bei https://console.aws.amazon.com/lambda/home#/functions.
     - Wählen Sie in der Liste Funktionen den Namen der Lambda-Funktion aus, die Sie in Schritt 4 (zum Beispiel **`tutorial-lambda-convert`**) erstellt haben.
     - Wählen Sie die Option CopyARN Kopieren aus.

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

```
        {
            "Sid": "S3Get",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-source-bucket/*",
                "arn:aws:s3:::amzn-s3-demo-destination-bucket2/*"
            ]
        },
        {
            "Sid": "S3PutJobCompletionReport",
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket1/*"
        },
        {
            "Sid": "S3BatchOperationsInvokeLambda",
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": [
                "arn:aws:lambda:us-west-2:111122223333:function:tutorial-lambda-
convert"
            ]
        }
    ]
}
```

6. Wählen Sie Next: Tags (Weiter: Tags) aus.
7. Klicken Sie auf Weiter: Prüfen.
8. Geben Sie im Feld Name **`tutorial-s3batch-policy`** ein.
9. Wählen Sie Create Policy (Richtlinie erstellen) aus.

Erstellen Sie eine IAM S3-Rolle für Batch Operations und fügen Sie Berechtigungsrichtlinien hinzu

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.

2. Wählen Sie im linken Navigationsbereich Roles (Rollen) und dann Create Role (Rolle erstellen) aus.
3. Wählen Sie den AWS-Service-Rollentyp und danach den Service S3 aus.
4. Wählen Sie unter Ihren Anwendungsfall auswählen S3-Batch-Vorgänge.
5. Wählen Sie Weiter: Berechtigungen aus.
6. Geben Sie unter Zugriffsrichtlinien anhängen den Namen der IAM Richtlinie, die Sie zuvor erstellt haben (z. B.**`tutorial-s3batch-policy`**), in das Suchfeld ein, um die Liste der Richtlinien zu filtern. Aktivieren Sie das Kontrollkästchen neben dem Namen der Richtlinie (z. B. **`tutorial-s3batch-policy`**).
7. Wählen Sie Next: Tags (Weiter: Tags) aus.
8. Wählen Sie Weiter: Prüfen aus.
9. Geben Sie für Role name (Rollenname) den Namen **`tutorial-s3batch-role`** ein.
10. Wählen Sie Create role (Rolle erstellen) aus.

    Nachdem Sie die IAM Rolle für S3 Batch Operations erstellt haben, wird die folgende Vertrauensrichtlinie automatisch an die Rolle angehängt. Diese Vertrauensrichtlinie ermöglicht es dem Dienstprinzipal von S3 Batch Operations, die IAM Rolle zu übernehmen.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"batchoperations.s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

Schritt 7: Einrichten und Ausführen eines Auftrags für S3-Batchvorgänge

Um einen S3-Batchvorgänge-Auftrag zu erstellen, um die Eingabevideos in Ihrem S3-Quell-Bucket zu verarbeiten, müssen Sie Parameter für diesen bestimmten Auftrag angeben.

> **Note**
>
> Um mit der Erstellung eines S3-Batchvorgänge-Auftrags zu beginnen, stellen Sie sicher, dass die Schaltfläche Auftrag aus Manifest erstellen aktiviert ist. Weitere Informationen finden Sie unter Überprüfen Sie den Lagerbestandsbericht für Ihren S3-Videoquellen-Bucket. Wenn die Schaltfläche Auftrag aus Manifest erstellen deaktiviert ist, wurde der erste Inventarbericht nicht übermittelt und Sie müssen warten, bis die Schaltfläche aktiviert ist. Nachdem Sie Amazon S3-Bestand für Ihren S3-Quell-Bucket in Schritt 5 konfiguriert haben, kann es bis zu 48 Stunden dauern, bis der erste Bestandsbericht übermittelt wird.

Teilschritte

- Erstellen eines S3-Batchvorgangsauftrags
- Führen Sie den S3-Batchvorgänge-Auftrag aus, um Ihre Lambda-Funktion aufzurufen.
- (Optional) Prüfen Sie Ihren Abschlussbericht
- (Optional) Überwachen Sie jeden Lambda-Aufruf in der Lambda-Konsole
- (Optional) Überwachen Sie jeden MediaConvert Video-Transcodierungsauftrag in der Konsole MediaConvert

Erstellen eines S3-Batchvorgangsauftrags

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Klicken Sie im linken Navigationsbereich auf Batchvorgänge.
3. Wählen Sie Job erstellen aus.
4. Wählen Sie für AWS-Region die Region aus, in der Sie Ihren Auftrag erstellen möchten.

   In diesem Tutorial müssen Sie den S3-Batchvorgänge-Auftrag zum Aufrufen einer Lambda-Funktion in derselben Region wie den S3-Videoquellen-Bucket erstellen, in dem sich die im Manifest referenzierten Objekte befinden.
5. Gehen Sie im Abschnitt Manifest wie folgt vor:

   a. Wählen Sie für Manifestformat S3-Inventory-Bericht (manifest.json) aus.

   b. Wählen Sie für Manifest-Objekt S3 durchsuchen, um den Bucket zu finden, den Sie in Schritt 5 zum Speichern von Bestandsberichten (z. B. *amzn-s3-demo-destination-*

bucket2) erstellt haben. Navigieren Sie auf der Seite Manifest-Objekt durch die Objektnamen, bis Sie eine `manifest.json`-Datei für ein bestimmtes Datum finden. Diese Datei listet die Informationen zu allen Videos auf, die Sie Batch-Transcodieren möchten. Wenn Sie die `manifest.json`-Datei gefunden haben, die Sie verwenden möchten, wählen Sie die Optionsschaltfläche daneben. Wählen Sie dann Pfad wählen aus.

c. (Optional) Geben Sie für Versions-ID des Manifestobjekts – optional die Versions-ID des Manifestobjekts ein, wenn Sie eine andere Version als die aktuelle verwenden möchten.

6. Wählen Sie Next (Weiter).
7. Um die Lambda-Funktion zum Transcodieren aller Objekte zu verwenden, die in der ausgewählten `manifest.json`-Datei aufgelistet sind, wählen Sie unter Vorgangstyp AWS Lambda -Funktion aufrufen aus.
8. Führen Sie im Abschnitt Aufrufen einer Lambda-Funktion Folgendes aus:

   a. Klicken Sie auf Wählen Sie aus den Funktionen Ihres Kontos aus.

   b. Wählen Sie für Lambda-Funktion die Lambda-Funktion aus, die Sie in Schritt 4 (zum Beispiel **`tutorial-lambda-convert`**) erstellt haben.

   c. Behalten Sie für die Lambda-Funktionsversion den Standardwert $ LATEST bei.

9. Wählen Sie Weiter. Die Seite Konfigurieren zusätzlicher Optionen wird geöffnet.
10. Behalten Sie im Abschnitt Zusätzliche Optionen die Standardeinstellungen bei.

    Weitere Informationen zu diesen Optionen finden Sie unter Batch-Vorgangsauftrag-Anforderungselemente.

11. Wählen Sie im Abschnitt Abschlussbericht für Pfad zum Ziel des Abschlussberichts S3 durchsuchen aus. Finden Sie den Bucket, den Sie in Schritt 1 für die Ausgabe-Mediendateien erstellte haben (z. B. *amzn-s3-demo-destination-bucket1*). Wählen Sie das Optionsfeld neben dem Namen dieses Buckets aus. Wählen Sie dann Pfad wählen aus.

    Lassen Sie die restlichen Einstellungen des Abschlussberichts auf die Standardwerte eingestellt. Weitere Informationen zu den Einstellungen des Fertigstellungsberichts finden Sie unter Batch-Vorgangsauftrag-Anforderungselemente. Ein Abschlussbericht führt eine Aufzeichnung der Details des Auftrags und der ausgeführten Vorgänge.

12. Wählen Sie im Abschnitt Berechtigungen die Option Aus vorhandenen IAM Rollen auswählen aus. Wählen Sie unter IAMRolle die IAM Rolle für Ihren S3 Batch Operations-Job aus, den Sie in Schritt 6 erstellt haben (z. B.**`tutorial-s3batch-role`**).
13. Wählen Sie Weiter.

14. Überprüfen Sie die Einstellungen auf der Seite Review. Wählen Sie dann Auftrag erstellen aus.

    Nachdem S3 das Manifest Ihres S3-Batchvorgänge-Auftrags gelesen hat, wechselt es den Auftrag in den Zustand Awaiting your confirmation to run (Wartet auf Ihre Bestätigung zur Ausführung). Um Aktualisierungen des Auftragsstatus anzuzeigen, aktualisieren Sie die Seite. Sie können Ihren Job erst ausführen, wenn der Status lautet Wartet auf Ihre Bestätigung zur Ausführung.

Führen Sie den S3-Batchvorgänge-Auftrag aus, um Ihre Lambda-Funktion aufzurufen.

Führen Sie Ihren Batchvorgänge-Auftrag aus, um Ihre Lambda-Funktion für die Videotranscodierung aufzurufen. Wenn Ihr Auftrag fehlschlägt, können Sie den Abschlussbericht überprüfen, um die Ursache zu ermitteln.

So führen Sie den S3-Batchvorgänge-Auftrag aus

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Klicken Sie im linken Navigationsbereich auf Batchvorgänge.
3. Wählen Sie aus der Liste Aufträge die Auftrags-ID des Auftrags in der ersten Zeile aus, bei der es sich um den vorher erstellten S3-Batchvorgänge-Auftrag handelt.
4. Wählen Sie Auftrag ausführen aus.
5. Überprüfen Sie die Auftrags-Parameter erneut, und bestätigen Sie, dass der Wert für Insgesamt im Manifest aufgelistete Objekte mit der Anzahl der Objekte im Manifest identisch ist. Wählen Sie dann Auftrag ausführen.

   Ihre Seite des S3-Batchvorgänge-Auftrags wird geöffnet.
6. Nachdem der Auftrag gestartet wurde, kontrollieren Sie unter Status den Fortschritt Ihres S3-Batchvorgänge-Auftrags, wie z. B. Status, % abgeschlossen, Gesamt erfolgreich (Rate), Gesamt fehlgeschlagen (Rate), Beendigungsdatum, und Grund für die Beendigung.

   Wenn der S3-Batchvorgänge-Auftrag abgeschlossen ist, zeigen Sie die Daten auf Ihrer Auftragaseite an, um zu bestätigen, dass der Auftrag wie erwartet abgeschlossen wurde.

   Wenn mehr als 50 Prozent der Objektvorgänge eines S3-Batchvorgänge-Auftrags fehlschlagen, nachdem mehr als 1000 Vorgänge versucht wurden, schlägt der Auftrag automatisch fehl. Um Ihren Abschlussbericht zu überprüfen, um die Ursache der Fehler zu ermitteln, verwenden Sie das folgenden optionale Verfahren.

(Optional) Prüfen Sie Ihren Abschlussbericht

Sie können Ihren Abschlussbericht verwenden, um festzustellen, welche Objekte fehlgeschlagen sind und welche Ursache der Fehler sind.

Um Ihren Abschlussbericht auf Details zu fehlgeschlagenen Objekten zu überprüfen:

1. Blättern Sie auf der Seite Ihres S3-Batchvorgänge-Auftrags zum Abschnitt Abschlussbericht und wählen Sie den Link unter Ziel des Abschlussberichts aus.

   Die Seite des S3-Ausgabe-Ziel-Buckets wird geöffnet.
2. Wählen Sie auf der Registerkarte Objekte auf den Ordner, dessen Name mit der Auftrags-ID des vorher erstellten S3-Batchvorgänge-Auftrags endet.
3. Klicken Sie auf Ergebnis/se.
4. Aktivieren Sie das Kontrollkästchen neben der `.csv`-Datei.
5. Wählen Sie Öffnen oder Herunterladen, um den Auftragsbericht anzuzeigen.

(Optional) Überwachen Sie jeden Lambda-Aufruf in der Lambda-Konsole

Nachdem der S3-Batchvorgänge-Auftrag gestartet wurde, ruft der Auftrag, die Lambda-Funktion für jedes Eingabe-Video-Objekt auf. S3 schreibt Protokolle jedes Lambda-Aufrufs in Logs. CloudWatch Sie können das Monitoring-Dashboard der Lambda-Konsole verwenden, um Ihre Lambda-Funktion zu überwachen.

1. Öffnen Sie die AWS Lambda Konsole unter. https://console.aws.amazon.com/lambda/
2. Wählen Sie im linken Navigationsbereich die Option Functions (Funktionen) aus.
3. Wählen Sie in der Liste Funktionen den Namen der Lambda-Funktion aus, die Sie in Schritt 4 (zum Beispiel **`tutorial-lambda-convert`**) erstellt haben.
4. Wählen Sie den Tab Überwachung.
5. Finden Sie unter Metriken die Laufzeitmetriken für Ihre Lambda-Funktion.
6. Sehen Sie sich unter Logs die Protokolldaten für jeden Lambda-Aufruf über CloudWatch Logs Insights an.

> **Note**
>
> Wenn Sie S3-Batch-Vorgänge mit einer Lambda-Funktion verwenden, wird die Lambda-Funktion für jedes Objekt aufgerufen. Wenn Ihr S3-Batchvorgänge-Auftrag groß ist, kann

er mehrere Lambda-Funktionen gleichzeitig aufrufen, was zu einer Spitze der Lambda-Gleichzeitigkeit führt.
Jedes AWS-Konto hat eine Lambda-Parallelitätsquote pro Region. Weitere Informationen finden Sie unter Funktionsskalierung AWS Lambda im AWS Lambda -Entwicklerhandbuch. Eine bewährte Methode für die Verwendung von Lambda-Funktionen mit S3-Batchvorgängen besteht darin, eine Gleichzeitigkeitsgrenze für die Lambda-Funktion selbst festzulegen. Ein Parallelitäts-Limit aufzustellen verhindert, dass Ihr Auftrag den größten Teil Ihrer Lambda-Parallelität verbraucht und möglicherweise andere Funktionen in Ihrem Konto drosselt. Weitere Informationen finden Sie unter Verwalten reservierter Parallelität von Lambda im AWS Lambda -Entwicklerhandbuch.

(Optional) Überwachen Sie jeden MediaConvert Video-Transcodierungsauftrag in der Konsole MediaConvert

Ein MediaConvert Job erledigt die Transcodierung einer Mediendatei. Wenn Ihr S3 Batch Operations-Job Ihre Lambda-Funktion für jedes Video aufruft, erstellt jeder Lambda-Funktionsaufruf einen MediaConvert Transcodierungsauftrag für jedes Eingabevideo.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Konsole unter. MediaConvert https://console.aws.amazon.com/mediaconvert/
2. Wenn die MediaConvert Einführungsseite angezeigt wird, wählen Sie Erste Schritte.
3. Zeige Sie aus der Liste Aufträge jede Zeile an, um die Transcodierungsaufgabe für jedes Eingabe-Video zu überwachen.
4. Geben Sie die Zeile eines Auftrags an, den Sie überprüfen möchten, und wählen Sie den Auftrags-ID-Link, um die Seite Auftrags-Details zu öffnen.
5. Wählen Sie auf der Seite mit der Jobübersicht unter Ausgaben den Link für die AusgabeHLS,MP4, oder Miniaturansichten aus, je nachdem, was von Ihrem Browser unterstützt wird, um zum S3-Ziel-Bucket für die Ausgabemediendateien zu gelangen.
6. Wählen Sie im entsprechenden Ordner (HLSMP4, oder Miniaturansichten) Ihres Ziel-Buckets für die S3-Ausgabe den Namen des Objekts der Ausgabemediendatei aus.

   Die Detailseite des Objekts wird geöffnet.
7. Wählen Sie auf der Detailseite des Objekts unter Objektübersicht den Link unter Objekt aus, URL um die transkodierte Ausgabemediendatei anzusehen.

Schritt 8: Überprüfen Sie die Ausgabe-Mediendateien aus Ihrem S3-Ziel-Bucket

So überprüfen Sie die Ausgabe-Mediendateien aus Ihrem S3-Ziel-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie aus der Liste Buckets den Namen des S3-Ziel-Buckets für Ausgabe-Mediendateien aus, die Sie im Abschnitt Schritt 1 (zum Beispiel *amzn-s3-demo-destination-bucket1*) erstellt haben.
4. Im Tab Objekte enthält jedes Eingabe-Video einen Ordner mit dem Namen des Eingabe-Videos. Jeder Ordner enthält die transcodierten Ausgabe-Mediendateien für ein Eingabe-Video.

   Gehen Sie wie folgt vor, um die Ausgabe-Mediendateien für ein Eingabe-Video zu überprüfen:

   a. Wählen Sie den Ordner mit dem Namen des Eingabe-Videos, das Sie überprüfen möchten.
   b. Wählen Sie den Ordner Standard/ aus.
   c. Wählen Sie den Ordner für ein transkodiertes Format (HLS,MP4, oder Miniaturansichten in diesem Tutorial).
   d. Wählen Sie den Namen der Ausgabemediendatei aus.
   e. Um die transkodierte Datei anzusehen, klicken Sie auf der Detailseite des Objekts auf den Link unter Objekt. URL

      Ausgabemediendateien in diesem HLS Format sind in kurze Segmente aufgeteilt. Um diese Videos abzuspielen, betten Sie das Objekt URL der `.m3u8` Datei in einen kompatiblen Player ein.

Schritt 9: Bereinigen

Wenn Sie Videos mit S3 Batch Operations, Lambda und MediaConvert nur zu Lernzwecken transkodiert haben, löschen Sie die AWS Ressourcen, die Sie zugewiesen haben, sodass keine Gebühren mehr anfallen.

Teilschritte

- Löschen der S3-Bestandskonfiguration für Ihren S3-Quell-Bucket
- Löschen Sie die Lambda-Funktion

- Löschen Sie die CloudWatch Protokollgruppe
- Löschen Sie die IAM Rollen zusammen mit den Inline-Richtlinien für die IAM Rollen
- Löschen Sie die vom Kunden verwaltete Richtlinie IAM
- Leeren Sie die S3-Buckets
- Löschen der S3-Buckets

Löschen der S3-Bestandskonfiguration für Ihren S3-Quell-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen Ihres Quell-Buckets (z. B. *amzn-s3-demo-source-bucket*).
4. Wählen Sie den Tab Management.
5. Wählen Sie im Abschnitt Bestandskonfigurationen die Optionsschaltfläche neben der Bestandskonfiguration aus, die Sie in Schritt 5 (zum Beispiel **tutorial-inventory-config**) erstellt haben.
6. Wählen Sie Löschen und dann Bestätigen aus.

Löschen Sie die Lambda-Funktion

1. Öffnen Sie die AWS Lambda Konsole unter https://console.aws.amazon.com/lambda/.
2. Wählen Sie im linken Navigationsbereich die Option Functions (Funktionen) aus.
3. Aktivieren Sie das Kontrollkästchen neben der Funktion, die Sie im Abschnitt Schritt 4 (zum Beispiel **tutorial-lambda-convert**) erstellt haben.
4. Wählen Sie Aktionen und anschließend Löschen aus.
5. Wählen Sie im Bestätigungsdialogfeld Delete (Löschen) die Option Delete (Löschen) aus.

Löschen Sie die CloudWatch Protokollgruppe

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im linken Navigationsbereich Protokolle und dann Protokollgruppen aus.

3. Aktivieren Sie das Kontrollkästchen neben der Protokollgruppe, deren Name mit der Lambda-Funktion endet, die Sie im Abschnitt Schritt 4 (zum Beispiel **`tutorial-lambda-convert`**) erstellt haben.
4. Wählen Sie Actions (Aktionen) und dann Delete log group(s) (Protokollgruppe(n) löschen) aus.
5. Wählen Sie im Dialogfeld Delete log group(s) (Protokollgruppe(n) löschen) die Option Delete (Löschen) aus.

Löschen Sie die IAM Rollen zusammen mit den Inline-Richtlinien für die IAM Rollen

Gehen Sie wie folgt vor, um die IAM Rollen zu löschen, die Sie in Schritt 2, Schritt 3 und Schritt 6 erstellt haben:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Roles (Rollen), und aktivieren Sie dann die Kontrollkästchen neben den Rollennamen, die Sie löschen möchten.
3. Wählen Sie oben auf der Seite Delete (Löschen) aus.
4. Geben Sie in das Bestätigungsfeld die erforderliche Antwort basierend auf der Eingabeaufforderung in das Texteingabefeld ein, und wählen Sie dann Löschen aus.

Löschen Sie die vom Kunden verwaltete Richtlinie IAM

Gehen Sie wie folgt vor, um die vom Kunden verwaltete IAM Richtlinie zu löschen, die Sie in Schritt 6 erstellt haben:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im linken Navigationsbereich Richtlinien aus.
3. Wählen Sie das Optionsfeld neben der Richtlinie aus, die Sie in Schritt 6 (zum Beispiel **`tutorial-s3batch-policy`**) erstellt haben. Sie können über das Suchfeld die Liste der Richtlinien filtern.
4. Wählen Sie Aktionen und anschließend Löschen.
5. Bestätigen Sie, dass Sie diese Richtlinie löschen möchten, indem Sie den Namen in das angezeigte Textfeld eingeben und dann Delete (Löschen) wählen.

Leeren Sie die S3-Buckets

Um die S3-Buckets zu leeren, die Sie in Voraussetzungen, Schritt 1 und Schritt 5 erstellt haben, tun Sie Folgendes:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets das Optionsfeld neben dem Namen des Buckets aus, den Sie leeren möchten. Wählen Sie anschließend Empty (Leeren) aus.
4. Bestätigen Sie auf der Seite Empty bucket (Bucket leeren), dass Sie den Bucket leeren möchten, indem Sie **`permanently delete`** in das Textfeld eingeben und dann Empty (Leeren) auswählen.

Löschen der S3-Buckets

Um die S3-Buckets zu löschen, die Sie in Voraussetzungen, Schritt 1 und Schritt 5 erstellt haben, tun Sie Folgendes:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets das Optionsfeld neben dem Namen des Buckets aus, den Sie löschen möchten.
4. Wählen Sie Löschen.
5. Bestätigen Sie auf der Seite Delete bucket (Bucket löschen), dass Sie den Bucket löschen möchten. Geben Sie dazu den Bucket-Namen in das Textfeld ein und wählen Sie Delete bucket (Bucket löschen).

Nächste Schritte

Nachdem Sie dieses Tutorial abgeschlossen haben, können Sie weitere relevante Anwendungsfälle untersuchen:

- Sie können Amazon verwenden CloudFront , um die transkodierten Mediendateien an Zuschauer auf der ganzen Welt zu streamen. Weitere Informationen finden Sie unter Tutorial: Hosten von On-Demand-Streaming-Videos mit Amazon S3 CloudFront, Amazon und Amazon Route 53.

- Sie können Videos in dem Moment transcodieren, indem Sie sie in den S3-Quell-Bucket hochladen. Dazu können Sie einen Amazon S3 S3-Event-Trigger konfigurieren, der automatisch die Lambda-Funktion aufruft, mit der neue Objekte in S3 transkodiert werden. MediaConvert Weitere Informationen finden Sie unter Tutorial: Verwenden eines Amazon S3-Auslösers zum Aufrufen einer Lambda-Funktion im AWS Lambda -Entwicklerhandbuch.

Problembehandlung bei Batch-Vorgängen

In den folgenden Themen sind häufig auftretende Fehler aufgeführt, um Ihnen bei der Behebung von Problemen zu helfen, die bei der Arbeit mit Amazon S3 Batch Operations auftreten können.

Informationen zur Behebung von Problemen mit S3 Batch Replication finden Sie unterthe section called “Fehler bei der Batchreplikation”.

Häufige Fehler

- Der Jobbericht wird nicht zugestellt, wenn ein Berechtigungsproblem vorliegt oder ein S3 Object Lock-Aufbewahrungsmodus aktiviert ist
- Bei Batch-Vorgängen scheitern Objekte mit dem Fehler 400 InvalidRequest: Aufgabe ist aufgrund fehlender Objekte fehlgeschlagen VersionId
- Fehler bei Auftragserstellung mit aktivierter Auftrags-Tag-Option
- Zugriff zum Lesen des Manifests verweigert

Der Jobbericht wird nicht zugestellt, wenn ein Berechtigungsproblem vorliegt oder ein S3 Object Lock-Aufbewahrungsmodus aktiviert ist

Der folgende Fehler tritt auf, wenn die erforderlichen Berechtigungen fehlen oder ein Object Lock-Aufbewahrungsmodus (entweder Governance-Modus oder Compliance-Modus) für den Ziel-Bucket aktiviert ist.

Fehler: Fehlerursachen. Der Auftragsbericht konnte nicht in Ihren Berichts-Bucket geschrieben werden. Überprüfen Sie Ihre Berechtigungen.

Die Rollen- und Vertrauensrichtlinie AWS Identity and Access Management (IAM) müssen so konfiguriert werden, dass Batch Operations `s3:PutObject` Zugriff auf PUT Objekte in dem Bucket erhält, in den der Bericht übermittelt wird. Wenn diese erforderlichen Berechtigungen fehlen, schlägt die Zustellung des Jobberichts fehl.

Wenn ein Aufbewahrungsmodus aktiviert ist, ist der Bucket geschützt write-once-read-many (WORM). Object Lock mit aktiviertem Aufbewahrungsmodus für den Ziel-Bucket wird nicht unterstützt, sodass Versuche zur Übermittlung von Berichten zum Abschluss von Aufträgen fehlschlagen. Wählen Sie zum Beheben dieses Problems einen Ziel-Bucket für Ihre Abschlussberichte zu Aufträgen aus, für den kein Aufbewahrungsmodus der Objektsperre aktiviert ist.

Bei Batch-Vorgängen scheitern Objekte mit dem Fehler 400 InvalidRequest: Aufgabe ist aufgrund fehlender Objekte fehlgeschlagen VersionId

Der folgende Beispielfehler tritt auf, wenn ein Batch-Operations-Auftrag Aktionen für Objekte in einem versionsfähigen Bucket ausführt und im Manifest ein Objekt mit leerem Versions-ID-Feld feststellt.

Fehler: *`bucket_name,prefix/file_name`*, fehlgeschlagen,400,, Die Aufgabe ist fehlgeschlagenInvalidRequest, weil sie fehlt VersionId

Dieser Fehler tritt auf, weil das Versions-ID-Feld im Manifest eine leere Zeichenfolge und nicht die Literalzeichenfolge `null` ist.

Batch Operations schlägt für diese(s) spezielle(n) Objekt(e) fehl, aber nicht für den gesamten Auftrag. Dieses Problem tritt auf, wenn das Manifestformat so konfiguriert ist, dass die Version IDs während des Vorgangs verwendet wird. Bei Aufträgen ohne Versionierung tritt dieses Problem nicht auf, da sie nur mit der neuesten Version jedes Objekts ausgeführt werden und die Version IDs im Manifest ignoriert wird.

Um dieses Problem zu beheben, konvertieren Sie die leere Version IDs in Zeichenketten. `null` Weitere Informationen finden Sie unter the section called “Konvertieren leerer Versions-ID-Zeichenfolgen in Null-Zeichenfolgen”.

Fehler bei Auftragserstellung mit aktivierter Auftrags-Tag-Option

Ohne die Berechtigung `s3:PutJobTagging` führt das Erstellen von Batch-Operations-Aufträgen mit aktivierter Auftrags-Tag-Option zum Fehler `403 access denied`.

Um Batch Operations-Jobs mit aktivierter Job-Tag-Option zu erstellen, muss der Benutzer AWS Identity and Access Management (IAM), der den Batch Operations-Job erstellt, zusätzlich zur `s3:PutJobTagging` Berechtigung auch über die `s3:CreateJob` entsprechende Berechtigung verfügen.

Weitere Informationen zu den erforderlichen Berechtigungen für Batch Operations finden Sie unter the section called “Gewähren von Berechtigungen”.

Zugriff zum Lesen des Manifests verweigert

Wenn Sie versuchen, einen Batch-Operations-Auftrag zu erstellen und Batch Operations die Manifestdatei nicht lesen kann, können die folgenden Fehler auftreten.

AWS CLI

Grund des Fehlers Das Manifest zu lesen ist verboten: AccessDenied

Amazon S3-Konsole

Warnung: Die des Manifest-Objekts konnten nicht abgerufen ETag werden. Geben Sie ein anderes Objekt an, um fortzufahren.

Führen Sie die folgenden Schritte aus, um dieses Problem zu beheben:

- Stellen Sie sicher, dass die IAM Rolle für die AWS-Konto , mit der Sie den Batch Operations-Job erstellt haben, über die `s3:GetObject` entsprechende Berechtigung verfügt. Die IAM Rolle des Kontos muss über die `s3:GetObject` Berechtigung verfügen, Batch Operations das Lesen der Manifestdatei zu gestatten.

  Weitere Informationen zu den erforderlichen Berechtigungen für Batch Operations finden Sie unter the section called “Gewähren von Berechtigungen”.
- Überprüfen Sie die Metadaten der Manifestobjekte auf etwaige Zugriffskonflikte mit S3 Object Ownership. Informationen zu S3 Object Ownership finden Sie unter the section called “Steuern der Objekteigentümerschaft”.
- Prüfen Sie, ob AWS Key Management Service (AWS KMS) -Schlüssel zum Verschlüsseln der Manifestdatei verwendet werden.

  Batch Operations unterstützt CSVInventarberichte, die AWS KMS-verschlüsselt sind. Batch Operations unterstützt jedoch keine CSVManifestdateien, die AWS KMS-verschlüsselt sind. Weitere Informationen erhalten Sie unter Konfigurieren von Amazon S3 Inventory und Angeben eines Manifests.

Abfragen vorhandener Daten mit Amazon S3 Select

> **Important**
>
> Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. Weitere Informationen

Mit Amazon S3 Select können Sie Structured Query Language (SQL) -Anweisungen verwenden, um den Inhalt eines Amazon S3 S3-Objekts zu filtern und nur die Teilmenge der Daten abzurufen, die Sie benötigen. Wenn Sie die Daten mit Amazon S3 Select filtern, wird die von Amazon S3 übertragene Datenmenge reduziert, wodurch sich die Kosten und Latenzzeiten für diesen Datenabruf verringern.

Mit Amazon S3 Select können Sie jeweils nur ein Objekt abfragen. Es funktioniert mit einem Objekt, das in CSVJSON, oder gespeichert ist Apache Parquet . Es funktioniert auch mit einem Objekt, das mit GZIP oder komprimiert ist BZIP2 (nur für CSV JSON Und-Objekte), und einem serverseitig verschlüsselten Objekt. Sie können das Format der Ergebnisse entweder als CSV oder angeben und festlegenJSON, wie die Datensätze im Ergebnis getrennt werden.

In der Anfrage übergeben Sie SQL Ausdrücke an Amazon S3. Amazon S3 Select unterstützt eine Teilmenge von. SQL Weitere Informationen zu den SQL Elementen, die von Amazon S3 Select unterstützt werden, finden Sie unterSQLReferenz für Amazon S3 Select.

Sie können SQL Abfragen mithilfe der Amazon S3 S3-Konsole, der AWS Command Line Interface (AWS CLI), der `SelectObjectContent` REST API Operation oder der ausführen AWS SDKs.

> **Note**
>
> Bei der Amazon S3-Konsole ist die zurückgegebene Datenmenge auf 40 MB begrenzt. Um weitere Daten abzurufen, verwenden Sie das AWS CLI oder dasAPI.

Voraussetzungen und Einschränkungen

Für die Verwendung von Amazon S3 Select müssen folgende Voraussetzungen erfüllt sein:

- Sie haben die Berechtigung `s3:GetObject` für das abzufragende Objekt.

- Wenn das Objekt, das Sie abfragen, mit serverseitiger Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) verschlüsselt ist, müssen Sie den Verschlüsselungsschlüssel verwenden `https` und in der Anfrage angeben.

Für die Verwendung von Amazon S3 Select gelten die folgenden Einschränkungen:

- S3 Select kann nur ein Objekt pro Anfrage abfragen.
- Die maximale Länge eines SQL Ausdrucks beträgt 256 KB.
- Die Maximallänge eines Datensatzes in der Eingabe oder im Ergebnis liegt bei 1 MB.
- Amazon S3 Select kann nur verschachtelte Daten mithilfe des JSON Ausgabeformats ausgeben.
- Sie können kein Objekt abfragen, das in den Speicherklassen S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive oder Reduced Redundancy Storage (RRS) gespeichert ist. Sie können auch kein Objekt abfragen, das auf der Stufe S3 Intelligent-Tiering Archive Access oder der Stufe S3 Intelligent-Tiering Deep Archive Access gespeichert ist. Weitere Informationen über Speicherklassen finden Sie unter Amazon S3 S3-Speicherklassen verstehen und verwalten.

Zusätzliche Einschränkungen gelten bei der Verwendung von Amazon S3 Select mit einem Parquet Objekt:

- Amazon S3 Select unterstützt nur die spaltenförmige Komprimierung mit oder GZIP Snappy. Amazon S3 Select unterstützt keine Komprimierung des gesamten Objekts für ein Parquet Objekt.
- Amazon S3 Select unterstützt nicht Parquet Ausgabe. Sie müssen das Ausgabeformat als CSV oder angebenJSON.
- Die Maximalgröße für unkomprimierte Zeilengruppen beträgt 512 MB.
- Sie müssen die im Schema des Objekts angegebenen Datentypen verwenden.
- Select-Anfragen für ein wiederholtes Feld geben ausschließlich den letzten Wert zurück.

Erstellen einer Anforderung

Beim Erstellen einer Anforderung geben Sie die Details des abzufragenden Objekts mithilfe eines `InputSerialization`-Objekts an. Die Details zur Rückgabe der Ergebnisse geben Sie mit einem `OutputSerialization`-Objekt an. Sie geben auch den SQL Ausdruck an, den Amazon S3 verwendet, um die Anfrage zu filtern.

Weitere Informationen zur Erstellung einer Amazon S3 Select-Anfrage finden Sie unter SelectObjectContentin der Amazon Simple Storage Service API Reference. In den folgenden Abschnitten finden Sie auch eines der SDK Codebeispiele.

Anforderungen mittels Scanbereichen

Mit Amazon S3 Select können Sie eine Teilmenge eines Objekts scannen, indem Sie einen abzufragenden Byte-Bereich angeben. Mit dieser Fähigkeit können Sie das gesamte Objekt parallel scannen, indem Sie die Arbeit in separate Amazon S3 Select-Anforderungen für eine Reihe von sich nicht überlappenden Scanbereichen aufteilen.

Scanbereiche müssen nicht an Datensatzgrenzen ausgerichtet werden. Eine Scanbereichsanforderung von Amazon S3 Select wird über den angegebenen Byte-Bereich ausgeführt. Ein Datensatz, der im angegebenen Scanbereich beginnt, aber darüber hinaus reicht, wird von der Abfrage bearbeitet. Im Folgenden wird beispielsweise ein Amazon S3 S3-Objekt gezeigt, das eine Reihe von Datensätzen in einem zeilengetrennten Format CSV enthält:

```
A,B
C,D
D,E
E,F
G,H
I,J
```

Gehen wir einmal davon aus, dass Sie den Amazon-S3-Select-Parameter `ScanRange` verwenden, bei (Byte) 1 beginnen und bei (Byte) 4 enden. Der Scanbereich würde also bei „`,`“ beginnen und der Scan würde bis zum Ende des Datensatzes fortgesetzt, der bei „`C`“ beginnt. Ihre Scanbereichsanfrage gibt das Ergebnis „`C, D`“ zurück, da dies das Ende des Datensatzes ist.

Amazon S3 Select Scan Range fordert Unterstützung an Parquet, CSV (ohne Trennzeichen in Anführungszeichen) oder JSON Objekte (nur im `LINES` Modus). CSVund JSON Objekte müssen unkomprimiert sein. Wenn für zeilenbasierte JSON Objekte CSV und Objekte ein Scanbereich als Teil der Amazon S3 Select-Anfrage angegeben wird, werden alle Datensätze verarbeitet, die innerhalb des Scanbereichs beginnen. Wählen Sie in der &Snowconsole; Ihren Auftrag aus der Tabelle. Parquet Bei Objekten werden alle Zeilengruppen verarbeitet, die innerhalb des angeforderten Scanbereichs beginnen.

Amazon S3 Select-Scanbereichsanforderungen können mit Amazon S3 API und verwendet AWS SDKs werden. AWS CLI Sie können den `ScanRange`-Parameter in der Amazon S3 Select-Anfrage

als diese Funktion verwenden. Weitere Informationen finden Sie unter SelectObjectContentin der Amazon Simple Storage Service API Reference.

Fehler

Amazon S3 Select gibt einen Fehlercode und eine zugehörige Fehlermeldung zurück, wenn beim Versuch, eine Abfrage auszuführen, ein Problem auftritt. Eine Liste der Fehlercodes und Beschreibungen finden Sie im Abschnitt Liste der Fehlercodes für SELECT Objektinhalte auf der Seite Error Responses in der Amazon Simple Storage Service API Reference.

Weitere Informationen zu Amazon S3 Select finden Sie in den folgenden Themen:

Themen

- Beispiele für die Verwendung von Amazon S3 Select für ein Objekt
- SQLReferenz für Amazon S3 Select

Beispiele für die Verwendung von Amazon S3 Select für ein Objekt

> **Important**
>
> Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. Weitere Informationen

Sie können S3 Select verwenden, um Inhalte aus einem Objekt auszuwählen, indem Sie die Amazon S3 S3-Konsole, die RESTAPI, und die verwenden AWS SDKs.

Weitere Informationen zu den unterstützten SQL Funktionen für S3 Select finden Sie unterSQLFunktionen.

Verwenden der S3-Konsole

Inhalte aus einem Objekt in der Amazon S3-Konsole auswählen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie den Bucket aus, der das Objekt enthält, aus dem Sie Inhalte auswählen möchten, und wählen Sie dann den Namen des Objekts aus.

4. Wählen Sie Objektaktionen und anschließend Abfragen mit S3 Select aus.
5. Konfigurieren Sie die Eingabeeinstellungen basierend auf dem Format Ihrer Eingabedaten.
6. Konfigurieren Sie die Ausgabeeinstellungen basierend auf dem Format der Ausgabe, die Sie empfangen möchten.
7. Um Datensätze aus dem ausgewählten Objekt zu extrahieren, geben Sie unter SQLAbfrage den SELECT SQL Befehle. Weitere Informationen zum Schreiben von SQL Befehlen finden Sie unterSQLReferenz für Amazon S3 Select.
8. Nachdem Sie SQL Abfragen eingegeben haben, wählen Sie SQLAbfrage ausführen. Anschließend können Sie unter Abfrageergebnisse die Ergebnisse Ihrer SQL Abfragen sehen.

Verwenden Sie den REST API

Sie können den verwenden AWS SDKs, um Inhalt aus einem Objekt auszuwählen. Wenn Ihre Anwendung dies jedoch erfordert, können Sie REST Anfragen direkt senden. Weitere Informationen zum Anfrage- und Antwortformat finden Sie unter SelectObjectContent.

Unter Verwendung des AWS SDKs

Sie können Amazon S3 Select verwenden, um mithilfe der `selectObjectContent` Methode einen Teil des Inhalts eines Objekts auszuwählen. Wenn diese Methode erfolgreich ist, gibt sie die Ergebnisse des SQL Ausdrucks zurück.

Java

Der folgende Java-Code gibt den Wert der ersten Spalte für jeden Datensatz zurück, der in einem Objekt gespeichert ist, das Daten enthält, die im CSV Format gespeichert sind. Im Beispiel müssen auch `Progress`- und `Stats`-Meldungen zurückgegeben werden. Sie müssen einen gültigen Bucket-Namen und ein Objekt angeben, das Daten im CSV Format enthält.

Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
package com.amazonaws;

import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CSVInput;
import com.amazonaws.services.s3.model.CSVOutput;
import com.amazonaws.services.s3.model.CompressionType;
```

```
import com.amazonaws.services.s3.model.ExpressionType;
import com.amazonaws.services.s3.model.InputSerialization;
import com.amazonaws.services.s3.model.OutputSerialization;
import com.amazonaws.services.s3.model.SelectObjectContentEvent;
import com.amazonaws.services.s3.model.SelectObjectContentEventVisitor;
import com.amazonaws.services.s3.model.SelectObjectContentRequest;
import com.amazonaws.services.s3.model.SelectObjectContentResult;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.concurrent.atomic.AtomicBoolean;

import static com.amazonaws.util.IOUtils.copy;

/**
 * This example shows how to query data from S3Select and consume the response in
 the form of an
 * InputStream of records and write it to a file.
 */

public class RecordInputStreamExample {

    private static final String BUCKET_NAME = "${my-s3-bucket}";
    private static final String CSV_OBJECT_KEY = "${my-csv-object-key}";
    private static final String S3_SELECT_RESULTS_PATH = "${my-s3-select-results-
path}";
    private static final String QUERY = "select s._1 from S3Object s";

    public static void main(String[] args) throws Exception {
        final AmazonS3 s3Client = AmazonS3ClientBuilder.defaultClient();

        SelectObjectContentRequest request = generateBaseCSVRequest(BUCKET_NAME,
 CSV_OBJECT_KEY, QUERY);
        final AtomicBoolean isResultComplete = new AtomicBoolean(false);

        try (OutputStream fileOutputStream = new FileOutputStream(new File
 (S3_SELECT_RESULTS_PATH));
             SelectObjectContentResult result =
 s3Client.selectObjectContent(request)) {
            InputStream resultInputStream =
 result.getPayload().getRecordsInputStream(
                    new SelectObjectContentEventVisitor() {
```

```
                        @Override
                        public void visit(SelectObjectContentEvent.StatsEvent event)
                        {
                            System.out.println(
                                    "Received Stats, Bytes Scanned: " +
event.getDetails().getBytesScanned()
                                            +  " Bytes Processed: " +
event.getDetails().getBytesProcessed());
                        }

                        /*
                         * An End Event informs that the request has finished
successfully.
                         */
                        @Override
                        public void visit(SelectObjectContentEvent.EndEvent event)
                        {
                            isResultComplete.set(true);
                            System.out.println("Received End Event. Result is
complete.");
                        }
                    }
            );

            copy(resultInputStream, fileOutputStream);
        }

        /*
         * The End Event indicates all matching records have been transmitted.
         * If the End Event is not received, the results may be incomplete.
         */
        if (!isResultComplete.get()) {
            throw new Exception("S3 Select request was incomplete as End Event was
not received.");
        }
    }

    private static SelectObjectContentRequest generateBaseCSVRequest(String bucket,
String key, String query) {
        SelectObjectContentRequest request = new SelectObjectContentRequest();
        request.setBucketName(bucket);
        request.setKey(key);
        request.setExpression(query);
        request.setExpressionType(ExpressionType.SQL);
```

```
        InputSerialization inputSerialization = new InputSerialization();
        inputSerialization.setCsv(new CSVInput());
        inputSerialization.setCompressionType(CompressionType.NONE);
        request.setInputSerialization(inputSerialization);

        OutputSerialization outputSerialization = new OutputSerialization();
        outputSerialization.setCsv(new CSVOutput());
        request.setOutputSerialization(outputSerialization);

        return request;
    }
}
```

JavaScript

Ein JavaScript Beispiel, bei dem der `SelectObjectContent` API Vorgang AWS SDK for JavaScript with the S3 verwendet wird, um Datensätze JSON und CSV Dateien auszuwählen, die in Amazon S3 gespeichert sind, finden Sie im Blogbeitrag Einführung in die Unterstützung von Amazon S3 Select in der AWS SDK for JavaScript.

Python

Ein Python-Beispiel für die Verwendung von SQL Abfragen zum Durchsuchen von Daten, die mithilfe von S3 Select als Datei mit kommagetrennten Werten (CSV) in Amazon S3 geladen wurden, finden Sie im Blogbeitrag Abfragen von Daten ohne Server oder Datenbanken mit Amazon S3 Select.

SQLReferenz für Amazon S3 Select

> **Important**
>
> Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. Weitere Informationen

Diese Referenz enthält eine Beschreibung der Elemente der strukturierten Abfragesprache (SQL), die von Amazon S3 Select unterstützt werden.

Themen

- SELECT command
- Datentypen
- Operatoren
- Reservierte Schlüsselwörter
- SQLFunktionen

SELECT command

> **Important**
>
> Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. Weitere Informationen

Amazon S3 Select unterstützt nur den `SELECT` SQL Befehl. Die folgenden ANSI Standardklauseln werden unterstützt für`SELECT`:

- `SELECT` table
- `FROM`-Klausel
- `WHERE`-Klausel
- `LIMIT`-Klausel

> **Note**
>
> Amazon S3 Select-Abfragen unterstützen derzeit keine Unterabfragen oder Joins.

SELECT auflisten

Die `SELECT`-Liste nennt die Spalten, Funktionen und Ausdrücke, die die Abfrage zurückgeben soll. Der Liste stellt die Ausgabe der Abfrage dar.

```
SELECT *
SELECT projection1 AS column_alias_1, projection2 AS column_alias_2
```

Bei der ersten SELECT-Form mit dem * (Sternchen) werden alle Zeilen zurückgegeben, die die Bedingung der WHERE-Klausel erfüllen. Bei der zweiten SELECT-Form wird für jede Spalte eine Zeile mit benutzerdefinierten skalaren Ausgabeausdrücken ***projection1*** und ***projection2*** erstellt.

FROM Satz

Amazon S3 Select unterstützt die folgenden Formen der FROM-Klausel:

```
FROM table_name
FROM table_name alias
FROM table_name AS alias
```

In jeder Form der FROM-Klausel steht table_name für das S3Object, das abgefragt wird. Benutzer traditioneller relationaler Datenbanken können sich dies als Datenbankschema mit mehreren Ansichten einer Tabelle vorstellen.

Gemäß dem Standard SQL erstellt die FROM Klausel Zeilen, die in der WHERE Klausel gefiltert und in die SELECT Liste projiziert werden.

Für JSON Objekte, die in Amazon S3 Select gespeichert sind, können Sie auch die folgenden Formen der FROM Klausel verwenden:

```
FROM S3Object[*].path
FROM S3Object[*].path alias
FROM S3Object[*].path AS alias
```

Mit dieser Form der FROM Klausel können Sie aus Arrays oder Objekten innerhalb eines JSON Objekts auswählen. Sie können „path“ unter Verwendung einer der folgenden Formen angeben:

- Nach Name (in einem Objekt): .*name* oder ['*name*']
- Nach Index (in einem Array): [*index*]
- Nach Platzhalterzeichen (in einem Objekt): .*
- Nach Platzhalterzeichen (in einem Array): [*]

> **Note**
>
> - Diese Form der FROM Klausel funktioniert nur mit JSON Objekten.

- Platzhalterzeichen übermitteln stets mindestens einen Datensatz. Wenn es keinen übereinstimmenden Datensatz gibt, übermittelt Amazon S3 Select den Wert `MISSING`. Während der Ausgabeserialisierung (nach Abschluss der Abfrage) ersetzt Amazon S3 Select `MISSING`-Werte durch leere Datensätze.
- Aggregierte Funktionen (`AVG`, `COUNT`, `MAX`, `MIN` und `SUM`) überspringen `MISSING`-Werte.
- Wenn Sie bei Verwendung eines Platzhalterzeichens keinen Alias angeben, können Sie auf die Zeile verweisen, die das letzte Element im Pfad verwendet. Sie könnten beispielsweise alle Preise aus einer Liste von Büchern unter Verwendung der Abfrage `SELECT price FROM S3Object[*].books[*].price` auswählen. Wenn der Pfad mit einem Platzhalterzeichen anstelle eines Namens endet, können Sie den Wert `_1` verwenden, um auf die Zeile zu verweisen. Beispielsweise könnten Sie anstelle von `SELECT price FROM S3Object[*].books[*].price` die Abfrage `SELECT _1.price FROM S3Object[*].books[*]` verwenden.
- Amazon S3 Select behandelt ein JSON Dokument immer als ein Array von Werten auf Stammebene. Selbst wenn das JSON Objekt, das Sie abfragen, nur ein Stammelement hat, muss die `FROM` Klausel also mit beginnen. `S3Object[*]` Aus Kompatibilitätsgründen ermöglicht Amazon S3 Select Ihnen das Auslassen des Platzhalterzeichens, wenn Sie keinen Pfad einfügen. Daher ist die vollständige Klausel `FROM S3Object` gleichwertig mit `FROM S3Object[*] as S3Object`. Wenn Sie einen Pfad einfügen, müssen Sie auch das Platzhalterzeichen verwenden. Daher sind sowohl `FROM S3Object` als auch `FROM S3Object[*].`*`path`* gültige Klauseln, nicht aber `FROM S3Object.`*`path`*.

Example

Beispiele:

Beispiel 1

Dieses Beispiel zeigt Ergebnisse unter Verwendung des folgenden Datensatzes und der folgenden Abfrage:

```
{ "Rules": [ {"id": "1"}, {"expr": "y > x"}, {"id": "2", "expr": "z = DEBUG"} ]}
{ "created": "June 27", "modified": "July 6" }
```

```
SELECT id FROM S3Object[*].Rules[*].id
```

```
{"id":"1"}
{}
{"id":"2"}
{}
```

Amazon S3 Select produziert die einzelnen Ergebnisse aus den folgenden Gründen:

- `{"id":"id-1"}` – `S3Object[0].Rules[0].id` produzierte eine Übereinstimmung.
- `{}` – `S3Object[0].Rules[1].id` produzierte keine Übereinstimmung mit einem Datensatz. Daher übermittelte Amazon S3 Select den Wert „`MISSING`“, der anschließend während der Ausgabeserialisierung in einen leeren Datensatz geändert und zurückgegeben wurde.
- `{"id":"id-2"}` – `S3Object[0].Rules[2].id` produzierte eine Übereinstimmung.
- `{}` – `S3Object[1]` produzierte keine Übereinstimmung mit `Rules`. Daher übermittelte Amazon S3 Select den Wert „`MISSING`“, der anschließend während der Ausgabeserialisierung in einen leeren Datensatz geändert und zurückgegeben wurde.

Wenn Sie nicht möchten, dass Amazon S3 Select leere Datensätze zurückgibt, wenn keine Übereinstimmung gefunden wird, können Sie einen Test auf den -Wert ausführe `MISSING`. Die folgende Abfrage gibt dieselben Ergebnisse wie die vorherige Abfrage zurück, jedoch mit Auslassung der leeren Werte:

```
SELECT id FROM S3Object[*].Rules[*].id WHERE id IS NOT MISSING
```

```
{"id":"1"}
{"id":"2"}
```

Beispiel 2

Dieses Beispiel zeigt Ergebnisse unter Verwendung des folgenden Datensatzes und der folgenden Abfragen:

```
{ "created": "936864000", "dir_name": "important_docs", "files": [ { "name": "." },
 { "name": ".." }, { "name": ".aws" }, { "name": "downloads" } ], "owner": "Amazon
 S3" }
{ "created": "936864000", "dir_name": "other_docs", "files": [ { "name": "." },
 { "name": ".." }, { "name": "my stuff" }, { "name": "backup" } ], "owner": "User" }
```

```
SELECT d.dir_name, d.files FROM S3Object[*] d
```

```
{"dir_name":"important_docs","files":[{"name":"."},{"name":".."},{"name":".aws"},
{"name":"downloads"}]}
{"dir_name":"other_docs","files":[{"name":"."},{"name":".."},{"name":"my stuff"},
{"name":"backup"}]}
```

```
SELECT _1.dir_name, _1.owner FROM S3Object[*]
```

```
{"dir_name":"important_docs","owner":"Amazon S3"}
{"dir_name":"other_docs","owner":"User"}
```

WHERE Satz

Die `WHERE`-Klausel hat die folgende Syntax:

```
WHERE condition
```

Die `WHERE`-Klausel filtert Zeilen basierend auf den Wert für „*`condition`*“. Eine Bedingung ist ein Ausdruck, der als Ergebnis einen booleschen Wert zurückgibt. Nur Zeilen, deren Bedingung als `TRUE` ausgewertet wird, werden als Ergebnis zurückgegeben.

LIMIT Satz

Die `LIMIT`-Klausel hat die folgende Syntax:

```
LIMIT number
```

Die `LIMIT`-Klausel begrenzt die Anzahl der von der Abfrage zurückgegebenen Datensätze basierend auf den Wert für „*`number`*“.

Attributzugriff

Die `WHERE` Klauseln `SELECT` und können mithilfe einer der Methoden in den folgenden Abschnitten auf Datensatzdaten verweisen, je nachdem, ob es sich bei der abgefragten Datei um ein oder um ein Format handelt. CSV JSON

CSV

- Spaltennummern — Sie können auf die N-te Spalte einer Zeile mit dem Spaltennamen verweisen, wobei *_N N* ist die Position der Spalte. Der Positionszähler beginnt mit 1. Die erste Spalte heißt demnach _1, die zweite Spalte heißt _2.

  Sie können mit *_N* oder *alias*.*_N* auf eine Spalte verweisen. Beispielsweise sind sowohl _2 und myAlias._2 gültige Möglichkeiten, um auf eine Spalte in der SELECT-Liste und der WHERE-Klausel zu verweisen.
- Spaltenüberschriften — Bei Objekten im CSV Format, die über eine Kopfzeile verfügen, sind die Überschriften für die SELECT Liste und WHERE die Klausel verfügbar. Insbesondere können Sie, wie bei herkömmlichen AusdrückenSQL, innerhalb SELECT - und WHERE Klauselausdrücken, mit oder auf die Spalten verweisen. *alias*.*column_name column_name*

JSON

- Dokument — Sie können auf JSON Dokumentfelder zugreifen als*alias*.*name*. Auch der Zugriff auf verschachtelte Felder ist möglich, z. B *alias*.*name1*.*name2*.*name3*.
- Liste — Sie können auf Elemente in einer JSON Liste zugreifen, indem Sie nullbasierte Indizes mit dem Operator verwenden. [] Beispielsweise lässt sich das zweite Element einer Liste mithilfe von *alias*[1] aufrufen. Sie können den Zugriff auf Listenelemente mit dem Zugriff auf Felder kombinieren, z. B *alias*.*name1*.*name2*[1].*name3*.
- Beispiele: Betrachten Sie dieses JSON Objekt als Beispieldatensatz:

```
{"name": "Susan Smith",
"org": "engineering",
"projects":
    [
     {"project_name":"project1", "completed":false},
     {"project_name":"project2", "completed":true}
    ]
}
```

  Beispiel 1

  Die folgende Abfrage gibt die folgenden Ergebnisse zurück:

```
Select s.name from S3Object s
```

```
{"name":"Susan Smith"}
```

Beispiel 2

Die folgende Abfrage gibt die folgenden Ergebnisse zurück:

```
Select s.projects[0].project_name from S3Object s
```

```
{"project_name":"project1"}
```

Groß-/Kleinschreibung bei Header- und Attributnamen

Mit Amazon S3 Select können Sie doppelte Anführungszeichen verwenden, um anzugeben, dass bei Spaltenüberschriften (für CSV Objekte) und Attributen (für JSON Objekte) zwischen Groß- und Kleinschreibung unterschieden wird. Ohne doppelte Anführungszeichen muss die Groß-/Kleinschreibung bei Objekt-Headern und -attributen nicht berücksichtigt werden. Im Falle einer Zweideutigkeit wird ein Fehler ausgegeben.

Die folgenden Beispiele sind entweder 1) Amazon S3 S3-Objekte im CSV Format mit den angegebenen Spaltenüberschriften und mit der `FileHeaderInfo` Einstellung auf `"Use"` für die Abfrageanforderung oder 2) Amazon S3 S3-Objekte im JSON Format mit den angegebenen Attributen.

Beispiel 1: Das abzufragende Objekt hat den Header oder das Attribut „`NAME`“.

- Mit folgendem Ausdruck werden Werte von dem Objekt erfolgreich zurückgegeben. Da keine Anführungszeichen vorhanden sind, berücksichtigt die Abfrage die Groß- und Kleinschreibung nicht.

  ```
  SELECT s.name from S3Object s
  ```

- Der folgende Ausdruck führt zum Fehler 400 „`MissingHeaderName`“. Da Anführungszeichen vorhanden sind, berücksichtigt die Abfrage die Groß- und Kleinschreibung.

  ```
  SELECT s."name" from S3Object s
  ```

Beispiel 2: Das abzufragende Amazon S3-Objekt hat einen Header oder ein Attribut mit „NAME“ sowie einen anderen Header oder ein anderes Attribut mit „name“.

- Der folgende Ausdruck führt zum Fehler 400 „AmbiguousFieldName“. Da keine Anführungszeichen vorhanden sind, wird die Groß-/Kleinschreibung nicht beachtet, aber es gibt zwei Übereinstimmungen, weshalb der Fehler ausgegeben wird.

```
SELECT s.name from S3Object s
```

- Mit folgendem Ausdruck werden Werte von dem Objekt erfolgreich zurückgegeben. Da Anführungszeichen vorhanden sind, berücksichtigt die Abfrage die Groß- und Kleinschreibung, weshalb es keine Mehrdeutigkeit gibt.

```
SELECT s."NAME" from S3Object s
```

Verwenden von reservierten Schlüsselwörtern als benutzerdefinierte Begriffe

Amazon S3 Select verfügt über eine Reihe von reservierten Schlüsselwörtern, die zur Ausführung der SQL Ausdrücke benötigt werden, die zur Abfrage von Objektinhalten verwendet werden. Zu den reservierten Schlüsselwörtern zählen Funktionsnamen, Datentypen, Operatoren etc. In einigen Fällen können benutzerdefinierte Begriffe, wie die Spaltenüberschriften (für CSV Dateien) oder Attribute (für JSON Objekte), mit einem reservierten Schlüsselwort kollidieren. In diesem Fall geben Sie mithilfe von doppelten Anführungszeichen an, dass Sie absichtlich einen benutzerdefinierten Begriff verwenden, der mit einem reservierten Schlüsselwort übereinstimmt. Andernfalls wird ein 400-Parse-Fehler ausgegeben.

Eine vollständige Liste der reservierten Schlüsselwörter finden Sie unter Reservierte Schlüsselwörter.

Das folgende Beispiel ist entweder 1) ein Amazon S3 S3-Objekt im CSV Format mit den angegebenen Spaltenüberschriften, das `"Use"` für die Abfrageanforderung auf `FileHeaderInfo` gesetzt ist, oder 2) ein Amazon S3 S3-Objekt im JSON Format mit den angegebenen Attributen.

Beispiel: Das abzufragende Objekt hat einen Header oder ein Attribut mit dem Namen „CAST“, der ein reserviertes Schlüsselwort ist.

- Mit folgendem Ausdruck werden Werte von dem Objekt erfolgreich zurückgegeben. Da die Abfrage Anführungszeichen enthält, verwendet S3 Select den benutzerdefinierten Header oder das benutzerdefinierte Attribut.

```
SELECT s."CAST" from S3Object s
```

- Der folgende Ausdruck führt zu einem Analysefehler 400. Da in der Abfrage keine Anführungszeichen verwendet werden, steht CAST im Konflikt mit einem reservierten Schlüsselwort.

```
SELECT s.CAST from S3Object s
```

Skalare Ausdrücke

In der WHERE Klausel und der SELECT Liste können Sie SQL skalare Ausdrücke verwenden, d. h. Ausdrücke, die Skalarwerte zurückgeben. Sie haben das folgende Format:

- ***Literal***

  Ein SQL Literal.

- ***column_reference***

  Ein Verweis auf eine Spalte in der Form *column_name* oder *alias*.*column_name*.

- ***unary_op expression***

  In diesem Fall ***unary_op***ist ein unärer OperatorSQL.

- ***expression binary_op expression***

  In diesem Fall ***binary_op*** ist es ein SQL binärer Operator.

- ***func_name***

  In diesem Fall ist ***func_name*** der Name der aufzurufenden skalaren Funktion.

- ***expression*** [NOT] BETWEEN ***Ausdruck*** AND ***Ausdruck***

- ***expression*** LIKE ***Ausdruck*** [ESCAPE ***expression***]

Datentypen

> **Important**
>
> Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. Weitere Informationen

Amazon S3 Select unterstützt mehrere primitive Datentypen.

Datentypkonvertierungen

Allgemein sollte die Funktion `CAST` verwendet werden, wenn sie definiert ist. Falls `CAST` nicht definiert ist, werden alle Eingabedaten als Zeichenfolge betrachtet. In diesem Fall müssen Sie Ihre Eingabedaten bei Bedarf in die relevanten Datentypen umwandeln.

Weitere Informationen zur Funktion `CAST` finden Sie unter CAST.

Unterstützte Datentypen

Amazon S3 Select unterstützt die folgenden primitiven Datentypen.

Name	Beschreibung	Beispiele
`bool`	Ein boolescher Wert, entweder `TRUE` oder `FALSE`.	`FALSE`
`int`, `integer`	Eine 8-Byte-Ganzzahl im Bereich -9.223.372.036.854 .775.808 bis 9.223.372.036.854.775.807.	`100000`
`string`	Eine UTF8 -kodierte Zeichenfolge mit variabler Länge. Das Standardlimit ist 1 Zeichen. Das maximale Zeichenlimit beträgt 2.147.483.647.	`'xyz'`
`float`	Eine 8-Byte-Gleitkommazahl.	`CAST(0.456 AS FLOAT)`
`decimal`, `numeric`	Eine Zahl mit der Basis 10 mit einer maximalen Genauigkeit von 38 (d. h. maximale Anzahl signifikanter Ziffern) und einer Größe zwischen -2^{31} und $2^{31}-1$ (d. h. der Exponent der Basis 10).	`123.456`

Name	Beschreibung	Beispiele
	Note Amazon S3 Select ignoriert die Skalierung und Genauigkeit, wenn Sie beide gleichzeitig bereitstellen.	
`timestamp`	Zeitstempel stellen einen bestimmten Zeitpunkt dar, enthalten immer einen lokalen Versatz und können beliebig genau sein. Im Textformat befolgen Zeitstempel den W3C-Hinweis zu Datums- und Zeitformaten, müssen jedoch mit dem Buchstaben „T“ enden, wenn die Präzision der Zeitstempel nicht mindestens ein ganzer Tag ist. Sekundenbruchteile mit mindestens einer Stelle bis zu beliebig vielen Stellen sind zulässig. Offsets zur Ortszeit können entweder als Zeitversatz von Stunde zu Minute oder als Literalwert für die Ortszeit von UTC dargestellt werden. `Z` UTC Lokale Zeitversätze sind für Zeitstempel mit Uhrzeit erforderlich, für reine Datumswerte jedoch nicht zulässig.	`CAST('2007-04-05T14:30Z' AS TIMESTAMP)`

Unterstützt Parquet Typen

Amazon S3 Select unterstützt Folgendes Parquet Typen.

- `DATE`
- `DECIMAL`
- `ENUM`
- `INT(8)`
- `INT(16)`
- `INT(32)`
- `INT(64)`

- `LIST`

> **Note**
>
> Für `LIST` Parquet Ausgabetyp, Amazon S3 Select unterstützt nur JSON das Format. Wenn die Abfrage die Daten jedoch auf einfache Werte beschränkt, `LIST` Parquet Der Typ kann auch im CSV Format abgefragt werden.

- `STRING`
- `TIMESTAMP`-unterstützte Präzision (`MILLIS`/`MICROS`/`NANOS`)

> **Note**
>
> Zeitstempel, die als `INT(96)` gespeichert werden, werden nicht unterstützt.
> Aufgrund des Bereichs des Typs `INT(64)` können Zeitstempel in der Einheit `NANOS` nur Werte zwischen `1677-09-21 00:12:43` und `2262-04-11 23:47:16` darstellen. Werte, die außerhalb dieses Bereichs liegen, können mit der `NANOS`-Einheit nicht dargestellt werden.

Zuordnung von Parquet Typen zu unterstützten Datentypen in Amazon S3 Select

Parquet Typen	Unterstützte Datentypen
`DATE`	`timestamp`
`DECIMAL`	`decimal`, `numeric`
`ENUM`	`string`
`INT(8)`	`int`, `integer`
`INT(16)`	`int`, `integer`
`INT(32)`	`int`, `integer`

Parquet Typen	Unterstützte Datentypen
`INT(64)`	`decimal`, `numeric`
`LIST`	Jeder Parquet Der Typ in der Liste ist dem entsprechenden Datentyp zugeordnet.
`STRING`	`string`
`TIMESTAMP`	`timestamp`

Operatoren

> **Important**
>
> Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. Weitere Informationen

Amazon S3 Select unterstützt die folgenden Operatoren.

Logische Operatoren

- `AND`
- `NOT`
- `OR`

Vergleichsoperatoren

- `<`
- `>`
- `<=`
- `>=`
- `=`

- `<>`
- `!=`
- `BETWEEN`
- `IN` – Beispiel: `IN ('a', 'b', 'c')`

Mustervergleichsoperatoren

- `LIKE`
- `_` (Vergleicht ein beliebiges Zeichen)
- `%` (Vergleicht eine beliebige Zeichenreihenfolge)

Einheitliche Operatoren

- `IS NULL`
- `IS NOT NULL`

Mathematische Operatoren

Es werden Addition, Subtraktion, Multiplikation, Division und Modulo wie folgt unterstützt:

- `+`
- `-`
- `*`
- `/`
- `%`

Rangfolge der Operatoren

Die folgende Tabelle zeigt die Rangfolge der Operatoren in absteigender Reihenfolge.

Operator oder Element	Assoziativität	Erforderlich
-	rechts	unär minus

Operator oder Element	Assoziativität	Erforderlich
*, /, %	links	Multiplikation, Division und Modulo
+, -	links	Addition und Subtraktion
IN		Mitgliedschaft festlegen
BETWEEN		in Bereich enthalten
LIKE		Zeichenfolgenübereinstimmung
<>		kleiner als, größer als
=	rechts	Gleichheit, Zuweisung
NOT	rechts	logische Negation
AND	links	logische Verbindung
OR	links	logische Disjunktion

Reservierte Schlüsselwörter

> **Important**
>
> Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. Weitere Informationen

Nachfolgend finden Sie eine Liste der reservierten Schlüsselwörter für Amazon S3 Select. Zu diesen Schlüsselwörtern gehören Funktionsnamen, Datentypen, Operatoren usw., die zur Ausführung der SQL Ausdrücke benötigt werden, die zur Abfrage von Objektinhalten verwendet werden.

```
absolute
action
add
all
allocate
alter
and
any
are
as
asc
assertion
at
authorization
avg
bag
begin
between
bit
bit_length
blob
bool
boolean
both
by
cascade
cascaded
case
cast
catalog
```

```
char
char_length
character
character_length
check
clob
close
coalesce
collate
collation
column
commit
connect
connection
constraint
constraints
continue
convert
corresponding
count
create
cross
current
current_date
current_time
current_timestamp
current_user
cursor
date
day
deallocate
dec
decimal
declare
default
deferrable
deferred
delete
desc
describe
descriptor
diagnostics
disconnect
distinct
```

```
domain
double
drop
else
end
end-exec
escape
except
exception
exec
execute
exists
external
extract
false
fetch
first
float
for
foreign
found
from
full
get
global
go
goto
grant
group
having
hour
identity
immediate
in
indicator
initially
inner
input
insensitive
insert
int
integer
intersect
interval
```

```
into
is
isolation
join
key
language
last
leading
left
level
like
limit
list
local
lower
match
max
min
minute
missing
module
month
names
national
natural
nchar
next
no
not
null
nullif
numeric
octet_length
of
on
only
open
option
or
order
outer
output
overlaps
pad
```

```
partial
pivot
position
precision
prepare
preserve
primary
prior
privileges
procedure
public
read
real
references
relative
restrict
revoke
right
rollback
rows
schema
scroll
second
section
select
session
session_user
set
sexp
size
smallint
some
space
sql
sqlcode
sqlerror
sqlstate
string
struct
substring
sum
symbol
system_user
table
```

```
temporary
then
time
timestamp
timezone_hour
timezone_minute
to
trailing
transaction
translate
translation
trim
true
tuple
union
unique
unknown
unpivot
update
upper
usage
user
using
value
values
varchar
varying
view
when
whenever
where
with
work
write
year
zone
```

SQLFunktionen

> **Important**
>
> Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. Weitere Informationen

Amazon S3 Select unterstützt die folgenden SQL Funktionen.

Themen

- Aggregationsfunktionen
- Konditionale Funktionen
- Konvertierungs-Funktionen
- Datumsfunktionen
- Zeichenfolgenfunktionen

Aggregationsfunktionen

> **Important**
>
> Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. Weitere Informationen

Amazon S3 Select unterstützt die folgenden aggregierten Funktionen.

Funktion	Argumenttyp	Rückgabetyp
`AVG(`*`expressio`* *`n`* `)`	`INT`, `FLOAT`, `DECIMAL`	`DECIMAL` für ein `INT`-Argument, `FLOAT` für ein Gleitkomma-Argument; andernfal

Funktion	Argumenttyp	Rückgabetyp
		ls gleich dem Argumentd atentyp.
COUNT	-	INT
MAX(*expressio n*)	INT, DECIMAL	Entspricht dem Argumenttyp.
MIN(*expressio n*)	INT, DECIMAL	Entspricht dem Argumenttyp.
SUM(*expressio n*)	INT, FLOAT, DOUBLE, DECIMAL	INT für ein INT-Argument , FLOAT für ein Gleitkomma-Argumen t; andernfal ls gleich dem Argumentd atentyp.

SUM Beispiel

Verwenden Sie einen SUM-Ausdruck, um die gesamten Objektgrößen eines Ordners in einem S3-Bestandslistenbericht zusammenzufassen.

Der folgende S3-Inventarbericht ist eine CSV Datei, die mit komprimiert wurdeGZIP. Sie hat drei Spalten.

- Die erste Spalte enthält den Namen des S3-Buckets (*DOC-EXAMPLE-BUCKET*), für den der S3-Inventarbericht bestimmt ist.
- Die zweite Spalte enthält den Objektschlüsselnamen, der das Objekt eindeutig im Bucket identifiziert.

Der *example-folder/*-Wert in der ersten Zeile steht für den Ordner *example-folder*. Wenn Sie in Amazon S3 einen Ordner in Ihrem Bucket anlegen, erstellt S3 ein 0-Byte-Objekt mit einem Schlüssel, der auf den von Ihnen angegebenen Ordnernamen festgelegt ist.

Der *example-folder/object1*-Wert in der zweiten Zeile steht für das Objekt *object1* im Ordner *example-folder*.

Der *example-folder/object2*-Wert in der dritten Zeile steht für das Objekt *object2* im Ordner *example-folder*.

Weitere Informationen über S3-Ordner finden Sie unter Organisieren von Objekten in der Amazon S3-Konsole mithilfe von Ordnern.

- Die dritte Spalte enthält die Objektgröße in Byte.

```
"DOC-EXAMPLE-BUCKET","example-folder/","0"
"DOC-EXAMPLE-BUCKET","example-folder/object1","2011267"
"DOC-EXAMPLE-BUCKET","example-folder/object2","1570024"
```

Um einen SUM Ausdruck zur Berechnung der Gesamtgröße des Ordners zu verwenden*example-folder*, führen Sie die SQL Abfrage mit Amazon S3 Select aus.

```
SELECT SUM(CAST(_3 as INT)) FROM s3object s WHERE _2 LIKE 'example-folder/%' AND _2 !=
 'example-folder/';
```

Abfrageergebnis:

```
3581291
```

Konditionale Funktionen

> **Important**
>
> Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. Weitere Informationen

Amazon S3 Select unterstützt die folgenden bedingten Funktionen.

Themen

- CASE
- COALESCE
- NULLIF

CASE

Der CASE-Ausdruck ist ein bedingter Ausdruck, der sich mit den `if/then/else`-Anweisungen anderer Sprachen vergleichen lässt. CASE wird verwendet, um ein Ergebnis anzugeben, wenn es mehrere Bedingungen gibt. Es gibt zwei Arten von CASE-Ausdrücken: einfach und gesucht.

In einfachen CASE-Ausdrücken wird ein Ausdruck mit einem Wert verglichen. Wenn keine Übereinstimmung gefunden wird, wird die in der THEN-Klausel angegebene Aktion angewendet. Wenn keine Übereinstimmung gefunden wird, wird die in der ELSE-Klausel angegebene Aktion angewendet.

In gesuchten CASE-Ausdrücken wird jeder CASE-Ausdruck auf der Basis eines booleschen Ausdrucks evaluiert und die CASE-Anweisung gibt den ersten übereinstimmenden CASE-Ausdruck zurück. Wenn in den WHEN-Klauseln kein übereinstimmender CASE-Ausdruck gefunden wird, wird die Aktion in der ELSE-Klausel zurückgegeben.

Syntax

> **Note**
>
> Derzeit unterstützt Amazon S3 Select weder `ORDER BY` noch Abfragen, die neue Zeilen enthalten. Achten Sie darauf, Abfragen ohne Zeilenumbrüche zu verwenden.

Die folgende Zeichenfolge ist eine einfache CASE-Aussage, die verwendet wird, um Bedingungen abzugleichen:

```
CASE expression WHEN value THEN result [WHEN...] [ELSE result] END
```

Die folgende Zeichenfolge ist eine gesuchte CASE-Anweisung, die verwendet wird, um jede Bedingung auszuwerten:

```
CASE WHEN boolean condition THEN result [WHEN ...] [ELSE result] END
```

Beispiele

> **Note**
>
> Wenn Sie die folgenden Beispiele mit der Amazon S3 S3-Konsole ausführen und Ihre CSV Datei eine Kopfzeile enthält, wählen Sie Die erste CSV Datenzeile ausschließen.

Beispiel 1: Verwenden Sie einen einfachen CASE-Ausdruck, um `New York City` in einer Abfrage durch `Big Apple` zu ersetzen. Alle anderen Städtenamen werden durch `other` ersetzt.

```
SELECT venuecity, CASE venuecity WHEN 'New York City' THEN 'Big Apple' ELSE 'other' END
 FROM S3Object;
```

Abfrageergebnis:

```
venuecity        |   case
-----------------+-----------
Los Angeles      | other
New York City    | Big Apple
San Francisco    | other
Baltimore        | other
...
```

Beispiel 2: Verwenden Sie einen gesuchten CASE-Ausdruck, um Gruppennummern basierend auf dem `pricepaid`-Wert für einzelne Ticketverkäufe zuzuweisen:

```
SELECT pricepaid, CASE WHEN CAST(pricepaid as FLOAT) < 10000 THEN 'group 1' WHEN
 CAST(pricepaid as FLOAT) > 10000 THEN 'group 2' ELSE 'group 3' END FROM S3Object;
```

Abfrageergebnis:

```
pricepaid |  case
-----------+---------
12624.00 | group 2
10000.00 | group 3
10000.00 | group 3
9996.00 | group 1
9988.00 | group 1
...
```

COALESCE

COALESCE wertet die Argumente nacheinander aus und gibt den ersten unbekannten Wert zurück, das heißt, den ersten Wert, der nicht null oder nicht fehlend ist. Null- und fehlende Werte werden von der Funktion nicht übernommen.

Syntax

```
COALESCE ( expression, expression, ... )
```

Parameter

expression

Der Zielausdruck, der von der Funktion verwendet wird.

Beispiele

```
COALESCE(1)                -- 1
COALESCE(null)             -- null
COALESCE(null, null)       -- null
COALESCE(missing)          -- null
COALESCE(missing, missing) -- null
COALESCE(1, null)          -- 1
COALESCE(null, null, 1)    -- 1
COALESCE(null, 'string')   -- 'string'
COALESCE(missing, 1)       -- 1
```

NULLIF

Bei zwei Ausdrücken, die das gleiche Auswertungsergebnis haben, gibt NULLIF NULL zurück. Andernfalls gibt NULLIF das Auswertungsergebnis für den ersten Ausdruck zurück.

Syntax

```
NULLIF ( expression1, expression2 )
```

Parameter

expression1, expression2

Die Zielausdrücke, die von der Funktion verwendet werden.

Beispiele

```
NULLIF(1, 1)             -- null
NULLIF(1, 2)             -- 1
NULLIF(1.0, 1)           -- null
NULLIF(1, '1')           -- 1
NULLIF([1], [1])         -- null
NULLIF(1, NULL)          -- 1
NULLIF(NULL, 1)          -- null
NULLIF(null, null)       -- null
NULLIF(missing, null)    -- null
NULLIF(missing, missing) -- null
```

Konvertierungs-Funktionen

> **Important**
>
> Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. Weitere Informationen

Amazon S3 Select unterstützt die folgende Konvertierungsfunktion.

Themen

- CAST

CAST

Die CAST-Funktion konvertiert ein Element, z. B. einen Ausdruck zur Auswertung eines einzelnen Werts, von einem Typ in einen anderen.

Syntax

```
CAST ( expression AS data_type )
```

Parameter

expression

Eine Kombination aus einem oder mehreren Werten, Operatoren und SQL Funktionen, die einen Wert ergeben.

data_type

Der Zieldatentyp, z. B. INT, in den der Ausdruck umgewandelt werden soll Eine Liste der unterstützten Datentypen finden Sie unter Datentypen.

Beispiele

```
CAST('2007-04-05T14:30Z' AS TIMESTAMP)
CAST(0.456 AS FLOAT)
```

Datumsfunktionen

> **Important**
>
> Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. Weitere Informationen

Amazon S3 Select unterstützt die folgenden Datumsfunktionen.

Themen

- DATE_ADD
- DATE_DIFF
- EXTRACT
- TO_STRING
- TO_TIMESTAMP
- UTCNOW

DATE_ADD

Bei einem Datumsteil, einer Menge und einem Zeitstempel gibt DATE_ADD einen aktualisierten Zeitstempel zurück, indem der Datumsteil anhand der Menge modifiziert wird.

Syntax

```
DATE_ADD( date_part, quantity, timestamp )
```

Parameter

date_part

Gibt den zu modifizierenden Teil des Datums an. Dabei kann es sich um einen der folgenden Werte handeln:

- Jahr
- Monat
- Tag
- Stunde
- Minute
- Sekunde

quantity

Der Wert, der auf den aktualisierten Zeitstempel anzuwenden ist. Positive „*quantity*“-Werte werden zum „date_part“-Wert des Zeitstempels addiert, negative Werte werden subtrahiert.

timestamp

Der Zielzeitstempel, der von der Funktion verwendet wird.

Beispiele

```
DATE_ADD(year, 5, `2010-01-01T`)                -- 2015-01-01 (equivalent to
 2015-01-01T)
DATE_ADD(month, 1, `2010T`)                     -- 2010-02T (result will add precision
 as necessary)
DATE_ADD(month, 13, `2010T`)                    -- 2011-02T
DATE_ADD(day, -1, `2017-01-10T`)                -- 2017-01-09 (equivalent to
 2017-01-09T)
DATE_ADD(hour, 1, `2017T`)                      -- 2017-01-01T01:00-00:00
DATE_ADD(hour, 1, `2017-01-02T03:04Z`)          -- 2017-01-02T04:04Z
DATE_ADD(minute, 1, `2017-01-02T03:04:05.006Z`) -- 2017-01-02T03:05:05.006Z
DATE_ADD(second, 1, `2017-01-02T03:04:05.006Z`) -- 2017-01-02T03:04:06.006Z
```

DATE_DIFF

Bei einem Datumsteil und zwei gültigen Zeitstempeln gibt DATE_DIFF die Differenz in Datumsteilen an. Sofern der *date_part*-Wert von *timestamp1* größer ist als der *date_part*-Wert von

timestamp2, wird eine negative Ganzzahl zurückgegeben. Wenn der *date_part*-Wert von *timestamp1* kleiner ist als der *date_part*-Wert von *timestamp2*, wird eine positive Ganzzahl zurückgegeben.

Syntax

```
DATE_DIFF( date_part, timestamp1, timestamp2 )
```

Parameter

date_part

Gibt den zu vergleichenden Teil der Zeitstempel an. Die Definition von `date_part` finden Sie unter DATE_ADD.

timestamp1

Der erste Zeitstempel für den Vergleich.

timestamp2

Der zweite Zeitstempel für den Vergleich.

Beispiele

```
DATE_DIFF(year, `2010-01-01T`, `2011-01-01T`)           -- 1
DATE_DIFF(year, `2010T`, `2010-05T`)                    -- 4 (2010T is equivalent to
 2010-01-01T00:00:00.000Z)
DATE_DIFF(month, `2010T`, `2011T`)                      -- 12
DATE_DIFF(month, `2011T`, `2010T`)                      -- -12
DATE_DIFF(day, `2010-01-01T23:00`, `2010-01-02T01:00`) -- 0 (need to be at least 24h
 apart to be 1 day apart)
```

EXTRACT

Bei einem Datumsteil und einem Zeitstempel gibt `EXTRACT` den Datumsteilwert des Zeitstempels zurück.

Syntax

```
EXTRACT( date_part FROM timestamp )
```

Parameter

date_part

Gibt den zu extrahierenden Teil der Zeitstempel an. Dabei kann es sich um einen der folgenden Werte handeln:

- YEAR
- MONTH
- DAY
- HOUR
- MINUTE
- SECOND
- TIMEZONE_HOUR
- TIMEZONE_MINUTE

Zeitstempel

Der Zielzeitstempel, der von der Funktion verwendet wird.

Beispiele

```
EXTRACT(YEAR FROM `2010-01-01T`)                              -- 2010
EXTRACT(MONTH FROM `2010T`)                                   -- 1 (equivalent to
 2010-01-01T00:00:00.000Z)
EXTRACT(MONTH FROM `2010-10T`)                                -- 10
EXTRACT(HOUR FROM `2017-01-02T03:04:05+07:08`)                -- 3
EXTRACT(MINUTE FROM `2017-01-02T03:04:05+07:08`)              -- 4
EXTRACT(TIMEZONE_HOUR FROM `2017-01-02T03:04:05+07:08`)       -- 7
EXTRACT(TIMEZONE_MINUTE FROM `2017-01-02T03:04:05+07:08`)     -- 8
```

TO_STRING

Bei einem Zeitstempel und einem Formatmuster gibt TO_STRING eine Zeichenfolgendarstellung des Zeitstempels im angegebenen Format zurück.

Syntax

```
TO_STRING ( timestamp time_format_pattern )
```

Parameter

timestamp

Der Zielzeitstempel, der von der Funktion verwendet wird.

time_format_pattern

Eine Zeichenfolge mit folgenden speziellen Zeichenbedeutungen:

Format	Beispiel	Beschreibung
yy	69	Jahreszahl mit 2 Ziffern
y	1969	Jahreszahl mit 4 Ziffern
yyyy	1969	Jahreszahl mit 4 Ziffern, mit Nullen aufgefüllt
M	1	Monatsname
MM	01	Monatsname, mit Nullen aufgefüllt
MMM	Jan	Abkürzung des Monatsnamens
MMMM	January	Vollständiger Monatsnamen
MMMMM	J	Erster Buchstabe des Monats des Jahres

Format	Beispiel	Beschreibung
		(NOTE: Dieses Format ist für die Verwendung mit der `TO_TIMESTAMP` Funktion nicht gültig.)
d	2	Monatstag (1-31)
dd	02	Monatstag, mit Nullen aufgefüllt (01-31)
a	AM	AM oder PM
h	3	Stunde (1-12)
hh	03	Stunde, mit Nullen aufgefüllt (01-12)
H	3	Stunde (0-23)
HH	03	Stunde, mit Nullen aufgefüllt (00-23)
m	4	Minute (0-59)

Format	Beispiel	Beschreibung
mm	04	Minute, mit Nullen aufgefüllt (00-59)
s	5	Sekunde (0-59)
ss	05	Sekunde, mit Nullen aufgefüllt (00-59)
S	0	Sekundenbruchteil (Genauigkeit: 0,1, Bereich: 0,0-0,9)
SS	6	Sekundenbruchteil (Genauigkeit: 0,01, Bereich: 0,0-0,99)
SSS	60	Sekundenbruchteil (Genauigkeit: 0,001, Bereich: 0,0-0,999)
…	…	…

Format	Beispiel	Beschreibung
SSSSSSSSS	60000000	Sekundenbruchteil (max. Genauigkeit: 1 Nanosekunde, Bereich: 0,0-0,999999999)
n	60000000	Nanosekunde
X	+07 oder Z	Offset in Stunden oder „Z“ bei Offset = 0
XX oder XXXX	+0700 oder Z	Offset in Stunden und Minuten oder „Z“ bei Offset = 0
XXX oder XXXXX	+07:00 oder Z	Offset in Stunden und Minuten oder „Z“ bei Offset = 0
x	7	Offset in Stunden
xx oder xxxx	700	Offset in Stunden und Minuten

Format	Beispiel	Beschreibung
`xxx` oder `xxxxx`	`+07:00`	Offset in Stunden und Minuten

Beispiele

```
TO_STRING(`1969-07-20T20:18Z`,  'MMMM d, y')                    -- "July 20, 1969"
TO_STRING(`1969-07-20T20:18Z`, 'MMM d, yyyy')                   -- "Jul 20, 1969"
TO_STRING(`1969-07-20T20:18Z`, 'M-d-yy')                        -- "7-20-69"
TO_STRING(`1969-07-20T20:18Z`, 'MM-d-y')                        -- "07-20-1969"
TO_STRING(`1969-07-20T20:18Z`, 'MMMM d, y h:m a')               -- "July 20, 1969 8:18
 PM"
TO_STRING(`1969-07-20T20:18Z`, 'y-MM-dd''T''H:m:ssX')           --
 "1969-07-20T20:18:00Z"
TO_STRING(`1969-07-20T20:18+08:00Z`, 'y-MM-dd''T''H:m:ssX')     --
 "1969-07-20T20:18:00Z"
TO_STRING(`1969-07-20T20:18+08:00`, 'y-MM-dd''T''H:m:ssXXXX')   --
 "1969-07-20T20:18:00+0800"
TO_STRING(`1969-07-20T20:18+08:00`, 'y-MM-dd''T''H:m:ssXXXXX')  --
 "1969-07-20T20:18:00+08:00"
```

TO_TIMESTAMP

Bei einer Zeichenfolge wandelt `TO_TIMESTAMP` diese in einen Zeitstempel um. `TO_TIMESTAMP` ist der umgekehrte Vorgang von `TO_STRING`.

Syntax

```
TO_TIMESTAMP ( string )
```

Parameter

`string`

Die Zielzeichenfolge, die von der Funktion verwendet wird.

Beispiele

```
TO_TIMESTAMP('2007T')                          -- `2007T`
TO_TIMESTAMP('2007-02-23T12:14:33.079-08:00') -- `2007-02-23T12:14:33.079-08:00`
```

UTCNOW

`UTCNOW`gibt die aktuelle Uhrzeit UTC als Zeitstempel zurück.

Syntax

```
UTCNOW()
```

Parameter

`UTCNOW` nutzt keine Parameter.

Beispiele

```
UTCNOW() -- 2017-10-13T16:02:11.123Z
```

Zeichenfolgenfunktionen

> **Important**
>
> Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. Weitere Informationen

Amazon S3 Select unterstützt die folgenden String-Funktionen.

Themen

- CHAR_LENGTH, CHARACTER_LENGTH
- LOWER
- SUBSTRING
- TRIM
- UPPER

CHAR_LENGTH, CHARACTER_LENGTH

CHAR_LENGTH (oder CHARACTER_LENGTH) zählt die Anzahl der Zeichen in der angegebenen Zeichenfolge.

> **Note**
>
> CHAR_LENGTH und CHARACTER_LENGTH sind Synonyme.

Syntax

```
CHAR_LENGTH ( string )
```

Parameter

string

Die Zielzeichenfolge, die von der Funktion verwendet wird.

Beispiele

```
CHAR_LENGTH('')          -- 0
CHAR_LENGTH('abcdefg')   -- 7
```

LOWER

LOWER wandelt alle Großbuchstaben einer Zeichenfolge in Kleinbuchstaben um. Alle Zeichen, die keine Großbuchstaben sind, bleiben unverändert.

Syntax

```
LOWER ( string )
```

Parameter

Zeichenfolge

Die Zielzeichenfolge, die von der Funktion verwendet wird.

Beispiele

```
LOWER('AbCdEfG!@#$') -- 'abcdefg!@#$'
```

SUBSTRING

SUBSTRING gibt bei einer Zeichenfolge, einem Startindex und (optional) einer Länge die Teilzeichenfolge vom Startindex bis zum Ende der Zeichenfolge oder bis zur angegebenen Länge zurück.

> **Note**
>
> Das erste Zeichen der Eingabezeichenfolge hat die Indexposition 1.
>
> - Wenn `start` < 1 ist und keine Länge angegeben wird, dann wird die Indexposition auf 1 festgelegt.
> - Wenn `start` < 1 ist und eine Länge angegeben wird, dann wird die Indexposition auf `start + length -1` festgelegt.
> - Wenn `start + length -1` < 0 ist, dann wird eine leere Zeichenfolge zurückgegeben.
> - Wenn `start + length -1` >= 0 ist, dann wird die Teilzeichenfolge beginnend bei Indexposition 1 mit der Länge `start + length - 1` zurückgegeben.

Syntax

```
SUBSTRING( string FROM start [ FOR length ] )
```

Parameter

Zeichenfolge

Die Zielzeichenfolge, die von der Funktion verwendet wird.

start

Die Startposition der Zeichenfolge.

length

Die Länge der zurückzugebenden Teilzeichenfolge. Falls nicht angegeben, wird bis zum Ende der Zeichenfolge fortgefahren.

Beispiele

```
SUBSTRING("123456789", 0)      -- "123456789"
SUBSTRING("123456789", 1)      -- "123456789"
SUBSTRING("123456789", 2)      -- "23456789"
SUBSTRING("123456789", -4)     -- "123456789"
SUBSTRING("123456789", 0, 999) -- "123456789"
SUBSTRING("123456789", 1, 5)   -- "12345"
```

TRIM

Kürzt vorangestellte oder nachgestellte Zeichen aus einer Zeichenfolge. Standardmäßig wird ein Leerzeichen (`' '`) entfernt.

Syntax

```
TRIM ( [[LEADING | TRAILING | BOTH remove_chars] FROM] string )
```

Parameter

Zeichenfolge

Die Zielzeichenfolge, die von der Funktion verwendet wird.

LEADING | TRAILING | BOTH

Dieser Parameter gibt an, ob vorangestellte oder nachgestellte Zeichen oder vorangestellte und nachgestellte Zeichen gekürzt werden sollen.

remove_chars

Die zu entfernenden Zeichen. *remove_chars* kann eine Zeichenfolge mit einer Länge von > 1 sein. Diese Funktion gibt die Zeichenfolge mit einem beliebigen Zeichen von *remove_chars* zurück, der am Anfang oder am Ende der entfernten Zeichenfolge stand.

Beispiele

```
TRIM('       foobar         ')              -- 'foobar'
TRIM('      \tfoobar\t         ')           -- '\tfoobar\t'
TRIM(LEADING FROM '       foobar         ')  -- 'foobar         '
TRIM(TRAILING FROM '       foobar         ') -- '       foobar'
TRIM(BOTH FROM '       foobar         ')     -- 'foobar'
```

```
TRIM(BOTH '12' FROM '1112211foobar22211122') -- 'foobar'
```

UPPER

UPPER wandelt alle Kleinbuchstaben einer Zeichenfolge in Großbuchstaben um. Alle Zeichen, die keine Kleinbuchstaben sind, bleiben unverändert.

Syntax

```
UPPER ( string )
```

Parameter

Zeichenfolge

Die Zielzeichenfolge, die von der Funktion verwendet wird.

Beispiele

```
UPPER('AbCdEfG!@#$') -- 'ABCDEFG!@#$'
```

Arbeiten mit Verzeichnis-Buckets

Es gibt zwei Arten von Amazon-S3-Buckets: Allzweck-Buckets und Verzeichnis-Buckets. Wählen Sie den Bucket-Typ, der am besten zu Ihren Anwendungs- und Leistungsanforderungen passt:

- Allzweck-Buckets sind der ursprüngliche S3-Bucket-Typ und werden für die meisten Anwendungsfälle und Zugriffsmuster empfohlen. Buckets für allgemeine Zwecke ermöglichen auch Objekte, die in allen Speicherklassen gespeichert werden, mit Ausnahme von S3 Express One Zone in AWS Availability Zones.
- Verzeichnis-Buckets unterstützen die Erstellung von Buckets im folgenden Bucket-Speicherortyp: Availability Zone. Für Anwendungsfälle mit geringer Latenz können Sie einen Verzeichnis-Bucket in einer einzelnen Availability Zone erstellen, um Daten zu speichern. Verzeichnis-Buckets in Availability Zones unterstützen die Speicherklasse S3 Express One Zone. Die Speicherklasse S3 Express One Zone wird empfohlen, wenn Ihre Anwendung leistungsempfindlich ist und von einstelligen Millisekunden `PUT` und Latenzen profitiert. `GET`

Verzeichnis-Buckets organisieren Daten hierarchisch in Verzeichnissen, im Gegensatz zu der flachen Speicherstruktur von Allzweck-Buckets. Es gibt keine Präfixbeschränkungen für Verzeichnis-Buckets und einzelne Verzeichnisse können horizontal skaliert werden.

Sie können in jedem Ihrer bis zu 100 Verzeichnis-Buckets erstellen, wobei die Anzahl der Objekte AWS-Konten, die Sie in einem Bucket speichern können, unbegrenzt ist. Ihr Bucket-Kontingent wird au jede Region in Ihrem AWS-Konto angewendet. Wenn Ihre Anwendung eine Erhöhung dieses Limits erfordert, wenden Sie sich an AWS Support.

> **Important**
>
> Verzeichnis-Buckets, die über einen Zeitraum von mindestens 90 Tagen keine Anforderungsaktivität aufweisen, gehen in einen inaktiven Status über. Im inaktiven Zustand ist ein Verzeichnis-Bucket vorübergehend für Lese- und Schreibvorgänge nicht zugänglich. Inaktive Buckets behalten alle Speicher-, Objekt- und Bucket-Metadaten bei. Für inaktive Buckets fallen bestehende Speichergebühren an. Wenn Sie eine Zugriffsanfrage für einen inaktiven Bucket stellen, wechselt der Bucket in den aktiven Status, normalerweise innerhalb weniger Minuten. Während dieser Übergangsphase geben Lese- und Schreibvorgänge einen HTTP `503 (Service Unavailable)` Fehlercode zurück.

Die folgenden Themen enthalten Informationen zu Verzeichnis-Buckets. Weitere Informationen über Allzweck-Buckets finden Sie unter Bucket-Übersicht.

Weitere Informationen zu Directory-Buckets finden Sie in den folgenden Themen.

- Namen von Verzeichnis-Buckets
- Verzeichnisse
- Schlüsselnamen
- Zugriffsverwaltung

Namen von Verzeichnis-Buckets

Ein Verzeichnis-Bucket-Name besteht aus einem Basisnamen, den Sie angeben, und einem Suffix, das die ID der Zone (Availability Zone) enthält, in der sich Ihr Bucket befindet. Die Namen von Verzeichnis-Buckets müssen das folgende Format haben und den Benennungsregeln für Verzeichnis-Buckets entsprechen:

```
bucket-base-name--zone-id--x-s3
```

Der folgende Verzeichnis-Bucket-Name enthält beispielsweise die Availability Zone-ID `usw2-az1`:

```
bucket-base-name--usw2-az1--x-s3
```

Weitere Informationen finden Sie unter Regeln für die Benennung von Verzeichnis-Buckets.

Verzeichnisse

Verzeichnis-Buckets organisieren Daten hierarchisch in Verzeichnissen, im Gegensatz zu der flachen Sortierstruktur von Allzweck-Buckets.

Bei einem hierarchischen Namespace ist das Trennzeichen im Objektschlüssel wichtig. Das einzige unterstützte Trennzeichen ist der Schrägstrich (`/`). Verzeichnisse werden durch Trennzeichengrenzen bestimmt. Beispielsweise führt der Objektschlüssel `dir1/dir2/file1.txt` dazu, dass die Verzeichnisse `dir1/`und `dir2/` automatisch erstellt werden und das Objekt `file1.txt` dem `/dir2`-Verzeichnis im Pfad `dir1/dir2/file1.txt` hinzugefügt wird.

Das Indizierungsmodell für Verzeichnis-Buckets gibt unsortierte Ergebnisse für den Vorgang zurück. `ListObjectsV2` API Wenn Sie Ihre Ergebnisse auf einen Unterabschnitt Ihres Buckets

beschränken müssen, können Sie im `prefix`-Parameter einen Unterverzeichnispfad angeben, z. B. `prefix=dir1/`.

Schlüsselnamen

Bei Verzeichnis-Buckets werden Unterverzeichnisse, die mehreren Objektschlüsseln gemeinsam sind, mit dem ersten Objektschlüssel erstellt. Zusätzliche Objektschlüssel für dasselbe Unterverzeichnis verwenden das zuvor erstellte Unterverzeichnis. Dieses Modell bietet Ihnen Flexibilität bei der Auswahl von Objektschlüsseln, die für die Anwendung am besten geeignet sind, und unterstützt sowohl dünn besetzte als auch dichte Verzeichnisse.

Zugriffsverwaltung

Bei Verzeichnis-Buckets sind alle Einstellungen für das Blockieren des öffentlichen Zugriffs auf Bucket-Ebene standardmäßig aktiviert. S3 Object Ownership ist auf Bucket Owner Enforced gesetzt und die Zugriffskontrolllisten (ACLs) sind deaktiviert. Diese Einstellungen können nicht geändert werden.

Standardmäßig haben Benutzer keine Berechtigungen für Verzeichnis-Buckets. Um Zugriffsberechtigungen für Verzeichnis-Buckets IAM zu gewähren, können Sie Benutzer, Gruppen oder Rollen erstellen und diesen Identitäten Berechtigungen zuweisen. Weitere Informationen finden Sie unter Autorisieren eines regionalen Endpunkts APIs mit IAM.

Verzeichnis-Buckets erstellen und verwenden

Weitere Informationen zum Arbeiten mit Verzeichnis-Buckets finden Sie in den folgenden Themen.

- Anwendungsfälle für Directory-Buckets
- Unterschiede bei Directory-Buckets
- Netzwerke für Verzeichnis-Buckets
- Regeln für die Benennung von Verzeichnis-Buckets
- Anzeigen von Verzeichnis-Bucket-Eigenschaften
- Verwaltung von Directory-Bucket-Richtlinien
- Leeren eines Verzeichnis-Buckets
- Löschen eines Verzeichnis-Buckets
- Auflisten von Verzeichnis-Buckets

- Feststellen, ob Sie auf einen Verzeichnis-Bucket zugreifen können
- Arbeiten mit Objekten in einem Verzeichnis-Bucket
- Sicherheit für Verzeichnis-Buckets
- Optimierung der Leistung von Directory-Buckets
- Entwickeln mit Verzeichnis-Buckets

Anwendungsfälle für Directory-Buckets

Verzeichnis-Buckets unterstützen die Bucket-Erstellung im folgenden Bucket-Standorttyp: Availability Zone.

Für Anwendungsfälle mit geringer Latenz können Sie einen Verzeichnis-Bucket in einer einzelnen Availability Zone erstellen, um Daten zu speichern. Verzeichnis-Buckets in Availability Zones unterstützen die Speicherklasse S3 Express One Zone. Die Speicherklasse S3 Express One Zone wird empfohlen, wenn Ihre Anwendung leistungsempfindlich ist und von einstelligen Millisekunden `PUT` und Latenzen profitiert. `GET` Weitere Informationen zum Erstellen von Verzeichnis-Buckets in Availability Zones finden Sie unter. Leistungsstarke Workloads

Themen

- Leistungsstarke Workloads

Leistungsstarke Workloads

S3 Express One Zone

Sie können Amazon S3 Express One Zone für Hochleistungs-Workloads verwenden. S3 Express One Zone ist die erste S3-Speicherklasse, bei der Sie eine einzelne Availability Zone mit der Option auswählen können, Ihren Objektspeicher zusammen mit Ihren Rechenressourcen zu platzieren, was die höchstmögliche Zugriffsgeschwindigkeit bietet. Objekte in S3 Express One Zone werden in Verzeichnis-Buckets gespeichert, die sich in Availability Zones befinden. Weitere Informationen zu Verzeichnis-Buckets finden Sie unter Directory-Buckets.

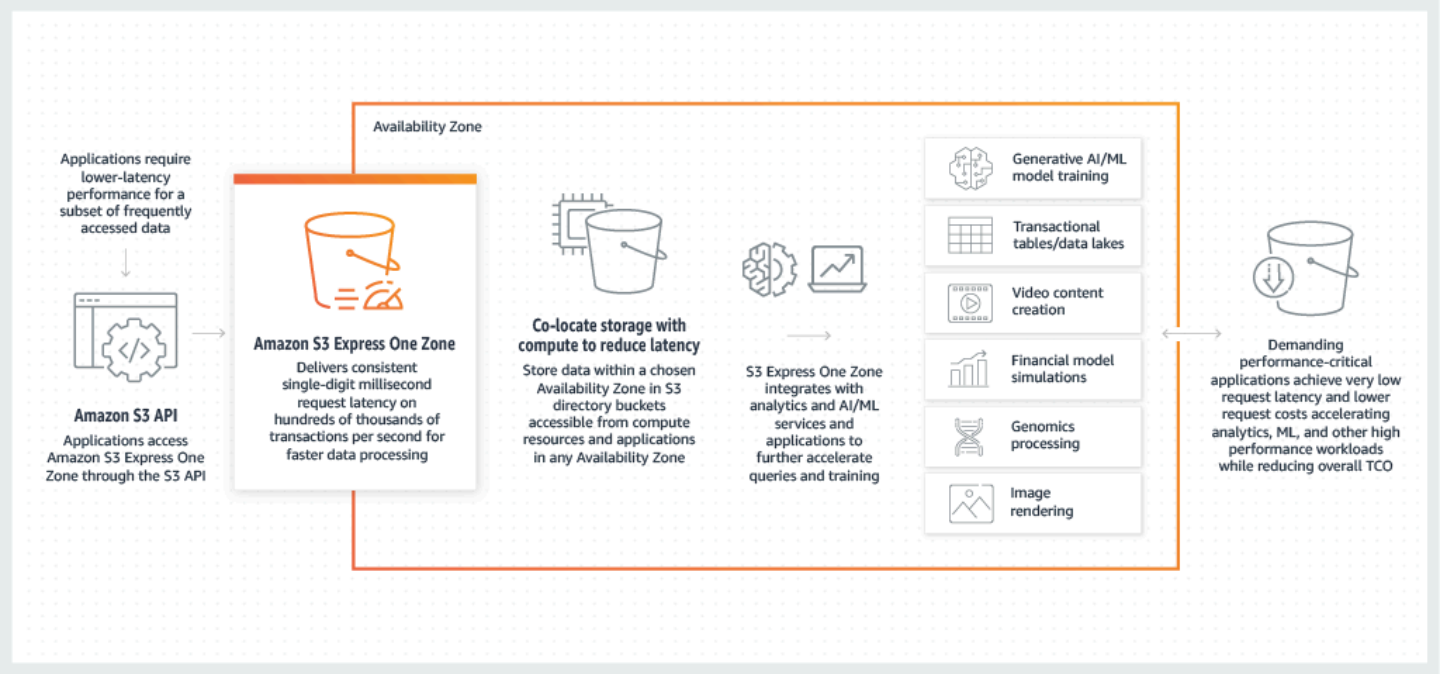
Amazon S3 Express One Zone ist eine leistungsstarke Amazon-S3-Speicherklasse mit einer einzelnen Zone, die speziell für den konsistenten Datenzugriff im einstelligen Millisekundenbereich für Ihre latenzempfindlichsten Anwendungen entwickelt wurde. S3 Express One Zone ist die derzeit verfügbare Cloud-Objektspeicherklasse mit der niedrigsten Latenz. Sie bietet

Datenzugriffsgeschwindigkeiten, die bis zu zehnmal schneller sind und die Anforderungskosten um 50 Prozent niedriger sind als bei S3 Standard. Anwendungen können sofort davon profitieren, dass Anfragen bis zu einer Größenordnung schneller bearbeitet werden. S3 Express One Zone bietet eine ähnliche Leistungselastizität wie andere S3-Speicherklassen. S3 Express One Zone wird für Workloads oder leistungskritische Anwendungen verwendet, die eine konsistente Latenz im einstelligen Millisekundenbereich erfordern.

Wie bei anderen Amazon S3 S3-Speicherklassen müssen Sie die Kapazitäts- oder Durchsatzanforderungen nicht im Voraus planen oder bereitstellen. Sie können Ihren Speicher je nach Bedarf nach oben oder unten skalieren und über Amazon S3 auf Ihre Daten zugreifenAPI.

Die Speicherklasse Amazon S3 Express One Zone ist für eine Verfügbarkeit von 99,95 Prozent innerhalb einer einzigen Availability Zone konzipiert und wird durch das Amazon S3 Service Level Agreement unterstützt. Mit S3 Express One Zone werden Ihre Daten redundant auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert. S3 Express One Zone wurde entwickelt, um mit gleichzeitigen Geräteausfällen umzugehen, indem verlorene Redundanz schnell erkannt und behoben werden kann. Wenn das vorhandene Gerät ausfällt, leitet S3 Express One Zone Anfragen automatisch an neue Geräte innerhalb einer Availability Zone weiter. Diese Redundanz trägt dazu bei, den unterbrechungsfreien Zugriff auf Ihre Daten innerhalb einer Availability Zone sicherzustellen.

S3 Express One Zone ist ideal für jede Anwendung geeignet, bei der es wichtig ist, die für den Zugriff auf ein Objekt erforderliche Latenz zu minimieren. Bei solchen Anwendungen kann es sich um interaktive Workflows handeln, z. B. bei der Videobearbeitung, bei der Kreative über ihre Benutzeroberflächen reaktionsschnellen Zugriff auf Inhalte benötigen. S3 Express One Zone bietet auch Vorteile für Analysen und maschinelles Lernen, die ähnliche Anforderungen an die Reaktionsfähigkeit ihrer Daten stellen, insbesondere Workloads mit vielen kleineren Zugriffen oder einer großen Anzahl zufälliger Zugriffe. S3 Express One Zone kann zusammen mit anderen AWS-Services zur Unterstützung von Workloads in den Bereichen Analytik, künstliche Intelligenz und maschinelles Lernen (KI/ML) wie AmazonEMR, Amazon und Amazon Athena SageMaker verwendet werden.



Bei den Verzeichnis-Buckets, die die Speicherklasse S3 Express One Zone verwenden, werden Daten auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert, Daten werden jedoch nicht redundant in Availability Zones gespeichert. Wenn Sie einen Directory-Bucket für die Verwendung der S3 Express One Zone-Speicherklasse erstellen, empfehlen wir Ihnen, eine Availability Zone AWS-Region und eine Availability Zone anzugeben, die lokal für Ihre Amazon-EC2, Amazon Elastic Kubernetes Service- oder Amazon Elastic Container Service (AmazonECS) Compute-Instances sind, um die Leistung zu optimieren.

Wenn Sie S3 Express One Zone verwenden, können Sie mit Ihrem Directory-Bucket in einer virtuellen privaten Cloud (VPC) interagieren, indem Sie einen VPC Gateway-Endpunkt verwenden. Mit einem Gateway-Endpunkt können Sie von Ihrem Computer aus auf S3 Express One Zone-Verzeichnis-Buckets zugreifen, VPC ohne dass Sie ein Internet-Gateway oder ein NAT Gerät VPC benötigen, und das ohne zusätzliche Kosten.

Sie können viele der gleichen Amazon S3 API S3-Operationen und -Funktionen mit Verzeichnis-Buckets verwenden, die Sie mit Allzweck-Buckets und anderen Speicherklassen verwenden. Dazu gehören Mountpoint für Amazon S3, serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3), serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-KMS), S3 Batch Operations und S3 Block Public Access. Sie können auf S3 Express One Zone zugreifen, indem Sie die Amazon S3-Konsole AWS Command Line Interface (AWS CLI) AWS SDKs, und Amazon S3 verwenden RESTAPI.

Weitere Informationen zu S3 Express One Zone finden Sie in den folgenden Themen.

- Übersicht
- Funktionen von S3 Express One Zone
- Zugehörige Services
- Nächste Schritte

Übersicht

Um die Leistung zu optimieren und die Latenz zu reduzieren, führt S3 Express One Zone die folgenden neuen Konzepte ein.

Availability Zones

Die Speicherklasse Amazon S3 Express One Zone ist für eine Verfügbarkeit von 99,95 Prozent innerhalb einer einzigen Availability Zone konzipiert und wird durch das Amazon S3 Service Level Agreement unterstützt. Mit S3 Express One Zone werden Ihre Daten redundant auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert. S3 Express One Zone wurde entwickelt, um mit gleichzeitigen Geräteausfällen umzugehen, indem verlorene Redundanz schnell erkannt und behoben werden kann. Wenn das vorhandene Gerät ausfällt, leitet S3 Express One Zone Anfragen automatisch an neue Geräte innerhalb einer Availability Zone weiter. Diese Redundanz trägt dazu bei, den unterbrechungsfreien Zugriff auf Ihre Daten innerhalb einer Availability Zone sicherzustellen.

Eine Availability Zone ist eines oder mehrere diskrete Rechenzentren mit redundanter Stromversorgung, Vernetzung und Konnektivität in einem AWS-Region. Wenn Sie einen Verzeichnis-Bucket erstellen, wählen Sie die Availability Zone und den AWS-Region Speicherort Ihres Buckets aus.

Einzelne Availability Zone

Wenn Sie einen Verzeichnis-Bucket erstellen, wählen Sie die Availability Zone und die AWS-Region.

Verzeichnis-Buckets verwenden die Speicherklasse S3 Express One Zone, die für die Verwendung durch leistungssensitive Anwendungen konzipiert wurde. S3 Express One Zone ist die erste S3-Speicherklasse, bei der Sie eine einzelne Availability Zone mit der Option auswählen können, Ihren Objektspeicher gemeinsam mit Ihren Computingressourcen zu platzieren, was die höchstmögliche Zugriffsgeschwindigkeit bietet.

Mit S3 Express One Zone werden Ihre Daten redundant auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert. S3 Express One Zone ist für eine Verfügbarkeit von 99,95

Prozent innerhalb einer einzigen Availability Zone konzipiert und wird durch das Amazon S3 Service Level Agreement unterstützt. Weitere Informationen finden Sie unter Availability Zones

Endpunkte und Gateway-Endpunkte VPC

APIBucket-Management-Operationen für Directory-Buckets sind über einen regionalen Endpunkt verfügbar und werden als regionale Endpunktoperationen bezeichnet. API Beispiele für regionale API Endpunktoperationen sind und. `CreateBucket` `DeleteBucket` Nachdem Sie einen Verzeichnis-Bucket erstellt haben, können Sie Zonal Endpoint API Operations verwenden, um die Objekte in Ihrem Verzeichnis-Bucket hochzuladen und zu verwalten. Zonale API Endpunktoperationen sind über einen zonalen Endpunkt verfügbar. Beispiele für API Operationen an zonalen Endpunkten sind und. `PutObject` `CopyObject`

Sie können über VPC Gateway-Endpunkte VPC von Ihrer aus auf S3 Express One Zone zugreifen. Nachdem Sie einen Gateway-Endpunkt erstellt haben, können Sie ihn als Ziel in Ihrer Routentabelle für den Verkehr hinzufügen, der von Ihrer VPC zur S3 Express One Zone bestimmt ist. Wie bei Amazon S3 fallen für die Nutzung von Gateway-Endpunkten keine zusätzlichen Gebühren an. Weitere Informationen zur Konfiguration von VPC Gateway-Endpunkten finden Sie unter Netzwerke für Verzeichnis-Buckets

Sitzungsbasierte Autorisierung

Mit S3 Express One Zone authentifizieren und autorisieren Sie Anfragen über einen neuen sitzungsbasierten Mechanismus, der für die geringste Latenz optimiert ist. Sie können mit `CreateSession` temporäre Anmeldeinformationen anfordern, die den Zugriff auf Ihren Bucket mit geringer Latenz ermöglichen. Diese temporären Anmeldeinformationen sind einem bestimmten S3-Verzeichnis-Bucket zugeordnet. Sitzungstoken werden nur bei zonalen Vorgängen (auf Objektebene) verwendet (mit Ausnahme von CopyObject). Weitere Informationen finden Sie unterAutorisieren von zonalen API Endpunktoperationen mit `CreateSession`.

Die AWS SDKsfür S3 Express One Zone unterstützte Zone kümmert sich in Ihrem Namen um die Einrichtung und Aktualisierung von Sitzungen. Um Ihre Sitzungen zu schützen, laufen temporäre Sicherheitsanmeldedaten nach 5 Minuten ab. Nachdem Sie die erforderlichen Berechtigungen heruntergeladen und installiert AWS SDKs und die erforderlichen AWS Identity and Access Management (IAM) Berechtigungen konfiguriert haben, können Sie sofort mit der Nutzung von API Operations beginnen.

Funktionen von S3 Express One Zone

Die folgenden S3-Funktionen sind für S3 Express One Zone verfügbar. Eine vollständige Liste der unterstützten API Operationen und nicht unterstützten Funktionen finden Sie unter. Unterschiede bei Directory-Buckets

Zugriffsverwaltung und Sicherheit

Sie können die folgenden Funktionen verwenden, um den Zugriff zu prüfen und zu verwalten. Standardmäßig sind Verzeichnis-Buckets privat und sind nur für Benutzer zugänglich, denen explizit Zugriff gewährt wurde. Im Gegensatz zu Allzweck-Buckets, bei denen die Zugriffskontrollgrenze auf Bucket-, Präfix- oder Objekt-Tag-Ebene festgelegt werden kann, wird die Zugriffskontrollgrenze für Verzeichnis-Buckets nur auf Bucket-Ebene festgelegt. Weitere Informationen finden Sie unter Autorisieren eines regionalen Endpunkts APIs mit IAM.

- S3 Block Public Access — Alle S3 Block Public Access-Einstellungen sind standardmäßig auf Bucket-Ebene aktiviert. Diese Standardeinstellung kann nicht geändert werden.
- S3 Object Ownership (Bucket-Besitzer wird standardmäßig erzwungen) — Zugriffskontrolllisten (ACLs) werden für Verzeichnis-Buckets nicht unterstützt. Directory-Buckets verwenden automatisch die vom Bucket-Besitzer erzwungene Einstellung für S3-Objekteigentum. Bucket Owner erzwungen bedeutet, dass diese deaktiviert ACLs sind. Der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. Diese Standardeinstellung kann nicht geändert werden.
- AWS Identity and Access Management (IAM) — IAM hilft Ihnen dabei, den Zugriff auf Ihre Verzeichnis-Buckets sicher zu kontrollieren. Mit dieser Aktion IAM können Sie Zugriff auf API Bucket-Management-Operationen (Regional) und API Objektmanagement-Operationen (Zonal) gewähren. `s3express:CreateSession` Weitere Informationen finden Sie unter Autorisieren eines regionalen Endpunkts APIs mit IAM. Im Gegensatz zu Objektverwaltungsaktionen können Bucket-Verwaltungsaktionen nicht kontoübergreifend sein. Nur der Bucket-Eigentümer kann diese Aktionen ausführen.
- Bucket-Richtlinien — Verwenden Sie eine IAM basierte Richtliniensprache, um ressourcenbasierte Berechtigungen für Ihre Verzeichnis-Buckets zu konfigurieren. Sie können sie auch verwendenIAM, um den Zugriff auf den `CreateSession` API Vorgang zu steuern, sodass Sie die zonalen Operationen oder die Objektverwaltung verwenden können. API Sie können über dasselbe Konto oder kontoübergreifenden Zugriff auf zonale Operationen gewähren. API Weitere Informationen zu den Berechtigungen und Richtlinien für S3 Express One Zone finden Sie unter. Autorisieren eines regionalen Endpunkts APIs mit IAM

- IAMAccess Analyzer for S3 — Evaluieren und überwachen Sie Ihre Zugriffsrichtlinien, um sicherzustellen, dass die Richtlinien nur den beabsichtigten Zugriff auf Ihre S3-Ressourcen ermöglichen.

Protokollierung und Überwachung

S3 Express One Zone verwendet die folgenden S3-Protokollierungs- und Überwachungstools, mit denen Sie die Nutzung Ihrer Ressourcen überwachen und kontrollieren können:

- CloudWatch Amazon-Metriken — Überwachen Sie Ihre AWS Ressourcen und Anwendungen CloudWatch , indem Sie Metriken sammeln und verfolgen. S3 Express One Zone verwendet denselben CloudWatch Namespace wie andere Amazon S3 S3-Speicherklassen (`AWS/S3`) und unterstützt tägliche Speichermetriken für Verzeichnis-Buckets: `BucketSizeBytes` und. `NumberOfObjects` Weitere Informationen finden Sie unter Metriken mit Amazon überwachen CloudWatch.
- AWS CloudTrail Logs — AWS CloudTrail ist ein Programm AWS-Service , das Sie bei der Durchführung von Betriebs- und Risikoprüfungen, Governance und Compliance unterstützt, AWS-Konto indem es die von einem Benutzer, einer Rolle oder einem ausgeführten Aktionen aufzeichnet. AWS-Service CloudTrail Erfasst für S3 Express One Zone regionale API Endpunktoperationen (zum Beispiel `CreateBucket` und`PutBucketPolicy`) als Verwaltungsereignisse und zonale API Operationen (zum Beispiel `GetObject` und`PutObject`) als Datenereignisse. Zu diesen Ereignissen gehören Maßnahmen AWS Management Console, die in den AWS API Vorgängen, AWS Command Line Interface (AWS CLI) AWS SDKs, und ergriffen wurden. Weitere Informationen finden Sie unter Protokollierung mit AWS CloudTrail für S3 Express One Zone.

> **Note**
>
> Amazon S3 S3-Serverzugriffsprotokolle werden mit S3 Express One Zone nicht unterstützt.

Verwaltung von Objekten

Sie können Ihren Objektspeicher mithilfe der Amazon S3 S3-Konsole AWS SDKs, und verwalten AWS CLI. Die folgenden Funktionen sind für die Objektverwaltung mit S3 Express One Zone verfügbar:

- S3-Stapeloperationen — Verwenden Sie Batch-Operationen, um Massenoperationen an Objekten in Verzeichnis-Buckets durchzuführen, z. B. die Funktionen Kopieren und Aufrufen AWS Lambda . Sie können mit Batch Operations beispielsweise Objekte zwischen Verzeichnis-Buckets und Allzweck-Buckets kopieren. Mit Batch Operations können Sie Milliarden von Objekten in großem Umfang mit einer einzigen S3-Anfrage verwalten, indem Sie das AWS SDKs oder AWS CLI oder einige Mausklicks in der Amazon S3 S3-Konsole verwenden.
- Import – Nachdem Sie einen Verzeichnis-Bucket erstellt haben, können Sie Ihren Bucket mithilfe der Importfunktion in der Amazon-S3-Konsole mit Objekten füllen. Import ist eine optimierte Methode zur Erstellung von Batch-Operations-Aufträgen zum Kopieren von Objekten aus Allzweck-Buckets in Verzeichnis-Buckets.

AWS SDKsund Client-Bibliotheken

Sie können Ihren Objektspeicher mithilfe der Clientbibliotheken AWS SDKs und verwalten.

- Mountpoint for Amazon S3 — Mountpoint for Amazon S3 ist ein Open-Source-Dateiclient, der Zugriff mit hohem Durchsatz ermöglicht und so die Rechenkosten für Data Lakes auf Amazon S3 senkt. Mountpoint für Amazon S3 übersetzt lokale Dateisystem-Aufrufe in API API S3-Objekt-Aufrufe wie und. `GET` `LIST` Es ist ideal für leseintensive Data Lake-Workloads, die Petabyte an Daten verarbeiten und den von Amazon S3 bereitgestellten hohen elastischen Durchsatz benötigen, um über Tausende von Instances nach oben und unten zu skalieren.
- S3A – S3A ist ein empfohlener Hadoop-kompatible Schnittstelle für den Zugriff auf Datenspeicher in Amazon S3. S3A ersetzt die S3N Hadoop Dateisystem-Client.
- PyTorch on AWS — PyTorch on AWS ist ein Open-Source-Deep-Learning-Framework, das die Entwicklung von Modellen für maschinelles Lernen und deren Einsatz in der Produktion erleichtert.
- AWS SDKs— Sie können das verwenden AWS SDKs, wenn Sie Anwendungen mit Amazon S3 entwickeln. Sie AWS SDKs vereinfachen Ihre Programmieraufgaben, indem Sie das zugrunde liegende Amazon S3 einschließen RESTAPI. Weitere Informationen zur Verwendung von Express One Zone AWS SDKs mit S3 finden Sie unterthe section called “AWS SDKs”.

Verschlüsselung und Datenschutz

Objekte in S3 Express One Zone werden automatisch durch serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) verschlüsselt. S3 Express One Zone unterstützt auch serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-). KMS S3 Express One Zone unterstützt keine serverseitige Verschlüsselung mit vom Kunden

bereitgestellten Verschlüsselungsschlüsseln (SSE-C) oder serverseitige Dual-Layer-Verschlüsselung mit (-). AWS KMS keys DSSE KMS Weitere Informationen finden Sie unter Datenschutz und Verschlüsselung.

S3 Express One Zone bietet Ihnen die Möglichkeit, den Prüfsummenalgorithmus auszuwählen, der zur Validierung Ihrer Daten beim Hoch- oder Herunterladen verwendet wird. Sie können einen der folgenden Algorithmen zur Prüfung der Datenintegrität zwischen Secure Hash Algorithms (SHA) oder Cyclic Redundancy Check (CRC) wählen:, -1 und -256. CRC32 CRC32C SHA SHA MD5basierte Prüfsummen werden von der S3 Express One Zone-Speicherklasse nicht unterstützt.

Weitere Informationen finden Sie unter Bewährte Methoden für zusätzliche S3-Prüfsummen.

AWS Signatur Version 4 (SigV4)

S3 Express One Zone verwendet AWS Signaturversion 4 (SigV4). SigV4 ist ein Signaturprotokoll, das zur Authentifizierung von Anfragen an Amazon S3 verwendet wird. HTTPS S3 Express One Zone signiert Anfragen mithilfe von AWS Sigv4. Weitere Informationen finden Sie unter Authentifizieren von Anfragen (AWS Signature Version 4) in der Amazon Simple Storage Service API Reference.

Starke Konsistenz

S3 Express One Zone bietet eine hohe read-after-write Konsistenz für alle Objekte `PUT` und `DELETE` Anfragen von Objekten in Ihren Verzeichnis-Buckets. AWS-Regionen Weitere Informationen finden Sie unter Amazon S3-Datenkonsistenzmodell.

Zugehörige Services

Sie können Folgendes AWS-Services mit der S3 Express One Zone-Speicherklasse verwenden, um Ihren spezifischen Anwendungsfall mit niedriger Latenz zu unterstützen.

- Amazon Elastic Compute Cloud (AmazonEC2) — Amazon EC2 bietet sichere und skalierbare Rechenkapazität in der AWS Cloud. Durch die Verwendung von Amazon EC2 müssen Sie nicht im Voraus in Hardware investieren, sodass Sie Anwendungen schneller entwickeln und bereitstellen können. Sie können Amazon verwendenEC2, um so viele oder so wenige virtuelle Server wie nötig zu starten, Sicherheit und Netzwerke zu konfigurieren und Speicher zu verwalten.
- AWS Lambda – Lambda ist ein Computingservice, mit dem Sie Code ausführen können, ohne Server bereitstellen oder verwalten zu müssen. Sie konfigurieren Benachrichtigungseinstellungen für einen Bucket und erteilen die Amazon-S3-Berechtigung zum Aufrufen einer Funktion in der ressourcenbasierten Berechtigungsrichtlinie der Funktion.

- Amazon Elastic Kubernetes Service (AmazonEKS) — Amazon EKS ist ein verwalteter Service, der die Installation, den Betrieb und die Wartung Ihres eigenen Services überflüssig macht Kubernetes Kontrollebene aktiviert. AWSKubernetesist ein Open-Source-System, das die Verwaltung, Skalierung und Bereitstellung von containerisierten Anwendungen automatisiert.
- Amazon Elastic Container Service (AmazonECS) — Amazon ECS ist ein vollständig verwalteter Container-Orchestrierungsservice, mit dem Sie containerisierte Anwendungen einfach bereitstellen, verwalten und skalieren können.
- AWS Key Management Service (AWS KMS) — AWS Der Key Management Service (AWS KMS) ist ein AWS verwalteter Service, mit dem Sie die Verschlüsselungsschlüssel, die zur Verschlüsselung Ihrer Daten verwendet werden, auf einfache Weise erstellen und kontrollieren können. Die AWS KMS Schlüssel, in denen Sie erstellen, AWS KMS werden durch FIPS 140-2 validierte Hardware-Sicherheitsmodule () geschützt. HSM Um Ihre KMS Schlüssel zu verwenden oder zu verwalten, interagieren Sie mit. AWS KMS
- Amazon Athena — Athena ist ein interaktiver Abfrageservice, der es einfach macht, Daten mithilfe von Standards direkt in Amazon S3 zu analysieren. SQL Sie können Athena auch verwenden, um Datenanalysen interaktiv auszuführen, indem Sie Apache Spark ohne Ressourcen planen, konfigurieren oder verwalten zu müssen. Wenn du rennst Apache Spark Bewerbungen auf Athena reichen Sie ein Spark Code zur Verarbeitung und direkt die Ergebnisse erhalten.
- Amazon SageMaker Training — Informieren Sie sich über die Optionen für Trainingsmodelle bei AmazonSageMaker, einschließlich integrierter Algorithmen, benutzerdefinierter Algorithmen, Bibliotheken und Modelle aus dem AWS Marketplace.
- AWS Glue— AWS Glue ist ein serverloser Datenintegrationsservice, der es Analytics-Benutzern leicht macht, Daten aus verschiedenen Quellen zu finden, aufzubereiten, zu verschieben und zu integrieren. Sie können ihn AWS Glue für Analysen, maschinelles Lernen und Anwendungsentwicklung verwenden. AWS Glue umfasst auch zusätzliche Produktivitäts- und Datenmanagement-Tools für die Erstellung, Ausführung von Jobs und Implementierung von Geschäftsabläufen.
- Amazon EMR — Amazon EMR ist eine verwaltete Cluster-Plattform, die den Betrieb von Big-Data-Frameworks vereinfacht, wie z. B. Apache Hadoop and Apache Spark, AWS um riesige Datenmengen zu verarbeiten und zu analysieren.
- AWS CloudTrail— AWS CloudTrail ist ein AWS Service, der Sie bei der Betriebs- und Risikoprüfung, Verwaltung und Einhaltung von Vorschriften für Ihr AWS Konto unterstützt. Von einem Benutzer, einer Rolle oder einem AWS Dienst ausgeführte Aktionen werden als Ereignisse in aufgezeichnet CloudTrail. Zu den Ereignissen gehören Aktionen, die in der AWS

Managementkonsole, der AWS Befehlszeilenschnittstelle und AWS SDKs und ausgeführt wurdenAPIs.

- AWS CloudFormation — ist ein Service, der Sie bei der Modellierung und Einrichtung Ihrer AWS Ressourcen unterstützt, sodass Sie weniger Zeit mit der Verwaltung dieser Ressourcen verbringen und sich mehr auf Ihre Anwendungen konzentrieren können, die in ausgeführt AWS werden. Sie erstellen eine Vorlage, die alle gewünschten AWS Ressourcen beschreibt (wie EC2 Amazon-Instances oder RDS Amazon-DB-Instances), und CloudFormation kümmert sich um die Bereitstellung und Konfiguration dieser Ressourcen für Sie. Sie müssen AWS Ressourcen nicht einzeln erstellen und konfigurieren und herausfinden, was wovon abhängt; CloudFormation kümmert sich darum.

Nächste Schritte

Weitere Informationen zum Arbeiten mit der Speicherklasse S3 Express One Zone und Verzeichnis-Buckets finden Sie in den folgenden Themen:

- Tutorial: Erste Schritte mit S3 Express One Zone
- Availability Zones und Regionen bei S3 Express One Zone
- Netzwerke für Verzeichnis-Buckets in einer Availability Zone
- Verzeichnis-Buckets in einer Availability Zone erstellen
- Regionale und zonale Endpunkte für Verzeichnis-Buckets in einer Availability Zone
- Optimieren der Leistung von S3 Express One Zone

Tutorial: Erste Schritte mit S3 Express One Zone

Amazon S3 Express One Zone ist die erste S3-Speicherklasse, bei der Sie eine einzelne Availability Zone mit der Option auswählen können, Ihren Objektspeicher zusammen mit Ihren Rechenressourcen zu platzieren, was die höchstmögliche Zugriffsgeschwindigkeit bietet. Daten in S3 Express One Zone werden in Verzeichnis-Buckets gespeichert, die sich in Availability Zones befinden. Weitere Informationen zu Verzeichnis-Buckets finden Sie unter Directory-Buckets.

S3 Express One Zone ist ideal für alle Anwendungen, bei denen es wichtig ist, die Latenz von Anfragen zu minimieren. Bei solchen Anwendungen kann es sich um Workflows handeln, bei denen Menschen interaktiv sind, z. B. bei der Videobearbeitung, bei denen Kreativprofis einen reaktionsschnellen Zugriff auf Inhalte über ihre Benutzeroberflächen benötigen. S3 Express One Zone bietet auch Vorteile für Analysen und maschinelles Lernen, die ähnliche Anforderungen an

die Reaktionsfähigkeit ihrer Daten stellen, insbesondere Workloads mit vielen kleineren Zugriffen oder einer großen Anzahl zufälliger Zugriffe. S3 Express One Zone kann mit anderen AWS Diensten wie AmazonEMR, Amazon Athena, AWS Glue Data Catalog und Amazon SageMaker Model Training verwendet werden, um Workloads für Analysen, künstliche Intelligenz und maschinelles Lernen (AI/ML) zu unterstützen. Sie können mit der Speicherklasse und den Verzeichnis-Buckets von S3 Express One Zone arbeiten, indem Sie die Amazon S3-Konsole AWS SDKs, die AWS Befehlszeilenschnittstelle (AWS CLI) und Amazon S3 REST API verwenden. Weitere Informationen finden Sie unter Was ist S3 Express One Zone? und Wie unterscheidet sich S3 Express One Zone? . Dies ist ein Workflow-Diagramm für S3 Express One Zone.

Ziel

In diesem Tutorial erfahren Sie, wie Sie einen Gateway-Endpunkt erstellen, eine IAM Richtlinie erstellen und anhängen, einen Verzeichnis-Bucket erstellen und dann mithilfe der Import-Aktion Ihren Verzeichnis-Bucket mit Objekten füllen, die derzeit in Ihrem Allzweck-Bucket gespeichert sind. Alternativ können Sie Objekte manuell in Ihren Verzeichnis-Bucket hochladen.

Themen

- Voraussetzungen
- Schritt 1: Konfigurieren Sie einen VPC Gateway-Endpunkt
- Schritt 2: Erstellen Sie einen Verzeichnis-Bucket
- Schritt 3: Daten in einen Verzeichnis-Bucket importieren
- Schritt 4: Laden Sie Objekte manuell in Ihren Verzeichnis-Bucket hoch
- Schritt 5: Leeren Sie Ihren Verzeichnis-Bucket
- Schritt 6: Löschen Sie Ihren Verzeichnis-Bucket
- Nächste Schritte

Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, benötigen Sie einen AWS-Konto , bei dem Sie sich als AWS Identity and Access Management (IAM) -Benutzer mit den richtigen Berechtigungen anmelden können.

Teilschritte

- Erstellen Sie ein AWS-Konto
- Erstellen Sie einen IAM Benutzer in Ihrer AWS-Konto (Konsole)

- Erstellen Sie eine IAM Richtlinie und fügen Sie sie einem IAM Benutzer oder einer Rolle (Konsole) hinzu

Erstellen Sie ein AWS-Konto

Um dieses Tutorial abzuschließen, benötigen Sie eine AWS-Konto. Wenn Sie sich für registrieren AWS, AWS-Konto ist Ihr automatisch für alle Services in angemeldet AWS, einschließlich Amazon S3. Berechnet werden Ihnen aber nur die Services, die Sie nutzen. Weitere Informationen zur Preisgestaltung finden Sie unter S3-Preise.

Erstellen Sie einen IAM Benutzer in Ihrer AWS-Konto (Konsole)

AWS Identity and Access Management (IAM) hilft Administratoren dabei AWS-Service , den Zugriff auf AWS Ressourcen sicher zu kontrollieren. IAMAdministratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um auf Objekte zuzugreifen und Verzeichnis-Buckets in S3 Express One Zone zu verwenden. Sie können es ohne IAM zusätzliche Kosten nutzen.

Standardmäßig sind Benutzer nicht berechtigt, auf Verzeichnis-Buckets zuzugreifen und S3 Express One Zone-Operationen auszuführen. Um Zugriffsberechtigungen für Verzeichnis-Buckets und S3 Express One Zone-Operationen IAM zu gewähren, können Sie Benutzer oder Rollen erstellen und diesen Identitäten Berechtigungen zuweisen. Weitere Informationen zum Erstellen eines IAM Benutzers finden Sie unter IAMBenutzer erstellen (Konsole) im IAMBenutzerhandbuch. Weitere Informationen zum Erstellen einer IAM Rolle finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen IAM Benutzer im IAMBenutzerhandbuch.

Der Einfachheit halber wird in diesem Tutorial ein IAM Benutzer erstellt und verwendet. Denken Sie nach Abschluss dieses Tutorials an Löschen Sie den IAM Benutzer. Für den produktiven Einsatz empfehlen wir, dass Sie die bewährten Sicherheitsmethoden IAM im IAM Benutzerhandbuch befolgen. Eine bewährte Methode ist, dass menschliche Benutzer den Verbund mit einem Identitätsanbieter verwenden, um mit temporären Anmeldeinformationen auf AWS zuzugreifen. Eine weitere bewährte Methode besteht darin, Workloads die Verwendung temporärer Anmeldeinformationen mit IAM Rollen für den Zugriff AWS vorzuschreiben. Weitere Informationen AWS IAM Identity Center zur Erstellung von Benutzern mit temporären Anmeldeinformationen finden Sie unter Erste Schritte im AWS IAM Identity Center Benutzerhandbuch.

> **Warning**
>
> IAMBenutzer verfügen über langfristige Anmeldeinformationen, was ein Sicherheitsrisiko darstellt. Um dieses Risiko zu minimieren, empfehlen wir, diesen Benutzern nur die Berechtigungen zu gewähren, die sie für die Ausführung der Aufgabe benötigen, und diese Benutzer zu entfernen, wenn sie nicht mehr benötigt werden.

Erstellen Sie eine IAM Richtlinie und fügen Sie sie einem IAM Benutzer oder einer Rolle (Konsole) hinzu

Standardmäßig haben Benutzer keine Berechtigungen für Verzeichnis-Buckets und S3-Express-One-Zone-Vorgänge. Um Zugriffsberechtigungen für Verzeichnis-Buckets IAM zu gewähren, können Sie Benutzer, Gruppen oder Rollen erstellen und diesen Identitäten Berechtigungen zuweisen. Verzeichnis-Buckets sind die einzige Ressource, die Sie in Bucket-Richtlinien oder IAM Identitätsrichtlinien für den Zugriff auf S3 Express One Zone aufnehmen können.

Um regionale API Endpunktoperationen (Operationen auf Bucket-Ebene oder auf Kontrollebene) mit S3 Express One Zone zu verwenden, verwenden Sie das IAM Autorisierungsmodell, das kein Sitzungsmanagement beinhaltet. Berechtigungen werden für Aktionen einzeln erteilt. Um zonale API Endpunktoperationen (Operationen auf Objektebene oder Datenebene) `CreateSession`zu verwenden, erstellen und verwalten Sie Sitzungen, die für die Autorisierung von Datenanforderungen mit geringer Latenz optimiert sind. Um ein Sitzungstoken abzurufen und zu verwenden, müssen Sie die `s3express:CreateSession`-Aktion für Ihren Verzeichnis-Bucket in einer identitätsbasierten Richtlinie oder einer Bucket-Richtlinie zulassen. Wenn Sie in der Amazon S3-Konsole, über die AWS Befehlszeilenschnittstelle (AWS CLI) oder mithilfe der auf S3 Express One Zone zugreifen AWS SDKs, erstellt S3 Express One Zone in Ihrem Namen eine Sitzung. Weitere Informationen finden Sie unter `CreateSession`Autorisierung und AWS Identity and Access Management (IAM) für S3 Express One Zone.

Um eine IAM Richtlinie zu erstellen und die Richtlinie einem IAM Benutzer (oder einer Rolle) zuzuordnen

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die IAM Management Console.
2. Wählen Sie im Navigationsbereich Richtlinien.
3. Wählen Sie Create Policy (Richtlinie erstellen) aus.
4. Wählen Sie JSON.

5. Kopieren Sie die nachstehende Richtlinie in das Fenster des Richtlinien-Editors. Bevor Sie Verzeichnis-Buckets erstellen oder S3 Express One Zone verwenden können, müssen Sie Ihrer Rolle oder Ihren Benutzern AWS Identity and Access Management (IAM) die erforderlichen Berechtigungen erteilen. Diese Beispielrichtlinie ermöglicht den Zugriff auf den `CreateSession` API Vorgang (zur Verwendung mit anderen Operationen auf Zonen- oder Objektebene) und alle API Operationen auf regionaler Endpunktebene (Bucket-Ebene). API Diese Richtlinie ermöglicht die Verwendung des `CreateSession` API Vorgangs mit allen Verzeichnis-Buckets, aber die regionalen API Endpunktoperationen sind nur für die Verwendung mit dem angegebenen Verzeichnis-Bucket zulässig. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre eigenen Informationen.

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Sid": "AllowAccessRegionalEndpointAPIs",
             "Effect": "Allow",
             "Action": [
                 "s3express:DeleteBucket",
                 "s3express:DeleteBucketPolicy",
                 "s3express:CreateBucket",
                 "s3express:PutBucketPolicy",
                 "s3express:GetBucketPolicy",
                 "s3express:ListAllMyDirectoryBuckets"
             ],

             "Resource": "arn:aws:s3express:region:account_id:bucket/bucket-base-
name--zone-id--x-s3/*"
         },
         {
             "Sid": "AllowCreateSession",
             "Effect": "Allow",
             "Action": "s3express:CreateSession",
             "Resource": "*"
         }
     ]
 }
```

6. Wählen Sie Weiter.
7. Geben Sie der Richtlinie einen Namen.

> **Note**
>
> Bucket-Tags werden für S3 Express One Zone nicht unterstützt.

8. Wählen Sie Create Policy.
9. Nachdem Sie eine IAM Richtlinie erstellt haben, können Sie sie einem IAM Benutzer zuordnen. Wählen Sie im Navigationsbereich Richtlinien.
10. Geben Sie in der Suchleiste den Namen Ihrer Richtlinie ein.
11. Wählen Sie im Menü Aktionen die Option Anhängen aus.
12. Wählen Sie unter Nach Entitätstyp filtern die Option IAMBenutzer oder Rollen aus.
13. Geben Sie im Suchfeld den Namen des Benutzers oder der Rolle ein, den Sie verwenden möchten.
14. Wählen Sie Richtlinie anfügen aus.

Schritt 1: Konfigurieren Sie einen VPC Gateway-Endpunkt

Sie können über Gateway-Endpunkte der virtuellen privaten Cloud (VPC) sowohl auf zonale als auch auf regionale API Operationen zugreifen. Gateway-Endpunkte können es ermöglichen, dass der Verkehr die S3 Express One Zone erreicht, ohne ein Gateway zu passieren. NAT Wir empfehlen dringend, Gateway-Endpunkte zu verwenden, da sie bei der Arbeit mit S3 Express One Zone den optimalsten Netzwerkpfad bieten. Sie können von Ihrem Computer aus auf S3 Express One Zone-Verzeichnis-Buckets zugreifen, VPC ohne ein Internet-Gateway oder ein NAT Gerät für Sie zu habenVPC, und das ohne zusätzliche Kosten. Gehen Sie wie folgt vor, um einen Gateway-Endpunkt zu konfigurieren, der eine Verbindung zu S3 Express One Zone-Speicherklassenobjekten und Verzeichnis-Buckets herstellt.

Für den Zugriff auf S3 Express One Zone verwenden Sie regionale und zonale Endpunkte, die sich von den standardmäßigen Amazon-S3-Endpunkten unterscheiden. Je nachdem, welchen Amazon S3 API S3-Betrieb Sie verwenden, ist entweder ein zonaler oder ein regionaler Endpunkt erforderlich. Eine vollständige Liste der unterstützten API Operationen nach Endpunkttyp finden Sie unter Von S3 Express One Zone unterstützte API Operationen. Sie müssen über einen Gateway-Endpunkt der virtuellen privaten Cloud (VPC) auf sowohl zonale als auch regionale Endpunkte zugreifen.

Gehen Sie wie folgt vor, um einen Gateway-Endpunkt zu erstellen, der eine Verbindung zu S3 Express One Zone-Speicherklassenobjekten und Verzeichnis-Buckets herstellt.

Um einen VPC Gateway-Endpunkt zu konfigurieren

1. Öffnen Sie die VPC Amazon-Konsole unterhttps://console.aws.amazon.com/vpc/.
2. Wählen Sie im seitlichen Navigationsbereich unter Virtual Private Cloud die Option Endpoints aus.
3. Wählen Sie Endpunkt erstellen aus.
4. Erstellen Sie einen Namen für Ihren Endpunkt.
5. Wählen Sie für Servicekategorie die Option AWS-Services aus.
6. Suchen Sie unter Dienste mit dem Filter Type=Gateway und wählen Sie dann die Optionsschaltfläche neben com.amazonaws aus. ***region***.s3express.
7. Wählen Sie für die VPC, VPC in der der Endpunkt erstellt werden soll.
8. Wählen Sie für Route tables (Routing-Tabellen) die Routing-Tabellen, die von dem Endpunkt verwendet werden sollen. Amazon fügt VPC automatisch eine Route hinzu, die den für den Service bestimmten Verkehr an die Netzwerkschnittstelle des Endpunkts weiterleitet.
9. Wählen Sie für Policy die Option Vollzugriff aus, um alle Operationen aller Principals auf allen Ressourcen über den VPC Endpunkt zuzulassen. Wählen Sie andernfalls Benutzerdefiniert aus, um eine VPC Endpunktrichtlinie anzuhängen, die die Berechtigungen steuert, die Prinzipale haben, um Aktionen an Ressourcen über den VPC Endpunkt auszuführen.
10. Wählen Sie Endpunkt erstellen aus.

Nachdem Sie einen Gateway-Endpunkt erstellt haben, können Sie regionale API Endpunkte und zonale API Endpunkte verwenden, um auf Amazon S3 Express One Zone-Speicherklassenobjekte und Verzeichnis-Buckets zuzugreifen.

Schritt 2: Erstellen Sie einen Verzeichnis-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen der aktuell angezeigten Datei aus AWS-Region. Wählen Sie als Nächstes die Region aus, in der Sie einen Bucket erstellen möchten.

> **Note**
>
> Wählen Sie eine Region in der Nähe aus, um Latenz und Kosten gering zu halten und behördliche Vorschriften zu erfüllen. In einer Region gespeicherte Objekte verbleiben

so lange in der Region, bis sie explizit in eine andere Region verschoben werden. Eine Liste von Amazon S3 finden Sie AWS-Regionen unter AWS-Service Endpoints in der Allgemeine Amazon Web Services-Referenz.

3. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
4. Wählen Sie Create Bucket (Bucket erstellen) aus.

   Anschließend wird die Seite Bucket erstellen geöffnet.
5. Sehen Sie sich unter Allgemeine Konfiguration an, AWS-Region wo Ihr Bucket erstellt wird.
6. Wählen Sie unter Bucket-Typ die Option Verzeichnis aus.

   > **Note**
   >
   > - Wenn Sie eine Region ausgewählt haben, die keine Verzeichnis-Buckets unterstützt, verschwindet die Option Bucket-Typ und der Bucket-Typ ist standardmäßig ein Allzweck-Bucket. Um einen Verzeichnis-Bucket zu erstellen, müssen Sie eine unterstützte Region auswählen. Eine Liste der Regionen, die Directory-Buckets und die Amazon S3 Express One Zone-Speicherklasse unterstützen, finden Sie unterthe section called “Availability Zones und Regionen bei S3 Express One Zone”.
   > - Nach dem Erstellen des Buckets kann der Bucket-Typ nicht mehr geändert werden.

   Wählen Sie für Availability Zone eine Availability Zone aus, die für Ihre Computing-Services lokal ist. Eine Liste der Availability Zones, die Directory-Buckets und die Speicherklasse S3 Express One Zone unterstützen, finden Sie unter. the section called “Availability Zones und Regionen bei S3 Express One Zone”

   > **Note**
   >
   > Die Availability Zone kann nicht geändert werden, nachdem der Bucket erstellt wurde.
7. Aktivieren Sie unter Availability Zone das Kontrollkästchen, um zu bestätigen, dass Ihre Daten im Falle eines Ausfalls der Availability Zone möglicherweise nicht verfügbar sind oder verloren gehen.

> **Important**
>
> Obwohl Verzeichnis-Buckets auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert werden, speichern Verzeichnis-Buckets Daten nicht redundant über Availability Zones hinweg.

8. Geben Sie unter Bucket-Name einen Namen für Ihren Verzeichnis-Bucket ein.

   Die folgenden Benennungsregeln gelten für Verzeichnis-Buckets.

   - Seien Sie innerhalb der ausgewählten Zone (AWS Availability Zone) einzigartig.
   - Der Name muss zwischen 3 (min) und 63 (maximal) Zeichen lang sein, einschließlich des Suffixes.
   - Nur aus Kleinbuchstaben, Zahlen und Bindestrichen bestehen.
   - Muss mit einer Zahl oder einem Buchstaben beginnen und enden.
   - Muss das folgende Suffix enthalten:. `--`*`zone-id`*`--x-s3`
   - Der Bucket-Name darf nicht mit dem Präfix `xn--` beginnen.
   - Der Bucket-Name darf nicht mit dem Präfix `sthree-` beginnen.
   - Der Bucket-Name darf nicht mit dem Präfix `sthree-configurator` beginnen.
   - Der Bucket-Name darf nicht mit dem Präfix `amzn-s3-demo-` beginnen.
   - Bucket-Namen dürfen nicht mit dem Suffix `-s3alias` enden. Dieses Suffix ist für Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter Verwenden eines Alias im Bucket-Stil für Ihren S3-Bucket-Zugriffspunkt.
   - Bucket-Namen dürfen nicht mit dem Suffix `--ol-s3` enden. Dieses Suffix ist für Objekt-Lambda-Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter So verwenden Sie einen Alias im Bucket-Stil für den Object Lambda Access Point Ihres S3-Buckets.
   - Bucket-Namen dürfen nicht mit dem Suffix `.mrap` enden. Dieses Suffix ist für Namen von Access Points mit mehreren Regionen reserviert. Weitere Informationen finden Sie unter Regeln zur Benennung von Amazon S3-Multi-Regions-Zugriffspunkten.

   Dem Basisnamen, den Sie angeben, wenn Sie mit der Konsole einen Verzeichnis-Bucket erstellen, wird automatisch ein Suffix hinzugefügt. Dieses Suffix enthält die ID der Availability Zone, die Sie ausgewählt haben.

Der Name eines einmal erstellter Buckets kann nicht nachträglich geändert werden. Weitere Informationen zur Benennung von Buckets finden Sie unter Allgemeine Benennungsregeln für Buckets.

> **Important**
>
> Nehmen Sie keine vertraulichen Informationen wie Kontonummern in den Bucket-Namen auf. Der Bucket-Name ist in dem Punkt sichtbarURLs, der auf die Objekte im Bucket verweist.

9. Unter Object Ownership ist die Einstellung Bucket Owner erforced automatisch aktiviert und alle Zugriffskontrolllisten (ACLs) sind deaktiviert. Kann für Verzeichnis-Buckets nicht ACLs aktiviert werden.

   ACLsdeaktiviert

   - Bucket Owner erforced (Standard) — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLswirkt sich nicht mehr auf die Zugriffsberechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet ausschließlich Richtlinien, um die Zugriffssteuerung zu definieren.

     Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

10. Unter „Einstellungen für öffentlichen Zugriff blockieren“ für diesen Bucket werden alle Einstellungen für „Öffentlichen Zugriff blockieren“ für Ihren Verzeichnis-Bucket automatisch aktiviert. Diese Einstellungen können für Directory-Buckets nicht geändert werden. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.
11. Unter Serverseitige Verschlüsselungseinstellungen verwenden Verzeichnis-Buckets serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3), um Daten zu verschlüsseln.
12. Wählen Sie Bucket erstellen aus.

    Nachdem Sie den Bucket erstellt haben, können Sie ihm Dateien und Ordner hinzufügen. Weitere Informationen finden Sie unter the section called “Arbeiten mit Objekten in einem Verzeichnis-Bucket”.

Der folgende Schritt zeigt, wie Sie die Import-Aktion in der Amazon S3 S3-Konsole verwenden, um Ihren Verzeichnis-Bucket mit Daten zu füllen.

Schritt 3: Daten in einen Verzeichnis-Bucket importieren

Um diesen Schritt abzuschließen, benötigen Sie einen Allzweck-Bucket, der Objekte enthält und sich im selben Verzeichnis befindet AWS-Region wie Ihr Verzeichnis-Bucket.

Nachdem Sie einen Verzeichnis-Bucket in Amazon S3 erstellt haben, können Sie den neuen Bucket mit Daten füllen, indem Sie die Import-Aktion in der Amazon S3 S3-Konsole verwenden. Import vereinfacht das Kopieren von Daten in Verzeichnis-Buckets, indem Sie ein Präfix oder einen Allzweck-Bucket auswählen können, aus dem Daten importiert werden sollen, ohne alle zu kopierenden Objekte einzeln angeben zu müssen. Beim Import werden S3 Batch Operations verwendet, bei denen die Objekte im ausgewählten Präfix oder im Allzweck-Bucket kopiert werden. Sie können den Fortschritt des Importkopierauftrags auf der Seite mit den Auftragsdetails für S3 Batch Operations überwachen.

Um die Importaktion zu verwenden

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen der aktuell angezeigten Datei aus AWS-Region. Wählen Sie als Nächstes die Region aus, die der Availability Zone zugeordnet ist, in der sich Ihr Directory-Bucket befindet.
3. Wählen Sie im linken Navigationsbereich Buckets und dann die Registerkarte Verzeichnis-Buckets aus. Wählen Sie den Verzeichnis-Bucket aus, in den Sie Objekte importieren möchten.
4. Wählen Sie Importieren aus.
5. Geben Sie unter Quelle den Allzweck-Bucket (oder den Bucket-Pfad einschließlich Präfix) ein, der die Objekte enthält, die Sie importieren möchten. Um einen vorhandenen Allzweck-Bucket aus einer Liste auszuwählen, wählen Sie S3 durchsuchen.
6. Im Bereich Berechtigungen können Sie festlegen, dass eine IAM Rolle automatisch generiert wird. Alternativ können Sie eine IAM Rolle aus einer Liste auswählen oder direkt eine IAM Rolle ARN eingeben.

    - Um Amazon S3 zu ermöglichen, in Ihrem Namen eine neue IAM Rolle zu erstellen, wählen Sie Neue IAM Rolle erstellen.

> **Note**
>
> Wenn Ihre Quellobjekte serverseitig mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-KMS) verschlüsselt sind, wählen Sie nicht die Option Neue IAM Rolle erstellen. Geben Sie stattdessen eine vorhandene IAM Rolle an, die über die `kms:Decrypt` entsprechende Berechtigung verfügt.
> Amazon S3 verwendet diese Berechtigung, um Ihre Objekte zu entschlüsseln. Während des Importvorgangs verschlüsselt Amazon S3 diese Objekte dann erneut, indem es serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verwendet.

- Um eine bestehende IAM Rolle aus einer Liste auszuwählen, wählen Sie Aus vorhandenen Rollen auswählen. IAM
- Um eine bestehende IAM Rolle durch Eingabe ihres Amazon-Ressourcennamens (ARN) anzugeben, wählen Sie IAM Rolle eingeben ARN und geben Sie dann das ARN in das entsprechende Feld ein.

7. Überprüfen Sie die Informationen, die in den Abschnitten Ziel und Kopierte Objekteinstellungen angezeigt werden. Wenn die Informationen im Bereich Ziel korrekt sind, wählen Sie Import, um den Kopierauftrag zu starten.

   Die Amazon-S3-Konsole zeigt den Status Ihres neuen Auftrags auf der Seite Batch-Operationen an. Für weitere Informationen über den Auftrag klicken Sie auf das Optionsfeld neben dem Auftragsnamen und wählen dann im Menü Aktionen die Option Details anzeigen. Um den Verzeichnis-Bucket zu öffnen, in den die Objekte importiert werden, wählen Sie Importziel anzeigen.

Schritt 4: Laden Sie Objekte manuell in Ihren Verzeichnis-Bucket hoch

Sie können Objekte auch manuell in Ihren Verzeichnis-Bucket hochladen.

Um Objekte manuell hochzuladen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

2. Wählen Sie in der Navigationsleiste in der oberen rechten Ecke der Seite den Namen der aktuell angezeigten Seite aus AWS-Region. Wählen Sie als Nächstes die Region aus, die der Availability Zone zugeordnet ist, in der sich Ihr Directory-Bucket befindet.
3. Wählen Sie im linken Navigationsbereich Buckets aus.
4. Wählen Sie die Registerkarte Verzeichnis-Buckets.
5. Wählen Sie den Namen des Buckets aus, in den Sie Ihre Ordner oder Dateien hochladen möchten.

   > **Note**
   >
   > Wenn Sie denselben Verzeichnis-Bucket ausgewählt haben, den Sie in den vorherigen Schritten dieses Tutorials verwendet haben, enthält Ihr Verzeichnis-Bucket die Objekte, die mit dem Import-Tool hochgeladen wurden. Beachten Sie, dass diese Objekte jetzt in der S3 Express One Zone-Speicherklasse gespeichert sind.

6. Wählen Sie in der Objektliste die Option Hochladen aus.
7. Führen Sie auf der Upload-Seite einen der folgenden Schritte aus:

   - Ziehen Sie Dateien und Ordner per Drag & Drop in den gepunkteten Upload-Bereich.
   - Wählen Sie „Dateien hinzufügen“ oder „Ordner hinzufügen“, wählen Sie die Dateien oder Ordner aus, die Sie hochladen möchten, und wählen Sie dann „Öffnen“ oder „Hochladen“.

8. Wählen Sie unter Prüfsummen die Prüfsummenfunktion aus, die Sie verwenden möchten.

   > **Note**
   >
   > Wir empfehlen die Verwendung CRC32 und CRC32C für die beste Leistung mit der Speicherklasse S3 Express One Zone. Weitere Informationen finden Sie unter Bewährte Methoden für zusätzliche S3-Prüfsummen.

   (Optional) Wenn Sie ein einzelnes Objekt hochladen, das weniger als 16 MB groß ist, können Sie auch einen vorab berechneten Prüfsummenwert angeben. Wenn Sie einen vorberechneten Wert angeben, vergleicht Amazon S3 ihn mit dem Wert, den es mithilfe der ausgewählten Prüfsummenfunktion berechnet. Wenn die Werte nicht übereinstimmen, startet der Upload nicht.

9. Die Optionen in den Abschnitten „Berechtigungen“ und „Eigenschaften“ werden automatisch auf die Standardeinstellungen gesetzt und können nicht geändert werden. Block Public Access ist

automatisch aktiviert, und S3 Versioning und S3 Object Lock können für Directory-Buckets nicht aktiviert werden.

(Optional) Wenn Sie Ihren Objekten Metadaten in Schlüssel-Wert-Paaren hinzufügen möchten, erweitern Sie den Abschnitt Eigenschaften und wählen Sie dann im Abschnitt Metadaten die Option Metadaten hinzufügen aus.

10. Um die aufgelisteten Dateien und Ordner hochzuladen, wählen Sie Hochladen.

    Amazon S3 lädt Ihre Objekte und Ordner hoch. Wenn der Upload abgeschlossen ist, wird auf der Seite Upload: Status eine Erfolgsmeldung angezeigt.

    Sie haben erfolgreich einen Verzeichnis-Bucket erstellt und Objekte in Ihren Bucket hochgeladen.

Schritt 5: Leeren Sie Ihren Verzeichnis-Bucket

Sie können Ihren Amazon S3 S3-Verzeichnis-Bucket mithilfe der Amazon S3 S3-Konsole leeren.

Um einen Verzeichnis-Bucket zu leeren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Navigationsleiste in der oberen rechten Ecke der Seite den Namen der aktuell angezeigten Seite aus AWS-Region. Wählen Sie als Nächstes die Region aus, die der Availability Zone zugeordnet ist, in der sich Ihr Directory-Bucket befindet.
3. Wählen Sie im linken Navigationsbereich Buckets aus.
4. Wählen Sie die Registerkarte Verzeichnis-Buckets.
5. Wählen Sie das Optionsfeld neben dem Namen des Buckets aus, den Sie leeren möchten, und wählen Sie dann Leer aus.
6. Bestätigen Sie auf der Seite Empty bucket (Bucket leeren), dass Sie den Bucket leeren möchten, indem Sie **`permanently delete`** in das Textfeld eingeben und dann Empty (Leeren) auswählen.
7. Überwachen Sie den Fortschritt beim Entleeren des Buckets auf der Seite Leerer Bucket: Status.

Schritt 6: Löschen Sie Ihren Verzeichnis-Bucket

Nachdem Sie Ihren Verzeichnis-Bucket geleert und alle laufenden mehrteiligen Uploads abgebrochen haben, können Sie Ihren Bucket mithilfe der Amazon S3 S3-Konsole löschen.

Um einen Verzeichnis-Bucket zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Navigationsleiste in der oberen rechten Ecke der Seite den Namen der aktuell angezeigten Seite aus AWS-Region. Wählen Sie als Nächstes die Region aus, die der Availability Zone zugeordnet ist, in der sich Ihr Directory-Bucket befindet.
3. Wählen Sie im linken Navigationsbereich Buckets aus.
4. Wählen Sie die Registerkarte Verzeichnis-Buckets.
5. Wählen Sie in der Liste der Directory-Buckets das Optionsfeld neben dem Bucket aus, den Sie löschen möchten.
6. Wählen Sie Löschen.
7. Geben Sie auf der Seite „Bucket löschen“ den Namen des Buckets in das Textfeld ein, um das Löschen Ihres Buckets zu bestätigen.

   > **Important**
   >
   > Das Löschen eines Verzeichnis-Buckets kann nicht rückgängig gemacht werden.

8. Um Ihren Verzeichnis-Bucket zu löschen, wählen Sie Bucket löschen aus.

Nächste Schritte

In diesem Tutorial haben Sie gelernt, wie Sie einen Verzeichnis-Bucket erstellen und die S3 Express One Zone-Speicherklasse verwenden. Nach Abschluss dieses Tutorials können Sie sich mit den zugehörigen AWS Diensten vertraut machen, die Sie mit der S3 Express One Zone-Speicherklasse verwenden können.

Sie können Folgendes AWS-Services mit der S3 Express One Zone-Speicherklasse verwenden, um Ihren spezifischen Anwendungsfall mit niedriger Latenz zu unterstützen.

- Amazon Elastic Compute Cloud (AmazonEC2) — Amazon EC2 bietet sichere und skalierbare Rechenkapazität in der AWS Cloud. Durch die Verwendung von Amazon EC2 müssen Sie nicht im

Voraus in Hardware investieren, sodass Sie Anwendungen schneller entwickeln und bereitstellen können. Sie können Amazon verwendenEC2, um so viele oder so wenige virtuelle Server zu starten, wie Sie benötigen, Sicherheit und Netzwerke zu konfigurieren und Speicher zu verwalten.

- AWS Lambda – Lambda ist ein Computingservice, mit dem Sie Code ausführen können, ohne Server bereitstellen oder verwalten zu müssen. Sie konfigurieren Benachrichtigungseinstellungen für einen Bucket und erteilen die Amazon-S3-Berechtigung zum Aufrufen einer Funktion in der ressourcenbasierten Berechtigungsrichtlinie der Funktion.
- Amazon Elastic Kubernetes Service (AmazonEKS) — Amazon EKS ist ein verwalteter Service, der die Installation, den Betrieb und die Wartung Ihres eigenen Services überflüssig macht Kubernetes Kontrollebene aktiviert. AWSKubernetesist ein Open-Source-System, das die Verwaltung, Skalierung und Bereitstellung von containerisierten Anwendungen automatisiert.
- Amazon Elastic Container Service (AmazonECS) — Amazon ECS ist ein vollständig verwalteter Container-Orchestrierungsservice, mit dem Sie containerisierte Anwendungen einfach bereitstellen, verwalten und skalieren können.
- Amazon EMR — Amazon EMR ist eine verwaltete Cluster-Plattform, die den Betrieb von Big-Data-Frameworks vereinfacht, wie Apache Hadoop and Apache Spark weiter AWS zur Verarbeitung und Analyse riesiger Datenmengen.
- Amazon Athena — Athena ist ein interaktiver Abfrageservice, der es einfach macht, Daten mithilfe von Standards direkt in Amazon S3 zu analysieren. SQL Sie können Athena auch verwenden, um Datenanalysen interaktiv auszuführen, indem Sie Apache Spark ohne Ressourcen planen, konfigurieren oder verwalten zu müssen. Wenn du rennst Apache Spark Bewerbungen auf Athena reichen Sie ein Spark Code für die Verarbeitung und direkt die Ergebnisse erhalten.
- AWS Glue Data Catalog — AWS Glue ist ein serverloser Datenintegrationsdienst, der es Analytics-Benutzern erleichtert, Daten aus mehreren Quellen zu entdecken, aufzubereiten, zu verschieben und zu integrieren. Sie können ihn AWS Glue für Analysen, maschinelles Lernen und Anwendungsentwicklung verwenden. AWS Glue Data Catalog ist ein zentrales Repository, das Metadaten zu den Datensätzen Ihres Unternehmens speichert. Es dient als Index für den Standort, das Schema und die Laufzeitmetriken Ihrer Datenquellen.
- Amazon SageMaker Runtime Model Training — Amazon SageMaker Runtime ist ein vollständig verwalteter Service für maschinelles Lernen. Mit SageMaker Runtime können Datenwissenschaftler und Entwickler schnell und einfach Modelle für maschinelles Lernen erstellen und trainieren und sie dann direkt in einer produktionsbereiten gehosteten Umgebung bereitstellen.

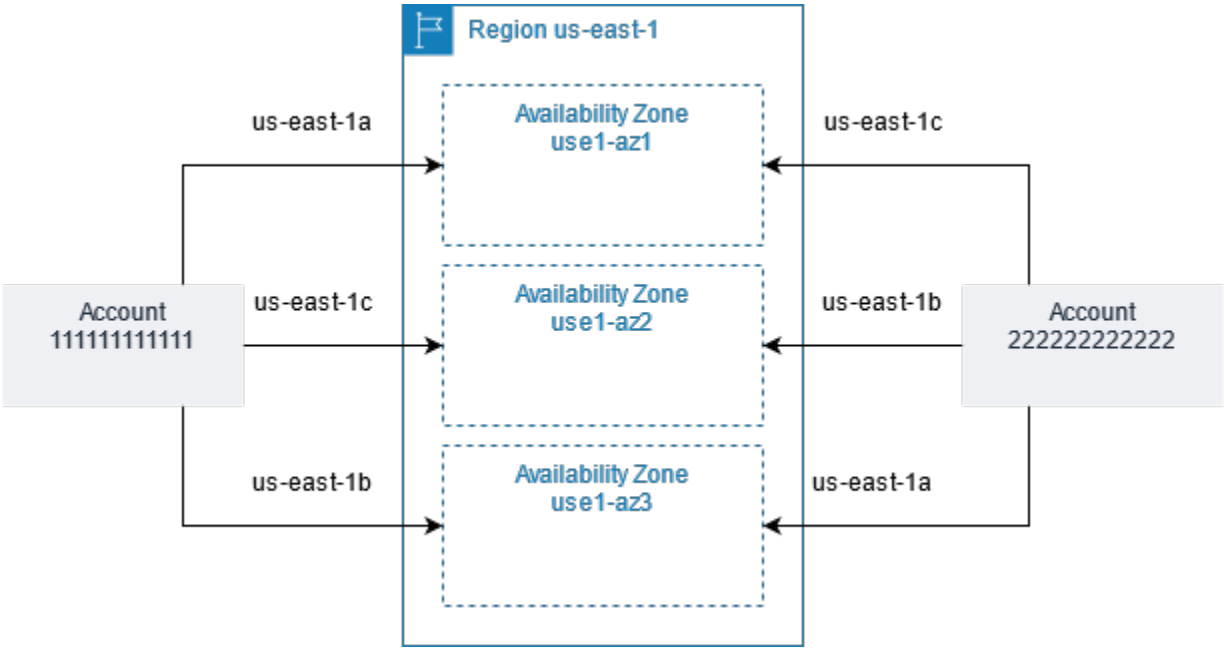
Weitere Informationen zu S3 Express One Zone finden Sie unter Was ist S3 Express One Zone? und wie unterscheidet sich S3 Express One Zone? .

Availability Zones und Regionen bei S3 Express One Zone

Eine Availability Zone ist eines oder mehrere diskrete Rechenzentren mit redundanter Stromversorgung, Vernetzung und Konnektivität in einem AWS-Region. Um Abrufe mit niedriger Latenz zu optimieren, werden Objekte in der Speicherklasse Amazon S3 Express One Zone redundant in S3-Verzeichnis-Buckets in einer einzigen Availability Zone gespeichert, die für Ihr Computing-Workload lokal ist. Wenn Sie einen Verzeichnis-Bucket erstellen, wählen Sie die Availability Zone und den AWS-Region Speicherort Ihres Buckets aus.

AWS ordnet die physischen Availability Zones nach dem Zufallsprinzip den jeweiligen Availability Zonen-Namen zu AWS-Konto. Dieser Ansatz hilft dabei, Ressourcen auf die Availability Zones in einer Region zu verteilen AWS-Region, anstatt dass sich die Ressourcen wahrscheinlich auf die erste Availability Zone für jede Region konzentrieren. Daher entspricht die Availability Zone `us-east-1a` für Sie AWS-Konto möglicherweise nicht demselben physischen Standort wie `us-east-1a` für eine andere AWS-Konto. Weitere Informationen finden Sie unter Regionen und Availability Zones im EC2Amazon-Benutzerhandbuch.

Um die Availability Zones kontenübergreifend zu koordinieren, müssen Sie die AZ-ID verwenden, die eine eindeutige und konsistente Kennung für eine Availability Zone ist. Dies `use1-az1` ist beispielsweise eine AZ-ID für die `us-east-1` Region und sie hat in jeder Region denselben physischen Standort AWS-Konto. Die folgende Abbildung zeigt, dass die AZ für jedes Konto identisch IDs sind, auch wenn die Namen der Availability Zone für jedes Konto unterschiedlich zugeordnet werden können.



Mit S3 Express One Zone werden Ihre Daten redundant auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert. S3 Express One Zone ist für eine Verfügbarkeit von 99,95 Prozent innerhalb einer einzigen Availability Zone konzipiert und wird durch das Amazon S3 Service Level Agreement unterstützt. Weitere Informationen finden Sie unter Availability Zones

Die folgende Tabelle zeigt die von S3 Express One Zone unterstützten Regionen und Availability Zones.

Name der Region	Regionscode	Availability Zone-ID				
USA Ost (Nord-Virginia)	us-east-1	use1-az4				
		use1-az5				
		use1-az6				
USA Ost (Ohio)	us-east-2	use2-az1				
		use2-az2				
USA West (Oregon)	us-west-2	usw2-az1				
		usw2-az3				
		usw2-az4				
Asien-Pazifik (Mumbai)	ap-south-1	aps1-az1				
		aps1-az3				
Asien-Pazifik (Tokio)	ap-northeast-1	apne1-az1				
		apne1-az4				
Europa (Irland)	eu-west-1	euw1-az1				
		euw1-az3				
Europa (Stockholm)	eu-north-1	eun1-az1				

Name der Region	Regionscode	Availability Zone-ID				
		eun1-az2				
		eun1-az3				

Netzwerke für Verzeichnis-Buckets in einer Availability Zone

In den folgenden Themen werden die Netzwerkanforderungen für den Zugriff auf S3 Express One Zone über einen VPC Gateway-Endpunkt beschrieben.

Themen

- Endpunkte für Verzeichnis-Buckets in Availability Zones
- Konfiguration von VPC Gateway-Endpunkten

Endpunkte für Verzeichnis-Buckets in Availability Zones

Die folgende Tabelle zeigt die regionalen und zonalen API Endpunkte, die für jede Region und Availability Zone verfügbar sind.

Konfiguration von VPC Gateway-Endpunkten

Gehen Sie wie folgt vor, um einen Gateway-Endpunkt zu erstellen, der eine Verbindung zu Objekten der Speicherklasse Amazon S3 Express One Zone und Verzeichnis-Buckets herstellt.

Um einen VPC Gateway-Endpunkt zu konfigurieren

1. Öffnen Sie die VPCAmazon-Konsole.
2. Wählen Sie im Navigationsbereich Endpunkte aus.
3. Wählen Sie Endpunkt erstellen.
4. Erstellen Sie einen Namen für Ihren Endpunkt.
5. Wählen Sie für Servicekategorie die Option AWS-Services aus.
6. Fügen Sie für Services den Filter Type=Gateway hinzu und wählen Sie dann die Optionsschaltfläche neben com.amazonaws.***region***.s3express.
7. Wählen Sie für die VPC, VPC in der der Endpunkt erstellt werden soll.

8. Wählen Sie für Route tables (Routing-Tabellen) die Routing-Tabellen, die von dem Endpunkt verwendet werden sollen. Amazon fügt VPC automatisch eine Route hinzu, die den für den Service bestimmten Verkehr an die Netzwerkschnittstelle des Endpunkts weiterleitet.
9. Wählen Sie für Policy die Option Vollzugriff aus, um alle Operationen aller Principals auf allen Ressourcen über den VPC Endpunkt zuzulassen. Wählen Sie andernfalls Benutzerdefiniert aus, um eine VPC Endpunktrichtlinie anzuhängen, die die Berechtigungen steuert, die Prinzipale haben, um Aktionen an Ressourcen über den VPC Endpunkt auszuführen.
10. (Optional) Sie fügen ein Tag hinzu, indem Sie Neues Tag hinzufügen auswählen und den Schlüssel und den Wert für das Tag eingeben.
11. Wählen Sie Endpunkt erstellen aus.

Nachdem Sie einen Gateway-Endpunkt erstellt haben, können Sie regionale API Endpunkte und zonale API Endpunkte verwenden, um auf Amazon S3 Express One Zone-Speicherklassenobjekte und Verzeichnis-Buckets zuzugreifen.

Verzeichnis-Buckets in einer Availability Zone erstellen

Um mit der Verwendung der Speicherklasse Amazon S3 Express One Zone zu beginnen, erstellen Sie einen Verzeichnis-Buckets. Die Speicherklasse S3 Express One Zone kann nur mit Verzeichnis-Buckets verwendet werden. Die Speicherklasse S3 Express One Zone unterstützt Anwendungsfälle mit niedriger Latenz und ermöglicht eine schnellere Datenverarbeitung innerhalb einer einzigen Availability Zone. Wenn Ihre Anwendung leistungsempfindlich ist und von `PUT`- und `GET`-Latenzen im einstelligen Millisekundenbereich profitiert, empfehlen wir, einen Verzeichnis-Bucket zu erstellen, damit Sie die Speicherklasse S3 Express One Zone verwenden können.

Es gibt zwei Arten von Amazon-S3-Buckets: Allzweck-Buckets und Verzeichnis-Buckets. Sie sollten den Bucket-Typ wählen, der am besten zu Ihren Anwendungs- und Leistungsanforderungen passt. Bei Allzweck-Buckets handelt es sich um den ursprünglichen S3-Bucket-Typ. Allzweck-Buckets werden für die meisten Anwendungsfälle und Zugriffsmuster empfohlen. Sie ermöglichen die Speicherung von Objekten in allen Speicherklassen, mit Ausnahme von S3 Express One Zone. Weitere Informationen über Allzweck-Buckets finden Sie unter Bucket-Übersicht.

Verzeichnis-Buckets verwenden die Speicherklasse S3 Express One Zone, die für Workloads oder leistungskritische Anwendungen konzipiert ist, die eine konsistente Latenz im einstelligen Millisekundenbereich erfordern. S3 Express One Zone ist die erste S3-Speicherklasse, bei der Sie eine einzelne Availability Zone mit der Option auswählen können, Ihren Objektspeicher gemeinsam mit Ihren Computingressourcen zu platzieren, was die höchstmögliche Zugriffsgeschwindigkeit bietet.

Wenn Sie einen Directory-Bucket erstellen, können Sie optional eine Availability Zone AWS-Region und eine Availability Zone angeben, die lokal für Ihre Amazon-EC2, Amazon Elastic Kubernetes Service- oder Amazon Elastic Container Service (AmazonECS) Compute-Instances sind, um die Leistung zu optimieren.

Mit S3 Express One Zone werden Ihre Daten redundant auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert. S3 Express One Zone ist für eine Verfügbarkeit von 99,95 Prozent innerhalb einer einzigen Availability Zone konzipiert und wird durch das Amazon S3 Service Level Agreement unterstützt. Weitere Informationen finden Sie unter Availability Zones

Verzeichnis-Buckets organisieren Daten hierarchisch in Verzeichnissen, im Gegensatz zu der flachen Speicherstruktur von Allzweck-Buckets. Es gibt keine Präfixbeschränkungen für Verzeichnis-Buckets und einzelne Verzeichnisse können horizontal skaliert werden.

Weitere Information zu Verzeichnis-Buckets finden Sie unter Arbeiten mit Verzeichnis-Buckets.

Namen von Verzeichnis-Buckets

Die Namen von Verzeichnis-Buckets müssen diesem Format folgen und den Regeln für die Benennung von Verzeichnis-Buckets entsprechen:

```
bucket-base-name--zone-id--x-s3
```

Der folgende Verzeichnis-Bucket-Name enthält beispielsweise die Availability Zone-ID `usw2-az1`:

```
bucket-base-name--usw2-az1--x-s3
```

Weitere Informationen zu den Benennungsregeln für Verzeichnis-Buckets finden Sie unter Regeln für die Benennung von Verzeichnis-Buckets.

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen der aktuell angezeigten Datei aus AWS-Region. Wählen Sie als Nächstes die Region aus, in der Sie einen Bucket erstellen möchten.

> **Note**
>
> Wählen Sie eine Region in der Nähe aus, um Latenz und Kosten gering zu halten und behördliche Vorschriften zu erfüllen. In einer Region gespeicherte Objekte verbleiben so lange in der Region, bis sie explizit in eine andere Region verschoben werden. Eine Liste von Amazon S3 finden Sie AWS-Regionen unter AWS-Service Endpoints in der Allgemeine Amazon Web Services-Referenz.

3. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
4. Wählen Sie Create Bucket (Bucket erstellen) aus.

   Anschließend wird die Seite Bucket erstellen geöffnet.
5. Sehen Sie sich unter Allgemeine Konfiguration an, AWS-Region wo Ihr Bucket erstellt wird.
6. Wählen Sie unter Bucket-Typ die Option Verzeichnis aus.

> **Note**
>
> - Wenn Sie eine Region ausgewählt haben, die keine Verzeichnis-Buckets unterstützt, verschwindet die Option Bucket-Typ und der Bucket-Typ ist standardmäßig ein Allzweck-Bucket. Um einen Verzeichnis-Bucket zu erstellen, müssen Sie eine unterstützte Region auswählen. Eine Liste der Regionen, die Directory-Buckets und die Amazon S3 Express One Zone-Speicherklasse unterstützen, finden Sie unterthe section called “Availability Zones und Regionen bei S3 Express One Zone”.
> - Nach dem Erstellen des Buckets kann der Bucket-Typ nicht mehr geändert werden.

Wählen Sie für Availability Zone eine Availability Zone aus, die für Ihre Computing-Services lokal ist. Eine Liste der Availability Zones, die Directory-Buckets und die Speicherklasse S3 Express One Zone unterstützen, finden Sie unter. the section called “Availability Zones und Regionen bei S3 Express One Zone”

> **Note**
>
> Die Availability Zone kann nicht geändert werden, nachdem der Bucket erstellt wurde.

7. Aktivieren Sie unter Availability Zone das Kontrollkästchen, um zu bestätigen, dass Ihre Daten im Falle eines Ausfalls der Availability Zone möglicherweise nicht verfügbar sind oder verloren gehen.

   > **Important**
   >
   > Obwohl Verzeichnis-Buckets auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert werden, speichern Verzeichnis-Buckets Daten nicht redundant über Availability Zones hinweg.

8. Geben Sie unter Bucket-Name einen Namen für Ihren Verzeichnis-Bucket ein.

   Die folgenden Benennungsregeln gelten für Verzeichnis-Buckets.

   - Seien Sie innerhalb der ausgewählten Zone (AWS Availability Zone) einzigartig.
   - Der Name muss zwischen 3 (min) und 63 (maximal) Zeichen lang sein, einschließlich des Suffixes.
   - Nur aus Kleinbuchstaben, Zahlen und Bindestrichen bestehen.
   - Muss mit einer Zahl oder einem Buchstaben beginnen und enden.
   - Muss das folgende Suffix enthalten:. `--`*`zone-id`*`--x-s3`
   - Der Bucket-Name darf nicht mit dem Präfix `xn--` beginnen.
   - Der Bucket-Name darf nicht mit dem Präfix `sthree-` beginnen.
   - Der Bucket-Name darf nicht mit dem Präfix `sthree-configurator` beginnen.
   - Der Bucket-Name darf nicht mit dem Präfix `amzn-s3-demo-` beginnen.
   - Bucket-Namen dürfen nicht mit dem Suffix `-s3alias` enden. Dieses Suffix ist für Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter Verwenden eines Alias im Bucket-Stil für Ihren S3-Bucket-Zugriffspunkt.
   - Bucket-Namen dürfen nicht mit dem Suffix `--ol-s3` enden. Dieses Suffix ist für Objekt-Lambda-Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter So verwenden Sie einen Alias im Bucket-Stil für den Object Lambda Access Point Ihres S3-Buckets.
   - Bucket-Namen dürfen nicht mit dem Suffix `.mrap` enden. Dieses Suffix ist für Namen von Access Points mit mehreren Regionen reserviert. Weitere Informationen finden Sie unter Regeln zur Benennung von Amazon S3-Multi-Regions-Zugriffspunkten.

Dem Basisnamen, den Sie angeben, wenn Sie mit der Konsole einen Verzeichnis-Bucket erstellen, wird automatisch ein Suffix hinzugefügt. Dieses Suffix enthält die ID der Availability Zone, die Sie ausgewählt haben.

Der Name eines einmal erstellter Buckets kann nicht nachträglich geändert werden. Weitere Informationen zur Benennung von Buckets finden Sie unter Allgemeine Benennungsregeln für Buckets.

> **Important**
>
> Nehmen Sie keine vertraulichen Informationen wie Kontonummern in den Bucket-Namen auf. Der Bucket-Name ist in dem Punkt sichtbarURLs, der auf die Objekte im Bucket verweist.

9. Unter Object Ownership ist die Einstellung Bucket Owner erforced automatisch aktiviert und alle Zugriffskontrolllisten (ACLs) sind deaktiviert. Kann für Verzeichnis-Buckets nicht ACLs aktiviert werden.

   ACLsdeaktiviert

   - Bucket Owner erforced (Standard) — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLswirkt sich nicht mehr auf die Zugriffsberechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet ausschließlich Richtlinien, um die Zugriffssteuerung zu definieren.

     Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

10. Unter „Einstellungen für öffentlichen Zugriff blockieren“ für diesen Bucket werden alle Einstellungen für „Öffentlichen Zugriff blockieren“ für Ihren Verzeichnis-Bucket automatisch aktiviert. Diese Einstellungen können für Directory-Buckets nicht geändert werden. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.
11. Unter Serverseitige Verschlüsselungseinstellungen verwenden Verzeichnis-Buckets serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3), um Daten zu verschlüsseln.

12. Wählen Sie Bucket erstellen aus.

    Nachdem Sie den Bucket erstellt haben, können Sie ihm Dateien und Ordner hinzufügen. Weitere Informationen finden Sie unter the section called “Arbeiten mit Objekten in einem Verzeichnis-Bucket”.

Unter Verwendung der AWS SDKs

SDK for Go

Dieses Beispiel zeigt, wie Sie mit dem einen Verzeichnis-Bucket erstellen AWS SDK for Go.

Example

```
var bucket = "..."

func runCreateBucket(c *s3.Client) {
    resp, err := c.CreateBucket(context.Background(), &s3.CreateBucketInput{
        Bucket: &bucket,
        CreateBucketConfiguration: &types.CreateBucketConfiguration{
            Location: &types.LocationInfo{
                Name: aws.String("usw2-az1"),
                Type: types.LocationTypeAvailabilityZone,
            },
            Bucket: &types.BucketInfo{
                DataRedundancy: types.DataRedundancySingleAvailabilityZone,
                Type:           types.BucketTypeDirectory,
            },
        },
    })
    var terr *types.BucketAlreadyOwnedByYou
    if errors.As(err, &terr) {
        fmt.Printf("BucketAlreadyOwnedByYou: %s\n", aws.ToString(terr.Message))
        fmt.Printf("noop...\n")
        return
    }
    if err != nil {
        log.Fatal(err)
    }

    fmt.Printf("bucket created at %s\n", aws.ToString(resp.Location))
}
```

SDK for Java 2.x

Dieses Beispiel zeigt, wie Sie einen Verzeichnis-Bucket mithilfe von erstellen AWS SDK for Java 2.x.

Example

```
public static void createBucket(S3Client s3Client, String bucketName) {

    //Bucket name format is {base-bucket-name}--{az-id}--x-s3
    //example: doc-example-bucket--usw2-az1--x-s3 is a valid name for a directory
 bucket created in
    //Region us-west-2, Availability Zone 2

    CreateBucketConfiguration bucketConfiguration =
 CreateBucketConfiguration.builder()
             .location(LocationInfo.builder()
                     .type(LocationType.AVAILABILITY_ZONE)
                     .name("usw2-az1").build()) //this must match the Region and
 Availability Zone in your bucket name
             .bucket(BucketInfo.builder()
                    .type(BucketType.DIRECTORY)
                    .dataRedundancy(DataRedundancy.SINGLE_AVAILABILITY_ZONE)
                    .build()).build();
    try {

             CreateBucketRequest bucketRequest =
 CreateBucketRequest.builder().bucket(bucketName).createBucketConfiguration(bucketConfigurat
             CreateBucketResponse response = s3Client.createBucket(bucketRequest);
             System.out.println(response);
    }

    catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
         }
    }
```

AWS SDK for JavaScript

Dieses Beispiel zeigt, wie Sie einen Verzeichnis-Bucket mithilfe von erstellen AWS SDK for JavaScript.

Example

```
// file.mjs, run with Node.js v16 or higher
// To use with the preview build, place this in a folder
// inside the preview build directory, such as /aws-sdk-js-v3/workspace/

import { S3 } from "@aws-sdk/client-s3";

const region = "us-east-1";
const zone = "use1-az4";
const suffix = `${zone}--x-s3`;

const s3 = new S3({ region });

const bucketName = `...--${suffix}`;

const createResponse = await s3.createBucket(
    { Bucket: bucketName,
      CreateBucketConfiguration: {Location: {Type: "AvailabilityZone", Name: zone},
      Bucket: { Type: "Directory", DataRedundancy: "SingleAvailabilityZone" }}
    }
   );
```

AWS SDK for .NET

Dieses Beispiel zeigt, wie Sie einen Verzeichnis-Bucket mithilfe von erstellen AWS SDK for .NET.

Example

```
using (var amazonS3Client = new AmazonS3Client())
{
    var putBucketResponse = await amazonS3Client.PutBucketAsync(new PutBucketRequest
    {

       BucketName = "DOC-EXAMPLE-BUCKET--usw2-az1--x-s3",
       PutBucketConfiguration = new PutBucketConfiguration
       {
         BucketInfo = new BucketInfo { DataRedundancy =
 DataRedundancy.SingleAvailabilityZone, Type = BucketType.Directory },
         Location = new LocationInfo { Name = "usw2-az1", Type =
 LocationType.AvailabilityZone }
       }
     }).ConfigureAwait(false);
```

```
}
```

SDK for PHP

Dieses Beispiel zeigt, wie Sie einen Verzeichnis-Bucket mithilfe von erstellen AWS SDK for PHP.

Example

```
require 'vendor/autoload.php';

$s3Client = new S3Client([

    'region'      => 'us-east-1',
]);


$result = $s3Client->createBucket([
    'Bucket' => 'doc-example-bucket--use1-az4--x-s3',
    'CreateBucketConfiguration' => [
        'Location' => ['Name'=> 'use1-az4', 'Type'=> 'AvailabilityZone'],
        'Bucket' => ["DataRedundancy" => "SingleAvailabilityZone" ,"Type" =>
 "Directory"]   ],
]);
```

SDK for Python

Dieses Beispiel zeigt, wie Sie einen Verzeichnis-Bucket mithilfe von erstellen AWS SDK for Python (Boto3).

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def create_bucket(s3_client, bucket_name, availability_zone):
    '''
    Create a directory bucket in a specified Availability Zone

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket to create; for example, 'doc-example-bucket--usw2-
az1--x-s3'
    :param availability_zone: String; Availability Zone ID to create the bucket in,
 for example, 'usw2-az1'
```

```
    :return: True if bucket is created, else False
    '''

    try:
        bucket_config = {
                'Location': {
                    'Type': 'AvailabilityZone',
                    'Name': availability_zone
                },
                'Bucket': {
                    'Type': 'Directory',
                    'DataRedundancy': 'SingleAvailabilityZone'
                }
            }
        s3_client.create_bucket(
            Bucket = bucket_name,
            CreateBucketConfiguration = bucket_config
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True


if __name__ == '__main__':
    bucket_name = 'BUCKET_NAME'
    region = 'us-west-2'
    availability_zone = 'usw2-az1'
    s3_client = boto3.client('s3', region_name = region)
    create_bucket(s3_client, bucket_name, availability_zone)
```

SDK for Ruby

Dieses Beispiel zeigt, wie Sie einen Verzeichnis-Bucket mithilfe von erstellen AWS SDK for Ruby.

Example

```
s3 = Aws::S3::Client.new(region:'us-west-2')
s3.create_bucket(
  bucket: "bucket_base_name--az_id--x-s3",
  create_bucket_configuration: {
    location: { name: 'usw2-az1', type: 'AvailabilityZone' },
    bucket: { data_redundancy: 'SingleAvailabilityZone', type: 'Directory' }
```

```
  }
)
```

Mit dem AWS CLI

Dieses Beispiel zeigt, wie Sie mit dem einen Verzeichnis-Bucket erstellen AWS CLI. Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

Wenn Sie einen Verzeichnis-Bucket erstellen, müssen Sie Konfigurationsdetails angeben und die folgende Benennungskonvention verwenden: *bucket-base-name*--*zone-id*--x-s3

```
aws s3api create-bucket
--bucket bucket-base-name--zone-id--x-s3
--create-bucket-configuration 'Location={Type=AvailabilityZone,Name=usw2-
az1},Bucket={DataRedundancy=SingleAvailabilityZone,Type=Directory}'
--region us-west-2
```

Weitere Informationen finden Sie unter create-bucket im. AWS Command Line Interface

Regionale und zonale Endpunkte für Verzeichnis-Buckets in einer Availability Zone

Für den Zugriff auf Ihre in S3 Express One Zone gespeicherten Objekte und Verzeichnis-Buckets verwenden Sie Gateway-Endpunkte. VPC Directory-Buckets verwenden regionale und zonale Endpunkte. API Je nachdem, welchen Amazon S3 API S3-Betrieb Sie verwenden, ist entweder ein regionaler oder ein zonaler Endpunkt erforderlich. Für die Nutzung von Gateway-Endpunkten fallen keine zusätzlichen Gebühren an.

APIOperationen auf Bucket-Ebene (oder Kontrollebene) sind über regionale Endpunkte verfügbar und werden als regionale Endpunktoperationen bezeichnet. API Beispiele für regionale API Endpunktoperationen sind und. `CreateBucket` `DeleteBucket`

Wenn Sie Verzeichnis-Buckets erstellen, die in S3 Express One Zone gespeichert werden, wählen Sie die Availability Zone aus, in der sich Ihr Bucket befinden soll. Sie können Zonal Endpoint API Operations verwenden, um die Objekte in Ihrem Verzeichnis-Bucket hochzuladen und zu verwalten.

APIOperationen auf Objektebene (oder Datenebene) sind über zonale Endpunkte verfügbar und werden als zonale Endpunktoperationen bezeichnet. API Beispiele für Operationen auf zonalen Endpunkten sind und. API `CreateSession` `PutObject`

Optimieren der Leistung von S3 Express One Zone

Amazon S3 Express One Zone ist eine leistungsstarke S3-Speicherklasse mit einer einzigen Availability Zone (AZ), die speziell für den konsistenten Datenzugriff im einstelligen Millisekundenbereich für Ihre latenzempfindlichsten Anwendungen entwickelt wurde. S3 Express One Zone ist die erste S3-Speicherklasse, mit der Sie leistungsstarke Objektspeicher- und AWS Rechenressourcen wie Amazon Elastic Compute Cloud, Amazon Elastic Kubernetes Service und Amazon Elastic Container Service innerhalb einer einzigen Availability Zone zusammenlegen können. Die gemeinsame Nutzung Ihrer Speicher- und Computingressourcen optimiert die Rechenleistung und die Kosten und sorgt für eine höhere Datenverarbeitungsgeschwindigkeit.

S3 Express One Zone bietet eine ähnliche Leistungselastizität wie andere S3-Speicherklassen, jedoch mit konsistenten Latenzen für Lese- und Schreibanfragen im ersten Byte im einstelligen Millisekundenbereich – bis zu zehnmal schneller als S3 Standard. S3 Express One Zone wurde von Grund auf für die Unterstützung von Burst-Durchsätzen bis hin zu sehr hohen Aggregatwerten konzipiert. Die Speicherklasse S3 Express One Zone verwendet eine maßgeschneiderte Architektur, um die Leistung zu optimieren und durch die Speicherung von Daten auf Hochleistungshardware eine konstant niedrige Anforderungslatenz zu gewährleisten. Das Objektprotokoll für S3 Express One Zone wurde verbessert, um die Authentifizierung und den Metadaten-Overhead zu optimieren.

Um die Zugriffsgeschwindigkeit weiter zu erhöhen und Hunderttausende von Anfragen pro Sekunde zu unterstützen, speichert S3 Express One Zone Daten außerdem in einem neuen Bucket-Typ, einem Amazon-S3-Verzeichnis-Bucket. Jeder S3-Verzeichnis-Bucket kann Hunderttausende von Transaktionen pro Sekunde unterstützen (). TPS

Die Kombination aus leistungsstarker, speziell entwickelter Hardware und Software, die Datenzugriffsgeschwindigkeiten im einstelligen Millisekundenbereich bietet, und Verzeichnis-Buckets, die für eine große Anzahl von Transaktionen pro Sekunde skalierbar sind, machen S3 Express One Zone zur besten Amazon-S3-Speicherklasse für anforderungsintensive Operationen oder leistungskritische Anwendungen.

Die folgenden Themen beschreiben bewährte Verfahren und Designmuster zur Optimierung der Leistung von Anwendungen, die die Speicherklasse Sr Express One Zone verwenden.

Themen

- Leistungsrichtlinien und Entwurfsmuster für S3 Express One Zone

Leistungsrichtlinien und Entwurfsmuster für S3 Express One Zone

Bei der Entwicklung von Anwendungen, die Objekte zu Amazon S3 Express One Zone hochladen und daraus abrufen, sollten Sie unsere bewährten Methoden befolgen, um die Leistung zu optimieren. Um die Speicherklasse S3 Express One Zone zu verwenden, müssen Sie einen S3-Verzeichnis-Bucket erstellen. Die Speicherklasse S3 Express One Zone wird für die Verwendung mit S3-Allzweck-Buckets nicht unterstützt.

Leistungsrichtlinien für alle anderen Amazon-S3-Speicherklassen und S3-Allzweck-Buckets finden Sie unter Bewährte Methoden für Designmuster: Optimieren der Leistung von Amazon S3.

Zur Erzielung der besten Leistung für Ihre Anwendung bei Verwendung der Speicherklasse S3 Express One Zone und von Verzeichnis-Buckets empfehlen wir die folgenden Vorgehensweisen und Designmuster.

Themen

- Platzieren Sie den S3 Express One Zone-Speicher gemeinsam mit Ihren Rechenressourcen AWS
- Verzeichnis-Buckets
- Parallelisierung horizontaler Skalierungsanforderungen für verzeichnis-Buckets
- Verwenden der sitzungsbasierten Authentifizierung
- Bewährte Methoden für zusätzliche S3-Prüfsummen
- Verwenden Sie die neueste Version der AWS SDKs und der gängigen Runtime-Bibliotheken
- Fehlerbehebung bei der Leistung

Platzieren Sie den S3 Express One Zone-Speicher gemeinsam mit Ihren Rechenressourcen AWS

Jeder Verzeichnis-Bucket wird in einer einzigen Availability Zone gespeichert, die Sie bei dessen Erstellung auswählen. Sie können damit beginnen, indem Sie neuen Verzeichnis-Bucket in einer Availability Zone erstellen, die für Ihre Computng-Workloads oder Ressourcen lokal ist. Sie können dann sofort mit Lese- und Schreibvorgängen mit sehr niedriger Latenz beginnen. Directory-Buckets sind die ersten S3-Buckets, in denen Sie die Availability Zone auswählen können, AWS-Region um die Latenz zwischen Datenverarbeitung und Speicher zu reduzieren.

Wenn Sie in verschiedenen Availability Zones auf Verzeichnis-Buckets zugreifen, erhöht sich die Latenz. Um die Leistung zu optimieren, empfehlen wir, dass Sie von Amazon Elastic Container Service-, Amazon Elastic Kubernetes Service- und Amazon Elastic Compute Cloud-Instances, die sich nach Möglichkeit in derselben Availability Zone befinden, auf einen Directory-Bucket zugreifen.

Verzeichnis-Buckets

Jeder Verzeichnis-Bucket kann Hunderttausende von Transaktionen pro Sekunde unterstützen ()TPS. Im Gegensatz zu Allzweck-Buckets organisieren Verzeichnis-Buckets Schlüssel hierarchisch in Verzeichnissen statt nach Präfixen. Ein Präfix ist eine Zeichenfolge am Anfang des Objektschlüsselnamens. Sie können sich Präfixe als eine Möglichkeit vorstellen, Ihre Daten ähnlich wie Verzeichnisse zu organisieren. Präfixe sind jedoch keine Verzeichnisse.

Präfixe organisieren Daten in einem flachen Namespace innerhalb von Allzweck-Buckets und die Anzahl der Präfixe in einem Allzweck-Bucket ist unbegrenzt. Mit jedem Präfix können mindestens 3.500`PUT`/`DELETE`oder 5.500 `POST`/`GET`/`HEAD`Anfragen pro Sekunde erreicht werden. Sie können Anfragen auch über mehrere Präfixe parallelisieren, um die Leistung zu skalieren. Die Skalierung erfolgt sowohl bei Lese- als auch bei Schreiboperationen schrittweise und nicht sofort. Allzweck-Buckets werden zwar auf Ihre neue höhere Anforderungsrate skaliert, Sie erhalten jedoch möglicherweise einige HTTP Statuscode-Fehler 503 (Service Unavailable).

Bei einem hierarchischen Namespace ist das Trennzeichen im Objektschlüssel wichtig. Das einzige unterstützte Trennzeichen ist der Schrägstrich (`/`). Verzeichnisse werden durch Trennzeichengrenzen bestimmt. Beispielsweise führt der Objektschlüssel `dir1/dir2/file1.txt` dazu, dass die Verzeichnisse `dir1`/und `dir2/` automatisch erstellt werden und das Objekt `file1.txt` dem `/dir2`-Verzeichnis im Pfad `dir1/dir2/file1.txt` hinzugefügt wird.

Für die Verzeichnisse, die beim Hochladen von Objekten in Verzeichnis-Buckets erstellt werden, gibt es keine TPS Beschränkungen pro Präfix. Sie werden automatisch vorskaliert, um die Wahrscheinlichkeit von HTTP 503-Fehlern (Service Unavailable) zu verringern. Diese automatische Skalierung ermöglicht Ihren Anwendungen, Lese- und Schreibanforderungen innerhalb von Verzeichnissen und zwischen Verzeichnissen nach Bedarf zu parallelisieren.

Parallelisierung horizontaler Skalierungsanforderungen für verzeichnis-Buckets

Sie erreichen die beste Leistung durch die Ausgabe mehrerer gleichzeitiger Anfragen an Verzeichnis-Buckets zur Verteilung Ihrer Anforderungen auf separate Verbindungen und zur Maximierung der verfügbaren Bandbreite. S3Express One Zone hat keine Beschränkungen für die Anzahl der Verbindungen, die mit Ihrem Verzeichnis-Bucket hergestellt werden. Einzelne Verzeichnisse können die Leistung horizontal und automatisch skalieren, wenn eine große Anzahl gleichzeitiger Schreibvorgänge in dasselbe Verzeichnis stattfindet.

Wenn ein Objektschlüssel zum ersten Mal erstellt wird und sein Schlüsselname ein Verzeichnis enthält, wird das Verzeichnis automatisch für das Objekt erstellt. Bei nachfolgenden Objekt-Uploads

in dasselbe Verzeichnis muss das Verzeichnis nicht erstellt werden, wodurch die Latenz beim Hochladen von Objekten in bestehende Verzeichnisse reduziert wird.

Obwohl sowohl flache als auch tiefe Verzeichnisstrukturen für das Speichern von Objekten in einem Verzeichnis-Bucket unterstützt werden, werden Verzeichnis-Buckets automatisch horizontal skaliert, wodurch die Latenz bei gleichzeitigen Uploads in dasselbe Verzeichnis oder in parallele „Verzeichnisgeschwister“ geringer ist.

Verwenden der sitzungsbasierten Authentifizierung

S3 Express One Zone und Verzeichnis-Buckets unterstützen einen neuen sitzungsbasierten Autorisierungsmechanismus zur Authentifizierung und Autorisierung von Anforderungen an einen Verzeichnis-Bucket. Bei der sitzungsbasierten Authentifizierung verwenden sie den `CreateSession` API Vorgang AWS SDKs automatisch, um ein temporäres Sitzungstoken zu erstellen, das für die Autorisierung von Datenanforderungen an einen Verzeichnis-Bucket mit geringer Latenz verwendet werden kann.

AWS SDKsSie verwenden den `CreateSession` API Vorgang, um temporäre Anmeldeinformationen anzufordern und anschließend automatisch alle 5 Minuten Token für Sie zu erstellen und zu aktualisieren. Um die Leistungsvorteile der Speicherklasse S3 Express One Zone zu nutzen, wurde empfohlen, die AWS SDKs zum Initiieren und Verwalten der `CreateSession` API Anfrage zu verwenden. Weitere Informationen zu diesem sitzungsbasierten Modell finden Sie unter Autorisieren von zonalen API Endpunktoperationen mit `CreateSession`.

Bewährte Methoden für zusätzliche S3-Prüfsummen

S3 Express One Zone bietet Ihnen die Möglichkeit, den Prüfsummenalgorithmus auszuwählen, der zur Validierung Ihrer Daten beim Hoch- oder Herunterladen verwendet wird. Sie können einen der folgenden Algorithmen zur Prüfung der Datenintegrität zwischen Secure Hash Algorithms (SHA) oder Cyclic Redundancy Check (CRC) wählen:CRC32,CRC32C, SHA -1 und -256. SHA MD5Auf Prüfsummen basierende Prüfsummen werden von der S3 Express One Zone-Speicherklasse nicht unterstützt.

CRC32ist die Standardprüfsumme, die AWS SDKs bei der Übertragung von Daten zu oder von der S3 Express One Zone verwendet wird. Wir empfehlen die Verwendung von CRC32 und CRC32C für die beste Leistung mit der Speicherklasse S3 Express One Zone.

Verwenden Sie die neueste Version der AWS SDKs und der gängigen Runtime-Bibliotheken

Einige von ihnen bieten AWS SDKs auch die AWS Common Runtime (CRT) -Bibliotheken, um die Leistung von S3-Clients weiter zu beschleunigen. Dazu SDKs gehören die AWS SDK for Java 2.x AWS SDK for C++, die und die AWS SDK for Python (Boto3). Der S3-Client CRT auf Basis von S3 Express One Zone überträgt Objekte mit verbesserter Leistung und Zuverlässigkeit, indem er automatisch den mehrteiligen API Upload-Vorgang und Abrufe im Bytebereich verwendet, um Verbindungen mit horizontaler Skalierung zu automatisieren.

Um die höchste Leistung mit der S3 Express One Zone-Speicherklasse zu erzielen, empfehlen wir, die neueste Version von zu verwenden, die die CRT Bibliotheken enthält AWS SDKs, oder die () zu verwenden. AWS Command Line Interface AWS CLI

Fehlerbehebung bei der Leistung

Wiederholung von Anforderungen für latzenzsensitive Anwendungen

S3 Express One Zone wurde speziell für gleichbleibende Leistung ohne zusätzliche Anpassungen entwickelt. Die Festlegung aggressiver Timeout-Werte und Wiederholungsversuche kann jedoch weiter zu einer gleichbleibenden Latenz und Leistung beitragen. Sie AWS SDKs verfügen über konfigurierbare Timeout- und Wiederholungswerte, die Sie an die Toleranzen Ihrer spezifischen Anwendung anpassen können.

AWS Allgemeine Runtime (CRT) -Bibliotheken und Paarung von EC2 Amazon-Instance-Typen

Anwendungen, die eine große Anzahl von Lese- und Schreibvorgängen ausführen, benötigen wahrscheinlich mehr Arbeitsspeicher oder Computing-Kapazitäten als Anwendungen, bei denen dies nicht der Fall ist. Wenn Sie Ihre Amazon Elastic Compute Cloud (AmazonEC2) -Instances für Ihre leistungsintensive Arbeitslast starten, wählen Sie Instance-Typen aus, die über die Menge dieser Ressourcen verfügen, die Ihre Anwendung benötigt. S3 Express One Zone Hochleistungsspeicher lässt sich ideal mit größeren und neueren Instance-Typen kombinieren, die über mehr Systemspeicher verfügen und leistungsfähiger sind CPUs und GPUs die Vorteile von leistungsfähigerem Speicher nutzen können. Wir empfehlen außerdem, die neuesten Versionen von CRT -enabled zu verwenden AWS SDKs, wodurch Lese- und Schreibanforderungen besser parallel beschleunigt werden können.

Verwenden Sie die sitzungsbasierte Authentifizierung in AWS SDKs anstelle von HTTP REST APIs

Mit Amazon S3 können Sie auch die Leistung bei der Verwendung von HTTP REST API Anfragen optimieren, indem Sie dieselben bewährten Methoden befolgen, die Teil von sind AWS SDKs. Angesichts des sitzungsbasierten Autorisierungs- und Authentifizierungsmechanismus, der von S3 Express One Zone verwendet wird, empfehlen wir Ihnen jedoch dringend, das To-Manage-

Sitzungstoken `CreateSession` und das AWS SDKs zugehörige Sitzungstoken zu verwenden. Sie erstellen und aktualisieren mithilfe des Vorgangs AWS SDKs automatisch Token in Ihrem Namen. `CreateSession` API Die Verwendung von AWS Identity and Access Management (IAM) `CreateSession` spart die Round-Trip-Latenz pro Anfrage, um jede Anfrage zu autorisieren.

Unterschiede bei Directory-Buckets

Wenn Sie Amazon S3 verwenden, können Sie den Bucket-Typ wählen, der Ihren Anwendungs- und Leistungsanforderungen am besten entspricht. Ein Verzeichnis-Bucket ist ein Bucket-Typ, der sich am besten für Anwendungsfälle mit niedriger Latenz eignet. Sie können einen Verzeichnis-Bucket in einer einzelnen Availability Zone erstellen, um Daten zu speichern. Sie können in jedem Ihrer AWS Konten bis zu 100 Verzeichnis-Buckets erstellen, wobei die Anzahl der Objekte, die Sie in einem Bucket speichern können, unbegrenzt ist. Ihr Bucket-Kontingent wird auf jede Region in Ihrem Konto angewendet. AWS Wenn für Ihre Anwendung eine Erhöhung dieses Limits erforderlich ist, wenden Sie sich an den AWS Support. Directory-Buckets unterstützen die Speicherklasse S3 Express One Zone. Die Speicherklasse S3 Express One Zone wird empfohlen, wenn Ihre Anwendung leistungsempfindlich ist und von einstelligen Millisekunden `PUT` und Latenzen profitiert. `GET` Weitere Informationen zur Speicherklasse S3 Express One Zone finden Sie unter S3 Express One Zone.

Weitere Informationen zu den Unterschieden zwischen Verzeichnis-Buckets finden Sie in den folgenden Themen.

Themen

- Unterschiede bei Directory-Buckets
- APIOperationen, die für Directory-Buckets unterstützt werden
- Amazon S3 S3-Funktionen, die von Directory-Buckets nicht unterstützt werden

Unterschiede bei Directory-Buckets

- Namen von Verzeichnis-Buckets
  - Ein Verzeichnis-Bucket-Name besteht aus einem Basisnamen, den Sie angeben, und einem Suffix, das die ID der Zone (Availability Zone) enthält, in der sich Ihr Bucket befindet. Die Namen von Verzeichnis-Buckets müssen ein bestimmtes Format verwenden und den Benennungsregeln für Directory-Buckets entsprechen. Eine Liste von Regeln und Beispiele für Verzeichnis-Bucket-Namen finden Sie unter Benennungsregeln für Verzeichnis-Buckets.
- **`ListObjectsV2`**Verhalten

  - Gibt bei Verzeichnis-Buckets `ListObjectsV2` keine Objekte in lexikografischer (alphabetischer) Reihenfolge zurück. Außerdem müssen Präfixe mit einem Trennzeichen enden und es kann nur „/“ als Trennzeichen angegeben werden.
  - Bei Verzeichnis-Buckets enthält die `ListObjectsV2` Antwort die Präfixe, die sich nur auf laufende mehrteilige Uploads beziehen.
- Löschverhalten – Wenn Sie ein Objekt in einem Verzeichnis-Bucket löschen, löscht Amazon S3 rekursiv alle leeren Verzeichnisse im Objektpfad. Wenn Sie beispielsweise den Objektschlüssel löschen`dir1/dir2/file1.txt`, löscht Amazon S3 ihn`file1.txt`. Wenn die Verzeichnisse `dir1/` und `dir2/` leer sind und keine anderen Objekte enthalten, löscht Amazon S3 auch diese Verzeichnisse.
- ETagsund Prüfsummen — Entitäts-Tags (ETags) für Verzeichnis-Buckets sind zufällige alphanumerische Zeichenketten und keine Prüfsummen. MD5
- Objektschlüssel in **`DeleteObjects`**-Anforderungen
  - Objektschlüssel in `DeleteObjects`-Anforderungen müssen mindestens ein Zeichen enthalten, das kein Leerzeichen ist. Zeichenfolgen, die ausschließlich Leerzeichen enthalten, werden in `DeleteObjects`-Anforderungen nicht unterstützt.
  - Objektschlüssel in `DeleteObjects`-Anforderungen dürfen keine Unicode-Steuerzeichen enthalten, mit Ausnahme von newline (`\n`), tab (`\t`) und carriage return (`\r`).
- Regionale und zonale Endpunkte — API Bucket-Management-Operationen für Directory-Buckets sind über einen regionalen Endpunkt verfügbar und werden als regionale Endpunktoperationen bezeichnet. API Beispiele für regionale Endpunktoperationen sind und. API CreateBucket DeleteBucket Nachdem Sie einen Verzeichnis-Bucket erstellt haben, können Sie Zonal Endpoint API Operations verwenden, um die Objekte in Ihrem Verzeichnis-Bucket hochzuladen und zu verwalten. Zonale API Endpunktoperationen sind über einen zonalen Endpunkt verfügbar. Beispiele für API Operationen an zonalen Endpunkten sind und. `PutObject` `CopyObject` Wenn Sie Verzeichnis-Buckets verwenden, müssen Sie die Region in allen Anfragen angeben. Für regionale Endpunkte geben Sie die Region an, zum Beispiel `s3express-control.us-west-2.amazonaws.com`. Für zonale Endpunkte geben Sie sowohl die Region als auch die Availability Zone an, zum Beispiel `s3express-usw2-az1.us-west-2.amazonaws.com`. Weitere Informationen finden Sie unter Regionale und zonale Endpunkte für Directory-Buckets.
- Mehrteilige Uploads — Sie können große Objekte, die in Verzeichnis-Buckets gespeichert sind, mithilfe des mehrteiligen Upload-Prozesses hochladen und kopieren. Im Folgenden sind jedoch einige Unterschiede aufgeführt, wenn der mehrteilige Upload-Prozess für Objekte verwendet wird, die in Verzeichnis-Buckets gespeichert sind. Weitere Informationen finden Sie unter the section called “Mehrteilige Uploads mit Verzeichnis-Buckets verwenden”.

  - Das Objekterstellungsdatum ist das Abschlussdatum des mehrteiligen Uploads.
  - Bei mehrteiligen Uploads müssen aufeinanderfolgende Teilenummern verwendet werden. Wenn Sie versuchen, eine mehrteilige Upload-Anfrage mit nicht aufeinanderfolgenden Artikelnummern abzuschließen, generiert Amazon S3 einen HTTP `400 (Bad Request)` Fehler.
  - Der Initiator eines mehrteiligen Uploads kann die Anforderung für den mehrteiligen Upload nur abbrechen, wenn ihm durch die entsprechende `s3express:CreateSession`-Berechtigung der Zugriff auf `AbortMultipartUpload` ausdrücklich gewährt wurde. Weitere Informationen finden Sie unter Autorisieren eines regionalen Endpunkts APIs mit IAM.
- Leeren eines Verzeichnis-Buckets — Der **`s3 rm`** Befehl über AWS Command Line Interface (CLI), der **`delete`** Vorgang über Mountpoint und die Optionsschaltfläche Bucket leeren über können in Bearbeitung befindliche mehrteilige Uploads in einem Verzeichnis-Bucket nicht gelöscht AWS Management Console werden. Um diese laufenden mehrteiligen Uploads zu löschen, verwenden Sie die `ListMultipartUploads` Operation, um die laufenden mehrteiligen Uploads im Bucket aufzulisten und mit der Operation alle laufenden mehrteiligen Uploads abzubrechen. `AbortMultipartUpload`
- Server-side encryption
  - Für Verzeichnis-Buckets können Sie zur Verschlüsselung Ihrer Daten mit serverseitiger Verschlüsselung entweder serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) (Standard) oder serverseitige Verschlüsselung mit () Schlüsseln AWS Key Management Service (AWS KMS-) verwenden. SSE KMS Wir empfehlen, dass die Standardverschlüsselung des Buckets die gewünschten Verschlüsselungskonfigurationen verwendet und dass Sie die Standardverschlüsselung des Buckets in Ihren Anfragen oder Objektanforderungen nicht überschreiben. `CreateSession` PUT Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Weitere Informationen zum Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets finden Sie unter Serverseitige Verschlüsselung AWS KMS für Uploads neuer Objekte angeben.
  - Ihre KMS Konfiguration SSE — kann für die gesamte Lebensdauer des Buckets nur einen vom Kunden verwalteten Schlüssel pro Verzeichnis-Bucket unterstützen. Die Von AWS verwalteter Schlüssel(`aws/s3`) wird nicht unterstützt. Außerdem können Sie, nachdem Sie einen vom Kunden verwalteten Schlüssel für SSE - angegeben habenKMS, den vom Kunden verwalteten Schlüssel für die KMS Konfiguration des SSE Buckets nicht überschreiben.

    Sie können den vom Kunden verwalteten Schlüssel, den Sie für die KMS Konfiguration SSE — des Buckets angegeben haben, wie folgt identifizieren:

- Sie stellen eine `HeadObject` API Operationsanfrage, um den Wert von `x-amz-server-side-encryption-aws-kms-key-id` in Ihrer Antwort zu ermitteln.

Um einen neuen, vom Kunden verwalteten Schlüssel für Ihre Daten zu verwenden, empfehlen wir, Ihre vorhandenen Objekte mit einem neuen kundenverwalteten Schlüssel in einen neuen Verzeichnis-Bucket zu kopieren.

- Bei APIVorgängen auf zonaler Endpunktebene (Objektebene) mit Ausnahme von CopyObjectund UploadPartCopyauthentifizieren und autorisieren Sie Anfragen über CreateSessionfür niedrige Latenz. Wir empfehlen, dass die Standardverschlüsselung des Buckets die gewünschte Verschlüsselungskonfiguration verwendet und dass Sie die Standardverschlüsselung des Buckets in Ihren `CreateSession` Anfragen oder PUT Objektanfragen nicht überschreiben. Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Um neue Objekte in einem Verzeichnis-Bucket mit SSE - zu verschlüsselnKMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom Kunden verwalteten Schlüssel) angeben. Wenn dann eine Sitzung für zonale API Endpunktoperationen erstellt wird, werden neue Objekte während der Sitzung automatisch mit SSE - und S3-Bucket-Schlüsseln ver KMS - und entschlüsselt. Weitere Informationen zum Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets finden Sie unter Serverseitige Verschlüsselung mit AWS KMS für Uploads neuer Objekte angeben.
- Um neue Objektkopien in einem Verzeichnis-Bucket mit SSE - zu verschlüsselnKMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom Kunden verwalteten Schlüssel) angeben. CopyObject Wenn Sie dann serverseitige Verschlüsselungseinstellungen für neue Objektkopien mit SSE - angeben, müssen Sie sicherstellenKMS, dass der Verschlüsselungsschlüssel derselbe vom Kunden verwaltete Schlüssel ist, den Sie für die Standardverschlüsselungskonfiguration des Directory-Buckets angegeben haben. Um neue Objektteilkopien in einem Verzeichnis-Bucket mit SSE - zu verschlüsselnKMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom Kunden verwalteten Schlüssel) angeben. UploadPartCopy Sie können keine serverseitigen Verschlüsselungseinstellungen für neue Objektteilkopien mit SSE - KMS in den UploadPartCopyAnforderungsheadern angeben. Außerdem müssen die Verschlüsselungseinstellungen, die Sie in der CreateMultipartUploadAnfrage angeben, mit der Standardverschlüsselungskonfiguration des Ziel-Buckets übereinstimmen.

- S3-Bucket-Keys sind immer für `GET` und für `PUT` Operationen in einem Verzeichnis-Bucket aktiviert und können nicht deaktiviert werden. S3-Bucket-Keys werden nicht unterstützt, wenn Sie KMS verschlüsselte Objekte von Allzweck-Buckets in Verzeichnis-Buckets, von Verzeichnis-Buckets in Allzweck-Buckets oder zwischen Verzeichnis-Buckets kopieren SSE CopyObject, UploadPartCopy, der Copy Operation in Batch-Operationen, oder die import Jobs. In diesem Fall ruft Amazon S3 AWS KMS jedes Mal, wenn eine Kopieranforderung für ein KMS -verschlüsseltes Objekt gestellt wird, auf.
- Wenn Sie in Ihrem Verzeichnis-Bucket einen vom AWS KMS Kunden verwalteten Schlüssel für die Verschlüsselung angeben, verwenden Sie nur die Schlüssel-ID oder den SchlüsselARN. Das Schlüsselaliasformat des KMS Schlüssels wird nicht unterstützt.

APIOperationen, die für Directory-Buckets unterstützt werden

Die Speicherklasse Amazon S3 Express One Zone unterstützt sowohl regionale (Bucket-Ebene oder Kontrollebene) als auch zonale (Objektebene oder Datenebene) API Endpunktoperationen. Weitere Informationen erhalten Sie unter Netzwerke für Verzeichnis-Buckets und Endpunkte und Gateway-Endpunkte VPC.

Regionale Endpunktoperationen API

Die folgenden regionalen API Endpunktoperationen werden für S3 Express One Zone unterstützt:

- CreateBucket
- DeleteBucket
- DeleteBucketPolicy
- GetBucketPolicy
- ListDirectoryBuckets
- PutBucketPolicy

Zonale Endpunktoperationen API

Die folgenden zonalen API Endpunktoperationen werden für die Verwendung mit Directory-Buckets unterstützt:

- CreateSession
- CopyObject

- DeleteObject
- DeleteObjects
- GetObject
- GetObjectAttributes
- HeadBucket
- HeadObject
- ListObjectsV2
- PutObject
- AbortMultipartUpload
- CompleteMultiPartUpload
- CreateMultipartUpload
- ListMultipartUploads
- ListParts
- UploadPart
- UploadPartCopy
- UploadPartCopy
- PutBucketEncryption
- GetBucketEncryption
- DeleteBucketEncryption

Amazon S3 S3-Funktionen, die von Directory-Buckets nicht unterstützt werden

Die folgenden Amazon-S3-Features werden von S3 Express One Zone nicht unterstützt:

- AWS verwaltete Richtlinien
- AWS PrivateLink für S3
- MD5Prüfsummen
- Multi-Faktor-Authentifizierung () löschen MFA
- S3-Objektsperre

- Zahlung durch den Anforderer
- S3 Access Grants
- S3-Zugriffspunkte
- Bucket-Tags
- CloudWatch Amazon-Anforderungskennzahlen
- S3-Ereignis-Benachrichtigungen
- S3-Lebenszyklus
- Multiregionale S3-Zugriffspunkte
- S3-Objekt-Lambda-Zugangspunkte
- S3-Versioning
- S3-Bestand
- S3-Replikation
- Objekt-Tags
- S3 Select
- Serverzugriffsprotokolle
- Statisches Website-Hosting
- S3 Storage Lens
- S3-Storage-Lens-Gruppen
- S3 Transfer Acceleration
- Zweischichtige serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln (DSSE-) KMS
- Serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE
- Die Option, die Einstellungen eines vorhandenen Buckets zu kopieren, wenn ein neuer Bucket in der Amazon S3 S3-Konsole erstellt wird
- Verbesserte Fehlermeldungen „Zugriff verweigert“ (HTTP`403 Forbidden`)

Netzwerke für Verzeichnis-Buckets

Um auf Verzeichnis-Buckets und Objekte darin zuzugreifen, verwenden Sie regionale und zonale API Endpunkte, die sich von den standardmäßigen Amazon S3 S3-Endpunkten unterscheiden. Je nachdem, welchen API S3-Vorgang Sie verwenden, ist entweder ein zonaler oder ein regionaler

Endpunkt erforderlich. Eine vollständige Liste der API Operationen nach Endpunkttyp finden Sie unterUnterschiede bei Directory-Buckets.

Sie können über Gateway-Endpunkte der virtuellen privaten Cloud (VPC) sowohl auf zonale als auch auf regionale API Operationen zugreifen.

In den folgenden Themen werden die Netzwerkanforderungen für den Zugriff auf S3 Express One Zone über einen VPC Gateway-Endpunkt beschrieben.

Themen

- Endpunkte
- Konfiguration von Gateway-Endpunkten VPC

Endpunkte

Sie können mithilfe von VPC Gateway-Endpunkten auf Verzeichnis-Buckets und Objekte innerhalb Ihrer VPC zugreifen. Verzeichnis-Buckets verwenden regionale und zonale Endpunkte. API Je nachdem, welchen Amazon S3 API S3-Betrieb Sie verwenden, ist entweder ein regionaler oder ein zonaler Endpunkt erforderlich. Für die Nutzung von Gateway-Endpunkten fallen keine zusätzlichen Gebühren an.

APIOperationen auf Bucket-Ebene (oder Kontrollebene) sind über regionale Endpunkte verfügbar und werden als regionale Endpunktoperationen bezeichnet. API Beispiele für regionale API Endpunktoperationen sind und. `CreateBucket` `DeleteBucket` Wenn Sie einen Verzeichnis-Bucket erstellen, wählen Sie eine einzelne Zone (Availability Zone) aus, in der Ihr Verzeichnis-Bucket erstellt werden soll. Nachdem Sie einen Verzeichnis-Bucket erstellt haben, können Sie Zonal Endpoint API Operations verwenden, um die Objekte in Ihrem Verzeichnis-Bucket hochzuladen und zu verwalten.

APIOperationen auf Objektebene (oder Datenebene) sind über zonale Endpunkte verfügbar und werden als zonale Endpunktoperationen bezeichnet. API Beispiele für Operationen auf zonalen Endpunkten sind und. API `CreateSession` `PutObject`

Weitere Informationen zu den Endpunkten und den Standorten, die Directory-Buckets in Availability Zones unterstützen, finden Sie unter. Endpunkte für Verzeichnis-Buckets in Availability Zones

Konfiguration von Gateway-Endpunkten VPC

Informationen zur Konfiguration von VPC Gateway-Endpunkten für Access Directory-Buckets in Availability Zones finden Sie unter. the section called “Konfiguration von VPC Gateway-Endpunkten”

Regeln für die Benennung von Verzeichnis-Buckets

Wenn Sie einen Verzeichnis-Bucket in Amazon S3 erstellen, gelten dafür die folgenden Benennungsregeln. Benennungsregeln für Allzweck-Buckets finden Sie unterAllgemeine Benennungsregeln für Buckets.

Ein Verzeichnis-Bucket-Name besteht aus einem Basisnamen, den Sie angeben, und einem Suffix, das die ID der AWS Zone (einer Availability Zone) enthält, in der sich Ihr Bucket befindet, und`--x-s3`. Das Tool *`zone-id`* kann die ID einer Availability Zone sein.

```
base-name--zone-id--x-s3
```

Der folgende Verzeichnis-Bucket-Name enthält beispielsweise die Availability Zone-ID `usw2-az1`:

```
bucket-base-name--usw2-az1--x-s3
```

> **Note**
>
> Wenn Sie mithilfe der Konsole einen Verzeichnis-Bucket erstellen, wird dem von Ihnen angegebenen Basisnamen automatisch ein Suffix hinzugefügt. Dieses Suffix enthält die Zonen-ID der Zone (Availability Zone), die Sie ausgewählt haben.
> Wenn Sie einen Verzeichnis-Bucket mithilfe von erstellenAPI, müssen Sie das vollständige Suffix, einschließlich der Zonen-ID, in Ihrer Anfrage angeben. Eine Liste der Zonen finden Sie IDs unterEndpunkte.

Die folgenden Benennungsregeln gelten für Verzeichnis-Buckets.

- Seien Sie innerhalb der ausgewählten Zone (AWS Availability Zone) einzigartig.
- Der Name muss zwischen 3 (min) und 63 (maximal) Zeichen lang sein, einschließlich des Suffixes.
- Nur aus Kleinbuchstaben, Zahlen und Bindestrichen bestehen.
- Muss mit einer Zahl oder einem Buchstaben beginnen und enden.
- Muss das folgende Suffix enthalten:. `--`*`zone-id`*`--x-s3`
- Der Bucket-Name darf nicht mit dem Präfix `xn--` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `sthree-` beginnen.
- Der Bucket-Name darf nicht mit dem Präfix `sthree-configurator` beginnen.

- Der Bucket-Name darf nicht mit dem Präfix `amzn-s3-demo-` beginnen.
- Bucket-Namen dürfen nicht mit dem Suffix `-s3alias` enden. Dieses Suffix ist für Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter Verwenden eines Alias im Bucket-Stil für Ihren S3-Bucket-Zugriffspunkt.
- Bucket-Namen dürfen nicht mit dem Suffix `--ol-s3` enden. Dieses Suffix ist für Objekt-Lambda-Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter So verwenden Sie einen Alias im Bucket-Stil für den Object Lambda Access Point Ihres S3-Buckets.
- Bucket-Namen dürfen nicht mit dem Suffix `.mrap` enden. Dieses Suffix ist für Namen von Access Points mit mehreren Regionen reserviert. Weitere Informationen finden Sie unter Regeln zur Benennung von Amazon S3-Multi-Regions-Zugriffspunkten.

Anzeigen von Verzeichnis-Bucket-Eigenschaften

Sie können die Eigenschaften für einen Amazon S3 S3-Verzeichnis-Bucket mithilfe der Amazon S3 S3-Konsole anzeigen und konfigurieren. Weitere Informationen finden Sie unter Arbeiten mit Verzeichnis-Buckets.

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie die Registerkarte Verzeichnis-Buckets.
4. Wählen Sie in der Liste Verzeichnis-Buckets den Namen des Buckets aus, dessen Eigenschaften Sie anzeigen möchten.
5. Wählen Sie die Registerkarte Eigenschaften aus.
6. Auf der Registerkarte Eigenschaften können Sie die folgenden Eigenschaften für den Bucket einsehen:

    - Übersicht über den Verzeichnis-Bucket — Sie können die AWS-Region Zone (Availability Zone), den Amazon-Ressourcennamen (ARN) und das Erstellungsdatum für den Bucket sehen.
    - Serverseitige Verschlüsselungseinstellungen — Amazon S3 wendet serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für alle S3-Buckets an. Amazon S3 verschlüsselt ein

Objekt, bevor es auf einer Festplatte gespeichert wird, und entschlüsselt das Objekt beim Herunterladen. Weitere Informationen finden Sie unter Einstellung und Überwachung der Standardverschlüsselung für Verzeichnis-Buckets.

Weitere Informationen zu unterstützten Features für Verzeichnis-Buckets finden Sie unter Verzeichnis-Buckets erstellen und verwenden.

Verwaltung von Directory-Bucket-Richtlinien

Sie können Bucket-Richtlinien für Amazon S3 S3-Verzeichnis-Buckets hinzufügen, löschen, aktualisieren und anzeigen, indem Sie die Amazon S3 S3-Konsole, die AWS SDKs und die AWS CLI verwenden. Weitere Informationen finden Sie unter den folgenden Themen. Weitere Informationen zu unterstützten AWS Identity and Access Management (IAM) Aktionen finden Sie unterAutorisieren eines regionalen Endpunkts APIs mit IAM. Beispiele für Bucket-Richtlinien für Verzeichnis-Buckets finden Sie unter Beispiel für Bucket-Richtlinien für Verzeichnis-Buckets.

Themen

- Hinzufügen einer Bucket-Richtlinie
- Anzeigen einer Bucket-Richtlinie
- Löschen einer Bucket-Richtlinie

Hinzufügen einer Bucket-Richtlinie

Um eine Bucket-Richtlinie zu einem Verzeichnis-Bucket hinzuzufügen, können Sie die Amazon S3 S3-Konsole AWS SDKs, die oder die verwenden AWS CLI.

Verwenden der S3-Konsole

Eine Bucket-Richtlinie erstellen oder bearbeiten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie die Registerkarte Verzeichnis-Buckets.
4. Wählen Sie in der Liste Verzeichnis-Buckets den Namen des Buckets aus, in den Ihre Ordner oder Dateien hochgeladen werden sollen.

5. Wählen Sie die Registerkarte Berechtigungen.
6. Wählen Sie unter Bucket-Richtlinie Bearbeiten aus. Die Seite Bucket-Richtlinie bearbeiten wird angezeigt.
7. Um eine Richtlinie automatisch zu generieren, wählen Sie Richtliniengenerator.

   Wenn Sie den Richtliniengenerator wählen, wird der AWS Richtliniengenerator in einem neuen Fenster geöffnet.

   Wenn Sie den AWS Richtliniengenerator nicht verwenden möchten, können Sie im Bereich Richtlinien JSON Aussagen hinzufügen oder bearbeiten.

   a. Wählen Sie auf der Seite AWS Policy Generator (Richtliniengenerator) unter Select Type of Policy (Richtlinientyp auswählen) die Option S3 Bucket Policy (S3-Bucket-Richtlinie) aus.
   b. Fügen Sie eine Anweisung hinzu, indem Sie die Informationen in die bereitgestellten Felder eingeben, und wählen Sie dann Anweisung hinzufügen. Wiederholen Sie diesen Schritt für so viele Aussagen, wie Sie hinzufügen möchten. Weitere Informationen zu diesen Feldern finden Sie in der Referenz zu den IAM JSON Richtlinienelementen im IAMBenutzerhandbuch.

      > **Note**
      >
      > Der Einfachheit halber wird auf der Seite Bucket-Richtlinie bearbeiten der Bucket ARN (Amazon-Ressourcenname) des aktuellen Buckets über dem Policy-Textfeld angezeigt. Sie können dies ARN zur Verwendung in den Kontoauszügen auf der Seite AWS Policy Generator kopieren.

   c. Wenn Sie mit dem Hinzufügen von Anweisungen fertig sind, wählen Sie Generieren von Richtlinien.
   d. Kopieren Sie den generierten Richtlinientext, wählen Sie Schließen und kehren Sie zur Seite Bucket-Richtlinie bearbeiten in der Amazon-S3-Konsole zurück.
8. Bearbeiten Sie im Feld Richtlinie die bestehende Richtlinie oder fügen Sie die Bucket-Richtlinie aus dem AWS Policy Generator ein. Beheben Sie Sicherheitswarnungen, Fehler, allgemeine Warnungen und Vorschläge bevor Sie Ihre Richtlinie speichern.

   > **Note**
   >
   > Bucket-Richtlinien sind auf eine Größe von 20 KB beschränkt.

9. Wählen Sie Save changes (Änderungen speichern) aus, wodurch Sie zur Registerkarte Permissions (Berechtigungen) zurückkehren.

Verwenden Sie den AWS SDKs

SDK for Java 2.x

Example

`PutBucketPolicy` AWS SDK for Java 2.x

```
public static void setBucketPolicy(S3Client s3Client, String bucketName, String
 policyText) {

      //sample policy text
      /**
       * policy_statement = {
        *         'Version': '2012-10-17',
        *         'Statement': [
        *             {
        *                 'Sid': 'AdminPolicy',
        *                 'Effect': 'Allow',
        *                 'Principal': {
        *                     "AWS": "111122223333"
        *                 },
        *                 'Action': 's3express:*',
        *                 'Resource':
 'arn:aws:s3express:region:111122223333:bucket/bucket-base-name--zone-id--x-s3'
        *             }
        *         ]
        *     }
        */
        System.out.println("Setting policy:");
        System.out.println("----");
        System.out.println(policyText);
        System.out.println("----");
        System.out.format("On Amazon S3 bucket: \"%s\"\n", bucketName);

        try {
            PutBucketPolicyRequest policyReq = PutBucketPolicyRequest.builder()
                    .bucket(bucketName)
                    .policy(policyText)
                    .build();
```

```
            s3Client.putBucketPolicy(policyReq);
            System.out.println("Done!");
        }

        catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

Mit dem AWS CLI

Dieses Beispiel zeigt, wie Sie mithilfe von eine Bucket-Richtlinie zu einem Verzeichnis-Bucket hinzufügen AWS CLI. Um den Befehl zu verwenden, ersetzen Sie den *`user input placeholders`* mit Ihren eigenen Informationen.

```
aws s3api put-bucket-policy --bucket bucket-base-name--zone-id--x-s3 --policy file://
bucket_policy.json
```

bucket_policy.json:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AdminPolicy",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111122223333"
            },
            "Action": "s3express*",
            "Resource": "arn:aws:s3express:us-west-2:111122223333:bucket/amzn-s3-demo-
bucket--usw2-az1--x-s3"
        }
    ]
}
```

Weitere Informationen finden Sie in der. put-bucket-policy AWS Command Line Interface

Anzeigen einer Bucket-Richtlinie

Verwenden Sie die folgenden Beispiele, um eine Bucket-Richtlinie für einen Verzeichnis-Bucket anzuzeigen.

Verwenden von AWS CLI

Dieses Beispiel zeigt, wie die einem Verzeichnis-Bucket zugeordnete Bucket-Richtlinie mithilfe von angezeigt wird AWS CLI. Um den Befehl zu verwenden, ersetzen Sie den *`user input placeholders`* mit Ihren eigenen Informationen.

```
aws s3api get-bucket-policy --bucket bucket-base-name--zone-id--x-s3
```

Weitere Informationen finden Sie get-bucket-policyin der AWS Command Line Interface.

Löschen einer Bucket-Richtlinie

Verwenden Sie die folgenden Beispiele, um eine Bucket-Richtlinie für einen Verzeichnis-Bucket zu löschen.

Verwenden der AWS SDKs

SDK for Java 2.x

Example

`DeleteBucketPolicy` AWS SDK for Java 2.x

```
public static void deleteBucketPolicy(S3Client s3Client, String bucketName) {
      try {
          DeleteBucketPolicyRequest deleteBucketPolicyRequest =
 DeleteBucketPolicyRequest
                  .builder()
                  .bucket(bucketName)
                  .build()
          s3Client.deleteBucketPolicy(deleteBucketPolicyRequest);
          System.out.println("Successfully deleted bucket policy");
      }

      catch (S3Exception e) {
          System.err.println(e.awsErrorDetails().errorMessage());
          System.exit(1);
```

```
    }
```

Mit dem AWS CLI

Dieses Beispiel zeigt, wie eine Bucket-Richtlinie für einen Verzeichnis-Bucket mithilfe von gelöscht wird AWS CLI. Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```
aws s3api delete-bucket-policy --bucket bucket-base-name--zone-id--x-s3
```

Weitere Informationen finden Sie delete-bucket-policyin der AWS Command Line Interface.

Leeren eines Verzeichnis-Buckets

Sie können einen Amazon S3 S3-Verzeichnis-Bucket mithilfe der Amazon S3 S3-Konsole leeren. Weitere Information zu Verzeichnis-Buckets finden Sie unter Arbeiten mit Verzeichnis-Buckets.

Bevor Sie einen Verzeichnis-Bucket leeren, beachten Sie Folgendes:

- Wenn Sie einen Verzeichnis-Bucket leeren, löschen Sie alle Objekte, behalten aber den Bucket selbst.
- Nachdem Sie einen Verzeichnis-Bucket geleert haben, kann die leere Aktion nicht rückgängig gemacht werden.
- Objekte, die dem Verzeichnis-Bucket hinzugefügt werden, während die Aktion „Leerer Bucket“ ausgeführt wird, werden möglicherweise gelöscht.

Wenn Sie den Bucket auch löschen möchten, beachten Sie Folgendes:

- Alle Objekte in einem Bucket müssen gelöscht werden, bevor der Bucket selbst gelöscht werden kann.
- Laufende mehrteilige Uploads in dem Verzeichnis-Bucket müssen abgebrochen werden, bevor der Bucket selbst gelöscht werden kann.

> **Note**
>
> Der `s3 rm` Befehl über AWS Command Line Interface (CLI), die `delete` Operation über Mountpoint und die Optionsschaltfläche „Bucket leeren“ über können in Bearbeitung

befindliche mehrteilige Uploads in einem Verzeichnis-Bucket nicht löschen. AWS Management Console Um diese laufenden mehrteiligen Uploads zu löschen, verwenden Sie den `ListMultipartUploads` Vorgang, um die in Bearbeitung befindlichen mehrteiligen Uploads im Bucket aufzulisten, und verwenden Sie den Vorgang, um alle laufenden mehrteiligen Uploads abzubrechen. `AbortMultipartUpload`

Informationen zum Löschen eines Verzeichnis-Buckets finden Sie unter Löschen eines Verzeichnis-Buckets. Informationen zum Abbrechen eines laufenden mehrteiligen Uploads finden Sie unter. the section called “Abbrechen eines mehrteiligen Uploads”

Informationen zum Leeren eines Allzweck-Buckets finden Sie unter Leeren eines Buckets.

Verwenden der S3-Konsole

So leeren Sie einen Verzeichnis-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie die Registerkarte Verzeichnis-Buckets.
4. Wählen Sie das Optionsfeld neben dem Namen des Buckets, den Sie leeren möchten, und wählen Sie dann Leer aus.
5. Bestätigen Sie auf der Seite Empty bucket (Bucket leeren), dass Sie den Bucket leeren möchten, indem Sie **`permanently delete`** in das Textfeld eingeben und dann Empty (Leeren) auswählen.
6. Überwachen Sie den Fortschritt beim Entleeren des Buckets auf der Seite Leerer Bucket: Status.

Löschen eines Verzeichnis-Buckets

Sie können nur leere Amazon S3 S3-Verzeichnis-Buckets löschen. Bevor Sie Ihren Verzeichnis-Bucket löschen, müssen Sie alle Objekte im Bucket löschen und alle laufenden mehrteiligen Uploads abbrechen.

Informationen zum Leeren eines Verzeichnis-Buckets finden Sie unter Leeren eines Verzeichnis-Buckets. Informationen zum Abbrechen eines laufenden mehrteiligen Uploads finden Sie unter. the section called “Abbrechen eines mehrteiligen Uploads”

Informationen zum Löschen eines Allzweck-Buckets finden Sie unter Löschen eines Buckets.

Verwenden der S3-Konsole

Nachdem Sie Ihren Verzeichnis-Bucket geleert und alle laufenden mehrteiligen Uploads abgebrochen haben, können Sie Ihren Bucket löschen.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie die Registerkarte Verzeichnis-Buckets.
4. Wählen Sie in der Liste der Directory-Buckets das Optionsfeld neben dem Bucket aus, den Sie löschen möchten.
5. Wählen Sie Löschen.
6. Geben Sie auf der Seite „Bucket löschen“ den Namen des Buckets in das Textfeld ein, um das Löschen Ihres Buckets zu bestätigen.

   > **Important**
   >
   > Das Löschen eines Verzeichnis-Buckets kann nicht rückgängig gemacht werden.

7. Um Ihren Verzeichnis-Bucket zu löschen, wählen Sie Bucket löschen aus.

Verwenden Sie den AWS SDKs

In den folgenden Beispielen wird ein Verzeichnis-Bucket mithilfe von AWS SDK for Java 2.x und gelöscht AWS SDK for Python (Boto3).

SDK for Java 2.x

Example

```
public static void deleteBucket(S3Client s3Client, String bucketName) {

    try {
        DeleteBucketRequest del = DeleteBucketRequest.builder()
                .bucket(bucketName)
                .build();
        s3Client.deleteBucket(del);
```

```
            System.out.println("Bucket " + bucketName + " has been deleted");
        }
        catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

SDK for Python

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def delete_bucket(s3_client, bucket_name):
    '''
    Delete a directory bucket in a specified Region

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket to delete; for example, 'doc-example-bucket--usw2-
az1--x-s3'
    :return: True if bucket is deleted, else False
    '''

    try:
        s3_client.delete_bucket(Bucket = bucket_name)
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    bucket_name = 'BUCKET_NAME'
    region = 'us-west-2'
    s3_client = boto3.client('s3', region_name = region)
```

Mit dem AWS CLI

Dieses Beispiel zeigt, wie ein Verzeichnis-Bucket mithilfe von gelöscht wird AWS CLI. Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```
aws s3api delete-bucket --bucket bucket-base-name--zone-id--x-s3 --region us-west-2
```

Weitere Informationen finden Sie unter delete-bucket im. AWS Command Line Interface

Auflisten von Verzeichnis-Buckets

Die folgenden Beispiele zeigen, wie Verzeichnis-Buckets mithilfe von und aufgelistet werden. AWS SDKs AWS CLI

Mit dem AWS SDKs

SDK for Java 2.x

Example

Das folgende Beispiel listet Verzeichnis-Buckets mithilfe von auf AWS SDK for Java 2.x.

```
public static void listBuckets(S3Client s3Client) {
       try {
           ListDirectoryBucketsRequest listDirectoryBucketsRequest =
ListDirectoryBucketsRequest.builder().build();
           ListDirectoryBucketsResponse response =
s3Client.listDirectoryBuckets(listDirectoryBucketsRequest);
           if (response.hasBuckets()) {
               for (Bucket bucket: response.buckets()) {
                   System.out.println(bucket.name());
                   System.out.println(bucket.creationDate());
               }
           }
       }

       catch (S3Exception e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

SDK for Python

Example

Im folgenden Beispiel werden Verzeichnis-Buckets mithilfe von aufgelistet. AWS SDK for Python (Boto3)

```
import logging
import boto3
from botocore.exceptions import ClientError

def list_directory_buckets(s3_client):
'''
Prints a list of all directory buckets in a Region

:param s3_client: boto3 S3 client
:return: True if there are buckets in the Region, else False
'''
try:
    response = s3_client.list_directory_buckets()
    for bucket in response['Buckets']:
        print (bucket['Name'])
except ClientError as e:
    logging.error(e)
    return False
return True


if __name__ == '__main__':
    region = 'us-east-1'
    s3_client = boto3.client('s3', region_name = region)
    list_directory_buckets(s3_client)
```

AWS SDK for .NET

Example

Im folgenden Beispiel werden Verzeichnis-Buckets mithilfe von aufgelistet. AWS SDK for .NET

```
var listDirectoryBuckets = await amazonS3Client.ListDirectoryBucketsAsync(new
 ListDirectoryBucketsRequest
```

```
{
  MaxDirectoryBuckets = 10
  }).ConfigureAwait(false);
```

SDK for PHP

Example

Im folgenden Beispiel werden Verzeichnis-Buckets mithilfe von aufgelistet. AWS SDK for PHP

```
require 'vendor/autoload.php';

$s3Client = new S3Client([
    'region'      => 'us-east-1',
]);
$result = $s3Client->listDirectoryBuckets();
```

SDK for Ruby

Example

Im folgenden Beispiel werden Verzeichnis-Buckets mithilfe von aufgelistet. AWS SDK for Ruby

```
s3 = Aws::S3::Client.new(region:'us-west-1')
s3.list_directory_buckets
```

Mit dem AWS CLI

Der folgende `list-directory-buckets` Beispielbefehl zeigt, wie Sie mit AWS CLI dem Ihre Verzeichnis-Buckets auflisten können *`us-east-1`* Region. Zum Ausführen dieses Befehls ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws s3api list-directory-buckets --region us-east-1
```

Weitere Informationen finden Sie unter list-directory-bucketsin der AWS CLI Befehlsreferenz.

Feststellen, ob Sie auf einen Verzeichnis-Bucket zugreifen können

Die folgenden AWS SDK Beispiele zeigen, wie Sie den `HeadBucket` API Vorgang verwenden, um festzustellen, ob ein Amazon S3 S3-Verzeichnis-Bucket vorhanden ist und ob Sie die Berechtigung haben, darauf zuzugreifen.

Mit dem AWS SDKs

Das folgende AWS SDK for Java 2.x Beispiel zeigt, wie Sie feststellen können, ob ein Bucket vorhanden ist und ob Sie über die erforderlichen Zugriffsrechte verfügen.

SDK for Java 2.x

Example

AWS SDK for Java 2.x

```
public static void headBucket(S3Client s3Client, String bucketName) {
   try {
        HeadBucketRequest headBucketRequest = HeadBucketRequest
                .builder()
                .bucket(bucketName)
                .build();
        s3Client.headBucket(headBucketRequest);
        System.out.format("Amazon S3 bucket: \"%s\" found.", bucketName);
   }

   catch (S3Exception e) {
       System.err.println(e.awsErrorDetails().errorMessage());
       System.exit(1);
   }
}
```

Mit dem AWS CLI

Der folgende `head-bucket` Beispielbefehl zeigt, wie Sie mit dem ermitteln können AWS CLI , ob ein Verzeichnis-Bucket vorhanden ist und ob Sie über die erforderlichen Zugriffsrechte verfügen. Um diesen Befehl auszuführen, ersetzen Sie die Platzhalter für Benutzereingaben durch Ihre eigenen Informationen.

```
aws s3api head-bucket --bucket bucket-base-name--zone-id--x-s3
```

Weitere Informationen finden Sie unter head-bucketin der AWS CLI Befehlsreferenz.

Arbeiten mit Objekten in einem Verzeichnis-Bucket

Nachdem Sie einen Amazon S3 S3-Verzeichnis-Bucket erstellt haben, können Sie mit Objekten arbeiten, indem Sie die Amazon S3 S3-Konsole verwenden, AWS Command Line Interface (AWS CLI) und die AWS SDKs.

Weitere Informationen zur Durchführung von Massenvorgängen, zum Importieren, Hochladen, Kopieren, Löschen und Herunterladen von Objekten in Verzeichnis-Buckets finden Sie in den folgenden Themen.

Themen

- Importieren von Objekten in einen Verzeichnis-Bucket
- Arbeiten mit S3 Lifecycle für Verzeichnis-Buckets
- Verwenden von Batch Operations mit Verzeichnis-Buckets.
- Daten an Objekte in Verzeichnis-Buckets anhängen
- Objekte in einen Verzeichnis-Bucket hochladen
- Objekte aus oder in einen Verzeichnis-Bucket kopieren
- Objekte aus einem Verzeichnis-Bucket löschen
- Ein Objekt aus einem Verzeichnis-Bucket herunterladen
- Objektmetadaten aus Verzeichnis-Buckets abrufen

Importieren von Objekten in einen Verzeichnis-Bucket

Nachdem Sie einen Verzeichnis-Bucket in Amazon S3 erstellt haben, können Sie den neuen Bucket mithilfe der Importaktion mit Daten füllen. Der Import ist eine optimierte Methode zur Erstellung von S3-Batch-Operations-Aufträgen zum Kopieren von Objekten aus Allzweck-Buckets in Verzeichnis-Buckets.

> **Note**
>
> Die folgenden Einschränkungen gelten für Importaufträge:

- Der Quell-Bucket und der Ziel-Bucket müssen sich im selben befinden AWS-Region und Konto.
- Der Quell-Bucket kann kein Verzeichnis-Bucket sein.
- Objekte, die größer als 5 GB sind, werden nicht unterstützt und beim Kopiervorgang nicht berücksichtigt.
- Objekte der Speicherklassen Glacier Flexible Retrieval, Glacier Deep Archive, Intelligent-Tiering Archive Access und Intelligent-Tiering Deep Archive müssen wiederhergestellt werden, bevor sie importiert werden können.
- Importierte Objekte mit MD5 Prüfsummenalgorithmen werden so konvertiert, dass sie CRC32 Prüfsummen verwenden.
- Importierte Objekte verwenden die Speicherklasse Express One Zone, die eine andere Preisstruktur hat als die Speicherklassen, die in Buckets für allgemeine Zwecke verwendet werden. Beachten Sie diesen Kostenunterschied, wenn Sie eine große Anzahl von Objekten importieren.

Wenn Sie einen Importauftrag konfigurieren, geben Sie den Quell-Bucket oder das Quellpräfix an, aus dem die vorhandenen Objekte kopiert werden. Sie stellen auch eine AWS Identity and Access Management (IAM) Rolle, die über Berechtigungen für den Zugriff auf die Quellobjekte verfügt. Amazon S3 startet dann einen Batch-Operations-Auftrag, der die Objekte kopiert und automatisch die entsprechenden Speicherklassen- und Prüfsummeneinstellungen anwendet.

Um Importaufträge zu konfigurieren, verwenden Sie die Amazon-S3-Konsole.

Verwenden der Amazon S3-Konsole

So importieren Sie Objekte in einen Verzeichnis-Bucket

1. Melden Sie sich an bei AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets und dann die Registerkarte Verzeichnis-Buckets aus. Wählen Sie die Optionsschaltfläche neben dem Verzeichnis-Bucket, in den Sie Objekte importieren möchten.
3. Wählen Sie Importieren aus.

4. Geben Sie unter Quelle den Allzweck-Bucket (oder den Bucket-Pfad einschließlich Präfix) ein, der die Objekte enthält, die Sie importieren möchten. Um einen vorhandenen Allzweck-Bucket aus einer Liste auszuwählen, wählen Sie S3 durchsuchen.
5. Gehen Sie für die Berechtigung zum Zugreifen auf und Kopieren von Quellobjekten auf eine der folgenden Arten vor, um eine IAM Rolle mit den erforderlichen Berechtigungen für den Import Ihrer Quellobjekte anzugeben:

   - Um Amazon S3 zu ermöglichen, in Ihrem Namen eine neue IAM Rolle zu erstellen, wählen Sie Neue IAM Rolle erstellen.
   - Um eine bestehende IAM Rolle aus einer Liste auszuwählen, wählen Sie Aus vorhandenen IAM Rollen auswählen.
   - Um eine bestehende IAM Rolle durch Eingabe ihres Amazon-Ressourcennamens (ARN) anzugeben, wählen Sie IAM Rolle eingeben ARN und geben Sie dann das ARN in das entsprechende Feld ein.
6. Überprüfen Sie die Informationen, die in den Abschnitten Ziel und Kopierte Objekteinstellungen angezeigt werden. Wenn die Informationen im Bereich Ziel korrekt sind, wählen Sie Import, um den Kopierauftrag zu starten.

   Die Amazon-S3-Konsole zeigt den Status Ihres neuen Auftrags auf der Seite Batch-Operationen an. Für weitere Informationen über den Auftrag klicken Sie auf das Optionsfeld neben dem Auftragsnamen und wählen dann im Menü Aktionen die Option Details anzeigen. Um den Verzeichnis-Bucket zu öffnen, in den die Objekte importiert werden, wählen Sie Importziel anzeigen.

Arbeiten mit S3 Lifecycle für Verzeichnis-Buckets

S3 Lifecycle hilft Ihnen dabei, Objekte in S3 Express One Zone kostengünstig in Verzeichnis-Buckets zu speichern, indem abgelaufene Objekte in Ihrem Namen gelöscht werden. Um den Lebenszyklus Ihrer Objekte zu verwalten, erstellen Sie eine S3-Lifecycle-Konfiguration für Ihren Directory-Bucket. Eine S3-Lebenszyklus-Konfiguration besteht aus einer Reihe von Regeln, mit denen Aktionen definiert werden, die Amazon S3 auf eine Gruppe von Objekten anwendet. Sie können eine Amazon S3-Lifecycle-Konfiguration für einen Verzeichnis-Bucket einrichten, indem Sie die AWS Befehlszeilenschnittstelle (AWS CLI) AWS SDKs, den Amazon S3 REST API und verwenden AWS CloudFormation.

In Ihrer Lebenszykluskonfiguration verwenden Sie Regeln, um Aktionen zu definieren, die Amazon S3 für Ihre Objekte ausführen soll. Für Objekte, die in Verzeichnis-Buckets gespeichert sind, können

Sie Lebenszyklusregeln erstellen, damit Objekte mit zunehmendem Alter ablaufen. Sie können auch Lebenszyklusregeln erstellen, um unvollständige mehrteilige Uploads in Verzeichnis-Buckets täglich zu löschen.

Wenn Sie einem Bucket eine Lebenszyklus-Konfiguration hinzufügen, gelten die Konfigurationsregeln für vorhandene Objekte und für Objekte, die Sie später hinzufügen. Wenn Sie heute beispielsweise eine Lebenszyklus-Konfigurationsregel mit einer Ablaufaktion hinzufügen, die bewirkt, dass Objekte mit einem bestimmten Präfix 30 Tage nach der Erstellung ablaufen, stellt S3 eine Warteschlange für die Entfernung aller vorhandenen Objekte, die älter als 30 Tage sind und das angegebene Präfix haben, in eine Warteschlange.

Worin unterscheidet sich der S3-Lebenszyklus für Verzeichnis-Buckets

Für Objekte in Verzeichnis-Buckets können Sie Lebenszyklusregeln erstellen, um Objekte ablaufen zu lassen und unvollständige mehrteilige Uploads zu löschen. S3 Lifecycle für Verzeichnis-Buckets unterstützt jedoch keine Übergangsaktionen zwischen Speicherklassen.

CreateSession

Lifecycle verwendet öffentliche `DeleteObject` `DeleteObjects` API Operationen, um Objekte in Verzeichnis-Buckets ablaufen zu lassen. Um diese API Operationen zu verwenden, verwendet S3 Lifecycle die, `CreateSession` API um temporäre Sicherheitsanmeldedaten für den Zugriff auf die Objekte in den Verzeichnis-Buckets einzurichten. Weitere Informationen finden Sie `CreateSession`in der Amazon S3 API S3-Referenz.

Wenn Sie über eine aktive Richtlinie verfügen, die dem Lifecycle-Principal Löschberechtigungen verweigert, verhindert dies, dass Sie S3 Lifecycle erlauben, Objekte in Ihrem Namen zu löschen.

Verwenden Sie eine Bucket-Richtlinie, um dem S3 Lifecycle-Dienstprinzipal Berechtigungen zu erteilen

Die folgende Bucket-Richtlinie gewährt die Erlaubnis, CreateSession Aufrufe mit der `ReadWrite` Standardsitzung zuzulassen, und erlaubt dem Lifecycle-Serviceprinzipal.

Example – Bucket-Richtlinie, um **`CreateSession`**-Aufrufe mit der **`ReadWrite`**-Standardsitzung zuzulassen

```
{
  "Version":"2008-10-17",
```

```
   "Statement":[
      {
         "Effect":"Allow",
         "Principal": {
            "Service":"lifecycle.s3.amazonaws.com"
          },
          "Action":"s3express:CreateSession",
          "Condition": {
             "StringEquals": {
                "s3express:SessionMode": "ReadWrite"
              }
           },
           "Resource":"arn:aws:s3express:us-east-2:412345678921:bucket/amzn-s3-demo-
bucket--use2-az2--x-s3"
       }
   ]
}
```

Überwachung der Lebenszyklusregeln

Für Objekte, die in Verzeichnis-Buckets gespeichert sind, generiert S3 Lifecycle AWS CloudTrail Verwaltungs- und Datenereignisprotokolle. Weitere Informationen finden Sie in den Beispielen für CloudTrail Protokolldateien für S3 Express One Zone.

Weitere Informationen zur Erstellung von Lebenszykluskonfigurationen und zur Behebung von Problemen im Zusammenhang mit S3 Lifecycle finden Sie in den folgenden Themen:

Themen

- Eine Lifecycle-Konfiguration für Ihren Directory-Bucket erstellen und verwalten
- Behebung von S3-Lifecycle-Problemen für Directory-Buckets

Eine Lifecycle-Konfiguration für Ihren Directory-Bucket erstellen und verwalten

Sie können eine Lebenszykluskonfiguration für Verzeichnis-Buckets erstellen, indem Sie AWS Command Line Interface (AWS CLI) AWS SDKs und REST APIs verwenden.

Mit dem AWS CLI

Sie können die folgenden AWS CLI Befehle verwenden, um S3 Lifecycle-Konfigurationen zu verwalten:

- `put-bucket-lifecycle-configuration`
- `get-bucket-lifecycle-configuration`
- `delete-bucket-lifecycle`

Anweisungen zur Einrichtung von finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS CLI in der Amazon S3 API S3-Referenz. AWS CLI

Die Amazon S3 Lifecycle-Konfiguration ist eine XML Datei. Aber wenn Sie die verwenden AWS CLI, können Sie das XML Format nicht angeben. Sie müssen stattdessen das JSON Format angeben. Im Folgenden finden Sie Beispiele für XML Lebenszykluskonfigurationen und die entsprechenden JSON Konfigurationen, die Sie in einem AWS CLI Befehl angeben können.

Im folgenden AWS CLI Beispiel wird eine Lebenszykluskonfigurationsrichtlinie auf einen Verzeichnis-Bucket angewendet. Diese Richtlinie legt fest, dass alle Objekte mit dem markierten Präfix (myprefix) und der definierten Objektgröße nach 7 Tagen ablaufen. Um dieses Beispiel zu verwenden, ersetzen Sie jedes *`user input placeholder`* mit Ihren eigenen Informationen.

Speichern Sie die Lebenszyklus-Konfigurationsrichtlinie in einer JSON Datei. In diesem Beispiel hat die Datei den Namen lifecycle cycle1.json.

Example

JSON

```
{
    "Rules": [
        {
        "Expiration": {
            "Days": 7
        },
        "ID": "Lifecycle expiration rule",
        "Filter": {
            "And": {
                "Prefix": "myprefix/",
                "ObjectSizeGreaterThan": 500,
                "ObjectSizeLessThan": 64000
            }
        },
        "Status": "Enabled"
    }
```

```
    ]
}
```

Reichen Sie die JSON Datei als Teil des Befehls ein. `put-bucket-lifecycle-configuration` CLI Um diesen Befehl zu verwenden, ersetzen Sie jeden *`user input placeholder`* mit Ihren eigenen Informationen.

```
aws s3api put-bucket-lifecycle-configuration --region us-west-2 --profile default
 --bucket amzn-s3-demo-bucket--usw2-az1--x-s3 --lifecycle-configuration file://lc-
policy.json --checksum-algorithm crc32c
```

Example

XML

```
<LifecycleConfiguration>
    <Rule>
        <ID>Lifecycle expiration rule</ID>
        <Filter>
            <And>
                <Prefix>myprefix/</Prefix>
                <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
                <ObjectSizeLessThan>64000</ObjectSizeLessThan>
            </And>
        </Filter>
        <Status>Enabled</Status>
        <Expiration>
             <Days>7</Days>
        </Expiration>
    </Rule>
</LifecycleConfiguration>
```

Mit dem AWS SDKs

SDK for Java

Example

```
import
 software.amazon.awssdk.services.s3.model.PutBucketLifecycleConfigurationRequest;
```

```
import
 software.amazon.awssdk.services.s3.model.PutBucketLifecycleConfigurationResponse;
import software.amazon.awssdk.services.s3.model.ChecksumAlgorithm;
import software.amazon.awssdk.services.s3.model.BucketLifecycleConfiguration;
import software.amazon.awssdk.services.s3.model.LifecycleRule;
import software.amazon.awssdk.services.s3.model.LifecycleRuleFilter;
import software.amazon.awssdk.services.s3.model.LifecycleExpiration;
import software.amazon.awssdk.services.s3.model.LifecycleRuleAndOperator;
import
 software.amazon.awssdk.services.s3.model.GetBucketLifecycleConfigurationResponse;
import
 software.amazon.awssdk.services.s3.model.GetBucketLifecycleConfigurationRequest;
import software.amazon.awssdk.services.s3.model.DeleteBucketLifecycleRequest;
import software.amazon.awssdk.services.s3.model.DeleteBucketLifecycleResponse;
import software.amazon.awssdk.services.s3.model.AbortIncompleteMultipartUpload;

// PUT a Lifecycle policy
LifecycleRuleFilter objectExpirationFilter =
 LifecycleRuleFilter.builder().and(LifecycleRuleAndOperator.builder().prefix("dir1/").object
LifecycleRuleFilter mpuExpirationFilter =
 LifecycleRuleFilter.builder().prefix("dir2/").build();

LifecycleRule objectExpirationRule =
 LifecycleRule.builder().id("lc").filter(objectExpirationFilter).status("Enabled").expiratio
                    .days(10)
                    .build())
                .build();
LifecycleRule mpuExpirationRule = LifecycleRule.builder().id("lc-
mpu").filter(mpuExpirationFilter).status("Enabled").abortIncompleteMultipartUpload(AbortInco
                        .daysAfterInitiation(10)
                        .build())
                .build();

PutBucketLifecycleConfigurationRequest putLifecycleRequest =
 PutBucketLifecycleConfigurationRequest.builder()
                .bucket("amzn-s3-demo-bucket--usw2-az1--x-s3")
                .checksumAlgorithm(ChecksumAlgorithm.CRC32)
                .lifecycleConfiguration(
                        BucketLifecycleConfiguration.builder()
                                .rules(objectExpirationRule, mpuExpirationRule)
                                .build()
                ).build();
```

```
PutBucketLifecycleConfigurationResponse resp =
 client.putBucketLifecycleConfiguration(putLifecycleRequest);

// GET the Lifecycle policy
GetBucketLifecycleConfigurationResponse getResp =
 client.getBucketLifecycleConfiguration(GetBucketLifecycleConfigurationRequest.builder().buc
s3-demo-bucket--usw2-az1--x-s3").build());

// DELETE the Lifecycle policy
DeleteBucketLifecycleResponse delResp =
 client.deleteBucketLifecycle(DeleteBucketLifecycleRequest.builder().bucket("amzn-
s3-demo-bucket--usw2-az1--x-s3").build());
```

SDK for Go

Example

```
package main

import (
    "context"
    "log"

    "github.com/aws/aws-sdk-go-v2/aws"
    "github.com/aws/aws-sdk-go-v2/config"
    "github.com/aws/aws-sdk-go-v2/service/s3"
    "github.com/aws/aws-sdk-go-v2/service/s3/types"
)
// PUT a Lifecycle policy
func putBucketLifecycleConfiguration(client *s3.Client, bucketName string ) error {
    lifecycleConfig := &s3.PutBucketLifecycleConfigurationInput{
        Bucket: aws.String(bucketName),
        LifecycleConfiguration: &types.BucketLifecycleConfiguration{
            Rules: []types.LifecycleRule{
                {
                    ID:     aws.String("lc"),
                    Filter: &types.LifecycleRuleFilter{
                        And: &types.LifecycleRuleAndOperator{
                            Prefix: aws.String("foo/"),
                            ObjectSizeGreaterThan: aws.Int64(1000000),
                            ObjectSizeLessThan:    aws.Int64(100000000),
                            },
                        },
```

```
                    Status: types.ExpirationStatusEnabled,
                    Expiration: &types.LifecycleExpiration{
                        Days: aws.Int32(int32(1)),
                    },
                },
                {
                    ID:     aws.String("abortmpu"),
                    Filter: &types.LifecycleRuleFilter{
                        Prefix: aws.String("bar/"),
                    },
                    Status: types.ExpirationStatusEnabled,
                    AbortIncompleteMultipartUpload:
 &types.AbortIncompleteMultipartUpload{
                        DaysAfterInitiation: aws.Int32(int32(5)),
                    },
                },
            },
        },
    }
    _, err := client.PutBucketLifecycleConfiguration(context.Background(),
 lifecycleConfig)
    return err
}
// Get the Lifecycle policy
func getBucketLifecycleConfiguration(client *s3.Client, bucketName string) error {
    getLifecycleConfig := &s3.GetBucketLifecycleConfigurationInput{
        Bucket: aws.String(bucketName),
    }

    resp, err := client.GetBucketLifecycleConfiguration(context.Background(),
 getLifecycleConfig)
    if err != nil {
        return err
    }
    return nil
}
// Delete the Lifecycle policy
func deleteBucketLifecycleConfiguration(client *s3.Client, bucketName string) error
 {
    deleteLifecycleConfig := &s3.DeleteBucketLifecycleInput{
        Bucket: aws.String(bucketName),
    }
    _, err := client.DeleteBucketLifecycle(context.Background(),
 deleteLifecycleConfig)
```

```
    return err
}
func main() {
    cfg, err := config.LoadDefaultConfig(context.Background(),
 config.WithRegion("us-west-2")) // Specify your region here
    if err != nil {
        log.Fatalf("unable to load SDK config, %v", err)
    }
    s3Client := s3.NewFromConfig(cfg)
    bucketName := "amzn-s3-demo-bucket--usw2-az1--x-s3"
    putBucketLifecycleConfiguration(s3Client, bucketName)
    getBucketLifecycleConfiguration(s3Client, bucketName)
    deleteBucketLifecycleConfiguration(s3Client, bucketName)
    getBucketLifecycleConfiguration(s3Client, bucketName)
}
```

SDK for .NET

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class LifecycleTest
    {
        private const string bucketName = "amzn-s3-demo-bucket--usw2-az1--x-s3";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;
        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            AddUpdateDeleteLifecycleConfigAsync().Wait();
        }

        private static async Task AddUpdateDeleteLifecycleConfigAsync()
        {
```

```
            try
            {
                var lifeCycleConfiguration = new LifecycleConfiguration()
                {
                    Rules = new List <LifecycleRule>
                        {
                            new LifecycleRule
                            {
                                 Id = "delete rule",
                                  Filter = new LifecycleFilter()
                                 {
                                     LifecycleFilterPredicate = new
LifecyclePrefixPredicate()
                                     {
                                         Prefix = "projectdocs/"
                                     }
                                 },
                                 Status = LifecycleRuleStatus.Enabled,
                                 Expiration = new LifecycleRuleExpiration()
                                 {
                                       Days = 10
                                 }
                            }
                        }
                };

                // Add the configuration to the bucket.
                await AddExampleLifecycleConfigAsync(client,
lifeCycleConfiguration);

                // Retrieve an existing configuration.
                lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);

                // Add a new rule.
                lifeCycleConfiguration.Rules.Add(new LifecycleRule
                {
                    Id = "mpu abort rule",
                    Filter = new LifecycleFilter()
                    {
                        LifecycleFilterPredicate = new LifecyclePrefixPredicate()
                        {
                            Prefix = "YearlyDocuments/"
                        }
                    },
```

```
                    Expiration = new LifecycleRuleExpiration()
                    {
                        Days = 10
                    },
                    AbortIncompleteMultipartUpload = new
LifecycleRuleAbortIncompleteMultipartUpload()
                    {
                        DaysAfterInitiation = 10
                    }
                });

                // Add the configuration to the bucket.
                await AddExampleLifecycleConfigAsync(client,
lifeCycleConfiguration);

                // Verify that there are now two rules.
                lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);
                Console.WriteLine("Expected # of rulest=2; found:{0}",
lifeCycleConfiguration.Rules.Count);

                // Delete the configuration.
                await RemoveLifecycleConfigAsync(client);

                // Retrieve a nonexistent configuration.
                lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);

            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered ***. Message:'{0}' when writing
an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }

        static async Task AddExampleLifecycleConfigAsync(IAmazonS3 client,
LifecycleConfiguration configuration)
        {
```

```
            PutLifecycleConfigurationRequest request = new
 PutLifecycleConfigurationRequest
            {
                BucketName = bucketName,
                Configuration = configuration
            };
            var response = await client.PutLifecycleConfigurationAsync(request);
        }

        static async Task <LifecycleConfiguration>
 RetrieveLifecycleConfigAsync(IAmazonS3 client)
        {
            GetLifecycleConfigurationRequest request = new
 GetLifecycleConfigurationRequest
            {
                BucketName = bucketName
            };
            var response = await client.GetLifecycleConfigurationAsync(request);
            var configuration = response.Configuration;
            return configuration;
        }

        static async Task RemoveLifecycleConfigAsync(IAmazonS3 client)
        {
            DeleteLifecycleConfigurationRequest request = new
 DeleteLifecycleConfigurationRequest
            {
                BucketName = bucketName
            };
            await client.DeleteLifecycleConfigurationAsync(request);
        }
    }
}
```

SDK for Python

Example

```
import boto3

client = boto3.client("s3", region_name="us-west-2")
bucket_name = 'amzn-s3-demo-bucket--usw2-az1--x-s3'

client.put_bucket_lifecycle_configuration(
```

```
    Bucket=bucket_name,
    ChecksumAlgorithm='CRC32',
    LifecycleConfiguration={
        'Rules': [
            {
                'ID': 'lc',
                'Filter': {
                    'And': {
                        'Prefix': 'foo/',
                        'ObjectSizeGreaterThan': 1000000,
                        'ObjectSizeLessThan': 100000000,
                    }
                },
                'Status': 'Enabled',
                'Expiration': {
                    'Days': 1
                }
            },
            {
                'ID': 'abortmpu',
                'Filter': {
                    'Prefix': 'bar/'
                },
                'Status': 'Enabled',
                'AbortIncompleteMultipartUpload': {
                    'DaysAfterInitiation': 5
                }
            }
        ]
    }
)

result = client.get_bucket_lifecycle_configuration(
    Bucket=bucket_name
)

client.delete_bucket_lifecycle(
    Bucket=bucket_name
)
```

Behebung von S3-Lifecycle-Problemen für Directory-Buckets

Themen

- Ich habe meine Lebenszykluskonfiguration eingerichtet, aber Objekte in meinem Verzeichnis-Bucket laufen nicht ab
- Wie überwache ich die Maßnahmen, die im Rahmen meiner Lebenszyklusregeln ergriffen wurden?

Ich habe meine Lebenszykluskonfiguration eingerichtet, aber Objekte in meinem Verzeichnis-Bucket laufen nicht ab

S3 Lifecycle für Verzeichnis-Buckets verwendet public, APIs um Objekte in S3 Express One Zone zu löschen. Um Public auf Objektebene zu verwendenAPIs, müssen Sie S3 Lifecycle die Erlaubnis zum Löschen Ihrer Objekte erteilen `CreateSession` und diese auch gewähren. Wenn Sie über eine aktive Richtlinie verfügen, die Löschungen verweigert, verhindert dies, dass Sie S3 Lifecycle erlauben, Objekte in Ihrem Namen zu löschen.

Es ist wichtig, dass Sie Ihre Bucket-Richtlinien korrekt konfigurieren, um sicherzustellen, dass die Objekte, die Sie löschen möchten, ablaufen können. Du kannst in deinen AWS CloudTrail Logs nach Aufrufen von `AccessDenied` `CreateSession` API Trails CloudTrail nach Aufrufen nachsehen, ob der Zugriff verweigert wurde. Das Überprüfen deiner CloudTrail Logs kann dir dabei helfen, Probleme mit dem Zugriff zu beheben und die Hauptursache für Fehler mit der Zugriffsverweigerung zu ermitteln. Anschließend können Sie Ihre falschen Zugriffskontrollen korrigieren, indem Sie die entsprechenden Richtlinien aktualisieren.

Wenn Sie sich vergewissern, dass Ihre Bucket-Richtlinien korrekt festgelegt sind und weiterhin Probleme auftreten, empfehlen wir Ihnen, die Lebenszyklusregeln zu überprüfen, um sicherzustellen, dass sie auf die richtige Teilmenge von Objekten angewendet werden.

Wie überwache ich die Maßnahmen, die im Rahmen meiner Lebenszyklusregeln ergriffen wurden?

Sie können AWS CloudTrail Datenereignisprotokolle verwenden, um die von S3 Lifecycle ausgeführten Aktionen in Verzeichnis-Buckets zu überwachen. Weitere Informationen finden Sie unter Beispiele für CloudTrail Protokolldateien.

Verwenden von Batch Operations mit Verzeichnis-Buckets.

Sie können Amazon S3 Batch Operations verwenden, um umfangreiche Vorgänge für in S3 gespeicherte Objekte durchzuführen. Weitere Informationen zu S3 Batch Operations finden Sie unter Durchführen umfangreicher Stapelvorgänge für Amazon-S3-Objekte.

In den folgenden Themen wird die Durchführung von Batch-Vorgängen für Objekte beschrieben, die in der Speicherklasse S3 Express One Zone in Verzeichnis-Buckets gespeichert sind.

Themen

- Verwenden von Batch Operations mit Verzeichnis-Buckets.
- Die wichtigsten Unterschiede:

Verwenden von Batch Operations mit Verzeichnis-Buckets.

Sie können die Operationen „Kopieren“ und „ AWS Lambda Funktion aufrufen“ für Objekte ausführen, die in Verzeichnis-Buckets gespeichert sind. Mit Copy können Sie Objekte zwischen Buckets desselben Typs kopieren (z. B. von einem Verzeichnis-Bucket in einen Verzeichnis-Bucket). Sie können auch zwischen Verzeichnis-Buckets und Allzweck-Buckets kopieren. Mit der AWS Lambda Funktion Invoke können Sie eine Lambda-Funktion verwenden, um Aktionen für Objekte in Ihrem Verzeichnis-Bucket mit von Ihnen definiertem Code auszuführen.

Objekte kopieren

Sie können zwischen demselben Bucket-Typ oder zwischen Verzeichnis-Buckets und Allzweck-Buckets kopieren. Wenn Sie in einen Verzeichnis-Bucket kopieren, müssen Sie das richtige Amazon-Ressourcenname (ARN) -Format für diesen Bucket-Typ verwenden. Das ARN Format für einen Verzeichnis-Bucket ist`arn:aws:s3express:`*`region`*`:`*`account-id`*`:bucket/`*`bucket-base-name`*`--x-s3`.

Sie können Ihren Verzeichnis-Bucket auch mit Daten füllen, indem Sie die Aktion Import in der S3-Konsole verwenden. Import ist eine optimierte Methode zur Erstellung von Batch-Operations-Aufträgen zum Kopieren von Objekten aus Allzweck-Buckets in Verzeichnis-Buckets. Für Import-Kopieraufträge aus Allzweck-Buckets in Verzeichnis-Buckets generiert S3 automatisch ein Manifest. Weitere Informationen finden Sie unter Objekte in einen Verzeichnis-Bucket importieren und Ein Manifest angeben.

Lambda-Funktionen () aufrufen **`LambdaInvoke`**

Für die Verwendung von Batch Operations zum Aufrufen von Lambda-Funktionen, die auf Verzeichnis-Buckets wirken, gelten besondere Anforderungen. Beispielsweise müssen Sie Ihre Lambda-Anfrage strukturieren, indem Sie ein v2 JSONAufrufschema und spezifizieren InvocationSchemaVersion 2.0 wenn Sie den Job erstellen. Weitere Informationen finden Sie unter AWS Lambda Funktion aufrufen.

Die wichtigsten Unterschiede:

Im Folgenden finden Sie eine Liste der wichtigsten Unterschiede, wenn Sie Batch Operations verwenden, um Massenoperationen an Objekten durchzuführen, die in Verzeichnis-Buckets mit der S3 Express One Zone-Speicherklasse gespeichert sind:

- Für Verzeichnis-Buckets werden SSE -S3 und serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-KMS) unterstützt. Wenn Sie eine `CopyObject` Anfrage stellen, die angibt, serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) für einen Verzeichnis-Bucket (Quelle oder Ziel) zu verwenden, gibt die Antwort einen Fehler zurück. HTTP `400 (Bad Request)`

  Wir empfehlen, dass die Standardverschlüsselung des Buckets die gewünschte Verschlüsselungskonfiguration verwendet und dass Sie die Standardverschlüsselung des Buckets in Ihren `CreateSession` Anfragen oder `PUT` Objektanforderungen nicht überschreiben. Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Weitere Informationen zum Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets und zum Verschlüsseln neuer Objektkopien in einem Verzeichnis-Bucket mit SSE - KMS finden Sie unter Serverseitige Verschlüsselung mit AWS KMS für neue Objekt-Uploads angeben.

  S3-Bucket-Keys werden nicht unterstützt, wenn Sie KMS verschlüsselte Objekte aus Allzweck-Buckets in SSE Verzeichnis-Buckets, von Verzeichnis-Buckets in Allzweck-Buckets oder zwischen Verzeichnis-Buckets kopieren, und zwar über Copy Betrieb in Batch-Operationen. In diesem Fall ruft Amazon S3 AWS KMS jedes Mal, wenn eine Kopieranforderung für ein KMS - verschlüsseltes Objekt gestellt wird, auf. Weitere Informationen zur Verwendung von SSE - in KMS Verzeichnis-Buckets finden Sie unter Einstellung und Überwachung der Standardverschlüsselung für Verzeichnis-Buckets und. Serverseitige Verschlüsselung mit AWS KMS Schlüsseln (SSE-KMS) in Verzeichnis-Buckets verwenden
- Objekte in Verzeichnis-Buckets können nicht mit Tags markiert werden. Sie können nur einen leeren Tag-Satz angeben. Standardmäßig kopiert Batch Operations Tags. Wenn Sie ein Objekt mit Tags zwischen Allzweck-Buckets und Verzeichnis-Buckets kopieren, erhalten Sie eine Antwort. `501 (Not Implemented)`
- S3 Express One Zone bietet Ihnen die Möglichkeit, den Prüfsummenalgorithmus auszuwählen, der zur Validierung Ihrer Daten bei Uploads oder Downloads verwendet wird. Sie können einen der folgenden Algorithmen zur Prüfung der Datenintegrität zwischen Secure Hash Algorithms

(SHA) oder Cyclic Redundancy Check (CRC) wählen:CRC32,CRC32, SHA -1 und -256. SHA MD5basierte Prüfsummen werden von der S3 Express One Zone-Speicherklasse nicht unterstützt.

- Standardmäßig setzen alle Amazon S3 S3-Buckets die Einstellung S3 Object Ownership auf Bucket Owner erforced und die Zugriffskontrolllisten (ACLs) sind deaktiviert. Für Verzeichnis-Buckets kann diese Einstellung nicht geändert werden. Sie können ein Objekt aus Allzweck-Buckets in Verzeichnis-Bucket kopieren. Sie können die Standardeinstellung jedoch nicht überschreiben, ACL wenn Sie in oder aus einem Verzeichnis-Bucket kopieren.
- Unabhängig davon, wie Sie Ihr Manifest angeben, muss die Liste selbst in einem Allzweck-Bucket gespeichert werden. Batch Operations kann keine vorhandenen Manifeste aus Verzeichnis-Buckets importieren (oder generierte Manifeste in Verzeichnis-Buckets speichern). Im Manifest beschriebene Objekte können jedoch in Verzeichnis-Buckets gespeichert werden.
- Batch Operations kann keinen Verzeichnis-Bucket als Speicherort in einem S3-Inventarbericht angeben. Inventarberichte unterstützen keine Verzeichnis-Buckets. Sie können eine Manifestdatei für Objekte innerhalb eines Verzeichnis-Buckets erstellen, indem Sie den `ListObjectsV2` API Vorgang zum Auflisten der Objekte verwenden. Anschließend können Sie die Liste in eine CSV Datei einfügen.

Gewähren von -Zugriff

Um Kopieraufträge durchzuführen müssen Sie über die folgenden Berechtigungen verfügen:

- Um Objekte von einem Verzeichnis-Bucket zu einem anderen Verzeichnis-Bucket zu kopieren, benötigen Sie die `s3express:CreateSession`-Berechtigung.
- Um Objekte aus Verzeichnis-Buckets zu Allzweck-Buckets zu kopieren, benötigen Sie die `s3express:CreateSession`-Berechtigung und `s3:PutObject`-Berechtigung zum Schreiben der Objektkopie in den Ziel-Bucket.
- Um Objekte aus Allzweck-Buckets in Verzeichnis-Buckets zu kopieren, benötigen Sie die `s3express:CreateSession` Berechtigung und die Berechtigung, das `s3:GetObject` kopierte Quellobjekt zu lesen.

  Weitere Informationen finden Sie unter CopyObjectin der Amazon Simple Storage Service API Reference.
- Um eine Lambda-Funktion aufzurufen, müssen Sie Ihrer Ressource basierend auf Ihrer Lambda-Funktion Berechtigungen erteilen. Um festzustellen, welche Berechtigungen erforderlich sind, überprüfen Sie die entsprechenden API Vorgänge.

Daten an Objekte in Verzeichnis-Buckets anhängen

Sie können Daten an das Ende vorhandener Objekte anhängen, die in der S3 Express One Zone-Speicherklasse in Verzeichnis-Buckets gespeichert sind. Es wird empfohlen, die Funktion zum Anhängen von Daten an ein Objekt zu nutzen, wenn die Daten über einen bestimmten Zeitraum kontinuierlich geschrieben werden oder wenn Sie das Objekt lesen müssen, während Sie in das Objekt schreiben. Das Anhängen von Daten an Objekte wird häufig in Anwendungsfällen eingesetzt, z. B. beim Hinzufügen neuer Protokolleinträge zu Protokolldateien oder beim Hinzufügen neuer Videosegmente zu Videodateien, wenn diese transkodiert und dann gestreamt werden. Durch das Anhängen von Daten an Objekte können Sie Anwendungen vereinfachen, die zuvor Daten im lokalen Speicher kombiniert haben, bevor das endgültige Objekt nach Amazon S3 kopiert wurde.

Es gibt keine Mindestgröße für die Daten, die Sie an ein Objekt anhängen können. Die maximale Größe der Daten, die Sie in einer einzigen Anfrage an ein Objekt anhängen können, beträgt jedoch 5 GB. Dies ist das gleiche Limit wie die größte Anforderungsgröße beim Hochladen von Daten mit einem beliebigen Amazon S3API.

Bei jedem erfolgreichen Anfügevorgang erstellen Sie einen Teil des Objekts, und jedes Objekt kann bis zu 10.000 Teile haben. Das bedeutet, dass Sie Daten bis zu 10.000 Mal an ein Objekt anhängen können. Wenn ein Objekt mithilfe des mehrteiligen S3-Uploads erstellt wird, wird jeder hochgeladene Teil auf die Gesamtzahl von 10.000 Teilen angerechnet. Sie können beispielsweise bis zu 9.000 Mal an ein Objekt anhängen, das durch mehrteiligen Upload aus 1.000 Teilen erstellt wurde.

> **Note**
>
> Wenn Sie das Limit an Teilen erreichen, erhalten Sie eine Fehlermeldung. TooManyParts Sie können den verwenden `CopyObject`API, um die Anzahl zurückzusetzen.

Wenn Sie Teile parallel in ein Objekt hochladen möchten und die Teile nicht lesen müssen, während die Teile hochgeladen werden, empfehlen wir Ihnen, den mehrteiligen Upload von Amazon S3 zu verwenden. Weitere Informationen finden Sie unter Mehrteiligen Upload verwenden.

Das Anhängen von Daten an Objekte wird nur für Objekte in Verzeichnis-Buckets unterstützt, die in der S3 Express One Zone-Speicherklasse gespeichert sind. Weitere Informationen zur S3 Express One Zone finden Sie unter Erste Schritte mit S3 Express One Zone.

Um mit dem Anhängen von Daten an Objekte in Ihren Verzeichnis-Buckets zu beginnen, können Sie AWS SDKs AWS CLI, und den verwenden. `PutObject` API Wenn Sie eine `PutObject` Anfrage

stellen, legen Sie den `x-amz-write-offset-bytes` Header auf die Größe des Objekts fest, an das Sie etwas anhängen möchten. Um den `PutObject` API Vorgang verwenden zu können, müssen Sie die verwenden, `CreateSession` API um temporäre Sicherheitsanmeldeinformationen für den Zugriff auf die Objekte in Ihren Verzeichnis-Buckets einzurichten. Weitere Informationen `CreateSession`und `PutObject`in der Amazon S3 API S3-Referenz.

Jeder erfolgreiche Anfügevorgang wird als Anfrage abgerechnet. `PutObject` Weitere Informationen zur Preisgestaltung finden Sie unter. `Amazon S3 pricing`

Ab Version 1.12 unterstützt Mountpoint für Amazon S3 das Anhängen von Daten an Objekte, die in S3 Express One Zone gespeichert sind. Um zu beginnen, müssen Sie sich anmelden, indem Sie die Einstellung von. `--incremental-upload` Weitere Informationen zu Mountpoint finden Sie unter Arbeiten mit Mountpoint.

Wenn Sie beim Hochladen der angehängten Daten einen Algorithmus CRC (Cyclic Redundancy Check) verwenden, können Sie mit der OR-Anfrage vollständige objektbasierte Prüfsummen abrufen. CRC `HeadObject` `GetObject` Wenn Sie beim Hochladen Ihrer angehängten Daten den Algorithmus SHA -1 oder SHA -256 verwenden, können Sie eine Prüfsumme der angehängten Teile abrufen und ihre Integrität anhand der Prüfsummen überprüfen, die bei früheren Antworten zurückgegeben wurden. SHA PutObject Weitere Informationen finden Sie unter Datenschutz und Verschlüsselung.

Anfügen von Daten an Ihre Objekte mithilfe von AWS CLI, AWS SDKs und REST API

Sie können Daten an Ihre Objekte anhängen, indem Sie AWS Command Line Interface (AWS CLI) und verwenden. AWS SDKs REST API

Mit dem AWS CLI

Der folgende `put-object` Beispielbefehl zeigt, wie Sie mit dem Daten AWS CLI an ein Objekt anhängen können. Um diesen Befehl auszuführen, ersetzen Sie den *`user input placeholders`* mit Ihren eigenen Informationen

```
aws s3api put-object --bucket amzn-s3-demo-bucket--azid--x-s3 --key sampleinput/
file001.bin --body bucket-seed/file001.bin --write-offset-bytes size-of-sampleinput/
file001.bin
```

Mit dem AWS SDKs

SDK for Java

Sie können den verwenden AWS SDK for Java , um Daten an Ihre Objekte anzuhängen.

```
var putObjectRequestBuilder = PutObjectRequest.builder()
                                              .key(key)
                                              .bucket(bucketName)
                                              .writeOffsetBytes(9);
var response = s3Client.putObject(putObjectRequestBuilder.build());
```

SDK for Python

```
s3.put_object(Bucket='amzn-s3-demo-bucket--use2-az2--x-s3', Key='2024-11-05-sdk-
test', Body=b'123456789', WriteOffsetBytes=9)
```

Mit dem REST API

Sie können REST Anfragen zum Anhängen von Daten an ein Objekt senden. Weitere Informationen finden Sie unter `PutObject`.

Objekte in einen Verzeichnis-Bucket hochladen

Nachdem Sie einen Amazon S3-Verzeichnis-Bucket erstellt haben, können Sie Objekte in diesen hochladen. Die folgenden Beispiele zeigen, wie Sie mithilfe der S3-Konsole und der ein Objekt in einen Verzeichnis-Bucket hochladen AWS SDKs. Informationen zum Hochladen von Objekten in großen Mengen mit S3 Express One Zone finden Sie unterVerwaltung von Objekten.

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie die Registerkarte Verzeichnis-Buckets.
4. Wählen Sie den Namen des Buckets, in den Sie Ihre Ordner oder Dateien hochladen möchten.
5. Wählen Sie in der Objektliste die Option Hochladen aus.
6. Führen Sie auf der Upload-Seite einen der folgenden Schritte aus:

- Ziehen Sie Dateien und Ordner per Drag & Drop in den gepunkteten Upload-Bereich.
- Wählen Sie „Dateien hinzufügen“ oder „Ordner hinzufügen“, wählen Sie die Dateien oder Ordner aus, die Sie hochladen möchten, und wählen Sie dann „Öffnen“ oder „Hochladen“.

7. Wählen Sie unter Prüfsummen die Prüfsummenfunktion aus, die Sie verwenden möchten.

   (Optional) Wenn Sie ein einzelnes Objekt hochladen, das weniger als 16 MB groß ist, können Sie auch einen vorab berechneten Prüfsummenwert angeben. Wenn Sie einen vorberechneten Wert angeben, vergleicht Amazon S3 ihn mit dem Wert, den es mithilfe der ausgewählten Prüfsummenfunktion berechnet. Wenn die Werte nicht übereinstimmen, startet der Upload nicht.

8. Die Optionen in den Abschnitten „Berechtigungen“ und „Eigenschaften“ werden automatisch auf die Standardeinstellungen gesetzt und können nicht geändert werden. Block Public Access ist automatisch aktiviert, und S3 Versioning und S3 Object Lock können für Directory-Buckets nicht aktiviert werden.

   (Optional) Wenn Sie Ihren Objekten Metadaten in Schlüssel-Wert-Paaren hinzufügen möchten, erweitern Sie den Abschnitt Eigenschaften und wählen Sie dann im Abschnitt Metadaten die Option Metadaten hinzufügen aus.

9. Um die aufgelisteten Dateien und Ordner hochzuladen, wählen Sie Hochladen.

   Amazon S3 lädt Ihre Objekte und Ordner hoch. Wenn der Upload abgeschlossen ist, wird auf der Seite Upload: Status eine Erfolgsmeldung angezeigt.

Verwenden Sie den AWS SDKs

SDK for Java 2.x

Example

```
public static void putObject(S3Client s3Client, String bucketName, String objectKey,
 Path filePath) {
       //Using File Path to avoid loading the whole file into memory
       try {
           PutObjectRequest putObj = PutObjectRequest.builder()
                   .bucket(bucketName)
                   .key(objectKey)
                   //.metadata(metadata)
                   .build();
           s3Client.putObject(putObj, filePath);
```

```
            System.out.println("Successfully placed " + objectKey +" into bucket
 "+bucketName);

        }

        catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
}
```

SDK for Python

Example

```
import boto3
import botocore
from botocore.exceptions import ClientError


def put_object(s3_client, bucket_name, key_name, object_bytes):
    """
    Upload data to a directory bucket.
    :param s3_client: The boto3 S3 client
    :param bucket_name: The bucket that will contain the object
    :param key_name: The key of the object to be uploaded
    :param object_bytes: The data to upload
    """
    try:
        response = s3_client.put_object(Bucket=bucket_name, Key=key_name,
                            Body=object_bytes)
        print(f"Upload object '{key_name}' to bucket '{bucket_name}'.")
        return response
    except ClientError:
        print(f"Couldn't upload object '{key_name}' to bucket '{bucket_name}'.")
        raise

def main():
    # Share the client session with functions and objects to benefit from S3 Express
 One Zone auth key
    s3_client = boto3.client('s3')
    # Directory bucket name must end with --zone-id--x-s3
    resp = put_object(s3_client, 'doc-bucket-example--use1-az5--x-s3', 'sample.txt',
 b'Hello, World!')
```

```
    print(resp)

if __name__ == "__main__":
    main()
```

Mit dem AWS CLI

Der folgende `put-object` Beispielbefehl zeigt, wie Sie mit AWS CLI dem ein Objekt aus Amazon S3 hochladen können. Zum Ausführen dieses Befehls ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws s3api put-object --bucket bucket-base-name--zone-id--x-s3 --key sampleinut/
file001.bin --body bucket-seed/file001.bin
```

Weitere Informationen finden Sie unter put-objectin der AWS CLI Befehlsreferenz.

Themen

- Mehrteilige Uploads mit Verzeichnis-Buckets verwenden

Mehrteilige Uploads mit Verzeichnis-Buckets verwenden

Sie können den mehrteiligen Upload-Prozess verwenden, um ein einzelnes Objekt als Satz von Teilen hochzuladen. Jeder Teil ist ein zusammenhängender Teil der Daten des Objekts. Sie können diese Objektteile unabhängig und in beliebiger Reihenfolge hochladen. Wenn die Übertragung eines Teils fehlschlägt, können Sie das Teil erneut übertragen, ohne dass dies Auswirkungen auf andere Teile hat. Nachdem alle Teile Ihres Objekts hochgeladen sind, fügt Amazon S3 diese Teile zusammen und erstellt das Objekt. Wenn Ihre Objektgröße 100 MB erreicht, sollten Sie in der Regel mehrteilige Uploads verwenden, anstatt das Objekt in einem einzigen Vorgang hochzuladen.

Die Nutzung mehrteiliger Uploads bietet die folgenden Vorteile:

- Verbesserter Durchsatz – Sie können die Teile parallel hochladen, um den Durchsatz zu erhöhen.
- Schnelle Wiederherstellung bei Netzwerkproblemen — Kleinere Bauteile minimieren die Auswirkungen eines Neustarts eines aufgrund eines Netzwerkfehlers fehlgeschlagenen Uploads.
- Anhalten und Fortsetzen von Objekt-Uploads – Sie können Objektteile mit der Zeit hochladen. Nachdem Sie einen mehrteiligen Upload gestartet haben, gibt es kein Ablaufdatum. Sie müssen den mehrteiligen Upload explizit abschließen oder abbrechen.

- Starten Sie einen Upload, bevor Sie die endgültige Objektgröße kennen – Sie können ein Objekt hochladen, während Sie es noch erstellen.

Wir empfehlen, mehrteilige Uploads auf folgende Weise zu verwenden:

- Wenn Sie große Objekte über ein stabiles Netzwerk mit hoher Bandbreite hochladen, verwenden Sie mehrteilige Uploads, um die verfügbare Bandbreite optimal zu nutzen, indem Sie Objektteile parallel hochladen, um eine Multithread-Leistung zu erzielen.
- Wenn Sie über ein lückenhaftes Netzwerk hochladen, verwenden Sie mehrteilige Uploads, um die Widerstandsfähigkeit gegenüber Netzwerkfehlern zu erhöhen, indem Sie Neustarts des Uploads vermeiden. Wenn Sie mehrteilige Uploads verwenden, müssen Sie erneut versuchen, nur die Teile hochzuladen, die während des Uploads unterbrochen wurden. Sie müssen nicht das gesamte Objekt von Anfang an neu hochladen.

Wenn Sie mehrteilige Uploads verwenden, um Objekte in Verzeichnis-Buckets in die Amazon S3 Express One Zone-Speicherklasse hochzuladen, ähnelt der mehrteilige Upload-Vorgang dem Vorgang des mehrteiligen Uploads zum Hochladen von Objekten in Allzweck-Buckets. Es gibt jedoch einige grundlegende Unterschiede.

Weitere Informationen zur Verwendung von mehrteiligen Uploads zum Hochladen von Objekten in die S3 Express One Zone finden Sie in den folgenden Themen.

Themen

- Der mehrteilige Upload-Vorgang
- Prüfsummen mit mehrteiligen Upload-Operationen
- Gleichzeitige mehrteilige Upload-Vorgänge
- Mehrteilige Uploads und Preisgestaltung
- Operationen und Berechtigungen für mehrteilige Uploads API
- Beispiele

Der mehrteilige Upload-Vorgang

Ein mehrteiliger Upload besteht aus drei Schritten:

- Sie initiieren den Upload.
- Sie laden die Objektteile hoch.

- Nachdem Sie alle Teile hochgeladen haben, schließen Sie den mehrteiligen Upload ab.

Nach Erhalt der vollständigen mehrteiligen Upload-Anfrage erstellt Amazon S3 das Objekt aus den hochgeladenen Teilen, und Sie können dann wie auf jedes andere Objekt in Ihrem Bucket auf das Objekt zugreifen.

Initiieren des mehrteiligen Uploads

Wenn Sie eine Anfrage zum Initiieren eines mehrteiligen Uploads senden, gibt Amazon S3 eine Antwort mit einer Upload-ID zurück, als eindeutige Kennung für den Multipart-Upload. Sie müssen diese Upload-ID immer angeben, wenn Sie Teile hochladen, die Teile auflisten, einen Upload durchführen oder ihn abbrechen.

Teile hochladen

Beim Hochladen eines Teils müssen Sie zusätzlich zur Upload-ID eine Teilenummer angeben. Wenn Sie einen mehrteiligen Upload mit S3 Express One Zone verwenden, müssen die mehrteiligen Artikelnummern fortlaufende Artikelnummern sein. Wenn Sie versuchen, eine mehrteilige Upload-Anfrage mit nicht aufeinanderfolgenden Artikelnummern abzuschließen, wird ein Fehler `HTTP 400 Bad Request` (Ungültige Bestellung) generiert.

Eine Artikelnummer identifiziert eindeutig ein Teil und seine Position in dem Objekt, das Sie hochladen. Wenn Sie ein neues Teil hochladen und dabei dieselbe Artikelnummer wie ein zuvor hochgeladenes Teil verwenden, wird das zuvor hochgeladene Teil überschrieben.

Immer wenn Sie ein Teil hochladen, gibt Amazon S3 in seiner Antwort einen Entity-Tag (ETag) -Header zurück. Für jeden Upload eines Teils müssen Sie die Artikelnummer und den ETag Wert aufzeichnen. Die ETag Werte für alle Uploads von Objektteilen bleiben gleich, aber jedem Teil wird eine andere Artikelnummer zugewiesen. Sie müssen diese Werte in die spätere Anforderung einschließen, um den mehrteiligen Upload abzuschließen.

Amazon S3 verschlüsselt automatisch alle neuen Objekte, die in einen S3-Bucket hochgeladen werden. Wenn Sie bei einem mehrteiligen Upload in Ihrer Kopieranforderung keine Verschlüsselungsinformationen angeben, wird die Verschlüsselungseinstellung der hochgeladenen Teile auf die Standardverschlüsselungskonfiguration des Ziel-Buckets festgelegt. Die Standardverschlüsselungskonfiguration eines Amazon S3 S3-Buckets ist immer aktiviert und mindestens auf serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) eingestellt. Für Verzeichnis-Buckets werden SSE -S3 und serverseitige Verschlüsselung mit AWS

KMS Schlüsseln (SSE-) unterstützt. KMS Weitere Informationen finden Sie unter Datenschutz und Verschlüsselung.

Abschließen eines mehrteiligen Uploads

Wenn Sie einen mehrteiligen Upload abschließen, erstellt Amazon S3 das Objekt, indem es die Teile anhand der Artikelnummer in aufsteigender Reihenfolge verkettet. Nach einer erfolgreich ausgeführten Abschlussanforderung sind die Teile nicht mehr vorhanden.

Ihre vollständige mehrteilige Upload-Anfrage muss die Upload-ID und eine Liste der beiden Artikelnummern und der entsprechenden Werte enthalten. ETag Die Amazon S3 S3-Antwort enthält eineETag, die die kombinierten Objektdaten eindeutig identifiziert. Dies ETag ist kein MD5 Hash der Objektdaten.

Auflistungen mehrteiliger Uploads

Sie können alle Teile eines bestimmten Multipart-Uploads oder alle laufenden mehrteiligen Uploads auflisten. Die Operation für die Teileauflistung gibt die Teileinformationen zurück, die Sie für einen bestimmten mehrteiligen Upload hochgeladen haben. Für jeden Abruf einer Teileauflistung gibt Amazon S3 die Teileinformationen für einen angegebenen mehrteiligen Upload bis zu maximal 1.000 Teilen zurück. Wenn in dem mehrteiligen Upload mehr als 1 000 Teile vorhanden sind, müssen Sie die Paginierung verwenden, um alle Teile abzurufen.

Die zurückgegebene Teileliste enthält keine Teile, deren Upload noch nicht abgeschlossen ist. Bei Verwendung der Operation Mehrteilige Uploads auflisten können Sie eine Liste aller mehrteiligen Uploads erhalten, die sich in Bearbeitung befinden.

Ein laufender mehrteiliger Upload ist ein Upload, den Sie initiiert haben, der aber noch nicht abgeschlossen ist oder abgebrochen wurde. Jeder Anforderung gibt bis zu 1.000 mehrteilige Uploads zurück. Wenn mehr als 1 000 mehrteilige Uploads vorhanden sind, müssen Sie zusätzliche Anforderungen senden, um die verbleibenden mehrteiligen Uploads abzurufen. Verwenden Sie die zurückgegebene Liste nur zur Überprüfung. Verwenden Sie das Ergebnis dieser Auflistung nicht, wenn Sie eine Anforderung für den Abschluss eines mehrteiligen Uploads senden. Pflegen Sie stattdessen Ihre eigene Liste mit den Artikelnummern, die Sie beim Hochladen von Teilen angegeben haben, und den entsprechenden ETag Werten, die Amazon S3 zurückgibt.

Weitere Informationen zu mehrteiligen Upload-Angeboten finden Sie unter ListPartsin der Amazon Simple Storage Service API Reference.

Prüfsummen mit mehrteiligen Upload-Operationen

Wenn Sie ein Objekt hochladen, können Sie einen Prüfsummenalgorithmus angeben, um die Objektintegrität zu überprüfen. MD5wird für Verzeichnis-Buckets nicht unterstützt. Sie können einen der folgenden Algorithmen zur Prüfung der Datenintegrität mit Secure Hash Algorithms (SHA) oder Cyclic Redundancy Check (CRC) angeben:

- CRC32
- CRC32C
- SHA-1
- SHA-256

Sie können Amazon S3 REST API oder den verwenden, AWS SDKs um den Prüfsummenwert für einzelne Teile abzurufen, indem Sie `GetObject` oder `HeadObject` verwenden. Wenn Sie die Prüfsummenwerte für einzelne Teile von mehrteiligen Uploads abrufen möchten, die noch in Bearbeitung sind, können Sie `ListParts` verwenden.

> **Important**
>
> Wenn Sie die oben genannten Prüfsummenalgorithmen verwenden, müssen die mehrteiligen Artikelnummern aufeinanderfolgende Artikelnummern verwenden. Wenn Sie versuchen, eine mehrteilige Upload-Anfrage mit nicht aufeinanderfolgenden Artikelnummern abzuschließen, generiert Amazon S3 einen Fehler `HTTP 400 Bad Request` (Ungültige Teilbestellung).

Weitere Informationen darüber, wie Prüfsummen bei mehrteiligen Upload-Objekten funktionieren, finden Sie unter. Überprüfung der Objektintegrität

Gleichzeitige mehrteilige Upload-Vorgänge

In einer verteilten Entwicklungsumgebung kann Ihre Anwendung mehrere Aktualisierungen für dasselbe Objekt gleichzeitig initiieren. Beispielsweise könnte Ihre Anwendung mehrere mehrteilige Uploads mithilfe desselben Objektschlüssels initiieren. Für jeden dieser Uploads kann Ihre Anwendung Teile hochladen und eine Anfrage auf Abschluss des Uploads an Amazon S3 senden, um das Objekt zu erstellen. Für S3 Express One Zone entspricht die Objekterstellungszeit dem Abschlussdatum des mehrteiligen Uploads.

> **Important**
>
> Die Versionierung wird für Objekte, die in Verzeichnis-Buckets gespeichert sind, nicht unterstützt.

Mehrteilige Uploads und Preisgestaltung

Nachdem Sie einen mehrteiligen Upload gestartet haben, behält Amazon S3 alle Teile bei, bis Sie den Upload abschließen oder abbrechen. Während seiner gesamten Lebensdauer werden Ihnen der gesamte Speicher, die Bandbreite und die Anforderungen für diesen mehrteiligen Upload und die zugehörigen Teile in Rechnung gestellt. Wenn Sie den mehrteiligen Upload abbrechen, löscht Amazon S3 die Upload-Artefakte und alle Teile, die Sie hochgeladen haben, und sie werden Ihnen nicht mehr in Rechnung gestellt. Für das Löschen unvollständiger mehrteiliger Uploads fallen unabhängig von der angegebenen Speicherklasse keine Gebühren für vorzeitiges Löschen an. Weitere Informationen zu Preisen finden Sie unter Amazon-S3-Preise.

> **Important**
>
> Wenn die vollständige mehrteilige Upload-Anfrage nicht erfolgreich gesendet wird, werden die Objektteile nicht zusammengesetzt und es wird kein Objekt erstellt. Ihnen wird der gesamte Speicher in Rechnung gestellt, der mit hochgeladenen Teilen verknüpft ist. Es ist wichtig, dass Sie entweder den mehrteiligen Upload abschließen, damit das Objekt erstellt wird, oder den mehrteiligen Upload abbrechen, um alle hochgeladenen Teile zu entfernen.
> Bevor Sie einen Verzeichnis-Bucket löschen können, müssen Sie alle laufenden mehrteiligen Uploads abschließen oder abbrechen. Directory-Buckets unterstützen keine S3-Lifecycle-Konfigurationen. Bei Bedarf können Sie Ihre aktiven mehrteiligen Uploads auflisten, die Uploads dann abbrechen und dann Ihren Bucket löschen.

Operationen und Berechtigungen für mehrteilige Uploads API

Um den Zugriff auf API Objektverwaltungsvorgänge in einem Verzeichnis-Bucket zu ermöglichen, erteilen Sie die `s3express:CreateSession` Berechtigung in einer Bucket-Richtlinie oder einer AWS Identity and Access Management (IAM) identitätsbasierten Richtlinie.

Sie müssen über die erforderlichen Berechtigungen verfügen, um die Multipart-Upload-Vorgänge zu verwenden. Sie können Bucket-Richtlinien oder IAM identitätsbasierte Richtlinien verwenden, um IAM

Prinzipalen Berechtigungen zur Ausführung dieser Operationen zu gewähren. Die folgende Tabelle listet die erforderlichen Berechtigungen für verschiedene mehrteilige Uploadvorgänge auf.

Sie können den Initiator eines mehrteiligen Uploads anhand des Elements identifizieren. `Initiator` Wenn der Initiator ein ist AWS-Konto, liefert dieses Element dieselben Informationen wie das Element. `Owner` Wenn der Initiator ein IAM Benutzer ist, stellt dieses Element den Benutzer ARN und den Anzeigenamen bereit.

Aktion	Erforderliche Berechtigungen
Erstellen eines mehrteiligen Uploads	Um den mehrteiligen Upload zu erstellen, müssen Sie berechtigt sein, die `s3express:CreateSession` Aktion im Verzeichnis-Bucket auszuführen.
Initiieren Sie einen mehrteiligen Upload	Um den mehrteiligen Upload zu initiieren, müssen Sie berechtigt sein, die `s3express:CreateSession` Aktion für den Verzeichnis-Bucket auszuführen.
Laden Sie einen Teil hoch	Um ein Teil hochzuladen, müssen Sie berechtigt sein, die `s3express:CreateSession` Aktion im Verzeichnis-Bucket auszuführen. Damit der Initiator einen Teil hochladen kann, muss der Bucket-Besitzer dem Initiator erlauben, die `s3express:CreateSession` Aktion für den Verzeichnis-Bucket auszuführen.
Laden Sie einen Teil hoch (Kopie)	Um ein Teil hochzuladen, müssen Sie berechtigt sein, die `s3express:CreateSession` Aktion im Verzeichnis-Bucket auszuführen. Damit der Initiator einen Teil eines Objekts hochladen kann, muss der Bucket-Besitzer dem Initiator eine Genehmigung für die Ausführung der Aktion `s3express:CreateSession` für das Objekt erteilen.
Abschließen eines mehrteiligen Uploads	Um einen mehrteiligen Upload abzuschließen, müssen Sie berechtigt sein, die `s3express:CreateSession` Aktion im Verzeichnis-Bucket auszuführen.

Aktion	Erforderliche Berechtigungen
	Damit der Initiator einen mehrteiligen Upload abschließen kann, muss der Bucket-Besitzer dem Initiator gestatten, die `s3express:CreateSession` Aktion für das Objekt auszuführen.
Abbrechen eines mehrteiligen Uploads	Um einen mehrteiligen Upload abzubrechen, müssen Sie berechtigt sein, die Aktion auszuführen. `s3express:CreateSession` Damit der Initiator einen mehrteiligen Upload abbrechen kann, muss dem Initiator die ausdrückliche Zugriffsberechtigung für die Ausführung der Aktion gewährt werden. `s3express:CreateSession`
Teile auflisten	Um die Teile in einem mehrteiligen Upload aufzulisten, müssen Sie berechtigt sein, die `s3express:CreateSession` Aktion im Verzeichnis-Bucket auszuführen.
Laufende mehrteilige Uploads auflisten	Um die laufenden mehrteiligen Uploads in einen Bucket aufzulisten, müssen Sie berechtigt sein, die `s3:ListBucketMultipartUploads` Aktion für diesen Bucket auszuführen.

APIBetriebsunterstützung für mehrteilige Uploads

In den folgenden Abschnitten der Amazon Simple Storage Service API Reference werden die Amazon S3 REST API S3-Operationen für mehrteilige Uploads beschrieben.

- CreateMultipartUpload
- UploadPart
- UploadPartCopy
- CompleteMultipartUpload
- AbortMultipartUpload
- ListParts
- ListMultipartUploads

Beispiele

Um einen mehrteiligen Upload zu verwenden, um ein Objekt in die S3 Express One Zone in einem Verzeichnis-Bucket hochzuladen, sehen Sie sich die folgenden Beispiele an.

Themen

- Erstellen eines mehrteiligen Uploads
- Teile eines mehrteiligen Uploads hochladen
- Abschließen eines mehrteiligen Uploads
- Abbrechen eines mehrteiligen Uploads
- Erstellen eines mehrteiligen Upload-Kopiervorgangs
- Liste laufender mehrteiliger Uploads
- Auflisten der Teile eines mehrteiligen Uploads

Erstellen eines mehrteiligen Uploads

> **Note**
>
> Wenn Sie bei Verzeichnis-Buckets einen `CreateMultipartUpload` Vorgang und eine `UploadPartCopy` Operation ausführen, muss die Standardverschlüsselung des Buckets die gewünschte Verschlüsselungskonfiguration verwenden, und die Anforderungsheader, die Sie in der Anfrage angeben, müssen mit der `CreateMultipartUpload` Standardverschlüsselungskonfiguration des Ziel-Buckets übereinstimmen.

Die folgenden Beispiele zeigen, wie Sie einen mehrteiligen Upload erstellen.

Mit dem AWS SDKs

SDK for Java 2.x

Example

```
/**
 * This method creates a multipart upload request that generates a unique upload ID
 that is used to track
 * all the upload parts
 *
```

```
 * @param s3
 * @param bucketName - for example, 'doc-example-bucket--use1-az4--x-s3'
 * @param key
 * @return
 */
 private static String createMultipartUpload(S3Client s3, String bucketName, String
 key) {

     CreateMultipartUploadRequest createMultipartUploadRequest =
 CreateMultipartUploadRequest.builder()
             .bucket(bucketName)
             .key(key)
             .build();

     String uploadId = null;

     try {
         CreateMultipartUploadResponse response =
 s3.createMultipartUpload(createMultipartUploadRequest);
         uploadId = response.uploadId();
     }
     catch (S3Exception e) {
         System.err.println(e.awsErrorDetails().errorMessage());
         System.exit(1);
     }
     return uploadId;
```

SDK for Python

Example

```
def create_multipart_upload(s3_client, bucket_name, key_name):
    '''
   Create a multipart upload to a directory bucket

   :param s3_client: boto3 S3 client
   :param bucket_name: The destination bucket for the multipart upload
   :param key_name: The key name for the object to be uploaded
   :return: The UploadId for the multipart upload if created successfully, else None
   '''

   try:
        mpu = s3_client.create_multipart_upload(Bucket = bucket_name, Key =
 key_name)
```

```
        return mpu['UploadId']
    except ClientError as e:
        logging.error(e)
        return None
```

Mit dem AWS CLI

Dieses Beispiel zeigt, wie Sie mithilfe von einen mehrteiligen Upload in einen Verzeichnis-Bucket erstellen. AWS CLI Dieser Befehl startet einen mehrteiligen Upload in den Verzeichnis-Bucket *bucket-base-name*--*zone-id*--x-s3 für das Objekt *KEY_NAME*. Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```
aws s3api create-multipart-upload --bucket bucket-base-name--zone-id--x-s3 --
key KEY_NAME
```

Weitere Informationen finden Sie create-multipart-uploadin der AWS Command Line Interface.

Teile eines mehrteiligen Uploads hochladen

Die folgenden Beispiele zeigen, wie Teile eines mehrteiligen Uploads hochgeladen werden.

Mit dem AWS SDKs

SDK for Java 2.x

Das folgende Beispiel zeigt, wie Sie ein einzelnes Objekt in Teile aufteilen und diese Teile dann mithilfe von SDK für Java 2.x in einen Verzeichnis-Bucket hochladen.

Example

```
/**
 * This method creates part requests and uploads individual parts to S3 and then
 returns all the completed parts
 *
 * @param s3
 * @param bucketName
 * @param key
 * @param uploadId
 * @throws IOException
 */
 private static ListCompletedPartmultipartUpload(S3Client s3, String bucketName,
 String key, String uploadId, String filePath) throws IOException {
```

```
        int partNumber = 1;
        ListCompletedPart completedParts = new ArrayList<>();
        ByteBuffer bb = ByteBuffer.allocate(1024 * 1024 * 5); // 5 MB byte buffer

        // read the local file, breakdown into chunks and process
        try (RandomAccessFile file = new RandomAccessFile(filePath, "r")) {
            long fileSize = file.length();
            int position = 0;
            while (position < fileSize) {
                file.seek(position);
                int read = file.getChannel().read(bb);

                bb.flip(); // Swap position and limit before reading from the
buffer.
                UploadPartRequest uploadPartRequest = UploadPartRequest.builder()
                        .bucket(bucketName)
                        .key(key)
                        .uploadId(uploadId)
                        .partNumber(partNumber)
                        .build();

                UploadPartResponse partResponse = s3.uploadPart(
                        uploadPartRequest,
                        RequestBody.fromByteBuffer(bb));

                CompletedPart part = CompletedPart.builder()
                        .partNumber(partNumber)
                        .eTag(partResponse.eTag())
                        .build();
                completedParts.add(part);

                bb.clear();
                position += read;
                partNumber++;
            }
        }

        catch (IOException e) {
            throw e;
        }
        return completedParts;
    }
```

SDK for Python

Das folgende Beispiel zeigt, wie Sie ein einzelnes Objekt in Teile aufteilen und diese Teile dann mithilfe von SDK for Python in einen Verzeichnis-Bucket hochladen.

Example

```
def multipart_upload(s3_client, bucket_name, key_name, mpu_id, part_size):
    '''
    Break up a file into multiple parts and upload those parts to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Destination bucket for the multipart upload
    :param key_name: Key name for object to be uploaded and for the local file
 that's being uploaded
    :param mpu_id: The UploadId returned from the create_multipart_upload call
    :param part_size: The size parts that the object will be broken into, in bytes.
                      Minimum 5 MiB, Maximum 5 GiB. There is no minimum size for the
 last part of your multipart upload.
    :return: part_list for the multipart upload if all parts are uploaded
 successfully, else None
    '''

    part_list = []
    try:
        with open(key_name, 'rb') as file:
            part_counter = 1
            while True:
                file_part = file.read(part_size)
                if not len(file_part):
                    break
                upload_part = s3_client.upload_part(
                    Bucket = bucket_name,
                    Key = key_name,
                    UploadId = mpu_id,
                    Body = file_part,
                    PartNumber = part_counter
                )
                part_list.append({'PartNumber': part_counter, 'ETag':
 upload_part['ETag']})
                part_counter += 1
    except ClientError as e:
        logging.error(e)
        return None
```

```
    return part_list
```

Mit dem AWS CLI

Dieses Beispiel zeigt, wie Sie ein einzelnes Objekt in Teile aufteilen und diese Teile dann mithilfe von in einen Verzeichnis-Bucket hochladen AWS CLI. Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```
aws s3api upload-part --bucket bucket-base-name--zone-id--x-s3 --
key KEY_NAME --part-number 1 --body LOCAL_FILE_NAME --upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
```

Weitere Informationen finden Sie unter Upload-Teil im. AWS Command Line Interface

Abschließen eines mehrteiligen Uploads

Die folgenden Beispiele zeigen, wie Sie einen mehrteiligen Upload abschließen.

Mit dem AWS SDKs

SDK for Java 2.x

Die folgenden Beispiele zeigen, wie Sie einen mehrteiligen Upload mit dem SDK für Java 2.x abschließen.

Example

```
/**
 * This method completes the multipart upload request by collating all the upload
 parts
 * @param s3
 * @param bucketName - for example, 'doc-example-bucket--usw2-az1--x-s3'
 * @param key
 * @param uploadId
 * @param uploadParts
 */
 private static void completeMultipartUpload(S3Client s3, String bucketName, String
 key, String uploadId, ListCompletedPart uploadParts) {
        CompletedMultipartUpload completedMultipartUpload =
 CompletedMultipartUpload.builder()
                .parts(uploadParts)
                .build();
```

```
        CompleteMultipartUploadRequest completeMultipartUploadRequest =
                CompleteMultipartUploadRequest.builder()
                        .bucket(bucketName)
                        .key(key)
                        .uploadId(uploadId)
                        .multipartUpload(completedMultipartUpload)
                        .build();

        s3.completeMultipartUpload(completeMultipartUploadRequest);
    }

    public static void multipartUploadTest(S3Client s3, String bucketName, String
 key, String localFilePath)  {
        System.out.println("Starting multipart upload for: " + key);
        try {
            String uploadId = createMultipartUpload(s3, bucketName, key);
            System.out.println(uploadId);
            ListCompletedPart parts = multipartUpload(s3, bucketName, key, uploadId,
 localFilePath);
            completeMultipartUpload(s3, bucketName, key, uploadId, parts);
            System.out.println("Multipart upload completed for: " + key);
        }

        catch (Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

SDK for Python

Die folgenden Beispiele zeigen, wie Sie einen mehrteiligen Upload mithilfe von SDK for Python abschließen.

Example

```
def complete_multipart_upload(s3_client, bucket_name, key_name, mpu_id, part_list):
    '''
    Completes a multipart upload to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: The destination bucket for the multipart upload
    :param key_name: The key name for the object to be uploaded
```

```
    :param mpu_id: The UploadId returned from the create_multipart_upload call
    :param part_list: The list of uploaded part numbers with their associated ETags
    :return: True if the multipart upload was completed successfully, else False
    '''

    try:
        s3_client.complete_multipart_upload(
            Bucket = bucket_name,
            Key = key_name,
            UploadId = mpu_id,
            MultipartUpload = {
                'Parts': part_list
            }
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    MB = 1024 ** 2
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    key_name = 'OBJECT_NAME'
    part_size = 10 * MB
    s3_client = boto3.client('s3', region_name = region)
    mpu_id = create_multipart_upload(s3_client, bucket_name, key_name)
    if mpu_id is not None:
        part_list = multipart_upload(s3_client, bucket_name, key_name, mpu_id,
 part_size)
        if part_list is not None:
            if complete_multipart_upload(s3_client, bucket_name, key_name, mpu_id,
 part_list):
                print (f'{key_name} successfully uploaded through a ultipart upload
 to {bucket_name}')
            else:
                print (f'Could not upload {key_name} hrough a multipart upload to
 {bucket_name}')
```

Mit dem AWS CLI

Dieses Beispiel zeigt, wie Sie einen mehrteiligen Upload für einen Verzeichnis-Bucket mithilfe von abschließen. AWS CLI Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```
aws s3api complete-multipart-upload --bucket bucket-base-name--zone-id--x-s3 --
key KEY_NAME --upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
 --multipart-upload file://parts.json
```

In diesem Beispiel wird eine JSON Struktur verwendet, die die Teile des mehrteiligen Uploads beschreibt, die wieder zur vollständigen Datei zusammengefügt werden sollen. In diesem Beispiel wird das `file://` Präfix verwendet, um die JSON Struktur aus einer Datei im lokalen Ordner mit dem Namen zu laden. `parts`

parts.json:

```
parts.json
{
  "Parts": [
    {
      "ETag": "6b78c4a64dd641a58dac8d9258b88147",
      "PartNumber": 1
    }
  ]
}
```

Weitere Informationen finden Sie complete-multipart-uploadin der. AWS Command Line Interface

Abbrechen eines mehrteiligen Uploads

Die folgenden Beispiele zeigen, wie Sie einen mehrteiligen Upload abbrechen.

Mit dem AWS SDKs

SDK for Java 2.x

> Das folgende Beispiel zeigt, wie Sie einen mehrteiligen Upload mit dem SDK für Java 2.x abbrechen.

Example

```
public static void abortMultiPartUploads( S3Client s3, String bucketName ) {

        try {
            ListMultipartUploadsRequest listMultipartUploadsRequest =
 ListMultipartUploadsRequest.builder()
                    .bucket(bucketName)
                    .build();

            ListMultipartUploadsResponse response =
 s3.listMultipartUploads(listMultipartUploadsRequest);
            ListMultipartUpload uploads = response.uploads();

            AbortMultipartUploadRequest abortMultipartUploadRequest;
            for (MultipartUpload upload: uploads) {
                abortMultipartUploadRequest = AbortMultipartUploadRequest.builder()
                        .bucket(bucketName)
                        .key(upload.key())
                        .uploadId(upload.uploadId())
                        .build();

                s3.abortMultipartUpload(abortMultipartUploadRequest);
            }

        }

        catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

SDK for Python

Das folgende Beispiel zeigt, wie ein mehrteiliger Upload mithilfe von SDK for Python abgebrochen wird.

Example

```
import logging
import boto3
from botocore.exceptions import ClientError
```

```
def abort_multipart_upload(s3_client, bucket_name, key_name, upload_id):
    '''
    Aborts a partial multipart upload in a directory bucket.

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket where the multipart upload was initiated - for
 example, 'doc-example-bucket--usw2-az1--x-s3'
    :param key_name: Name of the object for which the multipart upload needs to be
 aborted
    :param upload_id: Multipart upload ID for the multipart upload to be aborted
    :return: True if the multipart upload was successfully aborted, False if not
    '''
    try:
        s3_client.abort_multipart_upload(
            Bucket = bucket_name,
            Key = key_name,
            UploadId = upload_id
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True


if __name__ == '__main__':
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    key_name = 'KEY_NAME'
        upload_id = 'UPLOAD_ID'
    s3_client = boto3.client('s3', region_name = region)
    if abort_multipart_upload(s3_client, bucket_name, key_name, upload_id):
        print (f'Multipart upload for object {key_name} in {bucket_name} bucket has
 been aborted')
    else:
        print (f'Unable to abort multipart upload for object {key_name} in
 {bucket_name} bucket')
```

Mit dem AWS CLI

Das folgende Beispiel zeigt, wie Sie einen mehrteiligen Upload mit dem abbrechen. AWS CLI Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```
aws s3api abort-multipart-upload --bucket bucket-base-name--zone-id--x-s3 --
key KEY_NAME --upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
MAQAAAAB0OxUFeA7LTbWWFS8WYwhrxDxTIDN-pdEEq_agIHqsbg"
```

Weitere Informationen finden Sie abort-multipart-uploadin der AWS Command Line Interface.

Erstellen eines mehrteiligen Upload-Kopiervorgangs

> **Note**
>
> - Um neue Objektteilkopien in einem Verzeichnis-Bucket mit SSE - zu verschlüsselnKMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom Kunden verwalteten Schlüssel) angeben. Von AWS verwalteter Schlüssel(`aws/s3`) wird nicht unterstützt. Ihre KMS Konfiguration unterstützt für die gesamte Lebensdauer des Buckets nur einen vom Kunden verwalteten Schlüssel pro SSE Verzeichnis-Bucket. Nachdem Sie einen vom Kunden verwalteten Schlüssel für SSE - angegeben habenKMS, können Sie den vom Kunden verwalteten Schlüssel für die KMS Konfiguration des SSE Buckets nicht überschreiben. Sie können keine serverseitigen Verschlüsselungseinstellungen für neue Objektteilkopien mit SSE - KMS in den UploadPartCopyAnforderungsheadern angeben. Außerdem müssen die Anforderungsheader, die Sie in der `CreateMultipartUpload` Anfrage angeben, der Standardverschlüsselungskonfiguration des Ziel-Buckets entsprechen.
> - S3-Bucket-Keys werden nicht unterstützt, wenn Sie KMS verschlüsselte Objekte von Allzweck-Buckets in Verzeichnis-Buckets, von Verzeichnis-Buckets in Allzweck-Buckets oder zwischen Verzeichnis-Buckets kopieren SSE UploadPartCopy. In diesem Fall ruft Amazon S3 AWS KMS jedes Mal, wenn eine Kopieranforderung für ein KMS - verschlüsseltes Objekt gestellt wird, auf.

Die folgenden Beispiele zeigen, wie Objekte mithilfe eines mehrteiligen Uploads von einem Bucket in einen anderen kopiert werden.

Mit dem AWS SDKs

SDK for Java 2.x

Das folgende Beispiel zeigt, wie ein mehrteiliger Upload verwendet wird, um mithilfe von SDK für Java 2.x ein Objekt programmgesteuert von einem Bucket in einen anderen zu kopieren.

Example

```
/**
 * This method creates a multipart upload request that generates a unique upload ID
 that is used to track
 * all the upload parts.
 *
 * @param s3
 * @param bucketName
 * @param key
 * @return
 */
 private static String createMultipartUpload(S3Client s3, String bucketName, String
 key) {
        CreateMultipartUploadRequest createMultipartUploadRequest =
 CreateMultipartUploadRequest.builder()
                .bucket(bucketName)
                .key(key)
                .build();
        String uploadId = null;
        try {
            CreateMultipartUploadResponse response =
 s3.createMultipartUpload(createMultipartUploadRequest);
            uploadId = response.uploadId();
        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return uploadId;
  }

  /**
   * Creates copy parts based on source object size and copies over individual parts
   *
```

```
  * @param s3
  * @param sourceBucket
  * @param sourceKey
  * @param destnBucket
  * @param destnKey
  * @param uploadId
  * @return
  * @throws IOException
  */
   public static ListCompletedPart multipartUploadCopy(S3Client s3, String
sourceBucket, String sourceKey, String destnBucket, String destnKey, String
uploadId) throws IOException {

       // Get the object size to track the end of the copy operation.
       HeadObjectRequest headObjectRequest = HeadObjectRequest
               .builder()
               .bucket(sourceBucket)
               .key(sourceKey)
               .build();
       HeadObjectResponse response = s3.headObject(headObjectRequest);
       Long objectSize = response.contentLength();

       System.out.println("Source Object size: " + objectSize);

       // Copy the object using 20 MB parts.
       long partSize = 20 * 1024 * 1024;
       long bytePosition = 0;
       int partNum = 1;
       ListCompletedPart completedParts = new ArrayList<>();
       while (bytePosition < objectSize) {
           // The last part might be smaller than partSize, so check to make sure
           // that lastByte isn't beyond the end of the object.
           long lastByte = Math.min(bytePosition + partSize - 1, objectSize - 1);

           System.out.println("part no: " + partNum + ", bytePosition: " +
bytePosition + ", lastByte: " + lastByte);

           // Copy this part.
           UploadPartCopyRequest req = UploadPartCopyRequest.builder()
                   .uploadId(uploadId)
                   .sourceBucket(sourceBucket)
                   .sourceKey(sourceKey)
                   .destinationBucket(destnBucket)
                   .destinationKey(destnKey)
```

```
                    .copySourceRange("bytes="+bytePosition+"-"+lastByte)
                    .partNumber(partNum)
                    .build();
            UploadPartCopyResponse res = s3.uploadPartCopy(req);
            CompletedPart part = CompletedPart.builder()
                    .partNumber(partNum)
                    .eTag(res.copyPartResult().eTag())
                    .build();
            completedParts.add(part);
            partNum++;
            bytePosition += partSize;
        }
        return completedParts;
    }


    public static void multipartCopyUploadTest(S3Client s3, String srcBucket, String
srcKey, String destnBucket, String destnKey)  {
        System.out.println("Starting multipart copy for: " + srcKey);
        try {
            String uploadId = createMultipartUpload(s3, destnBucket, destnKey);
            System.out.println(uploadId);
            ListCompletedPart parts = multipartUploadCopy(s3, srcBucket,
srcKey,destnBucket,  destnKey, uploadId);
            completeMultipartUpload(s3, destnBucket, destnKey, uploadId, parts);
            System.out.println("Multipart copy completed for: " + srcKey);
        } catch (Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

SDK for Python

Das folgende Beispiel zeigt, wie ein mehrteiliger Upload verwendet wird, um mithilfe von for Python ein Objekt programmgesteuert von einem Bucket in einen anderen zu kopieren. SDK

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def head_object(s3_client, bucket_name, key_name):
```

```
    '''
    Returns metadata for an object in a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket that contains the object to query for metadata
    :param key_name: Key name to query for metadata
    :return: Metadata for the specified object if successful, else None
    '''

    try:
        response = s3_client.head_object(
            Bucket = bucket_name,
            Key = key_name
        )
        return response
    except ClientError as e:
        logging.error(e)
        return None

def create_multipart_upload(s3_client, bucket_name, key_name):
    '''
    Create a multipart upload to a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Destination bucket for the multipart upload
    :param key_name: Key name of the object to be uploaded
    :return: UploadId for the multipart upload if created successfully, else None
    '''

    try:
        mpu = s3_client.create_multipart_upload(Bucket = bucket_name, Key =
 key_name)
        return mpu['UploadId']
    except ClientError as e:
        logging.error(e)
        return None

def multipart_copy_upload(s3_client, source_bucket_name, key_name,
 target_bucket_name, mpu_id, part_size):
    '''
    Copy an object in a directory bucket to another bucket in multiple parts of a
 specified size

    :param s3_client: boto3 S3 client
```

```
    :param source_bucket_name: Bucket where the source object exists
    :param key_name: Key name of the object to be copied
    :param target_bucket_name: Destination bucket for copied object
    :param mpu_id: The UploadId returned from the create_multipart_upload call
    :param part_size: The size parts that the object will be broken into, in bytes.
                      Minimum 5 MiB, Maximum 5 GiB. There is no minimum size for the
 last part of your multipart upload.
    :return: part_list for the multipart copy if all parts are copied successfully,
 else None
    '''

    part_list = []
    copy_source = {
        'Bucket': source_bucket_name,
        'Key': key_name
    }
    try:
        part_counter = 1
        object_size = head_object(s3_client, source_bucket_name, key_name)
        if object_size is not None:
            object_size = object_size['ContentLength']
        while (part_counter - 1) * part_size <object_size:
            bytes_start = (part_counter - 1) * part_size
            bytes_end = (part_counter * part_size) - 1
            upload_copy_part = s3_client.upload_part_copy (
                Bucket = target_bucket_name,
                CopySource = copy_source,
                CopySourceRange = f'bytes={bytes_start}-{bytes_end}',
                Key = key_name,
                PartNumber = part_counter,
                UploadId = mpu_id
            )
            part_list.append({'PartNumber': part_counter, 'ETag':
 upload_copy_part['CopyPartResult']['ETag']})
            part_counter += 1
    except ClientError as e:
        logging.error(e)
        return None
    return part_list

def complete_multipart_upload(s3_client, bucket_name, key_name, mpu_id, part_list):
    '''
    Completes a multipart upload to a directory bucket
```

```
    :param s3_client: boto3 S3 client
    :param bucket_name: Destination bucket for the multipart upload
    :param key_name: Key name of the object to be uploaded
    :param mpu_id: The UploadId returned from the create_multipart_upload call
    :param part_list: List of uploaded part numbers with associated ETags
    :return: True if the multipart upload was completed successfully, else False
    '''

    try:
        s3_client.complete_multipart_upload(
            Bucket = bucket_name,
            Key = key_name,
            UploadId = mpu_id,
            MultipartUpload = {
                'Parts': part_list
            }
        )
    except ClientError as e:
        logging.error(e)
        return False
    return True

if __name__ == '__main__':
    MB = 1024 ** 2
    region = 'us-west-2'
    source_bucket_name = 'SOURCE_BUCKET_NAME'
    target_bucket_name = 'TARGET_BUCKET_NAME'
    key_name = 'KEY_NAME'
    part_size = 10 * MB
    s3_client = boto3.client('s3', region_name = region)
    mpu_id = create_multipart_upload(s3_client, target_bucket_name, key_name)
    if mpu_id is not None:
        part_list = multipart_copy_upload(s3_client, source_bucket_name, key_name,
 target_bucket_name, mpu_id, part_size)
        if part_list is not None:
            if complete_multipart_upload(s3_client, target_bucket_name, key_name,
 mpu_id, part_list):
                print (f'{key_name} successfully copied through multipart copy from
 {source_bucket_name} to {target_bucket_name}')
            else:
                print (f'Could not copy {key_name} through multipart copy from
 {source_bucket_name} to {target_bucket_name}')
```

Mit dem AWS CLI

Das folgende Beispiel zeigt, wie ein mehrteiliger Upload verwendet wird, um mithilfe von programmgesteuert ein Objekt von einem Bucket in einen Verzeichnis-Bucket zu kopieren. AWS CLI Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```
aws s3api upload-part-copy --bucket bucket-base-name--zone-id--x-s3 --
key TARGET_KEY_NAME --copy-source SOURCE_BUCKET_NAME/SOURCE_KEY_NAME --part-number 1 --
upload-id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
```

Weitere Informationen finden Sie upload-part-copyin der AWS Command Line Interface.

Liste laufender mehrteiliger Uploads

Um in Bearbeitung befindliche mehrteilige Uploads in einen Verzeichnis-Bucket aufzulisten, können Sie den, oder den verwenden. AWS SDKs AWS CLI

Verwenden Sie den AWS SDKs

SDK for Java 2.x

Die folgenden Beispiele zeigen, wie in Bearbeitung befindliche (unvollständige) mehrteilige Uploads mithilfe von SDK für Java 2.x aufgelistet werden.

Example

```
public static void listMultiPartUploads( S3Client s3, String bucketName) {
       try {
           ListMultipartUploadsRequest listMultipartUploadsRequest =
ListMultipartUploadsRequest.builder()
                .bucket(bucketName)
                .build();

           ListMultipartUploadsResponse response =
s3.listMultipartUploads(listMultipartUploadsRequest);
           List MultipartUpload uploads = response.uploads();
           for (MultipartUpload upload: uploads) {
               System.out.println("Upload in progress: Key = \"" + upload.key() +
"\", id = " + upload.uploadId());
           }
     }
```

```
        catch (S3Exception e) {
              System.err.println(e.getMessage());
              System.exit(1);
        }
    }
```

SDK for Python

Die folgenden Beispiele zeigen, wie in Bearbeitung befindliche (unvollständige) mehrteilige Uploads mithilfe von for Python aufgelistet werden. SDK

Example

```
import logging
import boto3
from botocore.exceptions import ClientError

def list_multipart_uploads(s3_client, bucket_name):
    '''
    List any incomplete multipart uploads in a directory bucket in e specified gion

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket to check for incomplete multipart uploads
    :return: List of incomplete multipart uploads if there are any, None if not
    '''

    try:
        response = s3_client.list_multipart_uploads(Bucket = bucket_name)
        if 'Uploads' in response.keys():
            return response['Uploads']
        else:
            return None
    except ClientError as e:
        logging.error(e)

if __name__ == '__main__':
    bucket_name = 'BUCKET_NAME'
    region = 'us-west-2'
    s3_client = boto3.client('s3', region_name = region)
    multipart_uploads = list_multipart_uploads(s3_client, bucket_name)
    if multipart_uploads is not None:
        print (f'There are {len(multipart_uploads)} ncomplete multipart uploads for
 {bucket_name}')
```

```
    else:
        print (f'There are no incomplete multipart uploads for {bucket_name}')
```

Mit dem AWS CLI

Die folgenden Beispiele zeigen, wie Sie laufende (unvollständige) mehrteilige Uploads mithilfe von auflisten können. AWS CLI Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```
aws s3api list-multipart-uploads --bucket bucket-base-name--zone-id--x-s3
```

Weitere Informationen finden Sie list-multipart-uploadsin der AWS Command Line Interface.

Auflisten der Teile eines mehrteiligen Uploads

Die folgenden Beispiele zeigen, wie die Teile eines mehrteiligen Uploads in einen Verzeichnis-Bucket aufgelistet werden.

Mit dem AWS SDKs

SDK for Java 2.x

Die folgenden Beispiele zeigen, wie die Teile eines mehrteiligen Uploads in einen Verzeichnis-Bucket mithilfe von Java SDK 2.x aufgelistet werden.

```
public static void listMultiPartUploadsParts( S3Client s3, String bucketName, String
 objKey, String uploadID) {

        try {
            ListPartsRequest listPartsRequest = ListPartsRequest.builder()
                .bucket(bucketName)
                .uploadId(uploadID)
                .key(objKey)
                .build();

            ListPartsResponse response = s3.listParts(listPartsRequest);
            ListPart parts = response.parts();
            for (Part part: parts) {
                System.out.println("Upload in progress: Part number = \"" +
 part.partNumber() + "\", etag = " + part.eTag());
            }
```

```
        }

        catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }


    }
```

SDK for Python

Die folgenden Beispiele zeigen, wie die Teile eines mehrteiligen Uploads in einen Verzeichnis-Bucket mithilfe von SDK for Python aufgelistet werden.

```
import logging
import boto3
from botocore.exceptions import ClientError

def list_parts(s3_client, bucket_name, key_name, upload_id):
    '''
    Lists the parts that have been uploaded for a specific multipart upload to a
 directory bucket.

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket that multipart uploads parts have been uploaded to
    :param key_name: Name of the object that has parts uploaded
    :param upload_id: Multipart upload ID that the parts are associated with
    :return: List of parts associated with the specified multipart upload, None if
 there are no parts
    '''
    parts_list = []
    next_part_marker = ''
    continuation_flag = True
    try:
        while continuation_flag:
            if next_part_marker == '':
                response = s3_client.list_parts(
                    Bucket = bucket_name,
                    Key = key_name,
                    UploadId = upload_id
                )
```

```
            else:
                response = s3_client.list_parts(
                    Bucket = bucket_name,
                    Key = key_name,
                    UploadId = upload_id,
                    NextPartMarker = next_part_marker
                )
            if 'Parts' in response:
                for part in response['Parts']:
                    parts_list.append(part)
                if response['IsTruncated']:
                    next_part_marker = response['NextPartNumberMarker']
                else:
                    continuation_flag = False
            else:
                continuation_flag = False
        return parts_list
    except ClientError as e:
        logging.error(e)
        return None

if __name__ == '__main__':
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    key_name = 'KEY_NAME'
    upload_id = 'UPLOAD_ID'
    s3_client = boto3.client('s3', region_name = region)
    parts_list = list_parts(s3_client, bucket_name, key_name, upload_id)
    if parts_list is not None:
        print (f'{key_name} has {len(parts_list)} parts uploaded to {bucket_name}')
    else:
        print (f'There are no multipart uploads with that upload ID for
 {bucket_name} bucket')
```

Mit dem AWS CLI

Die folgenden Beispiele zeigen, wie die Teile eines mehrteiligen Uploads in einen Verzeichnis-Bucket mithilfe von aufgelistet werden. AWS CLI Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```
aws s3api list-parts --bucket bucket-base-name--zone-id--x-s3 --key KEY_NAME --upload-
id
 "AS_mgt9RaQE9GEaifATue15dAAAAAAAAAAEMAAAAAAAAADQwNzI4MDU0MjUyMBYAAAAAAAAAAA0AAAAAAAAAAH2AfYAA
```

Weitere Informationen finden Sie in der Liste der Teile. AWS Command Line Interface

Objekte aus oder in einen Verzeichnis-Bucket kopieren

Die Kopieroperation erzeugt eine Kopie eines Objekts, das bereits in Amazon S3 gespeichert ist. Sie können Objekte zwischen Verzeichnis-Buckets und Allzweck-Buckets kopieren. Sie können Objekte auch innerhalb eines Buckets und zwischen Buckets desselben Typs kopieren, z. B. von Verzeichnis-Bucket zu Verzeichnis-Bucket.

Sie können in einem einzigen atomaren Vorgang eine Kopie eines Objekts mit einer Größe von bis zu 5 GB erstellen. Um jedoch ein Objekt zu kopieren, das größer als 5 GB ist, müssen Sie die mehrteiligen API Upload-Operationen verwenden. Weitere Informationen finden Sie unter Mehrteilige Uploads mit Verzeichnis-Buckets verwenden.

Berechtigungen

Um Objekte zu kopieren, müssen Sie über die folgenden Berechtigungen verfügen:

- Um Objekte von einem Verzeichnis-Bucket zu einem anderen Verzeichnis-Bucket zu kopieren, benötigen Sie die `s3express:CreateSession`-Berechtigung.
- Um Objekte aus Verzeichnis-Buckets zu Allzweck-Buckets zu kopieren, benötigen Sie die `s3express:CreateSession`-Berechtigung und `s3:PutObject`-Berechtigung zum Schreiben der Objektkopie in den Ziel-Bucket.
- Um Objekte aus Allzweck-Buckets in Verzeichnis-Buckets zu kopieren, benötigen Sie die `s3express:CreateSession` `s3:GetObject` Erlaubnis und das Recht, das kopierte Quellobjekt zu lesen.

  Weitere Informationen finden Sie unter CopyObjectin der Amazon Simple Storage Service API Reference.

Verschlüsselung

Amazon S3 verschlüsselt automatisch alle neuen Objekte, die in einen S3-Bucket hochgeladen werden. Die Standardverschlüsselungskonfiguration eines S3-Buckets ist immer aktiviert und

mindestens auf serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) eingestellt.

Für Verzeichnis-Buckets werden SSE -S3 und serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-) unterstützt. KMS Wenn es sich bei dem Ziel-Bucket um einen Verzeichnis-Bucket handelt, empfehlen wir, dass die Standardverschlüsselung des Ziel-Buckets die gewünschte Verschlüsselungskonfiguration verwendet und Sie die Standardverschlüsselung des Buckets nicht überschreiben. Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Außerdem werden S3-Bucket-Keys nicht unterstützt, wenn Sie KMS verschlüsselte Objekte von Allzweck-Buckets in Verzeichnis-Buckets, von Verzeichnis-Buckets in Allzweck-Buckets oder zwischen Verzeichnis-Buckets kopieren SSE CopyObject. In diesem Fall ruft Amazon S3 AWS KMS jedes Mal, wenn eine Kopieranforderung für ein KMS -verschlüsseltes Objekt gestellt wird, auf. Weitere Informationen zum Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets finden Sie unter Serverseitige Verschlüsselung AWS KMS für Uploads neuer Objekte angeben.

Für allgemeine Buckets können Sie SSE -S3 (Standard), serverseitige Verschlüsselung mit () Schlüsseln (-), zweischichtige serverseitige Verschlüsselung mit AWS Key Management Service Schlüsseln (SSE-KMS) oder serverseitige Verschlüsselung mit vom Kunden AWS KMS bereitgestellten Schlüsseln (DSSEKMS-C) verwenden.AWS KMS SSE

Wenn Sie eine Kopieranforderung stellen, die angibt, DSSE - KMS oder SSE -C für einen Verzeichnis-Bucket (entweder den Quell- oder den Ziel-Bucket) zu verwenden, gibt die Antwort einen Fehler zurück.

Tags

Verzeichnis-Buckets unterstützen keine Tags. Wenn Sie ein Objekt mit Tags aus einem Allzweck-Bucket in einen Verzeichnis-Bucket kopieren, erhalten Sie eine HTTP `501 (Not Implemented)` Antwort. Weitere Informationen finden Sie unter CopyObjectin der Amazon Simple Storage Service API Reference.

ETags

Entitäts-Tags (ETags) für S3 Express One Zone sind zufällige alphanumerische Zeichenfolgen und keine MD5 Prüfsummen. Verwenden Sie zusätzliche Prüfsummen, um die Objektintegrität sicherzustellen.

Zusätzliche Prüfsummen

S3 Express One Zone bietet Ihnen die Möglichkeit, den Prüfsummenalgorithmus auszuwählen, der zur Validierung Ihrer Daten beim Hoch- oder Herunterladen verwendet wird. Sie können einen der folgenden Algorithmen zur Prüfung der Datenintegrität mit Secure Hash Algorithms (SHA) oder Cyclic Redundancy Check (CRC) auswählen:CRC32,CRC32C, SHA -1 und -256. SHA MD5basierte Prüfsummen werden von der S3 Express One Zone-Speicherklasse nicht unterstützt.

Weitere Informationen finden Sie unter Bewährte Methoden für zusätzliche S3-Prüfsummen.

Unterstützte Features

Weitere Informationen darüber, welche Amazon S3-Funktionen für S3 Express One Zone unterstützt werden, finden Sie unterUnterschiede bei Directory-Buckets.

Verwenden der S3-Konsole (Kopieren in einen Verzeichnis-Bucket)

> **Note**
>
> Die Einschränkungen und Einschränkungen beim Kopieren eines Objekts in einen Verzeichnis-Bucket mit der Konsole lauten wie folgt:
>
> - Die `Copy` Aktion gilt für alle Objekte in den angegebenen Ordnern (Präfixe). Objekte, die diesen Ordnern hinzugefügt wurden, während die Aktion ausgeführt wird, können betroffen sein.
> - Objekte, die mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) verschlüsselt wurden, können nicht mit der S3-Konsole kopiert werden. Verwenden Sie, oder Amazon S3 RESTAPI, um mit SSE -C verschlüsselte Objekte zu kopieren. AWS CLI AWS SDK
> - Bei kopierten Objekten werden die Objektsperren-Einstellungen der Originalobjekte nicht beibehalten.
> - Wenn der Bucket, aus dem Sie Objekte kopieren, die vom Bucket-Besitzer erzwungene Einstellung für S3-Objekteigentum verwendet, ACLs wird das Objekt nicht an das angegebene Ziel kopiert.
> - Wenn Sie Objekte in einen Bucket kopieren möchten, der die vom Bucket-Besitzer erzwungene Einstellung für S3-Objekteigentum verwendet, stellen Sie sicher, dass der Quell-Bucket auch die erzwungene Einstellung des Bucket-Besitzers verwendet, oder entfernen Sie alle ACL Objektzuweisungen an andere AWS Konten und Gruppen.
> - Objekte, die aus einem Allzweck-Bucket in einen Verzeichnis-Bucket kopiert wurden, ACLs behalten keine Objekt-Tags oder Etag-Werte bei. Prüfsummenwerte können kopiert

werden, entsprechen aber nicht einem Etag. Der Prüfsummenwert kann sich im Vergleich zum Zeitpunkt des Hinzufügens ändern.

- Für alle Objekte, die in einen Verzeichnis-Bucket kopiert werden, wird die Einstellung „Besitzer des Buckets“ für S3 Object Ownership erzwungen.

Um ein Objekt aus einem Allzweck-Bucket oder einem Verzeichnis-Bucket in einen Verzeichnis-Bucket zu kopieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie den Bucket aus, aus dem Sie Objekte kopieren möchten:
   - Um aus einem Allzweck-Bucket zu kopieren, wählen Sie die Registerkarte Allzweck-Buckets.
   - Um aus einem Verzeichnis-Bucket zu kopieren, wählen Sie den Tab Directory-Buckets.
4. Wählen Sie den Allzweck-Bucket oder den Verzeichnis-Bucket, der die Objekte enthält, die Sie kopieren möchten.
5. Wählen Sie dieObjekte-Tag. Aktivieren Sie auf der Seite Objekte das Kontrollkästchen links neben den Namen der Objekte, die Sie kopieren möchten.
6. Wählen Sie im Menü Actions (Aktionen) die Option Copy (Kopieren) aus.

   Die Seite Kopieren wird angezeigt.
7. Wählen Sie unter Ziel die Option Directory-Bucket für Ihren Zieltyp aus. Um den Zielpfad anzugeben, wählen Sie Browse S3 aus, navigieren Sie zum Ziel und wählen Sie dann das Optionsfeld links neben dem Ziel. Wählen Sie unten rechts Choose destination (Ziel auswählen) aus.

   Geben Sie alternativ den Zielpfad ein.
8. Wählen Sie unter Zusätzliche Kopiereinstellungen aus, ob Sie Quelleinstellungen kopieren, Keine Einstellungen angeben oder Einstellungen angeben möchten. Quelleinstellungen kopieren ist die Standardoption. Wenn Sie das Objekt nur ohne die Quelleinstellungsattribute kopieren möchten, wählen Sie „Keine Einstellungen angeben“. Wählen Sie „Einstellungen angeben“, um Einstellungen für serverseitige Verschlüsselung, Prüfsummen und Metadaten anzugeben.
9. Wählen Sie unten rechts Copy (Kopieren) aus. Amazon S3 kopiert Ihre Objekte in den Zielordner.

Verwenden der S3-Konsole (Kopieren zu einem Allzweck-Bucket)

> **Note**
>
> Beim Kopieren eines Objekts in einen Allzweck-Bucket mit der Konsole gelten folgende Einschränkungen und Einschränkungen:
>
> - Die `Copy` Aktion gilt für alle Objekte in den angegebenen Ordnern (Präfixe). Objekte, die diesen Ordnern hinzugefügt wurden, während die Aktion ausgeführt wird, können betroffen sein.
> - Objekte, die mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) verschlüsselt wurden, können nicht mit der S3-Konsole kopiert werden. Verwenden Sie, oder Amazon S3 RESTAPI, um mit SSE -C verschlüsselte Objekte zu kopieren. AWS CLI AWS SDK
> - Bei kopierten Objekten werden die Objektsperren-Einstellungen der Originalobjekte nicht beibehalten.
> - Wenn der Bucket, aus dem Sie Objekte kopieren, die vom Bucket-Besitzer erzwungene Einstellung für S3-Objekteigentum verwendet, ACLs wird das Objekt nicht an das angegebene Ziel kopiert.
> - Wenn Sie Objekte in einen Bucket kopieren möchten, der die vom Bucket-Besitzer erzwungene Einstellung für S3-Objekteigentum verwendet, stellen Sie sicher, dass der Quell-Bucket auch die erzwungene Einstellung des Bucket-Besitzers verwendet, oder entfernen Sie alle ACL Objektzuweisungen an andere AWS Konten und Gruppen.

So kopieren Sie ein Objekt aus einem Verzeichnis-Bucket in einen Allzweck-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie den Tab Directory-Buckets.
4. Wählen Sie den Verzeichnis-Bucket aus, der die Objekte enthält, die Sie kopieren möchten.
5. Wählen Sie dieObjekte-Tag. Aktivieren Sie auf der Seite Objekte das Kontrollkästchen links neben den Namen der Objekte, die Sie kopieren möchten.
6. Wählen Sie im Menü Actions (Aktionen) die Option Copy (Kopieren) aus.

7. Wählen Sie unter Ziel die Option Allzweck-Bucket für Ihren Zieltyp aus. Um den Zielpfad anzugeben, wählen Sie Browse S3, navigieren Sie zum Ziel und wählen Sie das Optionsfeld links neben dem Ziel. Wählen Sie unten rechts Choose destination (Ziel auswählen) aus.

   Geben Sie alternativ den Zielpfad ein.

8. Wählen Sie unter Zusätzliche Kopiereinstellungen aus, ob Sie Quelleinstellungen kopieren, Keine Einstellungen angeben oder Einstellungen angeben möchten. Quelleinstellungen kopieren ist die Standardoption. Wenn Sie das Objekt nur ohne die Quelleinstellungsattribute kopieren möchten, wählen Sie „Keine Einstellungen angeben“. Wählen Sie „Einstellungen angeben“, um Einstellungen für SpeicherklasseACLs, Objekt-Tags, Metadaten, serverseitige Verschlüsselung und zusätzliche Prüfsummen anzugeben.
9. Wählen Sie unten rechts Copy (Kopieren) aus. Amazon S3 kopiert Ihre Objekte in den Zielordner.

Mit dem AWS SDKs

SDK for Java 2.x

Example

```
public static void copyBucketObject (S3Client s3, String sourceBucket, String
objectKey, String targetBucket) {
     CopyObjectRequest copyReq = CopyObjectRequest.builder()
         .sourceBucket(sourceBucket)
         .sourceKey(objectKey)
         .destinationBucket(targetBucket)
         .destinationKey(objectKey)
         .build();
      String temp = "";

      try {
          CopyObjectResponse copyRes = s3.copyObject(copyReq);
          System.out.println("Successfully copied " + objectKey +" from bucket " +
sourceBucket +" into bucket "+targetBucket);
      }

      catch (S3Exception e) {
          System.err.println(e.awsErrorDetails().errorMessage());
          System.exit(1);
      }
```

```
}
```

Mit dem AWS CLI

Der folgende `copy-object` Beispielbefehl zeigt, wie Sie mit dem AWS CLI ein Objekt von einem Bucket in einen anderen Bucket kopieren können. Sie können Objekte zwischen Bucket-Typen kopieren. Um diesen Befehl auszuführen, ersetzen Sie die Platzhalter für Benutzereingaben durch Ihre eigenen Informationen.

```
aws s3api copy-object --copy-source SOURCE_BUCKET/SOURCE_KEY_NAME --key TARGET_KEY_NAME
 --bucket TARGET_BUCKET_NAME
```

Weitere Informationen finden Sie unter copy-objectin der AWS CLI Befehlsreferenz.

Objekte aus einem Verzeichnis-Bucket löschen

Sie können Objekte aus einem Amazon S3-Verzeichnis-Bucket löschen, indem Sie die Amazon S3 S3-Konsole verwenden, AWS Command Line Interface (AWS CLI) oder AWS SDKs. Weitere Informationen erhalten Sie unter Arbeiten mit Verzeichnis-Buckets und S3 Express One Zone.

Warning

- Das Löschen eines Objekts kann nicht rückgängig gemacht werden.
- Diese Aktion löscht alle angegebenen Objekte. Warten Sie beim Löschen von Ordnern, bis die Löschaktion abgeschlossen ist, bevor Sie dem Ordner neue Objekte hinzufügen. Andernfalls könnten auch neue Objekte gelöscht werden.

Note

Beachten Sie Folgendes, wenn Sie programmgesteuert mehrere Objekte aus einem Verzeichnis-Bucket löschen:

- Objektschlüssel in `DeleteObjects`-Anforderungen müssen mindestens ein Zeichen enthalten, das kein Leerzeichen ist. Zeichenketten, die ausschließlich Leerzeichen enthalten, werden nicht unterstützt.

- Objektschlüssel in `DeleteObjects` Anfragen dürfen keine Unicode-Steuerzeichen enthalten, mit Ausnahme von Newline (`\n`), Tab (`\t`) und Carriage Return (`\r`).

Verwenden der S3-Konsole

So löschen Sie Objekte

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie die Registerkarte Verzeichnis-Buckets.
4. Wählen Sie den Verzeichnis-Bucket aus, der die Objekte enthält, die Sie löschen möchten.
5. Wählen Sie dieObjekte-Tag. Aktivieren Sie in der Objektliste das Kontrollkästchen links neben dem Objekt oder den Objekten, die Sie löschen möchten.
6. Wählen Sie Löschen.
7. Geben Sie auf der Seite Objekte löschen den Text **`permanently delete`** in das Textfeld ein.
8. Wählen Sie Delete objects (Objekte löschen).

Verwenden Sie den AWS SDKs

SDK for Java 2.x

Example

Im folgenden Beispiel werden Objekte in einem Verzeichnis-Bucket mithilfe von gelöscht. AWS SDK for Java 2.x

```
static void deleteObject(S3Client s3Client, String bucketName, String objectKey) {



        try {

            DeleteObjectRequest del = DeleteObjectRequest.builder()
                    .bucket(bucketName)
                    .key(objectKey)
                    .build();
```

```
            s3Client.deleteObject(del);

            System.out.println("Object " + objectKey + " has been deleted");


        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }

    }
```

SDK for Python

Example

Im folgenden Beispiel werden Objekte in einem Verzeichnis-Bucket mithilfe von gelöscht. AWS SDK for Python (Boto3)

```
import logging
import boto3
from botocore.exceptions import ClientError

def delete_objects(s3_client, bucket_name, objects):
    '''
    Delete a list of objects in a directory bucket

    :param s3_client: boto3 S3 client
    :param bucket_name: Bucket that contains objects to be deleted; for example,
 'doc-example-bucket--usw2-az1--x-s3'
    :param objects: List of dictionaries that specify the key names to delete
    :return: Response output, else False
    '''

    try:
        response = s3_client.delete_objects(
            Bucket = bucket_name,
            Delete = {
                'Objects': objects
            }
        )
        return response
```

```
    except ClientError as e:
        logging.error(e)
        return False


if __name__ == '__main__':
    region = 'us-west-2'
    bucket_name = 'BUCKET_NAME'
    objects = [
        {
            'Key': '0.txt'
        },
        {
            'Key': '1.txt'
        },
        {
            'Key': '2.txt'
        },
        {
            'Key': '3.txt'
        },
        {
            'Key': '4.txt'
        }
    ]

    s3_client = boto3.client('s3', region_name = region)
    results = delete_objects(s3_client, bucket_name, objects)
    if results is not None:
        if 'Deleted' in results:
            print (f'Deleted {len(results["Deleted"])} objects from {bucket_name}')
        if 'Errors' in results:
            print (f'Failed to delete {len(results["Errors"])} objects from
 {bucket_name}')
```

Mit dem AWS CLI

Der folgende `delete-object` Beispielbefehl zeigt, wie Sie mit dem AWS CLI ein Objekt aus einem Verzeichnis-Bucket löschen können. Zum Ausführen dieses Befehls ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws s3api delete-object --bucket bucket-base-name--zone-id--x-s3 --key KEY_NAME
```

Weitere Informationen finden Sie unter delete-objectin der AWS CLI Befehlsreferenz.

Ein Objekt aus einem Verzeichnis-Bucket herunterladen

Die folgenden Codebeispiele zeigen, wie mithilfe der `GetObject` API Operation Daten aus einem Objekt in einem Amazon S3-Verzeichnis-Bucket gelesen (heruntergeladen) werden.

Mit dem AWS SDKs

SDK for Java 2.x

Example

Das folgende Codebeispiel zeigt, wie Daten aus einem Objekt in einem Verzeichnis-Bucket mithilfe von gelesen werden AWS SDK for Java 2.x.

```
public static void getObject(S3Client s3Client, String bucketName, String objectKey)
 {
     try {
         GetObjectRequest objectRequest = GetObjectRequest
            .builder()
            .key(objectKey)
            .bucket(bucketName)
            .build();

         ResponseBytes GetObjectResponse objectBytes =
 s3Client.getObjectAsBytes(objectRequest);
         byte[] data = objectBytes.asByteArray();

         //Print object contents to console
         String s = new String(data, StandardCharsets.UTF_8);
         System.out.println(s);
    }

    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
```

SDK for Python

Example

Das folgende Codebeispiel zeigt, wie Daten aus einem Objekt in einem Verzeichnis-Bucket mithilfe von gelesen werden AWS SDK for Python (Boto3).

```
import boto3
from botocore.exceptions import ClientError
from botocore.response import StreamingBody

def get_object(s3_client: boto3.client, bucket_name: str, key_name: str) ->
 StreamingBody:
    """
    Gets the object.
    :param s3_client:
    :param bucket_name: The bucket that contains the object.
    :param key_name: The key of the object to be downloaded.
    :return: The object data in bytes.
    """
    try:
        response = s3_client.get_object(Bucket=bucket_name, Key=key_name)
        body = response['Body'].read()
        print(f"Got object '{key_name}' from bucket '{bucket_name}'.")
    except ClientError:
        print(f"Couldn't get object '{key_name}' from bucket '{bucket_name}'.")
        raise
    else:
        return body

def main():
    s3_client = boto3.client('s3')
    resp = get_object(s3_client, 'doc-example-bucket--use1-az4--x-s3', 'sample.txt')
    print(resp)

if __name__ == "__main__":
     main()
```

Mit dem AWS CLI

Das folgende Beispiel für `get-object` veranschaulicht, wie Sie die AWS CLI verwenden können, um ein Objekt von Amazon S3 herunterzuladen. Mit diesem Befehl wird das Objekt *KEY_NAME* aus

dem Verzeichnis-Bucket abgerufen*bucket-base-name*--*zone-id*--x-s3. Das Objekt wird in eine Datei mit dem Namen *LOCAL_FILE_NAME* heruntergeladen. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3api get-object --bucket bucket-base-name--zone-id--x-s3 --
key KEY_NAME LOCAL_FILE_NAME
```

Weitere Informationen finden Sie unter get-objectin der AWS CLI Befehlsreferenz.

Objektmetadaten aus Verzeichnis-Buckets abrufen

Im Folgenden AWS SDK und in den AWS CLI Beispielen wird gezeigt, wie der `HeadObject` API Vorgang verwendet wird, um Metadaten von einem Objekt in einem Amazon S3 S3-Verzeichnis-Bucket abzurufen, ohne das Objekt selbst zurückzugeben.

Mit dem AWS SDKs

SDK for Java 2.x

Example

```
public static void headObject(S3Client s3Client, String bucketName, String
 objectKey) {
    try {
        HeadObjectRequest headObjectRequest = HeadObjectRequest
                .builder()
                .bucket(bucketName)
                .key(objectKey)
                .build();
        HeadObjectResponse response = s3Client.headObject(headObjectRequest);
        System.out.format("Amazon S3 object: \"%s\" found in bucket: \"%s\" with
 ETag: \"%s\"", objectKey, bucketName, response.eTag());
    }
    catch (S3Exception e) {
        System.err.println(e.awsErrorDetails().errorMessage());
```

Mit dem AWS CLI

Der folgende `head-object` Beispielbefehl zeigt, wie Sie mit AWS CLI dem Metadaten aus einem Objekt abrufen können. Zum Ausführen dieses Befehls ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws s3api head-object --bucket bucket-base-name--zone-id--x-s3 --key KEY_NAME
```

Weitere Informationen finden Sie unter head-objectin der AWS CLI Befehlsreferenz.

Sicherheit für Verzeichnis-Buckets

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen. Sicherheit ist eine gemeinsame AWS Verantwortung von Ihnen und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS-Services in der läuft AWS Cloud. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheit im Rahmen der AWS Compliance Programs.

  Weitere Informationen zu den Compliance-Programmen finden Sie unter AWS-Services in Scope by Compliance Program.
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS-Service , was Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, die Anforderungen Ihres Unternehmens und die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Verwendung von Directory-Buckets anwenden können. In den folgenden Themen erfahren Sie, wie Sie Directory-Buckets konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere verwenden können AWS-Services , die Ihnen bei der Überwachung und Sicherung Ihrer Objekte in Directory-Buckets helfen können.

Datenschutz und Verschlüsselung

Weitere Informationen dazu, wie Sie die Verschlüsselung für Verzeichnis-Buckets konfigurieren können, finden Sie in den folgenden Themen.

Themen

- Server-side encryption
- Einstellung und Überwachung der Standardverschlüsselung für Verzeichnis-Buckets
- Serverseitige Verschlüsselung mit AWS KMS Schlüsseln (SSE-KMS) in Verzeichnis-Buckets verwenden
- Verschlüsselung während der Übertragung
- Löschen von Daten

Server-side encryption

Für alle Verzeichnis-Buckets ist standardmäßig Verschlüsselung konfiguriert, und alle neuen Objekte, die in Verzeichnis-Buckets hochgeladen werden, werden im Ruhezustand automatisch verschlüsselt. Die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Standardverschlüsselungskonfiguration für jeden Verzeichnis-Bucket. Wenn Sie einen anderen Verschlüsselungstyp angeben möchten, können Sie die serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-KMS) verwenden, indem Sie die Standardverschlüsselungskonfiguration des Buckets festlegen. Weitere Hinweise zu SSE - KMS in Verzeichnis-Buckets finden Sie unter. Serverseitige Verschlüsselung mit AWS KMS Schlüsseln (SSE-KMS) in Verzeichnis-Buckets verwenden

Wir empfehlen, dass die Standardverschlüsselung des Buckets die gewünschte Verschlüsselungskonfiguration verwendet und dass Sie die Standardverschlüsselung des Buckets in Ihren `CreateSession` Anfragen oder `PUT` Objektanforderungen nicht überschreiben. Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Weitere Informationen zum Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets finden Sie unter Serverseitige Verschlüsselung AWS KMS für Uploads neuer Objekte angeben.

SSE- KMS mit Verzeichnis-Buckets unterscheidet sich von SSE - KMS in Allzweck-Buckets in den folgenden Aspekten.

- Ihre KMS Konfiguration SSE — unterstützt für die gesamte Lebensdauer des Buckets nur einen vom Kunden verwalteten Schlüssel pro Verzeichnis-Bucket. Die Von AWS verwalteter Schlüssel(`aws/s3`) wird nicht unterstützt. Außerdem können Sie, nachdem Sie einen vom Kunden verwalteten Schlüssel für SSE - angegeben habenKMS, den vom Kunden verwalteten Schlüssel für die KMS Konfiguration des SSE Buckets nicht überschreiben.

  Sie können den vom Kunden verwalteten Schlüssel, den Sie für die KMS Konfiguration SSE — des Buckets angegeben haben, wie folgt identifizieren:

  - Sie stellen eine `HeadObject` API Operationsanfrage, um den Wert von `x-amz-server-side-encryption-aws-kms-key-id` in Ihrer Antwort zu ermitteln.

  Um einen neuen kundenverwalteten Schlüssel für Ihre Daten zu verwenden, empfehlen wir, Ihre vorhandenen Objekte mit einem neuen kundenverwalteten Schlüssel in einen neuen Verzeichnis-Bucket zu kopieren.

- Bei APIVorgängen auf zonaler Endpunktebene (Objektebene) mit Ausnahme von CopyObjectund UploadPartCopyauthentifizieren und autorisieren Sie Anfragen mit geringer Latenz. CreateSession Wir empfehlen, dass die Standardverschlüsselung des Buckets die gewünschte Verschlüsselungskonfiguration verwendet und dass Sie die Standardverschlüsselung des Buckets in Ihren `CreateSession` Anfragen oder Objektanforderungen nicht überschreiben. PUT Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Um neue Objekte in einem Verzeichnis-Bucket mit SSE - zu verschlüsselnKMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom Kunden verwalteten Schlüssel) angeben. Wenn dann eine Sitzung für zonale API Endpunktoperationen erstellt wird, werden neue Objekte während der Sitzung automatisch mit SSE - und S3-Bucket-Schlüsseln ver KMS - und entschlüsselt. Weitere Informationen zum Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets finden Sie unter Serverseitige Verschlüsselung mit AWS KMS für Uploads neuer Objekte angeben.

  Bei API Aufrufen von zonalen Endpunkten (außer CopyObjectund UploadPartCopy) können Sie die Werte der Verschlüsselungseinstellungen (`x-amz-server-side-encryption`, `x-amz-server-side-encryption-aws-kms-key-idx-amz-server-side-encryption-context`, und`x-amz-server-side-encryption-bucket-key-enabled`) aus der Anforderung nicht überschreiben. `CreateSession` Sie müssen diese Werte für die Verschlüsselungseinstellungen nicht explizit in API Aufrufen von zonalen Endpunkten angeben, und Amazon S3 verwendet die Werte der Verschlüsselungseinstellungen aus der `CreateSession` Anfrage, um neue Objekte im Verzeichnis-Bucket zu schützen.

> **Note**
>
> Wenn Sie das AWS CLI oder das AWS SDKs, für verwenden, wird das Sitzungstoken automatisch aktualisiert`CreateSession`, um Serviceunterbrechungen zu vermeiden, wenn eine Sitzung abläuft. Das AWS CLI oder sie AWS SDKs verwenden die Standardverschlüsselungskonfiguration des Buckets für die `CreateSession` Anfrage. Es wird nicht unterstützt, die Werte der Verschlüsselungseinstellungen in der `CreateSession` Anfrage zu überschreiben. Außerdem wird es bei API Aufrufen von zonalen Endpunkten (außer CopyObjectund UploadPartCopy) nicht unterstützt, die Werte der Verschlüsselungseinstellungen aus der `CreateSession` Anfrage zu überschreiben.

- Um neue Objektkopien in einem Verzeichnis-Bucket mit SSE - zu verschlüsselnKMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom Kunden verwalteten Schlüssel) angeben. CopyObject Wenn Sie dann serverseitige Verschlüsselungseinstellungen für neue Objektkopien mit SSE - angeben, müssen Sie sicherstellenKMS, dass der Verschlüsselungsschlüssel derselbe vom Kunden verwaltete Schlüssel ist, den Sie für die Standardverschlüsselungskonfiguration des Directory-Buckets angegeben haben. Um neue Objektteilkopien in einem Verzeichnis-Bucket mit SSE - zu verschlüsselnKMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom Kunden verwalteten Schlüssel) angeben. UploadPartCopy Sie können keine serverseitigen Verschlüsselungseinstellungen für neue Objektteilkopien mit SSE - KMS in den UploadPartCopyAnforderungsheadern angeben. Außerdem müssen die Verschlüsselungseinstellungen, die Sie in der CreateMultipartUploadAnfrage angeben, mit der Standardverschlüsselungskonfiguration des Ziel-Buckets übereinstimmen.
- S3-Bucket-Keys sind immer für `GET` und für `PUT` Operationen in einem Verzeichnis-Bucket aktiviert und können nicht deaktiviert werden. S3-Bucket-Keys werden nicht unterstützt, wenn Sie KMS verschlüsselte Objekte von Allzweck-Buckets in Verzeichnis-Buckets, von Verzeichnis-Buckets in Allzweck-Buckets oder zwischen Verzeichnis-Buckets kopieren SSE CopyObject, UploadPartCopy, der Copy Operation in Batch-Operationen oder die import Jobs. In diesem Fall ruft Amazon S3 AWS KMS jedes Mal, wenn eine Kopieranforderung für ein KMS -verschlüsseltes Objekt gestellt wird, auf.
- Wenn Sie in Ihrem Verzeichnis-Bucket einen vom AWS KMS Kunden verwalteten Schlüssel für die Verschlüsselung angeben, verwenden Sie nur die Schlüssel-ID oder den SchlüsselARN. Das Schlüsselaliasformat des KMS Schlüssels wird nicht unterstützt.

Directory-Buckets unterstützen keine zweischichtige serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (DSSE-KMS) oder serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (-C). SSE

Einstellung und Überwachung der Standardverschlüsselung für Verzeichnis-Buckets

In Amazon S3 S3-Buckets ist die Bucket-Verschlüsselung standardmäßig aktiviert, und neue Objekte werden automatisch mithilfe serverseitiger Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt. Diese Verschlüsselung gilt für alle neuen Objekte in Ihren Amazon-S3-Buckets und es fallen keine Kosten für Sie an.

Wenn Sie mehr Kontrolle über Ihre Verschlüsselungsschlüssel benötigen, z. B. die Verwaltung der Schlüsselrotation und die Gewährung von Zugriffsrichtlinien, können Sie sich für die serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (-) entscheiden. SSE KMS

> **Note**
>
> - Wir empfehlen, dass die Standardverschlüsselung des Buckets die gewünschte Verschlüsselungskonfiguration verwendet und dass Sie die Standardverschlüsselung des Buckets in Ihren `CreateSession` Anfragen oder `PUT` Objektanforderungen nicht überschreiben. Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Weitere Informationen zum Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets finden Sie unter Serverseitige Verschlüsselung AWS KMS für Uploads neuer Objekte angeben.
> - Um neue Objekte in einem Verzeichnis-Bucket mit SSE - zu verschlüsselnKMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom Kunden verwalteten Schlüssel) angeben. Wenn dann eine Sitzung für zonale API Endpunktoperationen erstellt wird, werden neue Objekte während der Sitzung automatisch mit SSE - und S3-Bucket-Schlüsseln ver KMS - und entschlüsselt.
> - Wenn Sie die Standard-Bucket-Verschlüsselung auf SSE - setzenKMS, sind S3-Bucket-Keys immer für `GET` `PUT` Vorgänge in einem Verzeichnis-Bucket aktiviert und können nicht deaktiviert werden. S3-Bucket-Keys werden nicht unterstützt, wenn Sie KMS verschlüsselte Objekte von Allzweck-Buckets in Verzeichnis-Buckets, von Verzeichnis-Buckets in Allzweck-Buckets oder zwischen Verzeichnis-Buckets kopierenSSE, bis CopyObject, UploadPartCopy, der Copy Operation in Batch-Operationen oder die import Jobs. In

diesem Fall ruft Amazon S3 AWS KMS jedes Mal, wenn eine Kopieranforderung für ein KMS -verschlüsseltes Objekt gestellt wird, auf. Weitere Informationen darüber, wie S3 Bucket Keys Ihre AWS KMS Anforderungskosten reduzieren, finden Sie unterSenkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys.

- Wenn Sie einen vom AWS KMS Kunden verwalteten Schlüssel für die Verschlüsselung in Ihrem Verzeichnis-Bucket angeben, verwenden Sie nur die Schlüssel-ID oder den SchlüsselARN. Das Schlüsselaliasformat des KMS Schlüssels wird nicht unterstützt.
- Die zweischichtige serverseitige Verschlüsselung mit AWS KMS Schlüsseln (DSSE-KMS) und die serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) werden für die Standardverschlüsselung in Verzeichnis-Buckets nicht unterstützt.

Weitere Informationen zur Konfiguration der Standardverschlüsselung finden Sie unter. Konfigurieren der Standardverschlüsselung

Weitere Informationen zu den für die Standardverschlüsselung erforderlichen Berechtigungen finden Sie unter PutBucketEncryptionin der Amazon Simple Storage Service API Reference.

Sie können die Amazon S3-Standardverschlüsselung für einen S3-Bucket konfigurieren, indem Sie die AWS SDKs Amazon S3 S3-Konsole RESTAPI, Amazon S3 und AWS Command Line Interface (AWS CLI) verwenden.

Verwenden der S3-Konsole

Konfigurieren der Standardverschlüsselung für einen Amazon-S3-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des von Ihnen erstellten Buckets aus.
4. Wählen Sie die Registerkarte Eigenschaften aus.
5. Unter Serverseitige Verschlüsselungseinstellungen verwenden Verzeichnis-Buckets serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3).
6. Wählen Sie Änderungen speichern.

Mit dem AWS CLI

Diese Beispiele zeigen Ihnen, wie Sie die Standardverschlüsselung mithilfe von SSE -S3 oder mithilfe von SSE - KMS mit einem S3-Bucket-Key konfigurieren.

Weitere Informationen zur Standardverschlüsselung finden Sie unter Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets. Weitere Informationen AWS CLI zur Konfiguration der Standardverschlüsselung finden Sie unter put-bucket-encryption.

Example — Standardverschlüsselung mit SSE -S3

In diesem Beispiel wird die Standardverschlüsselung von Buckets mit von Amazon S3 verwalteten Schlüsseln konfiguriert. Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```
aws s3api put-bucket-encryption --bucket bucket-base-name--zone-id--x-s3 --server-side-
encryption-configuration '{
    "Rules": [
        {
            "ApplyServerSideEncryptionByDefault": {
                "SSEAlgorithm": "AES256"
            }
        }
    ]
}'
```

Example — Standardverschlüsselung mit SSE — KMS unter Verwendung eines S3-Bucket-Keys

In diesem Beispiel wird die Standard-Bucket-Verschlüsselung mit SSE - KMS unter Verwendung eines S3-Bucket-Keys konfiguriert. Um den Befehl zu verwenden, ersetzen Sie den *user input placeholders* mit Ihren eigenen Informationen.

```
aws s3api put-bucket-encryption --bucket bucket-base-name--zone-id--x-s3 --server-side-
encryption-configuration '{
    "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "aws:kms",
                    "KMSMasterKeyID": "KMS-Key-ARN"
                },
                "BucketKeyEnabled": true
            }
```

```
        ]
    }'
```

Mit dem REST API

Verwenden Sie den REST API `PutBucketEncryption` Vorgang, um die Standardverschlüsselung mit einem serverseitigen Verschlüsselungstyp festzulegen — SSE -S3 oder SSE -. KMS

Weitere Informationen finden Sie unter PutBucketEncryptionin der Amazon Simple Storage Service API Reference.

Mit dem AWS SDKs

Bei der Verwendung AWS SDKs können Sie Amazon S3 AWS KMS keys für die serverseitige Verschlüsselung anfordern. Folgendes AWS SDKs für Java und. NETBeispiele konfigurieren die Standardverschlüsselungskonfiguration für einen Verzeichnis-Bucket mit SSE - KMS und einem S3-Bucket-Key. Informationen zu anderen SDKs finden Sie im AWS Developer Center unter Beispielcode und Bibliotheken.

> **Important**
>
> Wenn Sie eine AWS KMS key für die serverseitige Verschlüsselung in Amazon S3 verwenden, müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel. Weitere Informationen zu diesen Schlüsseln finden Sie unter Symmetrische KMS Verschlüsselungsschlüssel im AWS Key Management Service Entwicklerhandbuch.

Java

Mit dem können Sie Amazon S3 auffordern AWS SDK for Java 2.x, eine zu verwenden, AWS KMS key indem Sie die `applyServerSideEncryptionByDefault` Methode verwenden, um die Standardverschlüsselungskonfiguration Ihres Verzeichnis-Buckets für die Datenverschlüsselung mit SSE - anzugebenKMS. Sie erstellen einen symmetrischen KMS Verschlüsselungsschlüssel und geben diesen in der Anfrage an.

```
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.PutBucketEncryptionRequest;
import software.amazon.awssdk.services.s3.model.ServerSideEncryption;
import software.amazon.awssdk.services.s3.model.ServerSideEncryptionByDefault;
```

```
import software.amazon.awssdk.services.s3.model.ServerSideEncryptionConfiguration;
import software.amazon.awssdk.services.s3.model.ServerSideEncryptionRule;

public class Main {
    public static void main(String[] args) {
        S3Client s3 = S3Client.create();
        String bucketName = "bucket-base-name--zoneid--x-s3";
        String kmsKeyId = "your-kms-customer-managed-key-id";

        // AWS managed KMS keys aren't supported. Only customer-managed keys are
 supported.
        ServerSideEncryptionByDefault serverSideEncryptionByDefault =
 ServerSideEncryptionByDefault.builder()
                .sseAlgorithm(ServerSideEncryption.AWS_KMS)
                .kmsMasterKeyID(kmsKeyId)
                .build();

        // The bucketKeyEnabled field is enforced to be true.
        ServerSideEncryptionRule rule = ServerSideEncryptionRule.builder()
                .bucketKeyEnabled(true)
                .applyServerSideEncryptionByDefault(serverSideEncryptionByDefault)
                .build();

        ServerSideEncryptionConfiguration serverSideEncryptionConfiguration =
 ServerSideEncryptionConfiguration.builder()
                .rules(rule)
                .build();

        PutBucketEncryptionRequest putRequest = PutBucketEncryptionRequest.builder()
                .bucket(bucketName)

 .serverSideEncryptionConfiguration(serverSideEncryptionConfiguration)
                .build();

        s3.putBucketEncryption(putRequest);

    }
}
```

Weitere Informationen zur Erstellung von kundenverwalteten Schlüsseln finden Sie unter Programmieren von AWS KMS API im AWS Key Management Service Entwicklerhandbuch.

Funktionierende Codebeispiele zum Hochladen eines Objekts finden Sie unter den folgenden Themen. Um diese Beispiele zu verwenden, müssen Sie die Codebeispiele aktualisieren und Verschlüsselungsinformationen wie im vorigen Codefragment gezeigt bereitstellen.

- Weitere Informationen zum Hochladen eines Objekts in einem einzigen Vorgang finden Sie unter Objekte in einen Verzeichnis-Bucket hochladen.
- Informationen zu mehrteiligen API Upload-Vorgängen finden Sie unterMehrteilige Uploads mit Verzeichnis-Buckets verwenden.

.NET

Mit dem können Sie Amazon S3 auffordern AWS SDK for .NET, eine zu verwenden, AWS KMS key indem Sie die `ServerSideEncryptionByDefault` Eigenschaft verwenden, um die Standardverschlüsselungskonfiguration Ihres Verzeichnis-Buckets für die Datenverschlüsselung mit SSE - anzugebenKMS. Sie erstellen einen vom Kunden verwalteten symmetrischen Verschlüsselungsschlüssel und geben diesen in der Anfrage an.

```
    // Set the bucket server side encryption to use AWSKMS with a customer-managed
 key id.
    // bucketName: Name of the directory bucket. "bucket-base-name--zonsid--x-s3"
    // kmsKeyId: The Id of the customer managed KMS Key. "your-kms-customer-managed-
key-id"
    // Returns True if successful.
    public static async Task<bool> SetBucketServerSideEncryption(string bucketName,
 string kmsKeyId)
    {
        var serverSideEncryptionByDefault = new ServerSideEncryptionConfiguration
        {
            ServerSideEncryptionRules = new List<ServerSideEncryptionRule>
            {
                new ServerSideEncryptionRule
                {
                    ServerSideEncryptionByDefault = new
 ServerSideEncryptionByDefault
                    {
                        ServerSideEncryptionAlgorithm =
 ServerSideEncryptionMethod.AWSKMS,
                        ServerSideEncryptionKeyManagementServiceKeyId = kmsKeyId
                    }
                }
```

```
            }
        };
        try
        {
            var encryptionResponse =await _s3Client.PutBucketEncryptionAsync(new
PutBucketEncryptionRequest
            {
                BucketName = bucketName,
                ServerSideEncryptionConfiguration = serverSideEncryptionByDefault,
            });

            return encryptionResponse.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (AmazonS3Exception ex)
        {
            Console.WriteLine(ex.ErrorCode == "AccessDenied"
                ? $"This account does not have permission to set encryption on
{bucketName}, please try again."
                : $"Unable to set bucket encryption for bucket {bucketName},
{ex.Message}");
        }
        return false;
    }
```

Weitere Informationen zur Erstellung von kundenverwalteten Schlüsseln finden Sie unter Programmieren von AWS KMS API im AWS Key Management Service Entwicklerhandbuch.

Funktionierende Codebeispiele zum Hochladen eines Objekts finden Sie unter den folgenden Themen. Um diese Beispiele zu verwenden, müssen Sie die Codebeispiele aktualisieren und Verschlüsselungsinformationen wie im vorigen Codefragment gezeigt bereitstellen.

- Weitere Informationen zum Hochladen eines Objekts in einem einzigen Vorgang finden Sie unter Objekte in einen Verzeichnis-Bucket hochladen.
- Informationen zu mehrteiligen API Upload-Vorgängen finden Sie unterMehrteilige Uploads mit Verzeichnis-Buckets verwenden.

Überwachung der Standardverschlüsselung für Verzeichnis-Buckets mit AWS CloudTrail

Sie können Anfragen zur Standardverschlüsselungskonfiguration für Amazon S3 S3-Verzeichnis-Buckets mithilfe von AWS CloudTrail Ereignissen verfolgen. Die folgenden API Ereignisnamen werden in CloudTrail Protokollen verwendet:

- `PutBucketEncryption`
- `GetBucketEncryption`
- `DeleteBucketEncryption`

> **Note**
>
> - EventBridge wird in Verzeichnis-Buckets nicht unterstützt.
> - Eine zweischichtige serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln (DSSE-KMS) oder serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C) werden in Verzeichnis-Buckets nicht unterstützt.

Weitere Hinweise zur Überwachung der Standardverschlüsselung mit finden Sie unter. AWS CloudTrailÜberwachung der Standardverschlüsselung mit AWS CloudTrail und Amazon EventBridge

Serverseitige Verschlüsselung mit AWS KMS Schlüsseln (SSE-KMS) in Verzeichnis-Buckets verwenden

Die darin enthaltenen Sicherheitskontrollen AWS KMS können Ihnen dabei helfen, die Compliance-Anforderungen im Zusammenhang mit der Verschlüsselung zu erfüllen. Sie können Verzeichnis-Buckets so konfigurieren, dass sie serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-KMS) verwenden und diese KMS Schlüssel verwenden, um Ihre Daten in Amazon S3-Verzeichnis-Buckets zu schützen. Weitere Informationen zu SSE - KMS finden Sie unter. Serverseitige Verschlüsselung mit AWS KMS Schlüsseln verwenden (SSE-) KMS

Berechtigungen

Um ein mit Amazon S3 verschlüsseltes Objekt hoch- oder von Amazon S3 hoch `kms:GenerateDataKey` - oder herunterzuladen, benötigen Sie `kms:Decrypt` Berechtigungen für den Schlüssel. AWS KMS key Weitere Informationen finden Sie im AWS Key Management Service

Entwicklerhandbuch unter Erlauben Sie KMS Schlüsselbenutzern, einen Schlüssel für kryptografische Operationen zu verwenden. Informationen zu den AWS KMS Berechtigungen, die für mehrteilige Uploads erforderlich sind, finden Sie unter. Mehrteiliger Upload API und Berechtigungen

Weitere Hinweise zu KMS Schlüsseln für SSE - finden Sie KMS unter. Geben Sie die serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) an

Themen

- AWS KMS keys
- Verwenden von SSE - KMS für kontenübergreifende Operationen
- Amazon-S3-Bucket-Schlüssel
- Erfordert SSE - KMS
- Verschlüsselungskontext
- Senden von Anfragen für AWS KMS verschlüsselte Objekte
- Auditing SSE — KMS Verschlüsselung in Verzeichnis-Buckets
- Serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) für neue Objekt-Uploads in Verzeichnis-Buckets angeben

AWS KMS keys

Ihre KMS Konfiguration SSE — unterstützt für die gesamte Lebensdauer des Buckets nur einen vom Kunden verwalteten Schlüssel pro Verzeichnis-Bucket. Die Von AWS verwalteter Schlüssel(`aws/s3`) wird nicht unterstützt. Außerdem können Sie, nachdem Sie einen vom Kunden verwalteten Schlüssel für SSE - angegeben habenKMS, den vom Kunden verwalteten Schlüssel für die KMS Konfiguration des SSE Buckets nicht überschreiben.

Sie können den vom Kunden verwalteten Schlüssel, den Sie für die KMS Konfiguration SSE — des Buckets angegeben haben, wie folgt identifizieren:

- Sie stellen eine `HeadObject` API Operationsanfrage, um den Wert von `x-amz-server-side-encryption-aws-kms-key-id` in Ihrer Antwort zu ermitteln.

Um einen neuen, vom Kunden verwalteten Schlüssel für Ihre Daten zu verwenden, empfehlen wir, Ihre vorhandenen Objekte mit einem neuen kundenverwalteten Schlüssel in einen neuen Verzeichnis-Bucket zu kopieren.

Wenn Sie einen vom AWS KMS Kunden verwalteten Schlüssel für die Verschlüsselung in Ihrem Verzeichnis-Bucket angeben, verwenden Sie nur die Schlüssel-ID oder den SchlüsselARN. Das Schlüsselaliasformat des KMS Schlüssels wird nicht unterstützt.

Weitere Hinweise zu KMS Schlüsseln für SSE - KMS finden Sie unterAWS KMS keys.

Verwenden von SSE - KMS für kontenübergreifende Operationen

Wenn Sie Verschlüsselung für kontoübergreifende Operationen in Verzeichnis-Buckets verwenden, sollten Sie Folgendes beachten:

- Wenn Sie kontoübergreifenden Zugriff auf Ihre S3-Objekte gewähren möchten, konfigurieren Sie eine Richtlinie mit einem vom Kunden verwalteten Schlüssel, um den Zugriff von einem anderen Konto aus zu ermöglichen.
- Um einen vom Kunden verwalteten Schlüssel anzugeben, müssen Sie einen vollqualifizierten KMS Schlüssel ARN verwenden.

Amazon-S3-Bucket-Schlüssel

S3-Bucket-Keys sind immer für `GET` `PUT` Operationen in einem Verzeichnis-Bucket aktiviert und können nicht deaktiviert werden. S3-Bucket-Keys werden nicht unterstützt, wenn Sie KMS verschlüsselte Objekte von Allzweck-Buckets in Verzeichnis-Buckets, von Verzeichnis-Buckets in Allzweck-Buckets oder zwischen Verzeichnis-Buckets kopieren SSE CopyObject, UploadPartCopy, der Copy Operation in Batch-Operationen, oder die import Jobs. In diesem Fall ruft Amazon S3 AWS KMS jedes Mal, wenn eine Kopieranforderung für ein KMS -verschlüsseltes Objekt gestellt wird, auf.

Bei APIVorgängen auf zonaler Endpunktebene (Objektebene) mit Ausnahme von und authentifizieren CopyObjectund UploadPartCopyautorisieren Sie Anfragen über CreateSessionfür niedrige Latenz. Wir empfehlen, dass die Standardverschlüsselung des Buckets die gewünschte Verschlüsselungskonfiguration verwendet und dass Sie die Standardverschlüsselung des Buckets in Ihren `CreateSession` Anfragen oder `PUT` Objektanfragen nicht überschreiben. Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Um neue Objekte in einem Verzeichnis-Bucket mit SSE - zu verschlüsselnKMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom Kunden verwalteten Schlüssel) angeben. Wenn dann eine Sitzung für zonale API Endpunktoperationen erstellt wird, werden neue Objekte während der Sitzung automatisch mit SSE - und S3-Bucket-Schlüsseln ver KMS - und entschlüsselt. Weitere Informationen zum Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets finden Sie unter Serverseitige Verschlüsselung mit AWS KMS für Uploads neuer Objekte angeben.

S3-Bucket-Keys werden für einen begrenzten Zeitraum innerhalb von Amazon S3 verwendet, sodass Amazon S3 keine Anfragen mehr stellen muss, AWS KMS um Verschlüsselungsvorgänge abzuschließen. Weitere Informationen zur Verwendung von S3-Bucket Keys finden Sie unter Amazon-S3-Bucket-Schlüssel undSenkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys.

Erfordert SSE - KMS

Um SSE - KMS von allen Objekten in einem bestimmten Verzeichnis-Bucket zu verlangen, können Sie eine Bucket-Richtlinie verwenden. Wenn Sie den `CreateSession` API Vorgang beispielsweise verwenden, um die Erlaubnis zum Hochladen eines neuen Objekts (`PutObject`, und`CreateMultipartUpload`) zu erteilen`CopyObject`, verweigert die folgende Bucket-Richtlinie allen Benutzern die Berechtigung zum Hochladen von Objekten (`s3express:CreateSession`), wenn die `CreateSession` Anfrage keinen `x-amz-server-side-encryption-aws-kms-key-id` Header enthält, der SSE - KMS anfordert.

```
{
   "Version":"2012-10-17",
   "Id":"UploadObjectPolicy",
   "Statement":[{
         "Sid":"DenyObjectsThatAreNotSSEKMS",
         "Effect":"Deny",
         "Principal":"*",
         "Action":"s3express:CreateSession",
         "Resource":"arn:aws:s3express:region:account-id:bucket/bucket-base-name--zone-id--x-s3/*",
         "Condition":{
            "Null":{
               "s3express:x-amz-server-side-encryption-aws-kms-key-id":"true"
            }
         }
      }
   ]
}
```

Um zu verlangen, dass ein bestimmter AWS KMS key Code zum Verschlüsseln der Objekte in einem Bucket verwendet wird, können Sie den `s3express:x-amz-server-side-encryption-aws-kms-key-id` Bedingungsschlüssel verwenden. Um den KMS Schlüssel anzugeben, müssen Sie einen Schlüssel Amazon Resource Name (ARN) im folgenden `arn:aws:kms:region:acct-id:key/key-id` Format verwenden. AWS Identity and Access Management überprüft nicht, ob die

Zeichenfolge für `s3express:x-amz-server-side-encryption-aws-kms-key-id` existiert. Die AWS KMS Schlüssel-ID, die Amazon S3 für die Objektverschlüsselung verwendet, muss mit der AWS KMS Schlüssel-ID in der Richtlinie übereinstimmen, andernfalls lehnt Amazon S3 die Anfrage ab.

Weitere Informationen zur Verwendung SSE — KMS für Uploads neuer Objekte finden Sie unter. Serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) für neue Objekt-Uploads in Verzeichnis-Buckets angeben

Eine vollständige Liste der spezifischen Bedingungsschlüssel für Directory-Buckets finden Sie unter. Autorisieren eines regionalen Endpunkts APIs mit IAM

Verschlüsselungskontext

Bei Verzeichnis-Buckets besteht ein Verschlüsselungskontext aus einer Reihe von Schlüssel-Wert-Paaren, die kontextbezogene Informationen zu den Daten enthalten. Ein zusätzlicher Wert für den Verschlüsselungskontext wird nicht unterstützt. Weitere Informationen zum Thema Verschlüsselungskontext finden Sie unter Verschlüsselungskontext.

Wenn Sie SSE - KMS für einen Verzeichnis-Bucket verwenden, verwendet Amazon S3 standardmäßig den Bucket Amazon Resource Name (ARN) als Verschlüsselungskontext-Paar:

```
arn:aws:s3express:region:account-id:bucket/bucket-base-name--zone-id--x-s3
```

Stellen Sie sicher, dass Ihre IAM Richtlinien oder AWS KMS Schlüsselrichtlinien Ihren Bucket ARN als Verschlüsselungskontext verwenden.

Sie können optional ein explizites Verschlüsselungskontextpaar angeben, indem Sie den `x-amz-server-side-encryption-context` Header in einer zonalen API Endpunktanfrage verwenden, z. B. CreateSession Der Wert dieses Headers ist eine Base64-kodierte Zeichenfolge mit einer Kodierung von UTF -8JSON, die den Verschlüsselungskontext in Form von Schlüssel-Wert-Paaren enthält. Für Verzeichnis-Buckets muss der Verschlüsselungskontext mit dem Standard-Verschlüsselungskontext übereinstimmen — dem Bucket Amazon Resource Name (ARN). Da der Verschlüsselungskontext nicht verschlüsselt ist, sollten Sie außerdem sicherstellen, dass er keine vertraulichen Informationen enthält.

Sie können den Verschlüsselungskontext verwenden, um Ihre kryptografischen Vorgänge zu identifizieren und zu kategorisieren. Sie können auch den ARN Standardwert für den

Verschlüsselungskontext verwenden, um relevante Anfragen nachzuverfolgen, AWS CloudTrail indem Sie sich ansehen, welcher Verzeichnis-Bucket mit welchem Verschlüsselungsschlüssel verwendet ARN wurde.

Wenn Sie im `requestParameters` Feld einer CloudTrail Protokolldatei SSE - KMS für einen Verzeichnis-Bucket verwenden, entspricht der Wert für den Verschlüsselungskontext dem Wert ARN des Buckets.

```
"encryptionContext": {
    "aws:s3express:arn": "arn:aws:s3:::arn:aws:s3express:region:account-
id:bucket/bucket-base-name--zone-id--x-s3"
}
```

Bei der Objektverschlüsselung mit SSE - KMS in einem Verzeichnis-Bucket protokollieren Ihre AWS KMS CloudTrail Ereignisse außerdem Ihren ARN Bucket und nicht Ihr ObjektARN.

Senden von Anfragen für AWS KMS verschlüsselte Objekte

Auf Verzeichnis-Buckets kann nur über HTTPS (TLS) zugegriffen werden. Außerdem signieren Verzeichnis-Buckets Anfragen mithilfe von AWS Signature Version 4 (Sigv4). Weitere Hinweise zum Senden von Anfragen für AWS KMS verschlüsselte Objekte finden Sie unter. Senden von Anfragen für AWS KMS verschlüsselte Objekte

Wenn Ihr Objekt SSE - verwendetKMS, senden Sie keine Header für `GET` Verschlüsselungsanforderungen für Anfragen und `HEAD` Anfragen. Andernfalls erhalten Sie den Fehler HTTP400 Bad Request.

Auditing SSE — KMS Verschlüsselung in Verzeichnis-Buckets

Um die Verwendung Ihrer AWS KMS Schlüssel für Ihre SSE - KMS verschlüsselten Daten zu überprüfen, können Sie AWS CloudTrail Protokolle verwenden. Sie können Einblick in Ihre kryptografischen Operationen erhalten, wie zum Beispiel GenerateDataKey und Decrypt. CloudTrail unterstützt zahlreiche Attributwerte zum Filtern Ihrer Suche, einschließlich Ereignisname, Benutzername und Ereignisquelle.

Themen

- Serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) für neue Objekt-Uploads in Verzeichnis-Buckets angeben

Serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) für neue Objekt-Uploads in Verzeichnis-Buckets angeben

Für Verzeichnis-Buckets können Sie zur Verschlüsselung Ihrer Daten mit serverseitiger Verschlüsselung entweder serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) (Standard) oder serverseitige Verschlüsselung mit () Schlüsseln AWS Key Management Service (AWS KMS-) verwenden. SSE KMS Wir empfehlen, dass die Standardverschlüsselung des Buckets die gewünschte Verschlüsselungskonfiguration verwendet und dass Sie die Standardverschlüsselung des Buckets in Ihren Anfragen oder Objektanforderungen nicht überschreiben. `CreateSession` PUT Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Weitere Informationen zum Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets finden Sie unter Serverseitige Verschlüsselung AWS KMS für Uploads neuer Objekte angeben.

Für alle Amazon-S3-Buckets ist die Verschlüsselung standardmäßig konfiguriert und alle neuen Objekte, die in einen S3-Bucket hochgeladen werden, werden im Ruhezustand automatisch verschlüsselt. Die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Standardverschlüsselungskonfiguration für jeden Bucket in Amazon S3. Wenn Sie einen anderen Verschlüsselungstyp für einen Verzeichnis-Bucket angeben möchten, können Sie die serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-) verwenden. KMS Um neue Objekte in einem Verzeichnis-Bucket mit SSE - zu verschlüsselnKMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom Kunden verwalteten Schlüssel) angeben. Von AWS verwalteter Schlüssel(`aws/s3`) wird nicht unterstützt. Ihre KMS Konfiguration unterstützt für die gesamte Lebensdauer des Buckets nur einen vom Kunden verwalteten Schlüssel pro SSE Verzeichnis-Bucket. Nachdem Sie einen vom Kunden verwalteten Schlüssel für SSE - angegeben habenKMS, können Sie den vom Kunden verwalteten Schlüssel für die KMS Konfiguration des SSE Buckets nicht überschreiben. Wenn Sie dann serverseitige Verschlüsselungseinstellungen für neue Objekte mit SSE - angeben, müssen Sie sicherstellenKMS, dass der Verschlüsselungsschlüssel derselbe vom Kunden verwaltete Schlüssel ist, den Sie für die Standardverschlüsselungskonfiguration des Verzeichnis-Buckets angegeben haben. Um einen neuen, vom Kunden verwalteten Schlüssel für Ihre Daten zu verwenden, empfehlen wir, Ihre vorhandenen Objekte mit einem neuen kundenverwalteten Schlüssel in einen neuen Verzeichnis-Bucket zu kopieren.

Sie können die Verschlüsselung anwenden, wenn Sie entweder ein neues Objekt hochladen oder ein vorhandenes Objekt kopieren. Wenn Sie die Verschlüsselung eines Objekts ändern, wird ein neues Objekt erstellt, um das alte zu ersetzen.

Sie können SSE - angeben, KMS indem Sie die REST API Operationen AWS SDKs, und AWS Command Line Interface (AWS CLI) verwenden.

> **Note**
>
> - Bei Verzeichnis-Buckets ist das Verhalten beim Überschreiben der Verschlüsselung wie folgt:
>   - Wenn Sie CreateSessionmit der REST API zur Authentifizierung und Autorisierung von API Anfragen an zonale Endpunkte außer CopyObjectund verwenden UploadPartCopy, können Sie die Verschlüsselungseinstellungen auf SSE -S3 oder auf SSE — KMS nur dann überschreiben, wenn Sie zuvor die Standardverschlüsselung des Buckets mit angegeben haben. SSE KMS
>   - Wenn Sie die Option CreateSessionmit AWS CLI oder für die Authentifizierung und Autorisierung von zonalen API Endpunktanfragen außer CopyObjectund verwenden UploadPartCopy, können Sie die Verschlüsselungseinstellungen überhaupt nicht überschreiben. AWS SDKs
>   - Wenn Sie CopyObjectAnfragen stellen, können Sie die Verschlüsselungseinstellungen auf SSE -S3 oder auf SSE — KMS außer Kraft setzen, wenn Sie zuvor die Standardverschlüsselung des Buckets mit — angegeben haben. SSE KMS Wenn Sie UploadPartCopyAnfragen stellen, können Sie die Verschlüsselungseinstellungen nicht überschreiben.
> - Sie können Multiregion AWS KMS keys in Amazon S3 verwenden. Amazon S3 behandelt jedoch derzeit Multi-Regions-Schlüssel wie Einzel-Regions-Schlüssel und verwendet nicht die Multi-Regions-Funktionen des Schlüssels. Weitere Informationen finden Sie unter Using multi-Region keys (Verwenden von Multi-Regions-Zugriffpunkt-Schlüsseln) im AWS Key Management Service -Entwicklerhandbuch.
> - Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, benötigen Sie die Erlaubnis, den Schlüssel zu verwenden. Weitere Informationen zu kontoübergreifenden Berechtigungen für KMS Schlüssel finden Sie im AWS Key Management Service Entwicklerhandbuch unter Erstellen von KMS Schlüsseln, die andere Konten verwenden können.

Mit dem REST API

> **Note**
>
> Pro Verzeichnis-Bucket wird für die gesamte Lebensdauer des Buckets nur ein vom Kunden verwalteter Schlüssel unterstützt. Die Von AWS verwalteter Schlüssel(`aws/s3`) wird nicht unterstützt. Nachdem Sie SSE - KMS als Standardverschlüsselungskonfiguration Ihres Buckets mit einem vom Kunden verwalteten Schlüssel angegeben haben, können Sie den vom Kunden verwalteten Schlüssel für die KMS Konfiguration des SSE Buckets nicht mehr ändern.

Bei APIVorgängen auf zonaler Endpunktebene (Objektebene) mit Ausnahme von CopyObjectund UploadPartCopyauthentifizieren und autorisieren Sie Anfragen mit geringer Latenz. CreateSession Wir empfehlen, dass die Standardverschlüsselung des Buckets die gewünschten Verschlüsselungskonfigurationen verwendet und dass Sie die Standardverschlüsselung des Buckets in Ihren `CreateSession` Anfragen oder Objektanforderungen nicht überschreiben. PUT Anschließend werden neue Objekte automatisch mit den gewünschten Verschlüsselungseinstellungen verschlüsselt. Um neue Objekte in einem Verzeichnis-Bucket mit SSE - zu verschlüsselnKMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom Kunden verwalteten Schlüssel) angeben. Wenn dann eine Sitzung für zonale API Endpunktoperationen erstellt wird, werden neue Objekte während der Sitzung automatisch mit SSE - und S3-Bucket-Schlüsseln ver KMS - und entschlüsselt. Weitere Informationen zum Verhalten beim Überschreiben der Verschlüsselung in Verzeichnis-Buckets finden Sie unter Serverseitige Verschlüsselung mit AWS KMS für Uploads neuer Objekte angeben.

Bei API Aufrufen von zonalen Endpunkten (außer CopyObjectund UploadPartCopy) mit dem REST API können Sie die Werte der Verschlüsselungseinstellungen (`x-amz-server-side-encryption`, `x-amz-server-side-encryption-aws-kms-key-idx-amz-server-side-encryption-context`, und`x-amz-server-side-encryption-bucket-key-enabled`) aus der Anfrage nicht überschreiben. `CreateSession` Sie müssen diese Werte für die Verschlüsselungseinstellungen nicht explizit in API Aufrufen von zonalen Endpunkten angeben, und Amazon S3 verwendet die Werte der Verschlüsselungseinstellungen aus der `CreateSession` Anfrage, um neue Objekte im Verzeichnis-Bucket zu schützen.

> **Note**
>
> Wenn Sie das AWS CLI oder das AWS SDKs, für verwenden, wird das Sitzungstoken automatisch aktualisiert`CreateSession`, um Serviceunterbrechungen zu vermeiden, wenn eine Sitzung abläuft. Das AWS CLI oder sie AWS SDKs verwenden die Standardverschlüsselungskonfiguration des Buckets für die `CreateSession` Anfrage. Es wird nicht unterstützt, die Werte der Verschlüsselungseinstellungen in der `CreateSession` Anfrage zu überschreiben. Außerdem wird es bei API Aufrufen von zonalen Endpunkten (außer CopyObjectund UploadPartCopy) nicht unterstützt, die Werte der Verschlüsselungseinstellungen aus der `CreateSession` Anfrage zu überschreiben.

Um neue Objektkopien in einem Verzeichnis-Bucket mit SSE - zu verschlüsselnKMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom Kunden verwalteten Schlüssel) angeben. CopyObject Wenn Sie dann serverseitige Verschlüsselungseinstellungen für neue Objektkopien mit SSE - angeben, müssen Sie sicherstellenKMS, dass der Verschlüsselungsschlüssel derselbe vom Kunden verwaltete Schlüssel ist, den Sie für die Standardverschlüsselungskonfiguration des Directory-Buckets angegeben haben. Um neue Objektteilkopien in einem Verzeichnis-Bucket mit SSE - zu verschlüsselnKMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom Kunden verwalteten Schlüssel) angeben. UploadPartCopy Sie können keine serverseitigen Verschlüsselungseinstellungen für neue Objektteilkopien mit SSE - KMS in den UploadPartCopyAnforderungsheadern angeben. Außerdem müssen die Verschlüsselungseinstellungen, die Sie in der CreateMultipartUploadAnfrage angeben, mit der Standardverschlüsselungskonfiguration des Ziel-Buckets übereinstimmen.

Themen

- Amazon S3 REST API S3-Operationen, die unterstützen SSE - KMS
- Verschlüsselungskontext (x-amz-server-side-encryption-context)
- AWS KMS Schlüssel-ID () x-amz-server-side-encryption-aws-kms-key-id
- S3-Bucket-Schlüssel (x-amz-server-side-encryption-aws-bucket-key-enabled)

Amazon S3 REST API S3-Operationen, die unterstützen SSE - KMS

Die folgenden REST API Operationen auf Objektebene in Verzeichnis-Buckets akzeptieren die Header`x-amz-server-side-encryption`,`x-amz-server-side-encryption-aws-kms-key-id`, und `x-amz-server-side-encryption-context` Request-Header.

- CreateSession— Wenn Sie API Operationen an zonalen Endpunkten (Objektebene) (außer CopyObject und UploadPartCopy) verwenden, können Sie diese Anforderungsheader angeben.
- PutObject— Wenn Sie Daten mithilfe der PUT API Operation hochladen, können Sie diese Anforderungsheader angeben.
- CopyObject – Wenn Sie ein Objekt kopieren, erhalten Sie ein Quell- und ein Zielobjekt. Wenn Sie bei der `CopyObject` Operation SSE KMS -Header übergeben, werden sie nur auf das Zielobjekt angewendet.
- CreateMultipartUpload— Wenn Sie große Objekte mithilfe des mehrteiligen API Upload-Vorgangs hochladen, können Sie diese Header angeben. Sie geben diese Header in der Anfrage an. `CreateMultipartUpload`

Die Antwortheader der folgenden REST API Operationen geben den `x-amz-server-side-encryption` Header zurück, wenn ein Objekt mithilfe serverseitiger Verschlüsselung gespeichert wird.

- CreateSession
- PutObject
- CopyObject
- POST Object
- CreateMultipartUpload
- UploadPart
- UploadPartCopy
- CompleteMultipartUpload
- GetObject
- HeadObject

> **Important**
>
> - Alle `GET` `PUT` Anfragen für ein Objekt, das durch geschützt ist, AWS KMS schlagen fehl, wenn Sie diese Anfragen nicht mithilfe von Transport Layer Security (TLS) oder Signature Version 4 stellen.
> - Wenn Ihr Objekt SSE - verwendetKMS, senden Sie keine Header für `GET` Verschlüsselungsanfragen für Anfragen und `HEAD` Anfragen, da Sie sonst die BadRequest Fehlermeldung HTTP400 erhalten.

Verschlüsselungskontext (**`x-amz-server-side-encryption-context`**)

Wenn Sie angeben`x-amz-server-side-encryption:aws:kms`, API unterstützt Amazon S3 Sie dabei, optional einen expliziten Verschlüsselungskontext mit dem `x-amz-server-side-encryption-context` Header bereitzustellen. Bei Verzeichnis-Buckets besteht ein Verschlüsselungskontext aus einer Reihe von Schlüssel-Wert-Paaren, die kontextbezogene Informationen zu den Daten enthalten. Der Wert muss dem Standard-Verschlüsselungskontext entsprechen — dem Bucket Amazon Resource Name (ARN). Ein zusätzlicher Wert für den Verschlüsselungskontext wird nicht unterstützt.

Hinweise zum Verschlüsselungskontext in Verzeichnis-Buckets finden Sie unterVerschlüsselungskontext. Allgemeine Informationen zum Verschlüsselungs-Kontext finden Sie unter AWS Key Management Service Concepts – Encryption Context (Konzepte – Verschlüsselungs-Kontext) im AWS Key Management Service -Entwicklerhandbuch.

AWS KMS Schlüssel-ID () **`x-amz-server-side-encryption-aws-kms-key-id`**

Sie können den Header `x-amz-server-side-encryption-aws-kms-key-id` verwenden, um die ID des vom Kunden verwalteten Schlüssels anzugeben, der zum Schutz der Daten verwendet wird.

Ihre KMS Konfiguration kann für die gesamte Lebensdauer des Buckets nur einen vom Kunden verwalteten Schlüssel pro SSE Verzeichnis-Bucket unterstützen. Die Von AWS verwalteter Schlüssel(`aws/s3`) wird nicht unterstützt. Außerdem können Sie, nachdem Sie einen vom Kunden verwalteten Schlüssel für SSE - angegeben habenKMS, den vom Kunden verwalteten Schlüssel für die KMS Konfiguration des SSE Buckets nicht überschreiben.

Sie können den vom Kunden verwalteten Schlüssel, den Sie für die KMS Konfiguration SSE — des Buckets angegeben haben, wie folgt identifizieren:

- Sie stellen eine `HeadObject` API Operationsanfrage, um den Wert von `x-amz-server-side-encryption-aws-kms-key-id` in Ihrer Antwort zu ermitteln.

Um einen neuen, vom Kunden verwalteten Schlüssel für Ihre Daten zu verwenden, empfehlen wir, Ihre vorhandenen Objekte mit einem neuen kundenverwalteten Schlüssel in einen neuen Verzeichnis-Bucket zu kopieren.

Hinweise zum Verschlüsselungskontext in Verzeichnis-Buckets finden Sie unterAWS KMS keys.

S3-Bucket-Schlüssel (**`x-amz-server-side-encryption-aws-bucket-key-enabled`**)

S3-Bucket-Keys sind für `GET` `PUT` Vorgänge in einem Verzeichnis-Bucket immer aktiviert und können nicht deaktiviert werden. S3-Bucket-Keys werden nicht unterstützt, wenn Sie KMS verschlüsselte Objekte von Allzweck-Buckets in Verzeichnis-Buckets, von Verzeichnis-Buckets in Allzweck-Buckets oder zwischen Verzeichnis-Buckets kopieren SSE CopyObject, UploadPartCopy, der Copy Operation in Batch-Operationen oder die import Jobs. In diesem Fall ruft Amazon S3 AWS KMS jedes Mal, wenn eine Kopieranforderung für ein KMS -verschlüsseltes Objekt gestellt wird, auf. Informationen zu den S3-Bucket-Keys in Verzeichnis-Buckets finden Sie unter. Verschlüsselungskontext

Verwenden von AWS CLI

> **Note**
>
> Wenn Sie das AWS CLI, für verwenden, wird das Sitzungstoken automatisch aktualisiert`CreateSession`, um Dienstunterbrechungen zu vermeiden, wenn eine Sitzung abläuft. Es wird nicht unterstützt, die Werte der Verschlüsselungseinstellungen für die `CreateSession` Anfrage zu überschreiben. Außerdem wird es bei API Aufrufen von zonalen Endpunkten (außer CopyObjectund UploadPartCopy) nicht unterstützt, die Werte der Verschlüsselungseinstellungen aus der `CreateSession` Anfrage zu überschreiben.
> Um neue Objekte in einem Verzeichnis-Bucket mit SSE - zu verschlüsselnKMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom Kunden verwalteten Schlüssel) angeben. Wenn dann eine Sitzung für zonale API Endpunktoperationen erstellt wird, werden neue Objekte während der Sitzung automatisch mit SSE - und S3-Bucket-Schlüsseln ver KMS - und entschlüsselt.

Um die folgenden AWS CLI Beispielbefehle zu verwenden, ersetzen Sie sie durch Ihre eigenen Informationen. *`user input placeholders`*

Wenn Sie ein neues Objekt hochladen oder ein vorhandenes Objekt kopieren, können Sie angeben, dass serverseitige Verschlüsselung mit AWS KMS Schlüsseln zum Verschlüsseln Ihrer Daten verwendet werden soll. Verwenden Sie dazu den `put-bucket-encryption` Befehl, um die Standardverschlüsselungskonfiguration des Directory-Buckets auf SSE - KMS () `aws:kms` festzulegen. Fügen Sie der Anfrage insbesondere den `--server-side-encryption aws:kms` Header hinzu. Verwenden Sie den`--ssekms-key-id` *`example-key-id`*, um Ihren vom Kunden verwalteten AWS KMS Schlüssel hinzuzufügen, den Sie erstellt haben. Wenn Sie angeben`--server-side-encryption aws:kms`, müssen Sie eine AWS KMS Schlüssel-ID Ihres vom Kunden verwalteten Schlüssels angeben. Directory-Buckets verwenden keinen AWS verwalteten Schlüssel. Ein Beispielbefehl finden Sie unter Mit dem AWS CLI.

Wenn Sie dann ein neues Objekt mit dem folgenden Befehl hochladen, verwendet Amazon S3 die Bucket-Einstellungen für die Standardverschlüsselung, um das Objekt standardmäßig zu verschlüsseln.

```
aws s3api put-object --bucket bucket-base-name--zone-id--x-s3 --key example-object-key
 --body filepath
```

Sie müssen in Ihren Zonal Endpoint API Operations Befehle nicht `-\-bucket-key-enabled` explizit hinzufügen. S3-Bucket-Keys sind immer für `GET PUT` Operationen in einem Verzeichnis-Bucket aktiviert und können nicht deaktiviert werden. S3-Bucket-Keys werden nicht unterstützt, wenn Sie KMS verschlüsselte Objekte von Allzweck-Buckets in Verzeichnis-Buckets, von Verzeichnis-Buckets in Allzweck-Buckets oder zwischen Verzeichnis-Buckets kopieren SSE CopyObject, UploadPartCopy, der Copy Operation in Batch-Operationen oder die import Jobs. In diesem Fall ruft Amazon S3 AWS KMS jedes Mal, wenn eine Kopieranforderung für ein KMS -verschlüsseltes Objekt gestellt wird, auf.

Sie können ein Objekt aus einem Quell-Bucket (z. B. einem Allzweck-Bucket) in einen neuen Bucket (z. B. einen Verzeichnis-Bucket) kopieren und SSE — KMS Verschlüsselung für die Zielobjekte verwenden. Verwenden Sie dazu den `put-bucket-encryption` Befehl, um die Standardverschlüsselungskonfiguration des Ziel-Buckets (z. B. eines Verzeichnis-Buckets) auf SSE - KMS (`aws:kms`) festzulegen. Ein Beispielbefehl finden Sie unter Mit dem AWS CLI. Wenn Sie dann ein Objekt mit dem folgenden Befehl kopieren, verwendet Amazon S3 die Bucket-Einstellungen für die Standardverschlüsselung, um das Objekt standardmäßig zu verschlüsseln.

```
aws s3api copy-object --copy-source amzn-s3-demo-bucket/example-object-key --
bucket bucket-base-name--zone-id--x-s3 --key example-object-key
```

Mit dem AWS SDKs

Bei der Verwendung AWS SDKs können Sie Amazon S3 AWS KMS keys für die serverseitige Verschlüsselung anfordern. Die folgenden Beispiele zeigen die Verwendung von SSE - KMS mit dem AWS SDKs für Java und. NET. Informationen zu anderen SDKs finden Sie im AWS Developer Center unter Beispielcode und Bibliotheken.

> **Note**
>
> Wenn Sie AWS SDKs, für verwenden, wird das Sitzungstoken automatisch aktualisiert`CreateSession`, um Dienstunterbrechungen zu vermeiden, wenn eine Sitzung abläuft. Es wird nicht unterstützt, die Werte der Verschlüsselungseinstellungen für die `CreateSession` Anfrage zu überschreiben. Außerdem wird es bei API Aufrufen von zonalen Endpunkten (außer CopyObjectund UploadPartCopy) nicht unterstützt, die Werte der Verschlüsselungseinstellungen aus der `CreateSession` Anfrage zu überschreiben.
> Um neue Objekte in einem Verzeichnis-Bucket mit SSE - zu verschlüsselnKMS, müssen Sie SSE - KMS als Standardverschlüsselungskonfiguration des Verzeichnis-Buckets mit einem KMS Schlüssel (insbesondere einem vom Kunden verwalteten Schlüssel) angeben. Wenn dann eine Sitzung für zonale API Endpunktoperationen erstellt wird, werden neue Objekte während der Sitzung automatisch mit SSE - und S3-Bucket-Schlüsseln ver KMS - und entschlüsselt.
> Weitere Hinweise dazu, wie Sie AWS SDKs die Standardverschlüsselungskonfiguration eines Verzeichnis-Buckets auf SSE - festlegen könnenKMS, finden Sie unter. Mit dem AWS SDKs

> **Important**
>
> Wenn Sie eine AWS KMS key für die serverseitige Verschlüsselung in Amazon S3 verwenden, müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel. Weitere Informationen zu diesen Schlüsseln finden Sie unter Symmetrische KMS Verschlüsselungsschlüssel im AWS Key Management Service Entwicklerhandbuch.

Weitere Informationen zur Erstellung von kundenverwalteten Schlüsseln finden Sie unter Programmieren von AWS KMS API im AWS Key Management Service Entwicklerhandbuch.

Verschlüsselung während der Übertragung

Directory-Buckets verwenden regionale und zonale Endpunkte. API Je nachdem, welchen Amazon S3 API S3-Betrieb Sie verwenden, ist entweder ein regionaler oder ein zonaler Endpunkt erforderlich. Sie können über einen Gateway-Endpunkt der virtuellen privaten Cloud () VPC auf zonale und regionale Endpunkte zugreifen. Für die Nutzung von Gateway-Endpunkten fallen keine zusätzlichen Gebühren an. Weitere Informationen zu regionalen und zonalen API Endpunkten finden Sie unter. Netzwerke für Verzeichnis-Buckets

Löschen von Daten

Sie können ein oder mehrere Objekte direkt aus Ihren Verzeichnis-Buckets löschen, indem Sie die Amazon S3-Konsole, AWS SDKs, AWS Command Line Interface (AWS CLI) oder Amazon S3 REST API verwenden. Für alle Objekte in Ihrem S3-Bucket entstehen Speicherkosten, deshalb sollten Sie Objekte löschen, die Sie nicht mehr benötigen.

Beim Löschen eines Objekts, das in einem Verzeichnis-Bucket gespeichert ist, werden auch alle übergeordneten Verzeichnisse rekursiv gelöscht, sofern diese übergeordneten Verzeichnisse keine anderen Objekte als das Objekt enthalten, das gelöscht wird.

> **Note**
>
> Multi-Faktor-Authentifizierung (MFA), Löschen und S3-Versionierung werden für S3 Express One Zone nicht unterstützt.

Anfragen authentifizieren und autorisieren

Standardmäßig sind Verzeichnis-Buckets privat und sind nur für Benutzer zugänglich, denen explizit Zugriff gewährt wurde. Die Zugriffskontrollgrenze für Verzeichnis-Buckets wird nur auf Bucket-Ebene festgelegt. Im Gegensatz dazu kann die Zugriffskontrollgrenze für Allzweck-Buckets auf Bucket-, Präfix- oder Objekt-Tag-Ebene festgelegt werden. Dieser Unterschied bedeutet, dass Verzeichnis-Buckets die einzige Ressource sind, die Sie in Bucket-Richtlinien oder IAM Identitätsrichtlinien für den Zugriff auf S3 Express One Zone aufnehmen können.

Amazon S3 Express One Zone unterstützt sowohl die AWS Identity and Access Management (AWS IAM) -Autorisierung als auch die sitzungsbasierte Autorisierung:

- Um regionale API Endpunktoperationen (Operationen auf Bucket-Ebene oder Kontrollebene) mit S3 Express One Zone zu nutzen, verwenden Sie das IAM Autorisierungsmodell, das kein

Sitzungsmanagement beinhaltet. Berechtigungen werden für Aktionen einzeln erteilt. Weitere Informationen finden Sie unter Autorisieren eines regionalen Endpunkts APIs mit IAM.

- Um zonale API Endpunktoperationen (Operationen auf Objektebene oder Datenebene) mit Ausnahme von und zu verwenden, verwenden Sie den `CreateSession` API Vorgang, um Sitzungen zu erstellen `CopyObject` und zu verwaltenHeadBucket, die für die Autorisierung von Datenanforderungen mit niedriger Latenz optimiert sind. Um ein Sitzungstoken abzurufen und zu verwenden, müssen Sie die `s3express:CreateSession`-Aktion für Ihren Verzeichnis-Bucket in einer identitätsbasierten Richtlinie oder einer Bucket-Richtlinie zulassen. Weitere Informationen finden Sie unter Autorisieren eines regionalen Endpunkts APIs mit IAM. Wenn Sie in der Amazon S3-Konsole, über AWS Command Line Interface (AWS CLI) oder mithilfe der auf S3 Express One Zone zugreifen AWS SDKs, erstellt S3 Express One Zone eine Sitzung in Ihrem Namen.

Bei diesem `CreateSession` API Vorgang authentifizieren und autorisieren Sie Anfragen über einen neuen sitzungsbasierten Mechanismus. Sie können mit `CreateSession` temporäre Anmeldeinformationen anfordern, die den Zugriff auf Ihren Bucket mit geringer Latenz ermöglichen. Diese temporären Anmeldeinformationen sind auf einen bestimmten Verzeichnis-Bucket beschränkt.

Um damit zu arbeiten`CreateSession`, empfehlen wir die Verwendung der neuesten Version von AWS SDKs oder die Verwendung von (). AWS Command Line Interface AWS CLI Der Support AWS SDKs AWS CLI kümmert sich in Ihrem Namen um die Einrichtung, Aktualisierung und Beendigung der Sitzung.

Sie verwenden Sitzungstoken nur für zonale Operationen (Objektebene) (außer `CopyObject` und `HeadBucket`), um die mit der Autorisierung verbundene Latenz auf eine Reihe von Anforderungen in einer Sitzung zu verteilen. Für regionale API Endpunktoperationen (Operationen auf Bucket-Ebene) verwenden Sie die IAM Autorisierung, bei der keine Sitzung verwaltet wird. Weitere Informationen erhalten Sie unter Autorisieren eines regionalen Endpunkts APIs mit IAM und Autorisieren von zonalen API Endpunktoperationen mit `CreateSession`.

Wie werden API Operationen authentifiziert und autorisiert

In der folgenden Tabelle sind Authentifizierungs- und Autorisierungsinformationen für API Directory-Bucket-Operationen aufgeführt. Für jeden API Vorgang enthält die Tabelle den Namen des API Vorgangs, die IAM Richtlinienaktion, den Endpunkttyp (regional oder zonal) und den Autorisierungsmechanismus (IAModer sitzungsbasiert). In dieser Tabelle wird auch angegeben, ob der kontoübergreifende Zugriff unterstützt wird. Der Zugriff auf Aktionen auf Bucket-Ebene kann nur in IAM identitätsbasierten Richtlinien (Benutzer oder Rollen) gewährt werden, nicht in Bucket-Richtlinien.

API	Endpunkttyp	IAMAktion	Kontoübergreifender Zugriff
CreateBucket	Regional	s3express:CreateBucket	Nein
DeleteBucket	Regional	s3express:DeleteBucket	Nein
ListDirectoryBuckets	Regional	s3express:ListAllMyDirectoryBuckets	Nein
PutBucketPolicy	Regional	s3express:PutBucketPolicy	Nein
GetBucketPolicy	Regional	s3express:GetBucketPolicy	Nein
DeleteBucketPolicy	Regional	s3express:DeleteBucketPolicy	Nein
CreateSession	Zonal	s3express:CreateSession	Ja
CopyObject	Zonal	s3express:CreateSession	Ja
DeleteObject	Zonal	s3express:CreateSession	Ja
DeleteObjects	Zonal	s3express:CreateSession	Ja
HeadObject	Zonal	s3express:CreateSession	Ja
PutObject	Zonal	s3express:CreateSession	Ja
GetObjectAttributes	Zonal	s3express:CreateSession	Ja
ListObjectsV2	Zonal	s3express:CreateSession	Ja
HeadBucket	Zonal	s3express:CreateSession	Ja
CreateMultipartUpload	Zonal	s3express:CreateSession	Ja

API	Endpunkttyp	IAMAktion	Kontoübergreifender Zugriff
`UploadPart`	Zonal	`s3express:CreateSession`	Ja
`UploadPartCopy`	Zonal	`s3express:CreateSession`	Ja
`CompleteMultipartUpload`	Zonal	`s3express:CreateSession`	Ja
`AbortMultipartUpload`	Zonal	`s3express:CreateSession`	Ja
`ListParts`	Zonal	`s3express:CreateSession`	Ja
`ListMultipartUploads`	Zonal	`s3express:CreateSession`	Ja

Themen

- Autorisieren eines regionalen Endpunkts APIs mit IAM
- Autorisieren von zonalen API Endpunktoperationen mit CreateSession

Autorisieren eines regionalen Endpunkts APIs mit IAM

AWS Identity and Access Management (IAM) hilft Administratoren dabei AWS-Service , den Zugriff auf AWS Ressourcen sicher zu kontrollieren. IAMAdministratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Amazon S3 S3-Ressourcen in S3 Express One Zone zu verwenden. Sie können es ohne IAM zusätzliche Kosten nutzen.

Standardmäßig haben Benutzer keine Berechtigungen für Verzeichnis-Buckets und S3-Express-One-Zone-Vorgänge. Um Zugriffsberechtigungen für Verzeichnis-Buckets IAM zu gewähren, können Sie Benutzer, Gruppen oder Rollen erstellen und diesen Identitäten Berechtigungen zuweisen. Weitere Informationen IAM dazu finden Sie unter Bewährte Sicherheitsmethoden IAM im IAM Benutzerhandbuch.

Um Zugriff zu gewähren, können Sie Ihren Benutzern, Gruppen oder Rollen auf die folgenden Weisen Berechtigungen hinzufügen:

- Benutzer und Gruppen in AWS IAM Identity Center — Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter Erstellen eines Berechtigungssatzes im AWS IAM Identity Center -Benutzerhandbuch.
- Benutzer, IAM die über einen Identitätsanbieter verwaltet werden — Erstellen Sie eine Rolle für den Identitätsverbund. Folgen Sie den Anweisungen unter Erstellen einer Rolle für einen externen Identitätsanbieter (Federation) im IAMBenutzerhandbuch.
- IAMRollen und Benutzer — Erstellen Sie eine Rolle, die Ihr Benutzer übernehmen kann. Folgen Sie den Anweisungen unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen IAM Benutzer im IAMBenutzerhandbuch.

IAMWeitere Informationen zu S3 Express One Zone finden Sie in den folgenden Themen.

Themen

- Auftraggeber
- Ressourcen
- Aktionen für Verzeichnis-Buckets
- Bedingungsschlüssel für Verzeichnis-Buckets
- IAMidentitätsbasierte Richtlinien für Verzeichnis-Buckets
- Beispiel für Bucket-Richtlinien für Verzeichnis-Buckets

Auftraggeber

Wenn Sie eine ressourcenbasierte Richtlinie erstellen, um Zugriff auf Ihre Buckets zu gewähren, müssen Sie das `Principal`-Element verwenden, um die Person oder Anwendung anzugeben, die eine Anforderung für eine Aktion oder einen Vorgang auf dieser Ressource stellen kann. Für Verzeichnis-Bucket-Richtlinien können Sie die folgenden Prinzipale verwenden:

- Ein AWS Konto
- Ein IAM Nutzer
- Eine IAM Rolle
- Ein Verbundbenutzer

Weitere Informationen finden Sie unter Principal im IAM-Benutzerhandbuch.

Ressourcen

Amazon-Ressourcennamen (ARNs) für Verzeichnis-Buckets enthalten den `s3express` Namespace, die AWS-Region, die AWS Konto-ID und den Verzeichnis-Bucket-Namen, der die Availability Zone-ID enthält. Um auf Ihren Directory-Bucket zuzugreifen und Aktionen in Ihrem Verzeichnis-Bucket auszuführen, müssen Sie das folgende ARN Format verwenden:

```
arn:aws:s3express:region:account-id:bucket/base-bucket-name--zone-id--x-s3
```

Weitere Informationen zu finden ARNs Sie unter Amazon Resource Names (ARNs) im IAM-Benutzerhandbuch. Weitere Informationen zu Ressourcen finden Sie unter IAM JSON Richtlinienelemente: Resource im IAM-Benutzerhandbuch.

Aktionen für Verzeichnis-Buckets

In einer IAM identitäts- oder ressourcenbasierten Richtlinie definieren Sie, welche S3-Aktionen zulässig oder verweigert werden. Aktionen entsprechen bestimmten Vorgängen. API Wenn Sie Verzeichnis-Buckets verwenden, können Sie den S3 Express One Zone-Namespace verwenden, um Berechtigungen zu erteilen. Dieser Namespace ist `s3express`.

Wenn Sie die `s3express:CreateSession` Erlaubnis erteilen, ermöglicht dies dem `CreateSession` API Vorgang, Sitzungstoken abzurufen, wenn auf Operationen auf zonale Endpunkte API (oder Objektebene) zugegriffen wird. Diese Sitzungstoken geben Anmeldeinformationen zurück, die verwendet werden, um Zugriff auf alle anderen zonalen API Endpunktoperationen zu gewähren. Daher müssen Sie keine Zugriffsberechtigungen für zonale API Operationen mithilfe IAM von Richtlinien gewähren. Stattdessen ermöglicht das Sitzungstoken den Zugriff. Eine Liste der API Vorgänge und Berechtigungen für zonale Endgeräte finden Sie unter Anfragen authentifizieren und autorisieren.

Weitere Informationen zu zonalen und regionalen API Endpunktoperationen finden Sie unter. Netzwerke für Verzeichnis-Buckets Weitere Informationen zu diesem `CreateSession` API Vorgang finden Sie unter CreateSessionin der Amazon Simple Storage Service API Reference.

Sie können die folgenden Aktionen im `Action` Element einer IAM Grundsatzerklärung angeben. Verwenden Sie Richtlinien, um Berechtigungen zum Ausführen einer Operation in AWS zu erteilen. Wenn Sie eine Aktion in einer Richtlinie verwenden, gewähren oder verweigern Sie in der Regel den Zugriff auf den API Vorgang mit demselben Namen. In einigen Fällen steuert jedoch eine einzelne

Aktion den Zugriff auf mehr als einen API Vorgang. Der Zugriff auf Aktionen auf Bucket-Ebene kann nur über IAM identitätsbasierte Richtlinien (Benutzer oder Rolle) und nicht über Bucket-Richtlinien gewährt werden.

Die folgende Tabelle zeigt Aktionen und Bedingungsschlüssel.

Aktion	API	Beschreibung	Zugriffsebene	Bedingungsschlüssel

`s3express:CreateBucket`	`CreateBucket`	Gewährt die Berechtigung zum Erstellen eines neuen Buckets.	Schreibe	`s3express:authType`
				`s3express:LocationName`
				`s3express:ResourceAccount`
				`s3express:signatureversion`
				`s3express:TlsVersion`
				`s3express:x-amz-content-sha256`

`s3express:CreateSession`	Operationen an zonalen Endpunkten API	Erteilt die Berechtigung zum Erstellen eines Sitzungstokens, das verwendet wird, um Zugriff auf alle zonalen API Operationen (auf Objektebene) zu gewähren, z. B.`CopyObject` , `PutObject`	Schreibe	`s3express:authType` `s3express:SessionMode`

Aktion	API	Beschreibung	Zugriffsebene	Bedingungsschlüssel

		`GetObject` , usw. `HeadBucket`		`s3express:ResourceAccount`
				
				`s3express:signatureversion`
				
				`s3express:signatureAge`
				
				`s3express:TlsVersion`
				
				`s3express:x-amz-content-sha256`
				
				`s3express:x-amz-server-side-encryption`
				
				`s3express:x-amz-server-side-encryption-aws-kms-key-id`

Aktion	API	Beschreibung	Zugriffsebene	Bedingungsschlüssel
s3express:DeleteBucket	DeleteBucket	Erteilt die Berechtigung zum Löschen des Buckets mit dem Namen. URI	Schreibe	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Aktion	API	Beschreibung	Zugriffsebene	Bedingungsschlüssel
s3express:DeleteBucketPolicy	DeleteBucketPolicy	Gewährt die Berechtigung zum Löschen der Richtlinie für einen angegebenen Bucket.	Berechtigungsverwaltung	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Aktion	API	Beschreibung	Zugriffsebene	Bedingungsschlüssel

`s3express:GetBucketPolicy`	`GetBucketPolicy`	Gewährt die Berechtigung zum Zurückgeben der Richtlinie des angegebenen Buckets.	Lesen	`s3express:authType`
				`s3express:ResourceAccount`
				`s3express:signatureversion`
				`s3express:TlsVersion`
				`s3express:x-amz-content-sha256`

Aktion	API	Beschreibung	Zugriffsebene	Bedingungsschlüssel
s3express:GetEncryptionConfiguration	GetBucketEncryption	Erteilt die Berechtigung, die Standardverschlüsselungskonfiguration eines Verzeichnis-Buckets zurückzugeben.	Lesen	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Aktion	API	Beschreibung	Zugriffsebene	Bedingungsschlüssel
s3express:ListAllMyDirectoryBuckets	ListDirectoryBuckets	Gewährt die Berechtigung zum Auflisten aller Verzeichnis-Buckets, die dem authentifizierten Sender der Anforderung gehören.	Auflisten	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Aktion	API	Beschreibung	Zugriffsebene	Bedingungsschlüssel
s3express:PutBucketPolicy	PutBucketPolicy	Gewährt die Berechtigung zum Hinzufügen oder Ersetzen einer Bucket-Richtlinie für einen Bucket.	Berechtigungsverwaltung	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Aktion	API	Beschreibung	Zugriffsebene	Bedingungsschlüssel
s3express:PutBucketPolicy	PutBucketPolicy	Gewährt die Berechtigung zum Hinzufügen oder Ersetzen einer Bucket-Richtlinie für einen Bucket.	Berechtigungsverwaltung	s3express:authType s3express:ResourceAccount s3express:signatureversion s3express:TlsVersion s3express:x-amz-content-sha256

Aktion	API	Beschreibung	Zugriffsebene	Bedingungsschlüssel

`s3express:PutEncryptionConfiguration`	`PutBucketEncryption` oder `DeleteBucketEncryption`	Erteilt die Berechtigung, die Verschlüsselungskonfiguration für einen Verzeichnis-Bucket festzulegen	Schreibe	`s3express:authType`
				
				`s3express:ResourceAccount`
				
				`s3express:signatureversion`
				
				`s3express:TlsVersion`
				
				`s3express:x-amz-content-sha256`

Bedingungsschlüssel für Verzeichnis-Buckets

Im Folgenden finden Sie Bedingungsschlüssel, die im `Condition` Element einer IAM Richtlinie verwendet werden können. Diese Schlüssel können Sie verwenden, um die Bedingungen zu verfeinern, unter denen die Richtlinienanweisung angewendet wird.

Bedingungsschlüssel	Beschreibung	Typ
`s3express:authType`	Filtert den Zugriff nach Authentifizierungsmethode Um eingehende Anfragen auf die Verwendung einer bestimmten Authentifizierungsmethode zu beschränken, können Sie diesen optionalen Bedingungsschlüssel verwenden. Sie können diesen Bedingungsschlüssel beispielsweise verwenden, um	String

Bedingungsschlüssel	Beschreibung	Typ
	zuzulassen, dass nur der HTTP `Authorization` Header bei der Anforderungsauthentifizierung verwendet wird. Zulässige Werte: `REST-HEADER` , `REST-QUERY-STRING`	
`s3express:LocationName`	Filtert den Zugriff auf den `CreateBucket` API Vorgang nach einer bestimmten Availability Zone ID (AZ-ID), `usw2-az1` z. B. Beispielwert: `usw2-az1`	String
`s3express:ResourceAccount`	Filtert den Zugriff nach der AWS-Konto ID des Ressourcenbesitzers. Um den Benutzer-, Rollen- oder Anwendungszugriff auf die Verzeichnis-Buckets einzuschränken, die einer bestimmten AWS-Konto ID gehören, können Sie entweder den Bedingungsschlüssel `aws:ResourceAccount` oder den `s3express:ResourceAccount` Bedingungsschlüssel verwenden. Sie können diesen Bedingungsschlüssel entweder in AWS Identity and Access Management (IAM) Identitätsrichtlinien oder in Virtual Private Cloud (VPC) -Endpunktrichtlinien verwenden. Sie können diesen Bedingungsschlüssel beispielsweise verwenden, um zu verhindern, dass Clients in Ihrem VPC Unternehmen auf Buckets zugreifen, die Ihnen nicht gehören. Beispielwert: `111122223333`	String

Bedingungsschlüssel	Beschreibung	Typ
`s3express:SessionMode`	Filtert den Zugriff nach der vom `CreateSession` API Vorgang angeforderten Berechtigung. Standardmäßig ist die Sitzung auf `ReadWrite` eingestellt. Sie können diesen Bedingungsschlüssel verwenden, um den Zugriff auf `ReadOnly` zu beschränken oder den `ReadWrite` -Zugriff explizit zu verweigern. Weitere Informationen finden Sie unter Beispiel für Bucket-Richtlinien für Verzeichnis-Buckets und CreateSessionin der Amazon Simple Storage Service API Reference. Zulässige Werte: `ReadWrite` , `ReadOnly`	String
`s3express:signatureAge`	Filtert den Zugriff nach dem Alter der Anforderungssignatur in Millisekunden. Diese Bedingung gilt nur für vorsignierte Personen URLs. In AWS Signature Version 4 ist der Signaturschlüssel bis zu sieben Tage gültig. Daher sind die Signaturen auch bis zu sieben Tage lang gültig. Weitere Informationen finden Sie unter Einführung in das Signieren von Anfragen in der Amazon Simple Storage API Service-Referenz. Sie können diese Bedingung verwenden, um das Alter der Unterschrift weiter einzuschränken. Beispielwert: `600000`	Numerischer Wert
`s3express:signatureversion`	Identifiziert die Version von AWS Signature, die Sie für authentifizierte Anfragen unterstützen möchten. Für authentifizierte Anfragen wird Signature Version 4 unterstützt. Gültiger Wert: `"AWS4-HMAC-SHA256"` (identifiziert Signature Version 4)	String

Bedingungsschlüssel	Beschreibung	Typ
`s3express:TlsVersion`	Filtert den Zugriff nach der TLS Version, die vom Client verwendet wird. Sie können den `s3:TlsVersion` Bedingungsschlüssel verwendenIAM, um Richtlinien für virtuelle private Cloud-Endpunkte (VPCE) oder Buckets zu schreiben, die den Benutzer- oder Anwendungszugriff auf Verzeichnis-Buckets auf der Grundlage der vom Client verwendeten TLS Version einschränken. Sie können diesen Bedingungsschlüssel auch verwenden, um Richtlinien zu schreiben, für die eine TLS Mindestversion erforderlich ist. Beispielwert: `1.3`	Numerischer Wert

Bedingungsschlüssel	Beschreibung	Typ
`s3express:x-amz-content-sha256`	Filtert den Zugriff auf nicht signierte Inhalte in Ihrem Bucket. Sie können diesen Bedingungsschlüssel verwenden, um nicht signierte Inhalte in Ihrem Bucket zu verbieten. Wenn Sie Signature Version 4 verwenden, fügen Sie bei Anfragen, die den `Authorization` -Header verwenden, den `x-amz-content-sha256` -Header in die Signaturberechnung ein und setzen dann dessen Wert auf die Hash-Nutzlast. Sie können diesen Bedingungsschlüssel in Ihrer Bucket-Richtlinie verwenden, um alle Uploads zu verweigern, deren Nutzlasten nicht signiert sind. Beispielsweise: • Verweigern Sie Uploads, die den `Authorization` -Header zur Authentifizierung von Anfragen verwenden, aber die Nutzdaten nicht signieren. Weitere Informationen finden Sie unter Payload in einem einzigen Chunk übertragen in der Amazon Simple Storage Service API Reference. • Verweigern Sie Uploads, die vorsigniert verwenden. URLs Vorsignierte haben URLs immer eine. `UNSIGNED_PAYLOAD` Weitere Informationen finden Sie unter Authentifizieren von Anfragen und Authentifizierungsmethoden in der Amazon Simple Storage API Service-Referenz. Günstiger Wert: `UNSIGNED-PAYLOAD`	String

Bedingungsschlüssel	Beschreibung	Typ
`s3express:x-amz-server-side-encryption`	Filtert den Zugriff nach serverseitiger Verschlüsselung Zulässige Werte: `"AWS256"`, `aws:kms`	String
`s3express:x-amz-server-side-encryption-aws-kms-key-id`	Filtert den Zugriff nach dem vom AWS KMS Kunden verwalteten Schlüssel für die serversei tige Verschlüsselung Beispielwert: `"arn:aws:kms: region:acct-id:key/key-id"`	ARN

IAMidentitätsbasierte Richtlinien für Verzeichnis-Buckets

Bevor Sie Verzeichnis-Buckets erstellen können, müssen Sie Ihrer AWS Identity and Access Management (IAM) -Rolle oder Ihren Benutzern die erforderlichen Berechtigungen erteilen. Diese Beispielrichtlinie ermöglicht den Zugriff auf den `CreateSession` API Vorgang (zur Verwendung mit Operationen auf Zonenendpunkt [Objektebene]) und alle API Operationen auf regionalen Endpunkten (Bucket-Ebene). API Diese Richtlinie ermöglicht die Verwendung des `CreateSession` API Vorgangs mit allen Verzeichnis-Buckets, aber die regionalen API Endpunktoperationen sind nur für die Verwendung mit dem angegebenen Verzeichnis-Bucket zulässig. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre eigenen Informationen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAccessRegionalEndpointAPIs",
            "Effect": "Allow",
            "Action": [
                "s3express:DeleteBucket",
                "s3express:DeleteBucketPolicy",
                "s3express:CreateBucket",
                "s3express:PutBucketPolicy",
                "s3express:GetBucketPolicy",
                "s3express:ListAllMyDirectoryBuckets"
```

```
            ],

            "Resource": "arn:aws:s3express:region:account_id:bucket/bucket-base-
name--zone-id--x-s3/*"
        },
        {
            "Sid": "AllowCreateSession",
            "Effect": "Allow",
            "Action": "s3express:CreateSession",
            "Resource": "*"
        }
    ]
}
```

Beispiel für Bucket-Richtlinien für Verzeichnis-Buckets

Dieser Abschnitt enthält Beispiele für Richtlinien für Verzeichnis-Buckets. Wenn Sie diese Richtlinien verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre Informationen.

Die folgende Beispiel-Bucket-Richtlinie ermöglicht es AWS-Konto ID*111122223333*, den `CreateSession` API Vorgang mit der `ReadWrite` Standardsitzung für den angegebenen Verzeichnis-Bucket zu verwenden. Diese Richtlinie gewährt Zugriff auf die API Operationen am zonalen Endpunkt (Objektebene).

Example – Bucket-Richtlinie, um **`CreateSession`**-Aufrufe mit der **`ReadWrite`**-Standardsitzung zuzulassen

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadWriteAccess",
            "Effect": "Allow",
            "Resource": "arn:aws:s3express:us-west-2:account-id:bucket/bucket-base-
name--zone-id--x-s3",
            "Principal": {
                "AWS": [
                    "111122223333"
                ]
            },
            "Action": [
```

```
                "s3express:CreateSession"
            ]
        }
    ]
}
```

Example – Bucket-Richtlinie, um **`CreateSession`**-Aufrufe mit einer **`ReadOnly`**-Sitzung zuzulassen

Die folgende Beispiel-Bucket-Richtlinie ermöglicht es AWS-Konto ID*111122223333*, den `CreateSession` API Vorgang zu verwenden. Diese Richtlinie verwendet den `s3express:SessionMode`-Bedingungsschlüssel mit dem `ReadOnly`-Wert, um eine schreibgeschützte Sitzung einzurichten.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ReadOnlyAccess",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111122223333"
            },
            "Action": "s3express:CreateSession",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3express:SessionMode": "ReadOnly"
                }
            }
        }
    ]
}
```

Example – Bucket-Richtlinie, um den kontoübergreifenden Zugriff für **`CreateSession`**-Aufrufe zuzulassen

Die folgende Beispiel-Bucket-Richtlinie ermöglicht es AWS-Konto ID*111122223333*, den `CreateSession` API Vorgang für den angegebenen Verzeichnis-Bucket zu verwenden, der AWS-Konto ID gehört *444455556666*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CrossAccount",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111122223333"
            },
            "Action": [
                "s3express:CreateSession"
            ],
            "Resource": "arn:aws:s3express:us-west-2:444455556666:bucket/bucket-base-name--zone-id--x-s3"

        }
    ]
}
```

Autorisieren von zonalen API Endpunktoperationen mit `CreateSession`

Um zonale API Endpunktoperationen (Operationen auf Objektebene oder Datenebene) mit Ausnahme von und zu verwenden, verwenden Sie den `CreateSession` API Vorgang`HeadBucket`, um Sitzungen zu erstellen `CopyObject` und zu verwalten, die für die Autorisierung von Datenanforderungen mit geringer Latenz optimiert sind. Um ein Sitzungstoken abzurufen und zu verwenden, müssen Sie die `s3express:CreateSession`-Aktion für Ihren Verzeichnis-Bucket in einer identitätsbasierten Richtlinie oder einer Bucket-Richtlinie zulassen. Weitere Informationen finden Sie unter Autorisieren eines regionalen Endpunkts APIs mit IAM. Wenn Sie in der Amazon S3-Konsole, über AWS Command Line Interface (AWS CLI) oder mithilfe der auf S3 Express One Zone zugreifen AWS SDKs, erstellt S3 Express One Zone eine Sitzung in Ihrem Namen.

Wenn Sie Amazon S3 verwenden RESTAPI, können Sie den `CreateSession` API Vorgang dann verwenden, um temporäre Sicherheitsanmeldeinformationen zu erhalten, die eine Zugriffsschlüssel-ID, einen geheimen Zugriffsschlüssel, ein Sitzungstoken und eine Ablaufzeit enthalten. Die temporären Anmeldeinformationen bieten dieselben Berechtigungen wie langfristige Sicherheitsanmeldedaten, wie z. B. IAM Benutzeranmeldedaten, aber temporäre Sicherheitsanmeldedaten müssen ein Sitzungstoken enthalten.

Sitzungsmodus

Der Sitzungsmodus definiert den Umfang der Sitzung. In Ihrer Bucket-Richtlinie können Sie den `s3express:SessionMode`-Bedingungsschlüssel angeben, um zu steuern, wer eine `ReadWrite`- oder `ReadOnly`-Sitzung erstellen kann. Weitere Informationen zu `ReadWrite` `ReadOnly` Or-Sitzungen finden Sie im `x-amz-create-session-mode` Parameter für CreateSessionin der Amazon S3 API S3-Referenz. Informationen dazu, wie Sie eine Bucket-Richtlinie erstellen, finden Sie unter Beispiel für Bucket-Richtlinien für Verzeichnis-Buckets.

Sitzungs-Token

Wenn Sie einen Aufruf mit temporären Sicherheitsanmeldeinformationen tätigen, muss dieser ein Sitzungs-Token enthalten. Das Sitzungstoken wird zusammen mit den temporären Anmeldeinformationen zurückgegeben. Ein Sitzungstoken ist auf Ihren Verzeichnis-Bucket beschränkt und wird verwendet, um zu überprüfen, ob die Sicherheitsanmeldedaten gültig und nicht abgelaufen sind. Um Ihre Sitzungen zu schützen, laufen temporäre Sicherheitsanmeldedaten nach 5 Minuten ab.

`CopyObject` und **`HeadBucket`**

Temporäre Sicherheitsanmeldedaten sind auf einen bestimmten Verzeichnis-Bucket beschränkt und werden automatisch für alle zonalen API Operationsaufrufe (auf Objektebene) für einen bestimmten Verzeichnis-Bucket aktiviert. Im Gegensatz zu anderen zonalen API Endpunktoperationen verwenden sie keine AuthentifizierungCopyObject. `HeadBucket` `CreateSession` Alle `CopyObject` `HeadBucket` Anfragen müssen authentifiziert und mithilfe IAM von Anmeldeinformationen signiert werden. Sie `HeadBucket` werden jedoch `CopyObject` wie andere zonale API Endpunktoperationen weiterhin von `s3express:CreateSession` autorisiert.

Weitere Informationen finden Sie unter CreateSessionin der Amazon Simple Storage Service API Reference.

Bewährte Sicherheitsmethoden für Directory-Buckets

Bei der Arbeit mit Verzeichnis-Buckets sind eine Reihe von Sicherheitsfunktionen zu berücksichtigen. Die folgenden bewährten Methoden stellen allgemeine Richtlinien und keine vollständige Sicherheitslösung dar. Da diese bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Empfehlungen und nicht als bindend ansehen.

Standardeinstellungen für „Öffentlichen Zugriff blockieren“ und „Objekteigentümerschaft“

Verzeichnis-Buckets unterstützen S3 Block Public Access und S3 Object Ownership. Diese S3-Features werden für die Überwachung und Verwaltung des Zugriffs auf Ihre Buckets und Objekte verwendet.

Alle Einstellungen zum Blockieren jeglichen öffentlichen Zugriffs sind für Verzeichnis-Buckets standardmäßig aktiviert. Außerdem ist für Object Ownership der Wert Bucket Owner erforced festgelegt, was bedeutet, dass die Zugriffskontrolllisten (ACLs) deaktiviert sind. Diese Einstellungen können nicht geändert werden. Weitere Informationen zu diesen Features finden Sie unter the section called “Blockieren des öffentlichen Zugriffs” und the section called “Steuern der Objekteigentümerschaft”.

> **Note**
>
> Sie können keinen Zugriff auf Objekte gewähren, die in Verzeichnis-Buckets gespeichert sind. Sie können nur Zugriff auf Ihre Verzeichnis-Buckets gewähren. Das Autorisierungsmodell für S3 Express One Zone unterscheidet sich vom Autorisierungsmodell für Amazon S3. Weitere Informationen finden Sie unter Autorisieren von zonalen API Endpunktoperationen mit `CreateSession`.

Authentifizierung und Autorisierung

Die Authentifizierungs- und Autorisierungsmechanismen für Directory-Buckets unterscheiden sich, je nachdem, ob Sie Anfragen an zonale API Endpunktoperationen oder regionale Endpunktoperationen stellen. API Zonale API Operationen sind Operationen auf Objektebene (Datenebene). Bei regionalen API Operationen handelt es sich um Operationen auf Bucket-Ebene (Steuerungsebene).

Sie authentifizieren und autorisieren Anfragen an zonale API Endpunktoperationen mithilfe eines neuen sitzungsbasierten Mechanismus, der für die geringste Latenz optimiert wurde. Bei der sitzungsbasierten Authentifizierung AWS SDKs verwenden sie den `CreateSession` API Vorgang, um temporäre Anmeldeinformationen anzufordern, die den Zugriff auf Ihren Verzeichnis-Bucket mit geringer Latenz ermöglichen. Diese temporären Anmeldeinformationen sind auf einen bestimmten Verzeichnis-Bucket beschränkt und laufen nach 5 Minuten ab. Sie können diese temporären Anmeldeinformationen verwenden, um zonale Aufrufe (Objektebene) zu signieren.

API Weitere Informationen finden Sie unter Autorisieren von zonalen API Endpunktoperationen mit `CreateSession`.

Signieren von Anfragen mit Anmeldeinformationen für die Verwaltung von Verzeichnis-Buckets

Sie verwenden Ihre Anmeldeinformationen, um API Anfragen an zonale Endpunkte (Objektebene) mit AWS Signature Version 4 `s3express` als Dienstnamen zu signieren. Wenn Sie Ihre Anforderungen signieren, verwenden Sie den geheimen Schlüssel, der von `CreateSession`zurückgegeben wurde, und stellen auch das Sitzungstoken mit dem `x-amzn-s3session-token header` bereit. Weitere Informationen finden Sie unter CreateSession.

Die unterstützten AWS SDKs verwalten Anmeldeinformationen und signieren in Ihrem Namen. Wir empfehlen, die AWS SDKs zu verwenden, um Anmeldeinformationen zu aktualisieren und Anfragen für Sie zu signieren.

Anfragen mit IAM Anmeldeinformationen signieren

Alle regionalen API Aufrufe (auf Bucket-Ebene) müssen authentifiziert und mit AWS Identity and Access Management (IAM) -Anmeldeinformationen statt mit temporären Sitzungsanmeldedaten signiert werden. IAMAnmeldeinformationen bestehen aus der Zugriffsschlüssel-ID und dem geheimen Zugriffsschlüssel für die Identitäten. IAM Alle `CopyObject` `HeadBucket` Anfragen müssen außerdem authentifiziert und mithilfe IAM von Anmeldeinformationen signiert werden.

Um die geringste Latenz für Ihre zonalen Operationsaufrufe (auf Objektebene) zu erreichen, empfehlen wir, beim Signieren Ihrer Anfragen Anmeldeinformationen `CreateSession` zu verwenden, mit Ausnahme von Anfragen an und. `CopyObject` `HeadBucket`

Verwenden AWS CloudTrail

AWS CloudTrail bietet eine Aufzeichnung der Aktionen, die von einem Benutzer, einer Rolle oder einem AWS-Service in Amazon S3 ausgeführt wurden. Sie können die von gesammelten Informationen verwenden CloudTrail , um Folgendes zu ermitteln:

- Die Anforderung, die an Amazon S3 gestellt wurde
- Die IP-Adresse, von der die Anforderung erfolgt ist
- Wer die Anforderung gestellt hat
- Wann die Anforderung gestellt wurde
- Zusätzliche Details zur Anforderung

Wenn Sie Ihre einrichten AWS-Konto, sind CloudTrail Verwaltungsereignisse standardmäßig aktiviert. Die folgenden regionalen API Endpunktoperationen (Operationen auf Bucket-Ebene oder Kontrollebene) werden protokolliert. API CloudTrail

- CreateBucket
- DeleteBucket
- DeleteBucketPolicy
- PutBucketPolicy
- GetBucketPolicy
- ListDirectoryBuckets
- ListMultipartUploads
- PutBucketEncryption
- GetBucketEncryption
- DeleteBucketEncryption

> **Note**
>
> `ListMultipartUploads`ist ein Vorgang mit zonalem Endpunkt. API Es wird jedoch CloudTrail als Verwaltungsereignis protokolliert. Weitere Informationen finden Sie unter ListMultipartUploadsin der Amazon Simple Storage Service API Reference.

Standardmäßig protokollieren CloudTrail Trails keine Datenereignisse, aber Sie können Trails so konfigurieren, dass Datenereignisse für von Ihnen angegebene Verzeichnis-Buckets oder Datenereignisse für alle Directory-Buckets in Ihrem AWS Konto protokolliert werden. Die folgenden zonalen API Endpunktoperationen (Operationen auf Objektebene oder Datenebene) werden protokolliertAPI. CloudTrail

- AbortMultipartUpload
- CompleteMultipartUpload
- CreateSession
- CopyObject
- CreateMultipartUpload
- DeleteObject

- DeleteObjects
- GetObject
- GetObjectAttributes
- HeadBucket
- HeadObject
- ListObjectsV2
- ListParts
- PutObject
- UploadPart
- UploadPartCopy

Weitere Informationen zur Verwendung AWS CloudTrail mit Directory-Buckets finden Sie unter Logging with AWS CloudTrail for Directory-Buckets.

Implementieren Sie die Überwachung mithilfe AWS von Überwachungstools

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Sicherheit, Verfügbarkeit und Leistung von Amazon S3 und Ihren AWS Lösungen. AWS bietet verschiedene Tools und Dienste, mit denen Sie Amazon S3 und Ihre anderen überwachen können AWS-Services. Sie können beispielsweise CloudWatch Amazon-Metriken für Amazon S3 überwachen, insbesondere die `BucketSizeBytes` und `NumberOfObjects` Speichermetriken.

Objekte, die in den Verzeichnis-Buckets gespeichert sind, werden nicht in den `BucketSizeBytes` `NumberOfObjects` Speichermetriken für Amazon S3 berücksichtigt. Die `NumberOfObjects` Speichermetriken `BucketSizeBytes` und werden jedoch für Verzeichnis-Buckets unterstützt.
Um die Metriken Ihrer Wahl zu sehen, können Sie zwischen den Amazon S3 S3-Speicherklassen unterscheiden, indem Sie eine `StorageType` Dimension angeben. Weitere Informationen finden Sie unter Metriken mit Amazon überwachen CloudWatch.

Weitere Informationen erhalten Sie unter Metriken mit Amazon überwachen CloudWatch und Protokollierung und Überwachung in Amazon S3.

Protokollierung mit AWS CloudTrail für Verzeichnis-Buckets

Amazon S3 ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem AWS Service ausgeführten Aktionen bereitstellt. CloudTrail

erfasst alle API Aufrufe für Amazon S3 als Ereignisse. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Amazon S3 gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, den Zeitpunkt der Anfrage und weitere Details ermitteln. Wenn eine unterstützte Ereignisaktivität in Amazon S3 auftritt, wird diese Aktivität in einem CloudTrail Ereignis aufgezeichnet. Sie können AWS CloudTrail Trail verwenden, um Verwaltungsereignisse und Datenereignisse für Verzeichnis-Buckets zu protokollieren. Weitere Informationen finden Sie unter Amazon S3 CloudTrail S3-Ereignisse und Was ist AWS CloudTrail? im AWS CloudTrail Benutzerhandbuch.

CloudTrail Verwaltungsereignisse für Directory-Buckets

Standardmäßig werden Aktionen auf Bucket-Ebene für Verzeichnis-Buckets als Verwaltungsereignisse CloudTrail protokolliert. Der Wert `eventsource` für CloudTrail Verwaltungsereignisse für Verzeichnis-Buckets ist. `s3express.amazonaws.com` Wenn Sie Ihr AWS Konto einrichten, sind CloudTrail Verwaltungsereignisse standardmäßig aktiviert. Die folgenden regionalen API Endpunktoperationen (Operationen auf Bucket-Ebene oder Kontrollebene) werden protokolliert. API CloudTrail

- `CreateBucket`
- `DeleteBucket`
- `DeleteBucketPolicy`
- `PutBucketPolicy`
- `GetBucketPolicy`
- `ListDirectoryBuckets`
- `ListMultipartUploads`
- `GetBucketEncryption`
- `PutBucketEncryption`
- `DeleteBucketEncryption`

> **Note**
>
> `ListMultipartUploads`ist eine Operation mit zonalen Endpunkten. API Dieser API Vorgang wird jedoch CloudTrail als Verwaltungsereignis protokolliert. Weitere Informationen finden Sie `ListMultipartUploads`in der Amazon Simple Storage Service API Reference.

Weitere Informationen zu CloudTrail Verwaltungsereignissen finden Sie unter Protokollieren von Verwaltungsereignissen im AWS CloudTrail Benutzerhandbuch.

CloudTrail Datenereignisse für Verzeichnis-Buckets

Datenereignisse liefern Informationen über die Ressourcenoperationen, die auf oder in einer Ressource ausgeführt werden (z. B. Lesen oder Schreiben in ein Amazon-S3-Objekt). Sie werden auch als Vorgänge auf Datenebene bezeichnet. Datenereignisse sind oft Aktivitäten mit hohem Volume. Standardmäßig protokollieren CloudTrail Trails keine Datenereignisse, aber Sie können Trails so konfigurieren, dass Datenereignisse für Objekte protokolliert werden, die in Allzweck-Buckets und Verzeichnis-Buckets gespeichert sind. Weitere Informationen finden Sie unter Aktivieren der Protokollierung für Objekte in einem Bucket mithilfe der Konsole.

Wenn Sie Datenereignisse für einen Trail in protokollieren CloudTrail, können Sie wählen, ob Sie erweiterte oder einfache Event-Selektoren verwenden möchten. Um Datenereignisse für Objekte zu protokollieren, die in Verzeichnis-Buckets gespeichert sind, müssen Sie erweiterte Ereignisselektoren verwenden. Bei der Konfiguration erweiterter Ressourcenselektoren wählen Sie den Ressourcentyp aus oder geben diesen an. `AWS::S3Express::Object`

Die folgenden zonalen API Endpunktoperationen (Operationen auf Objektebene oder Datenebene) werden protokolliertAPI. CloudTrail

- `AbortMultipartUpload`
- `CompleteMultipartUpload`
- `CreateSession`
- `CopyObject`
- `CreateMultipartUpload`
- `DeleteObject`
- `DeleteObjects`
- `GetObject`
- `GetObjectAttributes`
- `HeadBucket`
- `HeadObject`
- `ListObjectsV2`
- `ListParts`
- `PutObject`

- UploadPart
- UploadPartCopy

Weitere Informationen zu CloudTrail Datenereignissen finden Sie unter Protokollieren von Datenereignissen im Benutzerhandbuch.AWS CloudTrail

Weitere Informationen zu CloudTrail Ereignissen für Directory-Buckets finden Sie in den folgenden Themen:

Themen

- CloudTrail Beispiele für Protokolldateien für Verzeichnis-Buckets

CloudTrail Beispiele für Protokolldateien für Verzeichnis-Buckets

Eine CloudTrail Protokolldatei enthält Informationen über den angeforderten API Vorgang, Datum und Uhrzeit des Vorgangs, Anforderungsparameter usw. Dieses Thema enthält Beispiele für CloudTrail Datenereignisse und Verwaltungsereignisse für Directory-Buckets.

Themen

- CloudTrail Beispiele für Datenereignisprotokolldateien für Directory-Buckets

CloudTrail Beispiele für Datenereignisprotokolldateien für Directory-Buckets

Das folgende Beispiel zeigt ein Beispiel für eine CloudTrail Protokolldatei, das Folgendes zeigt CreateSession.

```
{
    "eventVersion": "1.09",
    "userIdentity": {
      "type": "AssumedRole",
      "principalId": "AROAIDPPEZS35WEXAMPLE:AssumedRoleSessionName",
      "arn": "arn:aws:sts::111122223333assumed-role/RoleToBeAssumed/MySessionName",
      "accountId": "111122223333",
      "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
      "sessionContext": {
        "sessionIssuer": {
          "type": "Role",
          "principalId": "AROAIDPPEZS35WEXAMPLE",
          "arn": "arn:aws:iam::111122223333:role/RoleToBeAssumed",
```

```
              "accountId": "111122223333",
              "userName":"RoleToBeAssumed
            },

            "attributes": {
              "creationDate": "2024-07-02T00:21:16Z",
            "mfaAuthenticated": "false"
            }
          }
        },
        "eventTime": "2024-07-02T00:22:11Z",
        "eventSource": "s3express.amazonaws.com",
        "eventName": "CreateSession",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "72.21.198.68",
        "userAgent": "aws-sdk-java/2.20.160-SNAPSHOT
 Linux/5.10.216-225.855.amzn2.x86_64 OpenJDK_64-Bit_Server_VM/11.0.23+9-LTS
 Java/11.0.23 vendor/Amazon.com_Inc. md/internal exec-env/AWS_Lambda_java11 io/sync
 http/Apache cfg/retry-mode/standard",
        "requestParameters": {
          "bucketName": "bucket-base-name--usw2-az1--x-s3".
            "host": "bucket-base-name--usw2-az1--x-s3.s3express-usw2-az1.us-
west-2.amazonaws.com",
            "x-amz-create-session-mode": "ReadWrite"
        },
        "responseElements": {
            "credentials": {
                "accessKeyId": "AKIAI44QH8DHBEXAMPLE"
                "expiration": ""Mar 20, 2024, 11:16:09 PM",
                "sessionToken": "<session token string>"
           },
        },
        "additionalEventData": {
            "SignatureVersion": "SigV4",
            "cipherSuite": "TLS_AES_128_GCM_SHA256",
            "bytesTransferredIn": 0,
            "AuthenticationMethod": "AuthHeader",
            "xAmzId2": "q6xhNJYmhg",
            "bytesTransferredOut": 1815,
            "availabilityZone": "usw2-az1"
          },
          "requestID": "28d2faaf-3319-4649-998d-EXAMPLE72818",
          "eventID": "694d604a-d190-4470-8dd1-EXAMPLEe20c1",
          "readOnly": true,
```

```
          "resources": [
            {
              "type": "AWS::S3Express::Object",
              "ARNPrefix": "arn:aws:s3express:us-west-2:111122223333:bucket-base-name--
usw2-az1--x-s3"
            },
            {
              "accountId": "111122223333"
              "type": "AWS::S3Express::DirectoryBucket",
              "ARN": "arn:aws:s3express:us-west-2:111122223333:bucket-base-name--usw2-
az1--x-s3"
             }
           ],
           "eventType": "AwsApiCall",
           "managementEvent": false,
           "recipientAccountId": "111122223333",
           "eventCategory": "Data",
           "tlsDetails": {
             "tlsVersion": "TLSv1.3",
             "cipherSuite": "TLS_AES_128_GCM_SHA256",
             "clientProvidedHostHeader": "bucket-base-name--usw2-az1--x-s3.s3express-
usw2-az1.us-west-2.amazonaws.com"
            }
          }
```

Um zonale API Endpunktoperationen (Operationen auf Objektebene oder Datenebene) zu verwenden, können Sie den `CreateSession` API Vorgang verwenden, um Sitzungen zu erstellen und zu verwalten, die für die Autorisierung von Datenanforderungen mit geringer Latenz optimiert sind. Sie können ihn auch verwenden, um den Umfang der `CreateSession` Protokollierung zu reduzieren. Um zu ermitteln, welche zonalen API Operationen während einer Sitzung ausgeführt wurden, können Sie die Werte `accessKeyId` unter dem `responseElements` in Ihrer `CreateSession` Protokolldatei mit denen `accessKeyId` in der Protokolldatei anderer zonaler API Operationen abgleichen. Weitere Informationen finden Sie unter `CreateSession` Autorisierung.

Das folgende Beispiel zeigt ein Beispiel für eine CloudTrail Protokolldatei, die den `GetObject`APIVorgang veranschaulicht, der von `CreateSession` authentifiziert wurde.

```
    {
        "eventVersion": "1.09",
        "userIdentity": {
          "type": "AssumedRole",
          "principalId": "AROAIDPPEZS35WEXAMPLE:AssumedRoleSessionName",
```

```
          "arn": "arn:aws:sts::111122223333assumed-role/RoleToBeAssumed/MySessionName",
          "accountId": "111122223333",
          "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
          "sessionContext": {
            "attributes": {
              "creationDate": "2024-07-02T00:21:49Z"
            }
          }
        },
        "eventTime": "2024-07-02T00:22:01Z",
        "eventSource": "s3express.amazonaws.com",
        "eventName": "GetObject",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "72.21.198.68",
        "userAgent": "aws-sdk-java/2.25.66 Linux/5.10.216-225.855.amzn2.x86_64
 OpenJDK_64-Bit_Server_VM/17.0.11+9-LTS Java/17.0.11 vendor/Amazon.com_Inc. md/internal
 exec-env/AWS_Lambda_java17 io/sync http/Apache cfg/retry-mode/legacy",
        "requestParameters": {
          "bucketName": "bucket-base-name--usw2-az1--x-s3",
          "x-amz-checksum-mode": "ENABLED",
          "Host": "bucket-base-name--usw2-az1--x-s3.s3express-usw2-az1.us-
west-2.amazonaws.com",
          "key": "test-get-obj-with-checksum"
        },
        "responseElements": null,
        "additionalEventData": {
          "SignatureVersion": "Sigv4",
          "CipherSuite": "TLS_AES_128_GCM_SHA256",
          "bytesTransferredIn": 0,
          "AuthenticationMethod": "AuthHeader",
          "x-amz-id-2": "oOy6w8K7LFsyFN",
          "bytesTransferredOut": 9,
          "availabilityZone": "usw2-az1",
          "sessionModeApplied": "ReadWrite"
         },
          "requestID": "28d2faaf-3319-4649-998d-EXAMPLE72818",
          "eventID": "694d604a-d190-4470-8dd1-EXAMPLEe20c1",
          "readOnly": true,
          "resources": [
            {
              "type": "AWS::S3Express::Object",
              "ARNPrefix": "arn:aws:s3express:us-west-2:111122223333:bucket-base-name--
usw2-az1--x-s3"
            },
```

```
            {
              "accountId": "111122223333",
              "type": "AWS::S3Express::DirectoryBucket",
              "ARN": "arn:aws:s3express:us-west-2:111122223333:bucket-base-name--usw2-
az1--x-s3"
             }
           ],
           "eventType": "AwsApiCall",
           "managementEvent": false,
           "recipientAccountId": "111122223333",
           "eventCategory": "Data",
           "tlsDetails": {
             "tlsVersion": "TLSv1.3",
             "cipherSuite": "TLS_AES_128_GCM_SHA256",
             "clientProvidedHostHeader": "bucket-base-name--usw2-az1--x-s3.s3express-
usw2-az1.us-west-2.amazonaws.com"
            }
          }
```

Im obigen Beispiel für eine `GetObject` Protokolldatei entspricht `accessKeyId` (AKIAI44QH8DHBEXAMPLE) dem Wert `accessKeyId` unter dem `responseElements` im Beispiel der CreateSession Protokolldatei. Die Übereinstimmung `accessKeyId` gibt die Sitzung an, in der der `GetObject` Vorgang ausgeführt wurde.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der eine `DeleteObjects` Aktion in einem Verzeichnis-Bucket demonstriert, die von S3 Lifecycle aufgerufen wurde. Weitere Informationen finden Sie unter Working with S3 Lifecycle for directory buckets.

```
eventVersion:"1.09",
  userIdentity:{

    type:"AWSService",
    invokedBy:"lifecycle.s3.amazonaws.com"
  },
  eventTime:"2024-09-11T00:55:54Z",
  eventSource:"s3express.amazonaws.com",
  eventName:"DeleteObjects",
  awsRegion:"us-east-2",
  sourceIPAddress:"lifecycle.s3.amazonaws.com",
  userAgent:"gamma.lifecycle.s3.amazonaws.com",
  requestParameters:{
```

```
    bucketName:"amzn-s3-demo-bucket--use2-az2--x-s3",
    'x-amz-expected-bucket-owner':"637423581905",
    Host:"amzn-s3-demo-bucket--use2-az2--x-s3.gamma.use2-az2.express.s3.aws.dev",
    delete:"",
    'x-amz-sdk-checksum-algorithm':"CRC32C"
  },
  responseElements:null,
  additionalEventData:{

    SignatureVersion:"Sigv4",
    CipherSuite:"TLS_AES_128_GCM_SHA256",
    bytesTransferredIn:41903,
    AuthenticationMethod:"AuthHeader",
    'x-amz-id-2':"9H5YWZY0",
    bytesTransferredOut:35316,
    availabilityZone:"use2-az2",
    sessionModeApplied:"ReadWrite"
  },
  requestID:"011eeadd04000191",
  eventID:"d3d8b116-219d-4ee6-a072-5f9950733c74",
  readOnly:false,
  resources:[

    {

      type:"AWS::S3Express::Object",
      ARNPrefix:"arn:aws:s3express:us-east-2:637423581905:bucket/amzn-s3-demo-bucket--
use2-az2--x-s3/"
    },
    {

      accountId:"637423581905",
      type:"AWS::S3Express::DirectoryBucket",
      ARN:"arn:aws:s3express:us-east-2:637423581905:bucket/amzn-s3-demo-bucket--use2-
az2--x-s3"
    }
  ],
  eventType:"AwsApiCall",
  managementEvent:false,
  recipientAccountId:"637423581905",
  sharedEventID:"59f877ac-1dd9-415d-b315-9bb8133289ce",
  eventCategory:"Data"
}
```

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der eine `Access Denied` Anforderung für eine von S3 Lifecycle aufgerufene `CreateSession` Aktion demonstriert. Weitere Informationen finden Sie unter CreateSession.


```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "gamma.lifecycle.s3.amazonaws.com"
    },
    "eventTime": "2024-09-11T18:13:08Z",
    "eventSource": "s3express.amazonaws.com",
    "eventName": "CreateSession",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "gamma.lifecycle.s3.amazonaws.com",
    "userAgent": "gamma.lifecycle.s3.amazonaws.com",
    "errorCode": "AccessDenied",
    "errorMessage": "Access Denied",
    "requestParameters": {
        "bucketName": "amzn-s3-demo-bucket--use2-az2--x-s3",
        "Host": "amzn-s3-demo-bucket--use2-az2--x-s3.gamma.use2-
az2.express.s3.aws.dev",
        "x-amz-create-session-mode": "ReadWrite",
        "x-amz-server-side-encryption": "AES256"
    },
    "responseElements": null,
    "additionalEventData": {
        "SignatureVersion": "Sigv4",
        "CipherSuite": "TLS_AES_128_GCM_SHA256",
        "bytesTransferredIn": 0,
        "AuthenticationMethod": "AuthHeader",
        "x-amz-id-2": "zuDDC1VNbC4LoNwUIc5",
        "bytesTransferredOut": 210,
        "availabilityZone": "use2-az2"
    },
    "requestID": "010932f174000191e24a0",
    "eventID": "dce7cc46-4cd3-46c0-9a47-d1b8b70e301c",
    "readOnly": true,
    "resources": [{
            "type": "AWS::S3Express::Object",
            "ARNPrefix": "arn:aws:s3express:us-east-2:637423581905:bucket/amzn-s3-demo-
bucket--use2-az2--x-s3/"
        },
```

```
        {
            "accountId": "637423581905",
            "type": "AWS::S3Express::DirectoryBucket",
            "ARN": "arn:aws:s3express:us-east-2:637423581905:bucket/amzn-s3-demo-
bucket--use2-az2--x-s3"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "637423581905",
    "sharedEventID": "da96b5bd-6066-4a8d-ad8d-f7f427ca7d58",
    "eventCategory": "Data"
}
```

Optimierung der Leistung von Directory-Buckets

Amazon S3 Express One Zone ist eine leistungsstarke S3-Speicherklasse mit einer einzigen Availability Zone (AZ), die speziell für den konsistenten Datenzugriff im einstelligen Millisekundenbereich für Ihre latenzempfindlichsten Anwendungen entwickelt wurde. S3 Express One Zone ist die erste S3-Speicherklasse, die Ihnen die Möglichkeit bietet, leistungsstarke Objektspeicher- und AWS Rechenressourcen wie Amazon Elastic Compute Cloud, Amazon Elastic Kubernetes Service und Amazon Elastic Container Service innerhalb einer einzigen Availability Zone zusammenzulegen. Die gemeinsame Nutzung Ihrer Speicher- und Computingressourcen optimiert die Rechenleistung und die Kosten und sorgt für eine höhere Datenverarbeitungsgeschwindigkeit.

S3 Express One Zone bietet eine ähnliche Leistungselastizität wie andere S3-Speicherklassen, jedoch mit konsistenten Latenzen für Lese- und Schreibanfragen im ersten Byte im einstelligen Millisekundenbereich – bis zu zehnmal schneller als S3 Standard. S3 Express One Zone wurde von Grund auf für die Unterstützung von Burst-Durchsätzen bis hin zu sehr hohen Aggregatwerten konzipiert. Die Speicherklasse S3 Express One Zone verwendet eine maßgeschneiderte Architektur, um die Leistung zu optimieren und durch die Speicherung von Daten auf Hochleistungshardware eine konstant niedrige Anforderungslatenz zu gewährleisten. Das Objektprotokoll für S3 Express One Zone wurde verbessert, um die Authentifizierung und den Metadaten-Overhead zu optimieren.

Um die Zugriffsgeschwindigkeit weiter zu erhöhen und Hunderttausende von Anfragen pro Sekunde zu unterstützen, speichert S3 Express One Zone Daten außerdem in einem neuen Bucket-Typ, einem Amazon-S3-Verzeichnis-Bucket. Jeder S3-Verzeichnis-Bucket kann Hunderttausende von Transaktionen pro Sekunde unterstützen (). TPS

Die Kombination aus leistungsstarker, speziell entwickelter Hardware und Software, die Datenzugriffsgeschwindigkeiten im einstelligen Millisekundenbereich bietet, und Verzeichnis-Buckets, die für eine große Anzahl von Transaktionen pro Sekunde skalierbar sind, machen S3 Express One Zone zur besten Amazon-S3-Speicherklasse für anforderungsintensive Operationen oder leistungskritische Anwendungen.

Die folgenden Themen beschreiben bewährte Verfahren und Designmuster zur Optimierung der Leistung von Anwendungen, die die Speicherklasse Sr Express One Zone verwenden.

Themen

- Leistungsrichtlinien und Entwurfsmuster für S3 Express One Zone

Leistungsrichtlinien und Entwurfsmuster für S3 Express One Zone

Um die beste Leistung bei der Verwendung von Directory-Buckets zu erzielen, empfehlen wir die folgenden Richtlinien.

Themen

- Verwenden der sitzungsbasierten Authentifizierung
- Bewährte Methoden für zusätzliche S3-Prüfsummen
- Verwenden Sie die neueste Version der AWS SDKs und der gängigen Runtime-Bibliotheken

Verwenden der sitzungsbasierten Authentifizierung

Directory-Buckets unterstützen einen neuen sitzungsbasierten Autorisierungsmechanismus zur Authentifizierung und Autorisierung von Anfragen an einen Directory-Bucket. Bei der sitzungsbasierten Authentifizierung verwenden sie den `CreateSession` API Vorgang AWS SDKs automatisch, um ein temporäres Sitzungstoken zu erstellen, das für die Autorisierung von Datenanforderungen an einen Verzeichnis-Bucket mit geringer Latenz verwendet werden kann.

AWS SDKsSie verwenden den `CreateSession` API Vorgang, um temporäre Anmeldeinformationen anzufordern und anschließend automatisch alle 5 Minuten Token für Sie zu erstellen und zu aktualisieren. Um die Leistungsvorteile von Directory-Buckets nutzen zu können, empfehlen wir, dass Sie den verwenden, AWS SDKs um die `CreateSession` API Anfrage zu initiieren und zu verwalten. Weitere Informationen zu diesem sitzungsbasierten Modell finden Sie unter Autorisieren von zonalen API Endpunktoperationen mit `CreateSession`.

Bewährte Methoden für zusätzliche S3-Prüfsummen

Directory-Buckets bieten Ihnen die Möglichkeit, den Prüfsummenalgorithmus auszuwählen, der zur Validierung Ihrer Daten beim Hoch- oder Herunterladen verwendet wird. Sie können einen der folgenden Algorithmen zur Prüfung der Datenintegrität zwischen Secure Hash Algorithms (SHA) oder Cyclic Redundancy Check (CRC) wählen:CRC32,CRC32C, SHA -1 und -256. SHA MD5Auf Prüfsummen basierende Prüfsummen werden von der S3 Express One Zone-Speicherklasse nicht unterstützt.

CRC32ist die Standardprüfsumme, die von verwendet wird, AWS SDKs wenn Daten in oder aus Verzeichnis-Buckets übertragen werden. Wir empfehlen die Verwendung von CRC32 und, CRC32C um die beste Leistung mit Verzeichnis-Buckets zu erzielen.

Verwenden Sie die neueste Version der AWS SDKs und der gängigen Runtime-Bibliotheken

Einige von ihnen stellen AWS SDKs auch die AWS Common Runtime (CRT) -Bibliotheken bereit, um die Leistung von S3-Clients weiter zu beschleunigen. Dazu SDKs gehören die AWS SDK for Java 2.x AWS SDK for C++, die und die AWS SDK for Python (Boto3). Der CRT basierte S3-Client überträgt Objekte mit verbesserter Leistung und Zuverlässigkeit in und aus Verzeichnis-Buckets, indem er automatisch den mehrteiligen API Upload-Vorgang und Bytebereichs-Abrufe verwendet, um Verbindungen mit horizontaler Skalierung zu automatisieren.

Um die höchste Leistung mit den Verzeichnis-Buckets zu erzielen, empfehlen wir, die neueste Version von zu verwenden, die die CRT Bibliotheken enthält AWS SDKs, oder die () zu verwenden. AWS Command Line Interface AWS CLI

Entwickeln mit Verzeichnis-Buckets

Nachdem Sie Ihren Verzeichnis-Bucket erstellt haben, können Sie sofort mit Lese- und Schreibvorgängen mit sehr niedriger Latenz beginnen. Sie können mit Ihrem Verzeichnis-Bucket kommunizieren, indem Sie eine Endpunktverbindung über eine virtuelle private Cloud (VPC) verwenden, oder Sie können zonale und regionale API Operationen verwenden, um Ihre Objekte und Verzeichnis-Buckets zu verwalten. Sie können mit Verzeichnis-Buckets arbeiten AWS SDKs, indem Sie die Amazon S3-Konsole, die AWS Befehlszeilenschnittstelle (AWS CLI) und Amazon S3 REST APIs verwenden.

Themen

- Regionale und zonale Endpunkte für Directory-Buckets
- Arbeiten mit Verzeichnis-Buckets mithilfe der S3-Konsole AWS CLI, und AWS SDKs
- APIVerzeichnis-Bucket-Operationen

Regionale und zonale Endpunkte für Directory-Buckets

Um von Ihrer Virtual Private Cloud (VPC) aus auf die regionalen und zonalen Endpunkte für Verzeichnis-Buckets zuzugreifen, können Sie Gateway-Endpunkte verwenden. VPC Nachdem Sie einen Gateway-Endpunkt erstellt haben, können Sie ihn in Ihrer Routentabelle als Ziel für den Datenverkehr hinzufügen, der von Ihnen zu Ihrem Bucket geleitet wird. VPC Für die Nutzung von Gateway-Endpunkten fallen keine zusätzlichen Gebühren an. Weitere Informationen zur Konfiguration von VPC Gateway-Endpunkten finden Sie unter. Netzwerke für Verzeichnis-Buckets

APIOperationen auf Bucket-Ebene (Steuerungsebene) sind über einen regionalen Endpunkt verfügbar und werden als regionale Endpunktoperationen bezeichnet. API Beispiele für regionale API Endpunktoperationen sind `CreateBucket` und. `DeleteBucket`

Sie verwenden Zonal (APIEndpunktoperationen auf Objektebene oder Datenebene), um Ihre Objekte hochzuladen und zu verwalten. Zonale API Endpunktoperationen sind über einen zonalen Endpunkt verfügbar. Beispiele für zonale API Operationen sind und. `PutObject` `CopyObject`

Weitere Informationen zu regionalen und zonalen Endpunkten für Verzeichnis-Buckets in Availability Zones finden Sie unter. Regionale und zonale Endpunkte für Verzeichnis-Buckets in einer Availability Zone

Arbeiten mit Verzeichnis-Buckets mithilfe der S3-Konsole AWS CLI, und AWS SDKs

Sie können mit der Speicherklasse und den Verzeichnis-Buckets von S3 Express One Zone arbeiten AWS SDKs, indem Sie die Amazon S3-Konsole AWS Command Line Interface (AWS CLI) und Amazon S3 REST API verwenden.

S3-Konsole

Um mit der Verwendung der S3-Konsole zu beginnen, führen Sie diese Schritte aus:

- Verzeichnis-Buckets in einer Availability Zone erstellen

- Leeren eines Verzeichnis-Buckets
- Löschen eines Verzeichnis-Buckets

Ein vollständiges Tutorial finden Sie unter Tutorial: Erste Schritte mit S3 Express One Zone.

AWS SDKs

S3 Express One Zone unterstützt Folgendes AWS SDKs:

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java 2.x
- AWS SDK for JavaScript v3
- AWS SDK for .NET
- AWS SDK for PHP
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby
- AWS SDK for Kotlin
- AWS SDK for Rust

Wenn Sie mit S3 Express One Zone arbeiten, empfehlen wir die Verwendung der neuesten Version von AWS SDKs. Die AWS SDKs für S3 Express One Zone unterstützte Zone kümmert sich in Ihrem Namen um die Einrichtung, Aktualisierung und Beendigung von Sitzungen. Das bedeutet, dass Sie sofort mit dem API Betrieb beginnen können, nachdem Sie das heruntergeladen und installiert AWS SDKs und die erforderlichen IAM Berechtigungen konfiguriert haben. Weitere Informationen finden Sie unter Autorisieren eines regionalen Endpunkts APIs mit IAM.

Informationen zu den AWS SDKs, einschließlich Informationen zum Herunterladen und Installieren, finden Sie unter Tools, auf denen Sie aufbauen können AWS.

AWS SDK-Beispiele finden Sie nachfolgend.

- Verzeichnis-Buckets in einer Availability Zone erstellen
- Leeren eines Verzeichnis-Buckets
- Löschen eines Verzeichnis-Buckets

AWS Command Line Interface (AWS CLI)

Sie können die AWS Command Line Interface (AWS CLI) verwenden, um Verzeichnis-Buckets zu erstellen und unterstützte regionale und zonale API Endpunktoperationen für S3 Express One Zone zu verwenden.

Informationen zu den AWS CLI ersten Schritten finden Sie unter Erste Schritte mit dem AWS CLI in der AWS CLI Befehlsreferenz.

> **Note**
>
> Um Directory-Buckets mit den allgemeinen `aws s3` Befehlen zu verwenden, aktualisieren Sie Ihre Version AWS CLI auf die neueste Version. Weitere Informationen zur Installation und Konfiguration von finden Sie unter Installieren oder Aktualisieren der neuesten Version von AWS CLI in der AWS CLI Befehlsreferenz. AWS CLI

AWS CLI Beispiele finden Sie in den folgenden Abschnitten:

- Verzeichnis-Buckets in einer Availability Zone erstellen
- Leeren eines Verzeichnis-Buckets
- Löschen eines Verzeichnis-Buckets

APIVerzeichnis-Bucket-Operationen

Um Verzeichnis-Buckets zu verwalten, können Sie regionale API Endpunktoperationen (Bucket-Ebene oder Steuerungsebene) verwenden. Um Objekte in Ihren Verzeichnis-Buckets zu verwalten, können Sie zonale Endpunktoperationen (Objektebene oder Datenebene) verwenden. API Weitere Informationen erhalten Sie unter Netzwerke für Verzeichnis-Buckets und Endpunkte und Gateway-Endpunkte VPC.

Regionale Endpunktoperationen API

Die folgenden regionalen API Endpunktoperationen werden für S3 Express One Zone unterstützt:

- CreateBucket
- DeleteBucket
- DeleteBucketLifecycle

- DeleteBucketPolicy
- GetBucketLifecycleConfiguration
- GetBucketPolicy
- ListDirectoryBuckets
- PutBucketLifecycleConfiguration
- PutBucketPolicy

Zonale Endpunktoperationen API

Die folgenden zonalen API Endpunktoperationen werden für S3 Express One Zone unterstützt:

- CreateSession
- CopyObject
- DeleteObject
- DeleteObjects
- GetObject
- GetObjectAttributes
- HeadBucket
- HeadObject
- ListObjectsV2
- PutObject
- AbortMultipartUpload
- CompleteMultiPartUpload
- CreateMultipartUpload
- ListMultipartUploads
- ListParts
- UploadPart
- UploadPartCopy
- PutBucketEncryption
- GetBucketEncryption
- DeleteBucketEncryption

Zugriffskontrolle in Amazon S3

AWS In ist eine Ressource eine Entität, mit der Sie arbeiten können. In Amazon Simple Storage Service (S3) sind Buckets und Objekte die ursprünglichen Amazon S3 S3-Ressourcen. Jeder S3-Kunde hat wahrscheinlich Buckets mit Objekten darin. Als S3 um neue Funktionen erweitert wurde, kamen auch zusätzliche Ressourcen hinzu, aber nicht jeder Kunde nutzt diese funktionsspezifischen Ressourcen. Weitere Informationen zu Amazon S3 S3-Ressourcen finden Sie unterS3-Ressourcen.

Standardmäßig werden alle Amazon S3-Ressourcen als privat eingestuft. Außerdem können standardmäßig der Root-Benutzer der Ressource AWS-Konto , die die Ressource erstellt hat (Ressourcenbesitzer), und IAM Benutzer innerhalb dieses Kontos mit den erforderlichen Berechtigungen auf eine von ihnen erstellte Ressource zugreifen. Der Besitzer der Ressource entscheidet, wer sonst noch auf die Ressource zugreifen kann und welche Aktionen andere Benutzer an der Ressource ausführen dürfen. S3 verfügt über verschiedene Tools zur Zugriffsverwaltung, mit denen Sie anderen Zugriff auf Ihre S3-Ressourcen gewähren können.

Die folgenden Abschnitte bieten Ihnen einen Überblick über S3-Ressourcen, die verfügbaren S3-Zugriffsverwaltungstools und die besten Anwendungsfälle für die einzelnen Zugriffsverwaltungstools. Die Listen in diesen Abschnitten sollen umfassend sein und alle S3-Ressourcen, Zugriffsverwaltungstools und allgemeine Anwendungsfälle für die Zugriffsverwaltung enthalten. Gleichzeitig sind diese Abschnitte als Verzeichnisse konzipiert, die Sie zu den gewünschten technischen Details führen. Wenn Sie sich mit einigen der folgenden Themen gut auskennen, können Sie zu dem Abschnitt springen, der für Sie relevant ist.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

Topics

- S3-Ressourcen
- Identitäten
- Tools zur Zugriffsverwaltung
- Aktionen
- Anwendungsfälle für die Zugriffsverwaltung
- Fehlerbehebung bei der Zugriffsverwaltung

S3-Ressourcen

Die ursprünglichen Amazon S3 S3-Ressourcen sind Buckets und die Objekte, die sie enthalten. Wenn neue Funktionen zu S3 hinzugefügt werden, kommen auch neue Ressourcen hinzu. Im Folgenden finden Sie eine vollständige Liste der S3-Ressourcen und ihrer jeweiligen Funktionen.

Ressourcentyp	Amazon S3 S3-Funktion	Beschreibung
`bucket`	Kern-Features	Ein Bucket ist ein Container für Objekte. Um ein Objekt in S3 zu speichern, erstellen Sie einen Bucket und laden Sie dann ein oder mehrere Objekte in den Bucket hoch. Weitere Informationen finden Sie unter Erstellen, Konfigurieren und Arbeiten mit Amazon S3-Buckets.
`object`		Ein Objekt kann eine Datei und alle Metadaten sein, die diese Datei beschreiben. Wenn sich ein Objekt im Bucket befindet, können Sie es öffnen, herunterladen und verschieben. Weitere Informationen finden Sie unter Arbeiten mit Objekten in Amazon S3.
`accesspoint`	Zugriffspunkte	Access Points sind benannte Netzwerkendpunkte, die an Buckets angehängt sind, mit denen Sie Amazon S3 S3-Objektoperationen wie `GetObject` und `PutObject` ausführen können. Jeder Access Point verfügt über eigene Berechtigungen, Netzwerksteuerungen und eine benutzerdefinierte Zugriffspunktrichtlinie, die mit der Bucket-Richtlinie zusammenarbeitet, die dem zugrunde liegenden Bucket zugeordnet ist. Sie können jeden Access Point so konfigurieren, dass er nur Anfragen von einer Virtual Private Cloud (VPC) akzeptiert, oder benutzerdefinierte Einstellungen für den Block Public Access für jeden Access Point konfigurieren. Weitere Informationen finden Sie unter Verwaltung des Zugriffs auf gemeinsam genutzte Datensätze mit Access Points.

Ressourcentyp	Amazon S3 S3-Funktion	Beschreibung
`objectlambdaaccesspoint`		Ein Object Lambda Access Point ist ein Access Point für einen Bucket, der auch mit einer Lambda-Funktion verknüpft ist. Mit Object Lambda Access Point können Sie Amazon S3 Ihren eigenen Code und `HEAD` Anfragen zur Änderung und Verarbeitung von Daten hinzufügen `GETLIST`, wenn diese an eine Anwendung zurückgegeben werden. Weitere Informationen finden Sie unter Erstellen von Objekt-Lambda-Zugriffspunkten.
`multiregionaccesspoint`		Multi-Region-Access Points bieten einen globalen Endpunkt, über den Anwendungen Anfragen von Amazon S3 S3-Buckets bearbeiten können, die sich in mehreren AWS Regionen befinden. Sie können Multi-Regions Access Points verwenden, um Multi-Regions-Anwendungen mit derselben Architektur zu erstellen, die in einer einzelnen Region verwendet wird, und diese Anwendungen dann überall auf der Welt ausführen. Anstatt Anfragen über das überlastete öffentliche Internet zu senden, werden Anwendungsanfragen an einen globalen Multi-Region-Access Point-Endpunkt automatisch über das AWS globale Netzwerk an den nächstgelegenen Amazon S3 S3-Bucket weitergeleitet. Weitere Informationen finden Sie unter Verwaltung von regionsübergreifendem Verkehr mit multiregionalen Access Points.

Ressourcentyp	Amazon S3 S3-Funktion	Beschreibung
`job`	S3-BatchVorgänge	Ein Job ist eine Ressource der S3-Funktion Batch Operations. Sie können S3 Batch Operations verwenden, um umfangreiche Batch-Operationen für Listen von Amazon S3 S3-Objekten durchzuführen, die Sie angeben. Amazon S3 verfolgt den Fortschritt des Batch-Vorgangs, sendet Benachrichtigungen und speichert einen detaillierten Abschlussbericht über alle Aktionen, sodass Sie ein vollständig verwaltetes, überprüfbares und serverloses Erlebnis erhalten. Weitere Informationen finden Sie unter Massenausführung von Objektoperationen mit Batch Operations.
`storagelensconfiguration`	S3 Storage Lens	Eine S3 Storage Lens-Konfiguration erfasst unternehmensweite Speichermetriken und Benutzerdaten für alle Konten. S3 Storage Lens bietet Administratoren eine zentrale Ansicht der Objektspeichernutzung und der Aktivitäten von Hunderten oder sogar Tausenden von Konten in einem Unternehmen mit Details, um Erkenntnisse auf mehreren Aggregationsebenen zu gewinnen. Weitere Informationen finden Sie unter Bewerten Ihrer Speicheraktivität und -nutzung mit Amazon S3 Storage Lens.
`storagelensgroup`		Eine S3 Storage Lens-Gruppe aggregiert Metriken mithilfe benutzerdefinierter Filter, die auf Objektmetadaten basieren. Mithilfe von S3 Storage Lens-Gruppen können Sie Merkmale Ihrer Daten untersuchen, z. B. die Verteilung von Objekten nach Alter, Ihre häufigsten Dateitypen und mehr. Weitere Informationen finden Sie unter Arbeiten mit S3 Storage Lens-Gruppen zum Filtern und Aggregieren von Metriken.

Ressourcentyp	Amazon S3 S3-Funktion	Beschreibung
`accessgrantsinstance`	S3 Access Grants	Eine S3 Access Grants-Instanz ist ein Container für die von Ihnen erstellten S3-Grants. Mit S3 Access Grants können Sie Berechtigungen für Ihre Amazon S3 S3-Daten für IAM Identitäten in Ihrem Konto, IAM Identitäten in anderen Konten (kontoübergreifend) und Verzeichnisidentitäten erstellen, die AWS IAM Identity Center aus Ihrem Unternehmensverzeichnis hinzugefügt wurden. Weitere Informationen zu S3 Access Grants finden Sie unter. Verwalten des Zugriffs mit S3-Zugriffsberechtigungen
`accessgrantslocation`		Ein Access Grants-Standort ist ein Bucket, ein Präfix innerhalb eines Buckets oder ein Objekt, das Sie in Ihrer S3 Access Grants-Instanz registrieren. Sie müssen Standorte innerhalb der S3 Access Grants-Instanz registrieren, bevor Sie einen Zuschuss für diesen Standort erstellen können. Anschließend können Sie mit S3 Access Grants Zugriff auf den Bucket, das Präfix oder das Objekt für IAM Identitäten in Ihrem Konto, IAM Identitäten in anderen Konten (kontoübergreifend) und Verzeichnisidentitäten gewähren, die AWS IAM Identity Center aus Ihrem Unternehmensverzeichnis hinzugefügt wurden. Weitere Informationen zu S3 Access Grants finden Sie unter Verwalten des Zugriffs mit S3-Zugriffsberechtigungen
`accessgrant`		Ein Access Grant ist eine individuelle Gewährung Ihrer Amazon S3 S3-Daten. Mit S3 Access Grants können Sie Berechtigungen für Ihre Amazon S3 S3-Daten für IAM Identitäten in Ihrem Konto, IAM Identitäten in anderen Konten (kontoübergreifend) und Verzeichnisidentitäten erstellen, die AWS IAM Identity Center aus Ihrem Unternehmensverzeichnis hinzugefügt wurden. Weitere Informationen zu S3 Access Grants finden Sie unter Verwalten des Zugriffs mit S3-Zugriffsberechtigungen

Buckets

Es gibt zwei Arten von Amazon S3 S3-Buckets: Allzweck-Buckets und Verzeichnis-Buckets.

- Allzweck-Buckets sind der ursprüngliche S3-Bucket-Typ und werden für die meisten Anwendungsfälle und Zugriffsmuster empfohlen. Allzweck-Buckets ermöglichen auch Objekte, die in allen Speicherklassen gespeichert werden, mit Ausnahme von S3 Express One Zone. Weitere Informationen zu S3-Speicherklassen finden Sie unter. Amazon S3 S3-Speicherklassen verstehen und verwalten
- Directory-Buckets verwenden die Speicherklasse S3 Express One Zone. Dies wird empfohlen, wenn Ihre Anwendung leistungsempfindlich ist und von einstelligen Millisekunden und Latenzen profitiert. PUT GET Weitere Informationen finden Sie unter Arbeiten mit Verzeichnis-Buckets, S3 Express One Zone und Autorisieren eines regionalen Endpunkts APIs mit IAM.

Kategorisieren von S3-Ressourcen

Amazon S3 bietet Funktionen zur Kategorisierung und Organisation Ihrer S3-Ressourcen. Die Kategorisierung Ihrer Ressourcen ist nicht nur nützlich, um sie zu organisieren, sondern Sie können auch Zugriffsverwaltungsregeln auf der Grundlage der Ressourcenkategorien festlegen. Insbesondere Präfixe und Tagging sind zwei Funktionen zur Speicherorganisation, die Sie bei der Festlegung von Zugriffsverwaltungsberechtigungen verwenden können.

> **Note**
>
> Die folgenden Informationen beziehen sich auf Allzweck-Buckets. Verzeichnis-Buckets unterstützen kein Tagging und sie haben Präfixbeschränkungen. Weitere Informationen finden Sie unter Autorisieren eines regionalen Endpunkts APIs mit IAM.

- Präfixe — Ein Präfix in Amazon S3 ist eine Zeichenfolge am Anfang eines Objektschlüsselnamens, die verwendet wird, um die Objekte zu organisieren, die in Ihren S3-Buckets gespeichert sind. Sie können ein Trennzeichen verwenden, z. B. einen Schrägstrich (/), um das Ende des Präfixes innerhalb des Objektschlüsselnamens anzugeben. Beispielsweise können Sie Objektschlüsselnamen verwenden, die mit dem `engineering/` Präfix beginnen, oder Objektschlüsselnamen, die mit dem Präfix beginnen. `marketing/campaigns/` Die Verwendung eines Trennzeichens am Ende Ihres Präfixes, z. B. eines Schrägstrichs, / emuliert die Benennungskonventionen für Ordner und Dateien. In S3 ist das Präfix jedoch Teil des

Objektschlüsselnamens. In S3-Buckets für allgemeine Zwecke gibt es keine tatsächliche Ordnerhierarchie.

Amazon S3 unterstützt das Organisieren und Gruppieren von Objekten mithilfe ihrer Präfixe. Sie können den Zugriff auf Objekte auch anhand ihrer Präfixe verwalten. Sie können beispielsweise den Zugriff auf Objekte beschränken, deren Namen mit einem bestimmten Präfix beginnen.

Weitere Informationen finden Sie unter Organisieren von Objekten mit Präfixen. Die S3-Konsole verwendet das Konzept von Ordnern, bei denen es sich bei Buckets für allgemeine Zwecke im Wesentlichen um Präfixe handelt, die dem Objektschlüsselnamen vorangestellt werden. Weitere Informationen finden Sie unter Organisieren von Objekten in der Amazon S3-Konsole mithilfe von Ordnern.

- Tags — Jedes Tag ist ein Schlüssel-Wert-Paar, das Sie Ressourcen zuweisen. Sie können beispielsweise einige Ressourcen mit dem Tag kennzeichnen. `topicCategory=engineering` Sie können Tagging verwenden, um die Kostenzuweisung, Kategorisierung und Organisation sowie die Zugriffskontrolle zu erleichtern. Bucket-Tagging wird nur für die Kostenzuweisung verwendet. Sie können Objekte, S3 Storage Lens, Jobs und S3 Access Grants zu Organisationszwecken oder zur Zugriffskontrolle taggen. In S3 Access Grants können Sie Tagging auch für die Kostenzuweisung verwenden. Als Beispiel für die Steuerung des Zugriffs auf Ressourcen mithilfe ihrer Tags können Sie nur Objekte teilen, die über ein bestimmtes Tag oder eine Kombination von Tags verfügen.

  Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Steuern des Zugriffs auf AWS Ressourcen mithilfe von Ressourcen-Tags.

Identitäten

In Amazon S3 ist der Ressourcenbesitzer die Identität, die die Ressource erstellt hat, z. B. ein Bucket oder ein Objekt. Standardmäßig können nur der Root-Benutzer des Kontos, das die Ressource erstellt hat, und IAM Identitäten innerhalb des Kontos, die über die erforderlichen Berechtigungen verfügen, auf die S3-Ressource zugreifen. Ressourcenbesitzer können anderen Identitäten Zugriff auf ihre S3-Ressourcen gewähren.

Identitäten, denen eine Ressource nicht gehört, können Zugriff auf diese Ressource beantragen. Anfragen an eine Ressource sind entweder authentifiziert oder nicht authentifiziert. Authentifizierte Anfragen müssen einen Signaturwert enthalten, der den Absender der Anfrage authentifiziert. Für nicht authentifizierte Anfragen ist jedoch keine Signatur erforderlich. Wir empfehlen,

dass Sie nur authentifizierten Benutzern Zugriff gewähren. Weitere Informationen zur Anforderungsauthentifizierung finden Sie unter Anfragen stellen in der Amazon S3 API S3-Referenz.

> **Important**
>
> Wir empfehlen, dass Sie die AWS-Konto Root-Benutzeranmeldedaten nicht verwenden, um authentifizierte Anfragen zu stellen. Erstellen Sie stattdessen eine IAM Rolle und gewähren Sie dieser Rolle vollen Zugriff. Wir bezeichnen Benutzer mit dieser Rolle als Administratorbenutzer. Sie können die der Administratorrolle zugewiesenen Anmeldeinformationen anstelle von AWS-Konto Root-Benutzeranmeldedaten verwenden, um mit Aufgaben wie dem Erstellen eines Buckets, dem Erstellen von Benutzern AWS und dem Erteilen von Berechtigungen zu interagieren und diese auszuführen. Weitere Informationen finden Sie unter AWS-Konto Root-Benutzeranmeldedaten und IAM Benutzeranmeldedaten im Allgemeine AWS-Referenzund unter Bewährte Sicherheitsmethoden IAM im IAM Benutzerhandbuch.

Es kann sich um eine der folgenden Identitäten handeln, die auf Ihre Daten in Amazon S3 zugreifen:

AWS-Konto owner

AWS-Konto Derjenige, der die Ressource erstellt hat. Zum Beispiel das Konto, das den Bucket erstellt hat. Dieses Konto besitzt die Ressource. Weitere Informationen finden Sie unter Root-Benutzer für das AWS Konto.

IAMIdentitäten im selben Konto des Besitzers AWS-Konto

Beim Einrichten von Konten für neue Teammitglieder, die S3-Zugriff benötigen, kann der AWS-Konto Besitzer AWS Identity and Access Management (IAM) verwenden, um Benutzer, Gruppen und Rollen zu erstellen. Der AWS-Konto Besitzer kann dann Ressourcen mit diesen IAM Identitäten teilen. Der Kontoinhaber kann auch die Berechtigungen zur Vergabe der IAM Identitäten angeben, mit denen die Aktionen, die mit den gemeinsam genutzten Ressourcen ausgeführt werden können, zugelassen oder verweigert werden.

IAMIdentitäten bieten erweiterte Funktionen, einschließlich der Möglichkeit, von Benutzern die Eingabe von Anmeldeinformationen zu verlangen, bevor sie auf gemeinsam genutzte Ressourcen zugreifen können. Durch die Verwendung von IAM Identitäten können Sie eine Form der IAM Multi-Faktor-Authentifizierung (MFA) implementieren, um eine starke Identitätsbasis zu unterstützen. Eine IAM bewährte Methode besteht darin, Rollen für die Zugriffsverwaltung zu erstellen, anstatt jedem

einzelnen Benutzer Berechtigungen zu erteilen. Sie weisen einzelnen Benutzern die entsprechende Rolle zu. Weitere Informationen finden Sie unter Bewährte Sicherheitsmethoden unter IAM.

Andere AWS Kontoinhaber und ihre IAM Identitäten (kontoübergreifender Zugriff)

Der AWS-Konto Inhaber kann auch anderen AWS Kontoinhabern oder IAM Identitäten, die zu einem anderen AWS Konto gehören, Zugriff auf Ressourcen gewähren.

> **Note**
>
> Delegierung von Berechtigungen — Wenn ein Benutzer AWS-Konto Eigentümer einer Ressource ist, kann er diese Berechtigungen einem anderen AWS-Konto gewähren. Dieses Konto kann dann diese Berechtigungen oder einen Teil davon an Benutzer desselben Kontos delegieren. Dies wird auch als Berechtigungsdelegation bezeichnet. Ein Konto, das Berechtigungen von einem anderen Konto erhält, kann diese Berechtigungen jedoch nicht „kontoübergreifend“ an ein anderes delegieren. AWS-Konto

Anonyme Benutzer (öffentlicher Zugriff)

Der AWS-Konto Besitzer kann Ressourcen öffentlich machen. Wenn eine Ressource veröffentlicht wird, wird die Ressource technisch gesehen mit dem anonymen Benutzer geteilt. Buckets, die seit April 2023 erstellt wurden, blockieren standardmäßig den gesamten öffentlichen Zugriff, sofern Sie diese Einstellung nicht ändern. Wir empfehlen, dass Sie Ihre Buckets so einrichten, dass sie den öffentlichen Zugriff blockieren, und dass Sie nur authentifizierten Benutzern Zugriff gewähren. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

AWS-Services

Der Ressourcenbesitzer kann einem anderen AWS Service Zugriff auf eine Amazon S3 S3-Ressource gewähren. Sie können dem AWS CloudTrail Service beispielsweise die `s3:PutObject` Erlaubnis erteilen, Protokolldateien in Ihren Bucket zu schreiben. Weitere Informationen finden Sie unter Zugriff auf einen AWS Dienst bereitstellen.

Identitäten im Unternehmensverzeichnis

Der Eigentümer der Ressource kann Benutzern oder Rollen aus Ihrem Unternehmensverzeichnis mithilfe von S3 Access Grants Zugriff auf eine S3-Ressource gewähren. Weitere Informationen zum

Hinzufügen Ihres Unternehmensverzeichnisses finden Sie AWS IAM Identity Center unter Was ist IAM Identity Center? .

Besitzer von Buckets oder Ressourcen

Derjenige AWS-Konto , den Sie verwenden, um Buckets zu erstellen und Objekte hochzuladen, besitzt diese Ressourcen. Ein Bucket-Eigentümer kann einem anderen AWS-Konto (oder Benutzern in einem anderen Konto) kontenübergreifende Berechtigungen erteilen, um Objekte hochzuladen.

Wenn ein Bucket-Besitzer einem anderen Konto erlaubt, Objekte in einen Bucket hochzuladen, besitzt standardmäßig der Bucket-Besitzer alle Objekte, die in seinen Bucket hochgeladen wurden. Wenn jedoch sowohl die Einstellungen für den Bucket Owner erzwungen als auch den AWS-Konto Bucket Preferred Owner deaktiviert sind, besitzt derjenige, der die Objekte hochlädt, diese Objekte, und der Bucket-Besitzer hat keine Berechtigungen für die Objekte, die einem anderen Konto gehören, mit den folgenden Ausnahmen:

- Der Bucket-Eigentümer zahlt die Rechnungen. Ein Bucket-Eigentümer kann explizit den Zugriff auf Objekte verweigern oder Objekte im Bucket löschen, unabhängig davon, wem sie gehören.
- Der Bucket-Besitzer kann beliebige Objekte archivieren oder archivierte Objekte wiederherstellen, unabhängig davon, wem sie gehören. Die Archivierung bezieht sich auf die Speicherklasse, die beim Speichern der Objekte verwendet wurde. Weitere Informationen finden Sie unter Verwaltung des Lebenszyklus von Objekten.

Tools zur Zugriffsverwaltung

Amazon S3 enthält eine Vielzahl von Sicherheitsfunktionen und -tools. Im Folgenden finden Sie eine umfassende Liste dieser Funktionen und Tools. Sie benötigen nicht alle diese Tools zur Zugriffsverwaltung, aber Sie müssen eines oder mehrere verwenden, um Zugriff auf Ihre Amazon S3 S3-Ressourcen zu gewähren. Durch die richtige Anwendung dieser Tools kann sichergestellt werden, dass Ihre Ressourcen nur den vorgesehenen Benutzern zugänglich sind.

Das am häufigsten verwendete Zugriffsverwaltungstool ist eine Zugriffsrichtlinie. Eine Zugriffsrichtlinie kann eine ressourcenbasierte Richtlinie sein, die an eine AWS Ressource angehängt ist, z. B. eine Bucket-Richtlinie für einen Bucket. Eine Zugriffsrichtlinie kann auch eine identitätsbasierte Richtlinie sein, die mit einer AWS Identity and Access Management (IAM) Identität verknüpft ist, z. B. einem IAM Benutzer, einer Gruppe oder einer Rolle. Schreiben Sie eine Zugriffsrichtlinie, um IAM Benutzern, Gruppen AWS-Konten und Rollen die Erlaubnis zu erteilen, Operationen an

einer Ressource auszuführen. Sie können beispielsweise einem anderen Konto die `PUT Object` Erlaubnis erteilen, AWS-Konto sodass das andere Konto Objekte in Ihren Bucket hochladen kann.

Eine Zugriffsrichtlinie beschreibt, wer Zugriff auf welche Dinge hat. Wenn Amazon S3 eine Anfrage erhält, muss es alle Zugriffsrichtlinien auswerten, um zu entscheiden, ob die Anfrage autorisiert oder abgelehnt werden soll. Weitere Informationen dazu, wie Amazon S3 diese Richtlinien auswertet, finden Sie unter Wie Amazon S3 eine Anforderung autorisiert.

Im Folgenden sind die in Amazon S3 verfügbaren Zugriffsverwaltungstools aufgeführt.

Bucket-Richtlinie

Eine Amazon S3 S3-Bucket-Richtlinie ist eine JSON -formatierte AWS Identity and Access Management (IAM) ressourcenbasierte Richtlinie, die an einen bestimmten Bucket angehängt ist. Verwenden Sie Bucket-Richtlinien, um anderen Benutzern AWS-Konten oder IAM Identitäten Berechtigungen für den Bucket und die darin enthaltenen Objekte zu gewähren. Viele Anwendungsfälle für die S3-Zugriffsverwaltung können durch die Verwendung einer Bucket-Richtlinie erfüllt werden. Mit Bucket-Richtlinien können Sie den Bucket-Zugriff personalisieren, um sicherzustellen, dass nur die von Ihnen genehmigten Identitäten auf Ressourcen zugreifen und darin Aktionen ausführen können. Weitere Informationen finden Sie unter Bucket-Richtlinien für Amazon S3.

Hier finden Sie ein Beispiel für eine Bucket-Richtlinie. Sie drücken die Bucket-Richtlinie mithilfe einer JSON Datei aus. Diese Beispielrichtlinie gewährt einer IAM Rolle Leseberechtigung für alle Objekte im Bucket. Sie enthält eine Anweisung mit dem Namen`BucketLevelReadPermissions`, die die `s3:GetObject` Aktion (Leseberechtigung) für Objekte in einem Bucket mit dem Namen ermöglicht`amzn-s3-demo-bucket1`. Durch die `Principal` Angabe einer IAM Rolle als „gewährt diese Richtlinie jedem IAM Benutzer mit dieser Rolle Zugriff. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre eigenen Informationen.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid":"BucketLevelReadPermissions",
      "Effect":"Allow",
      "Principal": {
     "AWS": "arn:aws:iam::123456789101:role/s3-role"
      },
      "Action":["s3:GetObject"],
```

```
      "Resource":["arn:aws:s3:::amzn-s3-demo-bucket1/*"]
    }]
  }
```

> **Note**
>
> Vermeiden Sie beim Erstellen von Richtlinien die Verwendung von Platzhalterzeichen (*) im `Principal`-Element, da ein Platzhalterzeichen allen Benutzer den Zugriff auf Ihre Amazon-S3-Ressourcen ermöglicht. Führen Sie stattdessen explizit Benutzer oder Gruppen auf, die auf den Bucket zugreifen dürfen, oder führen Sie mithilfe einer Bedingungsklausel in der Richtlinie Bedingungen auf, die erfüllt werden müssen. Erteilen Sie Ihren Benutzern oder Gruppen außerdem gegebenenfalls spezifische Berechtigungen, anstatt Platzhalterzeichen für die Aktionen zu verwenden.

Identitätsbasierte Richtlinien

Eine identitätsbasierte Richtlinie oder IAM Benutzerrichtlinie ist eine Art von Richtlinie AWS Identity and Access Management () IAM. Eine identitätsbasierte Richtlinie ist eine JSON -formatierte Richtlinie, die IAM Benutzern, Gruppen oder Rollen in Ihrem Konto zugeordnet ist. AWS Sie können identitätsbasierte Richtlinien verwenden, um einer IAM Identität Zugriff auf Ihre Buckets oder Objekte zu gewähren. Sie können IAM Benutzer, Gruppen und Rollen in Ihrem Konto erstellen und ihnen Zugriffsrichtlinien zuordnen. Sie können dann Zugriff auf AWS Ressourcen gewähren, einschließlich Amazon S3 S3-Ressourcen. Weitere Informationen finden Sie unter Identitätsbasierte Richtlinien für Amazon S3.

Im Folgenden finden Sie ein Beispiel für eine identitätsbasierte Richtlinie. Die Beispielrichtlinie ermöglicht es der zugehörigen IAM Rolle, sechs verschiedene Amazon S3 S3-Aktionen (Berechtigungen) für einen Bucket und die darin enthaltenen Objekte auszuführen. Wenn Sie diese Richtlinie einer IAM Rolle in Ihrem Konto zuordnen und die Rolle einigen Ihrer IAM Benutzer zuweisen, können die Benutzer mit dieser Rolle diese Aktionen mit den in Ihrer Richtlinie angegebenen Ressourcen (Buckets) ausführen. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre eigenen Informationen.

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

```
    "Sid": "AssignARoleActions",
    "Effect": "Allow",
    "Action": [
   "s3:PutObject",
   "s3:GetObject",
   "s3:ListBucket",
   "s3:DeleteObject",
   "s3:GetBucketLocation"
    ],
    "Resource": [
   "arn:aws:s3:::amzn-s3-demo-bucket1/*",
   "arn:aws:s3:::amzn-s3-demo-bucket1"
    ]
    },
  {
    "Sid": "AssignARoleActions2",
    "Effect": "Allow",
    "Action": "s3:ListAllMyBuckets",
    "Resource": "*"
  }
  ]
}
```

S3 Access Grants

Verwenden Sie S3 Access Grants, um Zugriffsberechtigungen für Ihre Amazon S3 S3-Daten für beide Identitäten in Unternehmensidentitätsverzeichnissen zu erstellen, z. B. Active Directory, und zu AWS Identity and Access Management (IAM) Identitäten. S3 Access Grants unterstützt Sie bei der Verwaltung von Datenberechtigungen in großem Umfang. Darüber hinaus protokolliert S3 Access Grants die Identität des Endbenutzers und die Anwendung, die für den Zugriff auf die S3-Daten verwendet wurde. AWS CloudTrail Dies bietet einen detaillierten Auditverlauf bis hin zur Endbenutzeridentität für den gesamten Zugriff auf die Daten in Ihren S3-Buckets. Weitere Informationen finden Sie unter Verwalten des Zugriffs mit S3-Zugriffsberechtigungen.

Zugriffspunkte

Amazon S3 Access Points vereinfacht die Verwaltung des Datenzugriffs in großem Umfang für Anwendungen, die gemeinsam genutzte Datensätze auf S3 verwenden. Access Points werden als Netzwerkendpunkte bezeichnet, die an einen Bucket angeschlossen sind. Sie können Access Points verwenden, um S3-Objektoperationen in großem Umfang durchzuführen, wie z. B. das Hochladen und Abrufen von Objekten. An einen Bucket können bis zu 10.000 Access Points angeschlossen

sein, und für jeden Access Point können Sie eigene Berechtigungen und Netzwerkkontrollen durchsetzen, um den Zugriff auf Ihre S3-Objekte detailliert zu kontrollieren. S3 Access Points können Buckets im selben Konto oder in einem anderen vertrauenswürdigen Konto zugeordnet werden. Access Point-Richtlinien sind ressourcenbasierte Richtlinien, die in Verbindung mit der zugrunde liegenden Bucket-Richtlinie bewertet werden. Weitere Informationen finden Sie unter Verwaltung des Zugriffs auf gemeinsam genutzte Datensätze mit Access Points.

Liste der Zugriffskontrollen () ACL

An ACL ist eine Liste von Zuschüssen, in der der Empfänger und die erteilte Genehmigung identifiziert werden. ACLsanderen AWS-Konten Personen grundlegende Lese- oder Schreibberechtigungen gewähren. ACLsverwenden Sie ein Amazon S3 S3-spezifisches SchemaXML. An ACL ist eine Art von Richtlinie AWS Identity and Access Management () IAM. Ein Objekt ACL wird verwendet, um den Zugriff auf ein Objekt zu verwalten, und ein Bucket ACL wird verwendet, um den Zugriff auf einen Bucket zu verwalten. Bei Bucket-Richtlinien gibt es eine einzige Richtlinie für den gesamten Bucket, aber Objekte ACLs werden für jedes Objekt angegeben. Es wird empfohlen, die Option ACLs ausgeschaltet zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Weitere Informationen zur Verwendung finden ACLs Sie unterKontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

> **Warning**
>
> Die meisten modernen Anwendungsfälle in Amazon S3 erfordern nicht die Verwendung vonACLs.

Im Folgenden finden Sie ein Beispiel für einen BucketACL. Der Grant im ACL zeigt einen Bucket-Besitzer, der über Vollzugriff verfügt.

```
<?xml version="1.0" encoding="UTF-8"?>
 <AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
   <ID>Owner-Canonical-User-ID</ID>
   <DisplayName>owner-display-name</DisplayName>
  </Owner>
 <AccessControlList>
  <Grant>
   <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Canonical
 User">
    <ID>Owner-Canonical-User-ID</ID>
```

```
    <DisplayName>display-name</DisplayName>
   </Grantee>
   <Permission>FULL_CONTROL</Permission>
  </Grant>
 </AccessControlList>
</AccessControlPolicy>
```

Eigentum am Objekt

Um den Zugriff auf Ihre Objekte zu verwalten, müssen Sie der Eigentümer des Objekts sein. Sie können die Einstellung „Objekteigentum“ auf Bucketebene verwenden, um den Besitz von Objekten zu kontrollieren, die in Ihren Bucket hochgeladen wurden. Verwenden Sie auch „Objekteigentum“, um die Option zu aktivieren. ACLs Standardmäßig ist für Object Ownership die Einstellung Bucket Owner erforced festgelegt und alle Optionen ACLs sind deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet ausschließlich den Zugriff auf Daten. Um den Zugriff zu verwalten, verwendet der Bucket-Besitzer Richtlinien oder ein anderes Zugriffsverwaltungstool, ausgenommenACLs. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

Object Ownership verfügt über drei Einstellungen, mit denen Sie sowohl den Besitz von Objekten kontrollieren können, die in Ihren Bucket hochgeladen werden, als auch um sie zu aktivierenACLs:

ACLsausgeschaltet

- Bucket-Besitzer erzwungen (Standard) — ACLs sind ausgeschaltet, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLswirken sich nicht auf die Berechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet ausschließlich Richtlinien, um die Zugriffssteuerung zu definieren.

ACLseingeschaltet

- Bevorzugter Bucket-Besitzer — Der Bucket-Besitzer besitzt und hat die volle Kontrolle über neue Objekte, die andere Accounts in den Bucket mit den gespeicherten Objekten `bucket-owner-full-control` schreibenACL.
- Objekt-Writer — AWS-Konto Derjenige, der ein Objekt hochlädt, besitzt das Objekt, hat die volle Kontrolle darüber und kann anderen Benutzern Zugriff darauf gewähren. ACLs

Weitere bewährte Methoden

Erwägen Sie die Verwendung der folgenden Bucket-Einstellungen und Tools, um Daten während der Übertragung und Speicherung zu schützen. Beide sind entscheidend für die Aufrechterhaltung der Integrität und Zugänglichkeit Ihrer Daten:

- Öffentlichen Zugriff blockieren — Deaktivieren Sie nicht die Standardeinstellung „Öffentlichen Zugriff blockieren“ auf Bucket-Ebene. Diese Einstellung blockiert standardmäßig den öffentlichen Zugriff auf Ihre Daten. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.
- S3-Versionierung — Aus Gründen der Datenintegrität können Sie die S3-Versioning-Bucket-Einstellung implementieren, mit der Ihre Objekte bei Aktualisierungen versioniert werden, anstatt sie zu überschreiben. Sie können S3 Versioning verwenden, um bei Bedarf eine frühere Version beizubehalten, abzurufen und wiederherzustellen. Weitere Informationen über S3 Versioning finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.
- S3 Object Lock — S3 Object Lock ist eine weitere Einstellung, die Sie implementieren können, um Datenintegrität zu erreichen. Diese Funktion kann ein write-once-read-many (WORM) -Modell implementieren, um Objekte unveränderlich zu speichern. Weitere Informationen über die Objektsperre finden Sie unter Sperren von Objekten mit Object Lock.
- Objektverschlüsselung — Amazon S3 bietet mehrere Objektverschlüsselungsoptionen, die Daten während der Übertragung und Speicherung schützen. Die serverseitige Verschlüsselung verschlüsselt Ihr Objekt, bevor es auf Festplatten in seinen Rechenzentren gespeichert wird, und entschlüsselt es dann, wenn Sie die Objekte herunterladen. Wenn Sie Ihre Anfrage authentifizieren und über Zugriffsberechtigungen verfügen, gibt es keinen Unterschied in der Art und Weise, wie Sie auf verschlüsselte oder unverschlüsselte Objekte zugreifen. Weitere Informationen finden Sie unter Schützen von Daten mit serverseitiger Verschlüsselung. S3 verschlüsselt neu hochgeladene Objekte standardmäßig. Weitere Informationen finden Sie unter Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets. Client-seitige Verschlüsselung ist der Vorgang, bei dem Daten verschlüsselt werden, bevor sie an Amazon S3 gesendet werden. Weitere Informationen finden Sie unter Schützen von Daten mithilfe der clientseitigen Verschlüsselung.
- Signaturmethoden — Bei Signature Version 4 werden Authentifizierungsinformationen zu AWS Anfragen hinzugefügt, die von HTTP gesendet wurden. Aus Sicherheitsgründen AWS müssen die meisten Anfragen mit einem Zugriffsschlüssel signiert werden, der aus einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel besteht. Diese beiden Schlüssel werden in der Regel als Sicherheitsanmeldeinformationen bezeichnet. Weitere Informationen finden Sie unter Authentifizieren von Anfragen (AWS Signature Version 4) und Signaturprozess für Signature Version 4.

Aktionen

Eine vollständige Liste der S3-Berechtigungen und Bedingungsschlüssel finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

Aktionen

Die AWS Identity and Access Management (IAM) -Aktionen für Amazon S3 sind die möglichen Aktionen, die für einen S3-Bucket oder ein S3-Objekt ausgeführt werden können. Sie weisen diese Aktionen Identitäten zu, damit sie auf Ihre S3-Ressourcen einwirken können. Beispiele für S3-Aktionen sind `s3:GetObject` das Lesen von Objekten in einem Bucket und `s3:PutObject` das Schreiben von Objekten in einen Bucket.

Bedingungsschlüssel

Zusätzlich zu Aktionen sind IAM Bedingungsschlüssel darauf beschränkt, Zugriff nur dann zu gewähren, wenn eine Bedingung erfüllt ist. Bedingungsschlüssel sind optional.

> **Note**
>
> In einer ressourcenbasierten Zugriffsrichtlinie, z. B. einer Bucket-Richtlinie, oder in einer identitätsbasierten Richtlinie können Sie Folgendes angeben:
>
> - Eine Aktion oder eine Reihe von Aktionen im `Action` Element der Grundsatzerklärung.
> - Im `Effect` Element der Grundsatzerklärung können Sie angeben, ob die aufgelisteten Aktionen gewährt werden `Allow` sollen, oder Sie können angeben, `Deny` dass die aufgelisteten Aktionen blockiert werden sollen. Um die Praxis der geringsten Rechte weiter aufrechtzuerhalten, sollten die `Deny` Aussagen im `Effect` Element der Zugangsrichtlinie so weit wie möglich und die `Allow` Aussagen so eng wie möglich sein. `Deny`Die mit der `s3:*` Maßnahme verbundenen Effekte sind eine weitere gute Möglichkeit, bewährte Verfahren für die Zustimmung zu den Identitäten umzusetzen, die in den Erklärungen zu den politischen Bedingungen enthalten sind.
> - Ein Bedingungsschlüssel im `Condition` Element einer Grundsatzerklärung.

Anwendungsfälle für die Zugriffsverwaltung

Amazon S3 bietet Ressourcenbesitzern eine Vielzahl von Tools, mit denen sie Zugriff gewähren können. Welches S3-Zugriffsverwaltungstool Sie verwenden, hängt von den S3-Ressourcen ab, die Sie teilen möchten, den Identitäten, denen Sie Zugriff gewähren, und den Aktionen, die Sie zulassen oder verweigern möchten. Möglicherweise möchten Sie eines oder eine Kombination von S3-Zugriffsverwaltungstools verwenden, um den Zugriff auf Ihre S3-Ressourcen zu verwalten.

In den meisten Fällen können Sie eine Zugriffsrichtlinie verwenden, um Berechtigungen zu verwalten. Eine Zugriffsrichtlinie kann eine ressourcenbasierte Richtlinie sein, die an eine Ressource, z. B. einen Bucket, oder eine andere Amazon S3 S3-Ressource () S3-Ressourcen angehängt ist. Eine Zugriffsrichtlinie kann auch eine identitätsbasierte Richtlinie sein, die einem AWS Identity and Access Management (IAM) Benutzer, einer Gruppe oder einer Rolle in Ihrem Konto zugeordnet ist. Möglicherweise stellen Sie fest, dass eine Bucket-Richtlinie für Ihren Anwendungsfall besser geeignet ist. Weitere Informationen finden Sie unter Bucket-Richtlinien für Amazon S3. Alternativ können Sie mit AWS Identity and Access Management (IAM) IAM Benutzer, Gruppen und Rollen innerhalb Ihres eigenen erstellen und deren Zugriff auf Buckets AWS-Konto und Objekte mithilfe identitätsbasierter Richtlinien verwalten. Weitere Informationen finden Sie unter Identitätsbasierte Richtlinien für Amazon S3.

Um Ihnen den Umgang mit diesen Zugriffsverwaltungsoptionen zu erleichtern, finden Sie im Folgenden allgemeine Anwendungsfälle und Empfehlungen für die einzelnen S3-Zugriffsverwaltungstools von Amazon S3 S3-Kunden.

Der AWS-Konto Besitzer möchte Buckets nur mit Benutzern innerhalb desselben Kontos teilen

Alle Tools zur Zugriffsverwaltung können diesen grundlegenden Anwendungsfall erfüllen. Für diesen Anwendungsfall empfehlen wir die folgenden Tools zur Zugriffsverwaltung:

- Bucket-Richtlinie — Wenn Sie Zugriff auf einen Bucket oder eine kleine Anzahl von Buckets gewähren möchten oder wenn Ihre Bucket-Zugriffsberechtigungen von Bucket zu Bucket ähnlich sind, verwenden Sie eine Bucket-Richtlinie. Mit Bucket-Richtlinien verwalten Sie eine Richtlinie für jeden Bucket. Weitere Informationen finden Sie unter Bucket-Richtlinien für Amazon S3.
- Identitätsbasierte Richtlinie — Wenn Sie eine sehr große Anzahl von Buckets mit unterschiedlichen Zugriffsberechtigungen für jeden Bucket haben und nur wenige Benutzerrollen verwalten müssen, können Sie eine IAM Richtlinie für Benutzer, Gruppen oder Rollen verwenden. IAMRichtlinien sind auch eine gute Option, wenn Sie den Benutzerzugriff auf andere AWS Ressourcen sowie auf

Amazon S3 S3-Ressourcen verwalten. Weitere Informationen finden Sie unter Beispiel 1: Bucket-Eigentümer erteilt seinen Benutzern Bucket-Berechtigungen.

- S3 Access Grants — Sie können S3 Access Grants verwenden, um Zugriff auf Ihre S3-Buckets, -Präfixe oder Objekte zu gewähren. Mit S3 Access Grants können Sie unterschiedliche Berechtigungen auf Objektebene nach Bedarf festlegen. Bucket-Richtlinien sind dagegen auf eine Größe von 20 KB begrenzt. Weitere Informationen finden Sie unter Erste Schritte mit S3 Access Grants.
- Access Points — Sie können Access Points verwenden, bei denen es sich um so genannte Netzwerkendpunkte handelt, die an einen Bucket angeschlossen sind. An einen Bucket können bis zu 10.000 Access Points angeschlossen sein, und für jeden Access Point können Sie eigene Berechtigungen und Netzwerkkontrollen durchsetzen, um Ihnen eine detaillierte Kontrolle über den Zugriff auf Ihre S3-Objekte zu geben. Weitere Informationen finden Sie unter Verwaltung des Zugriffs auf gemeinsam genutzte Datensätze mit Access Points.

Der AWS-Konto Besitzer möchte Buckets oder Objekte mit Benutzern aus einem anderen AWS Konto teilen (kontoübergreifend)

Um einem anderen Benutzer die Erlaubnis zu erteilen AWS-Konto, müssen Sie eine Bucket-Richtlinie oder eines der folgenden empfohlenen Zugriffsverwaltungstools verwenden. Für diesen Anwendungsfall können Sie keine identitätsbasierte Zugriffsrichtlinie verwenden. Weitere Informationen zur Gewährung von kontenübergreifendem Zugriff finden Sie unter Wie stelle ich kontenübergreifenden Zugriff auf Objekte bereit, die sich in Amazon S3 S3-Buckets befinden?

Für diesen Anwendungsfall empfehlen wir die folgenden Tools zur Zugriffsverwaltung:

- Bucket-Richtlinie — Mit Bucket-Richtlinien verwalten Sie eine Richtlinie für jeden Bucket. Weitere Informationen finden Sie unter Bucket-Richtlinien für Amazon S3.
- S3 Access Grants — Sie können S3 Access Grants verwenden, um kontoübergreifende Berechtigungen für Ihre S3-Buckets, -Präfixe oder Objekte zu gewähren. Sie können S3 Access Grants verwenden, um unterschiedliche Berechtigungen auf Objektebene nach Bedarf festzulegen. Bucket-Richtlinien sind dagegen auf eine Größe von 20 KB begrenzt. Weitere Informationen finden Sie unter Erste Schritte mit S3 Access Grants.
- Access Points — Sie können Access Points verwenden, bei denen es sich um so genannte Netzwerkendpunkte handelt, die an einen Bucket angeschlossen sind. An einen Bucket können bis zu 10.000 Access Points angeschlossen sein, und für jeden Access Point können Sie eigene Berechtigungen und Netzwerkkontrollen durchsetzen, um Ihnen eine detaillierte Kontrolle über

den Zugriff auf Ihre S3-Objekte zu geben. Weitere Informationen finden Sie unter Verwaltung des Zugriffs auf gemeinsam genutzte Datensätze mit Access Points.

Der AWS-Konto Besitzer oder Bucket-Besitzer muss Berechtigungen auf Objekt- oder Präfix-Ebene gewähren, und diese Berechtigungen variieren von Objekt zu Objekt oder von Präfix zu Präfix

In einer Bucket-Richtlinie können Sie beispielsweise Zugriff auf die Objekte innerhalb eines Buckets gewähren, die ein bestimmtes Schlüsselnamenpräfix oder ein bestimmtes Tag haben. Sie können Leseberechtigungen für Objekte gewähren, die mit dem Schlüsselnamenpräfix beginnen`logs/`. Wenn Ihre Zugriffsberechtigungen jedoch je nach Objekt variieren, ist es möglicherweise nicht praktikabel, einzelnen Objekten mithilfe einer Bucket-Richtlinie Berechtigungen zu gewähren, zumal Bucket-Richtlinien auf eine Größe von 20 KB begrenzt sind.

Für diesen Anwendungsfall empfehlen wir die folgenden Tools zur Zugriffsverwaltung:

- S3 Access Grants — Sie können S3 Access Grants verwenden, um Berechtigungen auf Objekt- oder Präfix-Ebene zu verwalten. Im Gegensatz zu Bucket-Richtlinien können Sie S3 Access Grants verwenden, um unterschiedliche Berechtigungen auf Objektebene nach Bedarf festzulegen. Bucket-Richtlinien sind auf eine Größe von 20 KB beschränkt. Weitere Informationen finden Sie unter Erste Schritte mit S3 Access Grants.
- Access Points — Sie können Access Points verwenden, um Berechtigungen auf Objekt- oder Präfix-Ebene zu verwalten. Access Points werden als Netzwerkendpunkte bezeichnet, die an einen Bucket angeschlossen sind. An einen Bucket können bis zu 10.000 Access Points angeschlossen sein, und für jeden Access Point können Sie eigene Berechtigungen und Netzwerkkontrollen durchsetzen, um eine detaillierte Kontrolle über den Zugriff auf Ihre S3-Objekte zu erhalten. Weitere Informationen finden Sie unter Verwaltung des Zugriffs auf gemeinsam genutzte Datensätze mit Access Points.
- ACLs— Wir raten davon ab, Zugriffskontrolllisten (ACLs) zu verwenden, vor allem, weil sie auf 100 Grants pro Objekt begrenzt ACLs sind. Wenn Sie sich jedoch dafür entscheiden, diese Option zu aktivierenACLs, setzen Sie in Ihren Bucket-Einstellungen die Option Objekteigentum auf „Bevorzugter und ACLs aktivierter Bucket Owner“. Mit dieser Einstellung gehören neue Objekte, die mit dem gespeicherten Objekt `bucket-owner-full-control` geschrieben ACL werden, automatisch dem Bucket-Besitzer und nicht dem Objektschreiber. Sie können dann Objekt verwendenACLs, bei dem es sich um eine Zugriffsrichtlinie im XML -Format handelt, um anderen

Benutzern Zugriff auf das Objekt zu gewähren. Weitere Informationen finden Sie unter Übersicht über die Zugriffskontrollliste (ACL).

Der AWS-Konto Besitzer oder Bucket-Besitzer möchte den Bucket-Zugriff nur auf ein bestimmtes Konto beschränken IDs

Für diesen Anwendungsfall empfehlen wir die folgenden Tools zur Zugriffsverwaltung:

- Bucket-Richtlinie — Mit Bucket-Richtlinien verwalten Sie eine Richtlinie für jeden Bucket. Weitere Informationen finden Sie unter Bucket-Richtlinien für Amazon S3.
- Access Points — Access Points werden als Netzwerkendpunkte bezeichnet, die an einen Bucket angeschlossen sind. An einen Bucket können bis zu 10.000 Access Points angeschlossen sein, und für jeden Access Point können Sie eigene Berechtigungen und Netzwerkkontrollen durchsetzen, um eine detaillierte Kontrolle über den Zugriff auf Ihre S3-Objekte zu erhalten. Weitere Informationen finden Sie unter Verwaltung des Zugriffs auf gemeinsam genutzte Datensätze mit Access Points.

Der AWS-Konto Besitzer oder Bucket-Besitzer möchte für jeden Benutzer oder jede Anwendung, die auf seine Daten zugreift, eigene Endpunkte

Für diesen Anwendungsfall empfehlen wir das folgende Zugriffsverwaltungstool:

- Access Points — Access Points werden als Netzwerkendpunkte bezeichnet, die an einen Bucket angeschlossen sind. An einen Bucket können bis zu 10.000 Access Points angeschlossen sein, und für jeden Access Point können Sie eigene Berechtigungen und Netzwerkkontrollen durchsetzen, um eine detaillierte Kontrolle über den Zugriff auf Ihre S3-Objekte zu erhalten. Jeder Zugriffspunkt erzwingt eine benutzerdefinierte Zugriffspunktrichtlinie, die in Verbindung mit der Bucket-Richtlinie funktioniert, die dem zugrunde liegenden Bucket zugeordnet ist. Weitere Informationen finden Sie unter Verwaltung des Zugriffs auf gemeinsam genutzte Datensätze mit Access Points.

Der AWS-Konto Besitzer oder Bucket-Besitzer muss den Zugriff von Virtual Private Cloud (VPC) -Endpunkten für S3 verwalten

Virtual Private Cloud (VPC) -Endpunkte für Amazon S3 sind logische Entitäten innerhalb einerVPC, die nur Konnektivität zu S3 ermöglichen. Für diesen Anwendungsfall empfehlen wir die folgenden Tools zur Zugriffsverwaltung:

- Buckets in einer VPC Einstellung — Sie können eine Bucket-Richtlinie verwenden, um zu kontrollieren, wer auf Ihre Buckets zugreifen darf und auf welche VPC Endpunkte sie zugreifen können. Weitere Informationen finden Sie unter Steuern des Zugriffs von VPC Endpunkten aus mit Bucket-Richtlinien.
- Access Points — Wenn Sie Access Points einrichten möchten, können Sie eine Zugriffspunkt-Richtlinie verwenden. Sie können jeden Access Point so konfigurieren, dass er nur Anfragen von einer virtuellen privaten Cloud (VPC) akzeptiert, um den Amazon S3 S3-Datenzugriff auf ein privates Netzwerk einzuschränken. Sie können auch benutzerdefinierte Block Public Access-Einstellungen für jeden Zugriffspunkt konfigurieren. Weitere Informationen finden Sie unter Verwaltung des Zugriffs auf gemeinsam genutzte Datensätze mit Access Points.

Der AWS-Konto Besitzer oder Bucket-Besitzer muss eine statische Website öffentlich zugänglich machen

Mit S3 können Sie eine statische Website hosten und jedem ermöglichen, den Inhalt der Website anzusehen, die von einem S3-Bucket aus gehostet wird.

Für diesen Anwendungsfall empfehlen wir die folgenden Tools zur Zugriffsverwaltung:

- Amazon CloudFront — Mit dieser Lösung können Sie eine statische Amazon S3 S3-Website für die Öffentlichkeit hosten und gleichzeitig den gesamten öffentlichen Zugriff auf den Inhalt eines Buckets blockieren. Wenn Sie alle vier S3 Block Public Access-Einstellungen aktiviert lassen und eine statische S3-Website hosten möchten, können Sie Amazon CloudFront Origin Access Control (OAC) verwenden. Amazon CloudFront bietet die Funktionen, die für die Einrichtung einer sicheren statischen Website erforderlich sind. Außerdem können statische Amazon S3 S3-Websites, die diese Lösung nicht verwenden, nur HTTP Endgeräte unterstützen. CloudFront verwendet den dauerhaften Speicher von Amazon S3 und bietet gleichzeitig zusätzliche Sicherheitsheader, wie HTTPS z. HTTPSerhöht die Sicherheit, indem es eine normale HTTP Anfrage verschlüsselt und vor gängigen Cyberangriffen schützt.

Weitere Informationen finden Sie unter Erste Schritte mit einer sicheren statischen Website im Amazon CloudFront Developer Guide.

- Ihren Amazon S3 S3-Bucket öffentlich zugänglich machen — Sie können einen Bucket so konfigurieren, dass er als öffentlich zugängliche statische Website verwendet wird.

  > **Warning**
  >
  > Wir empfehlen diese Methode nicht. Stattdessen empfehlen wir Ihnen, statische Amazon S3 S3-Websites als Teil von Amazon zu verwenden CloudFront. Weitere Informationen finden Sie in der vorherigen Option oder unter Erste Schritte mit einer sicheren statischen Website.

  Um eine statische Amazon S3 S3-Website ohne Amazon CloudFront zu erstellen, müssen Sie zunächst alle Block Public Access-Einstellungen deaktivieren. Wenn Sie die Bucket-Richtlinie für Ihre statische Website schreiben, achten Sie darauf, nur `s3:GetObject`-Aktionen und keine `ListObject`- oder `PutObject`-Berechtigungen zuzulassen. Dadurch wird sichergestellt, dass Benutzer nicht alle Objekte in Ihrem Bucket anzeigen oder eigene Inhalte hinzufügen können. Weitere Informationen finden Sie unter Festlegen von Berechtigungen für den Website-Zugriff.

Der AWS-Konto Besitzer oder Bucket-Besitzer möchte den Inhalt eines Buckets öffentlich zugänglich machen

Wenn Sie einen neuen Amazon S3 S3-Bucket erstellen, ist die Einstellung Block Public Access standardmäßig aktiviert. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

Wir empfehlen nicht, öffentlichen Zugriff auf Ihren Bucket zu gewähren. Wenn Sie dies jedoch für einen bestimmten Anwendungsfall tun müssen, empfehlen wir das folgende Zugriffsverwaltungstool für diesen Anwendungsfall:

- Einstellung „Öffentlichen Zugriff blockieren“ deaktivieren — Ein Bucket-Besitzer kann nicht authentifizierte Anfragen an den Bucket zulassen. Nicht authentifizierte PUTObjektanforderungen sind beispielsweise zulässig, wenn für einen Bucket eine öffentliche Bucket-Richtlinie gilt oder wenn ein Bucket öffentlichen Zugriff gewährt. ACL Alle nicht authentifizierten Anfragen werden von anderen beliebigen AWS Benutzern oder sogar von nicht authentifizierten, anonymen Benutzern gestellt. Dieser Benutzer wird durch die spezifische ACLs kanonische Benutzer-ID repräsentiert.

`65a011a29cdf8ec533ec3d1ccaae921c` Wenn ein Objekt in ein `WRITE` oder hochgeladen wird`FULL_CONTROL`, gewährt dies ausdrücklich Zugriff auf die Gruppe „Alle Benutzer“ oder den anonymen Benutzer. Weitere Informationen zu öffentlichen Bucket-Richtlinien und öffentlichen Zugriffskontrolllisten (ACLs) finden Sie unterDie Bedeutung von „öffentlich“.

Der AWS-Konto Besitzer oder Bucket-Besitzer hat die Größenbeschränkungen der Zugriffsrichtlinie überschritten

Sowohl für Bucket-Richtlinien als auch für identitätsbasierte Richtlinien gilt eine Größenbeschränkung von 20 KB. Wenn Ihre Anforderungen an Zugriffsberechtigungen komplex sind, überschreiten Sie möglicherweise diese Größenbeschränkung.

Wir haben für diesen Anwendungsfall die folgenden Tools zur Zugriffsverwaltung empfohlen:

- Access Points — Verwenden Sie Access Points, wenn dies für Ihren Anwendungsfall funktioniert. Bei Access Points hat jeder Bucket mehrere benannte Netzwerkendpunkte, von denen jeder seine eigene Zugriffspunktrichtlinie hat, die mit der zugrundeliegenden Bucket-Richtlinie funktioniert. Access Points können jedoch nur auf Objekte, nicht auf Buckets, einwirken und unterstützen keine regionsübergreifende Replikation. Weitere Informationen finden Sie unter Verwaltung des Zugriffs auf gemeinsam genutzte Datensätze mit Access Points.
- S3 Access Grants — Verwenden Sie S3 Access Grants, das eine sehr große Anzahl von Zugriffen unterstützt, die Zugriff auf Buckets, Präfixe oder Objekte gewähren. Weitere Informationen finden Sie unter Erste Schritte mit S3 Access Grants.

Die AWS-Konto Inhaber- oder Administratorrolle möchte Benutzern oder Gruppen in einem Unternehmensverzeichnis direkt Bucket-, Präfix- oder Objektzugriff gewähren

Anstatt Benutzer, Gruppen und Rollen über AWS Identity and Access Management (IAM) zu verwalten, können Sie Ihr Unternehmensverzeichnis zu hinzufügen AWS IAM Identity Center. Weitere Informationen finden Sie unter Was ist IAM Identity Center? .

Nachdem Sie Ihr Unternehmensverzeichnis hinzugefügt haben, empfehlen wir Ihnen AWS IAM Identity Center, das folgende Zugriffsverwaltungstool zu verwenden, um den Identitäten des Unternehmensverzeichnisses Zugriff auf Ihre S3-Ressourcen zu gewähren:

- S3 Access Grants — Verwenden Sie S3 Access Grants, das die Gewährung von Zugriff für Benutzer oder Rollen in Ihrem Unternehmensverzeichnis unterstützt. Weitere Informationen finden Sie unter Erste Schritte mit S3 Access Grants.

Der AWS-Konto Besitzer oder Bucket-Besitzer möchte dem AWS CloudFront Service Zugriff gewähren, um CloudFront Logs in einen S3-Bucket zu schreiben

Für diesen Anwendungsfall haben wir das folgende Zugriffsverwaltungstool empfohlen:

- Bucket ACL — Der einzig empfohlene Anwendungsfall für Bucket ACLs besteht darin AWS-Services, bestimmten Personen Berechtigungen zu gewähren, z. B. dem CloudFront `awslogsdelivery` Amazon-Konto. Wenn Sie eine Distribution erstellen oder aktualisieren und die CloudFront Protokollierung aktivieren, wird der Bucket CloudFront aktualisiert, ACL sodass das `awslogsdelivery` Konto `FULL_CONTROL` berechtigt ist, Logs in Ihren Bucket zu schreiben. Weitere Informationen finden Sie unter Erforderliche Berechtigungen für die Konfiguration der Standardprotokollierung und für den Zugriff auf Ihre Protokolldateien im Amazon CloudFront Developer Guide. Wenn der Bucket, in dem die Logs gespeichert werden, die Einstellung Bucket Ownerforced für die Deaktivierung von S3 Object Ownership verwendetACLs, CloudFront können keine Logs in den Bucket geschrieben werden. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

Sie als Bucket-Besitzer möchten die volle Kontrolle über Objekte behalten, die dem Bucket von anderen Benutzern hinzugefügt werden

Sie können anderen Konten Zugriff gewähren, um Objekte in Ihren Bucket hochzuladen, indem Sie eine Bucket-Richtlinie, einen Access Point oder S3 Access Grants verwenden. Wenn Sie kontoübergreifenden Zugriff auf Ihren Bucket gewährt haben, können Sie sicherstellen, dass Sie weiterhin die volle Kontrolle über alle in Ihren Bucket hochgeladenen Objekte haben.

Für diesen Anwendungsfall haben wir das folgende Zugriffsverwaltungstool empfohlen:

- Objekteigentum — Behalten Sie für die Einstellung Object Ownership auf Bucket-Ebene die Standardeinstellung Bucket Ownerforced bei.

Fehlerbehebung bei der Zugriffsverwaltung

Die folgenden Ressourcen können Ihnen bei der Behebung von Problemen mit der S3-Zugriffsverwaltung helfen:

Beheben von Fehlern aufgrund einer Zugriffsverweigerung (403 Forbidden)

Wenn Sie auf Probleme mit der Zugriffsverweigerung stoßen, überprüfen Sie die Einstellungen auf Konto- und Bucket-Ebene. Überprüfen Sie außerdem die Zugriffsverwaltungsfunktion, mit der Sie den Zugriff gewähren, um sicherzustellen, dass die Richtlinie, Einstellung oder Konfiguration korrekt ist. Weitere Informationen zu den häufigsten Ursachen für den Fehler „Zugriff verweigert“ (403 Forbidden) in Amazon S3 finden Sie unter Problembehandlung bei verweigertem Zugriff beheben (403 Forbidden) Fehler in Amazon S3.

IAMAccess Analyzer für S3

Wenn Sie keine Ihrer Ressourcen öffentlich zugänglich machen oder den öffentlichen Zugriff auf Ihre Ressourcen einschränken möchten, können Sie IAM Access Analyzer für S3 verwenden. Verwenden Sie IAM Access Analyzer for S3 auf der Amazon S3-Konsole, um alle Buckets zu überprüfen, die über Bucket-Zugriffskontrolllisten (ACLs), Bucket-Richtlinien oder Access Point-Richtlinien verfügen, die öffentlichen oder gemeinsamen Zugriff gewähren. IAMAccess Analyzer for S3 warnt Sie vor Buckets, die so konfiguriert sind, dass sie allen Personen im Internet oder auf anderen Websites AWS-Konten, auch AWS-Konten außerhalb Ihrer Organisation, Zugriff gewähren. Für jeden öffentlichen oder freigegebenen Bucket erhalten Sie Ergebnisse, die die Quelle und die Ebene des öffentlichen oder freigegebenen Zugriffs melden.

In IAM Access Analyzer for S3 können Sie den gesamten öffentlichen Zugriff auf einen Bucket mit einer einzigen Aktion blockieren. Wir empfehlen Ihnen, den gesamten öffentlichen Zugriff auf Ihre Buckets zu blockieren, es sei denn, Sie benötigen öffentlichen Zugriff, um einen bestimmten Anwendungsfall zu unterstützen. Bevor Sie den gesamten öffentlichen Zugriff blockieren, stellen Sie sicher, dass Ihre Anwendungen auch ohne öffentlichen Zugriff ordnungsgemäß funktionieren. Weitere Informationen finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

Sie können auch Ihre Berechtigungseinstellungen auf Bucket-Ebene überprüfen, um detaillierte Zugriffsebenen zu konfigurieren. Für bestimmte und geprüfte Anwendungsfälle, die öffentlichen oder freigegebenen Zugriff erfordern, können Sie Ihre Absicht bestätigen und aufzeichnen, dass der Bucket öffentlich oder freigegeben bleibt, indem Sie die Ergebnisse für den Bucket archivieren. Diese Bucket-Konfigurationen sind jederzeit aufrufbar und änderbar. Sie können Ihre Ergebnisse zu Prüfungszwecken auch als CSV Bericht herunterladen.

IAMAccess Analyzer for S3 ist ohne zusätzliche Kosten auf der Amazon S3 S3-Konsole verfügbar. IAMAccess Analyzer für S3 wird von AWS Identity and Access Management (IAM) IAM Access Analyzer unterstützt. Um IAM Access Analyzer for S3 auf der Amazon S3 S3-Konsole zu verwenden, müssen Sie die IAMKonsole aufrufen und in IAM Access Analyzer für jede einzelne Region einen Analyzer auf Kontoebene erstellen.

Weitere Informationen zu IAM Access Analyzer for S3 finden Sie unter. Überprüfung des Bucket-Zugriffs mit IAM Access Analyzer für S3

Protokollierung und Überwachung

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer Amazon S3 S3-Lösungen, sodass Sie einen Zugriffsfehler einfacher debuggen können. Die Protokollierung kann Aufschluss darüber geben, welche Fehler Benutzer erhalten und wann und welche Anfragen gestellt werden. AWS bietet verschiedene Tools zur Überwachung Ihrer Amazon S3 S3-Ressourcen, wie z. B. die folgenden:

- AWS CloudTrail
- Amazon-S3-Zugriffsprotokolle
- AWS Trusted Advisor
- Amazon CloudWatch

Weitere Informationen finden Sie unter Protokollierung und Überwachung in Amazon S3.

Identity and Access Management für Amazon S3

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf AWS Ressourcen sicher zu kontrollieren. IAMAdministratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Amazon S3 S3-Ressourcen zu verwenden. IAMist eine AWS-Service , die Sie ohne zusätzliche Kosten verwenden können.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

> **Note**
>
> Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter S3 Express One Zone und Arbeiten mit Verzeichnis-Buckets.

Themen

- Zielgruppe
- Authentifizierung mit Identitäten
- Verwalten des Zugriffs mit Richtlinien
- So funktioniert Amazon S3 mit IAM
- Wie Amazon S3 eine Anforderung autorisiert
- Erforderliche Berechtigungen für Amazon S3 API S3-Operationen
- Richtlinien und Berechtigungen in Amazon S3
- Bucket-Richtlinien für Amazon S3
- Identitätsbasierte Richtlinien für Amazon S3
- Exemplarische Vorgehensweisen, die Richtlinien verwenden, um den Zugriff auf Ihre Amazon S3 S3-Ressourcen zu verwalten
- Verwenden von serviceverknüpften Rollen für Amazon S3 Storage Lens
- Problembehandlung Amazon S3 S3-Identität und -Zugriff
- AWS verwaltete Richtlinien für Amazon S3

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, die Sie in Amazon S3 ausführen.

Servicebenutzer — Wenn Sie den Amazon S3 S3-Service für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen zur Verfügung, die Sie benötigen. Da Sie für Ihre Arbeit mehr Amazon S3 S3-Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Wenn

Sie auf eine Funktion in Amazon S3 nicht zugreifen können, finden Sie weitere Informationen unterProblembehandlung Amazon S3 S3-Identität und -Zugriff.

Service-Administrator — Wenn Sie in Ihrem Unternehmen für Amazon S3-Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf Amazon S3. Es ist Ihre Aufgabe, zu bestimmen, auf welche Funktionen und Ressourcen von Amazon S3 Ihre Servicebenutzer zugreifen sollen. Anschließend müssen Sie Anfragen an Ihren IAM Administrator senden, um die Berechtigungen Ihrer Servicebenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die grundlegenden Konzepte von zu verstehenIAM. Weitere Informationen darüber, wie Ihr Unternehmen Amazon S3 nutzen IAM kann, finden Sie unterSo funktioniert Amazon S3 mit IAM.

IAMAdministrator — Wenn Sie ein IAM Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien schreiben können, um den Zugriff auf Amazon S3 zu verwalten. Beispiele für identitätsbasierte Amazon S3 S3-Richtlinien, die Sie in verwenden könnenIAM, finden Sie unter. Identitätsbasierte Richtlinien für Amazon S3

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM Rolle übernehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAMIdentity Center-) Nutzer, die Single-Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als föderierte Identität anmelden, hat Ihr Administrator zuvor einen Identitätsverbund mithilfe von Rollen eingerichtet. IAM Wenn Sie AWS mithilfe eines Verbunds darauf zugreifen, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter So melden Sie sich bei Ihrem an AWS-Konto im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, mit der Sie Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch signieren können. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu signieren, finden Sie im IAMBenutzerhandbuch unter AWS Signature Version 4 für API Anfragen.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen bereitstellen. AWS Empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter Multi-Faktor-Authentifizierung im AWS IAM Identity Center Benutzerhandbuch und AWS Multi-Faktor-Authentifizierung IAM im IAM Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen. Verwenden Sie diese nur, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie im Benutzerhandbuch unter Aufgaben, für die Root-Benutzeranmeldedaten erforderlich sind. IAM

Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAMBenutzer ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wir empfehlen, sich nach Möglichkeit auf temporäre Anmeldeinformationen zu verlassen, anstatt IAM Benutzer mit langfristigen Anmeldeinformationen wie Passwörtern und Zugriffsschlüsseln zu erstellen. Wenn Sie jedoch spezielle Anwendungsfälle haben, für die langfristige Anmeldeinformationen von IAM Benutzern erforderlich sind, empfehlen wir, die Zugriffsschlüssel abwechselnd zu verwenden. Weitere Informationen finden Sie im Benutzerhandbuch unter Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, für die IAM langfristige Anmeldeinformationen erforderlich sind.

Eine IAMGruppe ist eine Identität, die eine Sammlung von IAM Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise eine Gruppe benennen IAMAdminsund dieser Gruppe Berechtigungen zur Verwaltung von IAM Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie im Benutzerhandbuch unter Anwendungsfälle für IAM IAM Benutzer.

IAMRollen

Eine IAMRolle ist eine Identität innerhalb von Ihnen AWS-Konto , für die bestimmte Berechtigungen gelten. Sie ähnelt einem IAM Benutzer, ist jedoch keiner bestimmten Person zugeordnet. Um vorübergehend eine IAM Rolle in der zu übernehmen AWS Management Console, können Sie von einem Benutzer zu einer IAM Rolle (Konsole) wechseln. Sie können eine Rolle übernehmen, indem Sie eine AWS CLI AWS API OR-Operation aufrufen oder eine benutzerdefinierte Operation verwendenURL. Weitere Informationen zu Methoden zur Verwendung von Rollen finden Sie unter Methoden zur Übernahme einer Rolle im IAMBenutzerhandbuch.

IAMRollen mit temporären Anmeldeinformationen sind in den folgenden Situationen nützlich:

- Verbundbenutzerzugriff – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie im IAMBenutzerhandbuch unter Erstellen einer

Rolle für einen externen Identitätsanbieter (Federation). Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Um zu kontrollieren, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in. IAM Informationen zu Berechtigungssätzen finden Sie unter Berechtigungssätze im AWS IAM Identity Center -Benutzerhandbuch.

- Temporäre IAM Benutzerberechtigungen — Ein IAM Benutzer oder eine Rolle kann eine IAM Rolle übernehmen, um vorübergehend verschiedene Berechtigungen für eine bestimmte Aufgabe zu übernehmen.
- Kontoübergreifender Zugriff — Sie können eine IAM Rolle verwenden, um jemandem (einem vertrauenswürdigen Principal) in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zum Unterschied zwischen Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie IAMim Benutzerhandbuch unter Kontoübergreifender Ressourcenzugriff. IAM
- Serviceübergreifender Zugriff — Einige AWS-Services verwenden Funktionen in anderen. AWS-Services Wenn Sie beispielsweise einen Service aufrufen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Zugriffssitzungen weiterleiten (FAS) — Wenn Sie einen IAM Benutzer oder eine Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FASverwendet die Berechtigungen des Prinzipals, der an aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen AWS-Service an nachgelagerte Dienste zu stellen. FASAnfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien beim Stellen von FAS Anfragen finden Sie unter Zugriffssitzungen weiterleiten.
  - Servicerolle — Eine Servicerolle ist eine IAMRolle, die ein Dienst übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM Administrator kann eine Servicerolle von innen heraus erstellen, ändern und löschenIAM. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Erstellen einer Rolle zum Delegieren von Berechtigungen AWS-Service an eine.
  - Dienstbezogene Rolle — Eine dienstverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem

Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM Administrator kann die Berechtigungen für dienstbezogene Rollen anzeigen, aber nicht bearbeiten.

- Auf Amazon ausgeführte Anwendungen EC2 — Sie können eine IAM Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance ausgeführt werden und AWS API Anfragen stellen AWS CLI . Dies ist dem Speichern von Zugriffsschlüsseln innerhalb der EC2 Instance vorzuziehen. Um einer EC2 Instanz eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instanzprofil, das an die Instanz angehängt ist. Ein Instanzprofil enthält die Rolle und ermöglicht Programmen, die auf der EC2 Instanz ausgeführt werden, temporäre Anmeldeinformationen abzurufen. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Verwenden einer IAM Rolle, um Berechtigungen für Anwendungen zu erteilen, die auf EC2 Amazon-Instances ausgeführt werden.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Die Berechtigungen in den Richtlinien legen fest, ob eine Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in AWS Form von JSON Dokumenten gespeichert. Weitere Informationen zur Struktur und zum Inhalt von JSON Richtliniendokumenten finden Sie im IAMBenutzerhandbuch unter Überblick über JSON Richtlinien.

Administratoren können mithilfe von AWS JSON Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Um Benutzern die Erlaubnis zu erteilen, Aktionen mit den Ressourcen durchzuführen, die sie benötigen, kann ein IAM Administrator IAM Richtlinien erstellen. Der Administrator kann dann die IAM Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen übernehmen.

IAMRichtlinien definieren Berechtigungen für eine Aktion, unabhängig von der Methode, mit der Sie den Vorgang ausführen. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die

`iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen aus dem AWS Management Console AWS CLI, dem oder dem abrufen AWS API.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind Dokumente mit JSON Berechtigungsrichtlinien, die Sie an eine Identität anhängen können, z. B. an einen IAM Benutzer, eine Benutzergruppe oder eine Rolle. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen einer identitätsbasierten Richtlinie finden Sie im Benutzerhandbuch unter Definieren benutzerdefinierter IAM Berechtigungen mit vom Kunden verwalteten Richtlinien. IAM

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können. AWS-Konto Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen zur Auswahl zwischen einer verwalteten Richtlinie oder einer Inline-Richtlinie finden Sie im IAMBenutzerhandbuch unter Wählen Sie zwischen verwalteten Richtlinien und Inline-Richtlinien.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM Rollenvertrauensrichtlinien und Amazon S3 S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien nicht IAM in einer ressourcenbasierten Richtlinie verwenden.

Zugriffskontrolllisten () ACLs

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLsähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON Richtliniendokumentformat.

Amazon S3 AWS WAF, und VPC sind Beispiele für Dienste, die unterstützenACLs. Weitere Informationen finden Sie unter Übersicht über ACLs die Zugriffskontrollliste (ACL) im Amazon Simple Storage Service Developer Guide.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- Berechtigungsgrenzen — Eine Berechtigungsgrenze ist eine erweiterte Funktion, mit der Sie die maximalen Berechtigungen festlegen, die eine identitätsbasierte Richtlinie einer IAM Entität (IAMBenutzer oder Rolle) gewähren kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen zu Berechtigungsgrenzen finden Sie im IAMBenutzerhandbuch unter Berechtigungsgrenzen für IAM Entitäten.
- Dienststeuerungsrichtlinien (SCPs) — SCPs sind JSON Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen AWS Organizations. AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Geräte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Entitäten Root-Benutzer des AWS-Kontos. Weitere Informationen zu Organizations und SCPs finden Sie unter Richtlinien zur Servicesteuerung im AWS Organizations Benutzerhandbuch.
- Richtlinien zur Ressourcenkontrolle (RCPs) — RCPs sind JSON Richtlinien, mit denen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten festlegen können, ohne die IAM Richtlinien aktualisieren zu müssen, die jeder Ressource zugeordnet sind, deren Eigentümer Sie sind. Sie RCP schränken die Berechtigungen für Ressourcen in Mitgliedskonten ein und können sich auf die effektiven Berechtigungen für Identitäten auswirken, einschließlich der Root-Benutzer des AWS-Kontos, unabhängig davon, ob sie zu Ihrer Organisation gehören. Weitere Informationen zu OrganizationsRCPs, einschließlich einer Liste AWS-Services dieser Support-LeistungenRCPs, finden Sie unter Resource Control Policies (RCPs) im AWS Organizations Benutzerhandbuch.

- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Sitzungsrichtlinien.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAMBenutzerhandbuch unter Bewertungslogik für Richtlinien.

So funktioniert Amazon S3 mit IAM

Informieren Sie sich vor der Verwendung IAM zur Verwaltung des Zugriffs auf Amazon S3 darüber, welche IAM Funktionen für Amazon S3 verfügbar sind.

IAMFunktionen, die Sie mit Amazon S3 verwenden können

IAMFunktion	Amazon-S3-Support
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Ja
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (services pezifisch)	Ja
ACLs	Ja
ABAC(Tags in Richtlinien)	Teilweise

IAMFunktion	Amazon-S3-Support
Temporäre Anmeldeinformationen	Ja
Zugriffssitzungen weiterleiten (FAS)	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Teilweise

Einen allgemeinen Überblick darüber, wie Amazon S3 und andere AWS Dienste mit den meisten IAM Funktionen funktionieren, finden Sie IAM im IAMBenutzerhandbuch unter AWS Dienste, die mit funktionieren.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

Identitätsbasierte Richtlinien für Amazon S3

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind Dokumente mit JSON Berechtigungsrichtlinien, die Sie an eine Identität anhängen können, z. B. an einen IAM Benutzer, eine Benutzergruppe oder eine Rolle. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen einer identitätsbasierten Richtlinie finden Sie im Benutzerhandbuch unter Definieren benutzerdefinierter IAM Berechtigungen mit vom Kunden verwalteten Richtlinien. IAM

Mit IAM identitätsbasierten Richtlinien können Sie zulässige oder verweigerte Aktionen und Ressourcen sowie die Bedingungen angeben, unter denen Aktionen zulässig oder verweigert werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Weitere Informationen zu allen Elementen, die Sie in einer JSON Richtlinie verwenden können, finden Sie im IAMBenutzerhandbuch unter Referenz zu IAM JSON Richtlinienelementen.

Beispiele für identitätsbasierte Richtlinien für Amazon S3

Beispiele für identitätsbasierte Amazon S3 S3-Richtlinien finden Sie unter. Identitätsbasierte Richtlinien für Amazon S3

Ressourcenbasierte Richtlinien in Amazon S3

Unterstützt ressourcenbasierte Richtlinien: Ja

Ressourcenbasierte Richtlinien sind JSON Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM Rollenvertrauensrichtlinien und Amazon S3 S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um den kontoübergreifenden Zugriff zu ermöglichen, können Sie in einer ressourcenbasierten Richtlinie ein ganzes Konto oder IAM Entitäten in einem anderen Konto als Prinzipal angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM Administrator des vertrauenswürdigen Kontos auch der Prinzipalentität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource gewähren. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie IAMim IAMBenutzerhandbuch unter Kontenübergreifender Ressourcenzugriff.

Der Amazon S3 S3-Service unterstützt Bucket-Richtlinien, Zugriffspunkt-Richtlinien und Zugriffsberechtigungen:

- Bucket-Richtlinien sind ressourcenbasierte Richtlinien, die an einen Amazon S3 S3-Bucket angehängt sind. Eine Bucket-Richtlinie definiert, welche Principals Aktionen für den Bucket ausführen können.
- Access Point-Richtlinien sind ressourcenbasierte Richtlinien, die in Verbindung mit der zugrunde liegenden Bucket-Richtlinie bewertet werden.
- Zugriffsberechtigungen sind ein vereinfachtes Modell zur Definition von Zugriffsberechtigungen für Daten in Amazon S3 nach Präfix, Bucket oder Objekt. Informationen zu S3-Zugriffsberechtigungen finden Sie unterVerwalten des Zugriffs mit S3-Zugriffsberechtigungen.

Prinzipien für Bucket-Richtlinien

Das `Principal` Element gibt den Benutzer, das Konto, den Dienst oder eine andere Entität an, der der Zugriff auf eine Ressource entweder gewährt oder verweigert wird. Die folgenden sind Beispiele legen den `Principal` fest. Weitere Informationen finden Sie unter Principal im IAMBenutzerhandbuch.

Erteilen Sie Berechtigungen für einen AWS-Konto

Um einem Berechtigungen zu erteilen AWS-Konto, identifizieren Sie das Konto anhand des folgenden Formats.

```
"AWS":"account-ARN"
```

Im Folgenden sind einige Beispiele aufgeführt.

```
"Principal":{"AWS":"arn:aws:iam::AccountIDWithoutHyphens:root"}
```

```
"Principal":{"AWS":
["arn:aws:iam::AccountID1WithoutHyphens:root","arn:aws:iam::AccountID2WithoutHyphens:root"]}
```

Erteilen Sie einem IAM Benutzer Berechtigungen

Um einem IAM Benutzer in Ihrem Konto Berechtigungen zu erteilen, müssen Sie ein `"AWS":"user-ARN"` Name-Wert-Paar angeben.

```
"Principal":{"AWS":"arn:aws:iam::account-number-without-hyphens:user/username"}
```

Ausführliche Beispiele mit step-by-step Anweisungen finden Sie unter Beispiel 1: Bucket-Eigentümer erteilt seinen Benutzern Bucket-Berechtigungen und. Beispiel 3: Bucket-Eigentümer, der Berechtigungen für Objekte erteilt, die ihm nicht gehören

> **Note**
>
> Wenn eine IAM Identität gelöscht wird, nachdem Sie Ihre Bucket-Richtlinie aktualisiert haben, zeigt die Bucket-Richtlinie anstelle einer eindeutigen Kennung im Hauptelement anARN. Diese eindeutigen Daten IDs werden niemals wiederverwendet, sodass Sie Prinzipale mit eindeutigen Kennungen problemlos aus all Ihren Richtlinienerklärungen entfernen können.

Weitere Informationen zu eindeutigen Kennungen finden Sie unter IAMIdentifikatoren im Benutzerhandbuch. IAM

Erteilen anonymer Berechtigungen

> **Warning**
>
> Seien Sie vorsichtig, wenn Sie anonymen Zugriff auf Ihren Amazon-S3-Bucket erteilen. Wenn Sie anonymen Zugriff gewähren, kann jeder auf der ganzen Welt auf Ihren Bucket zugreifen. Wir empfehlen dringend, nie einen anonymen Schreibzugriff auf Ihren S3-Bucket zu gewähren.

Um jedem Benutzer Berechtigung zu erteilen, auch als anonymer Zugriff bezeichnet, legen Sie den Platzhalter (`"*"`) auf den Wert `Principal` fest. Wenn Sie beispielsweise ihren Bucket als Website konfigurieren, müssen Sie alle Objekte im Bucket öffentlich zugänglich machen.

```
"Principal":"*"
```

```
"Principal":{"AWS":"*"}
```

Durch die Verwendung `"Principal": "*"` mit `Allow` Wirkung in einer ressourcenbasierten Richtlinie kann jeder, auch wenn er nicht angemeldet ist AWS, auf Ihre Ressource zugreifen.

Wenn Sie die Option `"Principal" : { "AWS" : "*" }` mit `Allow` Wirkung in einer ressourcenbasierten Richtlinie verwenden, können alle Root-Benutzer, IAM Benutzer, Sitzungen mit übernommener Rolle oder Verbundbenutzer in einem beliebigen Konto in derselben Partition auf Ihre Ressource zugreifen.

Für anonyme Benutzer sind diese beiden Methoden gleichwertig. Weitere Informationen finden Sie im Benutzerhandbuch unter Alle Hauptbenutzer. IAM

Sie können keinen Platzhalter verwenden, um einem Teil eines Prinzipalnamens oder zuzuordnen. ARN

> **Important**
>
> Da jeder eine erstellen kann AWS-Konto, ist die Sicherheitsstufe dieser beiden Methoden gleichwertig, auch wenn sie unterschiedlich funktionieren.

Ressourcenberechtigungen beschränken

Sie können die Ressourcenrichtlinie auch verwenden, um den Zugriff auf Ressourcen einzuschränken, die sonst für IAM Prinzipale verfügbar wären. Verwenden Sie eine `Deny`-Anweisung, um den Zugriff zu verhindern.

Das folgende Beispiel blockiert den Zugriff, wenn kein sicheres Transportprotokoll verwendet wird:

```
{"Effect": "Deny",
 "Principal": "*",
 "Action": "s3:*",
 "Resource": <bucket ARN>,
 "Condition": {
   "Boolean": { "aws:SecureTransport" : "false"}
 }
}
```

Es hat sich für diese Richtlinie bewährt, `"Principal": "*"` so zu verwenden, dass diese Einschränkung für alle gilt, anstatt zu versuchen, mit dieser Methode nur bestimmten Konten oder Prinzipalen den Zugriff zu verweigern.

Erfordern Sie Zugriff über CloudFront URLs

Sie können verlangen, dass Ihre Benutzer nur auf Ihre Amazon S3-Inhalte zugreifen, indem sie Amazon S3 CloudFront URLs anstelle von Amazon S3 verwendenURLs. Erstellen Sie dazu eine CloudFront Origin-Zugriffskontrolle (OAC). Ändern Sie dann die Berechtigungen für Ihre S3-Daten. In Ihrer Bucket-Richtlinie können Sie Folgendes CloudFront als Principal festlegen:

```
"Principal":{"Service":"cloudfront.amazonaws.com"}
```

Verwenden Sie ein `Condition` Element in der Richtlinie, um nur dann Zugriff auf den Bucket CloudFront zu gewähren, wenn die Anfrage im Namen der CloudFront Distribution erfolgt, die den S3-Ursprung enthält.

```
        "Condition": {
           "StringEquals": {
              "AWS:SourceArn":
"arn:aws:cloudfront::111122223333:distribution/CloudFront-distribution-ID"
           }
        }
```

Weitere Informationen zur Anforderung von S3-Zugriff über CloudFront URLs finden Sie unter Beschränken des Zugriffs auf einen Amazon Simple Storage Service-Ursprung im Amazon CloudFront Developer Guide. Weitere Informationen zu den Sicherheits- und Datenschutzvorteilen der Nutzung von Amazon CloudFront finden Sie unter Konfiguration des sicheren Zugriffs und Beschränkung des Zugriffs auf Inhalte.

Beispiele für ressourcenbasierte Richtlinien für Amazon S3

- Richtlinienbeispiele für Amazon S3 S3-Buckets finden Sie unterBucket-Richtlinien für Amazon S3.
- Richtlinienbeispiele für Access Points finden Sie unterKonfiguration von IAM Richtlinien für die Verwendung von Access Points.

Politische Maßnahmen für Amazon S3

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON Richtlinien festlegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das `Action` Element einer JSON Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, für die nur eine Genehmigung erforderlich ist und für die es keinen entsprechenden Vorgang gibt. API Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Im Folgenden werden verschiedene Arten von Zuordnungsbeziehungen zwischen API S3-Vorgängen und den erforderlichen Richtlinienaktionen dargestellt.

- One-to-one Zuordnung mit demselben Namen. Um beispielsweise den `PutBucketPolicy` API Vorgang verwenden zu können, ist die `s3:PutBucketPolicy` Richtlinienaktion erforderlich.
- One-to-one Zuordnung mit unterschiedlichen Namen. Um den `ListObjectsV2` API Vorgang verwenden zu können, ist beispielsweise die `s3:ListBucket` Richtlinienaktion erforderlich.
- One-to-many Kartierung. Um beispielsweise die `HeadObject` API Operation verwenden zu können, `s3:GetObject` ist der erforderlich. Wenn Sie S3 Object Lock verwenden und den Status oder die Aufbewahrungseinstellungen eines Objekts abrufen möchten, sind außerdem die entsprechenden Aktionen `s3:GetObjectLegalHold` oder die `s3:GetObjectRetention` Richtlinienaktionen erforderlich, bevor Sie den `HeadObject` API Vorgang verwenden können.
- Many-to-one Zuordnung. Um beispielsweise die `HeadBucket` API Operationen `ListObjectsV2` oder zu verwenden, ist die `s3:ListBucket` Richtlinienaktion erforderlich.

Eine Liste der Amazon S3-Aktionen zur Verwendung in Richtlinien finden Sie unter Von Amazon S3 definierte Aktionen in der Service Authorization Reference. Eine vollständige Liste der Amazon S3 API S3-Operationen finden Sie unter Amazon S3 API S3-Aktionen in der Amazon Simple Storage API Service-Referenz.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

Richtlinienaktionen in Amazon S3 verwenden das folgende Präfix vor der Aktion:

```
s3
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
      "s3:action1",
      "s3:action2"
         ]
```

Bucket-Operationen

Bucket-Operationen sind API S3-Operationen, die auf dem Bucket-Ressourcentyp ausgeführt werden. Beispiel: `CreateBucket`, `ListObjectsV2` und `PutBucketPolicy`. S3-Richtlinienaktionen für Bucket-Operationen erfordern, dass das `Resource` Element in Bucket-Richtlinien oder IAM identitätsbasierten Richtlinien der Amazon Resource Name (ARN) -Bezeichner vom Typ S3-Bucket im folgenden Beispielformat ist.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
```

Die folgende Bucket-Richtlinie gewährt dem Benutzer *Akua* mit Konto *12345678901* die `s3:ListBucket` Erlaubnis, Folgendes durchzuführen ListObjectsV2APIOperation und Listet Objekte in einem S3-Bucket auf.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Akua to list objects in the bucket",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::12345678901:user/Akua"
            },
            "Action": [
            "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
        }
    ]
}
```

Bucket-Operationen in Access Point-Richtlinien

In einer Zugriffspunktrichtlinie gewährte Berechtigungen sind nur wirksam, wenn der zugrunde liegende Bucket dieselben Berechtigungen zulässt. Wenn Sie S3 Access Points verwenden, müssen Sie die Zugriffskontrolle vom Bucket an den Access Point delegieren oder dieselben Berechtigungen in den Access Point-Richtlinien zur Richtlinie des zugrunde liegenden Buckets hinzufügen. Weitere Informationen finden Sie unter Konfiguration von IAM Richtlinien für die Verwendung von Access Points. In Zugriffspunktrichtlinien erfordern S3-Richtlinienaktionen für Bucket-Operationen, dass Sie den Zugriffspunkt ARN für das `Resource` Element im folgenden Format verwenden.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-point"
```

Die folgende Zugriffspunktrichtlinie gewährt dem Benutzer *Akua* mit Konto *12345678901* die `s3:ListBucket` Erlaubnis, Folgendes auszuführen ListObjectsV2APIVorgang über den genannten S3-Zugangspunkt*example-access-point*. Diese Berechtigung ermöglicht es*Akua*, die Objekte in dem Bucket aufzulisten, dem sie zugeordnet sind*example-access-point*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Akua to list objects in the bucket through access point",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::12345678901:user/Akua"
            },
            "Action": [
            "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-
point"
        }
    ]
}
```

> **Note**
>
> Nicht alle Bucket-Operationen werden von S3 Access Point unterstützt. Weitere Informationen finden Sie unter Zugriffspunkt-Kompatibilität mit S3-Vorgänge.

Objektoperationen

Objektoperationen sind API S3-Operationen, die auf den Objektressourcentyp einwirken. Beispiel: `GetObject`, `PutObject` und `DeleteObject`. Für S3-Richtlinienaktionen für Objektoperationen muss das `Resource` Element in den Richtlinien das S3-Objekt ARN in den folgenden Beispielformaten sein.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
```

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/*"
```

> **Note**
>
> Das Objekt ARN muss nach dem Bucket-Namen einen Schrägstrich enthalten, wie in den vorherigen Beispielen gezeigt.

Die folgende Bucket-Richtlinie gewährt dem Benutzer *Akua* mit Konto *12345678901* die `s3:PutObject` entsprechende Berechtigung. Diese Erlaubnis ermöglicht *Akua* die Verwendung von PutObjectAPIVorgang zum Hochladen von Objekten in den genannten S3-Bucket*amzn-s3-demo-bucket*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Akua to upload objects",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::12345678901:user/Akua"
            },
            "Action": [
            "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Objektoperationen in Zugriffspunktrichtlinien

Wenn Sie S3 Access Points verwenden, um den Zugriff auf Objektoperationen zu steuern, können Sie Zugriffspunktrichtlinien verwenden. Wenn Sie Zugriffspunktrichtlinien verwenden, müssen Sie bei S3-Richtlinienaktionen für Objektoperationen den Zugriffspunkt ARN für das `Resource` Element im folgenden Format verwenden:`arn:aws:s3:`*region*`:`*account-id*`:accesspoint/`*access-point-name*`/object/`*resource*. Bei Objektoperationen, die Zugriffspunkte verwenden, müssen Sie den `/object/` Wert hinter dem gesamten Zugriffspunkt ARN in das `Resource` Element aufnehmen. Hier sind einige Beispiele.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-point/object/
*"
```

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-point/
object/prefix/*"
```

Die folgende Zugriffspunktrichtlinie gewährt dem Benutzer *Akua* mit Konto *12345678901* die `s3:GetObject` entsprechende Berechtigung. Diese Berechtigung ermöglicht *Akua* die Ausführung von GetObjectAPIVorgang über den Access Point, *example-access-point* der für alle Objekte im Bucket benannt ist, der dem Access Point zugeordnet ist.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow Akua to get objects through access point",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::12345678901:user/Akua"
            },
            "Action": [
            "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-
point/object/*"
        }
    ]
}
```

> **Note**
>
> Nicht alle Objektoperationen werden von Access Points unterstützt. Weitere Informationen finden Sie unter Zugriffspunkt-Kompatibilität mit S3-Vorgänge.

Operationen an Zugriffspunkten

Access Point-Operationen sind API S3-Operationen, die auf dem `accesspoint` Ressourcentyp ausgeführt werden. Beispiel: `CreateAccessPoint`, `DeleteAccessPoint` und `GetAccessPointPolicy`. S3-Richtlinienaktionen für Zugriffspunktoperationen können

nur in IAM identitätsbasierten Richtlinien verwendet werden, nicht in Bucket-Richtlinien oder Zugriffspunktrichtlinien. Für Access Point-Operationen muss das `Resource` Element der Access Point ARN im folgenden Beispielformat sein.

```
"Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/example-access-point"
```

Die folgende IAM identitätsbasierte Richtlinie gewährt die `s3:GetAccessPointPolicy` Erlaubnis zur Durchführung von GetAccessPointPolicyAPIVorgang auf dem genannten S3-Zugriffspunkt. *`example-access-point`*

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Grant permission to retrieve the access point policy of access
 point example-access-point",
            "Effect": "Allow",
            "Action": [
            "s3:GetAccessPointPolicy"
            ],
            "Resource": "arn:aws:s3:*:123456789012:accesspoint/example-access-point"
        }
    ]
}
```

Wenn Sie Access Points verwenden, um den Zugriff auf Bucket-Operationen zu steuern, sieheBucket-Operationen in Access Point-Richtlinien; Informationen zur Steuerung des Zugriffs auf Objektoperationen finden Sie unterObjektoperationen in Zugriffspunktrichtlinien. Weitere Informationen zur Konfiguration von Zugriffspunktrichtlinien finden Sie unterKonfiguration von IAM Richtlinien für die Verwendung von Access Points.

Object Lambda Access Point-Operationen

Mit S3 Object Lambda können Sie `GET`-, `LIST`- und `HEAD`-Anforderungen von Amazon S3 eigenen Code hinzufügen, um Daten zu ändern und zu verarbeiten, wenn sie an eine Anwendung zurückgegeben werden. Sie können Anfragen über einen Object Lambda Access Point stellen, was genauso funktioniert wie Anfragen über andere Access Points. Weitere Informationen finden Sie unter Transformieren von Objekten mit S3 Object Lambda.

Weitere Informationen zur Konfiguration von Richtlinien für Object Lambda Access Point-Operationen finden Sie unterKonfiguration von IAM Richtlinien für Object Lambda Access Points.

Access Point-Operationen in mehreren Regionen

Ein Access Point mit mehreren Regionen bietet einen globalen Endpunkt, über den Anwendungen Anfragen von S3-Buckets bearbeiten können, die sich in mehreren befinden. AWS-Region Sie können einen Multi-Region-Access Point verwenden, um multiregionale Anwendungen mit derselben Architektur zu erstellen, die in einer einzelnen Region verwendet wird, und diese Anwendungen dann überall auf der Welt ausführen. Weitere Informationen finden Sie unter Verwaltung von regionsübergreifendem Verkehr mit multiregionalen Access Points.

Weitere Informationen zur Konfiguration von Richtlinien für Access Point-Operationen in mehreren Regionen finden Sie unter. Beispielrichtlinien für Multi-Region Access Points

Batch-Job-Operationen

Auftragsoperationen (Batch Operations) sind API S3-Operationen, die auf dem Ressourcentyp Job ausgeführt werden. Beispiel: `DescribeJob` und `CreateJob`. S3-Richtlinienaktionen für Auftragsoperationen können nur in IAM identitätsbasierten Richtlinien verwendet werden, nicht in Bucket-Richtlinien. Für Auftragsoperationen muss das `Resource` Element in IAM identitätsbasierten Richtlinien außerdem das `job` ARN folgende Beispielformat haben.

```
"Resource": "arn:aws:s3:*:123456789012:job/*"
```

Die folgende IAM identitätsbasierte Richtlinie gewährt die `s3:DescribeJob` Erlaubnis, den DescribeJobAPIVorgang für den genannten Auftrag S3 Batch Operations auszuführen. *example-job*

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow describing the Batch operation job example-job",
            "Effect": "Allow",
            "Action": [
            "s3:DescribeJob"
            ],
            "Resource": "arn:aws:s3:*:123456789012:job/example-job"
        }
    ]
}
```

Konfigurationsvorgänge für S3 Storage Lens

Weitere Informationen zur Konfiguration von S3 Storage Lens-Konfigurationsvorgängen finden Sie unterAmazon S3 Storage Lens-Berechtigungen einrichten.

Kontovorgänge

Kontovorgänge sind API S3-Operationen, die auf Kontoebene ausgeführt werden. Zum Beispiel `GetPublicAccessBlock` (für ein Konto). Konto ist kein von Amazon S3 definierter Ressourcentyp. S3-Richtlinienaktionen für Kontooperationen können nur in IAM identitätsbasierten Richtlinien verwendet werden, nicht in Bucket-Richtlinien. Außerdem ist für Kontooperationen das `Resource` Element in IAM identitätsbasierten Richtlinien erforderlich. `"*"`

Die folgende IAM identitätsbasierte Richtlinie erteilt die `s3:GetAccountPublicAccessBlock` Erlaubnis, die Ausführung auf Kontoebene durchzuführen GetPublicAccessBlockAPIBetrieb und Abrufen der Einstellungen für die Sperrung des öffentlichen Zugriffs auf Kontoebene.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"Allow retrieving the account-level Public Access Block settings",
         "Effect":"Allow",
         "Action":[
            "s3:GetAccountPublicAccessBlock"
         ],
         "Resource":[
            "*"
         ]
       }
    ]
}
```

Richtlinienbeispiele für Amazon S3

- Beispiele für identitätsbasierte Amazon S3 S3-Richtlinien finden Sie unter. Identitätsbasierte Richtlinien für Amazon S3
- Beispiele für ressourcenbasierte Amazon S3 S3-Richtlinien finden Sie unter Bucket-Richtlinien für Amazon S3 und. Konfiguration von IAM Richtlinien für die Verwendung von Access Points

Richtlinienressourcen für Amazon S3

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON Richtlinien festlegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das `Resource` JSON Richtlinienelement gibt das Objekt oder die Objekte an, für die die Aktion gilt. Anweisungen müssen entweder ein – `Resource`oder ein `NotResource`-Element enthalten. Es hat sich bewährt, eine Ressource mit ihrem Amazon-Ressourcennamen (ARN) anzugeben. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Einige Amazon S3 API S3-Aktionen unterstützen mehrere Ressourcen. `s3:GetObject`Zugreift beispielsweise auf *`example-resource-1`* und zu*`example-resource-2`*, sodass ein Principal über Berechtigungen für den Zugriff auf beide Ressourcen verfügen muss. Um mehrere Ressourcen in einer einzigen Anweisung anzugeben, trennen Sie sie ARNs durch Kommas, wie im folgenden Beispiel gezeigt.

```
"Resource": [
      "example-resource-1",
      "example-resource-2"
```

Ressourcen in Amazon S3 sind Buckets, Objekte, Zugriffspunkte oder Jobs. Verwenden Sie in einer Richtlinie den Amazon-Ressourcennamen (ARN) des Buckets, Objekts, Access Points oder Jobs, um die Ressource zu identifizieren.

Eine vollständige Liste der Amazon S3-Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter Von Amazon S3 definierte Ressourcen in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie die ARN einzelnen Ressourcen spezifizieren können, finden Sie unter Von Amazon S3 definierte Aktionen.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

Platzhalterzeichen in der Ressource ARNs

Sie können Platzhalterzeichen als Teil der Ressource verwenden. ARN Sie können die Platzhalterzeichen (*und?) in jedem ARN Segment (die durch Doppelpunkte getrennten Teile) verwenden. Ein Sternchen (*) steht für kein Zeichen oder eine beliebige Kombination von mehreren Zeichen und ein Fragezeichen (?) entspricht einem beliebigen einzelnen Zeichen. Sie können in jedem Segment mehrere ? Zeichen * oder verwenden. Ein Platzhalterzeichen kann sich jedoch nicht über mehrere Segmente erstrecken.

- Im Folgenden ARN wird das * Platzhalterzeichen im `relative-ID` Teil von verwendet, ARN um alle Objekte im Bucket zu identifizieren. *amzn-s3-demo-bucket*

```
arn:aws:s3:::amzn-s3-demo-bucket/*
```

- Im Folgenden werden ARN alle * S3-Buckets und -Objekte bezeichnet.

```
arn:aws:s3:::*
```

- Im Folgenden ARN werden beide Platzhalterzeichen * und?, im `relative-ID` Teil verwendet. Dadurch ARN werden alle Objekte in Buckets identifiziert, z. B. *amzn-s3-demo-example1bucket*,, und so weiter. *amzn-s3-demo-example2bucket* *amzn-s3-demo-example3bucket*

```
arn:aws:s3:::amzn-s3-demo-example?bucket/*
```

Richtlinienvariablen für Ressourcen ARNs

Sie können Richtlinienvariablen in Amazon S3 verwendenARNs. Bei der Überprüfung der Richtlinien werden diese vordefinierten Variablen durch ihre entsprechenden Werte ersetzt. Angenommen, Sie organisieren Ihren Bucket als eine Sammlung von Ordnern mit einem Ordner für jeden Ihrer Benutzer. Der Ordnername entspricht dem Benutzernamen. Um Benutzern Berechtigungen für ihre Ordner zu gewähren, können Sie eine Richtlinienvariable in der Ressource angebenARN:

```
arn:aws:s3:::bucket_name/developers/${aws:username}/
```

Zur Laufzeit, wenn die Richtlinie ausgewertet wird, ARN wird die Variable `${aws:username}` in der Ressource durch den Benutzernamen der Person ersetzt, die die Anfrage stellt.

Richtlinienbeispiele für Amazon S3

- Beispiele für identitätsbasierte Amazon S3 S3-Richtlinien finden Sie unter. Identitätsbasierte Richtlinien für Amazon S3
- Beispiele für ressourcenbasierte Amazon S3 S3-Richtlinien finden Sie unter Bucket-Richtlinien für Amazon S3 und. Konfiguration von IAM Richtlinien für die Verwendung von Access Points

Schlüssel für Richtlinienbedingungen für Amazon S3

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen `AND`-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen `OR` Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Sie können einem IAM Benutzer beispielsweise nur dann Zugriff auf eine Ressource gewähren, wenn sie mit seinem IAM Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter IAMRichtlinienelemente: Variablen und Tags.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontext-Schlüssel für AWS globale Bedingungen im IAMBenutzerhandbuch.

Jeder Amazon S3 S3-Bedingungsschlüssel ist demselben Anforderungsheader mit demselben Namen zugeordnet, der von API dem zugelassen wird, für den die Bedingung gesetzt werden kann. Amazon-S3-spezifische Bedingungsschlüssel bestimmen das Verhalten der Anforderungs-Header mit demselben Namen. Beispielsweise definiert der Bedingungsschlüssel, der `s3:VersionId` verwendet wird, um die bedingte Berechtigung zu `s3:GetObjectVersion` erteilen, das Verhalten des `versionId` Abfrageparameters, den Sie in einer GET Objektanforderung festlegen.

Eine Liste der Amazon S3-Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von Amazon S3 definierte Aktionen.

Beispiel: Beschränkung von Objekt-Uploads auf Objekte mit einer bestimmten Speicherklasse

Angenommen, Konto A, dargestellt durch die Konto-ID *123456789012*, besitzt einen Bucket. Das Konto, das der Administrator einschränken möchte *Dave*, ein Benutzer in Konto A, so dass *Dave* kann Objekte nur dann in den Bucket hochladen, wenn das Objekt in der `STANDARD_IA` Speicherklasse gespeichert ist. Um das Hochladen von Objekten auf eine bestimmte Speicherklasse einzuschränken, kann der Administrator von Konto A den Bedingungsschlüssel `s3:x-amz-storage-class` verwenden wie in der folgenden Bucket-Beispielrichtlinie gezeigt.

```
{
                "Version": "2012-10-17",
                "Statement": [
                  {
                    "Sid": "statement1",
                    "Effect": "Allow",
                    "Principal": {
                      "AWS": "arn:aws:iam::123456789012:user/Dave"
                    },
                    "Action": "s3:PutObject",
                    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
                    "Condition": {
                      "StringEquals": {
                        "s3:x-amz-storage-class": [
                          "STANDARD_IA"
                        ]
                      }
                    }
```

```
                }
            ]
        }
```

Im Beispiel gibt der Block `Condition` die Bedingung `StringEquals` an, die auf das angegebene Schlüssel-Wert-Paar angewendet wird, `"s3:x-amz-acl":["public-read"]`. Es gibt einen Satz vordefinierter Schlüssel, die Sie zum Ausdruck einer Bedingung verwenden können. Das Beispiel verwendet den Bedingungsschlüssel `s3:x-amz-acl`. Diese Bedingung erfordert, dass der Benutzer den `x-amz-acl` Header mit dem Wert `public-read` in jede `PutObject` Anfrage einfügt.

Richtlinienbeispiele für Amazon S3

- Beispiele für identitätsbasierte Amazon S3 S3-Richtlinien finden Sie unter. Identitätsbasierte Richtlinien für Amazon S3
- Beispiele für ressourcenbasierte Amazon S3 S3-Richtlinien finden Sie unter Bucket-Richtlinien für Amazon S3 und. Konfiguration von IAM Richtlinien für die Verwendung von Access Points

ACLsauf Amazon S3

UnterstütztACLs: Ja

In Amazon S3 kontrollieren Zugriffskontrolllisten (ACLs), welche Berechtigungen für den Zugriff auf eine Ressource AWS-Konten haben. ACLsähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das Format des JSON Richtliniendokuments.

> **Important**
>
> Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs.

Informationen zur Verwendung ACLs zur Zugriffskontrolle in Amazon S3 finden Sie unterVerwaltung des Zugriffs mit ACLs.

ABACmit Amazon S3

Unterstützt ABAC (Tags in Richtlinien): Teilweise

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen auf der Grundlage von Attributen definiert werden. In werden AWS diese Attribute als Tags

bezeichnet. Sie können Tags an IAM Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt vonABAC. Anschließend entwerfen Sie ABAC Richtlinien, die Operationen zulassen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt, auf die er zugreifen möchte.

ABACist hilfreich in Umgebungen, die schnell wachsen, und hilft in Situationen, in denen die Richtlinienverwaltung umständlich wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen dazu finden Sie ABAC unter Definieren von Berechtigungen mit ABAC Autorisierung im IAMBenutzerhandbuch. Ein Tutorial mit Schritten zur Einrichtung finden Sie im ABAC Benutzerhandbuch unter Verwenden der attributebasierten Zugriffskontrolle (ABAC). IAM

Beispiele für identitätsbasierte Richtlinien zur Beschränkung des Zugriffs auf S3 Batch Operations-Jobs auf der Grundlage von Tags finden Sie unter. Steuern von Berechtigungen für Batch-Operationen mithilfe von Job-Tags

ABACund Objekt-Tags

In ABAC Richtlinien verwenden Objekte `s3:` Tags anstelle von `aws:` Tags. Um den Zugriff auf Objekte auf der Grundlage von Objekt-Tags zu steuern, geben Sie Tag-Informationen in der Condition Element einer Richtlinie unter Verwendung der folgenden Tags:

- `s3:ExistingObjectTag/`*`tag-key`*
- `s3:s3:RequestObjectTagKeys`
- `s3:RequestObjectTag/`*`tag-key`*

Informationen zur Verwendung von Objekt-Tags zur Zugriffskontrolle, einschließlich beispielhafter Berechtigungsrichtlinien, finden Sie unterMarkierungs- und Zugriffskontrollrichtlinien.

Temporäre Anmeldeinformationen mit Amazon S3 verwenden

Unterstützt temporäre Anmeldeinformationen: Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen darüber, AWS-Services wie Sie mit temporären Anmeldeinformationen arbeiten können AWS-Services , finden Sie IAM im IAMBenutzerhandbuch unter Informationen zum Arbeiten mit.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Kennwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Rollenwechsel finden Sie im Benutzerhandbuch unter Von einem Benutzer zu einer IAM Rolle (Konsole) wechseln. IAM

Mit dem AWS CLI oder können Sie manuell temporäre Anmeldeinformationen erstellen AWS API. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldeinformationen unter IAM.

Zugriffssitzungen für Amazon S3 weiterleiten

Unterstützt Forward-Access-Sitzungen (FAS): Ja

Wenn Sie einen IAM Benutzer oder eine Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FASverwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen AWS-Service an nachgelagerte Dienste zu stellen. FASAnfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien beim Stellen von FAS Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

- FASwird von Amazon S3 verwendet, um Aufrufe zur Entschlüsselung eines Objekts AWS KMS zu tätigen, wenn SSE - zum Verschlüsseln verwendet KMS wurde. Weitere Informationen finden Sie unter Serverseitige Verschlüsselung mit AWS KMS Schlüsseln verwenden (SSE-) KMS.
- S3 Access Grants verwendet auch. FAS Nachdem Sie eine Zugriffsberechtigung für Ihre S3-Daten für eine bestimmte Identität erstellt haben, fordert der Empfänger temporäre Anmeldeinformationen von S3 Access Grants an. S3 Access Grants erhält temporäre Anmeldeinformationen für den

Antragsteller von AWS STS und sendet die Anmeldeinformationen an den Antragsteller. Weitere Informationen finden Sie unter Anfordern des Zugriffs auf Amazon-S3-Daten über S3 Access Grants.

Servicerollen für Amazon S3

Unterstützt Servicerollen: Ja

Eine Servicerolle ist eine IAMRolle, die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM Administrator kann eine Servicerolle von innen heraus erstellen, ändern und löschenIAM. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Erstellen einer Rolle zum Delegieren von Berechtigungen AWS-Service an eine.

> **Warning**
>
> Durch das Ändern der Berechtigungen für eine Servicerolle kann die Amazon S3 S3-Funktionalität beeinträchtigt werden. Bearbeiten Sie Servicerollen nur, wenn Amazon S3 Sie dazu anleitet.

Servicebezogene Rollen für Amazon S3

Unterstützt serviceverknüpfte Rollen: Teilweise

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM Administrator kann die Berechtigungen für dienstbezogene Rollen anzeigen, aber nicht bearbeiten.

Amazon S3 unterstützt serviceverknüpfte Rollen für Amazon S3 Storage Lens. Einzelheiten zum Erstellen oder Verwalten von serviceverknüpften Amazon S3 S3-Rollen finden Sie unterVerwenden von serviceverknüpften Rollen für Amazon S3 Storage Lens.

Amazon S3 Service als Principal

Name des Dienstes in der Richtlinie	S3-Funktion	Weitere Informationen
`s3.amazonaws.com`	S3-Replikation	Überblick über die Einrichtung der Live-Replikation
`s3.amazonaws.com`	S3-Ereignisbenachrichtigungen	Amazon-S3-Ereignis-Benachrichtigungen
`s3.amazonaws.com`	S3-Bestand	Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory
`access-grants.s3.amazonaws.com`	S3 Access Grants	Registrieren eines Speicherorts
`batchoperations.s3.amazonaws.com`	S3-BatchVorgänge	Gewähren von Berechtigungen für Batchoperationen
`logging.s3.amazonaws.com`	Protokollierung des S3-Serverzugriffs	Aktivieren Sie die Amazon-S3-Server-Zugriffsprotokollierung
`storage-lens.s3.amazonaws.com`	S3 Storage Lens	Anzeigen von Amazon S3-Storage-Lens-Metriken mit einem Datenexport

Wie Amazon S3 eine Anforderung autorisiert

Wenn Amazon S3 eine Anforderung erhält – z. B. ein Bucket oder eine Objektoperation –, überprüft es zuerst, ob der Auftraggeber die erforderlichen Berechtigungen besitzt. Amazon S3 bewertet alle relevanten Zugriffsrichtlinien, Benutzerrichtlinien und ressourcenbasierten Richtlinien (Bucket-Richtlinie, Bucket-Zugriffskontrollliste (ACL) und ObjektACL), um zu entscheiden, ob die Anfrage autorisiert werden soll.

> **Note**
>
> Wenn bei der Amazon S3-Berechtigungsprüfung keine gültigen Berechtigungen gefunden werden, wird die Fehlermeldung Zugriff verweigert (403 Forbidden) Permission denied (403 Forbidden) zurückgegeben. Weitere Informationen finden Sie unter Problembehandlung bei Access Denied (403 Forbidden) in Amazon S3.

Um festzustellen, ob der Anforderer die Erlaubnis hat, den bestimmten Vorgang auszuführen, geht Amazon S3 bei Eingang einer Anfrage der Reihe nach wie folgt vor:

1. Konvertiert alle relevanten Zugriffsrichtlinien (Benutzerrichtlinie, Bucket-Richtlinie undACLs) zur Laufzeit in eine Reihe von Richtlinien zur Auswertung.
2. Es wertet in den folgenden Schritten die resultierende Richtlinienmenge aus. Amazon S3 wertet in jedem Schritt eine Untermenge der Richtlinien in einem spezifischen Kontext aus, basierend auf der Kontextautorität.
   a. Benutzerkontext – Im Benutzerkontext ist das übergeordnete Konto, zu dem der Benutzer gehört, die Kontextautorität.

      Amazon S3 wertet eine Untermenge der Richtlinien aus, die dem übergeordneten Konto gehören. Diese Untermenge beinhaltet die Benutzerrichtlinie, die das übergeordnete Konto dem Benutzer zuordnet. Wenn das übergeordnete Objekt auch Eigentümer der Ressource in der Anfrage ist (Bucket oder Objekt), bewertet Amazon S3 gleichzeitig auch die entsprechenden Ressourcenrichtlinien (Bucket-RichtlinieACL, Bucket und ObjektACL).

      Ein Benutzer benötigt die Berechtigung von dem übergeordneten Konto, um die Operation auszuführen.

Dieser Schritt wird nur angewendet, wenn die Anforderung von einem Benutzer in einem AWS-Konto gestellt wurde. Wenn die Anfrage mit den Root-Benutzeranmeldedaten eines gestellt wird AWS-Konto, überspringt Amazon S3 diesen Schritt.

b. Bucket-Kontext — Im Bucket-Kontext wertet Amazon S3 Richtlinien aus, die demjenigen gehören AWS-Konto , dem der Bucket gehört.

   Erfolgt die Anforderung für eine Bucket-Operation, muss der Auftraggeber die Berechtigung vom Bucket-Eigentümer besitzen. Erfolgt die Anforderung für ein Objekt, wertet Amazon S3 alle Richtlinien aus, die dem Bucket-Eigentümer gehören, um zu überprüfen, ob der Bucket-Eigentümer für das Objekt eine explizite Zugriffsverweigerung festgelegt hat. Wurde eine explizite Zugriffsverweigerung festgelegt, autorisiert Amazon S3 die Anforderung nicht.

c. Objektkontext – Erfolgt die Anforderung für ein Objekt, wertet Amazon S3 die Untermenge der Richtlinien aus, die dem Objekteigentümer gehören.

Im Folgenden finden Sie einige Beispielszenarien, die veranschaulichen, wie Amazon S3 eine Anfrage autorisiert.

Example — Der Anforderer ist ein Principal IAM

Wenn es sich bei dem Antragsteller AWS-Konto um einen IAM Principal handelt, muss Amazon S3 feststellen, ob das Mutterunternehmen, dem der Principal angehört, dem Principal die erforderliche Genehmigung zur Durchführung des Vorgangs erteilt hat. Erfolgt die Anforderung darüber hinaus für eine Bucket-Operation, wie beispielsweise eine Anforderung, den Bucket-Inhalt aufzulisten, muss Amazon S3 prüfen, ob der Bucket-Eigentümer dem Auftraggeber die Berechtigung erteilt hat, die Operation auszuführen. Um einen bestimmten Vorgang mit einer Ressource durchzuführen, benötigt ein IAM Principal die Genehmigung sowohl von der Muttergesellschaft, AWS-Konto zu der sie gehört, als auch von der Person AWS-Konto , der die Ressource gehört.

Example — Der Anforderer ist ein IAM Principal — Wenn es sich bei der Anfrage um einen Vorgang für ein Objekt handelt, das dem Bucket-Besitzer nicht gehört

Wenn es sich bei der Anfrage um einen Vorgang mit einem Objekt handelt, das nicht dem Bucket-Besitzer gehört, muss Amazon S3 nicht nur sicherstellen, dass der Anforderer über die Berechtigungen des Objekteigentümers verfügt, sondern auch die Bucket-Richtlinie überprüfen, um sicherzustellen, dass der Bucket-Besitzer nicht die explizite Ablehnung für das Objekt festgelegt hat. Ein Bucket-Eigentümer (der die Rechnung zahlt) kann explizit den Zugriff auf Objekte im Bucket

verweigern, unabhängig davon, wem dieser gehört. Der Bucket-Eigentümer kann auch ein beliebiges Objekt im Bucket löschen.

Wenn ein anderer Benutzer ein Objekt in Ihren S3-Bucket AWS-Konto hochlädt, besitzt dieses Konto (der Objektschreiber) standardmäßig das Objekt, hat Zugriff darauf und kann anderen Benutzern über Zugriffskontrolllisten (ACLs) Zugriff darauf gewähren. Mit Object Ownership können Sie dieses Standardverhalten so ändern, dass ACLs es deaktiviert ist und Sie als Bucket-Besitzer automatisch Eigentümer aller Objekte in Ihrem Bucket sind. Daher basiert die Zugriffskontrolle für Ihre Daten auf Richtlinien wie IAM Benutzerrichtlinien, S3-Bucket-Richtlinien, Virtual Private Cloud (VPC) -Endpunktrichtlinien und AWS Organizations Dienststeuerungsrichtlinien (SCPs). Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

Weitere Informationen darüber, wie Amazon S3 Zugriffsrichtlinien zur Autorisierung oder Ablehnung von Anfragen für Bucket-Vorgänge und Objekt-Vorgänge bewertet, finden Sie in den folgenden Themen:

Themen

- Wie Amazon S3 eine Anforderung für eine Bucket-Operation autorisiert
- Wie Amazon S3 eine Anforderung für eine Objekt-Operation autorisiert

Wie Amazon S3 eine Anforderung für eine Bucket-Operation autorisiert

Wenn Amazon S3 eine Anfrage für eine Bucket-Operation erhält, wandelt Amazon S3 alle relevanten Berechtigungen in eine Reihe von Richtlinien um, die zur Laufzeit ausgewertet werden sollen. Zu den relevanten Berechtigungen gehören ressourcenbasierte Berechtigungen (z. B. Bucket-Richtlinien und Bucket-Zugriffskontrolllisten) und Benutzerrichtlinien, falls die Anfrage von einem IAM Principal stammt. Amazon S3 bewertet dann den resultierenden Satz von Richtlinien in einer Reihe von Schritten entsprechend einem bestimmten Kontext — Benutzerkontext oder Bucket-Kontext:

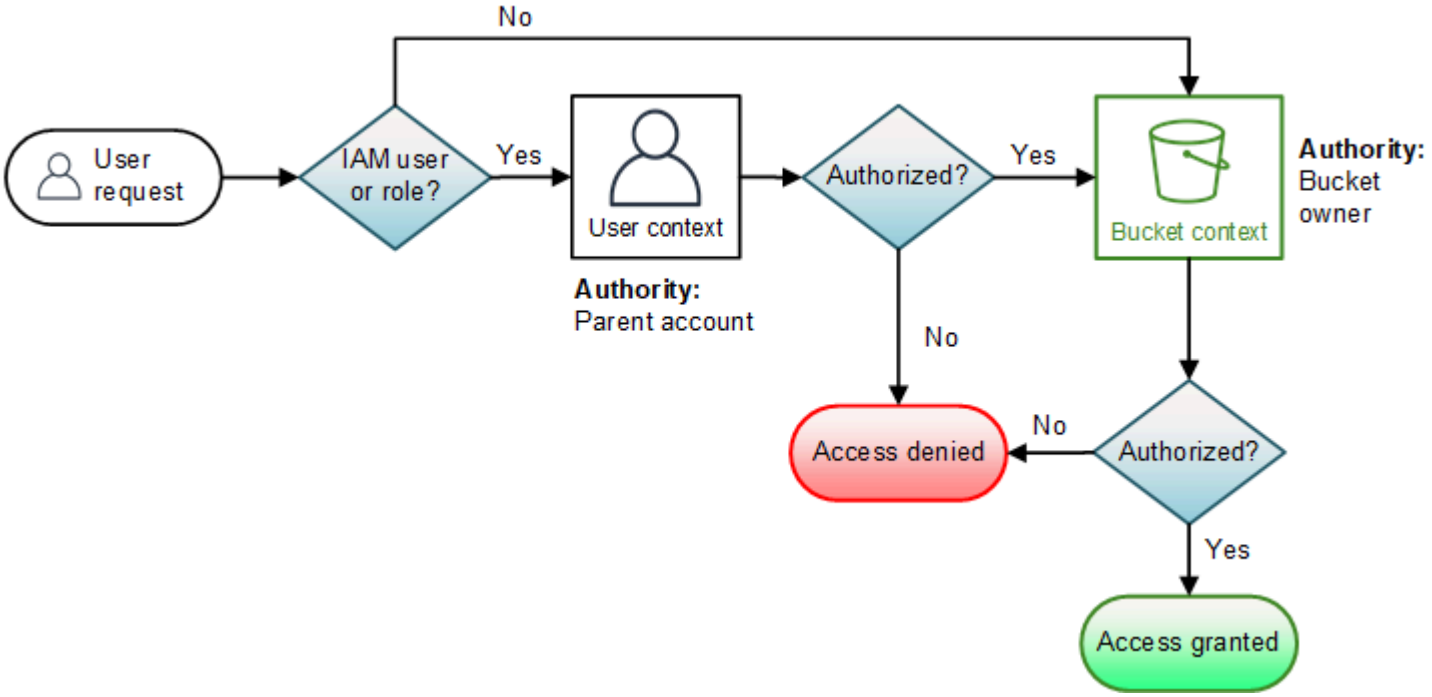
1. Benutzerkontext — Wenn der Anforderer ein IAM Principal ist, benötigt der Principal die Genehmigung des übergeordneten AWS-Konto Unternehmens, dem er angehört. In diesem Schritt wertet Amazon S3 eine Untermenge der Richtlinien aus, die dem übergeordneten Konto gehören (auch als Kontextautorität bezeichnet). Diese Richtlinienuntermenge beinhaltet die Benutzerrichtlinie, die das übergeordnete Konto dem Prinzipal zuordnet. Wenn das Elternteil auch Eigentümer der Ressource in der Anfrage ist (in diesem Fall der Bucket), bewertet Amazon S3 gleichzeitig auch die entsprechenden Ressourcenrichtlinien (Bucket-Richtlinie und BucketACL). Immer wenn eine Anforderung für eine Bucket-Operation gemacht wird, zeichnen die Server-

Zugriffsprotokolle die kanonische Benutzer-ID des Anforderers auf. Weitere Informationen finden Sie unter Protokollieren von Anfragen mit Server-Zugriffsprotokollierung.

2. Bucket-Kontext – Der Anforderer muss die Berechtigung vom Bucket-Eigentümer besitzen, eine spezifische Bucket-Operation auszuführen. In diesem Schritt bewertet Amazon S3 eine Teilmenge von Richtlinien, die demjenigen gehören, dem der AWS-Konto Bucket gehört.

   Der Bucket-Besitzer kann die Erlaubnis mithilfe einer Bucket-Richtlinie oder eines Buckets erteilen. ACL Wenn der AWS-Konto Besitzer des Buckets auch das übergeordnete Konto eines IAM Prinzipals ist, kann er Bucket-Berechtigungen in einer Benutzerrichtlinie konfigurieren.

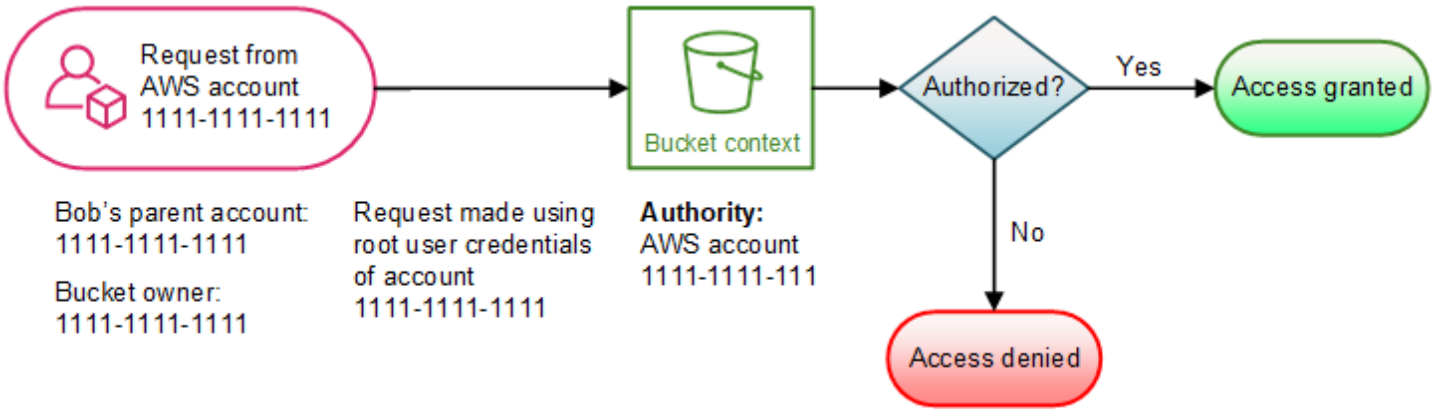
Nachfolgend sehen Sie eine grafische Darstellung der kontextbasierten Auswertung für Bucket-Operationen.



Die folgenden Beispiele veranschaulichen die Auswertungslogik.

Beispiel 1: Vom Bucket-Eigentümer angeforderte Bucket-Operation

In diesem Beispiel sendet der Bucket-Eigentümer eine Anforderung einer Bucket-Operation unter Verwendung der Root-Anmeldeinformationen des AWS-Konto.

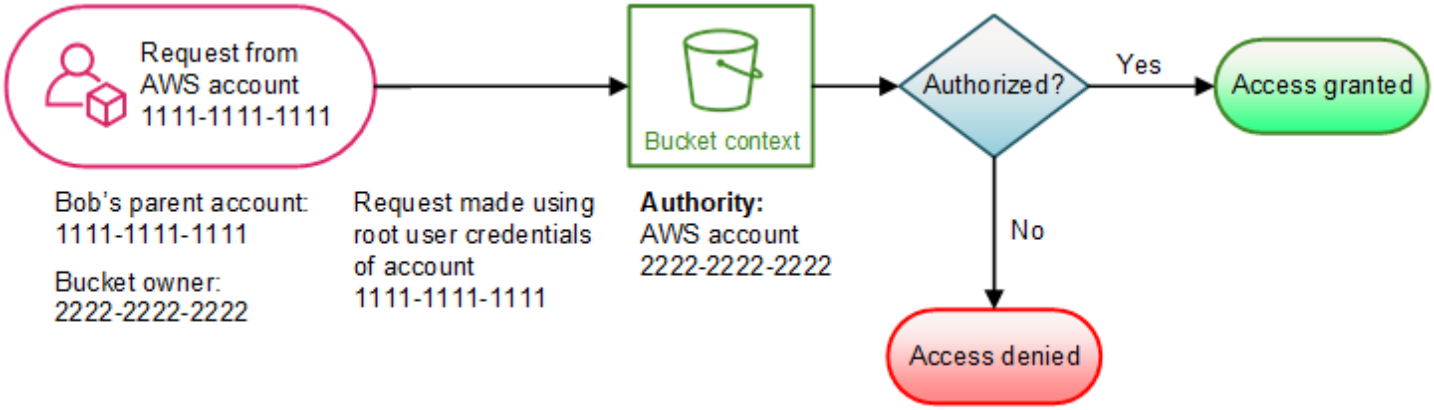


Amazon S3 führt die Kontextauswertung wie folgt durch:

1. Da die Anforderung unter Verwendung der Anmeldeinformationen des Root-Benutzers eines AWS-Konto erfolgt, wird der Benutzerkontext nicht ausgewertet.
2. Im Bucket-Kontext überprüft Amazon S3 die Bucket-Richtlinie, um festzustellen, ob der Auftraggeber die Berechtigung besitzt, die Operation auszuführen. Amazon S3 autorisiert die Anforderung.

Beispiel 2: Bucket-Vorgang, der von einem Benutzer angefordert wurde AWS-Konto , der nicht der Bucket-Besitzer ist

In diesem Beispiel wird eine Anforderung unter Verwendung der Anmeldeinformationen des Root-Benutzers des AWS-Konto 1111-1111-1111 für eine Bucket-Operation gestellt, die dem AWS-Konto 2222-2222-2222 gehört. An dieser Anfrage sind keine IAM Benutzer beteiligt.



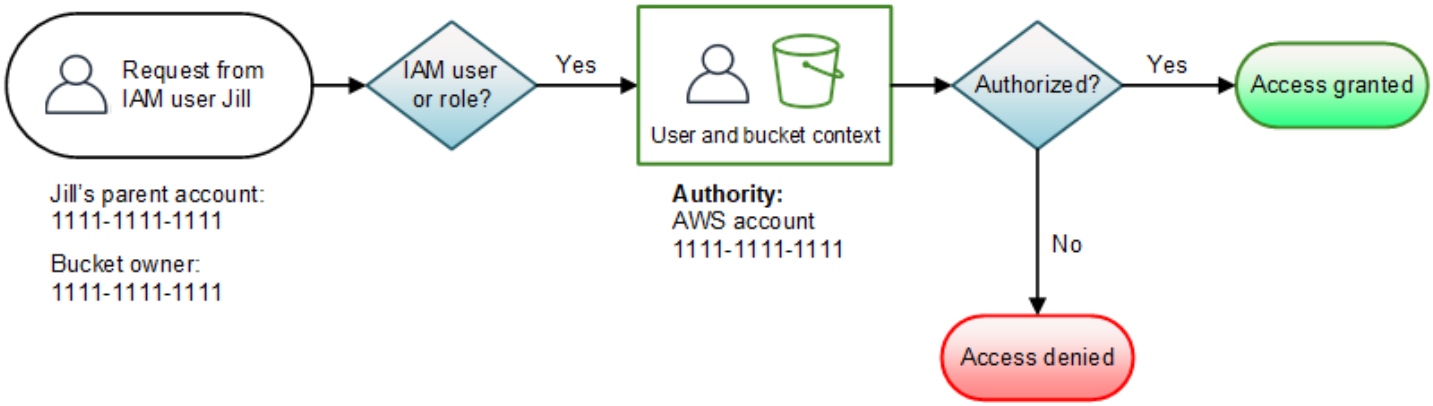
In diesem Beispiel bewertet Amazon S3 den Kontext wie folgt:

1. Da die Anfrage unter Verwendung der Root-Benutzeranmeldedaten von gestellt wird AWS-Konto, wird der Benutzerkontext nicht ausgewertet.

2. Im Bucket-Kontext wertet Amazon S3 die Bucket-Richtlinie aus. Wenn der Bucket-Besitzer (AWS-Konto 2222-2222-2222) AWS-Konto 1111-1111 nicht autorisiert hat, den angeforderten Vorgang auszuführen, lehnt Amazon S3 die Anfrage ab. Andernfalls genehmigt Amazon S3 die Anforderung und führt die Operation aus.

Beispiel 3: Bucket-Vorgang, der von einem Principal angefordert wurde, dessen Elternteil auch der Bucket-Besitzer ist IAM AWS-Konto

Im Beispiel wird die Anfrage von Jill gesendet, einer IAM Benutzerin in AWS-Konto 1111-1111-1111, der auch der Bucket gehört.



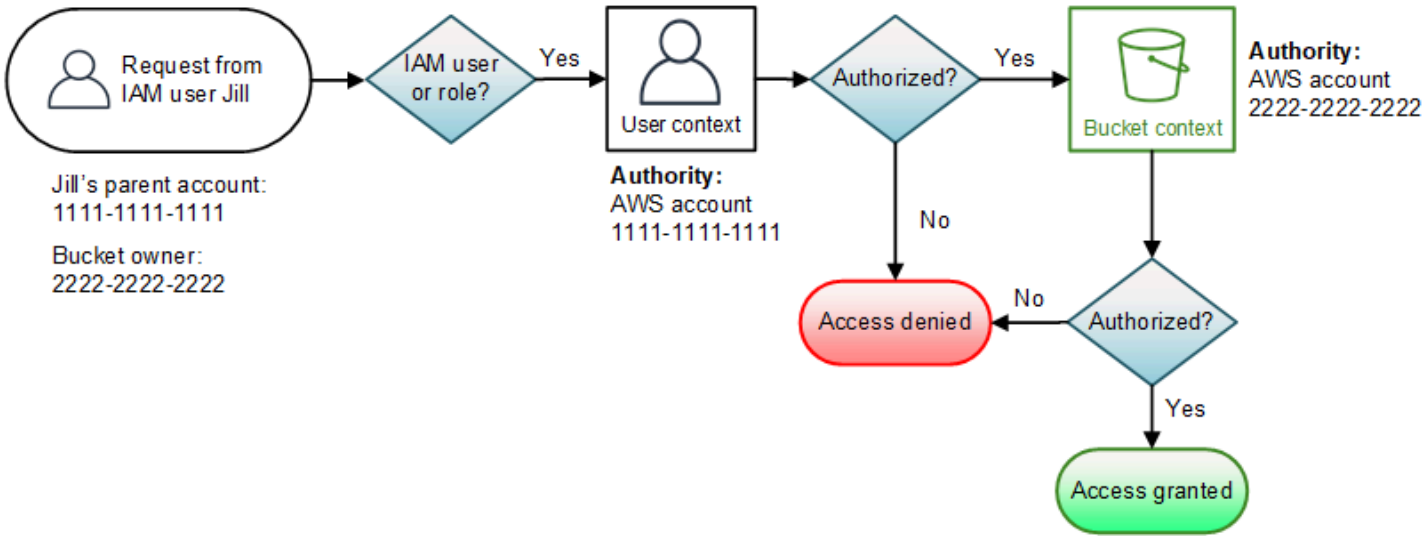
Amazon S3 führt die folgende Kontextauswertung durch:

1. Da die Anfrage von einem IAM Principal stammt, bewertet Amazon S3 im Benutzerkontext alle Richtlinien, die dem übergeordneten Unternehmen gehören, AWS-Konto um festzustellen, ob Jill berechtigt ist, den Vorgang auszuführen.

   In diesem Beispiel ist das übergeordnete Objekt AWS-Konto 1111-1111-1111, zu dem der Principal gehört, auch der Bucket-Besitzer. Daher bewertet Amazon S3 zusätzlich zur Benutzerrichtlinie auch die Bucket-Richtlinie und den Bucket ACL im selben Kontext, da sie zu demselben Konto gehören.
2. Da Amazon S3 die Bucket-Richtlinie und den Bucket ACL als Teil des Benutzerkontextes bewertet hat, bewertet es den Bucket-Kontext nicht.

Beispiel 4: Bucket-Vorgang, der von einem IAM Principal angefordert wurde, dessen übergeordnetes Element nicht der Bucket-Besitzer AWS-Konto ist

In diesem Beispiel wird die Anfrage von Jill gesendet, einer IAM Benutzerin, deren übergeordnetes Element 1111-1111-1111 AWS-Konto ist, der Bucket jedoch einem anderen gehört, 2222-2222-2222. AWS-Konto



Jill benötigt Berechtigungen sowohl vom Elternteil AWS-Konto als auch vom Bucket-Besitzer. Amazon S3 wertet den Kontext wie folgt aus:

1. Da die Anfrage von einem IAM Principal stammt, bewertet Amazon S3 den Benutzerkontext, indem es die vom Konto verfassten Richtlinien überprüft, um sicherzustellen, dass Jill über die erforderlichen Berechtigungen verfügt. Wenn Jill die Erlaubnis hat, bewertet Amazon S3 den Bucket-Kontext weiter. Wenn Jill keine Genehmigung hat, lehnt sie die Anfrage ab.
2. Im Bucket-Kontext überprüft Amazon S3, ob der Bucket-Besitzer 2222-2222-2222 Jill (oder ihren Eltern AWS-Konto) die Erlaubnis erteilt hat, den angeforderten Vorgang durchzuführen. Wenn sie über diese Genehmigung verfügt, erteilt Amazon S3 der Anfrage und führt den Vorgang durch. Andernfalls lehnt Amazon S3 die Anforderung ab.

Wie Amazon S3 eine Anforderung für eine Objekt-Operation autorisiert

Wenn Amazon S3 eine Anfrage für einen Objektvorgang erhält, konvertiert es alle relevanten Berechtigungen — ressourcenbasierte Berechtigungen (Object Access Control List (ACL), Bucket-Richtlinie, BucketACL) und IAM Benutzerrichtlinien — in eine Reihe von Richtlinien, die zur Laufzeit ausgewertet werden. Anschließend wertet es in mehreren Schritten die resultierende

Richtlinienmenge aus. In jedem Schritt bewertet es eine Teilmenge von Richtlinien in drei spezifischen Kontexten — Benutzerkontext, Bucket-Kontext und Objektkontext:

1. Benutzerkontext — Wenn es sich bei dem Anforderer um einen IAM Principal handelt, muss der Principal die Erlaubnis des übergeordneten Unternehmens haben, dem er angehört. AWS-Konto In diesem Schritt wertet Amazon S3 eine Untermenge der Richtlinien aus, die dem übergeordneten Konto gehören (auch als Kontextautorität bezeichnet). Diese Richtlinienuntermenge beinhaltet die Benutzerrichtlinie, die das übergeordnete Konto dem Prinzipal zuordnet. Wenn das übergeordnete Objekt auch Eigentümer der Ressource in der Anfrage ist (Bucket oder Objekt), bewertet Amazon S3 gleichzeitig die entsprechenden Ressourcenrichtlinien (Bucket-RichtlinieACL, Bucket und ObjektACL).

   > **Note**
   >
   > Wenn das übergeordnete Objekt AWS-Konto Eigentümer der Ressource (Bucket oder Objekt) ist, kann es seinem IAM Principal Ressourcenberechtigungen gewähren, indem es entweder die Benutzerrichtlinie oder die Ressourcenrichtlinie verwendet.

2. Bucket-Kontext – In diesem Kontext wertet Amazon S3 Richtlinien aus, die dem AWS-Konto gehören, dem der Bucket gehört.

   Wenn der AWS-Konto Besitzer des Objekts in der Anfrage nicht mit dem Bucket-Besitzer identisch ist, überprüft Amazon S3 die Richtlinien, ob der Bucket-Besitzer den Zugriff auf das Objekt explizit verweigert hat. Wurde eine explizite Zugriffsverweigerung für das Objekt festgelegt, autorisiert Amazon S3 die Anforderung nicht.

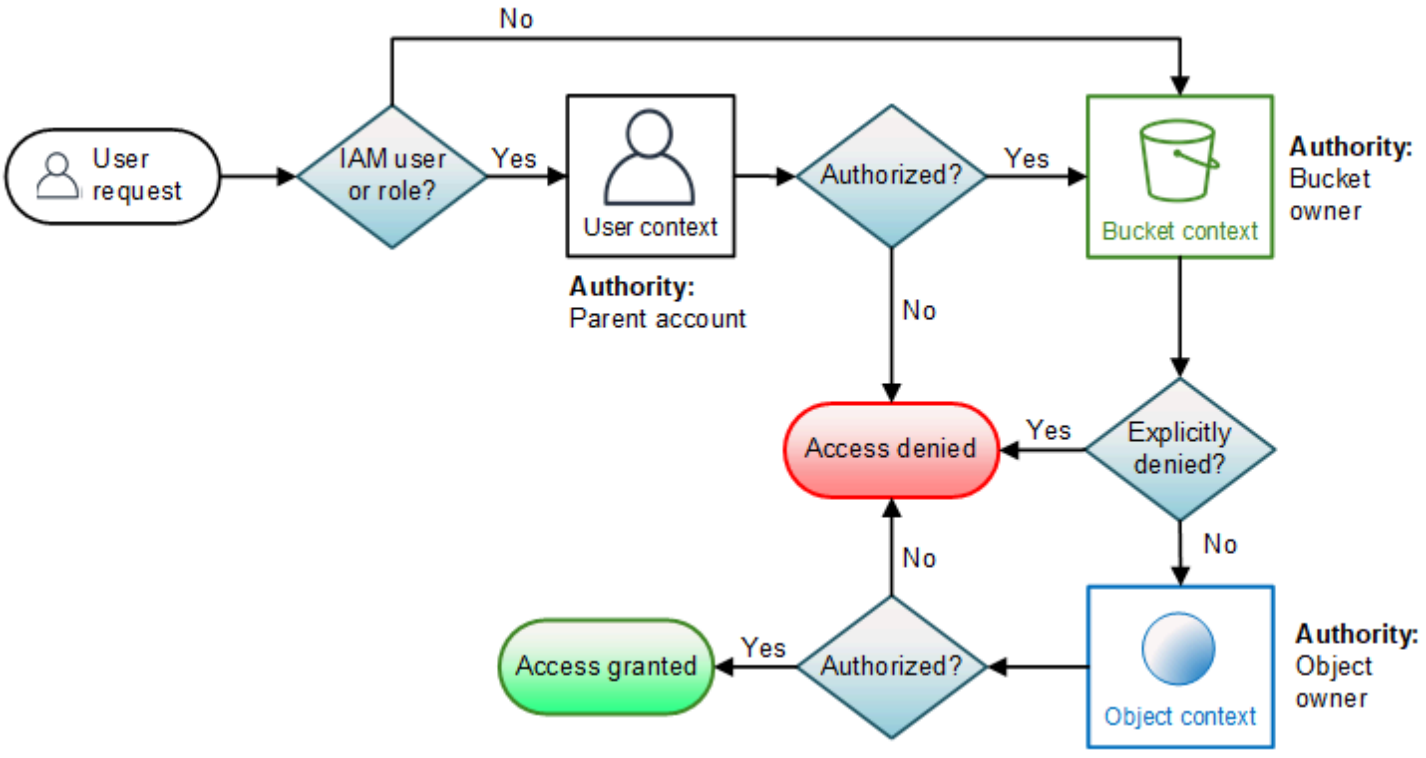
3. Objekt-Kontext – Der Anforderer muss die Berechtigung vom Objekt-Eigentümer besitzen, eine spezifische Objekt-Operation auszuführen. In diesem Schritt bewertet Amazon S3 das ObjektACL.

   > **Note**
   >
   > Ist der Bucket-Eigentümer gleich dem Objekt-Eigentümer ist, kann der Zugriff auf das Objekt in der Bucket-Richtlinie erteilt werden, die im Bucket-Kontext ausgewertet wird. Wenn es sich bei den Eigentümern um unterschiedliche Eigentümer handelt, müssen die Objekteigentümer ein Objekt verwendenACL, um Berechtigungen zu erteilen. Wenn es sich bei dem Konto AWS-Konto , dem das Objekt gehört, auch um das übergeordnete Konto handelt, zu dem der IAM Prinzipal gehört, kann es Objektberechtigungen in einer Benutzerrichtlinie konfigurieren, die im Benutzerkontext ausgewertet wird. Weitere Informationen über die Verwendung dieser Alternativen zu Zugriffsrichtlinien finden Sie

unter Exemplarische Vorgehensweisen, die Richtlinien verwenden, um den Zugriff auf Ihre Amazon S3 S3-Ressourcen zu verwalten.

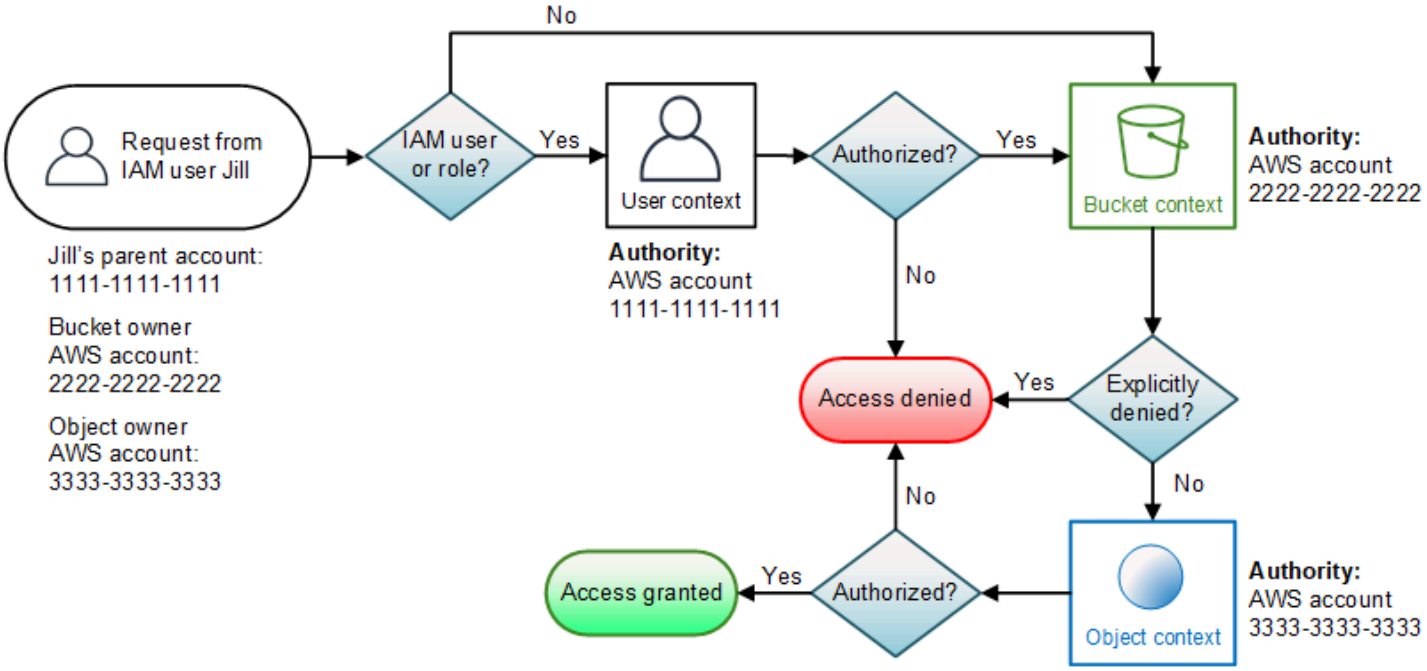
Wenn Sie als Bucket-Besitzer Eigentümer aller Objekte in Ihrem Bucket sein und Bucket-Richtlinien oder Richtlinien verwenden möchten, die auf der IAMto Verwaltung des Zugriffs auf diese Objekte basieren, können Sie die erzwungene Einstellung des Bucket-Besitzers für Objekteigentum anwenden. Mit dieser Einstellung besitzen Sie als Bucket-Eigentümer automatisch die volle Kontrolle über jedes Objekt in Ihrem Bucket. Bucket und Objekt ACLs können nicht bearbeitet werden und werden nicht mehr für den Zugriff berücksichtigt. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

Nachfolgend sehen Sie eine grafische Darstellung der kontextbasierten Auswertung für eine Objekt-Operation.



Beispiel für eine Anfrage zur Objektoperation

In diesem Beispiel sendet die IAM Benutzerin Jill, deren übergeordnetes Element 1111-1111-1111 AWS-Konto ist, eine Objektoperationsanforderung (z. B.`GetObject`) für ein Objekt, das AWS-Konto 3333-3333-3333 gehört, in einem Bucket, dem 2222-2222-2222 gehört. AWS-Konto



Jill benötigt die Erlaubnis des Elternteils AWS-Konto, des Bucket-Besitzers und des Objekteigentümers. Amazon S3 wertet den Kontext wie folgt aus:

1. Da die Anfrage von einem IAM Principal stammt, bewertet Amazon S3 den Benutzerkontext, um sicherzustellen, dass das übergeordnete Objekt AWS-Konto 1111-1111-1111 Jill die Erlaubnis erteilt hat, den angeforderten Vorgang auszuführen. Wenn sie die Berechtigung besitzt, wertet Amazon S3 den Bucket-Kontext aus. Andernfalls lehnt Amazon S3 die Anforderung ab.
2. Im Bucket-Kontext ist der Bucket-Besitzer, 2222-2222-2222, die Kontextautorität AWS-Konto . Amazon S3 wertet die Bucket-Richtlinie aus, um festzustellen, ob der Bucket-Eigentümer Jill explizit die Berechtigung entzogen hat, auf das Objekt zuzugreifen.
3. Im Objektkontext ist die Kontextautorität das AWS-Konto 3333-3333-3333, der Objekteigentümer. Amazon S3 bewertet das ObjektACL, um festzustellen, ob Jill berechtigt ist, auf das Objekt zuzugreifen. Ist dies der Fall, autorisiert Amazon S3 die Anfrage.

Erforderliche Berechtigungen für Amazon S3 API S3-Operationen

> **Note**
>
> Auf dieser Seite geht es um Amazon S3 S3-Richtlinienaktionen für Allzweck-Buckets. Weitere Informationen zu Amazon S3 S3-Richtlinienaktionen für Directory-Buckets finden Sie unterAktionen für Verzeichnis-Buckets.

Um einen API S3-Vorgang durchzuführen, benötigen Sie die richtigen Berechtigungen. Auf dieser Seite werden API S3-Operationen den erforderlichen Berechtigungen zugeordnet. Um Berechtigungen zur Ausführung eines API S3-Vorgangs zu erteilen, müssen Sie eine gültige Richtlinie (z. B. eine S3-Bucket-Richtlinie oder eine IAM identitätsbasierte Richtlinie) verfassen und entsprechende Aktionen im `Action` Element der Richtlinie angeben. Diese Aktionen werden als Richtlinienaktionen bezeichnet. Nicht jeder API S3-Vorgang wird durch eine einzige Berechtigung (eine einzelne Richtlinienaktion) repräsentiert, und einige Berechtigungen (einige Richtlinienaktionen) sind für viele verschiedene API Operationen erforderlich.

Wenn Sie Richtlinien erstellen, müssen Sie das `Resource` Element auf der Grundlage des richtigen Ressourcentyps angeben, der für die entsprechenden Amazon S3 S3-Richtlinienaktionen erforderlich ist. Auf dieser Seite werden die Berechtigungen für API S3-Operationen nach Ressourcentypen kategorisiert. Weitere Informationen zu den Ressourcentypen finden Sie unter Von Amazon S3 definierte Ressourcentypen in der Service Authorization Reference. Eine vollständige Liste der Amazon S3-Richtlinienaktionen, Ressourcen und Bedingungsschlüssel zur Verwendung in Richtlinien finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference. Eine vollständige Liste der Amazon S3 API S3-Operationen finden Sie unter Amazon S3 API S3-Aktionen in der Amazon Simple Storage API Service-Referenz.

Themen

- Bucket-Operationen und Berechtigungen
- Objektoperationen und Berechtigungen
- Vorgänge und Berechtigungen des Zugriffspunkts
- Operationen und Berechtigungen für Object Lambda Access Point
- Operationen und Berechtigungen für Access Points in mehreren Regionen
- Batch-Job-Operationen und Berechtigungen
- Konfiguration, Vorgänge und Berechtigungen von S3 Storage Lens

- S3 Storage Lens gruppiert Operationen und Berechtigungen
- Kontooperationen und Berechtigungen

Bucket-Operationen und Berechtigungen

Bucket-Operationen sind API S3-Operationen, die auf dem Bucket-Ressourcentyp ausgeführt werden. Sie müssen S3-Richtlinienaktionen für Bucket-Operationen in Bucket-Richtlinien oder IAM identitätsbasierten Richtlinien angeben.

In den Richtlinien muss das `Resource` Element der Bucket Amazon Resource Name (ARN) sein. Weitere Informationen zum `Resource` Elementformat und zu Beispielrichtlinien finden Sie unterBucket-Operationen.

> **Note**
>
> Beachten Sie Folgendes, um Berechtigungen für Bucket-Operationen in Zugriffspunktrichtlinien zu erteilen:
>
> - Berechtigungen, die in einer Zugriffspunktrichtlinie für Bucket-Operationen gewährt werden, sind nur wirksam, wenn der zugrunde liegende Bucket dieselben Berechtigungen zulässt. Wenn Sie einen Access Point verwenden, müssen Sie die Zugriffskontrolle vom Bucket an den Access Point delegieren oder die gleichen Berechtigungen in der Zugriffspunkt-Richtlinie zur Richtlinie des zugrunde liegenden Buckets hinzufügen.
> - In Zugriffspunktrichtlinien, die Berechtigungen für Bucket-Operationen gewähren, muss das `Resource` Element das `accesspoint` ARN sein. Weitere Informationen zum `Resource` Elementformat und zu Beispielrichtlinien finden Sie unterBucket-Operationen in Access Point-Richtlinien. Weitere Informationen zu Zugriffspunktrichtlinien finden Sie unterKonfiguration von IAM Richtlinien für die Verwendung von Access Points.
> - Nicht alle Bucket-Operationen werden von Access Points unterstützt. Weitere Informationen finden Sie unter Zugriffspunkt-Kompatibilität mit S3-Vorgänge.

Im Folgenden werden Bucket-Operationen und erforderliche Richtlinienaktionen zugeordnet.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
CreateBucket	(Erforderlich) `s3:CreateBucket`	Erforderlich, um einen neuen S3-Bucket zu erstellen.
	(Bedingt erforderlich) `s3:PutBucketAcl`	Erforderlich, wenn Sie die Zugriffskontrollliste (ACL) verwenden möchten, um bei einer `CreateBucket` Anfrage Berechtigungen für einen Bucket anzugeben.
	(Bedingt erforderlich)`s3:PutBucketObjectLockConfiguration` , `s3:PutBucketVersioning`	Erforderlich, wenn Sie beim Erstellen eines Buckets die Objektsperre aktivieren möchten.
	(Bedingt erforderlich) `s3:PutBucketOwnershipControls`	Erforderlich, wenn Sie beim Erstellen eines Buckets S3-Objekteigentümerschaft angeben möchten.
DeleteBucket	(Erforderlich) `s3:DeleteBucket`	Erforderlich, um einen S3-Bucket zu löschen.
DeleteBucketAnalyticsConfiguration	(Erforderlich) `s3:PutAnalyticsConfiguration`	Erforderlich, um eine S3-Analytics-Konfiguration aus einem S3-Bucket zu löschen.
DeleteBucketCors	(Erforderlich) `s3:PutBucketCORS`	Erforderlich, um die ursprungsübergreifende Resource Sharing (CORS) -Konfiguration für einen Bucket zu löschen.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
DeleteBucketEncryption	(Erforderlich) `s3:PutEncryptionConfiguration`	Erforderlich, um die Standardverschlüsselungskonfiguration für einen S3-Bucket als serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) zurückzusetzen.
DeleteBucketIntelligentTieringConfiguration	(Erforderlich) `s3:PutIntelligentTieringConfiguration`	Erforderlich, um die bestehende S3 Intelligent-Tiering-Konfiguration aus einem S3-Bucket zu löschen.
DeleteBucketInventoryConfiguration	(Erforderlich) `s3:PutInventoryConfiguration`	Erforderlich, um eine S3-Inventarkonfiguration aus einem S3-Bucket zu löschen.
DeleteBucketLifecycle	(Erforderlich) `s3:PutLifecycleConfiguration`	Erforderlich, um die S3-Lifecycle-Konfiguration für einen S3-Bucket zu löschen.
DeleteBucketMetricsConfiguration	(Erforderlich) `s3:PutMetricsConfiguration`	Erforderlich, um eine Metrikkonfiguration für die CloudWatch Amazon-Anforderungsmetriken aus einem S3-Bucket zu löschen.
DeleteBucketOwnershipControls	(Erforderlich) `s3:PutBucketOwnershipControls`	Erforderlich, um die Einstellung „Objekteigentum“ für einen S3-Bucket zu entfernen. Nach dem Entfernen wird die Einstellung Object Ownership zu`Object writer`.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
DeleteBucketPolicy	(Erforderlich) `s3:DeleteBucketPolicy`	Erforderlich, um die Richtlinie eines S3-Buckets zu löschen.
DeleteBucketReplication	(Erforderlich) `s3:PutReplicationConfiguration`	Erforderlich, um die Replikationskonfiguration eines S3-Buckets zu löschen.
DeleteBucketTagging	(Erforderlich) `s3:PutBucketTagging`	Erforderlich, um Tags aus einem S3-Bucket zu löschen.
DeleteBucketWebsite	(Erforderlich) `s3:DeleteBucketWebsite`	Erforderlich, um die Website-Konfiguration für einen S3-Bucket zu entfernen.
DeletePublicAccessBlock(Bucket-Ebene)	(Erforderlich) `s3:PutBucketPublicAccessBlock`	Erforderlich, um die Konfiguration zum Blockieren des öffentlichen Zugriffs für einen S3-Bucket zu entfernen.
GetBucketAccelerateConfiguration	(Erforderlich) `s3:GetAccelerateConfiguration`	Erforderlich, um die Accelerate-Unterressource zu verwenden, um den Amazon S3 Transfer Acceleration Acceleration-Status eines Buckets zurückzugeben, der entweder aktiviert oder ausgesetzt ist.
GetBucketAcl	(Erforderlich) `s3:GetBucketAcl`	Erforderlich, um die Zugriffskontrollliste (ACL) eines S3-Buckets zurückzugeben.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
GetBucketAnalyticsConfiguration	(Erforderlich) `s3:GetAnalyticsConfiguration`	Erforderlich, um eine Analytics-Konfiguration zurückzugeben, die durch die Analytics-Konfigurations-ID aus einem S3-Bucket identifiziert wurde.
GetBucketCors	(Erforderlich) `s3:GetBucketCORS`	Erforderlich, um die ursprungsübergreifende Resource Sharing (CORS) -Konfiguration für einen S3-Bucket zurückzugeben.
GetBucketEncryption	(Erforderlich) `s3:GetEncryptionConfiguration`	Erforderlich, um die Standardverschlüsselungskonfiguration für einen S3-Bucket zurückzugeben.
GetBucketIntelligentTieringConfiguration	(Erforderlich) `s3:GetIntelligentTieringConfiguration`	Erforderlich, um die S3 Intelligent-Tiering-Konfiguration eines S3-Buckets abzurufen.
GetBucketInventoryConfiguration	(Erforderlich) `s3:GetInventoryConfiguration`	Erforderlich, um eine Inventarkonfiguration zurückzugeben, die anhand der Inventarkonfigurations-ID aus dem Bucket identifiziert wurde.
GetBucketLifecycle	(Erforderlich) `s3:GetLifecycleConfiguration`	Erforderlich, um die S3-Lifecycle-Konfiguration des Buckets zurückzugeben.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
GetBucketLocation	(Erforderlich) `s3:GetBucketLocation`	Erforderlich, um das zurückzugeben AWS-Region , in dem sich ein S3-Bucket befindet.
GetBucketLogging	(Erforderlich) `s3:GetBucketLogging`	Erforderlich, um den Protokollierungsstatus eines S3-Buckets und die Berechtigungen zurückzugeben, über die Benutzer verfügen, um diesen Status anzuzeigen und zu ändern.
GetBucketMetricsConfiguration	(Erforderlich) `s3:GetMetricsConfiguration`	Erforderlich, um eine Metrikkonfiguration abzurufen , die durch die Metrik-Konfigurations-ID aus dem Bucket angegeben ist.
GetBucketNotificationConfiguration	(Erforderlich) `s3:GetBucketNotification`	Erforderlich, um die Benachrichtigungskonfiguration eines S3-Buckets zurückzugeben.
GetBucketOwnershipControls	(Erforderlich) `s3:GetBucketOwnershipControls`	Erforderlich, um die Einstellung „Objektbesitz“ für einen S3-Bucket abzurufen.
GetBucketPolicy	(Erforderlich) `s3:GetBucketPolicy`	Erforderlich, um die Richtlinie eines S3-Buckets zurückzugeben.
GetBucketPolicyStatus	(Erforderlich) `s3:GetBucketPolicyStatus`	Erforderlich, um den Richtlinienstatus für einen S3-Bucket abzurufen, der angibt, ob der Bucket öffentlich ist.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
GetBucketReplication	(Erforderlich) `s3:GetReplicationConfiguration`	Erforderlich, um die Replikationskonfiguration eines S3-Buckets zurückzugeben.
GetBucketRequestPayment	(Erforderlich) `s3:GetBucketRequestPayment`	Erforderlich, um die Konfiguration der Zahlungsanfrage für einen S3-Bucket zurückzugeben.
GetBucketVersioning	(Erforderlich) `s3:GetBucketVersioning`	Erforderlich, um den Versionsstatus eines S3-Buckets zurückzugeben.
GetBucketTagging	(Erforderlich) `s3:GetBucketTagging`	Erforderlich, um den Tagsatz zurückzugeben, der einem S3-Bucket zugeordnet ist.
GetBucketWebsite	(Erforderlich) `s3:GetBucketWebsite`	Erforderlich, um die Website-Konfiguration für einen S3-Bucket zurückzugeben.
GetObjectLockConfiguration	(Erforderlich) `s3:GetBucketObjectLockConfiguration`	Erforderlich, um die Object Lock-Konfiguration für einen S3-Bucket abzurufen.
GetPublicAccessBlock(Bucket-Ebene)	(Erforderlich) `s3:GetBucketPublicAccessBlock`	Erforderlich, um die Konfiguration für den öffentlichen Blockzugriff für einen S3-Bucket abzurufen.
HeadBucket	(Erforderlich) `s3:ListBucket`	Erforderlich, um festzustellen, ob ein Bucket vorhanden ist und ob Sie berechtigt sind, darauf zuzugreifen.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
ListBucketAnalyticsConfigurations	(Erforderlich) `s3:GetAnalyticsConfiguration`	Erforderlich, um die Analytics-Konfigurationen für einen S3-Bucket aufzulisten.
ListBucketIntelligentTieringConfigurations	(Erforderlich) `s3:GetIntelligentTieringConfiguration`	Erforderlich, um die S3 Intelligent-Tiering-Konfigurationen eines S3-Buckets aufzulisten.
ListBucketInventoryConfigurations	(Erforderlich) `s3:GetInventoryConfiguration`	Erforderlich, um eine Liste der Inventarkonfigurationen für einen S3-Bucket zurückzugeben.
ListBucketMetricsConfigurations	(Erforderlich) `s3:GetMetricsConfiguration`	Erforderlich, um die Metrikkonfigurationen für einen S3-Bucket aufzulisten.
ListObjects	(Erforderlich) `s3:ListBucket`	Erforderlich, um einige oder alle (bis zu 1.000) Objekte in einem S3-Bucket aufzulisten.
	(Bedingt erforderlich) `s3:GetObjectAcl`	Erforderlich, wenn Sie Informationen zum Objekteigentümer anzeigen möchten.
ListObjectsV2	(Erforderlich) `s3:ListBucket`	Erforderlich, um einige oder alle (bis zu 1.000) Objekte in einem S3-Bucket aufzulisten.
	(Bedingt erforderlich) `s3:GetObjectAcl`	Erforderlich, wenn Sie Informationen zum Objekteigentümer anzeigen möchten.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
ListObjectVersions	(Erforderlich) `s3:ListBucketVersions`	Erforderlich, um Metadaten zu allen Versionen von Objekten in einem S3-Bucket abzurufen.
PutBucketAccelerateConfiguration	(Erforderlich) `s3:PutAccelerateConfiguration`	Erforderlich, um die Beschleunigungskonfiguration eines vorhandenen Buckets festzulegen.
PutBucketAcl	(Erforderlich) `s3:PutBucketAcl`	Erforderlich, um Zugriffssteuerungslisten (ACLs) zu verwenden, um die Berechtigungen für einen vorhandenen Bucket festzulegen.
PutBucketAnalyticsConfiguration	(Erforderlich) `s3:PutAnalyticsConfiguration`	Erforderlich, um eine Analytics-Konfiguration für einen S3-Bucket festzulegen.
PutBucketCors	(Erforderlich) `s3:PutBucketCORS`	Erforderlich, um die Cross-Origin Resource Sharing (CORS)-Konfiguration für einen S3-Bucket festzulegen.
PutBucketEncryption	(Erforderlich) `s3:PutEncryptionConfiguration`	Erforderlich, um die Standardverschlüsselung für einen S3-Bucket zu konfigurieren.
PutBucketIntelligentTieringConfiguration	(Erforderlich) `s3:PutIntelligentTieringConfiguration`	Erforderlich, um die S3 Intelligent-Tiering-Konfiguration in einen S3-Bucket zu übernehmen.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
PutBucketInventoryConfiguration	(Erforderlich) `s3:PutInventoryConfiguration`	Erforderlich, um einem S3-Bucket eine Inventarkonfiguration hinzuzufügen.
PutBucketLifecycle	(Erforderlich) `s3:PutLifecycleConfiguration`	Erforderlich, um eine neue S3-Lifecycle-Konfiguration zu erstellen oder eine bestehende Lifecycle-Konfiguration für einen S3-Bucket zu ersetzen.
PutBucketLogging	(Erforderlich) `s3:PutBucketLogging`	Erforderlich, um die Protokollierungsparameter für einen S3-Bucket festzulegen und Berechtigungen dafür anzugeben, wer die Protokollierungsparameter anzeigen und ändern kann.
PutBucketMetricsConfiguration	(Erforderlich) `s3:PutMetricsConfiguration`	Erforderlich, um eine Metrikkonfiguration für die CloudWatch Amazon-Anforderungsmetriken eines S3-Buckets festzulegen oder zu aktualisieren.
PutBucketNotificationConfiguration	(Erforderlich) `s3:PutBucketNotification`	Erforderlich, um Benachrichtigungen über bestimmte Ereignisse für einen S3-Bucket zu aktivieren.
PutBucketOwnershipControls	(Erforderlich) `s3:PutBucketOwnershipControls`	Erforderlich, um die Einstellung „Objektbesitz“ für einen S3-Bucket zu erstellen oder zu ändern.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
PutBucketPolicy	(Erforderlich) `s3:PutBucketPolicy`	Erforderlich, um eine S3-Bucket-Richtlinie auf einen Bucket anzuwenden.
PutBucketReplication	(Erforderlich) `s3:PutReplicationConfiguration`	Erforderlich, um eine neue Replikationskonfiguration zu erstellen oder eine bestehende für einen S3-Bucket zu ersetzen.
PutBucketRequestPayment	(Erforderlich) `s3:PutBucketRequestPayment`	Erforderlich, um die Konfiguration der Zahlungsanfrage für einen Bucket festzulegen.
PutBucketTagging	(Erforderlich) `s3:PutBucketTagging`	Erforderlich, um einem S3-Bucket eine Reihe von Tags hinzuzufügen.
PutBucketVersioning	(Erforderlich) `s3:PutBucketVersioning`	Erforderlich, um den Versionsstatus eines S3-Buckets festzulegen.
PutBucketWebsite	(Erforderlich) `s3:PutBucketWebsite`	Erforderlich, um einen Bucket als Website zu konfigurieren und die Konfiguration der Website festzulegen.
PutObjectLockConfiguration	(Erforderlich) `s3:PutBucketObjectLockConfiguration`	Erforderlich, um die Object Lock-Konfiguration in einem S3-Bucket zu platzieren.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
PutPublicAccessBlock(Bucket-Ebene)	(Erforderlich) `s3:PutBucketPublicAccessBlock`	Erforderlich, um die Konfiguration für den öffentlichen Blockzugriff für einen S3-Bucket zu erstellen oder zu ändern.

Objektoperationen und Berechtigungen

Objektoperationen sind API S3-Operationen, die auf dem Objektressourcentyp ausgeführt werden. Sie müssen S3-Richtlinienaktionen für Objektoperationen in ressourcenbasierten Richtlinien (wie Bucket-Richtlinien, Zugriffspunktrichtlinien, Zugriffspunktrichtlinien für mehrere Regionen, VPC Endpunktrichtlinien) oder IAM identitätsbasierten Richtlinien angeben.

In den Richtlinien muss das `Resource` Element das Objekt sein. ARN Weitere Informationen zum `Resource` Elementformat und zu Beispielrichtlinien finden Sie unterObjektoperationen.

> **Note**
>
> - AWS KMS Richtlinienaktionen (`kms:GenerateDataKey`und`kms:Decrypt`) gelten nur für den AWS KMS Ressourcentyp und müssen in IAM identitätsbasierten Richtlinien und AWS KMS ressourcenbasierten Richtlinien (Schlüsselrichtlinien)AWS KMS angegeben werden. Sie können keine AWS KMS Richtlinienaktionen in ressourcenbasierten S3-Richtlinien angeben, z. B. in S3-Bucket-Richtlinien.
> - Wenn Sie Access Points verwenden, um den Zugriff auf Objektoperationen zu steuern, können Sie Zugriffspunktrichtlinien verwenden. Beachten Sie Folgendes, um Berechtigungen für Objektoperationen in Zugriffspunktrichtlinien zu erteilen:
>   - In Zugriffspunktrichtlinien, die Berechtigungen für Objektoperationen gewähren, muss das `Resource` Element das Element ARNs für Objekte sein, auf die über einen Zugriffspunkt zugegriffen wird. Weitere Informationen zum `Resource` Elementformat und zu Beispielrichtlinien finden Sie unterObjektoperationen in Zugriffspunktrichtlinien.
>   - Nicht alle Objektoperationen werden von Access Points unterstützt. Weitere Informationen finden Sie unter Zugriffspunkt-Kompatibilität mit S3-Vorgänge.

- Nicht alle Objektoperationen werden von Access Points mit mehreren Regionen unterstützt. Weitere Informationen finden Sie unter Kompatibilität von Multi-Region Access Points mit S3-Operationen.

Im Folgenden werden Objektoperationen und erforderliche Richtlinienaktionen zugeordnet.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
AbortMultipartUpload	(Erforderlich) `s3:AbortMultipartUpload`	Erforderlich, um einen mehrteiligen Upload abzubrechen.
CompleteMultipartUpload	(Erforderlich) `s3:PutObject`	Erforderlich, um einen mehrteiligen Upload abzuschließen.
	(Bedingt erforderlich) `kms:Decrypt`	Erforderlich, wenn Sie einen mehrteiligen Upload für ein vom AWS KMS Kunden verwaltetes verschlüsseltes Schlüsselobjekt abschließen möchten.
CopyObject	Für das Quellobjekt:	Für das Quellobjekt:
	(Erforderlich) Entweder `s3:GetObject` oder `s3:GetObjectVersion`	• `s3:GetObject` — Erforderlich, wenn Sie ein Objekt aus dem Quell-Bucket kopieren möchten, ohne es `versionId` in der Anfrage anzugeben. • `s3:GetObjectVersion` — Erforderlich, wenn Sie eine bestimmte Version eines Objekts aus dem

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
		Quell-Bucket kopieren möchten, indem Sie dies `versionId` in der Anfrage angeben.
	(Bedingt erforderlich) `kms:Decrypt`	Erforderlich, wenn Sie ein vom AWS KMS Kunden verwaltetes verschlüsseltes Schlüsselobjekt aus dem Quell-Bucket kopieren möchten.
	Für das Zielobjekt:	Für das Zielobjekt:
	(Erforderlich) `s3:PutObject`	Erforderlich, um das kopierte Objekt in den Ziel-Bucket zu legen.
	(Bedingt erforderlich) `s3:PutObjectAcl`	Erforderlich, wenn Sie das kopierte Objekt mit der Objektzugriffskontrollliste (ACL) in den Ziel-Bucket aufnehmen möchten, wenn Sie eine `CopyObject` Anfrage stellen.
	(Bedingt erforderlich) `s3:PutObjectTagging`	Erforderlich, wenn Sie das kopierte Objekt mit Objekt-Tagging im Ziel-Bucket platzieren möchten, wenn Sie eine `CopyObject` Anfrage stellen.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
	(Bedingt erforderlich) `kms:GenerateDataKey`	Erforderlich, wenn Sie das kopierte Objekt mit einem vom AWS KMS Kunden verwaltet en Schlüssel verschlüsseln und im Ziel-Bucket ablegen möchten.
	(Bedingt erforderlich) `s3:PutObjectRetent ion`	Erforderlich, wenn Sie eine Aufbewahrungskonfiguration für Object Lock für das neue Objekt festlegen möchten.
	(Bedingt erforderlich) `s3:PutObjectLegalH old`	Erforderlich, wenn Sie dem neuen Objekt eine gesetzliche Sperre zuweisen möchten.
CreateMultipartUpload	(Erforderlich) `s3:PutObj ect`	Erforderlich, um einen mehrteiligen Upload zu erstellen.
	(Bedingt erforderlich) `s3:PutObjectAcl`	Erforderlich, wenn Sie die Berechtigungen der Objektzug riffskontrollliste (ACL) für das hochgeladene Objekt festlegen möchten.
	(Bedingt erforderlich) `s3:PutObjectTagging`	Erforderlich, wenn Sie dem hochgeladenen Objekt Objekt-Tagging (s) hinzufügen möchten.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
	(Bedingt erforderlich) `kms:GenerateDataKey`	Erforderlich, wenn Sie einen vom AWS KMS Kunden verwalteten Schlüssel verwenden möchten, um ein Objekt zu verschlüsseln, wenn Sie einen mehrteiligen Upload initiieren.
	(Bedingt erforderlich) `s3:PutObjectRetention`	Erforderlich, wenn Sie eine Object Lock-Aufbewahrungskonfiguration für das hochgeladene Objekt festlegen möchten.
	(Bedingt erforderlich) `s3:PutObjectLegalHold`	Erforderlich, wenn Sie dem hochgeladenen Objekt eine gesetzliche Sperre für die Objektsperre zuweisen möchten.
DeleteObject	(Erforderlich) Entweder `s3:DeleteObject` oder `s3:DeleteObjectVersion`	• `s3:DeleteObject` — Erforderlich, wenn Sie ein Objekt entfernen möchten, ohne es `versionId` in der Anfrage anzugeben. • `s3:DeleteObjectVersion` — Erforderlich, wenn Sie eine bestimmte Version eines Objekts entfernen möchten, indem Sie dies `versionId` in der Anforderung angeben.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
	(Bedingt erforderlich) `s3:BypassGovernanceRetention`	Erforderlich, wenn Sie ein Objekt löschen möchten, das durch den Governance-Modus für die Aufbewahrung von Object Lock geschützt ist.
DeleteObjects	(Erforderlich) Entweder `s3:DeleteObject` oder `s3:DeleteObjectVersion`	• `s3:DeleteObject` — Erforderlich, wenn Sie ein Objekt entfernen möchten, ohne es `versionId` in der Anfrage anzugeben. • `s3:DeleteObjectVersion` — Erforderlich, wenn Sie eine bestimmte Version eines Objekts entfernen möchten, indem Sie dies `versionId` in der Anforderung angeben.
	(Bedingt erforderlich) `s3:BypassGovernanceRetention`	Erforderlich, wenn Sie Objekte löschen möchten, die durch den Governance-Modus für die Aufbewahrung von Object Lock geschützt sind.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
DeleteObjectTagging	(Erforderlich) Entweder `s3:DeleteObjectTagging` oder `s3:DeleteObjectVersionTagging`	• `s3:DeleteObjectTagging` — Erforderlich, wenn Sie den gesamten Tagsatz eines Objekts entfernen möchten, ohne dies `versionId` in der Anforderung anzugeben. • `s3:DeleteObjectVersionTagging` — Erforderlich, wenn Sie Tags einer bestimmten Objektversion löschen möchten, indem Sie sie `versionId` in der Anfrage angeben.
GetObject	(Erforderlich) Entweder `s3:GetObject` oder `s3:GetObjectVersion`	• `s3:GetObject` — Erforderlich, wenn Sie ein Objekt abrufen möchten, ohne es `versionId` in der Anfrage anzugeben. • `s3:GetObjectVersion` — Erforderlich, wenn Sie eine bestimmte Version eines Objekts abrufen möchten, indem Sie es `versionId` in der Anfrage angeben.
	(Bedingt erforderlich) `kms:Decrypt`	Erforderlich, wenn Sie ein vom AWS KMS Kunden verwaltetes verschlüsseltes Schlüsselobjekt abrufen und entschlüsseln möchten.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
	(Bedingt erforderlich) `s3:GetObjectTagging`	Erforderlich, wenn Sie das Tag-Set eines Objekts abrufen möchten, wenn Sie eine Anfrage stellen. `GetObject`
	(Bedingt erforderlich) `s3:GetObjectLegalHold`	Erforderlich, wenn Sie den aktuellen gesetzlichen Aufbewahrungsstatus eines Objekts für die Objektsperre abrufen möchten.
	(Bedingt erforderlich) `s3:GetObjectRetention`	Erforderlich, wenn Sie die Aufbewahrungseinstellungen von Object Lock für ein Objekt abrufen möchten.
GetObjectAcl	(Erforderlich) Entweder `s3:GetObjectAcl` oder `s3:GetObjectVersionAcl`	• `s3:GetObjectAcl` — Erforderlich, wenn Sie die Zugriffskontrollliste (ACL) eines Objekts abrufen möchten, ohne dies `versionId` in der Anforderung anzugeben. • `s3:GetObjectVersionAcl` — Erforderlich, wenn Sie die Zugriffskontrollliste (ACL) eines Objekts durch Angabe `versionId` in der Anforderung abrufen möchten.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
GetObjectAttributes	(Erforderlich) Entweder `s3:GetObject` oder `s3:GetObjectVersion`	• `s3:GetObject` — Erforderlich, wenn Sie Attribute abrufen möchten, die sich auf ein Objekt beziehen, ohne dies `versionId` in der Anforderung anzugeben. • `s3:GetObjectVersion` — Erforderlich, wenn Sie Attribute abrufen möchten, die sich auf eine bestimmte Objektversion beziehen, indem Sie sie `versionId` in der Anforderung angeben.
	(Bedingt erforderlich) `kms:Decrypt`	Erforderlich, wenn Sie Attribute abrufen möchten, die sich auf ein vom AWS KMS Kunden verwaltetes verschlüsseltes Schlüssel objekt beziehen.
GetObjectLegalHold	(Erforderlich) `s3:GetObjectLegalHold`	Erforderlich, um den aktuellen gesetzlichen Aufbewahrungsstatus eines Objekts für Object Lock abzurufen.
GetObjectRetention	(Erforderlich) `s3:GetObjectRetention`	Erforderlich, um die Object Lock-Aufbewahrungseinstellungen für ein Objekt abzurufen.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
GetObjectTagging	(Erforderlich) Entweder `s3:GetObjectTagging` oder `s3:GetObjectVersionTagging`	• `s3:GetObjectTagging` — Erforderlich, wenn Sie den Tagsatz eines Objekts ohne Angabe `versionId` in der Anfrage abrufen möchten. • `s3:GetObjectVersionTagging` — Erforderlich, wenn Sie die Tags einer bestimmten Objektversion durch Angabe `versionId` in der Anfrage abrufen möchten.
GetObjectTorrent	(Erforderlich) `s3:GetObject`	Erforderlich, um Torrent-Dateien eines Objekts zurückzugeben.
HeadObject	(Erforderlich) `s3:GetObject`	Erforderlich, um Metadaten von einem Objekt abzurufen, ohne das Objekt selbst zurückzugeben.
	(Bedingt erforderlich) `s3:GetObjectLegalHold`	Erforderlich, wenn Sie den aktuellen gesetzlichen Aufbewahrungsstatus eines Objekts für die Objektsperre abrufen möchten.
	(Bedingt erforderlich) `s3:GetObjectRetention`	Erforderlich, wenn Sie die Aufbewahrungseinstellungen von Object Lock für ein Objekt abrufen möchten.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
ListMultipartUploads	(Erforderlich) `s3:ListBucketMultipartUploads`	Erforderlich, um laufende mehrteilige Uploads in einem Bucket aufzulisten.
ListParts	(Erforderlich) `s3:ListMultipartUploadParts`	Erforderlich, um die Teile aufzulisten, die für einen bestimmten mehrteiligen Upload hochgeladen wurden.
	(Befriedigend erforderlich) `kms:Decrypt`	Erforderlich, wenn Sie Teile eines vom AWS KMS Kunden verwalteten, verschlüsselten, mehrteiligen Uploads auflisten möchten.
PutObject	(Erforderlich) `s3:PutObject`	Erforderlich, um ein Objekt zu platzieren.
	(Bedingt erforderlich) `s3:PutObjectAcl`	Erforderlich, wenn Sie die Objektzugriffskontrollliste (ACL) bei einer `PutObject` Anfrage angeben möchten.
	(Bedingt erforderlich) `s3:PutObjectTagging`	Erforderlich, wenn Sie bei einer Anfrage Objekt-Tagging angeben möchten. `PutObject`
	(Bedingt erforderlich) `kms:GenerateDataKey`	Erforderlich, wenn Sie ein Objekt mit einem vom AWS KMS Kunden verwalteten Schlüssel verschlüsseln möchten.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
	(Bedingt erforderlich) `s3:PutObjectRetention`	Erforderlich, wenn Sie eine Aufbewahrungskonfiguration für Object Lock für ein Objekt festlegen möchten.
	(Bedingt erforderlich) `s3:PutObjectLegalHold`	Erforderlich, wenn Sie eine Object Lock-Konfiguration mit legaler Aufbewahrungsfrist auf ein bestimmtes Objekt anwenden möchten.
PutObjectAcl	(Erforderlich) Entweder `s3:PutObjectAcl` oder `s3:PutObjectVersionAcl`	• `s3:PutObjectAcl` — Erforderlich, wenn Sie die Zugriffssteuerungsliste (ACL) -Berechtigungen für ein neues oder vorhandenes Objekt festlegen möchten, ohne dies `versionId` in der Anforderung anzugeben . • `s3:PutObjectVersionAcl` — Erforderlich, wenn Sie die Zugriffsberechtigungen für die Zugriffskontrollliste (ACL) für ein neues oder vorhandenes Objekt durch Angabe `versionId` in der Anforderung festlegen möchten.
PutObjectLegalHold	(Erforderlich) `s3:PutObjectLegalHold`	Erforderlich, um eine Object Lock Legal Hold-Konfiguration auf ein Objekt anzuwenden.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
PutObjectRetention	(Erforderlich) `s3:PutObjectRetention`	Erforderlich, um eine Object Lock-Aufbewahrungskonfiguration auf ein Objekt anzuwenden.
	(Bedingt erforderlich) `s3:BypassGovernanceRetention`	Erforderlich, wenn Sie den Verwaltungsmodus einer Object Lock-Aufbewahrungskonfiguration umgehen möchten.
PutObjectTagging	(Erforderlich) Entweder `s3:PutObjectTagging` oder `s3:PutObjectVersionTagging`	• `s3:PutObjectTagging` — Erforderlich, wenn Sie den angegebenen Tagsatz auf ein Objekt setzen möchten, das bereits in einem Bucket vorhanden ist, ohne dass dies `versionId` in der Anforderung angegeben wurde. • `s3:PutObjectVersionTagging` — Erforderlich, wenn Sie den angegebenen Tagsatz auf ein Objekt setzen möchten, das bereits in einem Bucket vorhanden ist, indem Sie dies `versionId` in der Anfrage angeben.
RestoreObject	(Erforderlich) `s3:RestoreObject`	Erforderlich, um eine Kopie eines archivierten Objekts wiederherzustellen.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
SelectObjectContent	(Erforderlich) `s3:GetObject`	Erforderlich, um den Inhalt eines S3-Objekts auf der Grundlage einer einfachen strukturierten Query Language (SQL) -Anweisung zu filtern.
	(Bedingt erforderlich) `kms:Decrypt`	Erforderlich, wenn Sie den Inhalt eines S3-Objekts filtern möchten, das mit einem vom AWS KMS Kunden verwaltet en Schlüssel verschlüsselt ist.
UploadPart	(Erforderlich) `s3:PutObject`	Erforderlich, um ein Teil in einem mehrteiligen Upload hochzuladen.
	(Befriedigend erforderlich) `kms:GenerateDataKey`	Erforderlich, wenn Sie einen Upload-Teil hochladen und mit einem vom AWS KMS Kunden verwalteten Schlüssel verschlüsseln möchten.
UploadPartCopy	Für das Quellobjekt:	Für das Quellobjekt:

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
	(Erforderlich) Entweder `s3:GetObject` oder `s3:GetObjectVersion`	• `s3:GetObject` — Erforderlich, wenn Sie ein Objekt aus dem Quell-Bucket kopieren möchten, ohne es `versionId` in der Anfrage anzugeben. • `s3:GetObjectVersion` — Erforderlich, wenn Sie eine bestimmte Version eines Objekts aus dem Quell-Bucket kopieren möchten, indem Sie dies `versionId` in der Anfrage angeben.
	(Bedingt erforderlich) `kms:Decrypt`	Erforderlich, wenn Sie ein vom AWS KMS Kunden verwaltetes verschlüsseltes Schlüsselobjekt aus dem Quell-Bucket kopieren möchten.
	Für den Zielteil:	Für den Zielteil:
	(Erforderlich) `s3:PutObject`	Erforderlich, um einen mehrteiligen Upload-Teil in den Ziel-Bucket hochzuladen.
	(Bedingt erforderlich) `kms:GenerateDataKey`	Erforderlich, wenn Sie ein Teil mit einem vom AWS KMS Kunden verwalteten Schlüssel verschlüsseln möchten, wenn Sie das Teil in den Ziel-Bucket hochladen.

Vorgänge und Berechtigungen des Zugriffspunkts

Access Point-Operationen sind API S3-Operationen, die auf dem `accesspoint` Ressourcentyp ausgeführt werden. Sie müssen S3-Richtlinienaktionen für Access Point-Operationen in IAM identitätsbasierten Richtlinien angeben, nicht in Bucket-Richtlinien oder Zugriffspunktrichtlinien.

In den Richtlinien muss das `Resource` Element das sein. `accesspoint` ARN Weitere Informationen zum `Resource` Elementformat und zu Beispielrichtlinien finden Sie unterOperationen an Zugriffspunkten.

> **Note**
>
> Wenn Sie Access Points verwenden möchten, um den Zugriff auf Bucket- oder Objektoperationen zu steuern, beachten Sie Folgendes:
>
> - Informationen zur Verwendung von Access Points zur Steuerung des Zugriffs auf Bucket-Operationen finden Sie unterBucket-Operationen in Access Point-Richtlinien.
> - Informationen zur Verwendung von Access Points zur Steuerung des Zugriffs auf Objektoperationen finden Sie unterObjektoperationen in Zugriffspunktrichtlinien.
> - Weitere Informationen zur Konfiguration von Zugriffspunktrichtlinien finden Sie unterKonfiguration von IAM Richtlinien für die Verwendung von Access Points.

Im Folgenden werden die Zugriffspunktvorgänge und die erforderlichen Richtlinienaktionen zugeordnet.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
CreateAccessPoint	(Erforderlich) `s3:CreateAccessPoint`	Erforderlich, um einen Access Point zu erstellen, der einem S3-Bucket zugeordnet ist.
DeleteAccessPoint	(Erforderlich) `s3:DeleteAccessPoint`	Erforderlich, um einen Access Point zu löschen.
DeleteAccessPointPolicy	(Erforderlich) `s3:DeleteAccessPointPolicy`	Erforderlich, um eine Zugriffspunktrichtlinie zu löschen.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
GetAccessPointPolicy	(Erforderlich) `s3:GetAccessPointPolicy`	Erforderlich, um eine Zugriffspunktrichtlinie abzurufen.
GetAccessPointPolicyStatus	(Erforderlich) `s3:GetAccessPointPolicyStatus`	Erforderlich, um die Informationen darüber abzurufen, ob für den angegebenen Zugriffspunkt derzeit eine Richtlinie gilt, die den öffentlichen Zugriff ermöglicht.
PutAccessPointPolicy	(Erforderlich) `s3:PutAccessPointPolicy`	Erforderlich, um eine Zugriffspunktrichtlinie festzulegen.

Operationen und Berechtigungen für Object Lambda Access Point

Object Lambda Access Point-Operationen sind API S3-Operationen, die auf dem `objectlambdaaccesspoint` Ressourcentyp ausgeführt werden. Weitere Informationen zur Konfiguration von Richtlinien für Object Lambda Access Point-Operationen finden Sie unterKonfiguration von IAM Richtlinien für Object Lambda Access Points.

Im Folgenden finden Sie die Zuordnung der Object Lambda Access Point-Operationen und der erforderlichen Richtlinienaktionen.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
CreateAccessPointForObjectLambda	(Erforderlich) `s3:CreateAccessPointForObjectLambda`	Erforderlich, um einen Object Lambda Access Point zu erstellen.
DeleteAccessPointForObjectLambda	(Erforderlich) `s3:DeleteAccessPointForObjectLambda`	Erforderlich, um einen angegebenen Object Lambda Access Point zu löschen.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
DeleteAccessPointPolicyForObjectLambda	(Erforderlich) `s3:DeleteAccessPointPolicyForObjectLambda`	Erforderlich, um die Richtlinie auf einem angegebenen Object Lambda Access Point zu löschen.
GetAccessPointConfigurationForObjectLambda	(Erforderlich) `s3:GetAccessPointConfigurationForObjectLambda`	Erforderlich, um die Konfiguration des Object Lambda Access Points abzurufen.
GetAccessPointForObjectLambda	(Erforderlich) `s3:GetAccessPointForObjectLambda`	Erforderlich, um Informationen über den Object Lambda Access Point abzurufen.
GetAccessPointPolicyForObjectLambda	(Erforderlich) `s3:GetAccessPointPolicyForObjectLambda`	Erforderlich, um die Zugriffspunktrichtlinie zurückzugeben, die dem angegebenen Object Lambda Access Point zugeordnet ist.
GetAccessPointPolicyStatusForObjectLambda	(Erforderlich) `s3:GetAccessPointPolicyStatusForObjectLambda`	Erforderlich, um den Richtlinienstatus für eine bestimmte Object Lambda Access Point-Richtlinie zurückzugeben.
PutAccessPointConfigurationForObjectLambda	(Erforderlich) `s3:PutAccessPointConfigurationForObjectLambda`	Erforderlich, um die Konfiguration des Object Lambda Access Points festzulegen.
PutAccessPointPolicyForObjectLambda	(Erforderlich) `s3:PutAccessPointPolicyForObjectLambda`	Erforderlich, um einem angegebenen Object Lambda Access Point eine Zugriffsrichtlinie zuzuordnen.

Operationen und Berechtigungen für Access Points in mehreren Regionen

Access Point-Operationen mit mehreren Regionen sind API S3-Operationen, die auf dem `multiregionaccesspoint` Ressourcentyp ausgeführt werden. Weitere Informationen zur Konfiguration von Richtlinien für Access Point-Operationen in mehreren Regionen finden Sie unter. Beispielrichtlinien für Multi-Region Access Points

Im Folgenden finden Sie die Zuordnung von Access Point-Vorgängen in mehreren Regionen und den erforderlichen Richtlinienaktionen.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
CreateMultiRegionAccessPoint	(Erforderlich) `s3:CreateMultiRegionAccessPoint`	Erforderlich, um einen Access Point mit mehreren Regionen zu erstellen und ihn mit S3-Buckets zu verknüpfen.
DeleteMultiRegionAccessPoint	(Erforderlich) `s3:DeleteMultiRegionAccessPoint`	Erforderlich, um einen Access Point mit mehreren Regionen zu löschen.
DescribeMultiRegionAccessPointOperation	(Erforderlich) `s3:DescribeMultiRegionAccessPointOperation`	Erforderlich, um den Status einer asynchronen Anfrage zur Verwaltung eines Access Points mit mehreren Regionen abzurufen.
GetMultiRegionAccessPoint	(Erforderlich) `s3:GetMultiRegionAccessPoint`	Erforderlich, um Konfigurationsinformationen über den angegebenen Multi-Region-Access Point zurückzugeben.
GetMultiRegionAccessPointPolicy	(Erforderlich) `s3:GetMultiRegionAccessPointPolicy`	Erforderlich, um die Zugriffskontrollrichtlinie des angegebenen Multi-Region-Access Points zurückzugeben.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
GetMultiRegionAccessPointPolicyStatus	(Erforderlich) `s3:GetMultiRegionAccessPointPolicyStatus`	Erforderlich, um den Richtlinienstatus für einen bestimmten Multi-Region-Access Point zurückzugeben und festzustellen, ob der angegebene Multi-Region-Access Point über eine Zugriffskontrollrichtlinie verfügt, die öffentlichen Zugriff ermöglicht.
GetMultiRegionAccessPointRoutes	(Erforderlich) `s3:GetMultiRegionAccessPointRoutes`	Erforderlich, um die Routing-Konfiguration für einen Access Point mit mehreren Regionen zurückzugeben.
PutMultiRegionAccessPointPolicy	(Erforderlich) `s3:PutMultiRegionAccessPointPolicy`	Erforderlich, um die Zugriffskontrollrichtlinie des angegebenen Multi-Region-Access Points zu aktualisieren.
SubmitMultiRegionAccessPointRoutes	(Erforderlich) `s3:SubmitMultiRegionAccessPointRoutes`	Erforderlich, um eine aktualisierte Routenkonfiguration für einen Access Point mit mehreren Regionen einzureichen.

Batch-Job-Operationen und Berechtigungen

Auftragsoperationen (Batch Operations) sind API S3-Operationen, die auf dem `job` Ressourcentyp ausgeführt werden. Sie müssen S3-Richtlinienaktionen für Auftragsoperationen in IAM identitätsbasierten Richtlinien angeben, nicht in Bucket-Richtlinien.

In den Richtlinien muss das `Resource` Element das sein. `job` ARN Weitere Informationen zum `Resource` Elementformat und zu Beispielrichtlinien finden Sie unterBatch-Job-Operationen.

Im Folgenden werden Batch-Job-Operationen und erforderliche Richtlinienaktionen zugeordnet.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
DeleteJobTagging	(Erforderlich) `s3:DeleteJobTagging`	Erforderlich, um Tags aus einem vorhandenen S3 Batch Operations-Job zu entfernen.
DescribeJob	(Erforderlich) `s3:DescribeJob`	Erforderlich, um die Konfigurationsparameter und den Status für einen Batch Operations-Job abzurufen.
GetJobTagging	(Erforderlich) `s3:GetJobTagging`	Erforderlich, um den Tagsatz eines vorhandenen S3 Batch Operations-Jobs zurückzugeben.
PutJobTagging	(Erforderlich) `s3:PutJobTagging`	Erforderlich, um einem vorhandenen S3 Batch Operations-Job Tags hinzuzufügen oder zu ersetzen.
UpdateJobPriority	(Erforderlich) `s3:UpdateJobPriority`	Erforderlich, um die Priorität eines vorhandenen Jobs zu aktualisieren.
UpdateJobStatus	(Erforderlich) `s3:UpdateJobStatus`	Erforderlich, um den Status für den angegebenen Job zu aktualisieren.

Konfiguration, Vorgänge und Berechtigungen von S3 Storage Lens

S3 Storage Lens-Konfigurationsvorgänge sind API S3-Operationen, die auf dem `storagelensconfiguration` Ressourcentyp ausgeführt werden. Weitere Informationen zur

Konfiguration von S3 Storage Lens-Konfigurationsvorgängen finden Sie unterAmazon S3 Storage Lens-Berechtigungen einrichten.

Im Folgenden werden die Konfigurationsvorgänge von S3 Storage Lens und die erforderlichen Richtlinienaktionen zugeordnet.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
DeleteStorageLensConfiguration	(Erforderlich) `s3:DeleteStorageLensConfiguration`	Erforderlich, um die S3 Storage Lens-Konfiguration zu löschen.
DeleteStorageLensConfigurationTagging	(Erforderlich) `s3:DeleteStorageLensConfigurationTagging`	Erforderlich, um die S3 Storage Lens-Konfigurations-Tags zu löschen.
GetStorageLensConfiguration	(Erforderlich) `s3:GetStorageLensConfiguration`	Erforderlich, um die S3 Storage Lens-Konfiguration abzurufen.
GetStorageLensConfigurationTagging	(Erforderlich) `s3:GetStorageLensConfigurationTagging`	Erforderlich, um die Tags der S3 Storage Lens-Konfiguration abzurufen.
PutStorageLensConfigurationTagging	(Erforderlich) `s3:PutStorageLensConfigurationTagging`	Erforderlich, um Tags in eine bestehende S3-Speicherlinsenkonfiguration einzufügen oder zu ersetzen.

S3 Storage Lens gruppiert Operationen und Berechtigungen

Bei Operationen mit S3 Storage Lens-Gruppen handelt es sich um API S3-Operationen, die auf dem `storagelensgroup` Ressourcentyp ausgeführt werden. Weitere Informationen zur Konfiguration von Gruppenberechtigungen für S3 Storage Lens finden Sie unterBerechtigungen für Storage-Lens-Gruppen.

Im Folgenden finden Sie die Zuordnung von S3 Storage Lens-Gruppenoperationen und erforderlichen Richtlinienaktionen.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
DeleteStorageLensGroup	(Erforderlich) `s3:DeleteStorageLensGroup`	Erforderlich, um eine bestehende S3 Storage Lens-Gruppe zu löschen.
GetStorageLensGroup	(Erforderlich) `s3:GetStorageLensGroup`	Erforderlich, um die Konfigurationsdetails der S3 Storage Lens-Gruppe abzurufen.
UpdateStorageLensGroup	(Erforderlich) `s3:UpdateStorageLensGroup`	Erforderlich, um die bestehende S3 Storage Lens-Gruppe zu aktualisieren.

Kontooperationen und Berechtigungen

Kontovorgänge sind API S3-Operationen, die auf Kontoebene ausgeführt werden. Konto ist kein von Amazon S3 definierter Ressourcentyp. Sie müssen S3-Richtlinienaktionen für Kontooperationen in IAM identitätsbasierten Richtlinien angeben, nicht in Bucket-Richtlinien.

In den Richtlinien muss das `Resource` Element sein. `"*"` Weitere Informationen zu Beispielrichtlinien finden Sie unterKontovorgänge.

Im Folgenden werden Kontovorgänge und erforderliche Richtlinienaktionen zugeordnet.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
CreateJob	(Erforderlich) `s3:CreateJob`	Erforderlich, um einen neuen S3 Batch Operations-Job zu erstellen.
CreateStorageLensGroup	(Erforderlich) `s3:CreateStorageLensGroup`	Erforderlich, um eine neue S3 Storage Lens-Gruppe

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
		zu erstellen und sie der angegebenen AWS-Konto ID zuzuordnen.
	(Bedingt erforderlich) `s3:TagResource`	Erforderlich, wenn Sie eine S3 Storage Lens-Gruppe mit AWS Ressourcen-Tags erstellen möchten.
DeletePublicAccessBlock(Auf Kontoebene)	(Erforderlich) `s3:PutAccountPublicAccessBlock`	Erforderlich, um die Konfiguration zum Blockieren des öffentlichen Zugriffs aus einem zu entfernen AWS-Konto.
GetAccessPoint	(Erforderlich) `s3:GetAccessPoint`	Erforderlich, um Konfigurationsinformationen über den angegebenen Zugriffspunkt abzurufen.
GetAccessPointPolicy(Auf Kontoebene)	(Erforderlich) `s3:GetAccountPublicAccessBlock`	Erforderlich, um die Konfiguration für den blockierten öffentlichen Zugriff für einen abzurufen AWS-Konto.
ListAccessPoints	(Erforderlich) `s3:ListAccessPoints`	Erforderlich, um Access Points eines S3-Buckets aufzulisten, die einem gehören AWS-Konto.
ListAccessPointsForObjectLambda	(Erforderlich) `s3:ListAccessPointsForObjectLambda`	Erforderlich, um die Object Lambda Access Points aufzulisten.

APIOperationen	Richtlinienaktionen	Beschreibung der politischen Maßnahmen
ListBuckets	(Erforderlich) `s3:ListAllMyBuckets`	Erforderlich, um eine Liste aller Buckets zurückzugeben, die dem authentifizierten Absender der Anfrage gehören.
ListJobs	(Erforderlich) `s3:ListJobs`	Erforderlich, um aktuelle Jobs und Jobs aufzulisten, die kürzlich beendet wurden.
ListMultiRegionAccessPoints	(Erforderlich) `s3:ListMultiRegionAccessPoints`	Erforderlich, um eine Liste der Multi-Region-Access Points zurückzugeben, die derzeit mit den angegebenen AWS-Konto Access Points verknüpft sind.
ListStorageLensConfigurations	(Erforderlich) `s3:ListStorageLensConfigurations`	Erforderlich, um eine Liste der S3 Storage Lens-Konfigurationen für eine abzurufen AWS-Konto.
ListStorageLensGroups	(Erforderlich) `s3:ListStorageLensGroups`	Erforderlich, um alle S3 Storage Lens-Gruppen im angegebenen Zuhause aufzulisten AWS-Region.
PutPublicAccessBlock(Kontoebene)	(Erforderlich) `s3:PutAccountPublicAccessBlock`	Erforderlich, um die Konfiguration für den öffentlichen Zugriff blockieren für einen zu erstellen oder zu ändern AWS-Konto.
PutStorageLensConfiguration	(Erforderlich) `s3:PutStorageLensConfiguration`	Erforderlich, um eine S3-Speicherlinsenkonfiguration einzurichten.

Richtlinien und Berechtigungen in Amazon S3

Diese Seite bietet einen Überblick über Bucket- und Benutzerrichtlinien in Amazon S3 und beschreibt die grundlegenden Elemente einer AWS Identity and Access Management (IAM) -Richtlinie. Jedes aufgelistete Element verweist auf weitere Details zu diesem Element und auf Beispiele für die Verwendung dieses Elements.

Eine vollständige Liste der Amazon S3-Aktionen, Ressourcen und Bedingungen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

In ihrer einfachsten Form enthält eine Richtlinie die folgenden Elemente:

- Resource— Der Amazon S3 S3-Bucket, das Objekt, der Access Point oder der Job, für den die Richtlinie gilt. Verwenden Sie den Amazon-Ressourcennamen (ARN) des Buckets, Objekts, Access Points oder Jobs, um die Ressource zu identifizieren.

  Ein Beispiel für Operationen auf Bucket-Ebene:

  `"Resource": "arn:aws:s3:::bucket_name"`

  Beispiele für Operationen auf Objektebene:

  - `"Resource": "arn:aws:s3:::bucket_name/*"`für alle Objekte im Bucket.
  - `"Resource": "arn:aws:s3:::bucket_name/prefix/*"`für Objekte unter einem bestimmten Präfix im Bucket.

  Weitere Informationen finden Sie unter Richtlinienressourcen für Amazon S3.

- Actions— Für jede Ressource unterstützt Amazon S3 eine Reihe von Vorgängen. Sie identifizieren RessourcenVorgänge, die Sie zulassen (oder ablehnen) können, indem Sie Aktionsschlüsselwörter verwenden.

  Die `s3:ListBucket` Berechtigung ermöglicht es dem Benutzer beispielsweise, Amazon S3 zu verwenden ListObjectsV2Betrieb. (Bei der `s3:ListBucket` Berechtigung handelt es sich um einen Fall, in dem der Aktionsname nicht direkt dem Operationsnamen zugeordnet ist.) Weitere Informationen zur Verwendung von Amazon-S3-Aktionen finden Sie unter Politische Maßnahmen

für Amazon S3. Eine vollständige Liste der Amazon S3 S3-Aktionen finden Sie unter Aktionen in der Amazon Simple Storage API Service-Referenz.

- Effect— Welche Auswirkung hat, wenn der Benutzer die spezifische Aktion anfordert — dies kann entweder oder `Allow` sein. `Deny`

  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 könnten Sie z. B. sicherstellen, dass ein Benutzer nicht auf die Ressource zugreifen kann, auch wenn der Zugriff durch eine andere Richtlinie gestattet wird. Weitere Informationen finden Sie unter IAMJSONPolicy Elements: Effect im IAMBenutzerhandbuch.
- Principal— Das Konto oder der Benutzer, dem Zugriff auf die Aktionen und Ressourcen in der Erklärung gewährt wird. In einer Bucket-Richtlinie ist der Prinzipal der Benutzer, das Konto, der Service oder eine andere Entität, der/die/das der Empfänger dieser Berechtigung ist. Weitere Informationen finden Sie unter Prinzipien für Bucket-Richtlinien.
- Condition— Bedingungen für den Zeitpunkt, zu dem eine Richtlinie in Kraft ist. Sie können AWS umfassende Schlüssel und Amazon S3-spezifische Schlüssel verwenden, um Bedingungen in einer Amazon S3 S3-Zugriffsrichtlinie festzulegen. Weitere Informationen finden Sie unter Beispiele für Bucket-Richtlinien mit Bedingungsschlüsseln.

Die folgende Beispiel-Bucket-Richtlinie zeigt die Elemente`Effect`,`Principal`, `Action` und. `Resource` Diese Richtlinie ermöglicht *`Akua`* einem Benutzer im Konto*`123456789012`*, `s3:GetObjects3:GetBucketLocation`, und `s3:ListBucket` Amazon S3 S3-Berechtigungen für den *`amzn-s3-demo-bucket1`* Bucket.

```
{
    "Version": "2012-10-17",
    "Id": "ExamplePolicy01",
    "Statement": [
        {
            "Sid": "ExampleStatement01",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Akua"
            },
            "Action": [
                "s3:GetObject",
                "s3:GetBucketLocation",
                "s3:ListBucket"
```

```
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1/*",
                "arn:aws:s3:::amzn-s3-demo-bucket1"
            ]
        }
    ]
}
```

Vollständige Informationen zur Sprache der Richtlinie finden Sie unter Richtlinien und Berechtigungen IAM und Referenzinformationen zu IAM JSON Richtlinien im IAMBenutzerhandbuch.

Berechtigungsdelegation

AWS-Konto Besitzt ein Benutzer eine Ressource, kann er diese Berechtigungen einem anderen gewähren AWS-Konto. Dieses Konto kann diese Berechtigungen oder eine Untermenge davon an Benutzer in dem Konto delegieren. Dies wird als Delegierung von Berechtigungen bezeichnet. Ein Konto, das Berechtigungen von einem anderen Konto erhält, kann Berechtigungen jedoch nicht kontoübergreifend an ein anderes delegieren. AWS-Konto

Amazon-S3-Bucket- und Objekt-Eigentümerschaft

Buckets und Objekte sind Amazon-S3-Ressourcen. Standardmäßig kann nur der jeweilige Eigentümer auf diese Ressourcen zugreifen. Der Ressourcenbesitzer bezieht sich auf den, der AWS-Konto die Ressource erstellt hat. Beispielsweise:

- Derjenige AWS-Konto , den Sie verwenden, um Buckets zu erstellen und Objekte hochzuladen, besitzt diese Ressourcen.
- Wenn Sie ein Objekt mit AWS Identity and Access Management (IAM) Benutzer- oder Rollenanmeldedaten hochladen, gehört AWS-Konto das Objekt dem Benutzer oder der Rolle, zu der der Benutzer oder die Rolle gehört.
- Ein Bucket-Eigentümer kann einem anderen AWS-Konto (oder Benutzern in einem anderen Konto) kontenübergreifende Berechtigungen erteilen, um Objekte hochzuladen. In diesem Fall gehören diese Objekte dem AWS-Konto , das sie hochlädt. Der Bucket-Besitzer hat keine Berechtigungen für die Objekte, die anderen Konten gehören, mit den folgenden Ausnahmen:
  - Der Bucket-Eigentümer zahlt die Rechnungen. Ein Bucket-Eigentümer kann explizit den Zugriff auf Objekte verweigern oder Objekte im Bucket löschen, unabhängig davon, wem sie gehören.
  - Ein Bucket-Eigentümer kann Objekte archivieren oder archivierte Objekte wiederherstellen, unabhängig davon, wem sie gehören. Die Archivierung bezieht sich auf die Speicherklasse,

die beim Speichern der Objekte verwendet wurde. Weitere Informationen finden Sie unter Verwaltung des Lebenszyklus von Objekten.

Eigentümerschaft und Anforderungsauthentifizierung

Alle Anforderungen nach einem Bucket sind authentifiziert oder nicht authentifiziert. Authentifizierte Anforderungen müssen einen Signaturwert enthalten, der den Absender der Anforderung authentifiziert, und für nicht authentifizierte Anforderungen gilt dies nicht. Weitere Informationen zur Anforderungsauthentifizierung finden Sie unter Anfragen stellen in der Amazon S3 API S3-Referenz.

Der Eigentümer eines Buckets kann nicht authentifizierte Anforderungen zulassen. Zum Beispiel nicht authentifiziert PutObjectAnfragen sind zulässig, wenn für einen Bucket eine öffentliche Bucket-Richtlinie gilt oder wenn ein Bucket speziell der `All Users` Gruppe `WRITE` oder dem anonymen Benutzer `FULL_CONTROL` Zugriff ACL gewährt. Weitere Informationen zu öffentlichen Bucket-Richtlinien und öffentlichen Zugriffskontrolllisten (ACLs) finden Sie unterDie Bedeutung von „öffentlich“.

Alle nicht authentifizierten Anforderungen werden vom anonymen Benutzer erstellt.
Dieser Benutzer wird ACLs durch die spezifische kanonische Benutzer-ID repräsentiert.
`65a011a29cdf8ec533ec3d1ccaae921c` Wenn ein Objekt mit einer nicht authentifizierten Anforderung zu einem Bucket hochgeladen wird, ist der anonyme Benutzer Eigentümer des Objekts. Das Standardobjekt ACL gewährt `FULL_CONTROL` dem anonymen Benutzer die Rolle des Eigentümers des Objekts. Daher erlaubt Amazon S3 nicht authentifizierte Anfragen, um das Objekt abzurufen oder zu ändern. ACL

Um zu verhindern, dass Objekte vom anonymen Benutzer geändert werden, empfehlen wir, keine Bucket-Richtlinien zu implementieren, die anonyme öffentliche Schreibvorgänge in Ihren Bucket zulassen, oder solche zu verwendenACLs, die anonymen Benutzern Schreibzugriff auf Ihren Bucket gewähren. Sie können diese empfohlene Verhaltensweise durch die Verwendung von Amazon S3 Block Public Access erzwingen.

Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher. Weitere Informationen zu finden ACLs Sie unterÜbersicht über die Zugriffskontrollliste (ACL).

> **Important**
>
> Es wird empfohlen, die AWS-Konto Root-Benutzeranmeldedaten nicht für authentifizierte Anfragen zu verwenden. Erstellen Sie stattdessen eine IAM Rolle und gewähren Sie dieser

Rolle vollen Zugriff. Wir bezeichnen Benutzer mit dieser Rolle als Administratorbenutzer. Sie können die der Administratorrolle zugewiesenen Anmeldeinformationen anstelle von AWS-Konto Root-Benutzeranmeldedaten verwenden, um mit Aufgaben wie dem Erstellen eines Buckets, dem Erstellen von Benutzern AWS und dem Erteilen von Berechtigungen zu interagieren und diese auszuführen. Weitere Informationen finden Sie unter AWS Sicherheitsanmeldedaten und bewährte Sicherheitsmethoden IAM im IAM Benutzerhandbuch.

Bucket-Richtlinien für Amazon S3

Eine Bucket-Richtlinie ist eine auf Ressourcen basierende Richtlinie, die Sie verwenden können, um Zugriffsberechtigungen für Ihren Amazon-S3-Bucket und die darin enthaltenen Objekte zu erteilen. Nur der Bucket-Eigentümer kann einem Bucket eine Richtlinie zuordnen. Die dem Bucket zugeordneten Berechtigungen gelten für alle Objekte im Bucket, die dem Bucket-Eigentümer gehören. Diese Berechtigungen gelten nicht für Objekte, die anderen gehören AWS-Konten.

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie den Besitz von Objekten kontrollieren können, die in Ihren Bucket hochgeladen wurden, und um Zugriffskontrolllisten zu deaktivieren oder zu aktivieren ()ACLs. Standardmäßig ist für Object Ownership die erzwungene Einstellung Bucket Owner festgelegt, und alle sind ACLs deaktiviert. Der Bucket-Eigentümer besitzt alle Objekte im Bucket und verwaltet den Datenzugriff ausschließlich mithilfe von Richtlinien.

Bucket-Richtlinien verwenden die Richtliniensprache JSON based AWS Identity and Access Management (IAM). Sie können Bucket-Richtlinien verwenden, um Berechtigungen für die Objekte in einem Bucket hinzuzufügen oder zu verweigern. Bucket-Richtlinien können Anforderungen basierend auf den Elementen in der Richtlinie erlauben oder verweigern. Diese Elemente umfassen den Anforderer, S3-Aktionen, Ressourcen und Aspekte oder Bedingungen der Anforderung (z. B. die IP-Adresse, die für die Anforderung verwendet wird).

Sie können z. B. eine Bucket-Richtlinie erstellen, die Folgendes bewirkt:

- Erteilt anderen Konten kontoübergreifende Berechtigungen für das Hochladen von Objekten in Ihren S3-Bucket
- Stellt sicher, dass Sie als Bucket-Eigentümer die volle Kontrolle über die hochgeladenen Objekte haben

Weitere Informationen finden Sie unter Beispiele für Amazon S3 S3-Bucket-Richtlinien.

> **Important**
>
> Sie können eine Bucket-Richtlinie nicht verwenden, um Löschungen oder Übergänge durch eine S3-Lebenszyklusregel zu verhindern. Selbst wenn Ihre Bucket-Richtlinie beispielsweise alle Aktionen für alle Principals ablehnt, funktioniert Ihre S3-Lifecycle-Konfiguration weiterhin wie gewohnt.

Die Themen in diesem Abschnitt enthalten Beispiele und zeigen Ihnen, wie Sie in der S3-Konsole eine Bucket-Richtlinie hinzufügen können. Informationen zu identitätsbasierten Richtlinien finden Sie unter. Identitätsbasierte Richtlinien für Amazon S3 Weitere Informationen zur Sprache der Bucket-Richtlinie finden Sie unter Richtlinien und Berechtigungen in Amazon S3.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unter. Erforderliche Berechtigungen für Amazon S3 API S3-Operationen

Themen

- Hinzufügen einer Bucket-Richtlinie mit der Amazon-S3-Konsole
- Steuern des Zugriffs von VPC Endpunkten aus mit Bucket-Richtlinien
- Beispiele für Amazon S3 S3-Bucket-Richtlinien
- Beispiele für Bucket-Richtlinien mit Bedingungsschlüsseln

Hinzufügen einer Bucket-Richtlinie mit der Amazon-S3-Konsole

Sie können die den AWS -Richtliniengenerator und die Amazon-S3-Konsole verwenden, um eine neue Bucket-Richtlinie hinzuzufügen oder eine vorhandene zu bearbeiten. Eine Bucket-Richtlinie ist eine ressourcenbasierte AWS Identity and Access Management (IAM) Richtlinie. Sie fügen einem Bucket eine Bucket-Richtlinie hinzu, um anderen IAM Benutzern AWS-Konten oder anderen Benutzern Zugriffsberechtigungen für den Bucket und die darin enthaltenen Objekte zu gewähren. Objektberechtigungen gelten nur für die Objekte, die der Bucket-Eigentümer erstellt. Weitere Informationen zu Bucket-Richtlinien finden Sie unter Identity and Access Management für Amazon S3.

Beheben Sie Sicherheitswarnungen, Fehler, allgemeine Warnungen und Vorschläge von AWS Identity and Access Management Access Analyzer bevor Sie Ihre Richtlinie speichern. IAMAccess Analyzer führt Richtlinienprüfungen durch, um Ihre Richtlinie anhand der IAM Richtliniengrammatik

und der bewährten Methoden zu überprüfen. Diese Prüfungen generieren Ergebnisse und bieten umsetzbare Empfehlungen, die Sie beim Erstellen von Richtlinien unterstützen, die funktionsfähig sind und den bewährten Methoden für Sicherheit entsprechen. Weitere Informationen zur Überprüfung von Richtlinien mithilfe von IAM Access Analyzer finden Sie unter IAMAccess Analyzer-Richtlinienvalidierung im IAMBenutzerhandbuch. Eine Liste der Warnungen, Fehler und Vorschläge, die von Access Analyzer zurückgegeben werden, finden Sie unter IAM IAMAccess Analyzer-Referenz zur Richtlinienüberprüfung.

Hinweise zur Behebung von Fehlern mit einer Richtlinie finden Sie unter Problembehandlung bei verweigertem Zugriff beheben (403 Forbidden) Fehler in Amazon S3.

Eine Bucket-Richtlinie erstellen oder bearbeiten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie eine Bucket-Richtlinie erstellen wollen oder dessen Bucket-Richtlinie Sie bearbeiten wollen.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Wählen Sie unter Bucket-Richtlinie Bearbeiten aus. Die Seite Edit bucket policy (Bucket-Richtlinie bearbeiten) wird angezeigt.
6. Führen Sie auf der Seite Edit bucket policy (Bucket-Richtlinie bearbeiten) einen der folgenden Schritte aus:

   - Um Beispiele für Bucket-Richtlinien zu sehen, wählen Sie Richtlinienbeispiele. Oder schauen Sie Beispiele für Amazon S3 S3-Bucket-Richtlinien im Amazon S3 S3-Benutzerhandbuch nach.
   - Um eine Richtlinie automatisch zu generieren oder zu bearbeiten, wählen Sie JSON im Abschnitt Richtlinie die Option Richtliniengenerator.

   Wenn Sie den Richtliniengenerator wählen, wird der AWS Richtliniengenerator in einem neuen Fenster geöffnet.

   a. Wählen Sie auf der Seite AWS Policy Generator (Richtliniengenerator) unter Select Type of Policy (Richtlinientyp auswählen) die Option S3 Bucket Policy (S3-Bucket-Richtlinie) aus.
   b. Fügen Sie eine Anweisung hinzu, indem Sie die Informationen in die bereitgestellten Felder eingeben, und wählen Sie dann Anweisung hinzufügen. Wiederholen Sie diesen

Vorgang für so viele Anweisungen, wie Sie hinzufügen möchten. Weitere Informationen zu diesen Feldern finden Sie in der Referenz zu den IAM JSON Richtlinienelementen im IAMBenutzerhandbuch.

> **Note**
>
> Der Einfachheit halber wird auf der Seite Bucket-Richtlinie bearbeiten der Bucket ARN (Amazon-Ressourcenname) des aktuellen Buckets über dem Policy-Textfeld angezeigt. Sie können dies ARN zur Verwendung in den Kontoauszügen auf der Seite AWS Policy Generator kopieren.

c. Wenn Sie mit dem Hinzufügen von Anweisungen fertig sind, wählen Sie Generieren von Richtlinien.

d. Kopieren Sie den generierten Richtlinientext, wählen Sie Schließen und kehren Sie zur Seite Bucket-Richtlinie bearbeiten in der Amazon-S3-Konsole zurück.

7. Bearbeiten Sie im Feld Richtlinie die bestehende Richtlinie oder fügen Sie die Bucket-Richtlinie aus dem AWS Policy Generator ein. Beheben Sie Sicherheitswarnungen, Fehler, allgemeine Warnungen und Vorschläge bevor Sie Ihre Richtlinie speichern.

> **Note**
>
> Bucket-Richtlinien sind auf eine Größe von 20 KB beschränkt.

8. (Optional) Wählen Sie unten rechts Preview external access (Vorschau des externen Zugriffs) aus, um eine Vorschau anzuzeigen, wie sich Ihre neue Richtlinie auf den öffentlichen und kontoübergreifenden Zugriff auf Ihre Ressource auswirkt. Bevor Sie Ihre Richtlinie speichern, können Sie überprüfen, ob sie neue IAM Access Analyzer-Ergebnisse einführt oder bestehende Ergebnisse behebt. Wenn Sie keinen aktiven Analyzer sehen, wählen Sie Go to Access Analyzer, um einen Account Analyzer in IAM Access Analyzer zu erstellen. Weitere Informationen finden Sie unter Vorschauzugriff im IAMBenutzerhandbuch.
9. Wählen Sie Save changes (Änderungen speichern) aus, wodurch Sie zur Registerkarte Permissions (Berechtigungen) zurückkehren.

Steuern des Zugriffs von VPC Endpunkten aus mit Bucket-Richtlinien

Sie können Amazon S3 S3-Bucket-Richtlinien verwenden, um den Zugriff auf Buckets von bestimmten Virtual Private Cloud (VPC) -Endpunkten oder bestimmten zu kontrollieren. VPCs Dieser

Abschnitt enthält Beispiel-Bucket-Richtlinien, mit denen Sie den Amazon S3 S3-Bucket-Zugriff von VPC Endpunkten aus steuern können. Informationen zum Einrichten von VPC Endpunkten finden Sie unter VPCEndgeräte im VPC Benutzerhandbuch.

A VPC ermöglicht es Ihnen, AWS Ressourcen in einem von Ihnen definierten virtuellen Netzwerk bereitzustellen. Mit einem VPC Endpunkt können Sie eine private Verbindung zwischen Ihrem Endgerät VPC und einem anderen herstellen AWS-Service. Für diese private Verbindung ist kein Zugriff über das Internet, über eine virtuelle private Netzwerkverbindung (VPN), über eine NAT Instanz oder über erforderlich AWS Direct Connect.

Ein VPC Endpunkt für Amazon S3 ist eine logische Einheit innerhalb einerVPC, die nur Konnektivität zu Amazon S3 ermöglicht. Der VPC Endpunkt leitet Anfragen an Amazon S3 weiter und leitet Antworten zurück an denVPC. VPCEndpunkte ändern nur die Art und Weise, wie Anfragen weitergeleitet werden. Öffentliche Endpunkte und DNS Namen von Amazon S3 funktionieren weiterhin mit VPC Endpunkten. Wichtige Informationen zur Verwendung von VPC Endpunkten mit Amazon S3 finden Sie unter Gateway-Endpunkte und Gateway-Endpunkte für Amazon S3 im VPC Benutzerhandbuch.

VPCEndpunkte für Amazon S3 bieten zwei Möglichkeiten, den Zugriff auf Ihre Amazon S3 S3-Daten zu kontrollieren:

- Sie können die Anfragen, Benutzer oder Gruppen steuern, die über einen bestimmten VPC Endpunkt zugelassen werden. Informationen zu dieser Art der Zugriffskontrolle finden Sie im Benutzerhandbuch unter Steuern des Zugriffs auf VPC Endgeräte mithilfe von VPC Endpunktrichtlinien.
- Mithilfe der Amazon S3 S3-Bucket-Richtlinien können Sie kontrollieren, welche VPCs oder welche VPC Endpunkte Zugriff auf Ihre Buckets haben. Beispiele für diese Art Zugriffssteuerung durch Bucket-Richtlinienfinden Sie in den folgenden Themen zur Zugriffsbeschränkung.

Themen

- Beschränken Sie den Zugriff auf einen bestimmten Endpunkt VPC
- Beschränken Sie den Zugriff auf einen bestimmten VPC

> **Important**
>
> Wenn Sie die in diesem Abschnitt beschriebenen Amazon S3 S3-Bucket-Richtlinien für VPC Endgeräte anwenden, können Sie Ihren Zugriff auf den Bucket unbeabsichtigt blockieren.

Bucket-Berechtigungen, mit denen der Bucket-Zugriff speziell auf Verbindungen beschränkt werden soll, die von Ihrem VPC Endpunkt ausgehen, können alle Verbindungen zum Bucket blockieren. Informationen zur Behebung dieses Problems finden Sie unter Wie behebe ich meine Bucket-Richtlinie, wenn sie die falsche VPC oder die VPC Endpunkt-ID hat? im AWS Support Knowledge Center.

Beschränken Sie den Zugriff auf einen bestimmten Endpunkt VPC

Im Folgenden finden Sie ein Beispiel für eine Amazon S3 S3-Bucket-Richtlinie, die den Zugriff auf einen bestimmten Bucket nur vom VPC Endpunkt mit der ID `vpce-1a2b3c4d` aus einschränkt. `awsexamplebucket1` Wenn der angegebene Endpunkt nicht verwendet wird, verweigert die Richtlinie jeglichen Zugriff auf den Bucket. Die `aws:SourceVpce` Bedingung gibt den Endpunkt an. Die `aws:SourceVpce` Bedingung erfordert keinen Amazon-Ressourcennamen (ARN) für die VPC Endpunktressource, sondern nur die VPC Endpunkt-ID. Weitere Informationen über die Verwendung von Bedingungen in einer Richtlinie finden Sie unter Beispiele für Bucket-Richtlinien mit Bedingungsschlüsseln.

Important

- Bevor Sie die folgende Beispielrichtlinie verwenden, ersetzen Sie die VPC Endpunkt-ID durch einen für Ihren Anwendungsfall geeigneten Wert. Andernfalls können Sie nicht auf Ihren Bucket zugreifen.
- Diese Richtlinie deaktiviert den Konsolenzugriff auf den angegebenen Bucket, da Konsolenanfragen nicht vom angegebenen VPC Endpunkt stammen.

```
{
   "Version": "2012-10-17",
   "Id": "Policy1415115909152",
   "Statement": [
     {
       "Sid": "Access-to-specific-VPCE-only",
       "Principal": "*",
       "Action": "s3:*",
       "Effect": "Deny",
       "Resource": ["arn:aws:s3:::awsexamplebucket1",
                    "arn:aws:s3:::awsexamplebucket1/*"],
```

```
        "Condition": {
          "StringNotEquals": {
            "aws:SourceVpce": "vpce-1a2b3c4d"
          }
        }
      }
   ]
}
```

Beschränken Sie den Zugriff auf einen bestimmten VPC

Mithilfe der Bedingung können Sie eine Bucket-Richtlinie erstellen, die den Zugriff auf eine bestimmte VPC Kategorie einschränkt. `aws:SourceVpc` Dies ist nützlich, wenn Sie mehrere VPC Endgeräte auf demselben VPC Gerät konfiguriert haben und den Zugriff auf Ihre Amazon S3 S3-Buckets für all Ihre Endgeräte verwalten möchten. Im Folgenden finden Sie ein Beispiel für eine Richtlinie, die Dritten den Zugriff auf `awsexamplebucket1` und ihre Objekte verweigert. VPC `vpc-111bbb22` Wenn die angegebene Option VPC nicht verwendet wird, verweigert die Richtlinie jeglichen Zugriff auf den Bucket. Diese Anweisung gewährt keinen Zugriff auf den Bucket. Um Zugriff zu gewähren, müssen Sie eine separate `Allow` Erklärung hinzufügen. Für den `vpc-111bbb22` Bedingungsschlüssel ist kein ARN für die VPC Ressource erforderlich, sondern nur die VPC ID.

> **Important**
>
> - Bevor Sie die folgende Beispielrichtlinie verwenden, ersetzen Sie die VPC ID durch einen Wert, der Ihrem Anwendungsfall entspricht. Andernfalls können Sie nicht auf Ihren Bucket zugreifen.
> - Diese Richtlinie deaktiviert den Konsolenzugriff auf den angegebenen Bucket, da Konsolenanfragen nicht aus dem angegebenen VPC Bucket stammen.

```
{
   "Version": "2012-10-17",
   "Id": "Policy1415115909153",
   "Statement": [
     {
       "Sid": "Access-to-specific-VPC-only",
       "Principal": "*",
       "Action": "s3:*",
       "Effect": "Deny",
```

```
        "Resource": ["arn:aws:s3:::awsexamplebucket1",
                     "arn:aws:s3:::awsexamplebucket1/*"],
        "Condition": {
          "StringNotEquals": {
            "aws:SourceVpc": "vpc-111bbb22"
          }
        }
      }
   ]
}
```

Beispiele für Amazon S3 S3-Bucket-Richtlinien

Mit Amazon-S3-Bucket-Richtlinien können Sie den Zugriff auf Objekte in Ihren Buckets sichern, sodass nur Benutzer mit den entsprechenden Berechtigungen darauf zugreifen können. Sie können sogar verhindern, dass authentifizierte Benutzer ohne die entsprechenden Berechtigungen auf Ihre Amazon-S3-Ressourcen zugreifen.

Dieser Abschnitt veranschaulicht Beispiele für typische Anwendungsfälle für Bucket-Richtlinien. Diese Beispielrichtlinien verwenden *amzn-s3-demo-bucket* als Ressourcenwert. Wenn Sie diese Richtlinien testen möchten, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen (z. B. Ihren Bucket-Namen).

Um einer Gruppe von Objekten Berechtigungen zu gewähren oder zu verweigern, können Sie Platzhalterzeichen (*) in Amazon-Ressourcennamen (ARNs) und anderen Werten verwenden. Sie können beispielsweise den Zugriff auf Objektgruppen steuern, die mit einem gemeinsamen Präfix beginnen oder mit einer bestimmten Erweiterung enden, z. B. `.html`

Weitere Hinweise zur Richtliniensprache AWS Identity and Access Management (IAM) finden Sie unterRichtlinien und Berechtigungen in Amazon S3.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

> **Note**
>
> Beim Testen von Berechtigungen unter Verwendung der Amazon-S3-Konsole erteilen Sie zusätzliche Berechtigungen, die die Konsole benötigt – `s3:ListAllMyBuckets`, `s3:GetBucketLocation` und `s3:ListBucket`. Ein detailliertes Beispiel für eine Richtlinie, die Berechtigungen für Benutzer erteilt und diese Berechtigungen unter Verwendung

der Konsole testet, finden Sie unter Kontrollieren des Zugriffs auf einen Bucket mit Benutzerrichtlinien.

Zu den zusätzlichen Ressourcen für die Erstellung von Bucket-Richtlinien gehören:

- Eine Liste der IAM Richtlinienaktionen, Ressourcen und Bedingungsschlüssel, die Sie bei der Erstellung einer Bucket-Richtlinie verwenden können, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference.
- Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.
- Anleitungen zur Erstellung einer S3-Richtlinie finden Sie unter Hinzufügen einer Bucket-Richtlinie mit der Amazon-S3-Konsole.
- Informationen zur Behebung von Fehlern mit einer Richtlinie finden Sie unter Problembehandlung bei verweigertem Zugriff beheben (403 Forbidden) Fehler in Amazon S3.

Themen

- Einem öffentlichen anonymen Benutzer nur Lesezugriff gewähren
- Erfordern von Verschlüsselung
- Verwaltung von Buckets mithilfe von Containern ACLs
- Verwaltung des Objektzugriffs mit Objektmarkierung
- Verwalten des Objektzugriffs mithilfe globaler Bedingungsschlüssel
- Verwaltung des Zugriffs auf der Grundlage HTTP unserer HTTPS Anfragen
- Verwalten des Benutzerzugriffs auf bestimmte Ordner
- Verwalten des Zugriffs für Zugriffsprotokolle
- Zugriff auf einen Amazon verwalten CloudFront OAI
- Verwalten des Zugriffs für Amazon S3 Storage Lens
- Verwalten von Berechtigungen für S3 Inventory, S3 Analytics und S3-Inventory-Berichte
- Erforderlich MFA
- Benutzer daran hindern, Objekte zu löschen

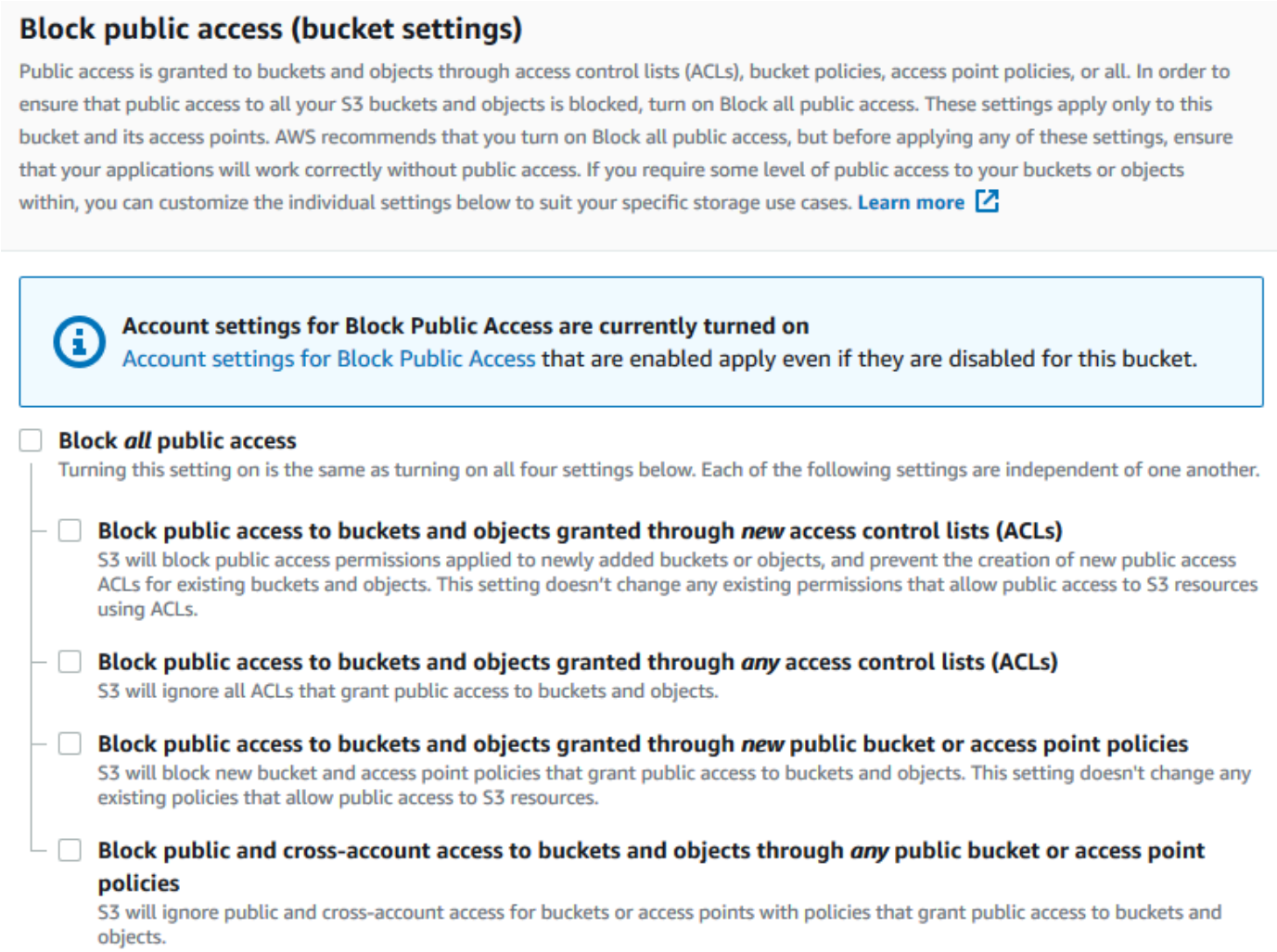
Einem öffentlichen anonymen Benutzer nur Lesezugriff gewähren

Sie können Ihre Richtlinieneinstellungen verwenden, um öffentlichen anonymen Benutzern Zugriff zu gewähren. Dies ist nützlich, wenn Sie Ihren Bucket als statische Website konfigurieren. Um öffentlichen anonymen Benutzern Zugriff zu gewähren, müssen Sie die Einstellungen „Öffentlichen Zugriff blockieren“ für Ihren Bucket deaktivieren. Weitere Informationen dazu und zu den erforderlichen Richtlinien finden Sie unterFestlegen von Berechtigungen für den Website-Zugriff. Informationen zum Einrichten restriktiverer Richtlinien für denselben Zweck finden Sie unter Wie kann ich öffentlichen Lesezugriff auf einige Objekte in meinem Amazon S3 S3-Bucket gewähren? im AWS Knowledge Center.

Standardmäßig blockiert Amazon S3 den öffentlichen Zugriff auf Ihr Konto und Ihre Buckets. Wenn Sie einen Bucket verwenden möchten, um eine statische Website zu hosten, können Sie diese Schritte verwenden, um Ihre Einstellungen für Block Public Access zu bearbeiten:

> **Warning**
>
> Bevor Sie diese Schritte ausführen, sollten Sie sicherstellen, Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher dass Sie die Risiken verstehen und akzeptieren, die mit der Gewährung eines öffentlichen Zugangs verbunden sind. Wenn Sie die Einstellungen für Block Public Access deaktivieren, um Ihren Bucket öffentlich zu machen, kann jeder im Internet auf Ihren Bucket zugreifen. Wir empfehlen Ihnen, den gesamten öffentlichen Zugriff auf Ihre Buckets zu blockieren.

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie den Namen des Buckets aus, den Sie als statische Website konfiguriert haben.
3. Wählen Sie Permissions (Berechtigungen).
4. Wählen Sie unter Block public access (bucket settings) (Öffentlichen Zugriff blockieren (Bucket-Einstellungen)), die Option Edit (Bearbeiten).
5. Löschen Sie Block all public access (Gesamten öffentlichen Zugriff blockieren) und wählen Sie Save (Speichern).



Amazon S3 deaktiviert die Einstellungen für den Block Public Access für Ihren Bucket. Um eine öffentliche statische Website zu erstellen, müssen Sie möglicherweise auch die Einstellungen zum Blockieren des öffentlichen Zugriffs für Ihr Konto bearbeiten, bevor Sie eine Bucket-Richtlinie hinzufügen. Wenn die Einstellungen „Öffentlichen Zugriff blockieren“ für Ihr Konto derzeit aktiviert sind, wird unter „Öffentlichen Zugriff blockieren“ (Bucket-Einstellungen) ein Hinweis angezeigt.

Erfordern von Verschlüsselung

Sie können eine serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-KMS) verlangen, wie in den folgenden Beispielen gezeigt.

Erforderlich SSE — KMS für alle Objekte, die in einen Bucket geschrieben werden

Die folgende Beispielrichtlinie verlangt, dass jedes Objekt, das in den Bucket geschrieben wird, mit serverseitiger Verschlüsselung unter Verwendung von AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-KMS) verschlüsselt wird. Wenn das Objekt nicht mit SSE - verschlüsselt istKMS, wird die Anfrage abgelehnt.

```
{
"Version": "2012-10-17",
"Id": "PutObjPolicy",
"Statement": [{
  "Sid": "DenyObjectsThatAreNotSSEKMS",
  "Principal": "*",
  "Effect": "Deny",
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
  "Condition": {
    "Null": {
      "s3:x-amz-server-side-encryption-aws-kms-key-id": "true"
    }
  }
}]
}
```

Erforderlich SSE — KMS mit einem spezifischen Wert AWS KMS key für alle Objekte, die in einen Bucket geschrieben werden

Die folgende Beispielrichtlinie verhindert, dass Objekte in den Bucket geschrieben werden, wenn sie nicht mit SSE - KMS unter Verwendung einer bestimmten KMS Schlüssel-ID verschlüsselt sind. Selbst wenn die Objekte mit SSE — KMS unter Verwendung einer Header- oder Bucket-Standardverschlüsselung pro Anfrage verschlüsselt sind, können die Objekte nicht in den Bucket geschrieben werden, wenn sie nicht mit dem angegebenen KMS Schlüssel verschlüsselt wurden. Achten Sie darauf, den in diesem Beispiel verwendeten KMS Schlüssel ARN durch Ihren eigenen KMS Schlüssel ARN zu ersetzen.

```
{
"Version": "2012-10-17",
"Id": "PutObjPolicy",
"Statement": [{
  "Sid": "DenyObjectsThatAreNotSSEKMSWithSpecificKey",
  "Principal": "*",
```

```
  "Effect": "Deny",
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
  "Condition": {
    "ArnNotEqualsIfExists": {
      "s3:x-amz-server-side-encryption-aws-kms-key-id": "arn:aws:kms:us-
east-2:111122223333:key/01234567-89ab-cdef-0123-456789abcdef"
    }
  }
}]
}
```

Verwaltung von Buckets mithilfe von Containern ACLs

Erteilen von Berechtigungen für mehrere Konten, um Objekte hochzuladen oder Objekte öffentlich zugänglich ACLs zu machen

Die folgende Beispielrichtlinie gewährt mehreren `s3:PutObject` Personen die `s3:PutObjectAcl` Berechtigungen und AWS-Konten. Außerdem verlangt die Beispielrichtlinie, dass alle Anfragen für diese Operationen die gespeicherte `public-read` Zugriffskontrollliste (ACL) enthalten müssen. Weitere Informationen erhalten Sie unter Politische Maßnahmen für Amazon S3 und Schlüssel für Richtlinienbedingungen für Amazon S3.

> **Warning**
>
> ACLMit der `public-read` Option „Gespeichert“ kann jeder auf der Welt die Objekte in Ihrem Bucket einsehen. Seien Sie vorsichtig, wenn Sie anonymen Zugriff auf Ihren Amazon-S3-Bucket gewähren oder die Block-Einstellungen für den öffentlichen Zugriff deaktivieren. Wenn Sie anonymen Zugriff gewähren, kann jeder auf der ganzen Welt auf Ihren Bucket zugreifen. Wir empfehlen Ihnen, niemals anonymen Zugriff auf Ihren Amazon-S3-Bucket zu gewähren, es sei denn, Sie müssen dies ausdrücklich tun, z. B. beim Hosting von statischen Websites. Informationen zum Aktivieren der Einstellungen zum Blockieren des öffentlichen Zugriffs für das Hosten statischer Websites finden Sie unter Tutorial: Konfiguration einer statischen Website in Amazon S3.

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

```
            "Sid": "AddPublicReadCannedAcl",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:root",
                    "arn:aws:iam::444455556666:root"
                ]
            },
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": [
                        "public-read"
                    ]
                }
            }
        }
    ]
}
```

Erteilung von kontoübergreifenden Berechtigungen für das Hochladen von Objekten, wobei sichergestellt wird, dass der Bucket-Eigentümer volle Kontrolle besitzt

Das folgende Beispiel zeigt, wie Sie einer anderen AWS-Konto Person das Hochladen von Objekten in Ihren Bucket ermöglichen und gleichzeitig sicherstellen können, dass Sie die volle Kontrolle über die hochgeladenen Objekte haben. Diese Richtlinie gewährt eine bestimmte AWS-Konto (*111122223333*) die Möglichkeit, Objekte nur hochzuladen, wenn das Konto beim Upload die `bucket-owner-full-control` ACL gescannten Objekte enthält. Die `StringEquals` Bedingung in der Richtlinie gibt den `s3:x-amz-acl` Bedingungsschlüssel an, mit dem die vordefinierte ACL Anforderung ausgedrückt werden soll. Weitere Informationen finden Sie unter Schlüssel für Richtlinienbedingungen für Amazon S3.

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

```
        "Principal":{"AWS":"111122223333"},
        "Action":"s3:PutObject",
        "Resource":"arn:aws:s3:::amzn-s3-demo-bucket/*",
        "Condition": {
          "StringEquals": {"s3:x-amz-acl":"bucket-owner-full-control"}
        }
      }
   ]
}
```

Verwaltung des Objektzugriffs mit Objektmarkierung

Einem Benutzer nur das Lesen von Objekten gestatten, die einen bestimmten Tag-Schlüssel und -Wert besitzen

Die folgende Berechtigungsrichtlinie beschränkt einen Benutzer darauf, nur Objekte zu lesen, die den Tag-Schlüssel und -Wert `environment: production` haben. Diese Richtlinie verwendet den `s3:ExistingObjectTag`-Bedingungsschlüssel, um den Tag-Schlüssel und -Wert anzugeben.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Principal":{
            "AWS":"arn:aws:iam::111122223333:role/JohnDoe"
         },
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket/*",
         "Condition":{
            "StringEquals":{
               "s3:ExistingObjectTag/environment":"production"
            }
         }
      }
   ]
}
```

Einschränken, welche Objekt-Tag-Schlüssel Benutzer hinzufügen können

Die folgende Beispielrichtlinie erteilt einem Benutzer die Berechtigungen, die `s3:PutObjectTagging`-Aktion auszuführen, die dem Benutzer gestattet, einem vorhandenen Objekt Markierungen hinzuzufügen. Die Bedingung verwendet den Bedingungsschlüssel `s3:RequestObjectTagKeys`, um die zulässigen Tag-Schlüssel zu spezifizieren, z. B. `Owner` oder `CreationDate`. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Erstellen einer Bedingung, die mehrere Schlüsselwerte testet.

Die Richtlinie stellt sicher, dass jeder in der Anfrage angegebene Tag-Schlüssel ein autorisierter Tag-Schlüssel ist. Der `ForAnyValue`-Qualifizierer in der Bedingung stellt sicher, dass mindestens einer der spezifizierten Schlüssel in der Anfrage enthalten ist.

```
{
   "Version": "2012-10-17",
  "Statement": [
    {"Principal":{"AWS":[
            "arn:aws:iam::111122223333:role/JohnDoe"
         ]
       },
 "Effect": "Allow",
      "Action": [
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ],
      "Condition": {"ForAnyValue:StringEquals": {"s3:RequestObjectTagKeys": [
            "Owner",
            "CreationDate"
          ]
        }
      }
    }
  ]
}
```

Einen spezifischen Tag-Schlüssel und -Wert verlangen, wenn Benutzern erlaubt wird, Objekt-Tags hinzuzufügen

Die folgende Beispielrichtlinie erteilt einem Benutzer die Berechtigungen, die `s3:PutObjectTagging`-Aktion auszuführen, die dem Benutzer gestattet, einem vorhandenen

Objekt Markierungen hinzuzufügen. Die Bedingung verlangt, dass der Benutzer einen spezifischen Tag-Schlüssel (z. B. *Project*) mit dem Wert *X* angibt.

```
{
   "Version": "2012-10-17",
  "Statement": [
    {"Principal":{"AWS":[
       "arn:aws:iam::111122223333:user/JohnDoe"
        ]
       },
      "Effect": "Allow",
      "Action": [
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ],
      "Condition": {"StringEquals": {"s3:RequestObjectTag/Project": "X"
        }
      }
    }
  ]
}
```

Einem Benutzer nur das Hinzufügen mit einem bestimmten Tag-Schlüssel und -Wert erlauben

Die folgende Beispielrichtlinie gewährt einem Benutzer die Berechtigung, die `s3:PutObject`-Aktion auszuführen, sodass er Objekte einem Bucket hinzufügen kann. Die `Condition`-Anweisung schränkt die Tag-Schlüssel und -Werte jedoch ein, die für die hochgeladenen Objekte zulässig sind. In diesem Beispiel kann der Benutzer dem Bucket nur Objekte hinzufügen, die den spezifischen Tag-Schlüssel (*Department*) mit dem Wert *Finance* haben.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Principal":{
            "AWS":[
                "arn:aws:iam::111122223333:user/JohnDoe"
         ]
        },
        "Effect": "Allow",
        "Action": [
```

```
            "s3:PutObject"
        ],
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket/*"
        ],
        "Condition": {
            "StringEquals": {
                "s3:RequestObjectTag/Department": "Finance"
            }
        }
    }]
}
```

Verwalten des Objektzugriffs mithilfe globaler Bedingungsschlüssel

Globale Bedingungsschlüssel sind Bedingungskontextschlüssel mit einem `aws` Präfix. AWS-Services kann globale Bedingungsschlüssel oder dienstspezifische Schlüssel unterstützen, die das Dienstpräfix enthalten. Sie können das `Condition` Element einer JSON Richtlinie verwenden, um die Schlüssel in einer Anfrage mit den Schlüsselwerten zu vergleichen, die Sie in Ihrer Richtlinie angeben.

Beschränken des Zugriffs nur auf Zugriffsprotokollbereitstellungen von Amazon S3 Server

Im folgenden Beispiel ist die Bucket-Richtlinie aws:SourceArnDer globale Bedingungsschlüssel wird verwendet, um den Amazon-Ressourcennamen (ARN) der Ressource zu vergleichen und eine service-to-service Anfrage mit ARN dem in der Richtlinie angegebenen zu stellen. Sie können diesen globalen `aws:SourceArn`-Bedingungsschlüssel verwenden, um zu verhindern, dass der Amazon-S3-Service bei Transaktionen zwischen Services als verwechselter Stellvertreter verwendet wird. Nur der Amazon-S3-Service darf dem Amazon-S3-Bucket Objekte hinzufügen.

Diese Bucket-Beispielrichtlinie gewährt nur dem Prinzipal des Protokollierungsservices (`logging.s3.amazonaws.com`) `s3:PutObject`-Berechtigungen.

> **Note**
>
> Die NotPrincipalElement kann nicht mit AWS-Service Prinzipalen in ressourcenbasierten Amazon S3 S3-Richtlinien, wie z. B. Bucket-Richtlinien, verwendet werden. Stattdessen empfehlen wir, den `aws:PrincipalServiceName` Bedingungsschlüssel zu verwenden, wie in der folgenden Richtlinie dargestellt.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutObjectS3ServerAccessLogsPolicy",
            "Principal": {
                "Service": "logging.s3.amazonaws.com"
            },
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket-logs/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111111111111"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::EXAMPLE-SOURCE-BUCKET"
                }
            }
        },
        {
            "Sid": "RestrictToS3ServerAccessLogs",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket-logs/*",
            "Condition": {
                "StringNotEqualsIfExists": {
                    "aws:PrincipalServiceName": "logging.s3.amazonaws.com"
                }
            }
        }
    ]
}
```

Erlauben des Zugriffs nur für Ihre Organisation

Wenn Sie möchten, dass alle IAMPrincipals, die auf eine Ressource zugreifen, von einem AWS-Konto Mitglied Ihrer Organisation stammen (einschließlich des AWS Organizations Verwaltungskontos), können Sie den `aws:PrincipalOrgID` globalen Bedingungsschlüssel verwenden.

Wenn Sie diese Art von Zugriff gewähren oder einschränken möchten, definieren Sie die `aws:PrincipalOrgID`-Bedingung und legen Sie den Wert in der Bucket-Richtlinie auf Ihre Organisations-ID fest. Die Organisations-ID wird verwendet, um den Zugriff auf den Bucket zu kontrollieren. Wenn Sie die `aws:PrincipalOrgID`-Bedingung verwenden, werden die Berechtigungen aus der Bucket-Richtlinie auch auf alle neuen Konten angewendet, die der Organisation hinzugefügt werden.

Hier ist ein Beispiel für eine ressourcenbasierte Bucket-Richtlinie, mit der Sie bestimmten IAM Prinzipalen in Ihrer Organisation direkten Zugriff auf Ihren Bucket gewähren können. Durch das Hinzufügen des globalen Bedingungsschlüssels `aws:PrincipalOrgID` zu Ihrer Bucket-Richtlinie muss sich das Prinzipal-Konto nun in Ihrer Organisation befinden, um Zugriff auf die Ressource zu erhalten. Selbst wenn Sie bei der Gewährung des Zugriffs versehentlich ein falsches Konto angeben, aws:PrincipalOrgID Der globale Bedingungsschlüssel dient als zusätzliche Absicherung. Wenn dieser globale Schüssel in einer Richtlinie verwendet wird, verhindert diese, dass alle Prinzipale außerhalb der angegebenen Organisation auf den Amazon-S3-Bucket zugreifen können. Nur Prinzipale von Konten in der aufgelisteten Organisation erhalten Zugriff auf die Ressource.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "AllowGetObject",
        "Principal": {
            "AWS": "*"
        },
        "Effect": "Allow",
        "Action": "s3:GetObject",
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
        "Condition": {
            "StringEquals": {
                "aws:PrincipalOrgID": ["o-aa111bb222"]
            }
        }
    }]
}
```

Verwaltung des Zugriffs auf der Grundlage HTTP unserer HTTPS Anfragen

Beschränken Sie den Zugriff nur auf HTTPS Anfragen

Wenn Sie verhindern möchten, dass potenzielle Angreifer den Netzwerkverkehr manipulieren, können Sie HTTPS (TLS) verwenden, um nur verschlüsselte Verbindungen zuzulassen

und gleichzeitig den Zugriff von HTTP Anfragen auf Ihren Bucket einzuschränken. Um festzustellen, ob es sich bei der Anfrage um HTTP oder handeltHTTPS, verwenden Sie aws:SecureTransportglobaler Bedingungsschlüssel in Ihrer S3-Bucket-Richtlinie. Der `aws:SecureTransport` Bedingungsschlüssel prüft, ob eine Anfrage mithilfe von gesendet wurdeHTTP.

Wenn eine Anfrage zurückkehrt`true`, wurde die Anfrage durchgesendetHTTPS. Wenn die Anfrage zurückkehrt`false`, wurde die Anfrage durchgesendetHTTP. Sie können dann den Zugriff auf Ihren Bucket basierend auf dem gewünschten Anforderungsschema erlauben oder verweigern.

Im folgenden Beispiel lehnt die Bucket-Richtlinie HTTP Anfragen ausdrücklich ab.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "RestrictToTLSRequestsOnly",
        "Action": "s3:*",
        "Effect": "Deny",
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket",
            "arn:aws:s3:::amzn-s3-demo-bucket/*"
        ],
        "Condition": {
            "Bool": {
                "aws:SecureTransport": "false"
            }
        },
        "Principal": "*"
    }]
}
```

Beschränken Sie den Zugriff auf einen bestimmten HTTP Referer

Angenommen, Sie haben eine Website mit dem Domainnamen *www.example.com* or *example.com* mit Links zu Fotos und Videos, die in Ihrem Bucket gespeichert sind*amzn-s3-demo-bucket*. Standardmäßig sind alle Amazon S3 S3-Ressourcen privat, sodass nur derjenige AWS-Konto darauf zugreifen kann, der die Ressourcen erstellt hat.

Wenn Sie Lesezugriff auf diese Objekte von Ihrer Website aus erlauben möchten, können Sie eine Bucket-Richtlinie hinzufügen, die die `s3:GetObject`-Berechtigung mit der Bedingung erlaubt, dass die `GET`-Anforderung von bestimmten Webseiten stammen muss. Die folgende Richtlinie

schränkt Anforderungen ein, indem die Bedingung `StringLike` zusammen mit dem `aws:Referer`-Bedingungsschlüssel verwendet wird.

```
{
  "Version":"2012-10-17",
  "Id":"HTTP referer policy example",
  "Statement":[
    {
      "Sid":"Allow only GET requests originating from www.example.com and
 example.com.",
      "Effect":"Allow",
      "Principal":"*",
      "Action":["s3:GetObject","s3:GetObjectVersion"],
      "Resource":"arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition":{
        "StringLike":{"aws:Referer":["http://www.example.com/*","http://example.com/
*"]}
      }
    }
  ]
}
```

Stellen Sie sicher, dass die von Ihnen verwendeten Browser den HTTP `referer` Header in die Anfrage aufnehmen.

> **Warning**
>
> Wir empfehlen, bei der Verwendung des `aws:Referer`-Bedingungsschlüssels vorsichtig zu sein. Es ist gefährlich, einen öffentlich bekannten HTTP Referer-Header-Wert einzufügen. Nicht autorisierte Parteien können mit modifizierten oder benutzerdefinierten Browsern einen beliebigen `aws:Referer`-Wert ihrer Wahl bereitstellen. Verwenden Sie es daher nicht, `aws:Referer` um zu verhindern, dass Unbefugte direkte AWS Anfragen stellen.
> Der `aws:Referer`-Bedingungsschlüssel wird nur bereitgestellt, damit Kunden ihre digitalen, in Amazon S3 gespeicherten Inhalte vor der Referenzierung auf nicht autorisierte Drittanbieter-Websites schützen können. Weitere Informationen finden Sie unter aws:Referer im IAM-Benutzerhandbuch.

Verwalten des Benutzerzugriffs auf bestimmte Ordner

Erteilen des Benutzerzugriffs auf bestimmte Ordner

Angenommen, Sie versuchen, Benutzern Zugriff auf einen bestimmten Ordner zu gewähren. Wenn der IAM Benutzer und der S3-Bucket zu demselben gehören AWS-Konto, können Sie dem Benutzer mithilfe einer IAM Richtlinie Zugriff auf einen bestimmten Bucket-Ordner gewähren. Bei diesem Ansatz müssen Sie Ihre Bucket-Richtlinie nicht aktualisieren, um Zugriff zu gewähren. Sie können die IAM Richtlinie einer IAM Rolle hinzufügen, zu der mehrere Benutzer wechseln können.

Wenn die IAM Identität und der S3-Bucket unterschiedlichen Kategorien angehören AWS-Konten, müssen Sie sowohl in der Richtlinie als auch in der IAM Bucket-Richtlinie kontoübergreifenden Zugriff gewähren. Weitere Informationen zum Gewähren von kontoübergreifendem Zugriff finden Sie unter Bucket-Eigentümer erteilt kontoübergreifende Bucket-Berechtigungen.

Die folgende Bucket-Beispielrichtlinie gewährt *JohnDoe* vollen Konsolenzugriff nur auf seinen Ordner (home/*JohnDoe*/). Indem Sie einen home-Ordner erstellen und Ihren Benutzern die entsprechenden Berechtigungen gewähren, können sich mehrere Benutzer einen einzelnen Bucket teilen. Diese Richtlinie besteht aus drei `Allow`-Anweisungen:

- *AllowRootAndHomeListingOfCompanyBucket*: Erlaubt dem Benutzer (*JohnDoe*), Objekte auf der Stammebene des *amzn-s3-demo-bucket*-Buckets und im home-Ordner aufzulisten. Diese Anweisung erlaubt dem Benutzer außerdem, mithilfe der Konsole nach dem Präfix home/ zu suchen.
- *AllowListingOfUserFolder*: Erlaubt dem Benutzer (*JohnDoe*), alle Objekte im home/*JohnDoe*/-Ordner und entsprechenden Unterordnern aufzulisten.
- *AllowAllS3ActionsInUserFolder*: Erlaubt dem Benutzer, alle Amazon-S3-Aktionen durchzuführen, indem `Read`-, `Write`- und `Delete`-Berechtigungen gewährt werden. Die Berechtigungen sind auf den Home-Ordner des Bucket-Eigentümers beschränkt.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRootAndHomeListingOfCompanyBucket",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:user/JohnDoe"
                ]
```

```
            },
            "Effect": "Allow",
            "Action": ["s3:ListBucket"],
            "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket"],
            "Condition": {
                "StringEquals": {
                    "s3:prefix": ["", "home/", "home/JohnDoe"],
                    "s3:delimiter": ["/"]
                }
            }
        },
        {
            "Sid": "AllowListingOfUserFolder",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:user/JohnDoe"
                ]
            },
            "Action": ["s3:ListBucket"],
            "Effect": "Allow",
            "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket"],
            "Condition": {
                "StringLike": {
                    "s3:prefix": ["home/JohnDoe/*"]
                }
            }
        },
        {
            "Sid": "AllowAllS3ActionsInUserFolder",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:user/JohnDoe"
                ]
            },
            "Action": ["s3:*"],
            "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket/home/JohnDoe/*"]
        }
    ]
}
```

Verwalten des Zugriffs für Zugriffsprotokolle

Gewähren des Zugriffs auf Application Load Balancer, um Zugriffsprotokolle zu aktivieren

Wenn Sie die Zugriffsprotokollierung für Ihren Application Load Balancer aktivieren, müssen Sie den Namen des S3-Buckets angeben, in dem der Load Balancer die Protokolle speichert. Dem Bucket muss eine Richtlinie angefügt sein, die Elastic Load Balancing die Berechtigung zum Schreiben in den Bucket gewährt.

Im folgenden Beispiel gewährt die Bucket-Richtlinie Elastic Load Balancing (ELB) die Berechtigung, die Zugriffs-Logs in den Bucket zu schreiben:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "AWS": "arn:aws:iam::elb-account-id:root"
      },
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/111122223333/*"
    }
  ]
}
```

> **Note**
>
> Stellen Sie sicher, dass Sie *elb-account-id* durch die AWS-Konto -ID für Elastic Load Balancing für Ihre AWS-Region ersetzen. Eine Liste der Regionen von Elastic Load Balancing finden Sie unter Hinzufügen von Richtlinien zu Ihrem S3-Bucket im Benutzerhandbuch für Elastic Load Balancing.

Wenn Ihre Region AWS-Region nicht in der Liste der unterstützten Elastic Load Balancing Balancing-Regionen aufgeführt ist, verwenden Sie die folgende Richtlinie, die dem angegebenen Log-Lieferdienst Berechtigungen gewährt.

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
       "Principal": {
         "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
          },
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/111122223333/*"
    }
  ]
}
```

Stellen Sie anschließend sicher, dass Sie Ihre Zugriffsprotokolle von Elastic Load Balancing konfigurieren, indem Sie sie aktivieren. Sie können Ihre Bucket-Berechtigungen überprüfen, indem Sie eine Testdatei erstellen.

Zugriff auf einen Amazon verwalten CloudFront OAI

Erteilen Sie einem Amazon die Erlaubnis CloudFront OAI

Die folgende Beispiel-Bucket-Richtlinie gewährt einer CloudFront Origin-Zugriffsidentität (OAI) die Berechtigung, alle Objekte in Ihrem S3-Bucket abzurufen (zu lesen). Sie können a verwenden CloudFront OAI, um Benutzern den Zugriff auf Objekte in Ihrem Bucket zu ermöglichen CloudFront , aber nicht direkt über Amazon S3. Weitere Informationen finden Sie unter Beschränken des Zugriffs auf Amazon S3 S3-Inhalte mithilfe einer Origin-Zugriffsidentität im Amazon CloudFront Developer Guide.

In der folgenden Richtlinie wird die ID OAI der Richtlinie als die ID der Richtlinie verwendet. `Principal` Weitere Informationen zur Verwendung von S3-Bucket-Richtlinien zur Gewährung des Zugriffs auf eine CloudFront OAI finden Sie unter Migration von der ursprünglichen Zugriffsidentität (OAI) zur Ursprungszugriffskontrolle (OAC) im Amazon CloudFront Developer Guide.

Zur Verwendung dieses Beispiels gehen Sie wie folgt vor:

- Ersetze es *EH1HDMB1FH2TC* durch OAI die ID. Um die OAI s-ID zu finden, besuche die Origin Access Identity-Seite auf der CloudFront Konsole oder verwende ListCloudFrontOriginAccessIdentitiesin der CloudFront API.
- Ersetzen Sie *amzn-s3-demo-bucket* durch den Namen von Ihrem Bucket.

```
{
```

```
    "Version": "2012-10-17",
    "Id": "PolicyForCloudFrontPrivateContent",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access
 Identity EH1HDMB1FH2TC"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Verwalten des Zugriffs für Amazon S3 Storage Lens

Erteilen von Berechtigungen für Amazon S3 Storage Lens

S3 Storage Lens aggregiert Ihre Metriken und zeigt die Informationen im Abschnitt Account snapshot (Konto-Snapshot) der Seite Buckets der Amazon-S3-Konsole an. S3 Storage Lens bietet außerdem ein interaktives Dashboard, mit dem Sie Erkenntnisse und Trends visualisieren, Ausreißer kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten erhalten und bewährte Datenschutzmethoden anwenden können. Das Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Organisations-, Konto-, AWS-Region-, Speicherklassen-, Bucket-, Präfix- oder Objekt- oder Storage-Lens-Gruppenebene zu generieren. Sie können auch einen täglichen Metrikexport in CSV oder senden Parquet in einen S3-Bucket formatieren.

S3 Storage Lens kann Ihre aggregierten Speichernutzungsmetriken in einen Amazon-S3-Bucket exportieren, um sie weiter zu analysieren. Der Bucket, in dem S3 Storage Lens seine Metrikexporte speichert, wird als Ziel-Bucket bezeichnet. Sie müssen eine Bucket-Richtlinie für den Ziel-Bucket haben, wenn Sie Ihren Metrikexport in S3 Storage Lens einrichten. Weitere Informationen finden Sie unter Bewerten Ihrer Speicheraktivität und -nutzung mit Amazon S3 Storage Lens.

Die folgende Bucket-Beispielrichtlinie erteilt Amazon S3 die Berechtigung, Objekte (PUT-Anforderungen) in einen Ziel-Bucket zu schreiben. Sie verwenden eine solche Bucket-Richtlinie für den Ziel-Bucket, wenn Sie einen S3-Storage-Lens-Metrik-Export einrichten.

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

```
        {
            "Sid": "S3StorageLensExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "storage-lens.s3.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-destination-bucket/destination-prefix/
StorageLens/111122223333/*"
            ],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": "111122223333",
                    "aws:SourceArn": "arn:aws:s3:region-code:111122223333:storage-
lens/storage-lens-dashboard-configuration-id"
                }
            }
        }
    ]
}
```

Verwenden Sie die folgende Änderung an der vorherigen `Resource`-Anweisung der Bucket-Richtlinie beim Einrichten eines Metrikexports von S3 Storage Lens auf Organisationsebene.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/destination-prefix/
StorageLens/your-organization-id/*",
```

Verwalten von Berechtigungen für S3 Inventory, S3 Analytics und S3-Inventory-Berichte

Gewähren von Berechtigungen für S3 Inventory und S3 Analytics

S3 Inventory erstellt Listen der Objekte in einem Bucket und der Speicherklassenanalyse-Export von S3 Analytics erstellt Ausgabedateien der in der Analyse verwendeten Daten. Der Bucket, dessen Objekte die Bestandserfassung auflistet, wird als Quell-Bucket bezeichnet. Der Bucket, in dem die Bestandsdatei und die Analyseexportdatei geschrieben werden, wird als Ziel-Bucket bezeichnet. Sie müssen eine Bucket-Richtlinie für den Ziel-Bucket erstellen, wenn Sie den Bestand oder einen Analyseexport einrichten. Weitere Informationen erhalten Sie unter Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory und Amazon S3 analytics – Speicherklassen-Analyse.

Das folgende Beispiel für eine Bucket-Richtlinie erteilt Amazon S3 die Berechtigung, Objekte aus dem Konto für den Quell-Bucket in den Ziel-Bucket zu schreiben (PUT-Anforderungen). Sie verwenden eine solche Bucket-Richtlinie für den Ziel-Bucket, wenn Sie einen S3-Inventory- und S3-Analytics-Export einrichten.

```
{
      "Version": "2012-10-17",
      "Statement": [
        {
            "Sid": "InventoryAndAnalyticsExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-DESTINATION-BUCKET/*"
            ],
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET"
                },
                "StringEquals": {
                    "aws:SourceAccount": "111122223333",
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        }
    ]
}
```

Steuern Sie die Erstellung der Konfiguration des S3-Inventarberichts

Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory erstellt Listen der Objekte in einem S3-Bucket sowie die Metadaten für jedes Objekt. Die `s3:PutInventoryConfiguration` Berechtigung ermöglicht es einem Benutzer, eine Inventarkonfiguration zu erstellen, die alle standardmäßig verfügbaren Objektmetadatenfelder enthält, und den Ziel-Bucket zum Speichern des Inventars anzugeben. Ein Benutzer mit Lesezugriff auf Objekte im Ziel-Bucket kann auf alle Objektmetadatenfelder zugreifen, die im Bestandsbericht verfügbar sind. Weitere Informationen über die Metadatenfelder, die in S3 Inventory verfügbar sind, finden Sie unterAmazon-S3-Inventory-Liste.

Um einen Benutzer daran zu hindern, einen S3-Inventarbericht zu konfigurieren, entziehen Sie dem Benutzer die `s3:PutInventoryConfiguration` Erlaubnis.

Einige Objektmetadatenfelder in Konfigurationen für S3-Inventarberichte sind optional, d. h. sie sind standardmäßig verfügbar, können aber eingeschränkt werden, wenn Sie einem Benutzer die `s3:PutInventoryConfiguration` entsprechende Berechtigung erteilen. Mithilfe des `s3:InventoryAccessibleOptionalFields` Bedingungsschlüssels können Sie steuern, ob Benutzer diese optionalen Metadatenfelder in ihre Berichte aufnehmen können. Eine Liste der optionalen Metadatenfelder, die in S3 Inventory verfügbar sind, finden Sie unter OptionalFieldsin der Amazon Simple Storage Service API Reference.

Um einem Benutzer die Erlaubnis zu erteilen, eine Inventarkonfiguration mit bestimmten optionalen Metadatenfeldern zu erstellen, verwenden Sie den `s3:InventoryAccessibleOptionalFields` Bedingungsschlüssel, um die Bedingungen in Ihrer Bucket-Richtlinie zu verfeinern.

Die folgende Beispielrichtlinie gewährt einem Benutzer (*Ana*) unter bestimmten Bedingungen die Berechtigung, eine Inventarkonfiguration zu erstellen. Die `ForAllValues:StringEquals` Bedingung in der Richtlinie verwendet den `s3:InventoryAccessibleOptionalFields` Bedingungsschlüssel, um die beiden zulässigen optionalen Metadatenfelder anzugeben, nämlich `Size` und`StorageClass`. Wenn *Ana* sie also eine Inventarkonfiguration erstellt, sind die einzigen optionalen Metadatenfelder, die sie einbeziehen kann, `Size` und`StorageClass`.

```
{
 "Id": "InventoryConfigPolicy",
 "Version": "2012-10-17",
 "Statement": [{
   "Sid": "AllowInventoryCreationConditionally",
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/Ana"
   },
   "Action":
    "s3:PutInventoryConfiguration",
   "Resource":
    "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET",
   "Condition": {
    "ForAllValues:StringEquals": {
     "s3:InventoryAccessibleOptionalFields": [
        "Size",
        "StorageClass"
        ]
```

```
      }
    }
   }
 ]
}
```

Um einen Benutzer daran zu hindern, einen S3-Inventarbericht zu konfigurieren, der bestimmte optionale Metadatenfelder enthält, fügen Sie der Bucket-Richtlinie für den Quell-Bucket eine ausdrückliche `Deny` Anweisung hinzu. Die folgende Beispiel-Bucket-Richtlinie verbietet *`Ana`* dem Benutzer, eine Inventarkonfiguration im Quell-Bucket zu erstellen, die *`DOC-EXAMPLE-SOURCE-BUCKET`* die optionalen Felder `ObjectAccessControlList` oder `ObjectOwner` Metadatenfelder enthält. Der Benutzer *`Ana`* kann weiterhin eine Inventarkonfiguration mit anderen optionalen Metadatenfeldern erstellen.

```
{
 "Id": "InventoryConfigSomeFields",
 "Version": "2012-10-17",
 "Statement": [{
   "Sid": "AllowInventoryCreation",
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/Ana"
   },
   "Action": "s3:PutInventoryConfiguration",
   "Resource":
    "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET",

  },
  {
   "Sid": "DenyCertainInventoryFieldCreation",
   "Effect": "Deny",
   "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/Ana"
   },
   "Action": "s3:PutInventoryConfiguration",
   "Resource":
     "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET",
   "Condition": {
    "ForAnyValue:StringEquals": {
     "s3:InventoryAccessibleOptionalFields": [
        "ObjectOwner",
        "ObjectAccessControlList"
```

```
        ]
      }
    }
  }
 ]
}
```

> **Note**
>
> Die Verwendung des `s3:InventoryAccessibleOptionalFields` Bedingungsschlüssels in Bucket-Richtlinien hat keinen Einfluss auf die Bereitstellung von Inventarberichten, die auf den vorhandenen Inventarkonfigurationen basieren.

> **Important**
>
> Wir empfehlen die Verwendung `ForAllValues` mit einem `Allow` Effekt oder `ForAnyValue` mit einem `Deny` Effekt, wie in den vorherigen Beispielen gezeigt.
> Verwenden Sie die Option weder `ForAllValues` mit `Deny` Wirkung noch `ForAnyValue` mit `Allow` Wirkung, da diese Kombinationen zu restriktiv sein und das Löschen der Inventarkonfiguration blockieren können.
> Weitere Informationen zu den Operatoren *`ForAllValues`* und *`ForAnyValue`* Bedingungssatzoperatoren finden Sie im Benutzerhandbuch unter Mehrwertige Kontexttasten. IAM

Erforderlich MFA

Amazon S3 unterstützt den MFA -geschützten API Zugriff, eine Funktion, die eine Multi-Faktor-Authentifizierung (MFA) für den Zugriff auf Ihre Amazon S3 S3-Ressourcen erzwingen kann. Die Multi-Faktor-Authentifizierung bietet eine zusätzliche Sicherheitsstufe, die Sie auf Ihre Umgebung anwenden können. AWS MFAist eine Sicherheitsfunktion, bei der Benutzer den physischen Besitz eines MFA Geräts nachweisen müssen, indem sie einen gültigen MFA Code angeben. Weitere Informationen finden Sie unter AWS Multi-Factor-Authentication. Sie können MFA für alle Anfragen den Zugriff auf Ihre Amazon S3 S3-Ressourcen verlangen.

Um die MFA Anforderung durchzusetzen, verwenden Sie den `aws:MultiFactorAuthAge` Bedingungsschlüssel in einer Bucket-Richtlinie. IAMBenutzer können mithilfe temporärer

Anmeldeinformationen, die von AWS Security Token Service (AWS STS) ausgestellt wurden, auf Amazon S3 S3-Ressourcen zugreifen. Sie geben den MFA Code zum Zeitpunkt der AWS STS Anfrage an.

Wenn Amazon S3 eine Anforderung mit Multi-Factor Authentication empfängt, liefert der `aws:MultiFactorAuthAge`-Bedingungsschlüssel einen numerischen Wert, der angibt, wie lange (in Sekunden) es her ist, dass die temporären Anmeldeinformationen erstellt wurden. Wenn die in der Anfrage angegebenen temporären Anmeldeinformationen nicht mithilfe eines MFA Geräts erstellt wurden, ist dieser Schlüsselwert Null (fehlt). In einer Bucket-Richtlinie können Sie eine Bedingung hinzufügen, um diesen Wert zu überprüfen, wie im folgenden Beispiel gezeigt.

Diese Beispielrichtlinie verweigert jegliche Amazon S3 S3-Operation auf dem */steuerdokumente* Ordner im *amzn-s3-demo-bucket* Bucket, wenn die Anfrage nicht mithilfe von authentifiziert wurde. MFA Weitere Informationen MFA dazu finden Sie unter Verwenden der Multi-Faktor-Authentifizierung (MFA) AWS im IAM Benutzerhandbuch.

```
{
    "Version": "2012-10-17",
    "Id": "123",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/taxdocuments/*",
        "Condition": { "Null": { "aws:MultiFactorAuthAge": true }}
      }
    ]
 }
```

Die `Null` Bedingung im `Condition` Block wird dahingehend ausgewertet, `true` ob der Wert des `aws:MultiFactorAuthAge` Bedingungsschlüssels Null ist, was darauf hinweist, dass die temporären Sicherheitsanmeldedaten in der Anforderung ohne ein MFA Gerät erstellt wurden.

Die folgende Bucket-Richtlinie ist eine Erweiterung der vorhergehenden Bucket-Richtlinie. Die folgende Richtlinie umfasst zwei Richtlinienerklärungen. Eine Anweisung gewährt jedem die `s3:GetObject`-Berechtigung für einen Bucket (*amzn-s3-demo-bucket*). Eine weitere Anweisung schränkt den Zugriff auf den *amzn-s3-demo-bucket/taxdocuments* Ordner im Bucket weiter ein, indem sie verlangtMFA.

```
{
    "Version": "2012-10-17",
    "Id": "123",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/taxdocuments/*",
        "Condition": { "Null": { "aws:MultiFactorAuthAge": true } }
      },
      {
        "Sid": "",
        "Effect": "Allow",
        "Principal": "*",
        "Action": ["s3:GetObject"],
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
      }
    ]
 }
```

Optional können Sie eine numerische Bedingung verwenden, um die Gültigkeitsdauer des `aws:MultiFactorAuthAge`-Schlüssels zu beschränken. Die Dauer, die Sie für den `aws:MultiFactorAuthAge`-Schlüssel angeben, ist unabhängig von der Lebensdauer der temporären Sicherheitsanmeldeinformationen, die für die Authentifizierung der Anforderung verwendet wurden.

Die folgende Bucket-Richtlinie verlangt beispielsweise nicht nur eine MFA Authentifizierung, sondern überprüft auch, wie lange es her ist, dass die temporäre Sitzung erstellt wurde. Die Richtlinie verweigert jede Operation, wenn der `aws:MultiFactorAuthAge`-Schlüsselwert angibt, dass die temporäre Sitzung vor mehr als einer Stunde erstellt wurde (3600 Sekunden).

```
{
    "Version": "2012-10-17",
    "Id": "123",
    "Statement": [
      {
        "Sid": "",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
```

```
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/taxdocuments/*",
        "Condition": {"Null": {"aws:MultiFactorAuthAge": true }}
      },
      {
        "Sid": "",
        "Effect": "Deny",
        "Principal": "*",
        "Action": "s3:*",
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/taxdocuments/*",
        "Condition": {"NumericGreaterThan": {"aws:MultiFactorAuthAge": 3600 }}
       },
       {
         "Sid": "",
         "Effect": "Allow",
         "Principal": "*",
         "Action": ["s3:GetObject"],
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
       }
    ]
}
```

Benutzer daran hindern, Objekte zu löschen

Standardmäßig besitzen Benutzer keine Berechtigungen. Beim Erstellen von Richtlinien können Sie Benutzern jedoch Berechtigungen gewähren, die Sie nicht gewähren wollten. Um solche Berechtigungslücken zu vermeiden, können Sie eine strengere Zugriffsrichtlinie verfassen, indem Sie eine ausdrückliche Verweigerung hinzufügen.

Um Benutzer oder Konten explizit am Löschen von Objekten zu hindern, müssen Sie einer Bucket-Richtlinie die folgenden Aktionen hinzufügen: `s3:DeleteObjects3:DeleteObjectVersion`, und Berechtigungen. `s3:PutLifecycleConfiguration` Alle drei Aktionen sind erforderlich, da Sie Objekte löschen können, indem Sie entweder die `DELETE Object` API Operationen explizit aufrufen oder ihren Lebenszyklus konfigurieren (siehe Verwaltung des Lebenszyklus von Objekten), sodass Amazon S3 die Objekte entfernen kann, wenn ihre Lebensdauer abgelaufen ist.

Im folgenden Richtlinienbeispiel verweigern Sie dem Benutzer ausdrücklich `DELETE Object` Berechtigungen *MaryMajor*. Eine ausdrückliche `Deny` Erklärung hat immer Vorrang vor allen anderen erteilten Genehmigungen.

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

```
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/MaryMajor"
      },
      "Action": [
        "s3:GetObjectVersion",
        "s3:GetBucketAcl"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1",
   "arn:aws:s3:::amzn-s3-demo-bucket1/*"
      ]
    },
    {
      "Sid": "statement2",
      "Effect": "Deny",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/MaryMajor"
      },
      "Action": [
        "s3:DeleteObject",
        "s3:DeleteObjectVersion",
        "s3:PutLifecycleConfiguration"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket1",
     "arn:aws:s3:::amzn-s3-demo-bucket1/*"
      ]
    }
  ]
}
```

Beispiele für Bucket-Richtlinien mit Bedingungsschlüsseln

Sie können die Sprache der Zugriffsrichtlinie verwenden, um Bedingungen anzugeben, wenn Sie Berechtigungen erteilen. Sie können das optionale `Condition` Element oder den `Condition` Block verwenden, um Bedingungen für das Inkrafttreten einer Richtlinie festzulegen.

Richtlinien, die Amazon-S3-Bedingungsschlüssel für Objekt- und Bucket-Vorgänge verwenden, finden Sie in den folgenden Beispielen. Weitere Informationen über Bedingungsschlüssel finden Sie unter Schlüssel für Richtlinienbedingungen für Amazon S3. Eine vollständige Liste der Amazon S3-

Aktionen, Bedingungsschlüssel und Ressourcen, die Sie in Richtlinien angeben können, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

Beispiele: Amazon S3 S3-Bedingungsschlüssel für Objektoperationen

Die folgenden Beispiele zeigen, wie Sie Amazon S3 S3-spezifische Bedingungsschlüssel für Objektoperationen verwenden können. Eine vollständige Liste der Amazon S3-Aktionen, Bedingungsschlüssel und Ressourcen, die Sie in Richtlinien angeben können, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

Einige der Beispielrichtlinien zeigen, wie Sie Bedingungsschlüssel mit PUTObjektoperationen verwenden können. PUTObjektoperationen ermöglichen Zugriffskontrolllisten (ACL) — Header, die Sie verwenden können, um bestimmte Berechtigungen zu erteilenACL. Mithilfe dieser Bedingungsschlüssel können Sie eine Bedingung festlegen, die bestimmte Zugriffsberechtigungen erfordert, wenn der Benutzer ein Objekt hochlädt. Sie können im Rahmen des ACL Vorgangs auch Zugriffsberechtigungen gewähren. PutObjectAcl Weitere Informationen finden Sie PutObjectAclin der Amazon S3 API S3-Referenz zu Amazon Simple Storage Service. Weitere Informationen zu finden ACLs Sie unterÜbersicht über die Zugriffskontrollliste (ACL).

Themen

- Beispiel 1: Erteilung einer s3:PutObject Berechtigung, die erfordert, dass Objekte mit serverseitiger Verschlüsselung gespeichert werden
- Beispiel 2: Erteilen der s3:PutObject Berechtigung zum Kopieren von Objekten mit einer Einschränkung der Kopierquelle
- Beispiel 3: Zugriff auf eine bestimmte Version eines Objekts gewähren
- Beispiel 4: Erteilen von Berechtigungen auf der Grundlage von Objekt-Tags
- Beispiel 5: Beschränkung des Zugriffs anhand der AWS-Konto ID des Bucket-Besitzers
- Beispiel 6: Eine TLS Mindestversion ist erforderlich
- Beispiel 7: Bestimmte Hauptpersonen aus einer Deny Anweisung ausschließen

Beispiel 1: Erteilung einer **`s3:PutObject`** Berechtigung, die erfordert, dass Objekte mit serverseitiger Verschlüsselung gespeichert werden

Angenommen, Konto A besitzt einen Bucket. Der Kontoadministrator möchte Jane, einer Benutzerin in Konto A, die Erlaubnis zum Hochladen von Objekten unter der Bedingung gewähren, dass Jane immer serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) anfordert. Der Konto-A-Administrator kann diese Anforderung mithilfe des `s3:x-amz-server-side-encryption` Bedingungsschlüssels wie in der Abbildung dargestellt spezifizieren. Das Schlüssel-Wert-Paar im folgenden `Condition` Block gibt den `s3:x-amz-server-side-encryption` Bedingungsschlüssel und SSE -S3 (`AES256`) als Verschlüsselungstyp an:

```
"Condition": {
     "StringNotEquals": {
         "s3:x-amz-server-side-encryption": "AES256"
     }}
```

Wenn Sie diese Berechtigung mit dem testen AWS CLI, müssen Sie die erforderliche Verschlüsselung mithilfe des `--server-side-encryption` Parameters hinzufügen, wie im folgenden Beispiel gezeigt. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key HappyFace.jpg --body c:
\HappyFace.jpg --server-side-encryption "AES256" --profile AccountAadmin
```

Beispiel 2: Erteilen der **`s3:PutObject`** Berechtigung zum Kopieren von Objekten mit einer Einschränkung der Kopierquelle

Wenn Sie in einer PUT Objektanforderung ein Quellobjekt angeben, handelt es sich bei der Anforderung um einen Kopiervorgang (siehe CopyObject). Dementsprechend kann der Bucket-Besitzer einem Benutzer die Erlaubnis erteilen, Objekte mit Einschränkungen in Bezug auf die Quelle zu kopieren, zum Beispiel:

- Erlaubt das Kopieren von Objekten nur aus dem angegebenen Quell-Bucket (z. B.*`amzn-s3-demo-source-bucket`*).
- Erlaubt das Kopieren von Objekten aus dem angegebenen Quell-Bucket und nur von Objekten, deren Schlüsselnamenpräfix mit einem bestimmten Präfix beginnt, z. B. *`öffentlich/`* (zum Beispiel*`amzn-s3-demo-source-bucket/public/*`*).

- Erlaubt das Kopieren nur eines bestimmten Objekts aus dem Quell-Bucket (z. B.*amzn-s3-demo-source-bucket/example.jpg*).

Die folgende Bucket-Richtlinie gewährt einem Benutzer (*Dave*) die `s3:PutObject` Berechtigung. Diese Richtlinie erlaubt es ihm, Objekte nur unter der Bedingung zu kopieren, dass die Anfrage den `s3:x-amz-copy-source` Header enthält und dass der Header-Wert das */amzn-s3-demo-source-bucket/public/** Schlüsselnamenpräfix angibt. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

```
{
    "Version": "2012-10-17",
    "Statement": [
       {
            "Sid": "cross-account permission to user in your own account",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Dave"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
        },
        {
            "Sid": "Deny your user permission to upload object if copy source is not /
bucket/prefix",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Dave"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-source-bucket/*",
            "Condition": {
                "StringNotLike": {
                    "s3:x-amz-copy-source": "amzn-s3-demo-source-bucket/public/*"
                }
            }
        }
    ]
}
```

Testen Sie die Richtlinie mit dem AWS CLI

Sie können die Berechtigung mit dem AWS CLI `copy-object` Befehl testen. Sie geben die Quelle an, indem Sie den Parameter `--copy-source` hinzufügen. Das Schlüsselnamenpräfix muss dem Präfix entsprechen, das in der Richtlinie zulässig ist. Sie müssen die Anmeldeinformationen des Benutzers Dave mit dem Parameter `--profile` angeben. Weitere Informationen zur Einrichtung von finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS CLI in der Amazon S3 API S3-Referenz. AWS CLI

```
aws s3api copy-object --bucket amzn-s3-demo-source-bucket --key HappyFace.jpg
--copy-source amzn-s3-demo-source-bucket/public/PublicHappyFace1.jpg --
profile AccountADave
```

Erteilen der Berechtigung, nur ein bestimmtes Objekt zu kopieren

Die vorherige Richtlinie verwendet die Bedingung `StringNotLike`. Um die Erlaubnis zu erteilen, nur ein bestimmtes Objekt zu kopieren, müssen Sie die Bedingung von `StringNotLike` bis ändern `StringNotEquals` und dann den genauen Objektschlüssel angeben, wie im folgenden Beispiel gezeigt. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
"Condition": {
       "StringNotEquals": {
           "s3:x-amz-copy-source": "amzn-s3-demo-source-bucket/public/
PublicHappyFace1.jpg"
       }
}
```

Beispiel 3: Zugriff auf eine bestimmte Version eines Objekts gewähren

Angenommen, Konto A besitzt einen Bucket mit aktivierter Versionierung. Der Bucket beinhaltet mehrere Versionen des Objekts *`HappyFace.jpg`*. Der Administrator des Kontos A möchte dem Benutzer nun die *`Dave`* Erlaubnis erteilen, nur eine bestimmte Version des Objekts abzurufen. Der Kontoadministrator kann dies erreichen, indem er dem Benutzer *`Dave`* die `s3:GetObjectVersion` Berechtigung unter bestimmten Bedingungen erteilt, wie im folgenden Beispiel gezeigt. Das Schlüssel-Wert-Paar im Block `Condition` gibt den Bedingungsschlüssel `s3:VersionId` an. In diesem Fall *`Dave`* muss die genaue Versions-ID des Objekts bekannt sein, um das Objekt aus dem angegebenen Bucket mit aktivierter Versionierung abzurufen. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre eigenen Informationen.

Weitere Informationen finden Sie unter GetObjectin der Amazon Simple Storage Service API Reference.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Dave"
            },
            "Action": "s3:GetObjectVersion",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/HappyFace.jpg"
        },
        {
            "Sid": "statement2",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Dave"
            },
            "Action": "s3:GetObjectVersion",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/HappyFace.jpg",
            "Condition": {
                "StringNotEquals": {
                    "s3:VersionId": "AaaHbAQitwiL_h47_44lRO2DDfLlBO5e"
                }
            }
        }
    ]
}
```

Testen Sie die Richtlinie mit AWS CLI

Sie können die Berechtigungen in dieser Richtlinie testen, indem Sie den AWS CLI `get-object` Befehl mit dem `--version-id` Parameter verwenden, um die spezifische Objektversion zu identifizieren, die abgerufen werden soll. Der Befehl ruft die angegebene Version des Objekts ab und speichert sie in der *`OutputFile.jpg`* Datei.

```
aws s3api get-object --bucket amzn-s3-demo-bucket --key HappyFace.jpg OutputFile.jpg --
version-id AaaHbAQitwiL_h47_44lRO2DDfLlBO5e --profile AccountADave
```

Beispiel 4: Erteilen von Berechtigungen auf der Grundlage von Objekt-Tags

Beispiele für die Verwendung von Bedingungsschlüsseln für Objekt-Tagging bei Amazon S3 S3-Vorgängen finden Sie unterMarkierungs- und Zugriffskontrollrichtlinien.

Beispiel 5: Beschränkung des Zugriffs anhand der AWS-Konto ID des Bucket-Besitzers

Sie können entweder den `s3:ResourceAccount` Bedingungsschlüssel `aws:ResourceAccount` oder verwenden, um IAM oder virtuelle private Cloud (VPC) -Endpunktrichtlinien zu schreiben, die den Benutzer-, Rollen- oder Anwendungszugriff auf die Amazon S3 S3-Buckets einschränken, die einer bestimmten AWS-Konto ID gehören. Sie können diese Bedingungsschlüssel verwenden, um zu verhindern, dass Clients in Ihrem VPC Unternehmen auf Buckets zugreifen, die Sie nicht besitzen.

Beachten Sie jedoch, dass einige AWS Dienste auf den Zugriff auf AWS verwaltete Buckets angewiesen sind. Daher kann sich die Verwendung der `s3:ResourceAccount` Taste `aws:ResourceAccount` oder in Ihrer IAM Richtlinie auch auf den Zugriff auf diese Ressourcen auswirken. Weitere Informationen finden Sie in den folgenden Ressourcen:

- Beschränken des Zugriffs auf Buckets in einem bestimmten AWS-Konto im AWS PrivateLink - Handbuch
- Beschränken Sie den Zugriff auf Buckets, die Amazon im Amazon ECR Guide ECR verwendet
- Geben Sie im Handbuch den erforderlichen Zugriff auf Systems Manager für AWS verwaltete Amazon S3 S3-Buckets an AWS Systems Manager

Weitere Informationen zu den Schlüsseln `aws:ResourceAccount` und den `s3:ResourceAccount` Bedingungsschlüsseln sowie Beispiele für deren Verwendung finden Sie unter Beschränken des Zugriffs auf Amazon S3 S3-Buckets, die bestimmten Personen gehören, AWS-Konten im AWS Storage-Blog.

Beispiel 6: Eine TLS Mindestversion ist erforderlich

Sie können den `s3:TlsVersion` Bedingungsschlüssel verwendenIAM, um Richtlinien für virtuelle private Cloud-Endpunkte (VPCE) oder Buckets zu schreiben, die den Benutzer- oder Anwendungszugriff auf Amazon S3 S3-Buckets je nach der vom Client verwendeten TLS Version einschränken. Sie können diesen Bedingungsschlüssel verwenden, um Richtlinien zu schreiben, für die eine TLS Mindestversion erforderlich ist.

Example

Die folgende Beispiel-Bucket-Richtlinie lehnt `PutObject` Anfragen von Clients ab, deren TLS Version älter als 1.2 ist, z. B. 1.1 oder 1.0. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre eigenen Informationen.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket1",
                "arn:aws:s3:::amzn-s3-demo-bucket1/*"
            ],
            "Condition": {
                "NumericLessThan": {
                    "s3:TlsVersion": 1.2
                }
            }
        }
    ]
}
```

Example

Die folgende Beispiel-Bucket-Richtlinie erlaubt `PutObject` Anfragen von Clients mit einer späteren TLS Version als 1.1, zum Beispiel 1.2, 1.3 oder höher:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": [
```

```
                "arn:aws:s3:::amzn-s3-demo-bucket1",
                "arn:aws:s3:::amzn-s3-demo-bucket1/*"
            ],
            "Condition": {
                "NumericGreaterThan": {
                    "s3:TlsVersion": 1.1
                }
            }
        }
    ]
}
```

Beispiel 7: Bestimmte Hauptpersonen aus einer **`Deny`** Anweisung ausschließen

Die folgende Bucket-Richtlinie verweigert den `s3:GetObject` Zugriff auf die*amzn-s3-demo-bucket*, außer für Principals mit der Kontonummer *123456789012*. Um diese Beispielrichtlinie zu verwenden, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyAccessFromPrincipalNotInSpecificAccount",
      "Principal": {
        "AWS": "*"
      },
      "Action": "s3:GetObject",
      "Effect": "Deny",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ],
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalAccount": [
            "123456789012"
          ]
        }
      }
    }
  ]
}
```

Beispiele: Amazon S3 S3-Bedingungsschlüssel für Bucket-Operationen

Die folgenden Beispielrichtlinien zeigen, wie Sie Amazon S3 S3-spezifische Bedingungsschlüssel für Bucket-Operationen verwenden können.

Themen

- Beispiel 1: Erteilen einer s3:GetObject Erlaubnis mit einer Bedingung für eine IP-Adresse
- Beispiel 2: Abrufen einer Liste von Objekten in einem Bucket mit einem bestimmten Präfix
- Beispiel 3: Festlegen der maximalen Anzahl von Schlüsseln

Beispiel 1: Erteilen einer **`s3:GetObject`** Erlaubnis mit einer Bedingung für eine IP-Adresse

Sie können authentifizierten Benutzern die Erlaubnis erteilen, die `s3:GetObject` Aktion zu verwenden, wenn die Anfrage von einem bestimmten IP-Adressbereich stammt (z. B.*`192.0.2.*`*), es sei denn, es handelt sich um eine IP-Adresse, die Sie ausschließen möchten (z. B.). *`192.0.2.188`* Im `Condition` Block `IpAddress` und `NotIpAddress` befinden sich Bedingungen, und jeder Bedingung wird ein Schlüssel-Wert-Paar zur Auswertung zugewiesen. Beide Schlüssel-Wert-Paare in diesem Beispiel verwenden den breiten Schlüssel. `aws:SourceIp` AWS Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre eigenen Informationen.

> **Note**
>
> Die im `Condition` Block angegebenen `NotIpAddress` Schlüsselwerte `IPAddress` und verwenden die CIDR Notation, wie in RFC 4632 beschrieben. Weitere Informationen finden Sie unter http://www.rfc-editor. org/rfc/rfc4632.txt.

```
{
    "Version": "2012-10-17",
    "Id": "S3PolicyId1",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": "*",
            "Action":"s3:GetObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition" : {
```

```
                "IpAddress" : {
                    "aws:SourceIp": "192.0.2.0/24"
                },
                "NotIpAddress" : {
                    "aws:SourceIp": "192.0.2.188/32"
                }
            }
        }
    ]
}
```

Sie können auch andere bedingte Schlüssel für AWS alle Bereiche in Amazon S3 S3-Richtlinien verwenden. Sie können beispielsweise die Bedingungsschlüssel `aws:SourceVpce` und die `aws:SourceVpc` Bedingungsschlüssel in Bucket-Richtlinien für VPC Endgeräte angeben. Spezifische Beispiele finden Sie unter Steuern des Zugriffs von VPC Endpunkten aus mit Bucket-Richtlinien.

> **Note**
>
> Für einige AWS globale Bedingungsschlüssel werden nur bestimmte Ressourcentypen unterstützt. Prüfen Sie daher, ob Amazon S3 den globalen Bedingungsschlüssel und den Ressourcentyp unterstützt, den Sie verwenden möchten, oder ob Sie stattdessen einen Amazon S3 S3-spezifischen Bedingungsschlüssel verwenden müssen. Eine vollständige Liste der unterstützten Ressourcentypen und Bedingungsschlüssel für Amazon S3 finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference.
>
> Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

Beispiel 2: Abrufen einer Liste von Objekten in einem Bucket mit einem bestimmten Präfix

Sie können den `s3:prefix` Bedingungsschlüssel verwenden, um die Antwort des ListObjectsV2APIOperation zur Eingabe von Schlüsselnamen mit einem bestimmten Präfix. Wenn Sie der Bucket-Besitzer sind, können Sie diesen Bedingungsschlüssel verwenden, um einen Benutzer darauf zu beschränken, den Inhalt eines bestimmten Präfixes im Bucket aufzulisten. Der `s3:prefix` Bedingungsschlüssel ist nützlich, wenn die Objekte im Bucket nach Schlüsselnamenpräfixen organisiert sind.

Die Amazon-S3-Konsole verwendet Schlüsselnamenpräfixe zum Anzeigen von Ordnerkonzepten. Nur die Konsole unterstützt das Konzept von Ordnern; Amazon S3 API unterstützt nur Buckets und Objekte. Zum Beispiel, wenn Sie zwei Objekte mit den Schlüsselnamen haben *public/object1.jpg* and *public/object2.jpg*, die Konsole zeigt die Objekte unter *öffentlich* folder. In Amazon S3 API sind dies Objekte mit Präfixen, keine Objekte in Ordnern. Weitere Informationen zur Verwendung von Präfixen und Trennzeichen zum Filtern von Zugriffsberechtigungen finden Sie unter Kontrollieren des Zugriffs auf einen Bucket mit Benutzerrichtlinien.

Im folgenden Szenario sind der Bucket-Besitzer und das übergeordnete Konto, zu dem der Benutzer gehört, identisch. Der Bucket-Besitzer kann also entweder eine Bucket-Richtlinie oder eine Benutzerrichtlinie verwenden, um Zugriff zu gewähren. Weitere Informationen zu anderen Bedingungsschlüsseln, die Sie für den `ListObjectsV2` API Vorgang verwenden können, finden Sie unter ListObjectsV2.

> **Note**
>
> Wenn für den Bucket die Versionsverwaltung aktiviert ist, müssen Sie, um die Objekte im Bucket aufzulisten, die `s3:ListBucketVersions` Berechtigung in den folgenden Richtlinien statt der `s3:ListBucket` Berechtigung erteilen. Die `s3:ListBucketVersions` Berechtigung unterstützt auch den `s3:prefix` Bedingungsschlüssel.

Richtlinie für Benutzer:

Die folgende Benutzerrichtlinie gewährt die `s3:ListBucket` Berechtigung (siehe ListObjectsV2) mit einer `Condition` Anweisung, nach der der Benutzer in der Anfrage ein Präfix mit dem Wert angeben muss*projects*. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Action": "s3:ListBucket",
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket",
         "Condition" : {
```

```
            "StringEquals" : {
                "s3:prefix": "projects"
            }
         }
       },
      {
         "Sid":"statement2",
         "Effect":"Deny",
         "Action": "s3:ListBucket",
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
         "Condition" : {
            "StringNotEquals" : {
                "s3:prefix": "projects"
            }
         }
       }
    ]
}
```

Die `Condition` Anweisung beschränkt den Benutzer darauf, nur Objektschlüssel aufzulisten, die das *`projects`* Präfix haben. Die hinzugefügte explizite `Deny` Anweisung verbietet dem Benutzer, Schlüssel mit einem anderen Präfix aufzulisten, unabhängig davon, welche anderen Berechtigungen der Benutzer möglicherweise hat. Es ist beispielsweise möglich, dass der Benutzer die Erlaubnis erhält, Objektschlüssel ohne Einschränkungen aufzulisten, entweder durch Aktualisierungen der vorherigen Benutzerrichtlinie oder durch eine Bucket-Richtlinie. Da explizite `Deny` Anweisungen immer `Allow` Anweisungen überschreiben, wird die Anfrage verweigert, wenn der Benutzer versucht, andere Schlüssel als die aufzulisten, die das *`projects`* Präfix haben.

Bucket-Richtlinie

Wenn Sie das `Principal` Element zur obigen Benutzerrichtlinie hinzufügen und den Benutzer identifizieren, verfügen Sie jetzt über eine Bucket-Richtlinie, wie im folgenden Beispiel gezeigt. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre eigenen Informationen.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Principal": {
```

```
            "AWS": "arn:aws:iam::123456789012:user/bucket-owner"
         },
         "Action":  "s3:ListBucket",
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
         "Condition" : {
             "StringEquals" : {
                 "s3:prefix": "projects"
             }
          }
       },
      {
         "Sid":"statement2",
         "Effect":"Deny",
         "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/bucket-owner"
         },
         "Action": "s3:ListBucket",
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
         "Condition" : {
             "StringNotEquals" : {
                 "s3:prefix": "projects"
             }
          }
       }
    ]
}
```

Testen Sie die Richtlinie mit dem AWS CLI

Sie können die Richtlinie mit dem folgenden `list-object` AWS CLI Befehl testen. Im Befehl geben Sie Benutzeranmeldeinformationen mit dem Parameter `--profile` an. Weitere Informationen zur Einrichtung und Verwendung von finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS CLI in der Amazon S3 API S3-Referenz. AWS CLI

```
aws s3api list-objects --bucket amzn-s3-demo-bucket --prefix projects --
profile AccountA
```

Beispiel 3: Festlegen der maximalen Anzahl von Schlüsseln

Sie können den `s3:max-keys` Bedingungsschlüssel verwenden, um die maximale Anzahl von Schlüsseln festzulegen, die ein Anforderer in einem ListObjectsV2 oder ListObjectVersions request. Standardmäßig geben diese API Operationen bis zu 1.000 Schlüssel zurück. Eine Liste der

numerischen Bedingungsoperatoren, die Sie zusammen mit `s3:max-keys` zugehörigen Beispielen verwenden können, finden Sie unter Numerische Bedingungsoperatoren im IAMBenutzerhandbuch.

Identitätsbasierte Richtlinien für Amazon S3

Standardmäßig sind Benutzer und Rollen nicht berechtigt, Amazon S3 S3-Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben mithilfe von AWS Management Console, AWS Command Line Interface (AWS CLI) oder ausführen AWS API. Um Benutzern die Berechtigung zu erteilen, Aktionen mit den Ressourcen durchzuführen, die sie benötigen, kann ein IAM Administrator IAM Richtlinien erstellen. Der Administrator kann dann die IAM Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen übernehmen.

Informationen zum Erstellen einer IAM identitätsbasierten Richtlinie anhand dieser JSON Beispieldokumente finden Sie unter IAM Richtlinien erstellen (Konsole) im IAMBenutzerhandbuch.

Einzelheiten zu den von Amazon S3 definierten Aktionen und Ressourcentypen, einschließlich des Formats ARNs für die einzelnen Ressourcentypen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

Themen

- Bewährte Methoden für Richtlinien
- Kontrollieren des Zugriffs auf einen Bucket mit Benutzerrichtlinien
- Beispiele für identitätsbasierte Richtlinien für Amazon S3

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand Amazon S3 S3-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder diese löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um Ihren Benutzern und Workloads zunächst Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen,

die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie AWS im IAMBenutzerhandbuch unter AWS Verwaltete Richtlinien oder Verwaltete Richtlinien für Jobfunktionen.

- Berechtigungen mit den geringsten Rechten anwenden — Wenn Sie Berechtigungen mit IAM Richtlinien festlegen, gewähren Sie nur die Berechtigungen, die für die Ausführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung IAM zum Anwenden von Berechtigungen finden Sie IAMim Benutzerhandbuch unter Richtlinien und Berechtigungen. IAM
- Verwenden Sie Bedingungen in IAM Richtlinien, um den Zugriff weiter einzuschränken — Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen einzuschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um anzugeben, dass alle Anfragen mit gesendet werden müssenSSL. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese über einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter IAMJSONRichtlinienelemente: Bedingung.
- Verwenden Sie IAM Access Analyzer, um Ihre IAM Richtlinien zu validieren, um sichere und funktionale Berechtigungen zu gewährleisten. IAM Access Analyzer validiert neue und bestehende Richtlinien, sodass die Richtlinien der IAM Richtliniensprache (JSON) und den IAM bewährten Methoden entsprechen. IAMAccess Analyzer bietet mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen, um Sie bei der Erstellung sicherer und funktionaler Richtlinien zu unterstützen. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Überprüfen von Richtlinien mit IAM Access Analyzer.
- Multi-Faktor-Authentifizierung erforderlich (MFA) — Wenn Sie ein Szenario haben, in dem IAM Benutzer oder ein Root-Benutzer erforderlich sind AWS-Konto, aktivieren Sie die Option MFA für zusätzliche Sicherheit. Um festzulegen, MFA wann API Operationen aufgerufen werden, fügen Sie MFA Bedingungen zu Ihren Richtlinien hinzu. Weitere Informationen finden Sie unter Sicherer API Zugriff mit MFA im IAMBenutzerhandbuch.

Weitere Informationen zu bewährten Methoden finden Sie unter Bewährte Sicherheitsmethoden IAM im IAM Benutzerhandbuch. IAM

Kontrollieren des Zugriffs auf einen Bucket mit Benutzerrichtlinien

Diese schrittweise Anleitung erklärt, wie Benutzerberechtigungen in Amazon S3 funktionieren. In diesem Beispiel erstellen Sie einen Bucket mit Ordnern. Anschließend erstellen Sie AWS Identity and Access Management IAM Benutzer in Ihrem AWS-Konto und gewähren diesen Benutzern inkrementelle Berechtigungen für Ihren Amazon S3 S3-Bucket und die darin enthaltenen Ordner.

Themen

- Grundlagen zu Buckets und Ordnern
- Walkthrough-Übersicht
- Vorbereitung auf den Walkthrough
- Schritt 1: Erstellen eines Buckets
- Schritt 2: IAM Benutzer und eine Gruppe erstellen
- Schritt 3: Stellen Sie sicher, dass IAM Benutzer keine Berechtigungen haben
- Schritt 4: Erteilen von Berechtigungen auf Gruppenebene
- Schritt 5: Erteilen Sie IAM der Benutzerin Alice spezifische Berechtigungen
- Schritt 6: Erteilen Sie dem IAM Benutzer Bob bestimmte Berechtigungen
- Schritt 7: Absichern des Ordners „Private“
- Schritt 8: Bereinigen
- Zugehörige Ressourcen

Grundlagen zu Buckets und Ordnern

Das Amazon-S3-Datenmodell ist eine flache Struktur: Sie erstellen einen Bucket und der Bucket speichert Objekte. Es gibt keine Hierarchie für Unter-Buckets oder Unterordner. Sie können aber eine Ordnerhierarchie nachbilden. Tools, wie die Amazon-S3-Konsole, können eine Ansicht dieser logischen Ordner und Unterordner in Ihrem Bucket präsentieren.

Die Konsole zeigt, dass sich in einem Bucket mit dem Namen `companybucket` die drei Ordner `Private`, `Development` und `Finance` und das Objekt `s3-dg.pdf` befinden. Die Konsole verwendet die Objektnamen (Schlüssel), um eine logische Hierarchie mit Ordnern und Unterordnern zu erzeugen. Betrachten Sie die folgenden Beispiele:

- Wenn Sie den Ordner `Development` anlegen, erstellt die Konsole ein Objekt mit dem Schlüssel `Development/`. Beachten Sie den als Trennzeichen dienenden abschließenden Schrägstrich (`/`).

- Wenn Sie ein Objekt mit dem Namen `Projects1.xls` in den Ordner `Development` hochladen, lädt die Konsole das Objekt hoch und weist ihm den Schlüssel `Development/Projects1.xls` zu.

  Im Schlüssel ist `Development` das Präfix und `/` ist das Trennzeichen. Amazon S3 API unterstützt Präfixe und Trennzeichen in seinem Betrieb. Beispielsweise können Sie eine Liste aller Objekte in einem Bucket mit einem bestimmten Präfix und Separator erhalten. Wenn Sie in der Konsole den Ordner `Development` öffnen, listet die Konsole die Objekte im betreffenden Ordner auf. Im folgenden Beispiel enthält der Ordner `Development` ein Objekt.

  Wenn die Konsole den Ordner `Development` im Bucket `companybucket` auflistet, sendet sie eine Anforderung an Amazon S3, in der das Präfix `Development` und das Trennzeichen `/` in der Anforderung angegeben werden. Die Antwort der Konsole sieht genauso aus, wie eine Ordnerliste im Dateisystem Ihres Computers. Das vorherige Beispiel zeigt, dass der Bucket `companybucket` ein Objekt mit dem Schlüssel `Development/Projects1.xls` enthält.

Die Konsole verwendet Objektschlüssel, um eine logische Hierarchie abzuleiten. Amazon S3 hat keine physische Hierarchie. Amazon S3 hat nur Buckets, die Objekte in einer flachen Dateistruktur enthalten. Wenn Sie Objekte mit Amazon S3 erstellenAPI, können Sie Objektschlüssel verwenden, die eine logische Hierarchie implizieren. Wenn Sie eine logische Hierarchie von Objekten erstellen, können Sie den Zugriff auf einzelne Ordner verwalten, wie in dieser exemplarischen Anleitung veranschaulicht.

Bevor Sie beginnen, müssen Sie mit dem Konzept des Bucket-Inhalts auf Stammebene vertraut sein. Angenommen, Ihr Bucket `companybucket` enthält die folgenden Objekte:

- `Private/privDoc1.txt`
- `Private/privDoc2.zip`
- `Development/project1.xls`
- `Development/project2.xls`
- `Finance/Tax2011/document1.pdf`
- `Finance/Tax2011/document2.pdf`
- `s3-dg.pdf`

Diese Objektschlüssel erzeugen eine logische Hierarchie mit `Private`, `Development` und `Finance` als Stammebenen-Ordner und `s3-dg.pdf` als Stammebenen-Objekt. Wenn Sie den Bucket-Namen

in der Amazon-S3-Konsole auswählen, erscheinen die Elemente der Stammebene. Die Konsole zeigt die Präfixe der obersten Ebene (`Private/`, `Development/` und `Finance/`) als Stammebenen-Ordner. Der Objektschlüssel `s3-dg.pdf` hat kein Präfix und erscheint daher als Stammebenen-Element.

Walkthrough-Übersicht

In dieser Anleitung erstellen Sie einen Bucket mit drei Ordnern (`Private`, `Development` und `Finance`).

Sie haben zwei User, Alice und Bob. Sie möchten, dass Alice nur auf den Ordner `Development` und Bob nur auf den Ordner `Finance` zugreift. Sie möchten, das der Inhalt des Ordners `Private` privat bleibt. In der exemplarischen Vorgehensweise verwalten Sie den Zugriff, indem Sie IAM Benutzer erstellen (im Beispiel werden die Benutzernamen Alice und Bob verwendet) und ihnen die erforderlichen Berechtigungen erteilen.

IAMunterstützt auch das Erstellen von Benutzergruppen und das Erteilen von Berechtigungen auf Gruppenebene, die für alle Benutzer in der Gruppe gelten. Dies hilft Ihnen bei der Verwaltung der Berechtigungen. In dieser Übung benötigen sowohl Alice als auch Bob einige gemeinsame Berechtigungen. Sie erstellen also eine Gruppe mit dem Namen `Consultants` und fügen dann Alice und Bob der Gruppe hinzu. Sie erteilen zunächst Berechtigungen, indem Sie der Gruppe eine Gruppenrichtlinie zuweisen. Fügen Sie dann benutzerspezifische Berechtigungen durch Zuweisen von Richtlinien zu bestimmten Benutzern hinzu.

> **Note**
>
> In der exemplarischen Vorgehensweise werden `companybucket` als Bucket-Name Alice und Bob als IAM Benutzer und `Consultants` als Gruppenname verwendet. Da Bucket-Namen in Amazon S3 global eindeutig sein müssen, müssen Sie den Bucket-Namen durch einen von Ihnen erstellten Namen ersetzen.

Vorbereitung auf den Walkthrough

In diesem Beispiel verwenden Sie Ihre AWS-Konto Anmeldeinformationen, um IAM Benutzer zu erstellen. Zu Beginn haben diese Benutzer keine Berechtigungen. Sie gewähren diesen Benutzern nach und nach Berechtigungen, damit sie spezifische Amazon-S3-Aktionen ausführen können. Um diese Berechtigungen zu testen, melden Sie sich mit den Anmeldeinformationen eines jeden

Benutzers bei der Konsole an. Da Sie als AWS-Konto Besitzer schrittweise Berechtigungen gewähren und als IAM Benutzer Testberechtigungen testen, müssen Sie sich jedes Mal mit unterschiedlichen Anmeldeinformationen an- und abmelden. Sie können diesen Test auch mit einem Browser durchführen, das Verfahren schreitet aber schneller fort, wenn Sie zwei verschiedene Browser verwenden. Verwenden Sie einen Browser, um AWS Management Console mit Ihren AWS-Konto Anmeldeinformationen eine Verbindung herzustellen, und einen anderen Browser, um eine Verbindung mit den IAM Benutzeranmeldedaten herzustellen.

Um sich AWS Management Console mit Ihren AWS-Konto Anmeldeinformationen bei der anzumelden, gehen Sie zu https://console.aws.amazon.com/. Ein IAM Benutzer kann sich nicht mit demselben Link anmelden. Ein IAM Benutzer muss eine Anmeldeseite verwenden, auf der IAM - aktiviert ist. Als Kontoinhaber können Sie diesen Link Ihren Benutzern bereitstellen.

Weitere Informationen zu finden Sie IAM unter Die AWS Management Console Anmeldeseite im IAM Benutzerhandbuch.

Um Benutzern einen Anmeldelink zur Verfügung zu stellen IAM

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich IAM Dashboard aus.
3. Beachten Sie den Link zur Anmeldung URL unter IAM Benutzern:. Sie geben IAM Benutzern diesen Link, damit sie sich mit ihrem IAM Benutzernamen und Passwort an der Konsole anmelden können.

Schritt 1: Erstellen eines Buckets

In diesem Schritt melden Sie sich mit Ihren AWS-Konto Anmeldeinformationen bei der Amazon S3-Konsole an, erstellen einen Bucket, fügen dem Bucket Ordner hinzu und laden in jeden Ordner ein oder zwei Beispieldokumente hoch.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Erstellen Sie einen Bucket.

   step-by-stepEine Anleitung finden Sie unterErstellen eines Buckets.
3. Laden Sie ein Dokument in den Bucket hoch.

Diese Übung geht davon aus, dass sich das Dokument `s3-dg.pdf` auf der Stammebene dieses Buckets befindet. Wenn Sie andere Dokumente hochladen, ändern Sie ihren Dateinamen in `s3-dg.pdf`.

4. Fügen Sie drei Ordner mit dem Namen `Private`, `Finance` and `Development` zum Bucket hinzu.

   step-by-stepAnweisungen zum Erstellen eines Ordners finden Sie unter Organisieren von Objekten in der Amazon S3-Konsole mithilfe von Ordnern > im Amazon Simple Storage Service-Benutzerhandbuch.

5. Laden Sie ein oder zwei Dokumente in jeden Ordner hoch.

   Für diese Übung wird angenommen, dass Sie einige Dokumente in jeden Ordner hochgeladen haben, sodass der Bucket Objekte mit den folgenden Schlüsseln enthält:

   - `Private/privDoc1.txt`
   - `Private/privDoc2.zip`
   - `Development/project1.xls`
   - `Development/project2.xls`
   - `Finance/Tax2011/document1.pdf`
   - `Finance/Tax2011/document2.pdf`
   - `s3-dg.pdf`

   step-by-stepAnweisungen finden Sie unterObjekte hochladen.

Schritt 2: IAM Benutzer und eine Gruppe erstellen

Verwenden Sie nun die IAMKonsole, um zwei IAM Benutzer, Alice und Bob, zu Ihrem hinzuzufügen AWS-Konto. step-by-stepAnweisungen finden Sie im IAM Benutzerhandbuch unter Einen Benutzer AWS-Konto in Ihrem erstellen. IAM

Erstellen Sie außerdem eine administrative Gruppe mit dem Namen`Consultants`. Fügen Sie dann beide Benutzer zur Gruppe hinzu. step-by-stepAnweisungen finden Sie unter IAMBenutzergruppen erstellen.

> **Warning**
>
> Wenn Sie die Benutzer und die Gruppe erstellen, fügen Sie keine Richtlinien an, die diesen Benutzern Berechtigungen erteilen. Anfänglich haben diese Benutzer keine Berechtigungen. In den folgenden Abschnitten gewähren Sie nach und nach Berechtigungen. Zunächst müssen Sie sicherstellen, dass Sie diesen IAM Benutzern Passwörter zugewiesen haben. Sie verwenden diese Benutzer-Anmeldeinformationen zum Testen der Amazon-S3-Aktionen und zum Überprüfen, ob die Berechtigungen wie erwartet funktionieren.

step-by-stepAnweisungen zum Erstellen eines neuen IAM Benutzers finden Sie im IAM Benutzerhandbuch unter Einen IAMBenutzer erstellen AWS-Konto in Ihrem. Wenn Sie die Benutzer für diese Anleitung erstellen, wählen Sie AWS Management Console -Zugriff aus und deaktivieren Sie Programmgesteuerter Zugriff.

step-by-stepAnweisungen zum Erstellen einer administrativen Gruppe finden Sie im Benutzerhandbuch unter Erstellen Ihres ersten IAM Admin-Benutzers und Ihrer ersten IAMAdministratorgruppe.

Schritt 3: Stellen Sie sicher, dass IAM Benutzer keine Berechtigungen haben

Wenn Sie zwei Browser verwenden, können Sie sich jetzt mit dem zweiten Browser mit einem der IAM Benutzeranmeldedaten an der Konsole anmelden.

1. Melden Sie sich über den IAM Benutzeranmeldelink (sieheUm Benutzern einen Anmeldelink zur Verfügung zu stellen IAM) AWS Management Console mit einem der IAM Benutzeranmeldedaten an.
2. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

   Überprüfen Sie die Konsolenmeldung, die Ihnen mitteilt, dass der Zugriff verweigert wurde.

Nun können Sie damit beginnen, den Benutzern schrittweise Berechtigungen zu erteilen. Sie weisen zunächst eine Gruppenrichtlinie zu, die beiden Benutzern die erforderlichen Berechtigungen gewährt.

Schritt 4: Erteilen von Berechtigungen auf Gruppenebene

Sie möchten den Benutzern Folgendes ermöglichen:

- Auflisten aller Buckets, die dem übergeordneten Konto gehören. Um das zu tun, müssen Bob und Alice die Berechtigung für die Aktion `s3:ListAllMyBuckets` besitzen.
- Auflisten aller Elemente, Ordner und Objekte auf Stammebene im Bucket `companybucket`. Um das zu tun, müssen Bob und Alice die Berechtigung für die Aktion `s3:ListBucket` im Bucket `companybucket` besitzen.

Zuerst erstellen Sie eine Richtlinie, die diese Berechtigungen gewährt, und dann weisen Sie sie der Gruppe `Consultants` zu.

Schritt 4.1: Erteilen der Berechtigung zum Auflisten aller Buckets

In diesem Schritt erstellen Sie eine verwaltete Richtlinie, die den Benutzern die Mindestberechtigungen für die Auflistung aller Buckets des übergeordneten Kontos erteilt. Dann weisen Sie die Richtlinie der Gruppe `Consultants` zu. Wenn Sie einem Benutzer oder einer Benutzergruppe die verwaltete Richtlinie zuordnen, erhält der Benutzer oder die Gruppe die Berechtigung, alle Buckets des übergeordneten AWS-Konto aufzulisten.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.

   > **Note**
   >
   > Da Sie Benutzerberechtigungen gewähren, melden Sie sich mit Ihren AWS-Konto Anmeldeinformationen an, nicht als IAM Benutzer.

2. Erstellen Sie die verwaltete Richtlinie.

   a. Wählen Sie links im Navigationsbereich Policies (Richtlinien) und dann Create Policy (Richtlinie erstellen) aus.

   b. Wählen Sie die JSONRegisterkarte.

   c. Kopieren Sie die folgende Zugriffsrichtlinie und fügen Sie sie in das Textfeld für die Richtlinie ein.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowGroupToSeeBucketListInTheConsole",
      "Action": ["s3:ListAllMyBuckets"],
```

```
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::*"]
    }
  ]
}
```

Eine Richtlinie ist ein JSON Dokument. Im Dokument ist ein `Statement` ein Array von Objekten, die jeweils eine Berechtigung unter Verwendung einer Sammlung von Namenswertpaaren beschreiben. Die vorangegangene Richtlinie beschreibt eine bestimmte Berechtigung. Die `Action` definiert den Zugriffstyp. In der Richtlinie ist `s3:ListAllMyBuckets` eine vordefinierte Amazon-S3-Aktion. Diese Aktion deckt den Amazon S3 GET Service-Vorgang ab, der eine Liste aller Buckets zurückgibt, die dem authentifizierten Absender gehören. Der Wert des `Effect`-Elements bestimmt, ob die spezifische Berechtigung gewährt oder verweigert wird.

d. Wählen Sie Review policy (Richtlinie überprüfen) aus. Geben Sie auf der nächsten Seite in das Feld Name `AllowGroupToSeeBucketListInTheConsole` ein und wählen Sie dann Create policy (Richtlinie erstellen).

> **Note**
>
> Der Eintrag Summary (Übersicht) enthält eine Nachricht, die angibt, dass die Richtlinie keinerlei Berechtigungen gewährt. Für diese Anleitung können Sie diese Nachricht getrost ignorieren.

3. Weisen Sie die von `AllowGroupToSeeBucketListInTheConsole` verwaltete Richtlinie, die Sie erstellt haben, der Gruppe `Consultants` zu.

   step-by-stepAnweisungen zum Anhängen einer verwalteten Richtlinie finden Sie unter Hinzufügen und Entfernen von IAM Identitätsberechtigungen im IAM Benutzerhandbuch.

   In der IAM Konsole hängen Sie Richtliniendokumente an IAM Benutzer und Gruppen an. Da Sie möchten, dass beide Benutzer die Buckets auflisten können, weisen Sie die Richtlinie der Gruppe zu.

4. Die Berechtigung testen.

   a. Melden Sie sich über den IAM Benutzeranmeldelink (siehe Um Benutzern einen Anmeldelink zur Verfügung zu stellen IAM) mit beliebigen IAM Benutzeranmeldedaten an der Konsole an.

   b. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

Die Konsole sollte nun alle Buckets auflisten, nicht aber die Objekte in den jeweiligen Buckets.

Schritt 4.2: Benutzern gestatten, dass sie den Bucket-Inhalt auf Stammebene auflisten

Als Nächstes gestatten Sie allen Benutzern in der Gruppe `Consultants`, Elemente im Bucket `companybucket` aufzulisten. Wenn ein Benutzer den Unternehmensbucket in der Amazon-S3-Konsole auswählt, werden die Elemente in der Stammebene des Buckets angezeigt.

> **Note**
>
> Dieses Beispiel verwendet `companybucket` zur Veranschaulichung. Sie müssen den Namen des Bucket verwenden, den Sie erstellt haben.

Um die Anfrage zu verstehen, die die Konsole an Amazon S3 sendet, wenn Sie einen Bucket-Namen wählen, die Antwort, die Amazon S3 zurückgibt, und wie die Konsole die Antwort interpretiert, sollten Sie den Ablauf etwas genauer untersuchen.

Wenn Sie einen Bucket-Namen wählen, sendet die Konsole die GETBucket-Anfrage (List Objects) an Amazon S3. Diese Anforderung enthält die folgenden Parameter:

- Der Parameter `prefix` mit einer leeren Zeichenfolge als Wert.
- Der Parameter `delimiter` mit `/` als Wert.

Es folgt ein Beispiel einer Anforderung.

```
GET ?prefix=&delimiter=/ HTTP/1.1
Host: companybucket.s3.amazonaws.com
Date: Wed, 01 Aug  2012 12:00:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:xQE0diMbLRepdf3YB+FIEXAMPLE=
```

Amazon S3 gibt eine Antwort zurück, die das folgende `<ListBucketResult/>`-Element enthält:

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>companybucket</Name>
  <Prefix></Prefix>
```

```
  <Delimiter>/</Delimiter>
   ...
  <Contents>
    <Key>s3-dg.pdf</Key>
    ...
  </Contents>
  <CommonPrefixes>
    <Prefix>Development/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>Finance/</Prefix>
  </CommonPrefixes>
  <CommonPrefixes>
    <Prefix>Private/</Prefix>
  </CommonPrefixes>
</ListBucketResult>
```

Das Schlüsselobjekt `s3-dg.pdf` enthält nicht das Schrägstrich-Trennzeichen (`/`), und Amazon S3 gibt den Schlüssel im `<Contents>`-Element zurück. Alle anderen Schlüssel in unserem Beispiel-Bucket enthalten jedoch das `/`-Trennzeichen. Amazon S3 gruppiert diese Schlüssel und gibt ein einziges `<CommonPrefixes>`-Element für jeden der verschiedenen Präfix-Werte `Development/`, `Finance/` und `Private/` zurück, eine Unterzeichenfolge vom Anfang dieser Schlüssel bis zum ersten Auftreten des angegebenen `/`-Trennzeichens.

Die Konsole interpretiert dieses Ergebnis und zeigt die Elemente auf der Stammebene als drei Ordner und einen Objektschlüssel an.

Wenn Bob oder Alice den Ordner Development öffnen, sendet die Konsole die GETBucket-Anfrage (List Objects) an Amazon S3, wobei die `prefix` `delimiter` Parameter auf die folgenden Werte gesetzt sind:

- Der Parameter `prefix` mit dem Wert `Development/`.
- Der Parameter `delimiter` mit dem Wert "`/`".

In der Antwort gibt Amazon S3 die Objektschlüssel mit dem angegebenen Präfix zurück.

```
<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Name>companybucket</Name>
  <Prefix>Development</Prefix>
  <Delimiter>/</Delimiter>
   ...
```

```
  <Contents>
    <Key>Project1.xls</Key>
    ...
  </Contents>
  <Contents>
    <Key>Project2.xls</Key>
    ...
  </Contents>
</ListBucketResult>
```

Die Konsole zeigt die Objektschlüssel an.

Kehren Sie nun wieder zu dem Vorgang zurück, Benutzern die Berechtigung zum Auflisten von Elementen auf der Stammebene zu erteilen. Damit der Bucket-Inhalt aufgelistet werden kann, müssen die Benutzer die Berechtigung zum Aufruf der `s3:ListBucket`-Aktion besitzen, wie in der folgenden Richtlinienanweisung dargestellt. Um sicherzustellen, dass sie nur den Inhalt auf der Stammebene sehen, können Sie als Bedingung hinzufügen, dass Benutzer in der Anforderung ein leeres `prefix` angeben müssen – d. h., dass es ihnen nicht gestattet sein soll, auf einen der Stammebenen-Ordner doppelzuklicken. Schließlich fügen Sie noch eine Bedingung hinzu, die den Zugriff im Ordnerstil dadurch vorschreibt, dass Benutzeranforderungen den Parameter `delimiter` mit dem Wert "/" enthalten müssen.

```
{
  "Sid": "AllowRootLevelListingOfCompanyBucket",
  "Action": ["s3:ListBucket"],
  "Effect": "Allow",
  "Resource": ["arn:aws:s3:::companybucket"],
  "Condition":{
         "StringEquals":{
             "s3:prefix":[""], "s3:delimiter":["/"]
                       }
              }
}
```

Wenn Sie auf der Amazon S3 S3-Konsole einen Bucket auswählen, sendet die Konsole zunächst die GETBucket-Standortanfrage, um herauszufinden, AWS-Region wo der Bucket bereitgestellt ist. Anschließend verwendet die Konsole den regionsspezifischen Endpunkt für den Bucket, um die Bucket-Anfrage (List-Objekte) zu senden. GET Wenn Benutzer die Konsole verwenden sollen, müssen Sie ihnen folglich die Berechtigung für die Aktion `s3:GetBucketLocation` gewähren, wie in der folgenden Richtlinienanweisung veranschaulicht.

```
{
   "Sid": "RequiredByS3Console",
   "Action": ["s3:GetBucketLocation"],
   "Effect": "Allow",
   "Resource": ["arn:aws:s3:::*"]
}
```

Den Benutzern gestatten, den Bucket-Inhalt auf Stammebene aufzulisten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter. https://console.aws.amazon.com/iam/

   Verwenden Sie Ihre AWS-Konto Anmeldeinformationen, nicht die eines IAM Benutzers, um sich an der Konsole anzumelden.

2. Ersetzen Sie die vorhandene `AllowGroupToSeeBucketListInTheConsole`-verwaltete Richtlinie, die der Gruppe `s3:ListBucket` zugeordnet ist, durch die folgende Richtlinie, durch die die Aktion `Consultants` ebenfalls gestattet wird. Denken Sie daran, zu ersetzen *`Firma Bucket`* in der Richtlinie `Resource` mit dem Namen Ihres Buckets.

   step-by-stepAnweisungen finden Sie im IAMBenutzerhandbuch unter IAMRichtlinien bearbeiten. Wenn Sie den step-by-step Anweisungen folgen, achten Sie darauf, dass Sie die Schritte befolgen, um Ihre Änderungen auf alle wichtigen Entitäten anzuwenden, denen die Richtlinie zugeordnet ist.

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
        {
          "Sid":
    "AllowGroupToSeeBucketListAndAlsoAllowGetBucketLocationRequiredForListBucket",
          "Action": [ "s3:ListAllMyBuckets", "s3:GetBucketLocation" ],
          "Effect": "Allow",
          "Resource": [ "arn:aws:s3:::*"  ]
        },
        {
          "Sid": "AllowRootLevelListingOfCompanyBucket",
          "Action": ["s3:ListBucket"],
          "Effect": "Allow",
          "Resource": ["arn:aws:s3:::companybucket"],
          "Condition":{
   ```

```
            "StringEquals":{
                  "s3:prefix":[""], "s3:delimiter":["/"]
                        }
                  }
     }
  ]
}
```

3. Testen Sie die aktualisierten Berechtigungen.

   a. Melden Sie sich über den IAM Benutzeranmeldelink (siehe Um Benutzern einen Anmeldelink zur Verfügung zu stellen IAM) bei der AWS Management Console an.

      Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

   b. Wählen Sie den von Ihnen erstellten Bucket aus. Die Konsole zeigt die Bucket-Elemente auf Stammebene an. Wenn Sie Ordner im Bucket auswählen, können Sie den Ordnerinhalt nicht sehen, da Sie diese Berechtigungen noch nicht gewährt haben.

Dieser Test ist erfolgreich, wenn Benutzer die Amazon-S3-Konsole verwenden. Wenn Sie einen Bucket in der Konsole auswählen, sendet die Konsolenimplementierung eine Anforderung mit dem Parameter `prefix` mit einer leeren Zeichenfolge als Wert und dem Parameter `delimiter` mit "/" als Wert.

Schritt 4.3: Übersicht über die Gruppenrichtlinie

Der Nettoeffekt der von Ihnen hinzugefügten Gruppenrichtlinie besteht darin, den IAM Benutzern Alice und Bob die folgenden Mindestberechtigungen zu gewähren:

- Auflisten aller Buckets, die dem übergeordneten Konto gehören.
- Ansicht der Elemente auf Stammebene im Bucket `companybucket`.

Die Benutzer können jedoch noch nicht sehr viele Aktionen ausführen. Als Nächstes erteilen Sie die folgenden benutzerspezifischen Berechtigungen:

- Erlauben Sie Bob, Objekte im Ordner `Development` aufzurufen und abzulegen.
- Erlauben Sie Bob, Objekte im Ordner `Finance` aufzurufen und abzulegen.

Für benutzerspezifische Berechtigungen fügen Sie eine Richtlinie zum spezifischen Benutzer hinzu, nicht für die Gruppe. Im folgenden Abschnitt erteilen Sie Alice die Berechtigung, mit dem Ordner

`Development` zu arbeiten. Sie können die Schritte wiederholen, um Bob eine ähnliche Berechtigung für das Arbeiten im Ordner `Finance` zu erteilen.

Schritt 5: Erteilen Sie IAM der Benutzerin Alice spezifische Berechtigungen

Nun gewähren Sie Alice zusätzliche Berechtigungen, damit sie den Inhalt des Ordners `Development` aufrufen und Objekte in diesem Ordner ablegen kann.

Schritt 5.1: Erteilen Sie der IAM Benutzerin Alice die Erlaubnis, den Inhalt des Entwicklungsordners aufzulisten

Damit Alice den `Development` Ordnerinhalt auflisten kann, müssen Sie dem Benutzer Alice eine Richtlinie zuweisen, die die Berechtigung für die `s3:ListBucket` Aktion im `companybucket` Bucket erteilt, vorausgesetzt, die Anfrage enthält das Präfix`Development/`. Da diese Richtlinie nur auf die Benutzerin Alice angewendet werden soll, verwenden Sie eine Inline-Richtlinie. Weitere Informationen zu Inline-Richtlinien finden Sie unter Verwaltete Richtlinien und Inline-Richtlinien im IAMBenutzerhandbuch.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.

   Verwenden Sie Ihre AWS-Konto Anmeldeinformationen, nicht die eines IAM Benutzers, um sich an der Konsole anzumelden.
2. Erstellen Sie eine Inline-Richtlinie, um der Benutzerin Alice die Berechtigung zu erteilen, den Inhalt des Ordners `Development` aufzulisten.

   a. Wählen Sie im Navigationsbereich auf der linken Seite Users (Benutzer) aus.

   b. Wählen Sie den Benutzernamen Alice.

   c. Wählen Sie auf der Benutzerdetailseite den Tab Permissions (Berechtigungen) und dann Add inline policy (Inline-Richtlinie hinzufügen).

   d. Wählen Sie den JSONTab.

   e. Kopieren Sie die folgende Richtlinie und fügen Sie sie in das Richtlinientextfeld ein.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
      "Sid": "AllowListBucketIfSpecificPrefixIsIncludedInRequest",
      "Action": ["s3:ListBucket"],
```

```
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::companybucket"],
      "Condition":{  "StringLike":{"s3:prefix":["Development/*"] }
       }
    }
  ]
}
```

f. Wählen Sie Review policy (Richtlinie überprüfen) aus. Geben Sie auf der nächsten Seite in das Feld Name einen Namen ein und wählen Sie dann Create policy (Richtlinie erstellen).

3. Testen Sie die geänderten Berechtigungen für Alice:

a. Melden Sie sich über den IAM Benutzeranmeldelink (siehe Um Benutzern einen Anmeldelink zur Verfügung zu stellen IAM) bei der AWS Management Console an.

b. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

c. Prüfen Sie in der Amazon-S3-Konsole, ob Alice die Liste der Objekte im Ordner `Development/` im Bucket sehen kann.

Wenn die Benutzerin den Ordner `/Development` auswählt, um die Liste der darin enthaltenen Objekte anzuzeigen, sendet die Amazon-S3-Konsole die Anforderung `ListObjects` zusammen mit dem Präfix `/Development` an Amazon S3. Weil die Benutzerin die Erlaubnis besitzt, die Objektliste mit dem Präfix `Development` und dem Trennzeichen `/` zu sehen, gibt Amazon S3 die Objektliste mit dem Schlüsselpräfix `Development/` zurück, und die Konsole gibt die Liste aus.

Schritt 5.2: Erteilen Sie der IAM Benutzerin Alice die Erlaubnis, Objekte abzurufen und im Entwicklungsordner abzulegen

Damit Alice Objekte im Ordner `Development` ablegen und aufrufen kann, benötigt sie die Berechtigung für die Aktionen `s3:GetObject` und `s3:PutObject`. Die folgenden Richtlinienanweisungen räumen diese Berechtigungen ein, vorausgesetzt die Anforderung enthält den Parameter `prefix` mit dem Wert `Development/`.

```
{
    "Sid":"AllowUserToReadWriteObjectData",
    "Action":["s3:GetObject", "s3:PutObject"],
    "Effect":"Allow",
    "Resource":["arn:aws:s3:::companybucket/Development/*"]
 }
```

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

   Verwenden Sie Ihre AWS-Konto Anmeldeinformationen, nicht die eines IAM Benutzers, um sich an der Konsole anzumelden.

2. Bearbeiten Sie die Inline-Richtlinie, die Sie im vorherigen Schritt erstellt haben.

   a. Wählen Sie im Navigationsbereich auf der linken Seite Users (Benutzer) aus.

   b. Klicken Sie auf den Benutzernamen Alice.

   c. Wählen Sie auf der Benutzerdetailseite den Tab Permissions (Berechtigungen) aus und erweitern Sie den Bereich Inline Policies (Inline-Richtlinien).

   d. Wählen Sie neben dem Namen der im vorherigen Schritt erstellten Richtlinie Edit Policy (Richtlinie bearbeiten) aus.

   e. Kopieren Sie die folgende Richtlinie und fügen Sie sie in das Textfeld für die Richtlinie ein, wobei die vorhandene Richtlinie ersetzt wird.

```
{
     "Version": "2012-10-17",
     "Statement":[
      {
         "Sid":"AllowListBucketIfSpecificPrefixIsIncludedInRequest",
         "Action":["s3:ListBucket"],
         "Effect":"Allow",
         "Resource":["arn:aws:s3:::companybucket"],
         "Condition":{
            "StringLike":{"s3:prefix":["Development/*"]
            }
         }
      },
      {
        "Sid":"AllowUserToReadWriteObjectDataInDevelopmentFolder",
        "Action":["s3:GetObject", "s3:PutObject"],
        "Effect":"Allow",
        "Resource":["arn:aws:s3:::companybucket/Development/*"]
      }
   ]
}
```

3. Testen Sie die aktualisierte Richtlinie:

a. Melden Sie sich mithilfe des IAM Benutzeranmeldelinks (sieheUm Benutzern einen Anmeldelink zur Verfügung zu stellen IAM) bei der AWS Management Console an.

b. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

c. Prüfen Sie in der Amazon-S3-Konsole, ob Alice nun im Ordner `Development` ein Objekt hinzufügen oder herunterladen kann.

Schritt 5.3: Verweigern Sie der IAM Benutzerin Alice ausdrücklich die Berechtigungen für alle anderen Ordner im Bucket

Die Benutzerin kann jetzt den Inhalt auf Stammebene im Bucket `companybucket` auflisten. Sie kann auch Objekte im Ordner `Development` aufrufen und ablegen. Wenn Sie die Zugriffsberechtigungen weiter verbessern möchten, können Sie Alice den Zugriff auf andere Ordner im Bucket explizit verweigern. Wenn es eine andere Richtlinie (Bucket-Richtlinie oderACL) gibt, die Alice Zugriff auf andere Ordner im Bucket gewährt, hat diese ausdrückliche Verweigerung Vorrang vor diesen Berechtigungen.

Sie können die folgende Anweisung zur Benutzerrichtlinie für Alice hinzufügen, die von allen Anfragen von Alice an Amazon S3 erfordert, dass der Parameter `prefix` enthalten ist, dessen Wert entweder `Development/*` oder eine leere Zeichenfolge ist.

```
{
   "Sid": "ExplicitlyDenyAnyRequestsForAllOtherFoldersExceptDevelopment",
   "Action": ["s3:ListBucket"],
   "Effect": "Deny",
   "Resource": ["arn:aws:s3:::companybucket"],
   "Condition":{  "StringNotLike": {"s3:prefix":["Development/*",""] },
                  "Null"         : {"s3:prefix":false }
    }
}
```

Beachten Sie, dass im Block `Condition` zwei bedingte Ausdrücke enthalten sind. Das Ergebnis dieser bedingten Ausdrücke wird durch das logische `AND` verknüpft. Wenn beide Bedingungen wahr sind, ist das Ergebnis der bedingten Ausdrücke wahr. Da der Wert für `Effect` in dieser Richtlinie `Deny` lautet, wenn `Condition` als "true" bewertet wird, können Benutzer die angegebene `Action` nicht durchführen.

- Der bedingte Ausdruck `Null` stellt sicher, dass die Anforderung von Alice den Parameter `prefix` enthält.

  Der Parameter `prefix` erfordert einen ordnerartigen Zugriff. Wenn Sie eine Anfrage ohne den Parameter `prefix` senden, gibt Amazon S3 alle Objektschlüssel zurück.

  Wenn die Anforderung den Parameter `prefix` mit einem Nullwert enthält, wird der Ausdruck als wahr ausgewertet, womit die gesamte Bedingung `Condition` wahr ist. Sie müssen eine leere Zeichenfolge für den Parameter `prefix` gestatten. Erinnern Sie sich an die vorangegangene Diskussion. Die leere Zeichenfolge lässt zu, dass Alice Bucket-Elemente auf Stammebene abrufen kann, wie es die Konsole in der vorhergehenden Diskussion macht. Weitere Informationen finden Sie unter Schritt 4.2: Benutzern gestatten, dass sie den Bucket-Inhalt auf Stammebene auflisten.
- Der bedingte Ausdruck `StringNotLike` stellt sicher, dass die Anforderung scheitert, wenn der angegebene Wert des Parameters `prefix` nicht `Development/*` ist.

Folgen Sie den Schritten im vorherigen Abschnitt und aktualisieren Sie die Inline-Richtlinie noch einmal, die Sie für die Benutzerin Alice erstellt haben.

Kopieren Sie die folgende Richtlinie und fügen Sie sie in das Textfeld für die Richtlinie ein, wobei die vorhandene Richtlinie ersetzt wird.

```
{
   "Version": "2012-10-17",
   "Statement":[
      {
         "Sid":"AllowListBucketIfSpecificPrefixIsIncludedInRequest",
         "Action":["s3:ListBucket"],
         "Effect":"Allow",
         "Resource":["arn:aws:s3:::companybucket"],
         "Condition":{
            "StringLike":{"s3:prefix":["Development/*"]
            }
         }
      },
      {
        "Sid":"AllowUserToReadWriteObjectDataInDevelopmentFolder",
        "Action":["s3:GetObject", "s3:PutObject"],
        "Effect":"Allow",
        "Resource":["arn:aws:s3:::companybucket/Development/*"]
      },
```

```
      {
         "Sid": "ExplicitlyDenyAnyRequestsForAllOtherFoldersExceptDevelopment",
         "Action": ["s3:ListBucket"],
         "Effect": "Deny",
         "Resource": ["arn:aws:s3:::companybucket"],
         "Condition":{  "StringNotLike": {"s3:prefix":["Development/*",""] },
                        "Null"         : {"s3:prefix":false }
          }
      }
   ]
}
```

Schritt 6: Erteilen Sie dem IAM Benutzer Bob bestimmte Berechtigungen

Sie können die Schritte wiederholen, um Bob eine ähnliche Berechtigung für den Ordner `Finance` zu erteilen. Führen Sie die Schritte aus, mit denen Sie zuvor Alice Berechtigungen erteilt haben, aber ersetzen Sie den Ordner `Development` durch den Ordner `Finance`. step-by-stepAnweisungen finden Sie unterSchritt 5: Erteilen Sie IAM der Benutzerin Alice spezifische Berechtigungen.

Schritt 7: Absichern des Ordners „Private“

In diesem Beispiel haben Sie nur zwei Benutzer. Sie haben auf Gruppenebene alle erforderlichen Mindestberechtigungen erteilt und auf Benutzerebene die Berechtigungen nur dann gewährt, wenn die einzelnen Benutzer die Berechtigungen wirklich benötigen. Dieser Ansatz minimiert den Aufwand beim Verwalten der Berechtigungen. Wenn die Anzahl der Benutzer steigt, kann das Verwalten der Berechtigungen mühsam werden. Sie möchten z. B. nicht, dass irgendwelche der Benutzer in diesem Beispiel auf den Inhalt des Ordners `Private` zugreifen können. Wie stellen Sie sicher, dass Sie einem Benutzer nicht versehentlich Berechtigungen für den `Private` Ordner erteilen? Sie fügen eine Richtlinie hinzu, die explizit den Zugriff auf den Ordner verweigert. Eine explizite Zugriffsverweigerung überschreibt alle anderen Berechtigungen.

Um sicherzustellen, dass der Ordner `Private` auch privat bleibt, können Sie die folgenden beiden Ablehnungsanweisungen zur Gruppenrichtlinie hinzufügen:

- Fügen Sie die folgende Anweisung hinzu, um jede Aktion auf die Ressourcen im `Private`-Ordner (`companybucket/Private/*`) zu verweigern.

```
{
  "Sid": "ExplictDenyAccessToPrivateFolderToEveryoneInTheGroup",
  "Action": ["s3:*"],
  "Effect": "Deny",
```

```
  "Resource":["arn:aws:s3:::companybucket/Private/*"]
}
```

- Sie verweigern auch die Berechtigung für die Aktion Objekte auflisten, wenn die Anforderung das Präfix `Private/` angibt. Wenn Bob oder Alice in der Konsole den Order `Private` öffnen, bewirkt diese Richtlinie, dass Amazon S3 eine Fehlermeldung zurückgibt.

```
{
  "Sid": "DenyListBucketOnPrivateFolder",
  "Action": ["s3:ListBucket"],
  "Effect": "Deny",
  "Resource": ["arn:aws:s3:::*"],
  "Condition":{
      "StringLike":{"s3:prefix":["Private/"]}
   }
}
```

Ersetzen Sie die Gruppenrichtlinie `Consultants` durch eine aktualisierte Richtlinie, die die vorherigen Ablehnungsanweisungen enthält. Nachdem die aktualisierte Richtlinie angewendet wurde, kann keiner der Benutzer in der Gruppe mehr auf den Order `Private` in Ihrem Bucket zugreifen.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

   Verwenden Sie Ihre AWS-Konto Anmeldeinformationen, nicht die eines IAM Benutzers, um sich an der Konsole anzumelden.

2. Ersetzen Sie die vorhandene von `AllowGroupToSeeBucketListInTheConsole` verwaltete Richtlinie, die der Gruppe `Consultants` zugeordnet ist, durch die folgende Richtlinie. Denken Sie daran, zu ersetzen *`Firma Bucket`* in der Richtlinie mit dem Namen Ihres Buckets.

   Anweisungen finden Sie im IAMBenutzerhandbuch unter Vom Kunden verwaltete Richtlinien bearbeiten. Wenn Sie die Anweisungen nachvollziehen, beachten Sie bitte die Anweisungen zum Ändern aller Hauptentitäten, denen die Richtlinie zugeordnet ist.

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid":
    "AllowGroupToSeeBucketListAndAlsoAllowGetBucketLocationRequiredForListBucket",
   ```

```
      "Action": ["s3:ListAllMyBuckets", "s3:GetBucketLocation"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::*"]
    },
    {
      "Sid": "AllowRootLevelListingOfCompanyBucket",
      "Action": ["s3:ListBucket"],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::companybucket"],
      "Condition":{
          "StringEquals":{"s3:prefix":[""]}
       }
    },
    {
      "Sid": "RequireFolderStyleList",
      "Action": ["s3:ListBucket"],
      "Effect": "Deny",
      "Resource": ["arn:aws:s3:::*"],
      "Condition":{
          "StringNotEquals":{"s3:delimiter":"/"}
       }
     },
    {
      "Sid": "ExplictDenyAccessToPrivateFolderToEveryoneInTheGroup",
      "Action": ["s3:*"],
      "Effect": "Deny",
      "Resource":["arn:aws:s3:::companybucket/Private/*"]
    },
    {
      "Sid": "DenyListBucketOnPrivateFolder",
      "Action": ["s3:ListBucket"],
      "Effect": "Deny",
      "Resource": ["arn:aws:s3:::*"],
      "Condition":{
          "StringLike":{"s3:prefix":["Private/"]}
       }
    }
  ]
}
```

Schritt 8: Bereinigen

Öffnen Sie zum Aufräumen die IAMKonsole und entfernen Sie die Benutzer Alice und Bob. step-by-stepAnweisungen finden Sie im IAMBenutzerhandbuch unter Löschen eines IAM Benutzers.

Um sicherzustellen, dass Sie für die Speicherung nicht künftig belastet werden, sollten Sie auch die Objekte und den Bucket löschen, die Sie für diese Übung erstellt haben.

Zugehörige Ressourcen

- Verwaltung von IAM Richtlinien im IAMBenutzerhandbuch

Beispiele für identitätsbasierte Richtlinien für Amazon S3

In diesem Abschnitt werden mehrere identitätsbasierte AWS Identity and Access Management (IAM) beispielhafte Richtlinien zur Steuerung des Zugriffs auf Amazon S3 gezeigt. Beispiele für Bucket-Richtlinien (ressourcenbasierte Richtlinien) finden Sie unter. Bucket-Richtlinien für Amazon S3 Informationen zur Sprache der IAM Richtlinien finden Sie unter. Richtlinien und Berechtigungen in Amazon S3

Die folgenden Beispielrichtlinien funktionieren, wenn Sie sie programmgesteuert verwenden. Um sie mit der Amazon-S3-Konsole verwenden zu können, müssen Sie aber zusätzliche Berechtigungen gewähren, die für die Konsole erforderlich sind. Informationen zur Verwendung von Richtlinien wie dieser mit der Amazon-S3-Konsole finden Sie unter Kontrollieren des Zugriffs auf einen Bucket mit Benutzerrichtlinien.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

Themen

- Einem IAM Benutzer Zugriff auf einen Ihrer Buckets gewähren
- Erlaubt jedem IAM Benutzer den Zugriff auf einen Ordner in einem Bucket
- Einer Gruppe erlauben, einen freigegebenen Ordner in Amazon S3 zu haben
- Erteilen der Erlaubnis für alle Benutzer, Objekte in einem Teil des Buckets zu lesen
- Erteilen der Erlaubnis für einen Partner, Dateien in einem bestimmten Bereich des Buckets abzulegen
- Beschränken des Zugriffs auf Amazon-S3-Buckets in einem bestimmten AWS-Konto
- Beschränken Sie den Zugriff auf Amazon S3 S3-Buckets innerhalb Ihrer Organisationseinheit

- Beschränken des Zugriffs auf Amazon-S3-Buckets innerhalb Ihrer Organisation
- Erteilung der Erlaubnis zum Abrufen der PublicAccessBlock Konfiguration für ein AWS-Konto
- Beschränken der Bucket-Erstellung auf eine Region

Einem IAM Benutzer Zugriff auf einen Ihrer Buckets gewähren

In diesem Beispiel möchten Sie einem IAM Benutzer in Ihrer Nähe AWS-Konto Zugriff auf einen Ihrer Buckets gewähren, *amzn-s3-demo-bucket1*, und dem Benutzer ermöglichen, Objekte hinzuzufügen, zu aktualisieren und zu löschen.

Zusätzlich zum Erteilen der Berechtigungen `s3:PutObject`, `s3:GetObject` und `s3:DeleteObject` für den Benutzer, gewährt die Richtlinie die Berechtigungen `s3:ListAllMyBuckets`, `s3:GetBucketLocation` und `s3:ListBucket`. Dies sind die zusätzlichen Berechtigungen, die von der Konsole benötigt werden. Außerdem sind die Aktionen `s3:PutObjectAcl` und `s3:GetObjectAcl` erforderlich, um Objekte in der Konsole kopieren, ausschneiden und einfügen zu können. Ein detailliertes Beispiel für eine Richtlinie, die Berechtigungen für Benutzer erteilt und sie unter Verwendung der Konsole testet, finden Sie unter Kontrollieren des Zugriffs auf einen Bucket mit Benutzerrichtlinien.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action": "s3:ListAllMyBuckets",
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":["s3:ListBucket","s3:GetBucketLocation"],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:GetObject",
            "s3:GetObjectAcl",
            "s3:DeleteObject"
```

```
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/*"
      }
   ]
}
```

Erlaubt jedem IAM Benutzer den Zugriff auf einen Ordner in einem Bucket

In diesem Beispiel möchten Sie, dass zwei IAM Benutzer, Mary und Carlos, Zugriff auf Ihren Bucket haben. *amzn-s3-demo-bucket1*, sodass sie Objekte hinzufügen, aktualisieren und löschen können. Allerdings möchten Sie den Zugriff beider Benutzer auf ein einzelnes Präfix (Ordner) im Bucket einschränken. Sie können Ordner mit Namen erstellen, die ihren Benutzernamen entsprechen.

```
amzn-s3-demo-bucket1
   Mary/
   Carlos/
```

Um jedem Benutzer nur den Zugriff auf den eigenen Ordner zu gewähren, können Sie für jeden Benutzer eine Richtlinie schreiben und ihnen einzeln zuweisen. Sie können beispielsweise die folgende Richtlinie der Benutzerin Mary zuweisen, um ihr spezifische Amazon-S3-Berechtigungen für den Ordner *amzn-s3-demo-bucket1*/*Mary* zu gewähren.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/Mary/*"
      }
   ]
}
```

Dann können Sie dem Benutzer Carlos eine ähnliche Richtlinie zuweisen, indem Sie den Ordner *Carlos* im Wert `Resource` angeben.

Statt Richtlinien einzelnen Benutzern zuzuweisen, können Sie auch eine einzelne Richtlinie mit einer Richtlinienvariable schreiben und dann die Richtlinie einer Gruppe zuweisen. Sie müssen zuerst eine Gruppe erstellen und die Benutzer Mary und Carlos in die Gruppe aufnehmen. Die folgende Beispielrichtlinie erlaubt eine Reihe von Amazon-S3-Berechtigungen für den Ordner *amzn-s3-demo-bucket1*`/${aws:username}`. Bei der Auswertung der Richtlinie wird die Richtlinienvariable `${aws:username}` durch den Benutzernamen des Anforderers ersetzt. Wenn Mary beispielsweise eine Anforderung zum Anlegen eines Objekts sendet, ist die Operation nur zulässig, wenn Mary das Objekt in den Ordner *amzn-s3-demo-bucket1*`/Mary` hochlädt.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/${aws:username}/*"
      }
   ]
}
```

> **Note**
>
> Bei der Verwendung von Richtlinienvariablen müssen Sie explizit die Version `2012-10-17` in der Richtlinie angeben. Die Standardversion der IAM Richtliniensprache, 17.10.2008, unterstützt keine Richtlinienvariablen.

Wenn Sie die vorherige Richtlinie in der Amazon-S3-Konsole testen möchten, erfordert die Konsole zusätzliche Berechtigungen, wie in der folgenden Richtlinie gezeigt. Weitere Informationen darüber, wie die Konsole diese Berechtigungen verwendet, finden Sie unter Kontrollieren des Zugriffs auf einen Bucket mit Benutzerrichtlinien.

```
{
 "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowGroupToSeeBucketListInTheConsole",
      "Action": [
       "s3:ListAllMyBuckets",
       "s3:GetBucketLocation"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::*"
    },
    {
      "Sid": "AllowRootLevelListingOfTheBucket",
      "Action": "s3:ListBucket",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1",
      "Condition":{
            "StringEquals":{
                    "s3:prefix":[""], "s3:delimiter":["/"]
                           }
                }
    },
    {
      "Sid": "AllowListBucketOfASpecificUserPrefix",
      "Action": "s3:ListBucket",
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1",
      "Condition":{  "StringLike":{"s3:prefix":["${aws:username}/*"] }
       }
    },
      {
     "Sid": "AllowUserSpecificActionsOnlyInTheSpecificUserPrefix",
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/${aws:username}/*"
      }
```

```
  ]
}
```

> **Note**
>
> In der 2012-10-17-Version der Richtlinie beginnen Richtlinienvariablen mit $. Diese Syntaxänderung kann möglicherweise einen Konflikt verursachen, wenn Ihr Objektschlüssel (Objektname) ein $-Zeichen enthält.
> Damit dieser Konflikt vermieden wird, geben Sie das Zeichen $ mithilfe von `${$}` an. Um beispielsweise einen Objektschlüssel `my$file` in eine Richtlinie aufzunehmen, geben Sie das Zeichen mit `my${$}file` an.

IAMBenutzernamen sind zwar benutzerfreundliche, für Menschen lesbare Kennungen, müssen aber nicht global eindeutig sein. Wenn beispielsweise der Benutzer Carlos die Organisation verlässt und ein anderer Carlos hinzukommt, könnte der neue Carlos auf die Informationen des vorherigen Carlos zugreifen.

Anstatt Benutzernamen zu verwenden, könnten Sie Ordner auf Benutzerbasis erstellen. IAM IDs Jede IAM Benutzer-ID ist einzigartig. In diesem Fall müssen Sie die vorherige Richtlinie ändern, und die Richtlinienvariable `${aws:userid}` verwenden. Weitere Informationen zu Benutzerkennungen finden Sie unter IAMIdentifikatoren im IAMBenutzerhandbuch.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/home/${aws:userid}/*"
      }
   ]
}
```

Erlauben Sie IAM Nichtbenutzern (Benutzern mobiler Apps) den Zugriff auf Ordner in einem Bucket

Angenommen, Sie möchten eine mobile App entwickeln, ein Spiel, in dem Benutzerdaten in einem S3-Bucket gespeichert werden. Sie möchten für jeden App-Benutzer einen Ordner in Ihrem Bucket erstellen. Sie möchten auch den Zugriff jedes Benutzers auf seinen eigenen Ordner einschränken. Sie können jedoch keine Ordner erstellen, bevor jemand Ihre App heruntergeladen und das Spiel gespielt hat, da Sie dessen Benutzer-ID nicht haben.

In diesem Fall können Sie von den Nutzern verlangen, dass Sie sich in Ihrer App über einen öffentlichen Identitätsanbieter anmelden, wie z. B. Login with Amazon, Facebook oder Google. Nachdem sich Benutzer über einen dieser Anbieter bei Ihrer App angemeldet haben, verfügen sie über eine Benutzer-ID, mit der Sie zur Laufzeit benutzerspezifische Ordner erstellen können.

Sie können dann den Web-Identitätsverbund verwenden AWS Security Token Service , um Informationen vom Identitätsanbieter in Ihre App zu integrieren und temporäre Sicherheitsanmeldeinformationen für jeden Benutzer zu erhalten. Anschließend können Sie IAM Richtlinien erstellen, die es der App ermöglichen, auf Ihren Bucket zuzugreifen und beispielsweise benutzerspezifische Ordner zu erstellen und Daten hochzuladen. Weitere Informationen zum Web Identity Federation finden Sie unter About Web Identity Federation im IAMBenutzerhandbuch.

Einer Gruppe erlauben, einen freigegebenen Ordner in Amazon S3 zu haben

Durch Anfügen der folgenden Richtlinie an die Gruppe erhalten alle Benutzer der Gruppe Zugriff auf den folgenden Ordner in Amazon S3: *amzn-s3-demo-bucket1*/share/marketing. Gruppenmitglieder dürfen nur auf die spezifischen Amazon-S3-Berechtigungen zugreifen, die in der Richtlinie gegeben sind, und nur für Objekte im angegebenen Ordner.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/share/marketing/*"
      }
```

```
   ]
}
```

Erteilen der Erlaubnis für alle Benutzer, Objekte in einem Teil des Buckets zu lesen

In diesem Beispiel erstellen Sie eine Gruppe mit dem Namen*AllUsers*, die alle IAM Benutzer enthält, deren Eigentümer der AWS-Konto. Anschließend fügen Sie eine Richtlinie hinzu, die der Gruppe Zugriff auf `GetObject` und `GetObjectVersion` gewährt, jedoch nur für Objekte im Ordner *amzn-s3-demo-bucket1*`/readonly`.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/readonly/*"
      }
   ]
}
```

Erteilen der Erlaubnis für einen Partner, Dateien in einem bestimmten Bereich des Buckets abzulegen

In diesem Beispiel erstellen Sie eine Gruppe namens *AnyCompany*, die eine Partnerfirma darstellt. Sie erstellen einen IAM Benutzer für die bestimmte Person oder Anwendung im Partnerunternehmen, die Zugriff benötigt, und fügen den Benutzer dann der Gruppe hinzu.

Sie fügen dann eine Richtlinie hinzu, die der Gruppe den `PutObject`-Zugriff auf den folgenden Ordner im Bucket erteilt:

amzn-s3-demo-bucket1`/uploads/anycompany`

Sie möchten verhindern, dass die *AnyCompany*-Gruppe andere Aktionen für den Bucket ausführt. Daher fügen Sie eine Anweisung hinzu, die explizit die Berechtigung für andere Amazon-S3-Aktionen verweigert, außer für `PutObject` in einer Amazon-S3-Ressource im AWS-Konto.

```
{
```

```
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"s3:PutObject",
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/uploads/anycompany/*"
      },
      {
         "Effect":"Deny",
         "Action":"s3:*",
         "NotResource":"arn:aws:s3:::amzn-s3-demo-bucket1/uploads/anycompany/*"
      }
   ]
}
```

Beschränken des Zugriffs auf Amazon-S3-Buckets in einem bestimmten AWS-Konto

Wenn Sie sicherstellen möchten, dass Ihre Amazon S3 S3-Prinzipale nur auf die Ressourcen zugreifen, die sich in einem vertrauenswürdigen System befinden AWS-Konto, können Sie den Zugriff einschränken. Diese identitätsbasierte IAM Richtlinie verwendet beispielsweise einen `Deny` Effekt, um den Zugriff auf Amazon S3 S3-Aktionen zu blockieren, es sei denn, die Amazon S3 S3-Ressource, auf die zugegriffen wird, befindet sich im Konto. *222222222222* Um zu verhindern, dass ein IAM Principal in einem AWS-Konto auf Amazon S3 S3-Objekte außerhalb des Kontos zugreift, fügen Sie die folgende IAM Richtlinie bei:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyS3AccessOutsideMyBoundary",
      "Effect": "Deny",
      "Action": [
        "s3:*"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": [
            "222222222222"
          ]
        }
      }
```

```
    }
  ]
}
```

> **Note**
>
> Diese Richtlinie ersetzt nicht Ihre bestehenden IAM Zugriffskontrollen, da sie keinen Zugriff gewährt. Stattdessen dient diese Richtlinie als zusätzliche Schutzmaßnahme für Ihre anderen IAM Berechtigungen, unabhängig davon, welche Berechtigungen im Rahmen anderer IAM Richtlinien gewährt wurden.

Ersetzen Sie unbedingt die Konto-ID *222222222222* in der Richtlinie durch Ihr eigenes AWS-Konto. Wenn Sie eine Richtlinie unter Beibehaltung dieser Einschränkung auf mehrere Konten anwenden möchten, ersetzen Sie die Konto-ID durch den Bedingungsschlüssel `aws:PrincipalAccount`. Diese Bedingung erfordert, dass sich der Prinzipal und die Ressource in demselben Konto befinden müssen.

Beschränken Sie den Zugriff auf Amazon S3 S3-Buckets innerhalb Ihrer Organisationseinheit

Wenn Sie eine Organisationseinheit (OU) eingerichtet haben AWS Organizations, möchten Sie möglicherweise den Zugriff auf den Amazon S3 S3-Bucket auf einen bestimmten Teil Ihrer Organisation beschränken. In diesem Beispiel wird der `aws:ResourceOrgPaths`-Schlüssel verwendet, um den Amazon-S3-Bucket-Zugriff auf eine OU in Ihrer Organisation einzuschränken. In diesem Beispiel lautet die OU-ID *ou-acroot-exampleou*. Stellen Sie sicher, dass Sie diesen Wert in Ihrer eigenen Richtlinie durch Ihre eigene Organisationseinheit ersetzenIDs.

```
{
   "Version": "2012-10-17",
   "Statement": [
     {
       "Sid": "AllowS3AccessOutsideMyBoundary",
       "Effect": "Allow",
       "Action": [
         "s3:*"
       ],
       "Resource": "*",
       "Condition": {
         "ForAllValues:StringNotLike": {
           "aws:ResourceOrgPaths": [
```

```
            "o-acorg/r-acroot/ou-acroot-exampleou/"
          ]
        }
      }
    }
  ]
}
```

> **Note**
>
> Diese Richtlinie gewährt keinen Zugriff. Stattdessen dient diese Richtlinie als Backstop für Ihre anderen IAM Berechtigungen und verhindert, dass Ihre Prinzipale außerhalb einer von der OU definierten Grenze auf Amazon S3 S3-Objekte zugreifen.

Die Richtlinie verweigert den Zugriff auf Amazon-S3-Aktionen, es sei denn, das Amazon-S3-Objekt, auf das zugegriffen wird, befindet sich in der OU *ou-acroot-exampleou* Ihrer Organisation. Die IAMRichtlinienbedingung erfordertaws:ResourceOrgPaths, dass ein mehrwertiger Bedingungsschlüssel einen der aufgelisteten OU-Pfade enthält. Die Richtlinie verwendet den ForAllValues:StringNotLike Operator, um die Werte von mit den aufgelisteten Werten aws:ResourceOrgPaths zu vergleichen, OUs ohne dass Groß- und Kleinschreibung beachtet wird.

Beschränken des Zugriffs auf Amazon-S3-Buckets innerhalb Ihrer Organisation

Um den Zugriff auf Amazon S3 S3-Objekte innerhalb Ihrer Organisation einzuschränken, fügen Sie dem Stamm der Organisation eine IAM Richtlinie hinzu, die auf alle Konten in Ihrer Organisation angewendet wird. Verwenden Sie eine Service-Control-Richtlinie (SCP), um von Ihren IAM Prinzipalen die Einhaltung dieser Regel zu verlangen. Wenn Sie sich für eine entscheidenSCP, stellen Sie sicher, dass Sie sie gründlich testen, SCP bevor Sie die Richtlinie dem Stammverzeichnis der Organisation zuordnen.

In der folgenden Beispielrichtlinie wird der Zugriff auf Amazon S3 S3-Aktionen verweigert, es sei denn, das Amazon S3 S3-Objekt, auf das zugegriffen wird, befindet sich in derselben Organisation wie der IAM Principal, der darauf zugreift:

```
{
   "Version": "2012-10-17",
   "Statement": [
     {
       "Sid": "DenyS3AccessOutsideMyBoundary",
```

```
      "Effect": "Deny",
      "Action": [
        "s3:*"
      ],
      "Resource": "arn:aws:s3:::*/*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceOrgID": "${aws:PrincipalOrgID}"
        }
      }
    }
  ]
}
```

> **Note**
>
> Diese Richtlinie gewährt keinen Zugriff. Stattdessen dient diese Richtlinie als Backstop für Ihre anderen IAM Berechtigungen und verhindert, dass Ihre Prinzipale auf Amazon S3 S3-Objekte außerhalb Ihrer Organisation zugreifen. Diese Richtlinie gilt auch für Amazon-S3-Ressourcen, die nach Inkrafttreten der Richtlinie erstellt werden.

Die IAMRichtlinienbedingung in diesem Beispiel erfordertaws:PrincipalOrgID, dass aws:ResourceOrgID und einander entsprechen. Bei dieser Anforderung müssen sich der Prinzipal, der die Anforderung stellt, und die Ressource, auf die zugegriffen wird, in derselben Organisation befinden.

Erteilung der Erlaubnis zum Abrufen der PublicAccessBlock Konfiguration für ein AWS-Konto

Das folgende Beispiel für eine identitätsbasierte Richtlinie erteilt einem Benutzer die s3:GetAccountPublicAccessBlock entsprechende Berechtigung. Für diese Berechtigungen legen Sie den Wert Resource auf "*" fest. Hinweise zur Ressource ARNs finden Sie unter. Richtlinienressourcen für Amazon S3

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Action":[
```

```
            "s3:GetAccountPublicAccessBlock"
         ],
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Beschränken der Bucket-Erstellung auf eine Region

Angenommen, ein AWS-Konto Administrator möchte seinem Benutzer (Dave) die Erlaubnis erteilen, einen Bucket nur in der Region Südamerika (São Paulo) zu erstellen. Der Kontoadministrator kann die folgende Benutzerrichtlinie anfügen, welche die Berechtigung `s3:CreateBucket` mit der angegebenen Bedingung gewährt. Das Schlüssel-Wert-Paar im Block `Condition` gibt den Schlüssel `s3:LocationConstraint` und die Region `sa-east-1` als seinen Wert an.

> **Note**
>
> In diesem Beispiel erteilt der Bucket-Eigentümer einem seiner Benutzer die Berechtigung, daher kann entweder eine Bucket-Richtlinie oder eine Benutzerrichtlinie verwendet werden. Dieses Beispiel zeigt eine Benutzerrichtlinie.

Die Liste der Amazon-S3-Regionen finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Referenz.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Action": "s3:CreateBucket",
         "Resource": "arn:aws:s3:::*",
         "Condition": {
             "StringLike": {
                 "s3:LocationConstraint": "sa-east-1"
             }
         }
       }
```

```
    ]
}
```

Hinzufügen einer expliziten Zugriffsverweigerung

Die vorangegangene Richtlinie hindert den Benutzer an der Erstellung eines Buckets in einer anderen Region als `sa-east-1`. Einige andere Richtlinien gewähren diesem Benutzer möglicherweise jedoch die Berechtigung, Buckets in einer anderen Region zu erstellen. Wenn der Benutzer beispielsweise zu einer Gruppe gehört, ist der Gruppe möglicherweise eine Richtlinie angefügt, die alle Benutzer der Gruppe zum Erstellen von Buckets in einer anderen Region berechtigt. Um sicherzustellen, dass der Benutzer keine Berechtigung zum Erstellen von Buckets in einer anderen Region erhält, können Sie der obigen Richtlinie eine ausdrückliche Ablehnungsklausel hinzufügen.

Die Anweisung `Deny` verwendet die Bedingung `StringNotLike`. Das bedeutet, dass, eine Anforderung zum Erstellen eines Buckets abgelehnt wird, wenn die Standorteinschränkung nicht `sa-east-1` ist. Die ausdrückliche Ablehnung erlaubt es dem Benutzer nicht, einen Bucket in einer anderen Region zu erstellen, unabhängig davon, welche anderen Berechtigungen der Benutzer erhält. Die folgende Richtlinie beinhaltet eine ausdrückliche Ablehnungsklausel.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"statement1",
         "Effect":"Allow",
         "Action": "s3:CreateBucket",
         "Resource": "arn:aws:s3:::*",
         "Condition": {
             "StringLike": {
                 "s3:LocationConstraint": "sa-east-1"
             }
         }
       },
      {
         "Sid":"statement2",
         "Effect":"Deny",
         "Action": "s3:CreateBucket",
         "Resource": "arn:aws:s3:::*",
         "Condition": {
             "StringNotLike": {
```

```
                "s3:LocationConstraint": "sa-east-1"
            }
         }
      }
   ]
}
```

Testen Sie die Richtlinie mit AWS CLI

Sie können die Richtlinie mit dem folgenden `create-bucket` AWS CLI Befehl testen. In diesem Beispiel wird die Datei `bucketconfig.txt` verwendet, um die Standortbeschränkung anzugeben. Beachten Sie den Windows Dateipfad. Sie müssen den Bucket-Namen und -Pfad entsprechend aktualisieren. Sie müssen die Anmeldeinformationen des Benutzers bereitstellen, indem Sie den Parameter `--profile` hinzufügen. Weitere Informationen zur Einrichtung und Verwendung von finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS CLI in der Amazon S3 API S3-Referenz. AWS CLI

```
aws s3api create-bucket --bucket examplebucket --profile AccountADave --create-bucket-
configuration file://c:/Users/someUser/bucketconfig.txt
```

Die Datei `bucketconfig.txt` gibt die Konfiguration wie folgt an.

```
{"LocationConstraint": "sa-east-1"}
```

Exemplarische Vorgehensweisen, die Richtlinien verwenden, um den Zugriff auf Ihre Amazon S3 S3-Ressourcen zu verwalten

In diesem Thema werden die folgenden einführenden Anleitungsbeispiele für das Gewähren des Zugriffs aufs Amazon-S3-Ressourcen bereitgestellt. In AWS Management Console diesen Beispielen werden Ressourcen (Buckets, Objekte, Benutzer) erstellt und ihnen Berechtigungen erteilt. Anschließend zeigen die Beispiele auf, wie Sie Berechtigungen mithilfe der Befehlszeilen-Tools überprüfen können, sodass Sie keinen Code schreiben müssen. Wir stellen Befehle bereit, die sowohl das AWS Command Line Interface (AWS CLI) als auch das AWS Tools for Windows PowerShell verwenden.

- Beispiel 1: Bucket-Eigentümer erteilt seinen Benutzern Bucket-Berechtigungen

  Die IAM Benutzer, die Sie in Ihrem Konto erstellen, haben standardmäßig keine Berechtigungen. In dieser Übung erteilen Sie einem Benutzer die Berechtigung, Bucket- und Objekt-Vorgänge auszuführen.
- Beispiel 2: Bucket-Eigentümer erteilt kontoübergreifende Bucket-Berechtigungen

  In dieser Übung gewährt ein Bucket-Eigentümer, Konto A, einem anderen AWS-Konto, Konto B, kontoübergreifende Berechtigungen. Konto B delegiert diese Berechtigungen anschließend an Benutzer in seinem Konto.
- Verwalten von Objektberechtigungen, wenn der Objekt- und der Bucket-Eigentümer nicht identisch sind

  Bei den Beispielszenarien in diesem Fall geht es um einen Bucket-Eigentümer, der anderen Objektberechtigungen erteilt, es gehören jedoch nicht alle Objekte im Bucket dem Bucket-Eigentümer. Welche Berechtigungen benötigt der Bucket-Eigentümer und wie kann er diese Berechtigungen delegieren?

  AWS-Konto Derjenige, der einen Bucket erstellt, wird als Bucket-Besitzer bezeichnet. Der Besitzer kann anderen Personen die AWS-Konten Erlaubnis erteilen, Objekte hochzuladen, und derjenige, der Objekte erstellt AWS-Konten , ist Eigentümer dieser Objekte. Der Bucket-Eigentümer hat keine Berechtigungen für diese Objekte, die von anderen AWS-Konten erstellt wurden. Wenn der Bucket-Besitzer eine Bucket-Richtlinie schreibt, die Zugriff auf Objekte gewährt, gilt die Richtlinie nicht für Objekte, die anderen Konten gehören.

  In diesem Fall muss der Objekteigentümer dem Bucket-Besitzer zunächst mithilfe eines Objekts Berechtigungen erteilenACL. Der Bucket-Besitzer kann diese Objektberechtigungen dann an

andere, an Benutzer in seinem eigenen Konto oder an einen anderen delegieren AWS-Konto, wie die folgenden Beispiele zeigen.

- Beispiel 3: Bucket-Eigentümer, der Berechtigungen für Objekte erteilt, die ihm nicht gehören

  In dieser Übung erhält der Bucket-Eigentümer zuerst Berechtigungen von dem Objekteigentümer. Der Bucket-Eigentümer delegiert diese Berechtigungen anschließend an Benutzer in seinem eigenen Konto.

- Beispiel 4: Bucket-Besitzer gewährt kontoübergreifende Berechtigungen für Objekte, die er nicht besitzt

  Nachdem der Bucket-Besitzer die Berechtigungen vom Objekteigentümer erhalten hat, kann er die Berechtigungen nicht an andere delegieren, AWS-Konten da die kontoübergreifende Delegierung nicht unterstützt wird (siehe). Berechtigungsdelegation Stattdessen kann der Bucket-Besitzer eine IAM Rolle mit Berechtigungen zum Ausführen bestimmter Operationen (z. B. zum Abrufen eines Objekts) erstellen und einer anderen Person erlauben, diese Rolle AWS-Konto zu übernehmen. Jeder, der diese Rolle annimmt, kann anschließend auf Objekte zugreifen. Dieses Beispiel zeigt, wie ein Bucket-Besitzer eine IAM Rolle verwenden kann, um diese kontoübergreifende Delegierung zu aktivieren.

Bevor Sie die beispiehalften Walkthroughs ausprobieren

In diesen Beispielen werden Ressourcen erstellt und Berechtigungen erteilt. AWS Management Console Um Berechtigungen zu testen, verwenden die Beispiele die Befehlszeilentools AWS CLI AWS Tools for Windows PowerShell, und, sodass Sie keinen Code schreiben müssen. Zum Testen der Berechtigungen müssen Sie eins dieser Tools einrichten. Weitere Informationen finden Sie unter Einrichtung der Tools für die exemplarischen Vorgehensweisen.

Darüber hinaus verwenden diese Beispiele beim Erstellen von Ressourcen keine Root-Benutzeranmeldedaten von AWS-Konto. Sie erstellen stattdessen einen Administratorbenutzer in diesen Konten, um diese Aufgaben auszuführen.

Informationen zur Verwendung eines Administratorbenutzers zum Erstellen von Ressourcen und Erteilen von Berechtigungen

AWS Identity and Access Management (IAM) empfiehlt, Ihre Root-Benutzeranmeldedaten nicht für Anfragen AWS-Konto zu verwenden. Erstellen Sie stattdessen einen IAM Benutzer oder eine Rolle, gewähren Sie ihnen vollen Zugriff und verwenden Sie dann ihre Anmeldeinformationen, um Anfragen zu stellen. Wir bezeichnen dies als Administratorbenutzer oder -rolle. Weitere Informationen

finden Sie unter Root-Benutzer des AWS-Kontos Anmeldeinformationen und IAM Identitäten im IAMBenutzerhandbuch Allgemeine AWS-Referenzund unter IAMBewährte Methoden.

Alle Beispielanleitungen in diesem Abschnitt verwenden die Anmeldeinformationen des Administratorbenutzers. Falls Sie noch keinen Administratorbenutzer für Ihr Konto erstellt haben, erfahren Sie in den Themen AWS-Konto, wie das geht.

Um sich AWS Management Console mit den Benutzeranmeldedaten bei anzumelden, müssen Sie die IAM Benutzeranmeldung URL verwenden. Die IAMKonsole bietet dies URL für Ihre AWS-Konto. Die Themen zeigen Ihnen, wie Sie die erhaltenURL.

Einrichtung der Tools für die exemplarischen Vorgehensweisen

In den einführenden Beispielen (sieheExemplarische Vorgehensweisen, die Richtlinien verwenden, um den Zugriff auf Ihre Amazon S3 S3-Ressourcen zu verwalten) werden Ressourcen erstellt und Berechtigungen erteilt. AWS Management Console Um Berechtigungen zu testen, verwenden die Beispiele die Befehlszeilentools AWS Command Line Interface (AWS CLI) und AWS Tools for Windows PowerShell, sodass Sie keinen Code schreiben müssen. Zum Testen der Berechtigungen müssen Sie eins dieser Tools einrichten.

Um das einzurichten AWS CLI

1. Herunterladen und Konfigurieren von AWS CLI. Eine Anleitung finden Sie unter den folgenden Themen im AWS Command Line Interface -Benutzerhandbuch:

   Installieren oder aktualisieren Sie auf die neueste Version von AWS Command Line Interface

   Beginnen Sie mit dem AWS Command Line Interface
2. Richten Sie das Standardprofil ein.

   Sie speichern Benutzeranmeldeinformationen in der AWS CLI Konfigurationsdatei. Erstellen Sie mit Ihren AWS-Konto Anmeldeinformationen ein Standardprofil in der Konfigurationsdatei. Anweisungen zum Suchen und Bearbeiten Ihrer AWS CLI Konfigurationsdatei finden Sie unter Einstellungen für die Konfiguration und die Anmeldeinformationsdatei.

   ```
   [default]
   aws_access_key_id = access key ID
   aws_secret_access_key = secret access key
   region = us-west-2
   ```

3. Überprüfen Sie die Einrichtung, indem Sie den folgenden Befehl in die Befehlszeile eingeben. Beide Befehle stellen nicht explizit Anmeldeinformationen bereit, daher werden die Anmeldeinformationen des Standardprofils verwendet.

   - Probieren Sie den `help` Befehl aus.

     ```
     aws help
     ```

   - Verwenden Sie den `aws s3 ls` Befehl, um eine Liste der Buckets für das konfigurierte Konto abzurufen.

     ```
     aws s3 ls
     ```

Während Sie die exemplarischen Vorgehensweisen durchgehen, erstellen Sie Benutzer und speichern Benutzeranmeldeinformationen in den Konfigurationsdateien, indem Sie Profile erstellen, wie das folgende Beispiel zeigt. Diese Profile haben die Namen und. `AccountAadmin` `AccountBadmin`

```
[profile AccountAadmin]
aws_access_key_id = User AccountAadmin access key ID
aws_secret_access_key = User AccountAadmin secret access key
region = us-west-2

[profile AccountBadmin]
aws_access_key_id = Account B access key ID
aws_secret_access_key = Account B secret access key
region = us-east-1
```

Um mit diesen Benutzeranmeldeinformationen einen Befehl auszuführen, fügen Sie den Parameter `--profile` hinzu, um den Profilnamen festzulegen. Mit dem folgenden AWS CLI Befehl wird eine Liste von Objekten abgerufen *`Beispiel/Bucket`* und spezifiziert das `AccountBadmin` Profil.

```
aws s3 ls s3://examplebucket --profile AccountBadmin
```

Alternativ können Sie eine Reihe von Anmeldeinformationen als Standardprofil konfigurieren, indem Sie die Umgebungsvariable `AWS_DEFAULT_PROFILE` von der Befehlszeile aus ändern. Nachdem Sie dies getan haben, AWS CLI verwendet der jedes Mal, wenn Sie AWS CLI Befehle ohne den `--`

`profile` Parameter ausführen, das Profil, das Sie in der Umgebungsvariablen festgelegt haben, als Standardprofil.

```
$ export AWS_DEFAULT_PROFILE=AccountAadmin
```

Zum Einrichten AWS Tools for Windows PowerShell

1. Herunterladen und Konfigurieren von AWS Tools for Windows PowerShell. Anweisungen dazu finden Sie AWS Tools for Windows PowerShell im AWS Tools for Windows PowerShell Benutzerhandbuch unter Installation von.

   > **Note**
   >
   > Um das AWS Tools for Windows PowerShell Modul zu laden, müssen Sie Folgendes aktivieren PowerShell Ausführung des Skripts. Weitere Informationen finden Sie unter Aktivieren der Skriptausführung im AWS Tools for Windows PowerShell Benutzerhandbuch.

2. Für diese exemplarischen Vorgehensweisen geben Sie die AWS Anmeldeinformationen pro Sitzung mithilfe des `Set-AWSCredentials` Befehls an. Der Befehl speichert die Anmeldeinformationen in einem persistenten Speicher (Parameter `-StoreAs`).

   ```
   Set-AWSCredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -
   storeas string
   ```

3. Überprüfen Sie die Einrichtung.

   - Um eine Liste verfügbarer Befehle abzurufen, die Sie für Amazon S3 S3-Operationen verwenden können, führen Sie den `Get-Command` Befehl aus.

     ```
     Get-Command -module awspowershell -noun s3* -StoredCredentials string
     ```

   - Um eine Liste der Objekte in einem Bucket abzurufen, führen Sie den `Get-S3Object` Befehl aus.

     ```
     Get-S3Object -BucketName bucketname -StoredCredentials string
     ```

Eine Liste der Befehle finden Sie unter AWS Tools for PowerShell Cmdlet Reference.

Jetzt sind Sie bereit, die exemplarischen Vorgehensweisen auszuprobieren. Folgen Sie den Links am Anfang jedes Abschnitts.

Beispiel 1: Bucket-Eigentümer erteilt seinen Benutzern Bucket-Berechtigungen

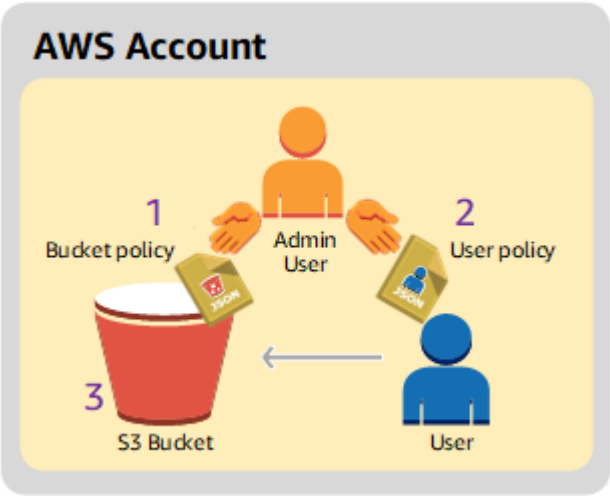
> **Important**
>
> Das Erteilen von Berechtigungen für IAM Rollen ist eine bessere Methode als das Erteilen von Berechtigungen für einzelne Benutzer. Weitere Informationen darüber, wie Sie IAM Rollen Berechtigungen gewähren, finden Sie unter. Grundlegendes zu kontoübergreifenden Berechtigungen und zur Verwendung von Rollen IAM

Themen

- Vorbereitung auf den Walkthrough
- Schritt 1: Erstellen Sie Ressourcen in Konto A und gewähren Sie Berechtigungen
- Schritt 2: Testen der Berechtigungen

In dieser exemplarischen Vorgehensweise AWS-Konto besitzt ein Benutzer einen Bucket, und das Konto umfasst einen IAM Benutzer. Standardmäßig hat der Benutzer keine Berechtigungen. Damit der Benutzer alle Aufgaben ausführen kann, muss das übergeordnete Konto ihm Berechtigungen erteilen. Der Bucket-Eigentümer und das übergeordnete Konto sind identisch. Um dem Benutzer Berechtigungen für den Bucket zu gewähren, AWS-Konto kann er daher eine Bucket-Richtlinie, eine Benutzerrichtlinie oder beides verwenden. Der Kontobesitzer gewährt einige Berechtigungen unter Verwendung einer Bucket-Richtlinie und andere Berechtigungen unter Verwendung einer Benutzerrichtlinie.

Die folgenden Schritte fasen das detaillierte Beispiel zusammen:



1. Der Kontoadministrator erstellt eine Bucket-Richtlinie, die dem Benutzer verschiedene Berechtigungen erteilt.
2. Der Kontoadministrator weist dem Benutzer eine Benutzerrichtlinie zu, die ihm zusätzliche Berechtigungen erteilt.
3. Anschließend probiert der Benutzer Berechtigungen aus, die über die Bucket-Richtlinie und die Benutzerrichtlinie erteilt wurden.

Für dieses Beispiel benötigen Sie eine AWS-Konto. Anstatt die Anmeldeinformationen des Root-Benutzers für das Konto zu verwenden, erstellen Sie einen Administrator-Benutzer (siehe Informationen zur Verwendung eines Administratorbenutzers zum Erstellen von Ressourcen und Erteilen von Berechtigungen). Wir beziehen uns auf den AWS-Konto und den Administratorbenutzer, wie in der folgenden Tabelle dargestellt.

Konto-ID	Konto bezeichnet als	Administratorbenutzer im Konto
1111-1111-1111	Konto A	AccountAadmin

> **Note**
>
> Der Administratorbenutzer in diesem Beispiel ist AccountAadmin, was sich auf Konto A bezieht, und nicht AccountAdmin.

Alle Aufgaben in Verbindung mit dem Erstellen von Benutzern und Gewähren von Berechtigungen werden in der AWS Management Console ausgeführt. Um die Berechtigungen zu überprüfen, verwendet die exemplarische Vorgehensweise die Befehlszeilentools AWS Command Line Interface (AWS CLI) und AWS Tools for Windows PowerShell, sodass Sie keinen Code schreiben müssen.

Vorbereitung auf den Walkthrough

1. Stellen Sie sicher, dass Sie über eine verfügen AWS-Konto und dass sie über einen Benutzer mit Administratorrechten verfügt.

   a. Melden Sie sich bei Bedarf für eine an AWS-Konto. Wir bezeichnen dieses Konto als Konto A.

i. Gehen Sie zu https://aws.amazon.com/s3 und wählen Sie AWS Konto erstellen.

ii. Folgen Sie den Anweisungen auf dem Bildschirm.

AWS benachrichtigt Sie per E-Mail, wenn Ihr Konto aktiv und für Sie verfügbar ist.

b. Erstellen Sie in Konto A einen Administratorbenutzer**`AccountAadmin`**. Melden Sie sich mit den Anmeldeinformationen von Konto A bei der IAMKonsole an und gehen Sie wie folgt vor:

i. Erstellen Sie einen Benutzer **`AccountAadmin`** und notieren Sie sich die Sicherheitsanmeldeinformationen des Benutzers.

Anweisungen finden Sie im IAM Benutzerhandbuch unter Einen Benutzer AWS-Konto in Ihrem erstellen. IAM

ii. Gewähren Sie Administratorrechte, AccountAadminindem Sie eine Benutzerrichtlinie anhängen, die vollen Zugriff gewährt.

Anweisungen finden Sie im IAMBenutzerhandbuch unter IAMRichtlinien verwalten.

iii. Notieren Sie sich die IAMBenutzeranmeldung URL für AccountAadmin. Sie müssen dies verwenden, URL wenn Sie sich bei der AWS Management Console anmelden. Weitere Informationen darüber, wo Sie die Anmeldung findenURL, finden Sie im Benutzerhandbuch unter AWS Management Console Als IAM Benutzer anmelden. IAM Beachten Sie die URL für jedes Konto.

2. Richten Sie entweder das AWS CLI oder das ein AWS Tools for Windows PowerShell. Stellen Sie sicher, dass Sie die Administrator-Benutzeranmeldedaten wie folgt speichern:

- Wenn Sie das verwenden AWS CLI, erstellen Sie ein Profil`AccountAadmin`,, in der Konfigurationsdatei.
- Wenn Sie die verwenden AWS Tools for Windows PowerShell, stellen Sie sicher, dass Sie die Anmeldeinformationen für die Sitzung als speichern`AccountAadmin`.

Detaillierte Anweisungen finden Sie unter Einrichtung der Tools für die exemplarischen Vorgehensweisen.

Schritt 1: Erstellen Sie Ressourcen in Konto A und gewähren Sie Berechtigungen

Melden Sie sich mit den Anmeldeinformationen des Benutzers `AccountAadmin` in Konto A und der speziellen IAM Benutzeranmeldung URL bei der an AWS Management Console und gehen Sie wie folgt vor:

1. Erstellen Sie Ressourcen aus einem Bucket und einem Benutzer IAM

   a. Erstellen Sie in der Amazon-S3-Konsole einen Bucket. Notieren Sie sich den Bucket, AWS-Region in dem Sie den Bucket erstellt haben. Detaillierte Anweisungen finden Sie unter Erstellen eines Buckets.

   b. Gehen Sie in der IAMKonsole wie folgt vor:

      i. Erstellen Sie einen Benutzer mit dem Namen Dave.

         step-by-stepAnweisungen finden Sie im Benutzerhandbuch unter IAMBenutzer erstellen (Konsole). IAM

      ii. Notieren Sie sich die `UserDave` Anmeldeinformationen.

      iii. Notieren Sie sich den Amazon-Ressourcennamen (ARN) für den Benutzer Dave. Wählen Sie in der IAMKonsole den Benutzer aus, und auf der Registerkarte Zusammenfassung wird der Benutzer angezeigtARN.

2. Erteilen Sie Berechtigungen.

   Da der Bucket-Besitzer und das übergeordnete Konto, zu dem der Benutzer gehört, identisch sind, AWS-Konto können sie Benutzerberechtigungen mithilfe einer Bucket-Richtlinie, einer Benutzerrichtlinie oder beidem gewähren. In diesem Beispiel machen Sie beides. Wenn das Objekt auch demselben Konto gehört, kann der Bucket-Besitzer Objektberechtigungen in der Bucket-Richtlinie (oder einer IAM Richtlinie) gewähren.

   a. Fügen Sie in der Amazon S3 S3-Konsole die folgende Bucket-Richtlinie hinzu *`awsexamplebucket1`*.

      Die Richtlinie enthält zwei Anweisungen.

      - Die erste Anweisung erteilt Dave Berechtigungen für die Bucket-Operationen `s3:GetBucketLocation` und `s3:ListBucket`.
      - Die zweite Anweisung erteilt die `s3:GetObject`-Berechtigung. Konto A gehört auch das Objekt, deshalb kann der Kontoadministrator die `s3:GetObject`-Berechtigung erteilen.

In der `Principal` Erklärung wird Dave von seinem Benutzer identifiziertARN. Weitere Informationen zu Richtlinienelementen finden Sie unter Richtlinien und Berechtigungen in Amazon S3.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "statement1",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
         },
         "Action": [
            "s3:GetBucketLocation",
            "s3:ListBucket"
         ],
         "Resource": [
            "arn:aws:s3:::awsexamplebucket1"
         ]
      },
      {
         "Sid": "statement2",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
         },
         "Action": [
             "s3:GetObject"
         ],
         "Resource": [
            "arn:aws:s3:::awsexamplebucket1/*"
         ]
      }
   ]
}
```

b. Erstellen Sie mithilfe der folgenden Richtlinie eine Inlinerichtlinie für den Benutzer Dave. Die Richtlinie erteilt dem Benutzer Dave die `s3:PutObject`-Berechtigung. Sie müssen die Richtlinie aktualisieren, indem Sie Ihren Bucket-Namen angeben.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "PermissionForObjectOperations",
         "Effect": "Allow",
         "Action": [
            "s3:PutObject"
         ],
         "Resource": [
            "arn:aws:s3:::awsexamplebucket1/*"
         ]
      }
   ]
}
```

Anweisungen finden Sie IAMpolicies im IAMBenutzerhandbuch unter Verwalten. Beachten Sie, dass Sie sich mit den Anmeldeinformationen von Konto A an der Konsole anmelden müssen.

Schritt 2: Testen der Berechtigungen

Überprüfen Sie unter Verwendung der Anmeldeinformationen von Dave, ob die Berechtigungen funktionieren. Sie haben die Wahl zwischen den folgenden beiden Verfahren.

Testen Sie die Berechtigungen mit dem AWS CLI

1. Aktualisieren Sie die AWS CLI Konfigurationsdatei, indem Sie das folgende `UserDaveAccountA` Profil hinzufügen. Weitere Informationen finden Sie unter Einrichtung der Tools für die exemplarischen Vorgehensweisen.

   ```
   [profile UserDaveAccountA]
   aws_access_key_id = access-key
   aws_secret_access_key = secret-access-key
   region = us-east-1
   ```

2. Überprüfen Sie, ob Dave Vorgänge ausführen kann, für die ihm in der Benutzerrichtlinie Berechtigungen erteilt wurden. Laden Sie mit dem folgenden AWS CLI `put-object` Befehl ein Beispielobjekt hoch.

Der Parameter `--body` im Befehl identifiziert die hochzuladende Quelldatei. Wenn sich die Datei beispielsweise im Stammverzeichnis des Laufwerks C: auf einem Windows Maschine, die Sie angeben`c:\HappyFace.jpg`. Der Parameter `--key` gibt den Schlüsselnamen für das Objekt an.

```
aws s3api put-object --bucket awsexamplebucket1 --key HappyFace.jpg --
body HappyFace.jpg --profile UserDaveAccountA
```

Führen Sie den folgenden AWS CLI Befehl aus, um das Objekt abzurufen.

```
aws s3api get-object --bucket awsexamplebucket1 --key HappyFace.jpg OutputFile.jpg
 --profile UserDaveAccountA
```

Testen Sie die Berechtigungen mit dem AWS Tools for Windows PowerShell

1. Speichern Sie Daves Anmeldeinformationen als`AccountADave`. Anschließend verwenden Sie diese Anmeldeinformationen, um PUT GET ein Objekt zu erstellen.

   ```
   set-awscredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -storeas
    AccountADave
   ```

2. Laden Sie mithilfe des AWS Tools for Windows PowerShell `Write-S3Object` Befehls ein Beispielobjekt hoch und verwenden Sie dabei die gespeicherten Anmeldeinformationen von Benutzer Dave.

   ```
   Write-S3Object -bucketname awsexamplebucket1 -key HappyFace.jpg -file HappyFace.jpg
    -StoredCredentials AccountADave
   ```

   Laden Sie das zuvor hochgeladene Objekt herunter.

   ```
   Read-S3Object -bucketname awsexamplebucket1 -key HappyFace.jpg -file Output.jpg -
   StoredCredentials AccountADave
   ```

Beispiel 2: Bucket-Eigentümer erteilt kontoübergreifende Bucket-Berechtigungen

> **Important**
>
> Das Erteilen von Berechtigungen für IAM Rollen ist eine bessere Methode als das Erteilen von Berechtigungen für einzelne Benutzer. Weitere Informationen zur Vorgehensweise finden Sie unter Grundlegendes zu kontoübergreifenden Berechtigungen und zur Verwendung von Rollen IAM.

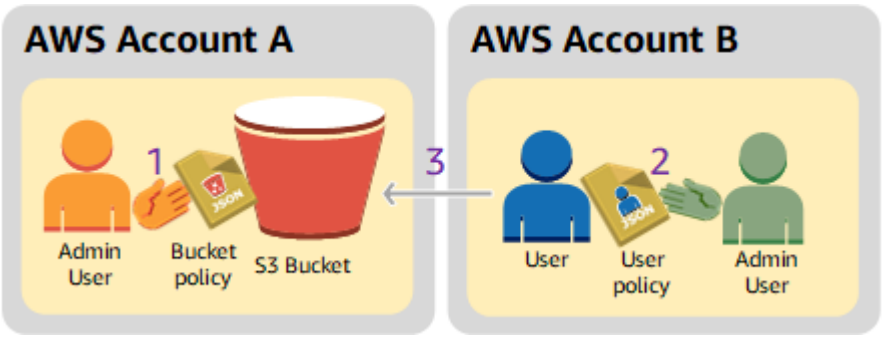
Themen

- Vorbereitung auf den Walkthrough
- Schritt 1: Erledigen der Aufgaben von Konto A
- Schritt 2: Erledigen der Aufgaben von Konto B
- Schritt 3: (Optional) Versuchen Sie eine explizite Zugriffsverweigerung
- Schritt 4: Bereinigen

Ein Konto AWS-Konto— beispielsweise Konto A — kann einem anderen Konto AWS-Konto, Konto B, die Erlaubnis erteilen, auf seine Ressourcen wie Buckets und Objekte zuzugreifen. Konto B kann diese Berechtigungen an Benutzer in seinem Konto delegieren. In diesem Beispielszenario erteilt ein Bucket-Eigentümer einem anderen Konto eine kontenübergreifende Berechtigung, um bestimmte Bucket-Vorgänge auszuführen.

> **Note**
>
> Konto A kann einem Benutzer in Konto B unter Verwendung einer Bucket-Richtlinie auch direkt Berechtigungen erteilen. Der Benutzer benötigt jedoch weiterhin die Erlaubnis des übergeordneten Kontos, Konto B, zu dem der Benutzer gehört, auch wenn Konto B keine Berechtigungen von Konto A hat. Solange der Benutzer sowohl vom Eigentümer der Ressource als auch vom übergeordneten Konto über die entsprechenden Berechtigungen verfügt, kann der Benutzer auf die Ressource zugreifen.

Nachfolgend finden Sie eine kurze Zusammenfassung der wichtigsten Details:



1. Der Administrator-Benutzer von Konto A ordnet eine Bucket-Richtlinie zu, die Konto B kontenübergreifende Berechtigungen erteilt, um bestimmte Bucket-Vorgänge auszuführen.

   Beachten Sie, dass der Administrator-Benutzer in Konto B die Berechtigungen automatisch erbt.
2. Der Administrator-Benutzer von Konto B ordnet dem Benutzer eine Benutzerrichtlinie zu, die die Berechtigungen an den Benutzer delegiert, die er von Konto A erhalten hat.
3. Der Benutzer in Konto B überprüft die Berechtigungen, indem er auf ein Objekt in dem Bucket zugreift, das Konto A gehört.

Für dieses Beispiel benötigen Sie zwei Konten. Die folgende Tabelle zeigt, wie wir auf diese Konten und die Administrator-Benutzer darin verweisen. Gemäß den IAM Richtlinien (siehe Informationen zur Verwendung eines Administratorbenutzers zum Erstellen von Ressourcen und Erteilen von Berechtigungen) verwenden wir in dieser exemplarischen Vorgehensweise keine Root-Benutzeranmeldedaten. Stattdessen erstellen Sie einen Administrator-Benutzer in jedem Konto, und verwenden dessen Anmeldeinformationen, um Ressourcen zu erstellen und ihnen Berechtigungen zu erteilen.

AWS-Konto ID	Konto bezeichnet als	Administratorbenutzer im Konto
1111-1111-1111	Konto A	AccountAadmin
2222-2222-2222	Konto B	AccountBadmin

Alle Aufgaben in Verbindung mit dem Erstellen von Benutzern und Gewähren von Berechtigungen werden in der AWS Management Console ausgeführt. Um die Berechtigungen zu überprüfen, werden in der exemplarischen Vorgehensweise die Befehlszeilentools AWS Command Line Interface (CLI) und verwendet AWS Tools for Windows PowerShell, sodass Sie keinen Code schreiben müssen.

Vorbereitung auf den Walkthrough

1. Stellen Sie sicher, dass Sie über zwei Benutzer verfügen AWS-Konten und dass jedes Konto über einen Administratorbenutzer verfügt, wie in der Tabelle im vorherigen Abschnitt dargestellt.

   a. Melden Sie sich bei Bedarf für einen an AWS-Konto.

   b. Melden Sie sich mit den Anmeldeinformationen von Konto A bei der IAMKonsole an, um den Administratorbenutzer zu erstellen:

      i. Erstellen Sie einen Benutzer **`AccountAadmin`** und notieren Sie sich die Sicherheitsanmeldedaten. Eine Anleitung dazu finden Sie im IAM Benutzerhandbuch unter Einen IAMBenutzer AWS-Konto in Ihrem erstellen.

      ii. Gewähren Sie Administratorrechte, AccountAadminindem Sie eine Benutzerrichtlinie anhängen, die vollen Zugriff gewährt. Anweisungen finden Sie im IAMBenutzerhandbuch unter Arbeiten mit Richtlinien.

   c. Notieren Sie sich die IAMBenutzeranmeldung URL auf dem Dashboard, während Sie sich in der IAM Konsole befinden. Alle Benutzer im Konto müssen dies URL bei der Anmeldung beim AWS Management Console verwenden.

      Weitere Informationen finden Sie im Benutzerhandbuch unter So melden sich IAM Benutzer bei Ihrem Konto an.

   d. Wiederholen Sie den vorherigen Schritt mit den Anmeldeinformationen für Konto B und erstellen Sie einen Administratorbenutzer**`AccountBadmin`**.

2. Richten Sie entweder die AWS Command Line Interface (AWS CLI) oder die ein AWS Tools for Windows PowerShell. Stellen Sie sicher, dass Sie die Administrator-Benutzeranmeldedaten wie folgt speichern:

   - Wenn Sie das verwenden AWS CLI, erstellen Sie zwei Profile, `AccountAadmin` und`AccountBadmin`, in der Konfigurationsdatei.
   - Wenn Sie die verwenden AWS Tools for Windows PowerShell, stellen Sie sicher, dass Sie die Anmeldeinformationen für die Sitzung als `AccountAadmin` und speichern`AccountBadmin`.

   Detaillierte Anweisungen finden Sie unter Einrichtung der Tools für die exemplarischen Vorgehensweisen.

3. Speichern Sie die Anmeldeinformationen des Administrator-Benutzers, auch als Profile bezeichnet. Sie können den Profilnamen verwenden, statt für jeden eingegebenen Befehl

Anmeldeinformationen anzugeben. Weitere Informationen finden Sie unter Einrichtung der Tools für die exemplarischen Vorgehensweisen.

a. Fügen Sie der AWS CLI Anmeldeinformationsdatei Profile für jeden der Administratorbenutzer `AccountAadmin` und `AccountBadmin` in den beiden Konten hinzu.

```
[AccountAadmin]
aws_access_key_id = access-key-ID
aws_secret_access_key = secret-access-key
region = us-east-1

[AccountBadmin]
aws_access_key_id = access-key-ID
aws_secret_access_key = secret-access-key
region = us-east-1
```

b. Wenn Sie den verwenden AWS Tools for Windows PowerShell, führen Sie den folgenden Befehl aus.

```
set-awscredentials -AccessKey AcctA-access-key-ID -SecretKey AcctA-secret-
access-key -storeas AccountAadmin
set-awscredentials -AccessKey AcctB-access-key-ID -SecretKey AcctB-secret-
access-key -storeas AccountBadmin
```

Schritt 1: Erledigen der Aufgaben von Konto A

Schritt 1.1: Melden Sie sich an AWS Management Console

Melden Sie sich mit der IAM Benutzeranmeldung URL für Konto A zunächst AWS Management Console als AccountAadminBenutzer an. Dieser Benutzer erstellt einen Bucket und ordnet ihm eine Richtlinie zu.

Schritt 1.2: Erstellen eines Buckets

1. Erstellen Sie in der Amazon-S3-Konsole einen Bucket. In dieser Übung wird davon ausgegangen, dass der Bucket im Osten der USA (Nord-Virginia) erstellt AWS-Region und benannt wurde*amzn-s3-demo-bucket*.

   Detaillierte Anweisungen finden Sie unter Erstellen eines Buckets.

2. Hochladen eines Beispielobjekts in den Bucket.

Anweisungen finden Sie unter Schritt 2: Hochladen eines Objekts in Ihren Bucket.

Schritt 1.3: Zuordnen einer Bucket-Richtlinie, um Konto B kontoübergreifende Berechtigungen zu erteilen

Die Bucket-Richtlinie gewährt Konto B die `s3:ListBucket` Berechtigungen `s3:GetLifecycleConfiguration` und. Es wird davon ausgegangen, dass Sie immer noch mit AccountAadminBenutzeranmeldedaten bei der Konsole angemeldet sind.

1. Weisen Sie *amzn-s3-demo-bucket* die folgende Bucket-Richtlinie zu. Die Richtlinie erteilt Konto B die Berechtigung für die Aktionen `s3:GetLifecycleConfiguration` und `s3:ListBucket`.

   Detaillierte Anweisungen finden Sie unter Hinzufügen einer Bucket-Richtlinie mit der Amazon-S3-Konsole.

   ```
   {
      "Version": "2012-10-17",
      "Statement": [
         {
            "Sid": "Example permissions",
            "Effect": "Allow",
            "Principal": {
               "AWS": "arn:aws:iam::AccountB-ID:root"
            },
            "Action": [
               "s3:GetLifecycleConfiguration",
               "s3:ListBucket"
            ],
            "Resource": [
               "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
         }
      ]
   }
   ```

2. Stellen Sie sicher, dass Konto B (und damit sein Administratorbenutzer) die Operationen ausführen kann.

   - Überprüfen Sie mit dem AWS CLI

```
aws s3 ls s3://amzn-s3-demo-bucket --profile AccountBadmin
aws s3api get-bucket-lifecycle-configuration --bucket amzn-s3-demo-bucket --
profile AccountBadmin
```

- Überprüfen Sie mit dem AWS Tools for Windows PowerShell

```
get-s3object -BucketName amzn-s3-demo-bucket -StoredCredentials AccountBadmin
get-s3bucketlifecycleconfiguration -BucketName amzn-s3-demo-bucket -
StoredCredentials AccountBadmin
```

Schritt 2: Erledigen der Aufgaben von Konto B

Jetzt erstellt der Administrator von Konto B einen Benutzer Dave und delegiert an Dave die Berechtigungen, die er von Konto A erhalten hat.

Schritt 2.1: Melden Sie sich an bei AWS Management Console

Melden Sie sich mit der IAM Benutzeranmeldung URL für Konto B zunächst AWS Management Console als AccountBadminBenutzer an.

Schritt 2.2: Erstellen des Benutzers Dave in Konto B

Erstellen Sie in der IAMKonsole einen Benutzer,**`Dave`**.

Anweisungen finden Sie im Benutzerhandbuch unter IAMBenutzer erstellen (Konsole). IAM

Schritt 2.3: Delegieren von Berechtigungen an den Benutzer Dave

Erstellen Sie mithilfe der folgenden Richtlinie eine Inlinerichtlinie für den Benutzer Dave. Sie müssen die Richtlinie aktualisieren, indem Sie Ihren Bucket-Namen angeben.

Es wird davon ausgegangen, dass Sie mit AccountBadminBenutzeranmeldedaten bei der Konsole angemeldet sind.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Example",
```

```
         "Effect": "Allow",
         "Action": [
            "s3:ListBucket"
         ],
         "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket"
         ]
      }
   ]
}
```

Anweisungen finden Sie im IAMBenutzerhandbuch unter IAMRichtlinien verwalten.

Schritt 2.4: Testen der Berechtigungen

Jetzt kann Dave in Konto B den Inhalt von *amzn-s3-demo-bucket* auflisten, der Konto A gehört. Sie können die Berechtigungen mit einem der folgenden Verfahren überprüfen.

Testen Sie die Berechtigungen mit dem AWS CLI

1. Fügen Sie das `UserDave` Profil der AWS CLI Konfigurationsdatei hinzu. Weitere Informationen zur Config-Datei finden Sie unte Einrichtung der Tools für die exemplarischen Vorgehensweisen.

   ```
   [profile UserDave]
   aws_access_key_id = access-key
   aws_secret_access_key = secret-access-key
   region = us-east-1
   ```

2. Geben Sie an der Befehlszeile den folgenden AWS CLI Befehl ein, um zu überprüfen, ob Dave jetzt eine Objektliste aus dem *amzn-s3-demo-bucket* Konto A abrufen kann. Beachten Sie, dass der Befehl das `UserDave` Profil angibt.

   ```
   aws s3 ls s3://amzn-s3-demo-bucket --profile UserDave
   ```

   Dave hat keine anderen Berechtigungen. Wenn er also eine andere Operation versucht, z. B. die folgende `get-bucket-lifecycle` Konfiguration, gibt Amazon S3 die Berechtigung verweigert zurück.

   ```
   aws s3api get-bucket-lifecycle-configuration --bucket amzn-s3-demo-bucket --profile
    UserDave
   ```

Testen Sie die Berechtigungen mit AWS Tools for Windows PowerShell

1. Speichern Sie Daves Anmeldeinformationen als`AccountBDave`.

```
set-awscredentials -AccessKey AccessKeyID -SecretKey SecretAccessKey -storeas
 AccountBDave
```

2. Probieren Sie den Befehl List Bucket aus.

```
get-s3object -BucketName amzn-s3-demo-bucket -StoredCredentials AccountBDave
```

   Dave hat keine anderen Berechtigungen. Wenn er also einen anderen Vorgang versucht, z. B. den folgenden, gibt `get-s3bucketlifecycleconfiguration` Amazon S3 die Berechtigung verweigert zurück.

```
get-s3bucketlifecycleconfiguration -BucketName amzn-s3-demo-bucket -
StoredCredentials AccountBDave
```

Schritt 3: (Optional) Versuchen Sie eine explizite Zugriffsverweigerung

Sie können sich Berechtigungen mithilfe einer Zugriffskontrollliste (ACL), einer Bucket-Richtlinie oder einer Benutzerrichtlinie gewähren lassen. Wenn jedoch entweder durch eine Bucket-Richtlinie oder eine Benutzerrichtlinie eine ausdrückliche Verweigerung festgelegt wird, hat die ausdrückliche Verweigerung Vorrang vor allen anderen Berechtigungen. Aktualisieren Sie zum Testen die Bucket-Richtlinie und verweigern Sie Konto B ausdrücklich die `s3:ListBucket` Erlaubnis. Die Richtlinie gewährt auch die `s3:ListBucket` Erlaubnis. Die ausdrückliche Ablehnung hat jedoch Vorrang, und Konto B oder Benutzer in Konto B können keine Objekte auflisten. *amzn-s3-demo-bucket*

1. Wenn Sie die Anmeldeinformationen des Benutzers `AccountAadmin` in Konto A verwenden, ersetzen Sie die Bucket-Richtlinie durch Folgendes.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Example permissions",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountB-ID:root"
```

```
            },
            "Action": [
               "s3:GetLifecycleConfiguration",
               "s3:ListBucket"
            ],
            "Resource": [
               "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
         },
         {
            "Sid": "Deny permission",
            "Effect": "Deny",
            "Principal": {
               "AWS": "arn:aws:iam::AccountB-ID:root"
            },
            "Action": [
               "s3:ListBucket"
            ],
            "Resource": [
               "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
         }
      ]
}
```

2. Wenn Sie nun versuchen, mithilfe von `AccountBadmin` Anmeldeinformationen eine Bucket-Liste zu erstellen, wird der Zugriff verweigert.

   - Führen Sie mit dem AWS CLI den folgenden Befehl aus:

     ```
     aws s3 ls s3://amzn-s3-demo-bucket --profile AccountBadmin
     ```

   - Führen Sie mit dem AWS Tools for Windows PowerShell den folgenden Befehl aus:

     ```
     get-s3object -BucketName amzn-s3-demo-bucket -StoredCredentials AccountBDave
     ```

Schritt 4: Bereinigen

1. Nachdem Sie mit dem Testen fertig sind, können Sie zum Aufräumen wie folgt vorgehen:

   - Melden Sie sich mit den Anmeldeinformationen von Konto A bei AWS Management Console (AWS Management Console) an und gehen Sie wie folgt vor:

- Entfernen Sie in der Amazon-S3-Konsole die an *amzn-s3-demo-bucket* angefügte Bucket-Richtlinie. Löschen Sie in den Bucket Properties die Richtlinie im Abschnitt Permissions.
- Wenn der Bucket für diese Übung erstellt wurde, löschen Sie in der Amazon-S3-Konsole die Objekte und dann den Bucket.
- Entfernen Sie den `AccountAadmin` Benutzer in der IAMKonsole.

2. Melden Sie sich mit den Anmeldeinformationen für Konto B bei der IAMKonsole an. Benutzer löschen`AccountBadmin`. step-by-stepAnweisungen finden Sie im IAMBenutzerhandbuch unter Löschen eines IAM Benutzers.

Beispiel 3: Bucket-Eigentümer, der Berechtigungen für Objekte erteilt, die ihm nicht gehören

> **Important**
>
> Das Erteilen von Berechtigungen für IAM Rollen ist eine bessere Methode als das Erteilen von Berechtigungen für einzelne Benutzer. Weitere Informationen zur Vorgehensweise finden Sie unter Grundlegendes zu kontoübergreifenden Berechtigungen und zur Verwendung von Rollen IAM.

Themen

- Schritt 0: Vorbereitung auf den Walkthrough
- Schritt 1: Erledigen der Aufgaben von Konto A
- Schritt 2: Erledigen der Aufgaben von Konto B
- Schritt 3: Testen der Berechtigungen
- Schritt 4: Bereinigen

Das Szenario für dieses Beispiel ist, dass ein Bucket-Besitzer die Erlaubnis zum Zugriff auf Objekte gewähren möchte, aber der Bucket-Besitzer besitzt nicht alle Objekte im Bucket. In diesem Beispiel versucht der Bucket-Eigentümer, Benutzern in seinem eigenen Konto eine Berechtigung zu erteilen.

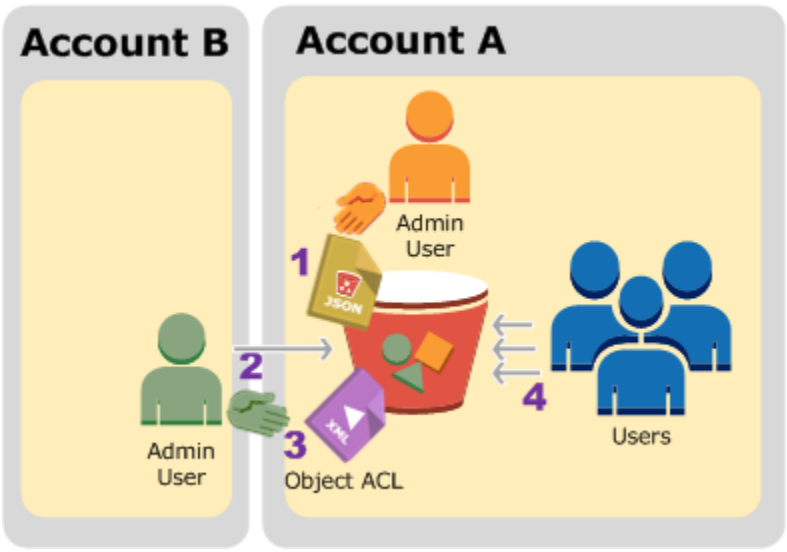
Ein Bucket-Besitzer kann anderen AWS-Konten das Hochladen von Objekten ermöglichen. Standardmäßig besitzt der Bucket-Eigentümer keine Objekte, die von einem anderen AWS-Konto

in einen Bucket geschrieben wurden. Objekte gehören den Konten, die sie in einen S3-Bucket schreiben. Wenn der Bucket-Besitzer keine Objekte im Bucket besitzt, muss der Objekteigentümer dem Bucket-Besitzer zunächst mithilfe einer Objektzugriffskontrollliste (ACL) die Erlaubnis erteilen. Anschließend kann der Bucket-Besitzer einem Objekt, das er nicht besitzt, Berechtigungen gewähren. Weitere Informationen finden Sie unter Amazon-S3-Bucket- und Objekt-Eigentümerschaft.

Wenn der Bucket-Eigentümer die vom Bucket-Eigentümer erzwungene Einstellung für S3 Object Ownership für den Bucket anwendet, besitzt der Bucket-Eigentümer alle Objekte im Bucket, einschließlich der Objekte, die von einem anderen AWS-Konto geschrieben wurden. Dieser Ansatz löst das Problem, dass Objekte nicht dem Bucket-Besitzer gehören. Anschließend können Sie die Berechtigung an Benutzer in Ihrem eigenen Konto oder an andere AWS-Konten.

> **Note**
>
> S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie sowohl das Eigentum an den Objekten kontrollieren können, die in Ihren Bucket hochgeladen werden, als auch um sie zu deaktivieren oder zu aktivieren. ACLs Standardmäßig ist für Object Ownership die erzwungene Einstellung des Bucket-Besitzers festgelegt, und alle sind ACLs deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff darauf ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.
> Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Wenn diese ACLs Option deaktiviert ist, können Sie mithilfe von Richtlinien den Zugriff auf alle Objekte in Ihrem Bucket steuern, unabhängig davon, wer die Objekte in Ihren Bucket hochgeladen hat. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

In diesem Beispiel gehen wir davon aus, dass der Bucket-Eigentümer die vom Bucket-Eigentümer erzwungene Einstellung für Object Ownership nicht angewendet hat. Der Bucket-Eigentümer delegiert die Berechtigung an Benutzer in seinem eigenen Konto. Im Folgenden finden Sie eine Zusammenfassung der exemplarischen Schritte:



1. Der Administrator-Benutzer von Konto A ordnet eine Bucket-Richtlinie mit zwei Anweisungen zu.
   - Kontoübergreifende Berechtigung für Konto B, um Objekte hochzuladen.
   - Berechtigung für einen Benutzer in seinem eigenen Konto, auf Objekte im Bucket zuzugreifen.
2. Der Administrator-Benutzer für Konto B lädt Objekte in den Bucket hoch, der Konto A gehört.
3. Der Administrator für Konto B aktualisiert die Zuweisung ACL zum Hinzufügen von Objekten, die dem Bucket-Besitzer die volle Kontrolle über das Objekt gewährt.
4. Der Benutzer in Konto A überprüft dies durch Zugriff auf Objekte im Bucket, unabhängig davon, wem diese gehören.

Für dieses Beispiel benötigen Sie zwei Konten. Die folgende Tabelle zeigt, wie wir auf diese Konten und die Administrator-Benutzer in diesen Konten verweisen. In dieser exemplarischen Vorgehensweise verwenden Sie die Root-Benutzeranmeldedaten des Kontos nicht gemäß den empfohlenen IAM Richtlinien. Weitere Informationen finden Sie unter Informationen zur Verwendung eines Administratorbenutzers zum Erstellen von Ressourcen und Erteilen von Berechtigungen. Stattdessen erstellen Sie einen Administrator in jedem Konto, und verwenden dessen Anmeldeinformationen, um Ressourcen zu erstellen und ihnen Berechtigungen zu erteilen.

AWS-Konto ID	Konto bezeichnet als	Administrator im Konto
1111-1111-1111	Konto A	AccountAadmin
2222-2222-2222	Konto B	AccountBadmin

Alle Aufgaben in Verbindung mit dem Erstellen von Benutzern und Gewähren von Berechtigungen werden in der AWS Management Console ausgeführt. Um die Berechtigungen zu überprüfen, werden in der exemplarischen Vorgehensweise die Befehlszeilentools AWS Command Line Interface (AWS CLI) und verwendet AWS Tools for Windows PowerShell, sodass Sie keinen Code schreiben müssen.

Schritt 0: Vorbereitung auf den Walkthrough

1. Stellen Sie sicher, dass Sie über zwei Konten verfügen AWS-Konten und dass jedes Konto über einen Administrator verfügt, wie in der Tabelle im vorherigen Abschnitt dargestellt.

   a. Melden Sie sich bei Bedarf für einen an AWS-Konto.

   b. Melden Sie sich mit den Anmeldeinformationen von Konto A bei der IAMKonsole an und gehen Sie wie folgt vor, um einen Administratorbenutzer zu erstellen:

      - Erstellen Sie einen Benutzer **`AccountAadmin`** und notieren Sie sich die Sicherheitsanmeldedaten des Benutzers. Weitere Informationen zum Hinzufügen von Benutzern finden Sie im IAM Benutzerhandbuch unter Erstellen eines IAMBenutzers AWS-Konto in Ihrem.
      - Gewähren Sie Administratorberechtigungen für, AccountAadminindem Sie eine Benutzerrichtlinie anhängen, die vollen Zugriff gewährt. Anweisungen finden Sie im IAMBenutzerhandbuch unter IAMRichtlinien verwalten.
      - Notieren Sie sich im IAMKonsolen-Dashboard die IAMBenutzeranmeldung URL. Benutzer in diesem Konto müssen dies URL bei der Anmeldung beim AWS Management Console verwenden. Weitere Informationen finden Sie im Benutzerhandbuch unter So melden sich IAM Benutzer bei Ihrem Konto an.

   c. Wiederholen Sie den vorherigen Schritt mit den Anmeldeinformationen für Konto B und erstellen Sie einen Administratorbenutzer**`AccountBadmin`**.

2. Richten Sie entweder die Tools AWS CLI oder die Tools für Windows ein PowerShell. Stellen Sie sicher, dass Sie die Administratoranmeldeinformationen wie folgt speichern:

   - Wenn Sie das verwenden AWS CLI, erstellen Sie zwei Profile, `AccountAadmin` und`AccountBadmin`, in der Konfigurationsdatei.
   - Wenn Sie die Tools für Windows verwenden PowerShell, stellen Sie sicher, dass Sie die Anmeldeinformationen für die Sitzung als `AccountAadmin` und speichern`AccountBadmin`.

Detaillierte Anweisungen finden Sie unter Einrichtung der Tools für die exemplarischen Vorgehensweisen.

Schritt 1: Erledigen der Aufgaben von Konto A

Führen Sie für Konto A die folgenden Schritte aus:

Schritt 1.1: Anmelden bei der Konsole

Melden Sie sich mit der IAM Benutzeranmeldung URL für Konto A beim **`AccountAadmin`** Benutzer AWS Management Console als Benutzer an. Dieser Benutzer erstellt einen Bucket und ordnet ihm eine Richtlinie zu.

Schritt 1.2: Erstellen eines Buckets und eines Benutzers und Hinzufügen einer Bucket-Richtlinie, um Benutzerberechtigungen zu erteilen

1. Erstellen Sie in der Amazon-S3-Konsole einen Bucket. In dieser Übung wird davon ausgegangen, dass der Bucket im Osten der USA (Nord-Virginia) AWS-Region erstellt wurde und der Name lautet*amzn-s3-demo-bucket1*.

   Detaillierte Anweisungen finden Sie unter Erstellen eines Buckets.
2. Erstellen Sie in der IAMKonsole einen Benutzer**`Dave`**.

   step-by-stepAnweisungen finden Sie im Benutzerhandbuch unter IAMBenutzer erstellen (Konsole). IAM
3. Notieren Sie sich die Anmeldeinformationen des Benutzers Dave.
4. Weisen Sie in der Amazon-S3-Konsole die folgende Bucket-Richtlinie dem *amzn-s3-demo-bucket1*-Bucket zu. Detaillierte Anweisungen finden Sie unter Hinzufügen einer Bucket-Richtlinie mit der Amazon-S3-Konsole. Folgen Sie den Schritten, um eine Bucket-Richtlinie hinzuzufügen. Informationen dazu, wie Sie ein Konto findenIDs, finden Sie unter Ihre AWS-Konto ID finden.

   Die Richtlinie erteilt Konto B die Berechtigung `s3:PutObject` und `s3:ListBucket`. Die Richtlinie gewährt `Dave` dem Benutzer auch die `s3:GetObject` Erlaubnis.

```
{
   "Version": "2012-10-17",
```

```
   "Statement": [
      {
         "Sid": "Statement1",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountB-ID:root"
         },
         "Action": [
            "s3:PutObject",
            "s3:ListBucket"
         ],
         "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket1/*",
            "arn:aws:s3:::amzn-s3-demo-bucket1"
         ]
      },
      {
         "Sid": "Statement3",
         "Effect": "Allow",
         "Principal": {
            "AWS": "arn:aws:iam::AccountA-ID:user/Dave"
         },
         "Action": [
            "s3:GetObject"
         ],
         "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket1/*"
         ]
      }
   ]
}
```

Schritt 2: Erledigen der Aufgaben von Konto B

Da Konto B nun über die erforderlichen Berechtigungen verfügt, um Vorgänge im Bucket von Konto A durchzuführen, geht der Administrator von Konto B wie folgt vor:

- Lädt ein Objekt in den Bucket von Konto A hoch
- Fügt dem Objekt einen Grant hinzuACL, um Account A, dem Bucket-Besitzer, die volle Kontrolle zu gewähren

Mit dem AWS CLI

1. Laden Sie mit dem `put-object` AWS CLI Befehl ein Objekt hoch. Der Parameter `--body` im Befehl identifiziert die hochzuladende Quelldatei. Wenn sich die Datei beispielsweise auf dem `C:` Laufwerk eines befindet Windows Maschine, spezifizierenc:\HappyFace.jpg. Der Parameter `--key` gibt den Schlüsselnamen für das Objekt an.

```
aws s3api put-object --bucket amzn-s3-demo-bucket1 --key HappyFace.jpg --body
 HappyFace.jpg --profile AccountBadmin
```

2. Fügen Sie dem Objekt einen Grant hinzuACL, um dem Bucket-Besitzer die volle Kontrolle über das Objekt zu gewähren. Informationen darüber, wie Sie eine kanonische Benutzer-ID finden, finden Sie unter Finden Sie die kanonische Benutzer-ID für Sie AWS-Konto im Referenzhandbuch zur AWS Kontoverwaltung.

```
aws s3api put-object-acl --bucket amzn-s3-demo-bucket1 --key HappyFace.jpg --grant-
full-control id="AccountA-CanonicalUserID" --profile AccountBadmin
```

Verwenden der Tools für Windows PowerShell

1. Laden Sie mit dem `Write-S3Object` Befehl ein Objekt hoch.

```
Write-S3Object -BucketName amzn-s3-demo-bucket1 -key HappyFace.jpg -file
 HappyFace.jpg -StoredCredentials AccountBadmin
```

2. Fügen Sie dem Objekt einen Grant hinzuACL, um dem Bucket-Besitzer die volle Kontrolle über das Objekt zu gewähren.

```
Set-S3ACL -BucketName amzn-s3-demo-bucket1 -Key HappyFace.jpg -CannedACLName
 "bucket-owner-full-control" -StoredCreden
```

Schritt 3: Testen der Berechtigungen

Überprüfen Sie nun, dass der Benutzer Dave in Konto A Zugriff auf das Objekt hat, das Konto B gehört.

Mit dem AWS CLI

1. Fügen Sie der AWS CLI Konfigurationsdatei die Anmeldeinformationen des Benutzers Dave hinzu und erstellen Sie ein neues Profil,`UserDaveAccountA`. Weitere Informationen finden Sie unter Einrichtung der Tools für die exemplarischen Vorgehensweisen.

```
[profile UserDaveAccountA]
aws_access_key_id = access-key
aws_secret_access_key = secret-access-key
region = us-east-1
```

2. Führen Sie den `get-object` CLI Befehl aus, um ihn herunterzuladen `HappyFace.jpg` und lokal zu speichern. Sie stellen dem Benutzer Dave Anmeldeinformationen bereit, indem Sie den Parameter `--profile` hinzufügen.

```
aws s3api get-object --bucket amzn-s3-demo-bucket1 --key
 HappyFace.jpg Outputfile.jpg --profile UserDaveAccountA
```

Verwenden der Tools für Windows PowerShell

1. Speichern Sie die AWS Anmeldeinformationen des Benutzers Dave als `UserDaveAccountA` im persistenten Speicher.

```
Set-AWSCredentials -AccessKey UserDave-AccessKey -SecretKey UserDave-
SecretAccessKey -storeas UserDaveAccountA
```

2. Führen Sie den Befehl `Read-S3Object` aus, um das Objekt `HappyFace.jpg` herunterzuladen und es lokal zu speichern. Sie stellen dem Benutzer Dave Anmeldeinformationen bereit, indem Sie den Parameter `-StoredCredentials` hinzufügen.

```
Read-S3Object -BucketName amzn-s3-demo-bucket1 -Key HappyFace.jpg -file
 HappyFace.jpg  -StoredCredentials UserDaveAccountA
```

Schritt 4: Bereinigen

1. Nachdem Sie mit dem Testen fertig sind, können Sie wie folgt aufräumen:

- Melden Sie sich mit den Anmeldeinformationen von Konto A an der AWS Management Console an und machen Sie Folgendes:
  - Entfernen Sie in der Amazon S3 S3-Konsole die Bucket-Richtlinie, die angehängt ist an *amzn-s3-demo-bucket1*. Löschen Sie im Bucket Properties die Richtlinie im Abschnitt Permissions.
  - Wenn der Bucket für diese Übung erstellt wurde, löschen Sie in der Amazon-S3-Konsole die Objekte und dann den Bucket.
  - Entfernen Sie den AccountAadminBenutzer in der IAMKonsole. step-by-step Anweisungen finden Sie im IAMBenutzerhandbuch unter Löschen eines IAM Benutzers.

2. Melden Sie sich mit den Anmeldeinformationen von Konto B an der AWS Management Console an. Löschen Sie den Benutzer in der IAMKonsole AccountBadmin.

Beispiel 4: Bucket-Besitzer gewährt kontoübergreifende Berechtigungen für Objekte, die er nicht besitzt

Themen

- Grundlegendes zu kontoübergreifenden Berechtigungen und zur Verwendung von Rollen IAM
- Schritt 0: Vorbereitung auf den Walkthrough
- Schritt 1: Erledigen der Aufgaben von Konto A
- Schritt 2: Erledigen der Aufgaben von Konto B
- Schritt 3: Führen Sie die Aufgaben von Konto C aus
- Schritt 4: Bereinigen
- Zugehörige Ressourcen

In diesem Beispielszenario besitzen Sie einen Bucket und haben anderen AWS-Konten das Hochladen von Objekten ermöglicht. Wenn Sie die vom Bucket-Eigentümer erzwungene Einstellung für S3 Object Ownership für den Bucket angewendet haben, besitzen Sie alle Objekte im Bucket, einschließlich der Objekte, die von einem anderen AWS-Konto geschrieben wurden. Dieser Ansatz behebt das Problem, dass Objekte nicht Ihnen, dem Bucket-Besitzer, gehören. Anschließend können Sie die Berechtigung an Benutzer in Ihrem eigenen Konto oder an andere AWS-Konten. Angenommen, die erzwungene Einstellung des Bucket-Eigentümers für S3 Object Ownership ist nicht aktiviert. Das bedeutet, Ihr Bucket kann Objekte enthalten, die anderen AWS-Konten gehören.

Angenommen, Sie müssen als Bucket-Eigentümer einem Benutzer in einem anderen Konto eine kontenübergreifende Berechtigung für Objekte erteilen, unabhängig davon, wer der Eigentümer ist. Dieser Benutzer könnte beispielsweise eine Buchhaltungsanwendung sein, die Zugriff auf Objekt-Metadaten benötigt. Es gibt zwei Kernprobleme:

- Der Bucket-Eigentümer hat keine Berechtigungen für diese Objekte, die von anderen AWS-Konten erstellt wurden. Damit der Bucket-Besitzer Berechtigungen für Objekte gewähren kann, die ihm nicht gehören, muss der Objekteigentümer zuerst dem Bucket-Besitzer die entsprechenden Rechte erteilen. Der Objekteigentümer ist derjenige AWS-Konto , der die Objekte erstellt hat. Der Bucket-Eigentümer kann diese Berechtigungen delegieren.
- Das Konto des Bucket-Besitzers kann Berechtigungen an Benutzer in seinem eigenen Konto delegieren (sieheBeispiel 3: Bucket-Eigentümer, der Berechtigungen für Objekte erteilt, die ihm nicht gehören). Das Konto des Bucket-Besitzers kann jedoch keine Berechtigungen an andere delegieren, AWS-Konten da die kontoübergreifende Delegierung nicht unterstützt wird.

In diesem Szenario kann der Bucket-Besitzer eine Rolle AWS Identity and Access Management (IAM) mit der Berechtigung für den Zugriff auf Objekte erstellen. Anschließend kann der Bucket-Besitzer eine weitere AWS-Konto Berechtigung zur Übernahme der Rolle gewähren, sodass er vorübergehend auf Objekte im Bucket zugreifen kann.

> **Note**
>
> S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie sowohl das Eigentum an den Objekten kontrollieren können, die in Ihren Bucket hochgeladen werden, als auch um sie zu deaktivieren oder zu aktivieren. ACLs Standardmäßig ist für Object Ownership die erzwungene Einstellung des Bucket-Besitzers festgelegt, und alle sind ACLs deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff darauf ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.
> Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Wenn diese ACLs Option deaktiviert ist, können Sie mithilfe von Richtlinien den Zugriff auf alle Objekte in Ihrem Bucket steuern, unabhängig davon, wer die Objekte in Ihren Bucket hochgeladen hat. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

Grundlegendes zu kontoübergreifenden Berechtigungen und zur Verwendung von Rollen IAM

IAMRollen ermöglichen verschiedene Szenarien, um den Zugriff auf Ihre Ressourcen zu delegieren, und der kontenübergreifende Zugriff ist eines der wichtigsten Szenarien. In diesem Beispiel verwendet der Bucket-Besitzer, Konto A, eine IAM Rolle, um vorübergehend den kontoübergreifenden Objektzugriff an Benutzer in einem anderen Konto AWS-Konto, Konto C, zu delegieren. Jeder IAM Rolle, die Sie erstellen, sind die folgenden zwei Richtlinien zugeordnet:

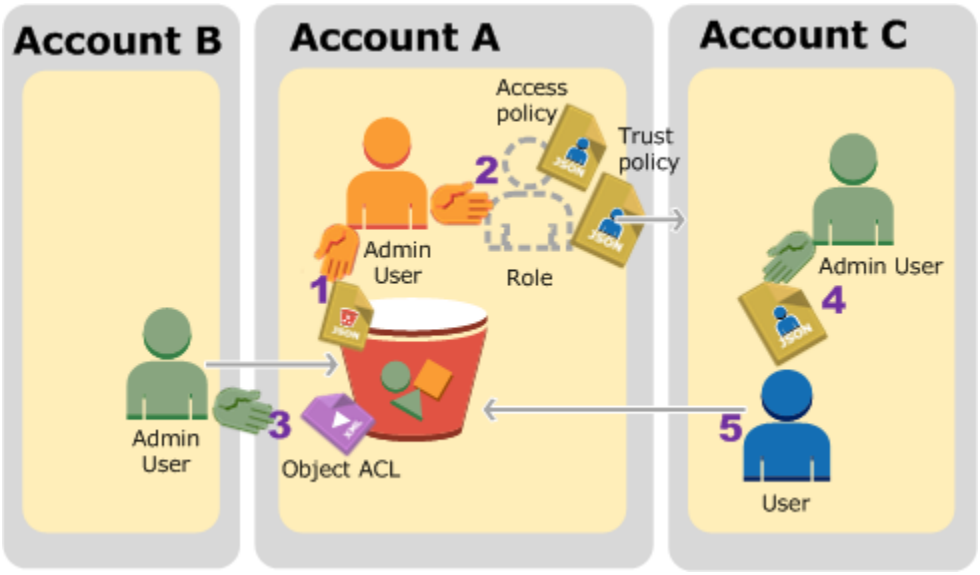
- Eine Vertrauensrichtlinie, die eine andere Person identifiziert AWS-Konto , die die Rolle übernehmen kann.
- Eine Zugriffsrichtlinie, die definiert, welche Berechtigungen – z. B. `s3:GetObject` – zulässig sind, wenn jemand die Rolle einnimmt. Eine Liste aller Berechtigungen, die Sie in einer Richtlinie angeben können, finden Sie unter Politische Maßnahmen für Amazon S3.

Die AWS-Konto in der Vertrauensrichtlinie angegebene Person erteilt ihrem Benutzer dann die Erlaubnis, die Rolle zu übernehmen. Der Benutzer kann dann wie folgt auf Objekte zugreifen:

- Die Rolle einnehmen und daraufhin temporäre Sicherheitsanmeldeinformationen erhalten.
- Verwenden der temporären Sicherheitsanmeldeinformationen, um auf die Objekte im Bucket zuzugreifen.

Weitere Informationen zu IAM IAMRollen finden Sie im IAMBenutzerhandbuch unter Rollen.

Im Folgenden finden Sie eine Zusammenfassung der exemplarischen Schritte:



1. Der Administrator-Benutzer von Konto A ordnet eine Bucket-Richtlinie zu, die Konto B die bedingte Berechtigung erteilt, Objekte hochzuladen.
2. Der Administrator von Konto A erstellt eine IAM Rolle und stellt so eine Vertrauensstellung mit Konto C her, sodass Benutzer in diesem Konto auf Konto A zugreifen können. Die der Rolle zugeordnete Zugriffsrichtlinie schränkt ein, was der Benutzer in Konto C tun kann, wenn der Benutzer auf Konto A zugreift.
3. Der Administrator von Konto B lädt ein Objekt in den Bucket hoch, der Konto A gehört, und erteilt dem Bucket-Eigentümer vollständige Berechtigungen.
4. Der Administrator von Konto C erstellt einen Benutzer und ordnet ihm eine Benutzerrichtlinie zu, die dem Benutzer gestattet, die Rolle zu übernehmen.
5. Der Benutzer in Konto C übernimmt zuerst die Rolle, womit er temporäre Sicherheitsanmeldeinformationen erhält. Unter Verwendung dieser temporären Sicherheitsanmeldeinformationen greift der Benutzer dann auf die Objekte im Bucket zu.

Für dieses Beispiel benötigen Sie drei Konten. Die folgende Tabelle zeigt, wie wir auf diese Konten und die Administrator-Benutzer in diesen Konten verweisen. Gemäß den IAM Richtlinien (siehe Informationen zur Verwendung eines Administratorbenutzers zum Erstellen von Ressourcen und Erteilen von Berechtigungen) verwenden wir die Root-Benutzer des AWS-Kontos Anmeldeinformationen in dieser exemplarischen Vorgehensweise nicht. Stattdessen erstellen Sie einen Administrator-Benutzer in jedem Konto, und verwenden dessen Anmeldeinformationen, um Ressourcen zu erstellen und ihnen Berechtigungen zu erteilen.

AWS-Konto Ausweis	Konto bezeichnet als	Administratorbenutzer im Konto
1111-1111-1111	Konto A	AccountAadmin
2222-2222-2222	Konto B	AccountBadmin
3333-3333-3333	Konto C	AccountCadmin

Schritt 0: Vorbereitung auf den Walkthrough

> **Note**
>
> Möglicherweise möchten Sie einen Texteditor öffnen und einige Informationen aufschreiben, während Sie die Schritte ausführen. Insbesondere benötigen Sie ein KontoIDs, einen kanonischen BenutzerIDs, eine IAM Benutzeranmeldung URLs für jedes Konto, um eine Verbindung zur Konsole herzustellen, sowie Amazon-Ressourcennamen (ARNs) der IAM Benutzer und Rollen.

1. Stellen Sie sicher, dass Sie drei Benutzer haben AWS-Konten und dass jedes Konto über einen Administratorbenutzer verfügt, wie in der Tabelle im vorherigen Abschnitt dargestellt.

   a. Melden Sie sich bei Bedarf für AWS-Konten an. Wir bezeichnen diese Konten als Konto A, Konto B und Konto C.

   b. Melden Sie sich mit den Anmeldeinformationen von Konto A bei der IAMKonsole an und gehen Sie wie folgt vor, um einen Administratorbenutzer zu erstellen:

      - Erstellen Sie einen Benutzer **`AccountAadmin`** und notieren Sie sich seine Sicherheitsanmeldedaten. Weitere Informationen zum Hinzufügen von Benutzern finden Sie unter Erstellen eines IAM Benutzers AWS-Konto in Ihrem IAM Benutzerhandbuch.
      - Gewähren Sie Administratorrechte, AccountAadminindem Sie eine Benutzerrichtlinie anhängen, die vollen Zugriff gewährt. Anweisungen finden Sie im IAMBenutzerhandbuch unter IAMRichtlinien verwalten.
      - Notieren Sie sich im IAM Konsolen-Dashboard die IAMBenutzeranmeldung URL. Benutzer in diesem Konto müssen dies URL bei der Anmeldung bei der AWS Management Console verwenden. Weitere Informationen finden Sie unter AWS Management Console Als IAM Benutzer anmelden im IAMBenutzerhandbuch.

   c. Wiederholen Sie den obigen Schritt, um Administrator-Benutzer in Konto B und Konto C zu erstellen.

2. Notieren Sie sich für Konto C die kanonische Benutzer-ID.

   Wenn Sie eine IAM Rolle in Konto A erstellen, erteilt die Vertrauensrichtlinie Konto C die Erlaubnis, die Rolle anzunehmen, indem die Konto-ID angegeben wird. Sie finden die Konteninformationen wie folgt:

a. Verwenden Sie Ihre AWS-Konto ID oder Ihren Kontoalias, Ihren IAM Benutzernamen und Ihr Passwort, um sich bei der Amazon S3 S3-Konsole anzumelden.

b. Wählen Sie den Namen eines Amazon-S3-Buckets aus, um die Details zu diesem Bucket anzuzeigen.

c. Klicken Sie auf den Tab Berechtigungen und anschließend auf Access Control List.

d. Im Abschnitt Access for your AWS-Konto(Zugriff für Ihr AWS-Konto) in der Spalte Konto befindet sich eine lange Kennung, z. B. `c1daexampleaaf850ea79cf0430f33d72579fd1611c97f7ded193374c0b163b6`. Dies ist Ihre kanonische Benutzer-ID.

3. Für das Erstellen einer Bucket-Richtlinie benötigen Sie die folgenden Informationen. Notieren Sie sich diese Werte:

   - Kanonische Benutzer-ID von Konto A – Wenn der Administrator von Konto A dem Administrator von Konto B die bedingte Berechtigung erteilt, Objekte hochzuladen, gibt die Bedingung die kanonische Benutzer-ID des Benutzers von Konto A an, der vollständige Kontrolle über die Objekte benötigt.

     > **Note**
     >
     > Die kanonische Benutzer-ID ist ein nur für Amazon S3 geltendes Konzept. Es handelt sich dabei um eine 64 Zeichen lange verschleierte Version der Konto-ID.

   - Benutzer ARN für Konto B-Administrator — Sie finden den Benutzer ARN in der IAMKonsole. Sie müssen den Benutzer auswählen und den Benutzer auf der Registerkarte „Übersicht“ suchen. ARN

     In der Bucket-Richtlinie erteilen Sie die `AccountBadmin` Berechtigung zum Hochladen von Objekten und geben den Benutzer an, der den ARN verwendet. Hier ist ein ARN Beispielwert:

     ```
     arn:aws:iam::AccountB-ID:user/AccountBadmin
     ```

4. Richten Sie entweder das AWS Command Line Interface (CLI) oder das ein AWS Tools for Windows PowerShell. Stellen Sie sicher, dass Sie die Administrator-Benutzeranmeldedaten wie folgt speichern:

   - Wenn Sie das verwenden AWS CLI, erstellen Sie die Profile `AccountAadmin` und `AccountBadmin` in der Konfigurationsdatei.

- Wenn Sie die verwenden AWS Tools for Windows PowerShell, stellen Sie sicher, dass Sie die Anmeldeinformationen für die Sitzung als `AccountAadmin` und speichern`AccountBadmin`.

Detaillierte Anweisungen finden Sie unter Einrichtung der Tools für die exemplarischen Vorgehensweisen.

Schritt 1: Erledigen der Aufgaben von Konto A

In diesem Beispiel ist Konto A der Bucket-Eigentümer. Der Benutzer AccountAadmin in Konto A wird also Folgendes tun:

- Erstellen Sie einen Bucket.
- Hängen Sie eine Bucket-Richtlinie an, die dem Konto B Administratorrechte zum Hochladen von Objekten gewährt.
- Erstellen Sie eine IAM Rolle, die Konto C die Berechtigung erteilt, die Rolle zu übernehmen, sodass es auf Objekte im Bucket zugreifen kann.

Schritt 1.1: Melden Sie sich an bei AWS Management Console

Melden Sie sich mit der IAM Benutzeranmeldung URL für Konto A zunächst AWS Management Console als **`AccountAadmin`** Benutzer an. Dieser Benutzer erstellt einen Bucket und ordnet ihm eine Richtlinie zu.

Schritt 1.2: Erstellen eines Buckets und Anfügen einer Richtlinie

Führen Sie in der Amazon-S3-Konsole die folgenden Schritte aus:

1. Erstellen Sie einen Bucket. Diese Übung setzt voraus, dass der Bucket-Name *`amzn-s3-demo-bucket1`* ist.

   Detaillierte Anweisungen finden Sie unter Erstellen eines Buckets.

2. Fügen Sie die folgende Bucket-Richtlinie bei. Die Richtlinie gewährt dem Administrator für Konto B bedingte Berechtigungen zum Hochladen von Objekten.

   Aktualisieren Sie die Richtlinie, indem Sie Ihre eigenen Werte für *`amzn-s3-demo-bucket1AccountB-ID`*, und die angeben*`CanonicalUserId-of-AWSaccountA-BucketOwner`*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "111",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*"
        },
        {
            "Sid": "112",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
            "Condition": {
                "StringNotEquals": {
                    "s3:x-amz-grant-full-control": "id=CanonicalUserId-of-
AWSaccountA-BucketOwner"
                }
            }
        }
    ]
}
```

Schritt 1.3: Erstellen Sie eine IAM Rolle, um Konto C den kontoübergreifenden Zugriff auf Konto A zu ermöglichen

Erstellen Sie in der IAMKonsole eine IAM Rolle (**`examplerole`**), die Konto C die Erlaubnis erteilt, die Rolle zu übernehmen. Stellen Sie sicher, dass Sie immer noch als Administrator für Konto A angemeldet sind, da die Rolle in Konto A erstellt werden muss.

1. Bevor Sie die Rolle erstellen, richten Sie die verwaltete Richtlinie ein, die die von der Rolle benötigten Berechtigungen definiert. Diese Richtlinie fügen Sie zu einem späteren Zeitpunkt der Rolle an.

a. Wählen Sie im Navigationsbereich auf der linken Seite Richtlinien und dann Richtlinie erstellen aus.

b. Klicken Sie neben Create Your Own Policy auf Select.

c. Geben Sie **`access-accountA-bucket`** in das Feld Policy Name ein.

d. Kopieren Sie die folgende Zugriffsrichtlinie und fügen Sie sie in das Feld Policy Document ein. Die Zugriffsrichtlinie gewährt der Rolle die `s3:GetObject` Berechtigung, sodass der Benutzer des Kontos C, wenn er die Rolle annimmt, nur den `s3:GetObject` Vorgang ausführen kann.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*"
    }
  ]
}
```

e. Wählen Sie Create Policy (Richtlinie erstellen) aus.

Die neue Richtlinie wird in der Liste der verwalteten Richtlinien angezeigt.

2. Wählen Sie im Navigationsbereich auf der linken Seite Rollen und dann Neue Rolle erstellen aus.

3. Wählen Sie unter Rollentyp auswählen die Option Rolle für kontoübergreifenden Zugriff aus und klicken AWS-Konten Sie dann auf die Schaltfläche Auswählen neben Eigenzugriff bereitstellen.

4. Geben Sie die Konto-ID von Konto C ein.

Für diese exemplarische Vorgehensweise müssen Sie nicht verlangen, dass Benutzer über eine Multi-Faktor-Authentifizierung (MFA) verfügen, um die Rolle übernehmen zu können. Wählen Sie diese Option also nicht aus.

5. Klicken Sie auf Next Step, um die mit der Rolle verknüpften Berechtigungen einzurichten.

6. Aktivieren Sie das Kontrollkästchen neben der Access-Account-A-Bucket-Richtlinie, die Sie erstellt haben, und wählen Sie dann Next Step aus.

Die Prüfseite wird angezeigt, sodass Sie die Einstellungen bestätigen können, bevor Sie die Rolle erstellen. Ein sehr wichtiges, auf dieser Seite zu beachtendes Element ist der Link, den Sie an die Benutzer senden können, die die Rolle verwenden müssen. Benutzer, die den Link verwenden, gelangen direkt zur Seite „Rolle wechseln“, auf der die Felder Konto-ID und Rollenname bereits ausgefüllt sind. Dieser Link wird auch auf der Seite Role Summary für beliebige kontoübergreifende Rollen angezeigt.

7. Geben Sie `examplerole` den Rollennamen ein und wählen Sie dann Next Step.
8. Nachdem Sie die Rolle überprüft haben, wählen Sie Create Role aus.

   Die Rolle `examplerole` wird in der Liste der Rollen angezeigt.
9. Wählen Sie den Rollennamen`examplerole`.
10. Wählen Sie den Tab Trust Relationships.
11. Wählen Sie Richtliniendokument anzeigen und überprüfen Sie, ob die angezeigte Vertrauensrichtlinie mit der folgenden Richtlinie übereinstimmt.

    Die folgende Vertrauensrichtlinie richtet eine Vertrauensbeziehung zu Konto C ein und gestattet ihm die Aktion `sts:AssumeRole`. Weitere Informationen finden Sie unter AssumeRolein der AWS Security Token Service APIReferenz.

    ```
    {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Sid": "",
          "Effect": "Allow",
          "Principal": {
            "AWS": "arn:aws:iam::AccountC-ID:root"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }
    ```
12. Notieren Sie sich den Amazon-Ressourcennamen (ARN) der `examplerole` Rolle, die Sie erstellt haben.

Später in den folgenden Schritten fügen Sie eine Benutzerrichtlinie hinzu, die es einem IAM Benutzer ermöglicht, diese Rolle anzunehmen, und Sie identifizieren die Rolle anhand des ARN Werts.

Schritt 2: Erledigen der Aufgaben von Konto B

Der Beispiel-Bucket, der Konto A gehört, benötigt Objekte, die anderen Konten gehören. In diesem Schritt lädt der Administrator von Konto B unter Verwendung der Befehlszeilen-Tools ein Objekt hoch.

- Laden Sie mithilfe des `put-object` AWS CLI Befehls ein Objekt in hoch*amzn-s3-demo-bucket1*.

  ```
  aws s3api put-object --bucket amzn-s3-demo-bucket1 --key HappyFace.jpg --
  body HappyFace.jpg --grant-full-control id="canonicalUserId-ofTheBucket0wner" --
  profile AccountBadmin
  ```

  Beachten Sie Folgendes:

  - Der `--Profile` Parameter gibt das `AccountBadmin` Profil an, sodass das Objekt dem Konto B gehört.
  - Der Parameter `grant-full-control` erteilt dem Bucket-Eigentümer vollständige Berechtigungen für das Objekt, wie für die Bucket-Richtlinie erforderlich.
  - Der Parameter `--body` identifiziert die hochzuladende Quelldatei. Wenn sich die Datei beispielsweise auf dem Laufwerk C: eines Windows Computer, den Sie angeben`c:\HappyFace.jpg`.

Schritt 3: Führen Sie die Aufgaben von Konto C aus

In den vorherigen Schritten hat Konto A bereits eine Rolle erstellt`examplerole`, die Vertrauen zu Konto C herstellt. Diese Rolle ermöglicht es Benutzern in Konto C, auf Konto A zuzugreifen. In diesem Schritt erstellt der Administrator von Konto C einen Benutzer (Dave) und delegiert ihm die von Konto A erhaltenen `sts:AssumeRole` Berechtigungen. Dieser Ansatz ermöglicht es Dave, das Konto A zu übernehmen `examplerole` und vorübergehend Zugriff auf Konto A zu erhalten. Die Zugriffsrichtlinie, die Konto A der Rolle beigefügt hat, schränkt ein, was Dave tun kann, wenn er auf Konto A zugreift, insbesondere Objekte in*amzn-s3-demo-bucket1*.

Schritt 3.1: Erstellen Sie einen Benutzer in Konto C und delegieren Sie die Zugriffsrechte examplerole

1. Melden Sie sich mit der IAM Benutzeranmeldung URL für Konto C zunächst AWS Management Console als **`AccountCadmin`** Benutzer an.

2. Erstellen Sie in der IAMKonsole einen Benutzer, Dave.

   step-by-stepAnweisungen finden Sie im Benutzerhandbuch unter IAMBenutzer erstellen (AWS Management Console). IAM

3. Notieren Sie sich die Dave-Anmeldeinformationen. Dave benötigt diese Anmeldeinformationen, um die Rolle `examplerole` zu übernehmen.

4. Erstellen Sie eine Inline-Richtlinie für den IAM Dave-Benutzer, um Dave die `sts:AssumeRole` Berechtigung für die `examplerole` Rolle in Konto A zu delegieren.

   a. Wählen Sie im Navigationsbereich auf der linken Seite Users (Benutzer).

   b. Wählen Sie den Benutzernamen Dave.

   c. Wählen Sie auf der Seite mit den Benutzerinformationen den Tab Permissions aus und erweitern Sie den Abschnitt Inline Policies.

   d. Wählen Sie hier klicken (oder Create User Policy).

   e. Wählen Sie Custom Policy und dann Select aus.

   f. Geben Sie einen Namen für die Richtlinie in das Feld Policy Name ein.

   g. Kopieren Sie die folgende Richtlinie in das Feld Policy Document:.

      Sie müssen die Richtlinie aktualisieren, indem Sie die angeben*`AccountA-ID`*.

      ```
      {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Action": ["sts:AssumeRole"],
            "Resource": "arn:aws:iam::AccountA-ID:role/examplerole"
          }
        ]
      }
      ```

   h. Klicken Sie auf Apply Policy (Richtlinie anwenden).

5. Speichern Sie Daves Anmeldeinformationen in der Konfigurationsdatei von, AWS CLI indem Sie ein weiteres Profil hinzufügen,`AccountCDave`.

```
[profile AccountCDave]
aws_access_key_id = UserDaveAccessKeyID
aws_secret_access_key = UserDaveSecretAccessKey
region = us-west-2
```

Schritt 3.2: Rolle übernehmen (examplerole) und auf Objekte zugreifen

Jetzt kann Dave auf Objekte in dem Bucket zugreifen, der Konto A gehört, nämlich wie folgt:

- Dave übernimmt zuerst die `examplerole` unter Verwendung seiner eigenen Anmeldeinformationen. Damit werden temporäre Anmeldeinformationen zurückgegeben.
- Unter Verwendung der temporären Anmeldeinformationen greift Dave dann auf die Objekte im Bucket von Konto A zu.

1. Führen Sie in der Befehlszeile den folgenden AWS CLI `assume-role` Befehl unter Verwendung des `AccountCDave` Profils aus.

   Sie müssen den ARN Wert im Befehl aktualisieren, indem Sie angeben*AccountA-ID*, wo definiert `examplerole` ist.

```
aws sts assume-role --role-arn arn:aws:iam::AccountA-ID:role/examplerole --profile
 AccountCDave --role-session-name test
```

   Als Antwort gibt AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldedaten zurück (Zugriffsschlüssel-ID, geheimer Zugriffsschlüssel und ein Sitzungstoken).

2. Speichern Sie die temporären Sicherheitsanmeldedaten in der AWS CLI Konfigurationsdatei unter dem `TempCred` Profil.

```
[profile TempCred]
aws_access_key_id = temp-access-key-ID
aws_secret_access_key = temp-secret-access-key
aws_session_token = session-token
region = us-west-2
```

3. Führen Sie in der Befehlszeile den folgenden AWS CLI Befehl aus, um mithilfe der temporären Anmeldeinformationen auf Objekte zuzugreifen. Der Befehl gibt beispielsweise das Head-Objekt an, API um Objektmetadaten für das `HappyFace.jpg` Objekt abzurufen.

```
aws s3api get-object --bucket amzn-s3-demo-bucket1 --
key HappyFace.jpg SaveFileAs.jpg --profile TempCred
```

Die `examplerole` zugeordnete Zugriffsrichtlinie erlaubt die Aktionen, deshalb verarbeitet Amazon S3 die Anforderung. Sie können das mit jeder anderen Aktion für jedes andere Objekt im Bucket ausprobieren.

Wenn Sie beispielsweise eine andere Aktion ausprobieren, wird `get-object-acl` Ihnen die Berechtigung verweigert, weil die Rolle diese Aktion nicht erlaubt.

```
aws s3api get-object-acl --bucket amzn-s3-demo-bucket1 --key HappyFace.jpg --
profile TempCred
```

Wir haben den Benutzer Dave verwendet, um die Rolle zu übernehmen und unter Verwendung temporärer Anmeldeinformationen auf das Objekt zuzugreifen. Es könnte auch eine Anwendung in Konto C sein, die auf Objekte in *amzn-s3-demo-bucket1* zugreift. Die Anwendung kann temporäre Anmeldeinformationen erhalten, und Konto C kann die Berechtigung der Anwendung delegieren, um `examplerole` zu übernehmen.

Schritt 4: Bereinigen

1. Wenn Sie mit dem Testen fertig sind, können Sie wie folgt aufräumen:

   - Melden Sie sich mit den Anmeldeinformationen von Konto A an der AWS Management Console an und machen Sie Folgendes:

     - Entfernen Sie in der Amazon-S3-Konsole die an *amzn-s3-demo-bucket1* angefügte Bucket-Richtlinie. Löschen Sie in den Bucket Properties die Richtlinie im Abschnitt Permissions.
     - Wenn der Bucket für diese Übung erstellt wurde, löschen Sie in der Amazon-S3-Konsole die Objekte und dann den Bucket.

- Entfernen Sie in der IAMKonsole die Datei, die `examplerole` Sie in Konto A erstellt haben. Eine step-by-step Anleitung dazu finden Sie im IAM Benutzerhandbuch unter Löschen eines IAM Benutzers.
- Entfernen Sie den AccountAadminBenutzer in der IAMKonsole.

2. Melden Sie sich mit den Anmeldeinformationen für Konto B bei der IAMKonsole an. Löschen Sie den Benutzer AccountBadmin.
3. Melden Sie sich mit den Anmeldeinformationen von Konto C bei der IAMKonsole an. Delete AccountCadminund der Benutzer Dave.

Zugehörige Ressourcen

Weitere Informationen zu dieser exemplarischen Vorgehensweise finden Sie in den folgenden Ressourcen im IAMBenutzerhandbuch:

- Eine Rolle erstellen, um Berechtigungen an einen Benutzer zu delegieren IAM
- Tutorial: Zugriff mithilfe von Rollen übergreifend AWS-Konten delegieren IAM
- Richtlinien verwalten IAM

Verwenden von serviceverknüpften Rollen für Amazon S3 Storage Lens

Um Amazon S3 Storage Lens zum Sammeln und Aggregieren von Metriken für alle Ihre Konten in AWS Organizations-Organisationen zu verwenden, müssen Sie zunächst sicherstellen, dass S3-Storage-Lens über vertrauenswürdigen Zugriff verfügt, der durch das Verwaltungskonto in Ihrer Organisation aktiviert ist. S3 Storage Lens erstellt eine dienstbezogene Rolle (SLR), damit es die Liste der AWS-Konten Mitglieder Ihrer Organisation abrufen kann. Diese Liste von Konten wird von S3 Storage Lens verwendet, um Metriken für S3-Ressourcen in allen Mitgliedskonten zu sammeln, wenn das S3-Storage-Lens-Dashboard oder die Konfigurationen erstellt oder aktualisiert werden.

Amazon S3 Storage Lens verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist ein einzigartiger IAM Rollentyp, der direkt mit S3 Storage Lens verknüpft ist. Dienstbezogene Rollen sind von S3 Storage Lens vordefiniert und beinhalten alle Berechtigungen, die der Dienst benötigt, um andere in AWS-Services Ihrem Namen anzurufen.

Eine serviceverknüpfte Rolle vereinfacht das Einrichten von S3 Storage Lens, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. S3 Storage Lens definiert die Berechtigungen seiner serviceverknüpften Rollen. Sofern keine andere Konfiguration festgelegt

wurde, kann nur S3 Storage Lens die Rollen übernehmen. Zu den definierten Berechtigungen gehören die Vertrauensrichtlinie und die Berechtigungsrichtlinie, und diese Berechtigungsrichtlinie kann keiner anderen IAM Entität zugeordnet werden.

Sie können diese serviceverknüpfte Rolle erst löschen, nachdem die zugehörigen Ressourcen gelöscht wurden. Dies schützt Ihre S3-Storage-Lens-Ressourcen, da Sie nicht versehentlich die Berechtigung für den Zugriff auf die Ressourcen entfernen können.

Informationen zu anderen Diensten, die dienstverknüpfte Rollen unterstützen, finden Sie unter AWS Dienste, die mit Dienstverknüpften Rollen funktionieren, IAM und suchen Sie nach Diensten, für die in der Spalte Dienstverknüpfte Rolle die Option Ja steht. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Berechtigungen für serviceverknüpfte Rollen für Amazon S3 Storage Lens

S3 Storage Lens verwendet die dienstgebundene Rolle mit dem Namen AWSServiceRoleForS3StorageLens— Diese ermöglicht den Zugriff auf AWS Dienste und Ressourcen, die von S3 Storage Lens verwendet oder verwaltet werden. Dadurch kann S3 Storage Lens in Ihrem Namen auf AWS Organizations Ressourcen zugreifen.

Die serviceverknüpfte Rolle S3 Storage Lens vertraut dem folgenden Service auf dem Speicher Ihres Unternehmens:

- `storage-lens.s3.amazonaws.com`

Die Rollenberechtigungsrichtlinie erlaubt S3 Storage Lens die Durchführung der folgenden Aktionen:

- `organizations:DescribeOrganization`

  `organizations:ListAccounts`

  `organizations:ListAWSServiceAccessForOrganization`

  `organizations:ListDelegatedAdministrators`

Sie müssen Berechtigungen konfigurieren, damit eine IAM Entität (z. B. ein Benutzer, eine Gruppe oder eine Rolle) eine dienstbezogene Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter Berechtigungen für dienstbezogene Rollen im IAMBenutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für S3 Storage Lens

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie eine der folgenden Aufgaben ausführen, während Sie beim AWS Organizations Management- oder Delegate-Administratorkonto angemeldet sind, erstellt S3 Storage Lens die dienstbezogene Rolle für Sie:

- Erstellen Sie in der Amazon-S3-Konsole eine S3-Storage-Lens-Dashboard-Konfiguration für Ihr Unternehmen.
- PUTeine S3 Storage Lens-Konfiguration für Ihre Organisation mithilfe von RESTAPI, und AWS CLI . SDKs

> **Note**
>
> S3 Storage Lens unterstützt maximal fünf delegierte Administratoren pro Organisation.

Wenn Sie diese serviceverknüpfte Rolle löschen, werden sie von den vorherigen Aktionen bei Bedarf neu erstellt.

Beispielrichtlinie für die serviceverknüpfte Rolle S3 Storage Lens

Example Berechtigungsrichtlinie für die serviceverknüpfte Rolle S3 Storage Lens

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AwsOrgsAccess",
            "Effect": "Allow",
            "Action": [
                "organizations:DescribeOrganization",
                "organizations:ListAccounts",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:ListDelegatedAdministrators"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
```

```
}
```

Bearbeiten einer serviceverknüpften Rolle für Amazon S3 Storage Lens

Mit S3 Storage Lens können Sie das nicht bearbeiten AWSServiceRoleForS3StorageLens Rolle, die mit dem Dienst verknüpft ist. Nachdem Sie eine serviceverknüpfte Rolle erstellt haben, können Sie den Namen der Rolle nicht mehr ändern, da verschiedene Entitäten auf die Rolle verweisen könnten. Sie können die Beschreibung der Rolle jedoch mit IAM bearbeiten. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Bearbeiten einer dienstbezogenen Rolle.

Löschen einer serviceverknüpften Rolle für Amazon S3 Storage Lens

Wenn Sie die serviceverknüpfte Rolle nicht mehr verwenden müssen, empfehlen wir, die Rolle zu 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 Amazon-S3-Storage-Lens-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um das zu löschen AWSServiceRoleForS3StorageLens Sie müssen alle S3 Storage Lens-Konfigurationen auf Organisationsebene löschen, die in allen vorhanden sind, AWS-Regionen indem Sie das AWS Organizations Management- oder das Delegate-Administratorkonto verwenden.

Die Ressourcen sind S3-Storage-Lens-Konfigurationen auf Organisationsebene. Verwenden Sie S3 Storage Lens, um die Ressourcen zu bereinigen, und verwenden Sie dann die IAMKonsole,CLI, oder RESTAPI, AWS SDK um die Rolle zu löschen.

In den Bereichen REST API AWS CLI SDKs, und können S3 Storage Lens-Konfigurationen anhand aller Regionen ermittelt werden, `ListStorageLensConfigurations` in denen Ihr Unternehmen S3 Storage Lens-Konfigurationen erstellt hat. Verwenden Sie die Aktion `DeleteStorageLensConfiguration`, um diese Konfigurationen zu löschen, damit Sie die Rolle dann löschen können.

> **Note**
>
> Um die serviceverknüpfte Rolle zu löschen, müssen Sie alle S3-Storage-Lens-Konfigurationen auf Organisationsebene in allen Regionen löschen, in denen sie existieren.

Um Amazon S3 Storage Lens-Ressourcen zu löschen, die verwendet werden von AWSServiceRoleForS3StorageLens SLR

1. Um eine Liste Ihrer Konfigurationen auf Organisationsebene zu erhalten, müssen Sie `ListStorageLensConfigurations` in jeder Region, in der Sie über S3 Storage Lens-Konfigurationen verfügen, die S3 Storage Lens verwenden. Diese Liste kann auch über die Amazon S3 S3-Konsole abgerufen werden.
2. Löschen Sie diese Konfigurationen von den entsprechenden regionalen Endpunkten, indem `DeleteStorageLensConfiguration` API Sie den Anruf aufrufen oder die Amazon S3 S3-Konsole verwenden.

Um die mit dem Service verknüpfte Rolle manuell zu löschen, verwenden Sie IAM

Nachdem Sie die Konfigurationen gelöscht haben, löschen Sie die AWSServiceRoleForS3StorageLens SLRvon der IAMKonsole aus oder durch IAM API `DeleteServiceLinkedRole` Aufrufen von oder mithilfe von AWS CLI oder AWS SDK. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Löschen einer serviceverknüpften Rolle.

Unterstützte Regionen für serviceverknüpfte Rollen mit S3 Storage Lens

S3 Storage Lens unterstützt die Verwendung von dienstbezogenen Rollen überall dort, AWS-Regionen wo der Dienst verfügbar ist. Weitere Informationen finden Sie unter Amazon-S3-Regionen und Endpunkte.

Problembehandlung Amazon S3 S3-Identität und -Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit Amazon S3 und auftreten könnenIAM.

Themen

- Ich habe die Fehlermeldung „Zugriff verweigert“ erhalten
- Ich bin nicht berechtigt, eine Aktion in Amazon S3 durchzuführen

- Ich bin nicht berechtigt, iam auszuführen: PassRole
- Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine Amazon S3 S3-Ressourcen ermöglichen
- Problembehandlung bei verweigertem Zugriff beheben (403 Forbidden) Fehler in Amazon S3

Ich habe die Fehlermeldung „Zugriff verweigert“ erhalten

Vergewissern Sie sich, dass weder in der Bucket-Richtlinie noch in der identitätsbasierten Richtlinie eine ausdrückliche `Deny` Erklärung gegen den Antragsteller vorliegt, dem Sie Berechtigungen erteilen möchten.

Ausführliche Informationen zur Behebung von Fehlern bei Zugriffsverweigerung finden Sie unter. Problembehandlung bei verweigertem Zugriff beheben (403 Forbidden) Fehler in Amazon S3

Ich bin nicht berechtigt, eine Aktion in Amazon S3 durchzuführen

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson` IAM Benutzer versucht, die Konsole zu verwenden, um Details zu einer fiktiven *`my-example-widget`* Ressource anzuzeigen, aber nicht über die fiktiven `s3:`*`GetWidget`* Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 s3:GetWidget on resource: my-example-widget
```

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `s3:`*`GetWidget`*-Aktion auf die *`my-example-widget`*-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht berechtigt sind, die `iam:PassRole` Aktion durchzuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an Amazon S3 übergeben können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Service zu übergeben, anstatt eine neue Servicerolle oder eine dienstbezogene Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in Amazon S3 auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine Amazon S3 S3-Ressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob Amazon S3 diese Funktionen unterstützt, finden Sie unterSo funktioniert Amazon S3 mit IAM.
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM Benutzerhandbuch unter Gewähren des Zugriffs auf einen anderen IAMBenutzer AWS-Konto , den Sie besitzen.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAMBenutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- Informationen dazu, wie Sie Zugriff über einen Identitätsverbund gewähren, finden Sie im Benutzerhandbuch unter Zugriff für extern authentifizierte Benutzer (Identitätsverbund). IAM

- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie IAMim Benutzerhandbuch unter Kontoübergreifender Ressourcenzugriff. IAM

Problembehandlung bei verweigertem Zugriff beheben (403 Forbidden) Fehler in Amazon S3

Fehler „Zugriff verweigert“ (HTTP`403 Forbidden`) treten auf, wenn eine Autorisierungsanfrage AWS explizit oder implizit abgelehnt wird.

- Eine ausdrückliche Ablehnung tritt auf, wenn eine Richtlinie eine `Deny` Anweisung für die bestimmte Aktion enthält. AWS
- Eine implizite Verweigerung tritt auf, wenn es keine entsprechende `Deny`-Anweisung, aber auch keine anwendbare `Allow`-Anweisung gibt.

Da eine AWS Identity and Access Management (IAM) -Richtlinie standardmäßig implizit einen IAM Prinzipal ablehnt, muss die Richtlinie dem Prinzipal ausdrücklich erlauben, eine Aktion auszuführen. Andernfalls verweigert die Richtlinie implizit den Zugriff. Weitere Informationen finden Sie im Benutzerhandbuch unter Der Unterschied zwischen expliziten und impliziten Verweigerungen. IAM Informationen zur Richtlinienbewertungslogik, die bestimmt, ob eine Zugriffsanforderung zulässig oder verweigert wird, finden Sie unter Bewertungslogik für Richtlinien im IAMBenutzerhandbuch.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

Die folgenden Themen behandeln die häufigsten Ursachen für Fehler mit Zugriffsverweigerung in Amazon S3.

> **Note**
>
> Bei Fehlern mit Zugriff verweigert (HTTP`403 Forbidden`) berechnet Amazon S3 dem Bucket-Besitzer keine Gebühren, wenn die Anfrage außerhalb des individuellen AWS Kontos des Bucket-Besitzers oder der AWS Organisation des Bucket-Besitzers initiiert wird.

Themen

- Beispiele für abgelehnte Nachrichten und deren Behebung

- Bucket-Richtlinien und IAM Richtlinien
- Amazon S3 ACL S3-Einstellungen
- S3–Block-Public-Access-Einstellungen
- Amazon-S3-Verschlüsselungseinstellungen
- S3-Einstellungen für die Objektsperre
- VPCEndpunktrichtlinien
- AWS Organizations Richtlinien
- Zugriffspunkteinstellungen

> **Note**
>
> Wenn Sie versuchen, ein Berechtigungsproblem zu beheben, beginnen Sie mit dem the section called “Beispiele für abgelehnte Nachrichten und deren Behebung” Abschnitt und fahren Sie dann mit dem ??? Abschnitt fort. Achte außerdem darauf, die Anweisungen unter zu befolgenthe section called “Tipps zum Überprüfen von Berechtigungen”.

Beispiele für abgelehnte Nachrichten und deren Behebung

Amazon S3 enthält jetzt zusätzlichen Kontext in Access Denied (HTTP`403 Forbidden`) -Fehler für Anfragen an Ressourcen innerhalb desselben AWS-Konto. Dieser neue Kontext umfasst die Art der Richtlinie, die den Zugriff verweigert hat, den Grund für die Ablehnung und Informationen über den IAM Benutzer oder die Rolle, der den Zugriff auf die Ressource angefordert hat.

Dieser zusätzliche Kontext hilft Ihnen bei der Behebung von Zugriffsproblemen, der Identifizierung der Hauptursache von Fehlern bei der Zugriffsverweigerung und bei der Behebung falscher Zugriffskontrollen, indem Sie die entsprechenden Richtlinien aktualisieren. Dieser zusätzliche Kontext ist auch in AWS CloudTrail Protokollen verfügbar. Verbesserte Fehlermeldungen mit dem Status „Zugriff verweigert“ für Anfragen mit demselben Konto sind jetzt in allen Regionen verfügbar AWS-Regionen, auch in den Regionen China AWS GovCloud (US) Regions und China.

Die meisten Zugriffsverweigerungs-Fehlermeldungen haben das Format `User` *`user-arn`* `is not authorized to perform` *`action`* `on "`*`resource-arn`*`" because` *`context`*. In diesem Beispiel: *`Benutzer-ARN`* ist der Amazon-Ressourcenname (ARN) des Benutzers, der keinen Zugriff erhält, *`action`* ist die Serviceaktion, die die Richtlinie verweigert, und *`resource-arn`* ist die ARN

Ressource, auf die sich die Richtlinie bezieht. Das Tool *context* Das Feld steht für zusätzlichen Kontext zum Richtlinientyp, der erklärt, warum die Richtlinie den Zugriff verweigert hat.

Wenn eine Richtlinie den Zugriff ausdrücklich verweigert, weil die Richtlinie eine Deny Erklärung enthält, enthält die Fehlermeldung „Zugriff verweigert“ den folgenden Satzwith an explicit deny in a *type* policy. Wenn die Richtlinie implizit den Zugriff verweigert, enthält die Fehlermeldung „Zugriff verweigert“ den folgenden Satz. because no *type* policy allows the *action* action

Important

- Meldungen zur erweiterten Zugriffsverweigerung werden nur bei Anfragen für dasselbe Konto zurückgegeben. Bei kontoübergreifenden Anfragen wird eine generische Nachricht zurückgegeben. Access Denied

  Informationen zur Bewertungslogik für Richtlinien, mit der bestimmt wird, ob eine kontenübergreifende Zugriffsanfrage zulässig oder verweigert wird, finden Sie im Benutzerhandbuch unter Logik der IAM kontenübergreifenden Richtlinienbewertung. Eine exemplarische Vorgehensweise, die zeigt, wie Sie kontenübergreifenden Zugriff gewähren, finden Sie unter. the section called “Erteilen von kontoübergreifenden Berechtigungen”
- Für Anfragen an Directory-Buckets werden keine Fehlermeldungen mit dem Status „Erweiterter Zugriff verweigert“ zurückgegeben. Directory-Bucket-Anfragen geben eine generische Access Denied Meldung zurück.
- Wenn mehrere Richtlinien desselben Richtlinientyps eine Autorisierungsanfrage ablehnen, gibt die Fehlermeldung „Zugriff verweigert“ nicht die Anzahl der Richtlinien an.
- Wenn mehrere Richtlinientypen eine Autorisierungsanfrage ablehnen, enthält die Fehlermeldung nur einen dieser Richtlinientypen.
- Wenn eine Zugriffsanfrage aus mehreren Gründen abgelehnt wird, enthält die Fehlermeldung nur einen der Gründe für die Ablehnung.

Die folgenden Beispiele zeigen das Format für die verschiedenen Arten von Fehlermeldungen mit Zugriffsverweigerung und die Problembehandlung der einzelnen Meldungstypen.

Der Zugriff wurde aufgrund einer Ressourcensteuerungsrichtlinie verweigert — explizite Ablehnung

1. Suchen Sie in Ihren Ressourcenkontrollrichtlinien nach einer `Deny` Erklärung zu der Aktion (RCPs). Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre, RCP indem Sie die `Deny` Erklärung entfernen. Weitere Informationen finden Sie im AWS Organizations Benutzerhandbuch unter Aktualisieren einer Ressourcenkontrollrichtlinie (RCP).

```
An error occurred (AccessDenied) when calling the GetObject operation:
User: arn:aws:iam::777788889999:user/MaryMajor is not authorized to perform:
s3:GetObject on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-name"
with an explicit deny in a resource control policy
```

Zugriffsverweigerung aufgrund einer Service-Kontrollrichtlinie – implizite Verweigerung

1. Suchen Sie in Ihren Richtlinien zur Servicesteuerung nach einer fehlenden `Allow` Angabe für die Aktion (SCPs). Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre, SCP indem Sie die `Allow` Erklärung hinzufügen. Weitere Informationen finden Sie unter Aktualisieren von SCP im AWS Organizations Benutzerhandbuch.

```
User: arn:aws:iam::777788889999:user/MaryMajor is not authorized to perform:
s3:GetObject because no service control policy allows the s3:GetObject action
```

Zugriffsverweigerung aufgrund einer Service-Kontrollrichtlinie – explizite Verweigerung

1. Suchen Sie in Ihren Service Control-Richtlinien (SCPs) nach einer `Deny` Erklärung zu der Aktion. Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre, SCP indem Sie die `Deny` Erklärung ändern, um dem Benutzer den erforderlichen Zugriff zu gewähren. Ein Beispiel dafür, wie Sie dies tun können, finden Sie im Benutzerhandbuch unter IAMVerhindern, dass Benutzer und Rollen bestimmte Änderungen vornehmen, mit einer Ausnahme für eine bestimmte Administratorrolle.AWS Organizations Weitere Informationen zur Aktualisierung Ihres SCP finden Sie unter Aktualisieren von SCP im AWS Organizations Benutzerhandbuch.

```
User: arn:aws:iam::777788889999:user/MaryMajor is not authorized to perform:
```

```
s3:GetObject with an explicit deny in a service control policy
```

Zugriff aufgrund einer VPC Endpunktrichtlinie verweigert — implizite Ablehnung

1. Suchen Sie in Ihren Virtual Private Cloud (VPC) -Endpunktrichtlinien nach einer fehlenden `Allow` Angabe für die Aktion. Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre VPC Endpunktrichtlinie, indem Sie die `Allow` Erklärung hinzufügen. Weitere Informationen finden Sie im AWS PrivateLink Handbuch unter Aktualisieren einer VPC Endpunktrichtlinie.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject because no VPC endpoint policy allows the s3:GetObject action
```

Zugriff aufgrund einer VPC Endpunktrichtlinie verweigert — explizite Ablehnung

1. Suchen Sie in Ihren Virtual Private Cloud (VPC) -Endpunktrichtlinien nach einer ausdrücklichen `Deny` Erklärung für die Aktion. Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre VPC Endpunktrichtlinie, indem Sie die `Deny` Erklärung ändern, um dem Benutzer den erforderlichen Zugriff zu gewähren. Sie können Ihre `Deny` Anweisung beispielsweise so aktualisieren, dass der `aws:PrincipalAccount` Bedingungsschlüssel zusammen mit dem `StringNotEquals` Bedingungsoperator verwendet wird, um dem jeweiligen Hauptbenutzer Zugriff zu gewähren, wie unter gezeigtthe section called “Beispiel 7: Ausschließen von Hauptpersonen aus Kontoauszügen Deny”. Weitere Informationen zur Aktualisierung Ihrer VPC Endpunktrichtlinie finden Sie im AWS PrivateLink Handbuch unter Aktualisieren einer VPC Endpunktrichtlinie.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-name" with
an explicit deny in a VPC endpoint policy
```

Zugriff aufgrund einer Berechtigungsgrenze verweigert – implizite Ablehnung

1. Überprüfen Sie, ob in Ihrer Berechtigungsgrenze eine `Allow`-Anweisung für die Aktion fehlt. Für das folgende Beispiel lautet die Aktion `s3:GetObject`.

2. Aktualisieren Sie Ihre Zugriffsgrenzen, indem Sie die `Allow` Erklärung zu Ihrer IAM Richtlinie hinzufügen. Weitere Informationen finden Sie unter Zugriffsgrenzen für IAM Entitäten und Bearbeiten von IAM Richtlinien im IAMBenutzerhandbuch.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-name"
because no permissions boundary allows the s3:GetObject action
```

Zugriff aufgrund einer Berechtigungsgrenze verweigert – explizite Ablehnung

1. Überprüfen Sie, ob in Ihren Berechtigungsgrenzen eine explizite `Deny`-Anweisung für die Aktion vorhanden ist. Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre Berechtigungsgrenze, indem Sie die `Deny` Aussage in Ihrer IAM Richtlinie ändern, um dem Benutzer den erforderlichen Zugriff zu gewähren. Sie können Ihre `Deny` Anweisung beispielsweise so aktualisieren, dass der `aws:PrincipalAccount` Bedingungsschlüssel zusammen mit dem `StringNotEquals` Bedingungsoperator verwendet wird, um dem jeweiligen Hauptbenutzer Zugriff zu gewähren, wie unter aws:PrincipalAccount im IAM-Benutzerhandbuch. Weitere Informationen finden Sie unter Berechtigungsgrenzen für IAM Entitäten und IAMBearbeitungsrichtlinien im IAMBenutzerhandbuch.

```
User: arn:aws:iam::777788889999:user/MaryMajor is not authorized to perform:
s3:GetObject with an explicit deny in a permissions boundary
```

Zugriff aufgrund von Sitzungsrichtlinien verweigert – implizite Ablehnung

1. Überprüfen Sie, ob in Ihren Sitzungsrichtlinien eine `Allow`-Anweisung für die Aktion fehlt. Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre Sitzungsrichtlinie, indem Sie die `Allow`-Anweisung hinzufügen. Weitere Informationen finden Sie unter Sitzungsrichtlinien und IAMBearbeitungsrichtlinien im IAMBenutzerhandbuch.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject because no session policy allows the s3:GetObject action
```

Zugriff aufgrund von Sitzungsrichtlinien verweigert – explizite Ablehnung

1. Überprüfen Sie, ob in Ihren Sitzungsrichtlinien eine explizite `Deny`-Anweisung für die Aktion vorhanden ist. Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre Sitzungsrichtlinie, indem Sie die `Deny` Erklärung ändern, um dem Benutzer den erforderlichen Zugriff zu gewähren. Sie können Ihre `Deny` Anweisung beispielsweise so aktualisieren, dass der `aws:PrincipalAccount` Bedingungsschlüssel zusammen mit dem `StringNotEquals` Bedingungsoperator verwendet wird, um dem jeweiligen Hauptbenutzer Zugriff zu gewähren, wie unter gezeigtthe section called “Beispiel 7: Ausschließen von Hauptpersonen aus Kontoauszügen Deny”. Weitere Informationen zur Aktualisierung Ihrer Sitzungsrichtlinie finden Sie unter Sitzungsrichtlinien und IAMBearbeitungsrichtlinien im IAMBenutzerhandbuch.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-name" with
an explicit deny in a session policy
```

Zugriff aufgrund ressourcenbasierter Richtlinien verweigert – implizite Ablehnung

> **Note**
>
> Unter ressourcenbasierten Richtlinien sind Richtlinien wie Bucket-Richtlinien und Zugriffspunktrichtlinien zu verstehen.

1. Überprüfen Sie, ob in Ihrer ressourcenbasierten Richtlinie eine `Allow`-Anweisung für die Aktion fehlt. Prüfen Sie auch, ob die Einstellung `IgnorePublicAcls` S3 Block Public Access auf Bucket-, Access Point- oder Kontoebene angewendet wird. Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre Richtlinie, indem Sie die `Allow`-Anweisung hinzufügen. Weitere Informationen finden Sie unter Ressourcenbasierte Richtlinien und IAMBearbeitungsrichtlinien im IAMBenutzerhandbuch.

   Möglicherweise müssen Sie auch Ihre Einstellung „Öffentlichen Zugriff `IgnorePublicAcls` blockieren“ für den Bucket, den Access Point oder das Konto anpassen. Weitere Informationen erhalten Sie unter the section called “Der Zugriff wurde aufgrund der Einstellungen zum Blockieren

des öffentlichen Zugriffs verweigert” und the section called “Konfigurieren von Bucket- und Zugriffspunktseinstellungen”.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject because no resource-based policy allows the s3:GetObject action
```

Zugriff aufgrund ressourcenbasierter Richtlinien verweigert – explizite Ablehnung

> **Note**
>
> Unter ressourcenbasierten Richtlinien sind Richtlinien wie Bucket-Richtlinien und Zugriffspunktrichtlinien zu verstehen.

1. Suchen Sie nach einer expliziten `Deny`-Anweisung für die Aktion in Ihrer ressourcenbasierten Richtlinie. Prüfen Sie auch, ob die Einstellung `RestrictPublicBuckets` S3 Block Public Access auf Bucket-, Access Point- oder Kontoebene angewendet wird. Für das folgende Beispiel lautet die Aktion `s3:GetObject`.
2. Aktualisieren Sie Ihre Richtlinie, indem Sie die `Deny` Erklärung ändern, um dem Benutzer den erforderlichen Zugriff zu gewähren. Sie können Ihre `Deny` Anweisung beispielsweise so aktualisieren, dass der `aws:PrincipalAccount` Bedingungsschlüssel zusammen mit dem `StringNotEquals` Bedingungsoperator verwendet wird, um dem jeweiligen Hauptbenutzer Zugriff zu gewähren, wie unter gezeigtthe section called “Beispiel 7: Ausschließen von Hauptpersonen aus Kontoauszügen `Deny`”. Weitere Informationen zur Aktualisierung Ihrer ressourcenbasierten Richtlinie finden Sie unter Ressourcenbasierte Richtlinien und IAM Bearbeitungsrichtlinien im Benutzerhandbuch. IAM

   Möglicherweise müssen Sie auch Ihre Einstellung „Öffentlichen Zugriff `RestrictPublicBuckets` sperren“ für den Bucket, den Access Point oder das Konto anpassen. Weitere Informationen erhalten Sie unter the section called “Der Zugriff wurde aufgrund der Einstellungen zum Blockieren des öffentlichen Zugriffs verweigert” und the section called “Konfigurieren von Bucket- und Zugriffspunktseinstellungen”.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-name" with
an explicit deny in a resource-based policy
```

Zugriff aufgrund identitätsbasierter Richtlinien verweigert – implizite Verweigerung

1. Überprüfen Sie ob in identitätsbasierten Richtlinien, die der Identität angefügt sind, eine `Allow`-Anweisung für die Aktion fehlt. Im folgenden Beispiel ist die Aktion an den Benutzer `s3:GetObject` angehängt`MaryMajor`.
2. Aktualisieren Sie Ihre Richtlinie, indem Sie die `Allow`-Anweisung hinzufügen. Weitere Informationen finden Sie unter Identitätsbasierte Richtlinien und IAMBearbeitungsrichtlinien im IAMBenutzerhandbuch.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject because no identity-based policy allows the s3:GetObject action
```

Zugriff aufgrund identitätsbasierter Richtlinien verweigert – explizite Ablehnung

1. Überprüfen Sie, ob in identitätsbasierten Richtlinien, die der Identität angefügt sind, eine explizite `Deny`-Anweisung für die Aktion vorhanden ist. Im folgenden Beispiel ist die Aktion an den Benutzer `s3:GetObject` angehängt. `MaryMajor`
2. Aktualisieren Sie Ihre Richtlinie, indem Sie die `Deny` Erklärung ändern, um dem Benutzer den erforderlichen Zugriff zu gewähren. Sie können Ihre `Deny` Erklärung beispielsweise so aktualisieren, dass der `aws:PrincipalAccount` Bedingungsschlüssel zusammen mit dem `StringNotEquals` Bedingungsoperator verwendet wird, um dem jeweiligen Hauptbenutzer Zugriff zu gewähren, wie unter aws:PrincipalAccount im IAM-Benutzerhandbuch. Weitere Informationen finden Sie unter Identitätsbasierte Richtlinien und IAMBearbeitungsrichtlinien im IAMBenutzerhandbuch.

```
User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
s3:GetObject on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-name" with
an explicit deny in an identity-based policy
```

Der Zugriff wurde aufgrund der Einstellungen zum Blockieren des öffentlichen Zugriffs verweigert

Die Amazon S3 Block Public Access-Funktion bietet Einstellungen für Zugriffspunkte, Buckets und Konten, mit denen Sie den öffentlichen Zugriff auf Amazon-S3-Ressourcen verwalten können. Weitere Informationen dazu, wie „öffentlich“ in Amazon S3 definiert ist, finden Sie unter Die Bedeutung von „öffentlich“.

Standardmäßig erlauben neue Buckets, Zugriffspunkte und Objekte keinen öffentlichen Zugriff. Benutzer können jedoch Bucket-Richtlinien, Zugriffspunktrichtlinien, IAM Benutzerrichtlinien, Objektberechtigungen oder Zugriffskontrolllisten (ACLs) ändern, um den öffentlichen Zugriff zu ermöglichen. Die Einstellungen von S3 Block Public Access haben Vorrang vor diesen Richtlinien, Berechtigungen undACLs. Seit April 2023 sind alle Einstellungen für Block Public Access standardmäßig für neue Buckets aktiviert.

Wenn Amazon S3 eine Anforderung zum Zugriff auf einen Bucket oder ein Objekt erhält, wird ermittelt, ob für den Bucket oder das Konto des Bucket-Eigentümers eine Block Public Access-Einstellung vorliegt. Wenn die Anforderung über einen Zugriffspunkt einging, prüft Amazon S3 auch auf Block Public Access-Einstellungen für den Zugriffspunkt. Wenn eine Block Public Access-Einstellung vorhanden ist, die den angeforderten Zugriff verbietet, lehnt Amazon S3 die Anforderung ab.

Amazon S3 Block Public Access bietet vier Einstellungen. Diese Einstellungen sind voneinander unabhängig und können in beliebiger Kombination verwendet werden. Jede Einstellung kann auf einen Access Point, einen Bucket oder ein ganzes AWS Konto angewendet werden. Wenn sich die Block Public Access-Einstellungen für den Zugriffspunkt, den Bucket oder das Konto unterscheiden, wendet Amazon S3 die restriktivste Kombination der Zugriffspunkt-, Bucket- und Kontoeinstellungen an.

Wenn Amazon S3 ermittelt, ob eine Operation von einer Block Public Access-Einstellung untersagt wird, werden alle Anforderungen abgelehnt, die gegen eine Zugriffspunkt-, Bucket- oder Konto-Einstellung verstoßen.

Die vier von Amazon S3 Block Public Access bereitgestellten Einstellungen lauten wie folgt:

- `BlockPublicAcls`— Diese Einstellung gilt für `PutBucketAclPutObjectAcl`,`PutObject`, `CreateBucketCopyObject`, und `POST Object` Anfragen. Die `BlockPublicAcls` Einstellung verursacht das folgende Verhalten:
  - `PutBucketAcl`und `PutObjectAcl` Aufrufe schlagen fehl, wenn die angegebene Zugriffskontrollliste (ACL) öffentlich ist.
  - `PutObject`Aufrufe schlagen fehl, wenn die Anfrage ein öffentliches Objekt beinhaltetACL.
  - Wenn diese Einstellung auf ein Konto angewendet wird, schlagen `CreateBucket` Aufrufe mit einer HTTP `400` (`Bad Request`) -Antwort fehl, wenn die Anfrage ein öffentliches Objekt beinhaltetACL.

Wenn beispielsweise der Zugriff auf eine `CopyObject` Anfrage aufgrund der `BlockPublicAcls` Einstellung verweigert wird, erhalten Sie die folgende Meldung:

```
An error occurred (AccessDenied) when calling the CopyObject operation:
User: arn:aws:sts::123456789012:user/MaryMajor is not authorized to
perform: s3:CopyObject on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-name"
because public access control lists (ACLs) are blocked by the BlockPublicAcls block
public access setting.
```

- `IgnorePublicAcls`— Die `IgnorePublicAcls` Einstellung veranlasst Amazon S3, alle öffentlichen ACLs Informationen in einem Bucket und alle darin enthaltenen Objekte zu ignorieren. Wenn die Genehmigung Ihrer Anfrage nur von einer Öffentlichkeit erteilt wirdACL, lehnt die `IgnorePublicAcls` Einstellung die Anfrage ab.

  Jede Ablehnung, die sich aus der `IgnorePublicAcls` Einstellung ergibt, ist implizit. Wenn Sie beispielsweise eine `GetObject` Anfrage aufgrund `IgnorePublicAcls` eines öffentlichen Zugriffs ablehnenACL, erhalten Sie die folgende Meldung:

  ```
  User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
  s3:GetObject because no resource-based policy allows the s3:GetObject action
  ```

- `BlockPublicPolicy`— Diese Einstellung gilt für `PutAccessPointPolicy` Anfragen `PutBucketPolicy` und.

  Die Einstellung `BlockPublicPolicy` für einen Bucket führt dazu, dass Amazon S3 Aufrufe ablehnt, `PutBucketPolicy` wenn die angegebene Bucket-Richtlinie öffentlichen Zugriff zulässt. Diese Einstellung veranlasst Amazon S3 auch, Anrufe an `PutAccessPointPolicy` alle Access Points des Buckets mit demselben Konto abzulehnen, wenn die angegebene Richtlinie öffentlichen Zugriff zulässt.

  Die Einstellung `BlockPublicPolicy` für einen Access Point veranlasst Amazon S3, Anrufe an `PutAccessPointPolicy` und über den Access Point getätigte Anrufe abzulehnen`PutBucketPolicy`, wenn die angegebene Richtlinie (entweder für den Access Point oder den zugrundeliegenden Bucket) den öffentlichen Zugriff zulässt.

  Wenn beispielsweise der Zugriff auf eine `PutBucketPolicy` Anfrage aufgrund der `BlockPublicPolicy` Einstellung verweigert wird, erhalten Sie die folgende Meldung:

  ```
  An error occurred (AccessDenied) when calling the PutBucketPolicy operation:
  ```

```
User: arn:aws:sts::123456789012:user/MaryMajor is not authorized to
perform: s3:PutBucketPolicy on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-
name"
because public policies are blocked by the BlockPublicPolicy block public
access setting.
```

- `RestrictPublicBuckets`— Die `RestrictPublicBuckets` Einstellung beschränkt den Zugriff auf einen Access Point oder Bucket mit einer öffentlichen Richtlinie nur auf AWS-Service Prinzipale und autorisierte Benutzer innerhalb des Kontos des Bucket-Besitzers und des Accounts des Access Point-Besitzers. Diese Einstellung blockiert den gesamten kontoübergreifenden Zugriff auf den Access Point oder Bucket (außer durch AWS-Service Principals), ermöglicht aber dennoch Benutzern innerhalb des Kontos, den Access Point oder Bucket zu verwalten. Diese Einstellung weist auch alle anonymen (oder unsignierten) Aufrufe zurück.

  Jede Ablehnung, die sich aus dieser `RestrictPublicBuckets` Einstellung ergibt, ist explizit. Wenn beispielsweise eine `GetObject` Anfrage aufgrund einer `RestrictPublicBuckets` öffentlichen Bucket- oder Access-Point-Richtlinie abgelehnt wird, erhalten Sie die folgende Meldung:

  ```
  User: arn:aws:iam::123456789012:user/MaryMajor is not authorized to perform:
  s3:GetObject on resource: "arn:aws:s3:::amzn-s3-demo-bucket1/object-name" with
  an explicit deny in a resource-based policy
  ```

Weitere Informationen zu diesen Einstellungen finden Sie unter the section called “Block Public Access-Einstellungen”. Informationen zum Überprüfen und Aktualisieren dieser Einstellungen finden Sie unterthe section called “Verwenden von Block Public Access”.

Bucket-Richtlinien und IAM Richtlinien

Vorgänge auf Bucket-Ebene

Wenn keine Bucket-Richtlinie vorhanden ist, erlaubt der Bucket implizit Anfragen von einer beliebigen AWS Identity and Access Management (IAM) Identität im Konto des Bucket-Besitzers. Der Bucket lehnt auch implizit Anfragen von anderen IAM Identitäten von anderen Konten sowie anonyme (unsignierte) Anfragen ab. Wenn jedoch keine IAM Benutzerrichtlinie vorhanden ist, wird dem Anforderer (es sei denn, er ist der AWS-Konto Root-Benutzer) implizit untersagt, Anfragen zu stellen. Weitere Informationen zu dieser Bewertungslogik finden Sie im IAMBenutzerhandbuch unter Feststellen, ob eine Anfrage innerhalb eines Kontos abgelehnt oder zulässig ist.

Vorgänge auf Objektebene

Wenn das Objekt dem Konto gehört, das den Bucket besitzt, funktionieren die Bucket-Richtlinie und die IAM Benutzerrichtlinie für Operationen auf Objektebene genauso wie für Operationen auf Bucket-Ebene. Wenn beispielsweise keine Bucket-Richtlinie vorhanden ist, lässt der Bucket implizit Objektanforderungen von beliebigen Identitäten im Konto des Bucket-Besitzers zu. IAM Der Bucket lehnt auch implizit Objektanfragen von anderen IAM Identitäten aus anderen Konten sowie anonyme (unsignierte) Anfragen ab. Wenn jedoch keine IAM Benutzerrichtlinie vorhanden ist, wird dem Anforderer (es sei denn, er ist der AWS-Konto Root-Benutzer) implizit untersagt, Objektanforderungen zu stellen.

Wenn das Objekt einem externen Konto gehört, kann der Zugriff auf das Objekt nur über Objektzugriffskontrolllisten () ACLs gewährt werden. Die Bucket-Richtlinie und die IAM Benutzerrichtlinie können weiterhin verwendet werden, um Objektanforderungen abzulehnen.

Um sicherzustellen, dass Ihre Bucket-Richtlinie oder IAM Benutzerrichtlinie nicht den Fehler Access Denied (403 Forbidden) verursacht, stellen Sie daher sicher, dass die folgenden Anforderungen erfüllt sind:

- Für den Zugriff auf dasselbe Konto darf es weder in der Bucket-Richtlinie noch in der Benutzerrichtlinie eine ausdrückliche `Deny` Erklärung gegen den Antragsteller geben, dem Sie Berechtigungen gewähren möchten. IAM Wenn Sie Berechtigungen nur mithilfe der Bucket-Richtlinie und der IAM Benutzerrichtlinie gewähren möchten, muss in einer dieser Richtlinien mindestens eine ausdrückliche `Allow` Aussage enthalten sein.
- Für kontoübergreifenden Zugriff darf es weder in der Bucket-Richtlinie noch in der IAM Benutzerrichtlinie eine ausdrückliche `Deny` Aussage gegen den Antragsteller geben, dem Sie Berechtigungen erteilen möchten. Wenn Sie kontoübergreifende Berechtigungen nur mithilfe der Bucket-Richtlinie und der IAM Benutzerrichtlinie gewähren möchten, stellen Sie sicher, dass sowohl die Bucket-Richtlinie als auch die IAM Benutzerrichtlinie des Anforderers eine ausdrückliche Erklärung enthalten. `Allow`

> **Note**
>
> `Allow`-Anweisungen in einer Bucket-Richtlinie gelten nur für Objekte, die demselben Konto gehören, das im Besitz des Buckets ist. `Deny`-Anweisungen in einer Bucket-Richtlinie gelten jedoch für alle Objekte, unabhängig von der Objekteigentümerschaft.

So überprüfen oder bearbeiten Sie Ihre Bucket-Richtlinie

> **Note**
>
> Um eine Bucket-Richtlinie anzuzeigen oder zu bearbeiten, benötigen Sie die Berechtigung `s3:GetBucketPolicy`.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie eine Bucket-Richtlinie anzeigen oder bearbeiten möchten.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Wählen Sie unter Bucket-Richtlinie Bearbeiten aus. Die Seite Bucket-Richtlinie bearbeiten wird angezeigt.

Um Ihre Bucket-Richtlinie mithilfe von AWS Command Line Interface (AWS CLI) zu überprüfen oder zu bearbeiten, verwenden Sie die get-bucket-policyBefehl.

> **Note**
>
> Wenn Sie aufgrund einer falschen Bucket-Richtlinie aus einem Bucket ausgesperrt werden, melden Sie sich mit Ihren AWS-Konto Root-Benutzeranmeldedaten bei dem AWS Management Console an. Um wieder Zugriff auf Ihren Bucket zu erhalten, stellen Sie sicher, dass Sie die falsche Bucket-Richtlinie mithilfe Ihrer AWS-Konto Root-Benutzeranmeldedaten löschen.

Tipps zum Überprüfen von Berechtigungen

Gehen Sie wie folgt vor, um zu überprüfen, ob der Anforderer über die erforderlichen Berechtigungen für die Ausführung einer Amazon-S3-Operation verfügt:

- Ermitteln Sie den Anforderer. Wenn es sich um eine unsignierte Anfrage handelt, handelt es sich um eine anonyme Anfrage ohne IAM Benutzerrichtlinie. Wenn es sich um eine Anfrage handelt, für

die eine vorsignierte Anfrage verwendet wirdURL, gilt dieselbe Benutzerrichtlinie wie für den IAM Benutzer oder die Rolle, die die Anfrage signiert hat.

- Stellen Sie sicher, dass Sie den richtigen IAM Benutzer oder die richtige Rolle verwenden. Sie können Ihren IAM Benutzer oder Ihre Rolle verifizieren, indem Sie in der oberen rechten Ecke von AWS Management Console oder nachsehen, indem Sie aws sts get-caller-identityBefehl.
- Überprüfen Sie die IAM Richtlinien, die sich auf den IAM Benutzer oder die Rolle beziehen. Sie können eine der folgenden Methoden verwenden:
  - Testen Sie IAM Richtlinien mit dem IAM Richtliniensimulator.
  - Überprüfen Sie die verschiedenen IAMRichtlinientypen.
- Bearbeiten Sie bei Bedarf Ihre IAM Benutzerrichtlinie.
- Sehen Sie sich die folgenden Beispiele für Richtlinien an, die den Zugriff explizit verweigern oder zulassen:
  - IAMBenutzerrichtlinie „Explizite Zulassung“ IAM: Erlaubt und verweigert programmgesteuert und in der Konsole den Zugriff auf mehrere Dienste
  - Explizite Bucket-Richtlinie „Zulassen“: Erteilung von Berechtigungen für mehrere Konten, um Objekte hochzuladen oder Objekte öffentlich zugänglich zu machen ACLs
  - Richtlinie „IAMBenutzer explizit verweigern“ AWS: Verweigert den Zugriff AWS auf die angeforderte Person AWS-Region
  - Explizite Richtlinie zum Ablehnen von Buckets: SSEErforderlich — KMS für alle Objekte, die in einen Bucket geschrieben werden

Amazon S3 ACL S3-Einstellungen

Überprüfen Sie bei der Überprüfung Ihrer ACL Einstellungen zunächst Ihre Einstellungen für Object Ownership, um zu überprüfen, ob ACLs sie für den Bucket aktiviert sind. Beachten Sie, dass ACL Berechtigungen nur zur Erteilung von Berechtigungen und nicht zur Ablehnung von Anfragen verwendet werden können. ACLskann auch nicht verwendet werden, um Anforderern Zugriff zu gewähren, die durch ausdrückliche Ablehnungen in Bucket-Richtlinien oder IAM Benutzerrichtlinien abgelehnt wurden.

Die Einstellung für Object Ownership ist auf „Bucket-Eigentümer erzwungen“ gesetzt.

Wenn die Einstellung „Bucket Owner erforced“ aktiviert ist, ist es unwahrscheinlich, dass die ACL Einstellungen zu einem Fehler „Zugriff verweigert“ (403 Forbidden) führen, da diese Einstellung alles deaktiviertACLs, was für Buckets und Objekte gilt. „Bucket-Eigentümer erzwungen“ ist die Standardeinstellung (und empfohlene Einstellung) für Amazon-S3-Buckets.

Die Einstellung für Object Ownership ist auf „Bucket-Eigentümer bevorzugt“ oder „Objektschreiber“ gesetzt.

ACLDie Berechtigungen sind weiterhin gültig, wenn die bevorzugte Einstellung für den Bucket-Besitzer oder die Einstellung für den Objekt-Writer gilt. Es gibt zwei Arten vonACLs: Bucket ACLs und ObjektACLs. Die Unterschiede zwischen diesen beiden Typen finden Sie unter Zuordnung von ACL Berechtigungen und Zugriffsrichtlinienberechtigungen. ACLs

Überprüfen Sie je nach der Aktion der abgelehnten Anfrage die ACL Berechtigungen für Ihren Bucket oder das Objekt:

- Wenn Amazon S3 ein `LIST` `PUT` Objekt oder eine `PutBucketAcl` Anfrage abgelehnt hat, überprüfen Sie die ACL Berechtigungen für Ihren Bucket. `GetBucketAcl`

  > **Note**
  >
  > Mit den ACL Bucket-Einstellungen können Sie keine `GET` Objektberechtigungen gewähren.

- Wenn Amazon S3 eine `GET` Anfrage für ein S3-Objekt abgelehnt hat, oder PutObjectAclAnfrage und überprüfen Sie anschließend die ACL Berechtigungen für das Objekt.

  > **Important**
  >
  > Wenn es sich bei dem Konto, dem das Objekt gehört, nicht um das Konto handelt, das im Besitz des Buckets ist, wird der Zugriff auf das Objekt nicht durch die Bucket-Richtlinie gesteuert.

Beheben eines Fehlers aufgrund einer Zugriffsverweigerung (403 Forbidden) bei einer **`GET`**-Objekt-Anforderung während einer kontoübergreifenden Objekteigentümerschaft

Überprüfen Sie die Einstellungen für Object Ownership des Buckets, um den Objekteigentümer zu ermitteln. Wenn Sie Zugriff auf das Objekt habenACLs, können Sie auch das Konto des Objektbesitzers überprüfen. (Um das Konto des Objekteigentümers einzusehen, überprüfen Sie die ACL Objekteinstellungen in der Amazon S3 S3-Konsole.) Alternativ können Sie auch eine `GetObjectAcl`-Anforderung stellen, um die kanonische ID des Objekteigentümers zu ermitteln und so das Konto des Objekteigentümers überprüfen zu können. Standardmäßig ACLs gewähren Sie dem Konto des Objekteigentümers explizite Genehmigungsberechtigungen für `GET` Anfragen.

Nachdem Sie bestätigt haben, dass der Objekteigentümer nicht mit dem Bucket-Eigentümer identisch ist, wählen Sie je nach Anwendungsfall und Zugriffsebene eine der folgenden Methoden aus, um den Fehler aufgrund einer Zugriffsverweigerung (403 Forbidden) zu beheben:

- Deaktivieren ACLs (empfohlen) — Diese Methode gilt für alle Objekte und kann vom Bucket-Besitzer ausgeführt werden. Bei dieser Methode erhält der Bucket-Eigentümer automatisch das Eigentum an jedem Objekt im Bucket und die volle Kontrolle darüber. Bevor Sie diese Methode implementieren, überprüfen Sie die Voraussetzungen für die Deaktivierung ACLs. Informationen dazu, wie Sie Ihren Bucket auf den Modus „Bucket-Eigentümer erzwungen“ (empfohlen) setzen, finden Sie unter Einstellung für Object Ownership für einen vorhandenen Bucket.

  > **Important**
  >
  > Um zu verhindern, dass der Fehler „Zugriff verweigert“ (403 Forbidden) angezeigt wird, müssen Sie die ACL Berechtigungen vor der Deaktivierung ACLs auf eine Bucket-Richtlinie migrieren. Weitere Informationen finden Sie unter Beispiele für Bucket-Richtlinien für die Migration von ACL Berechtigungen.

- Objekteigentümer in Bucket-Eigentümer ändern – Diese Methode kann auf einzelne Objekte angewendet werden, doch nur der Objekteigentümer (oder ein Benutzer mit den entsprechenden Berechtigungen) kann die Eigentümerschaft eines Objekts ändern. Zusätzliche PUT Kosten können anfallen. (Weitere Informationen finden Sie unter Amazon S3 – Preise.) Diese Methode gewährt dem Bucket-Eigentümer das vollständige Eigentum an dem Objekt, sodass der Bucket-Eigentümer den Zugriff auf das Objekt über eine Bucket-Richtlinie steuern kann.

  Führen Sie einen der folgenden Schritte aus, um die Objekteigentümerschaft zu ändern:

  - Sie (der Bucket-Eigentümer) können das Objekt wieder in den Bucket kopieren.
  - Sie können die Einstellung für Object Ownership des Buckets auf „Bucket-Eigentümer bevorzugt“ setzen. Wenn die Versionsverwaltung deaktiviert ist, werden die Objekte im Bucket überschrieben. Wenn die Versionsverwaltung aktiviert ist, werden doppelte Versionen desselben Objekts im Bucket angezeigt, für die der Bucket-Eigentümer eine Lebenszyklusregel festlegen kann, damit sie ablaufen. Anweisungen zum Ändern der Einstellung für Object Ownership finden Sie unter Einstellung für Object Ownership für einen vorhandenen Bucket.

> **Note**
>
> Wenn Sie Ihre Einstellung für Object Ownership in „Bucket-Eigentümer bevorzugt“ aktualisieren, wird die Einstellung nur auf neue Objekte angewendet, die in den Bucket hochgeladen werden.

- Sie können den Objekteigentümer das Objekt zusammen mit dem gespeicherten Objekt `bucket-owner-full-control` erneut hochladen lassenACL.

> **Note**
>
> Für kontoübergreifende Uploads können Sie das `bucket-owner-full-control` gespeicherte Objekt auch ACL in Ihrer Bucket-Richtlinie vorschreiben. Ein Beispiel für eine Bucket-Richtlinie finden Sie unter Erteilung von kontoübergreifenden Berechtigungen für das Hochladen von Objekten, wobei sichergestellt wird, dass der Bucket-Eigentümer volle Kontrolle besitzt.

- Objektschreiber als Objekteigentümer beibehalten – Mit dieser Methode wird der Objekteigentümer nicht geändert, Sie können jedoch den Zugriff auf Objekte einzeln gewähren. Um Zugriff auf ein Objekt zu gewähren, müssen Sie über die Berechtigung `PutObjectAcl` für das Objekt verfügen. Um dann den Fehler „Zugriff verweigert“ (403 Forbidden) zu beheben, fügen Sie den Antragsteller als Empfänger für den Zugriff auf das Objekt im Objekt hinzu. ACLs Weitere Informationen finden Sie unter Konfiguration ACLs.

S3–Block-Public-Access-Einstellungen

Wenn die fehlgeschlagene Anfrage öffentlichen Zugriff oder öffentliche Richtlinien beinhaltet, überprüfen Sie die S3-Einstellungen für Block Public Access in Ihrem Konto, Bucket oder Access Point. Weitere Informationen zur Behebung von Fehlern im Zusammenhang mit der Einstellung „Zugriff verweigert“ im Zusammenhang mit den Einstellungen zum Blockieren des öffentlichen Zugriffs in S3 finden Sie unterthe section called “Der Zugriff wurde aufgrund der Einstellungen zum Blockieren des öffentlichen Zugriffs verweigert”.

Amazon-S3-Verschlüsselungseinstellungen

Amazon S3 unterstützt die serverseitige Verschlüsselung in Ihrem Bucket. Serverseitige Verschlüsselung ist die Verschlüsselung von Daten am Zielort durch die Anwendung oder den

Service, der sie erhält. Amazon S3 verschlüsselt Ihre Daten auf Objektebene, wenn es sie auf Festplatten in AWS Rechenzentren schreibt, und entschlüsselt sie für Sie, wenn Sie darauf zugreifen.

Standardmäßig wendet Amazon S3 jetzt serverseitige Verschlüsselung mit verwalteten Amazon S3-Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Amazon S3 ermöglicht es Ihnen auch, die serverseitige Verschlüsselungsmethode beim Hochladen von Objekten anzugeben.

So überprüfen Sie den Status der serverseitigen Verschlüsselung und die Verschlüsselungseinstellungen Ihres Buckets

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Bucket aus, für den Sie die Verschlüsselungseinstellungen überprüfen möchten.
4. Wählen Sie die Registerkarte Eigenschaften aus.
5. Scrollen Sie nach unten zum Abschnitt Standardverschlüsselung und sehen Sie sich die Einstellungen für den Verschlüsselungstyp an.

Um Ihre Verschlüsselungseinstellungen mit dem zu überprüfen AWS CLI, verwenden Sie den get-bucket-encryptionBefehl.

So überprüfen Sie den Verschlüsselungsstatus eines Objekts

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.
4. Wählen Sie in der Liste Objekte den Namen des Objekts aus, für das Sie eine Verschlüsselung hinzufügen oder ändern möchten.

   Die Seite mit den Objektdetails wird angezeigt.
5. Scrollen Sie nach unten zum Abschnitt Serverseitige Verschlüsselungseinstellungen, um die serverseitigen Verschlüsselungseinstellungen des Objekts anzuzeigen.

Um den Status Ihrer Objektverschlüsselung mit dem zu überprüfen AWS CLI, verwenden Sie den head-objectBefehl.

Verschlüsselungs- und Berechtigungsanforderungen

Amazon S3 unterstützt drei Arten von serverseitiger Verschlüsselung:

- Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3)
- Serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln (-) SSE KMS
- Serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE

Stellen Sie unter Berücksichtigung Ihrer Verschlüsselungseinstellungen sicher, dass die folgenden Berechtigungen erfüllt sind:

- SSE-S3 — Es sind keine zusätzlichen Berechtigungen erforderlich.
- SSE- KMS (mit einem vom Kunden verwalteten Schlüssel) — Um Objekte hochzuladen, AWS KMS key ist die `kms:GenerateDataKey` Berechtigung für das erforderlich. Um Objekte herunterzuladen und mehrteilige Uploads von Objekten durchzuführen, ist die `kms:Decrypt` Genehmigung für den KMS Schlüssel erforderlich.
- SSE- KMS (mit einem Von AWS verwalteter Schlüssel) — Der Anforderer muss aus demselben Konto stammen, dem der Schlüssel gehört. `aws/s3` KMS Der Anforderer muss außerdem über die richtigen Amazon-S3-Berechtigungen verfügen, um auf das Objekt zugreifen zu können.
- SSE-C (mit einem vom Kunden bereitgestellten Schlüssel) — Es sind keine zusätzlichen Berechtigungen erforderlich. Sie können die Bucket-Richtlinie so konfigurieren, dass eine serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln für Objekte in Ihrem Bucket erforderlich und beschränkt ist.

Wenn das Objekt mit einem vom Kunden verwalteten Schlüssel verschlüsselt ist, stellen Sie sicher, dass die KMS Schlüsselrichtlinie es Ihnen ermöglicht, die `kms:GenerateDataKey` `kms:Decrypt` OR-Aktionen auszuführen. Anweisungen zur Überprüfung Ihrer KMS Schlüsselrichtlinie finden Sie unter Schlüsselrichtlinie anzeigen im AWS Key Management Service Entwicklerhandbuch.

S3-Einstellungen für die Objektsperre

Wenn in Ihrem Bucket die S3-Objektsperre aktiviert ist und das Objekt durch einen Aufbewahrungszeitraum oder eine rechtliche Aufbewahrungspflicht geschützt ist, gibt Amazon

S3 beim Versuch, das Objekt zu löschen, einen Fehler aufgrund einer Zugriffsverweigerung (403 Forbidden) zurück.

So überprüfen Sie, ob für den Bucket eine Objektsperre aktiviert ist

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie überprüfen möchten.
4. Wählen Sie die Registerkarte Eigenschaften aus.
5. Scrollen Sie nach unten zum Abschnitt Objektsperre. Überprüfen Sie, ob die Einstellung für Objektsperre Aktiviert oder Deaktiviert lautet.

Um festzustellen, ob das Objekt durch einen Aufbewahrungszeitraum oder eine rechtliche Aufbewahrungspflicht geschützt ist, zeigen Sie die Sperrinformationen für Ihr Objekt an.

Wenn das Objekt durch einen Aufbewahrungszeitraum oder eine rechtliche Aufbewahrungspflicht geschützt ist, überprüfen Sie Folgendes:

- Wenn die Objektversion durch den Compliance-Aufbewahrungsmodus geschützt ist, gibt es keine Möglichkeit, sie dauerhaft zu löschen. Eine permanente `DELETE` Anfrage von einem beliebigen Anforderer, einschließlich des AWS-Konto Root-Benutzers, führt zu der Fehlermeldung Access Denied (403 Forbidden). Beachten Sie außerdem, dass Amazon S3 eine Löschmarkierung für das Objekt erstellt, wenn Sie eine `DELETE`-Anforderung für ein Objekt einreichen, das durch den Compliance-Aufbewahrungsmodus geschützt ist.
- Wenn die Objektversion durch den Governance-Aufbewahrungsmodus geschützt ist und Sie über die Berechtigung `s3:BypassGovernanceRetention` verfügen, können Sie den Schutz umgehen und die Version dauerhaft löschen. Weitere Informationen finden Sie unter Umgehen des Governance-Modus.
- Wenn die Objektversion durch eine rechtliche Aufbewahrungspflicht geschützt ist, kann eine permanente `DELETE`-Anforderung zu einem Fehler aufgrund einer Zugriffsverweigerung (403 Forbidden) führen. Um die Objektversion dauerhaft zu löschen, müssen Sie die rechtliche Aufbewahrungspflicht für die Objektversion aufheben. Um eine rechtliche Aufbewahrungspflicht aufzuheben, benötigen Sie die Berechtigung `s3:PutObjectLegalHold`. Weitere Informationen zum Aufheben einer rechtlichen Aufbewahrungspflicht finden Sie unter Konfigurieren von S3 Object Lock.

VPCEndpunktrichtlinien

Wenn Sie über einen Virtual Private Cloud (VPC) -Endpunkt auf Amazon S3 zugreifen, stellen Sie sicher, dass die VPC Endpunktrichtlinie Sie nicht am Zugriff auf Ihre Amazon S3 S3-Ressourcen hindert. Standardmäßig erlaubt die VPC Endpunktrichtlinie alle Anfragen an Amazon S3. Sie können die VPC Endpunktrichtlinie auch so konfigurieren, dass bestimmte Anfragen eingeschränkt werden. Informationen dazu, wie Sie Ihre VPC Endpunktrichtlinie überprüfen können, finden Sie in den folgenden Ressourcen:

- the section called “Zugriff aufgrund einer VPC Endpunktrichtlinie verweigert — implizite Ablehnung”
- the section called “Zugriff aufgrund einer VPC Endpunktrichtlinie verweigert — explizite Ablehnung”
- Steuern Sie den Zugriff auf VPC Endgeräte mithilfe der Endpunktrichtlinien im Handbuch AWS PrivateLink

AWS Organizations Richtlinien

Wenn Sie zu einer Organisation AWS-Konto gehören, können AWS Organizations Richtlinien Sie daran hindern, auf Amazon S3 S3-Ressourcen zuzugreifen. Standardmäßig blockieren AWS Organizations Richtlinien keine Anfragen an Amazon S3. Stellen Sie jedoch sicher, dass Ihre AWS Organizations Richtlinien nicht so konfiguriert wurden, dass sie den Zugriff auf S3-Buckets blockieren. Anweisungen zur Überprüfung Ihrer AWS Organizations Richtlinien finden Sie in den folgenden Ressourcen:

- the section called “Zugriffsverweigerung aufgrund einer Service-Kontrollrichtlinie – implizite Verweigerung”
- the section called “Zugriffsverweigerung aufgrund einer Service-Kontrollrichtlinie – explizite Verweigerung”
- the section called “Der Zugriff wurde aufgrund einer Ressourcensteuerungsrichtlinie verweigert — explizite Ablehnung”
- Auflistung aller Richtlinien im AWS Organizations Benutzerhandbuch

Wenn Sie Ihre Bucket-Richtlinie für ein Mitgliedskonto falsch konfiguriert haben, um allen Benutzern den Zugriff auf Ihren S3-Bucket zu verweigern, können Sie den Bucket außerdem entsperren, indem Sie eine privilegierte Sitzung für das Mitgliedskonto in startenIAM. Sobald Sie eine privilegierte Sitzung gestartet haben, können Sie die falsch konfigurierte Bucket-Richtlinie löschen, um wieder Zugriff auf den Bucket zu erhalten. Weitere Informationen finden Sie im AWS Identity and Access

Management Benutzerhandbuch unter Ausführen einer privilegierten Aufgabe für ein AWS Organizations Mitgliedskonto.

Zugriffspunkteinstellungen

Wenn Sie bei Anforderungen über Amazon-S3-Zugriffspunkte einen Fehler aufgrund einer Zugriffsverweigerung (403 Forbidden) erhalten, müssen Sie möglicherweise Folgendes überprüfen:

- Die Konfigurationen Ihrer Zugriffspunkte
- Die IAM Benutzerrichtlinie, die für Ihre Access Points verwendet wird
- Die Bucket-Richtlinie, die zur Verwaltung oder Konfiguration Ihrer kontoübergreifenden Zugriffspunkte verwendet wird

Zugriffspunktkonfigurationen und -richtlinien

- Wenn Sie einen Zugriffspunkt erstellen, können Sie wählen, ob Sie das Internet oder VPCals Netzwerkquelle angeben möchten. Wenn der Netzwerkursprung auf VPC Nur gesetzt ist, lehnt Amazon S3 alle Anfragen an den Access Point ab, die nicht von dem angegebenen stammenVPC. Informationen zum Überprüfen des Netzwerkursprungs Ihres Zugriffspunkts finden Sie unter Erstellen von Zugriffspunkten, die auf eine Virtual Private Cloud beschränkt sind.
- Mit Zugriffspunkten können Sie auch benutzerdefinierte Block-Public-Access-Einstellungen konfigurieren, die ähnlich wie die Block-Public-Access-Einstellungen auf Bucket- oder Kontoebene funktionieren. Informationen zu Ihren benutzerdefinierten Block-Public-Access-Einstellungen finden Sie unter Verwalten des öffentlichen Zugriffs auf Zugriffspunkte.
- Um mithilfe von Access Points erfolgreiche Anfragen an Amazon S3 zu stellen, stellen Sie sicher, dass der Anforderer über die erforderlichen IAM Berechtigungen verfügt. Weitere Informationen finden Sie unter Konfiguration von IAM Richtlinien für die Verwendung von Access Points.
- Wenn die Anforderung kontoübergreifende Zugriffspunkte umfasst, stellen Sie sicher, dass der Bucket-Eigentümer die Bucket-Richtlinie aktualisiert hat, um Anforderungen vom Zugriffspunkt zu autorisieren. Weitere Informationen finden Sie unter Erteilen von Berechtigungen für kontoübergreifende Zugriffspunkte.

Wenn der Fehler Zugriff verweigert (403 Forbidden) weiterhin besteht, nachdem Sie alle Artikel in diesem Thema überprüft haben, rufen Sie Ihre Amazon S3 S3-Anforderungs-ID ab und wenden Sie sich an uns, um AWS Support weitere Informationen zu erhalten.

AWS verwaltete Richtlinien für Amazon S3

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird AWS. AWS Verwaltete Richtlinien sind so konzipiert, dass sie Berechtigungen für viele gängige Anwendungsfälle bereitstellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie allen AWS Kunden zur Verfügung stehen. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie kundenverwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service wird oder neue API Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie im IAMBenutzerhandbuch unter AWS Verwaltete Richtlinien.

AWS verwaltete Richtlinie: AmazonS3FullAccess

Sie können die `AmazonS3FullAccess` Richtlinie an Ihre IAM Identitäten anhängen. Diese Richtlinie gewährt Berechtigungen, die vollen Zugriff auf Amazon S3 ermöglichen.

Die Berechtigungen für diese Richtlinie finden Sie unter AmazonS3FullAccessin der AWS Management Console.

AWS verwaltete Richtlinie: AmazonS3ReadOnlyAccess

Sie können die `AmazonS3ReadOnlyAccess` Richtlinie an Ihre IAM Identitäten anhängen. Diese Richtlinie gewährt Berechtigungen, die einen schreibgeschützten Zugriff auf Amazon S3 erlauben.

Die Berechtigungen für diese Richtlinie finden Sie unter AmazonS3ReadOnlyAccessin der AWS Management Console.

AWS verwaltete Richtlinie: AmazonS3ObjectLambdaExecutionRolePolicy

Stellt AWS Lambda Funktionen die erforderlichen Berechtigungen zum Senden von Daten an S3 Object Lambda bereit, wenn Anfragen an einen S3 Object Lambda-Zugriffspunkt gestellt werden. Gewährt Lambda außerdem Berechtigungen zum Schreiben in CloudWatch Amazon-Logs.

Die Berechtigungen für diese Richtlinie finden Sie unter AmazonS3ObjectLambdaExecutionRolePolicy im AWS Management Console.

AWS verwaltete Richtlinie: S3UnlockBucketPolicy

Wenn Sie Ihre Bucket-Richtlinie für ein Mitgliedskonto falsch konfiguriert haben, um allen Benutzern den Zugriff auf Ihren S3-Bucket zu verweigern, können Sie diese AWS verwaltete Richtlinie (`S3UnlockBucketPolicy`) verwenden, um den Bucket zu entsperren. Weitere Informationen zum Entfernen einer falsch konfigurierten Bucket-Richtlinie, die allen Principals den Zugriff auf einen Amazon S3 S3-Bucket verweigert, finden Sie unter Ausführen einer privilegierten Aufgabe für ein AWS Organizations Mitgliedskonto im AWS Identity and Access Management Benutzerhandbuch.

Amazon S3 S3-Updates für AWS verwaltete Richtlinien

Sehen Sie sich Details zu Aktualisierungen der AWS verwalteten Richtlinien für Amazon S3 an, seit dieser Service begonnen hat, diese Änderungen zu verfolgen.

Änderung	Beschreibung	Datum
Amazon S3 `S3UnlockBucketPolicy` hinzugefügt	Amazon S3 hat eine neue AWS verwaltete Richtlinie hinzugefügt, mit der ein Bucket entsperrt und eine falsch konfigurierte Bucket-Richtlinie entfernt werden `S3UnlockBucketPolicy` soll, die allen Principals den Zugriff auf einen Amazon S3 S3-Bucket verweigert.	1. November 2024
Amazon S3 hat die Beschreibungsberechtigungen zu `AmazonS3ReadOnlyAccess` hinzugefügt.	Amazon S3 hat die `s3:Describe*` -Berechtigungen zu `AmazonS3ReadOnlyAccess` hinzugefügt.	11. August 2023
Amazon S3 hat S3 Objekt Lambda-Berechtigungen zu	Amazon S3 hat die `AmazonS3FullAccess` -	27. September 2021

Änderung	Beschreibung	Datum
`AmazonS3FullAccess` und `AmazonS3ReadOnlyAccess` hinzugefügt	und `AmazonS3ReadOnlyAccess` -Richtlinien zur Einbeziehung von Berechtigungen für S3-Objekt-Lambda aktualisiert.	
Amazon S3 `AmazonS3ObjectLambdaExecutionRolePolicy` hinzugefügt	Amazon S3 hat eine neue AWS verwaltete Richtlinie hinzugefügt`AmazonS3ObjectLambdaExecutionRolePolicy` , die Lambda-Funktionen Berechtigungen zur Interaktion mit S3 Object Lambda und zum Schreiben in Protokolle gewährt. CloudWatch	18. August 2021
Amazon S3 hat mit der Verfolgung von Änderungen begonnen	Amazon S3 hat damit begonnen, Änderungen für seine AWS verwalteten Richtlinien nachzuverfolgen.	18. August 2021

Verwaltung des Zugriffs auf gemeinsam genutzte Datensätze mit Access Points

Amazon S3 S3-Zugriffspunkte vereinfachen den Datenzugriff für alle AWS Service- oder Kundenanwendungen, die Daten in S3 speichern. Zugriffspunkte sind benannte Netzwerkendpunkte, die Buckets zugeordnet sind, mit denen Sie S3-Objekt-Vorgänge ausführen können, z. B. `GetObject` und `PutObject`. Jeder Zugriffspunkt verfügt über unterschiedliche Berechtigungen und Netzwerkkontrollen, die S3 für alle Anforderungen anwendet, die über diesen Zugriffspunkt eingehen. Jeder Zugriffspunkt erzwingt eine benutzerdefinierte Zugriffspunktrichtlinie, die in Verbindung mit der Bucket-Richtlinie funktioniert, die dem zugrunde liegenden Bucket zugeordnet ist. Sie können jeden Access Point so konfigurieren, dass er nur Anfragen von einer virtuellen privaten Cloud (VPC) akzeptiert, um den Amazon S3 S3-Datenzugriff auf ein privates Netzwerk einzuschränken.

Sie können auch benutzerdefinierte Block Public Access-Einstellungen für jeden Zugriffspunkt konfigurieren.

> **Note**
>
> - Sie können Zugriffspunkte nur zum Ausführen von Vorgängen an Objekten verwenden. Sie können mit Zugriffspunkten keine anderen Amazon S3-Vorgänge auszuführen, z. B. das Ändern oder Löschen von Buckets. Eine vollständige Liste der S3-Vorgänge, die Zugriffspunkte unterstützen, finden Sie unter Kompatibilität des Access Points mit Diensten AWS.
> - Access Points funktionieren mit einigen, aber nicht allen AWS Diensten und Funktionen. Beispielsweise können Sie die regionsübergreifende Replikation nicht so konfigurieren, dass sie über einen Zugriffspunkt ausgeführt wird. Eine vollständige Liste der AWS Dienste, die mit S3-Zugriffspunkten kompatibel sind, finden Sie unterKompatibilität des Access Points mit Diensten AWS.

In diesem Abschnitt wird erläutert, wie Sie mit Amazon S3-Zugriffspunkten arbeiten. Hinweise zum Arbeiten mit Buckets finden Sie unter Bucket-Übersicht. Weitere Informationen zur Arbeit mit Objekten finden Sie unter Übersicht über Amazon-S3-Objekte.

Themen

- Einschränkungen von Access Points
- Konfiguration von IAM Richtlinien für die Verwendung von Access Points
- Erstellen von Zugriffspunkten
- Verwenden von Zugriffspunkten

Einschränkungen von Access Points

Es gelten die folgenden Einschränkungen und Beschränkungen für Amazon S3-Zugriffspunkte:

- Jeder Zugriffspunkt ist genau einem Bucket zugeordnet, den Sie beim Erstellen des Zugriffspunkts angeben müssen. Nachdem Sie einen Zugriffspunkt erstellt haben, können Sie ihn keinem anderen Bucket zuordnen. Sie können jedoch einen Zugriffspunkt löschen und dann einen anderen mit demselben Namen erstellen. Diesen neuen Zugriffspunkt können Sie dann einem anderen Bucket zuordnen.

- Namen von Zugriffspunkten müssen bestimmte Bedingungen erfüllen. Weitere Informationen zur Benennung von Zugangspunkten finden Sie unter Regeln zur Benennung von Amazon S3-Zugriffspunkten.
- Nachdem Sie einen Access Point erstellt haben, können Sie seine Virtual Private Cloud (VPC) - Konfiguration nicht mehr ändern.
- Zugriffspunkt-Richtlinien sind auf eine Größe von 20 KB beschränkt.
- Sie können maximal 10.000 Access Points AWS-Konto pro Region erstellen. Wenn Sie mehr als 10 000 Zugriffspunkte für ein einzelnes Konto in einer Region benötigen, können Sie eine Erhöhung der Service Quotas beantragen. Weitere Informationen zu Service Quotas und zum Beantragen einer Erhöhung finden Sie unter AWS Service Quotas in der Allgemeine AWS-Referenz.
- Wenn Sie über mehr als 1.000 Access Points verfügen, können Sie in der Amazon S3 S3-Konsole nicht anhand des Namens nach einem Access Point suchen. AWS-Regionen
- Sie können einen Zugriffspunkt nicht als Ziel für die Replikation in S3 verwenden. Weitere Informationen zur Replikation finden Sie unter Objekte innerhalb und zwischen Regionen replizieren.
- Sie können S3-Zugriffspunkt-Aliase nicht als Quelle oder Ziel für Move-Operationen in der Amazon S3 S3-Konsole verwenden.
- Sie können Access Points nur ansprechen, indem Sie virtual-host-style URLs Weitere Hinweise virtual-host-style zur Adressierung finden Sie unterZugreifen auf einen Amazon S3 S3-Bucket.
- APIVorgänge, die die Funktionalität des Access Points steuern (z. B. `PutAccessPoint` und`GetAccessPointPolicy`), unterstützen keine kontenübergreifenden Anrufe.
- Sie müssen AWS Signature Version 4 verwenden, wenn Sie Anfragen an einen Access Point mit dem REST APIs stellen. Weitere Informationen zur Authentifizierung von Anfragen finden Sie unter Authentifizieren von Anfragen (AWS Signature Version 4) in der Amazon Simple Storage Service API Reference.
- Access Points unterstützen nur Anfragen über. HTTPS Amazon S3 antwortet automatisch mit einer HTTP Weiterleitung auf alle AnfragenHTTP, auf die Sie die Anfrage aktualisieren möchtenHTTPS.
- Zugriffspunkte unterstützen keinen anonymen Zugriff.
- Mit den kontoübergreifenden Zugriffspunkten erhalten Sie jedoch erst dann Zugriff auf Daten, wenn Ihnen vom Bucket-Eigentümer die entsprechenden Berechtigungen erteilt wurden. Der Bucket-Eigentümer behält immer die ultimative Kontrolle über den Datenzugriff und muss die Bucket-Richtlinie aktualisieren, um Anforderungen vom kontoübergreifenden Zugriffspunkt zu

autorisieren. Eine Bucket-Beispielrichtlinie finden Sie unter Konfiguration von IAM Richtlinien für die Verwendung von Access Points.

- Wenn Sie in der Amazon-S3-Konsole einen kontoübergreifenden Zugriffspunkt anzeigen, wird in der Spalte Zugriff Unbekannt angezeigt. Die Amazon-S3-Konsole kann nicht feststellen, ob öffentlicher Zugriff für den zugehörigen Bucket und die zugehörigen Objekte gewährt wurde. Sofern Sie keine öffentliche Konfiguration für einen bestimmten Anwendungsfall benötigen, empfehlen wir Ihnen und dem Bucket-Besitzer, den gesamten öffentlichen Zugriff auf den Access Point und den Bucket zu blockieren. Weitere Informationen finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

Konfiguration von IAM Richtlinien für die Verwendung von Access Points

Amazon S3 Access Points unterstützen Ressourcenrichtlinien AWS Identity and Access Management (IAM), mit denen Sie die Nutzung des Access Points nach Ressourcen, Benutzern oder anderen Bedingungen steuern können. Damit eine Anwendung oder ein Benutzer über einen Zugriffspunkt auf Objekte zugreifen kann, müssen sowohl der Zugriffspunkt als auch der zugrunde liegende Bucket die Anforderung zulassen.

> **Important**
>
> Das Hinzufügen eines S3-Zugriffspunkts zu einem Bucket ändert das Verhalten des Buckets nicht, wenn auf den Bucket direkt über den Namen des Buckets oder den Amazon-Ressourcennamen (ARN) zugegriffen wird. Alle vorhandenen Vorgänge für den Bucket funktionieren weiterhin wie zuvor. Einschränkungen, die Sie in eine Zugriffspunktrichtlinie einschließen, gelten nur für Anforderungen, die über diesen Zugriffspunkt eingehen.

Wenn Sie IAM Ressourcenrichtlinien verwenden, stellen Sie sicher, dass Sie Sicherheitswarnungen, Fehler, allgemeine Warnungen und Vorschläge AWS Identity and Access Management Access Analyzer vor dem Speichern Ihrer Richtlinie behoben haben. IAMAccess Analyzer führt Richtlinienprüfungen durch, um Ihre Richtlinie anhand der IAM Richtliniengrammatik und der bewährten Methoden zu überprüfen. Diese Prüfungen generieren Ergebnisse und bieten Empfehlungen, die Sie beim Erstellen von Richtlinien unterstützen, die funktionsfähig sind und den bewährten Methoden für Sicherheit entsprechen.

Weitere Informationen zur Überprüfung von Richtlinien mithilfe von IAM Access Analyzer finden Sie unter IAMAccess Analyzer-Richtlinienvalidierung im IAMBenutzerhandbuch. Eine Liste der

Warnungen, Fehler und Vorschläge, die von Access Analyzer zurückgegeben werden, finden Sie unter IAM IAMAccess Analyzer-Referenz zur Richtlinienüberprüfung.

Beispiele von -Zugriffspunktrichtlinien

Die folgenden Beispiele zeigen, wie IAM Richtlinien zur Steuerung von Anfragen über einen Access Point erstellt werden.

> **Note**
>
> Berechtigungen, die in einer Zugriffspunktrichtlinie erteilt werden, sind nur wirksam, wenn der zugrunde liegende Bucket auch denselben Zugriff zulässt. Sie können dies auf zwei Arten erreichen:
>
> 1. (Empfohlen) Delegieren Sie die Zugriffssteuerung vom Bucket an den Zugriffspunkt, wie unter Delegieren der Zugangskontrolle an Zugriffspunkte beschrieben.
> 2. Fügen Sie der Richtlinie des zugrunde liegenden Buckets dieselben Berechtigungen hinzu, die in der Zugriffspunktrichtlinie enthalten sind. Beispiel 1 für eine Zugriffspunktrichtlinie veranschaulicht, wie die zugrunde liegende Bucket-Richtlinie geändert wird, damit der erforderliche Zugriff gewährt wird.

Example 1 – Erteilung der Zugriffspunktberechtigung

Die folgende Zugriffspunktrichtlinie gewährt IAM Benutzern `GET` und *`Jane`* Benutzerkonten *`123456789012`* Berechtigungen für `PUT` Objekte mit dem Präfix *`Jane/`* über den Zugriffspunkt *`my-access-point`* im Konto *`123456789012`*.

```
{
    "Version":"2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/Jane"
        },
        "Action": ["s3:GetObject", "s3:PutObject"],
        "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-access-point/
object/Jane/*"
    }]
```

```
}
```

> **Note**
>
> Damit die Zugangspunkt-Richtlinie effektiv Zugriff gewährt *Jane*, muss der zugrundeliegende Bucket auch denselben Zugriff auf ermöglichen *Jane*. Sie können die Zugriffskontrolle vom Bucket an den Access Point delegieren, wie unter beschriebenDelegieren der Zugangskontrolle an Zugriffspunkte. Oder Sie können dem zugrunde liegenden Bucket die folgende Richtlinie hinzufügen, um Jane die erforderlichen Berechtigungen zu erteilen. Beachten Sie, dass sich der `Resource`-Eintrag zwischen den Zugriffspunkt- und Bucket-Richtlinien unterscheidet.
>
> ```
> {
>     "Version": "2012-10-17",
>     "Statement": [
>     {
>         "Effect": "Allow",
>         "Principal": {
>             "AWS": "arn:aws:iam::123456789012:user/Jane"
>         },
>         "Action": ["s3:GetObject", "s3:PutObject"],
>         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/Jane/*"
>     }]
> }
> ```

Example 2 – Zugriffspunktrichtlinie mit Tag-Bedingung

Die folgende Zugriffspunktrichtlinie gewährt IAM dem Benutzer *Mateo* auf Rechnung *123456789012* Berechtigungen für GET Objekte über den Access Point *my-access-point* auf dem Konto *123456789012* die den Tag-Schlüssel haben *data* gesetzt mit einem Wert von *finanzieren*.

```
{
    "Version":"2012-10-17",
    "Statement": [
    {
        "Effect":"Allow",
        "Principal" : {
            "AWS": "arn:aws:iam::123456789012:user/Mateo"
        },
```

```
        "Action":"s3:GetObject",
        "Resource" : "arn:aws:s3:us-west-2:123456789012:accesspoint/my-access-point/
object/*",
        "Condition" : {
            "StringEquals": {
                "s3:ExistingObjectTag/data": "finance"
            }
        }
    }]
}
```

Example 3 – Zugriffspunktrichtlinie, die eine Bucket-Auflistung ermöglicht

Die folgende Zugriffspunktrichtlinie erlaubt IAM Benutzern `Arnav` den Zugriff auf das Konto *123456789012* Berechtigung zum Anzeigen der Objekte, die sich in dem Bucket befinden, der dem Access Point zugrunde liegt *my-access-point* auf dem Konto *123456789012*.

```
{
    "Version":"2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::123456789012:user/Arnav"
        },
        "Action": "s3:ListBucket",
        "Resource": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-access-point"
    }]
}
```

Example 4 – Service-Kontrollrichtlinie

Gemäß der folgenden Dienststeuerungsrichtlinie müssen alle neuen Access Points mit einem Netzwerkursprung aus einer virtuellen privaten Cloud (VPC) erstellt werden. Mit dieser Richtlinie können Benutzer in Ihrer Organisation keine neuen Zugriffspunkte erstellen, auf die über das Internet zugegriffen werden kann.

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

```
        "Effect": "Deny",
        "Action": "s3:CreateAccessPoint",
        "Resource": "*",
        "Condition": {
            "StringNotEquals": {
                "s3:AccessPointNetworkOrigin": "VPC"
            }
        }
    }]
}
```

Example 5 — Bucket-Richtlinie zur Beschränkung von S3-Vorgängen auf VPC Netzwerkquellen

Die folgende Bucket-Richtlinie beschränkt den Zugriff auf alle S3-Objektoperationen für den Bucket *amzn-s3-demo-bucket* auf Zugriffspunkte mit VPC Netzwerkursprung.

> **Important**
>
> Bevor Sie eine Anweisung wie in diesem Beispiel gezeigt verwenden, stellen Sie sicher, dass Sie keine Funktionen verwenden müssen, die von Zugriffspunkten nicht unterstützt werden, z. B. regionsübergreifende Replikation.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:BypassGovernanceRetention",
                "s3:DeleteObject",
                "s3:DeleteObjectTagging",
                "s3:DeleteObjectVersion",
                "s3:DeleteObjectVersionTagging",
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:GetObjectLegalHold",
                "s3:GetObjectRetention",
                "s3:GetObjectTagging",
                "s3:GetObjectVersion",
```

```
                "s3:GetObjectVersionAcl",
                "s3:GetObjectVersionTagging",
                "s3:ListMultipartUploadParts",
                "s3:PutObject",
                "s3:PutObjectAcl",
                "s3:PutObjectLegalHold",
                "s3:PutObjectRetention",
                "s3:PutObjectTagging",
                "s3:PutObjectVersionAcl",
                "s3:PutObjectVersionTagging",
                "s3:RestoreObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "StringNotEquals": {
                    "s3:AccessPointNetworkOrigin": "VPC"
                }
            }
        }
    ]
}
```

Bedingungsschlüssel

S3-Zugriffspunkte verfügen über Bedingungsschlüssel, die Sie in IAM Richtlinien verwenden können, um den Zugriff auf Ihre Ressourcen zu kontrollieren. Die folgenden Bedingungsschlüssel stellen nur einen Teil einer IAM Richtlinie dar. Vollständige Richtlinienbeispiele finden Sie unter Beispiele von - Zugriffspunktrichtlinien, the section called “Delegieren der Zugangskontrolle an Zugriffspunkte” und the section called “Erteilen von Berechtigungen für kontoübergreifende Zugriffspunkte”.

`s3:DataAccessPointArn`

Dieses Beispiel zeigt eine Zeichenfolge, die Sie für den Abgleich auf einem Access Point verwenden könnenARN. Das folgende Beispiel entspricht allen Access Points für AWS-Konto *123456789012* in der Region *us-west-2*:

```
"Condition" : {
    "StringLike": {
        "s3:DataAccessPointArn": "arn:aws:s3:us-west-2:123456789012:accesspoint/*"
    }
}
```

`s3:DataAccessPointAccount`

Dieses Beispiel zeigt einen Zeichenfolgenoperator, mit dem Sie die Konto-ID des Besitzers eines Zugriffspunkts abgleichen können. Das folgende Beispiel entspricht allen Access Points, die dem gehören AWS-Konto *123456789012*.

```
"Condition" : {
    "StringEquals": {
        "s3:DataAccessPointAccount": "123456789012"
    }
}
```

`s3:AccessPointNetworkOrigin`

Dieses Beispiel zeigt einen Zeichenfolgenoperator, den Sie verwenden können, um für den Netzwerkursprung entweder `Internet` oder VPC abzugleichen. Das folgende Beispiel entspricht nur Access Points mit einem VPC Ursprung.

```
"Condition" : {
    "StringEquals": {
        "s3:AccessPointNetworkOrigin": "VPC"
    }
}
```

Weitere Informationen zur Verwendung von Bedingungsschlüsseln mit Amazon S3 finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

Delegieren der Zugangskontrolle an Zugriffspunkte

Sie können die Zugriffssteuerung für einen Bucket an die Zugriffspunkte des Buckets delegieren. Die folgende Bucket-Beispielrichtlinie ermöglicht Vollzugriff auf alle Zugriffspunkte, die dem Konto des Bucket-Eigentümers gehören. Somit wird der gesamte Zugriff auf diesen Bucket durch die an seine Zugriffspunkte angehängten Richtlinien gesteuert. Wir empfehlen, Ihre Buckets auf diese Weise für alle Anwendungsfälle zu konfigurieren, die keinen direkten Zugriff auf den Bucket erfordern.

Example 6 – Bucket-Richtlinie zum Delegieren der Zugriffskontrolle an Standardzugriffspunkte

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS": "*" },
        "Action" : "*",
        "Resource" : [ "Bucket ARN", "Bucket ARN/*"],
        "Condition": {
            "StringEquals" : { "s3:DataAccessPointAccount" : "Bucket owner's account
 ID" }
        }
    }]
}
```

Erteilen von Berechtigungen für kontoübergreifende Zugriffspunkte

Wenn Sie einen Zugriffspunkt für einen Bucket erstellen möchten, der einem anderen Konto gehört, müssen Sie zuerst den Zugriffspunkt erstellen, indem Sie den Bucket-Namen und die Kontobesitzer-ID angeben. Dann muss der Bucket-Eigentümer die Bucket-Richtlinie aktualisieren, um Anfragen vom Zugriffspunkt zu autorisieren. Das Erstellen eines Access Points ähnelt dem Erstellen eines Access Points DNS CNAME insofern, als der Access Point keinen Zugriff auf den Bucket-Inhalt bietet. Der gesamte Bucket-Zugriff wird durch die Bucket-Richtlinie kontrolliert. Die folgende Bucket-Beispielrichtlinie erlaubt `GET`- und `LIST`- Anfragen an den Bucket von einem Zugriffspunkt aus, der einem vertrauenswürdigen AWS-Konto gehört.

Ersetzen *`Bucket ARN`* mit dem ARN des Buckets.

Example 7 — Bucket-Richtlinie, mit der Berechtigungen an einen anderen delegiert werden AWS-Konto

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS": "*" },
        "Action" : ["s3:GetObject","s3:ListBucket"],
        "Resource" : [ "Bucket ARN", "Bucket ARN/*"],
```

```
        "Condition": {
            "StringEquals" : { "s3:DataAccessPointAccount" : "Access point owner's
 account ID" }
        }
    }]
}
```

Erstellen von Zugriffspunkten

Amazon S3 stellt Funktionen zum Erstellen und Verwalten von Zugriffspunkten bereit. Sie können S3-Zugriffspunkte mithilfe von AWS Management Console, AWS Command Line Interface (AWS CLI) AWS SDKs, oder Amazon S3 erstellen RESTAPI.

Standardmäßig können Sie für jedes Ihrer AWS-Konten bis zu 10 000 Zugriffspunkte pro Region erstellen. Wenn Sie mehr als 10 000 Zugriffspunkte für ein einzelnes Konto in einer Region benötigen, können Sie eine Erhöhung der Service Quotas beantragen. Weitere Informationen zu Service Quotas und zum Beantragen einer Erhöhung finden Sie unter AWS Service Quotas in der Allgemeine AWS-Referenz.

> **Note**
>
> Da Sie möglicherweise Ihren Zugriffspunktnamen veröffentlichen möchten, damit andere Benutzer den Zugriffspunkt verwenden können, sollten Sie vertrauliche Informationen im Namen des Zugriffspunkts vermeiden. Die Namen der Access Points werden in einer öffentlich zugänglichen Datenbank veröffentlicht, die als Domain Name System (DNS) bezeichnet wird.

Regeln zur Benennung von Amazon S3-Zugriffspunkten

Ein Zugriffspunktname muss die folgenden Bedingungen erfüllen:

- Muss innerhalb einer einzelnen AWS-Konto Region eindeutig sein
- Muss den DNS Benennungsbeschränkungen entsprechen
- Muss mit einer Zahl oder einem Kleinbuchstaben beginnen
- Muss zwischen 3 und 50 Zeichen lang sein.
- Dürfen nicht mit einem Bindestrich (-) beginnen oder enden

- Dürfen keine Unterstriche (_), Großbuchstaben oder Punkte (.) enthalten
- Kann nicht mit dem Suffix `-s3alias` enden. Dieses Suffix ist für Zugriffspunkt-Aliasnamen reserviert. Weitere Informationen finden Sie unter Verwenden eines Alias im Bucket-Stil für Ihren S3-Bucket-Zugriffspunkt.

Informationen zum Erstellen eines Zugriffspunkts finden Sie in den folgenden Themen.

Themen

- Erstellen eines Zugriffspunkts
- Erstellen von Zugriffspunkten, die auf eine Virtual Private Cloud beschränkt sind
- Verwalten des öffentlichen Zugriffs auf Zugriffspunkte

Erstellen eines Zugriffspunkts

Ein Zugriffspunkt ist genau einem Amazon S3-Bucket zugeordnet. Wenn Sie einen Bucket in Ihrem verwenden möchten AWS-Konto, müssen Sie zuerst einen Bucket erstellen. Weitere Informationen zum Erstellen von Buckets finden Sie unter Erstellen, Konfigurieren und Arbeiten mit Amazon S3-Buckets.

Sie können auch einen kontoübergreifenden Zugriffspunkt erstellen, der mit einem Bucket in einem anderen AWS-Konto verknüpft ist, sofern Sie den Bucket-Namen und die Konto-ID des Bucket-Eigentümers kennen. Wenn Sie kontoübergreifende Zugriffspunkte erstellen, erhalten Sie jedoch erst Zugriff auf Daten im Bucket, wenn Ihnen vom Bucket-Eigentümer die entsprechenden Berechtigungen erteilt wurden. Der Bucket-Eigentümer muss dem Konto des Zugriffspunktbesitzers (Ihr Konto) über die Bucket-Richtlinie Zugriff auf den Bucket gewähren. Weitere Informationen finden Sie unter Erteilen von Berechtigungen für kontoübergreifende Zugriffspunkte.

Standardmäßig können Sie für jedes Ihrer AWS-Konten bis zu 10 000 Zugriffspunkte pro Region erstellen. Wenn Sie mehr als 10 000 Zugriffspunkte für ein einzelnes Konto in einer Region benötigen, können Sie eine Erhöhung der Service Quotas beantragen. Weitere Informationen zu Service Quotas und zum Beantragen einer Erhöhung finden Sie unter AWS Service Quotas in der Allgemeine AWS-Referenz.

Die folgenden Beispiele zeigen, wie Sie einen Access Point mit der AWS CLI und der S3-Konsole erstellen. Weitere Informationen zum Erstellen von Access Points mithilfe von finden Sie unter REST API CreateAccessPointin der Amazon Simple Storage Service API Reference.

Verwenden der S3-Konsole

So erstellen Sie einen Zugriffspunkt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen der aktuell angezeigten Datei aus AWS-Region. Wählen Sie als Nächstes die Region aus, in der Sie einen Access Point erstellen möchten.
3. Wählen Sie im linken Navigationsbereich Zugriffspunkte aus.
4. Wählen Sie auf der Seite Zugriffspunkte die Option Zugriffspunkt erstellen aus.
5. Geben Sie im Feld Name des Zugriffspunkts den gewünschten Namen für den Zugriffspunkt ein. Weitere Benennungen zur Benennung von Zugangspunkten finden Sie unter Regeln zur Benennung von Amazon S3-Zugriffspunkten.
6. Geben Sie unter Bucket-Name den S3-Bucket an, den Sie mit dem Zugriffspunkt verwenden möchten.

   Um in einen Bucket in Ihrem Konto zu verwenden, wählen Sie Bucket in diesem Konto auswählen aus und geben Sie den Namen des Buckets ein oder suchen Sie danach.

   Um einen Bucket in einem anderen Konto zu verwenden AWS-Konto, wählen Sie Spezify a bucket in another account und geben Sie die AWS-Konto ID und den Namen des Buckets ein.

   > **Note**
   >
   > Wenn Sie einen Bucket in einem anderen Bucket verwenden AWS-Konto, muss der Bucket-Besitzer die Bucket-Richtlinie aktualisieren, um Anfragen vom Access Point zu autorisieren. Ein Beispiel für eine Bucket-Richtlinie finden Sie in Beispiel Erteilen von Berechtigungen für kontoübergreifende Zugriffspunkte.

7. Wählen Sie einen Ausgangspunkt des Netzwerks aus. Wenn Sie Virtual Private Cloud (VPC) wählen, geben Sie die VPCID ein, die Sie mit dem Access Point verwenden möchten.

   Weitere Informationen zu Netzwerkursprüngen für Zugangspunkte finden Sie unter Erstellen von Zugriffspunkten, die auf eine Virtual Private Cloud beschränkt sind.
8. Wählen Sie unter Block Public Access settings for this Access Point (Block-Public-Access-Einstellungen für diesen Zugriffspunkt) die Block-Public-Access-Einstellungen, die Sie für den

Zugriffspunkt anwenden möchten. Alle Einstellungen zum Blockieren des öffentlichen Zugriffs sind standardmäßig für neue Zugriffspunkte aktiviert. Wir empfehlen, alle Einstellungen aktiviert zu lassen, es sei denn, Sie wissen, dass Sie eine bestimmte Einstellung deaktivieren müssen.

> **Note**
>
> Nachdem Sie einen Zugriffspunkt erstellt haben, können Sie die Einstellungen zum Blockieren des öffentlichen Zugriffs nicht mehr ändern.

Weitere Informationen zur Verwendung von Amazon S3-Block-Public-Access mit Zugriffspunkten finden Sie unter Verwalten des öffentlichen Zugriffs auf Zugriffspunkte.

9. (Optional) Geben Sie unter Access point policy (Zugriffspunktrichtlinie) – optional die Zugriffspunktrichtlinie an. Beheben Sie vor dem Speichern Ihrer Richtlinie Sicherheitswarnungen, Fehler, allgemeine Warnungen und Vorschläge. Weitere Informationen zum Festlegen einer Zugriffspunktrichtlinie finden Sie unter Beispiele von -Zugriffspunktrichtlinien.
10. Wählen Sie Create access point (Zugriffspunkt erstellen) aus.

Verwenden Sie den AWS CLI

Der folgende Beispielbefehl erstellt einen Access Point mit dem Namen *example-ap* für den Eimer *amzn-s3-demo-bucket* auf dem Konto *111122223333*. Um den Access Point zu erstellen, senden Sie eine Anfrage an Amazon S3, die Folgendes spezifiziert:

- Den Namen des Zugriffspunkts Informationen zu Benennungsregeln finden Sie unter the section called “Regeln zur Benennung von Amazon S3-Zugriffspunkten”.
- Den Name des Buckets, dem Sie den Zugriffspunkt zuweisen möchten
- Die Konto-ID des AWS-Konto Besitzers des Access Points.

```
aws s3control create-access-point --name example-ap --account-id 111122223333 --
bucket amzn-s3-demo-bucket
```

Wenn Sie einen Access Point mithilfe eines Buckets in einem anderen erstellen AWS-Konto, geben Sie den `--bucket-account-id` Parameter an. Der folgende Beispielbefehl erstellt einen Access Point im AWS-Konto *111122223333*, unter Verwendung des Buckets *amzn-s3-demo-bucket2*, das ist in der AWS-Konto *444455556666*.

```
aws s3control create-access-point --name example-ap --account-id 111122223333 --
bucket amzn-s3-demo-bucket --bucket-account-id 444455556666
```

Erstellen von Zugriffspunkten, die auf eine Virtual Private Cloud beschränkt sind

Wenn Sie einen Access Point erstellen, können Sie wählen, ob der Access Point über das Internet zugänglich ist, oder Sie können angeben, dass alle über diesen Access Point gestellten Anfragen aus einer bestimmten virtuellen privaten Cloud stammen müssen (VPC). Ein Zugangspunkt, der über das Internet zugänglich ist, soll einen Netzwerkursprung von `Internet` haben. Er kann von überall im Internet verwendet werden, vorbehaltlich anderer Zugriffsbeschränkungen für den Zugriffspunkt, den zugrunde liegenden Bucket und die zugehörige Ressourcen, z. B. die angeforderten Objekte. Ein Access Point, auf den nur von einem bestimmten Access Point aus zugegriffen werden kannVPC, VPC hat einen Netzwerkursprung von, und Amazon S3 lehnt alle Anfragen an den Access Point ab, die nicht von diesem VPC stammen.

> **Important**
>
> Sie können den Netzwerkursprung eines Zugriffspunkts nur angeben, wenn Sie den Zugriffspunkt erstellen. Nachdem Sie den Zugriffspunkt erstellt haben, können Sie seinen Netzwerkursprung nicht mehr ändern.

Um einen Access Point auf den VPC reinen Zugriff zu beschränken, fügen Sie den `VpcConfiguration` Parameter der Anfrage zur Erstellung des Access Points bei. Im `VpcConfiguration` Parameter geben Sie die VPC ID an, mit der Sie den Access Point verwenden möchten. Wenn eine Anfrage über den Access Point gestellt wird, muss die Anfrage von dem stammen, VPC andernfalls lehnt Amazon S3 sie ab.

Sie können den Netzwerkursprung eines Access Points mithilfe von AWS CLI AWS SDKs, oder abrufen RESTAPIs. Wenn für einen Access Point eine VPC Konfiguration angegeben ist, lautet sein NetzwerkursprungVPC. Andernfalls ist der Netzwerkursprung des Zugriffspunkts `Internet`.

Example

Beispiel: Erstellen Sie einen Access Point, der auf den VPC Zugriff beschränkt ist

Im folgenden Beispiel wird ein Access Point mit dem Namen `example-vpc-ap` Bucket *amzn-s3-demo-bucket* in Account erstellt`123456789012`, der den Zugriff nur über den ermöglicht

`vpc-1a2b3c`VPC. Das Beispiel überprüft dann, ob der neue Zugriffspunkt den Netzwerkursprung VPC hat.

AWS CLI

```
aws s3control create-access-point --name example-vpc-ap --account-id 123456789012 --
bucket amzn-s3-demo-bucket --vpc-configuration VpcId=vpc-1a2b3c
```

```
aws s3control get-access-point --name example-vpc-ap --account-id 123456789012

{
    "Name": "example-vpc-ap",
    "Bucket": "amzn-s3-demo-bucket",
    "NetworkOrigin": "VPC",
    "VpcConfiguration": {
        "VpcId": "vpc-1a2b3c"
    },
    "PublicAccessBlockConfiguration": {
        "BlockPublicAcls": true,
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "RestrictPublicBuckets": true
    },
    "CreationDate": "2019-11-27T00:00:00Z"
}
```

Um einen Access Point mit a zu verwendenVPC, müssen Sie die Zugriffsrichtlinie für Ihren VPC Endpunkt ändern. VPCEndpunkte ermöglichen den Datenfluss von Ihnen VPC zu Amazon S3. Sie verfügen über Zugriffskontrollrichtlinien, die VPC kontrollieren, wie Ressourcen innerhalb von mit Amazon S3 interagieren dürfen. Anfragen von Ihnen VPC an Amazon S3 werden nur dann über einen Access Point erfolgreich ausgeführt, wenn die VPC Endpunktrichtlinie Zugriff sowohl auf den Access Point als auch auf den zugrunde liegenden Bucket gewährt.

> **Note**
>
> Um den Zugriff auf Ressourcen nur innerhalb eines zu ermöglichenVPC, stellen Sie sicher, dass Sie eine private, gehostete Zone für Ihren VPC Endpunkt einrichten. Um eine private gehostete Zone zu verwenden, ändern Sie Ihre VPC Einstellungen so, dass die

VPCNetzwerkattribute `enableDnsHostnames` und auf eingestellt `enableDnsSupport` sind`true`.

In der folgenden Beispiel-Richtlinienanweisung wird ein VPC Endpunkt so konfiguriert, dass Aufrufe `GetObject` für einen Bucket mit dem Namen `awsexamplebucket1` und einen Access Point mit dem Namen `example-vpc-ap` möglich sind.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Principal": "*",
        "Action": [
            "s3:GetObject"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::awsexamplebucket1/*",
            "arn:aws:s3:us-west-2:123456789012:accesspoint/example-vpc-ap/object/*"
        ]
    }]
}
```

Note

Die `"Resource"` Deklaration in diesem Beispiel verwendet einen Amazon-Ressourcennamen (ARN), um den Access Point zu spezifizieren. Weitere Informationen zum Access Point ARNs finden Sie unterVerwenden von Zugriffspunkten.

Weitere Informationen zu VPC Endpunktrichtlinien finden Sie unter Verwenden von Endpunktrichtlinien für Amazon S3 im VPCBenutzerhandbuch.

Verwalten des öffentlichen Zugriffs auf Zugriffspunkte

Amazon S3-Zugriffspunkte unterstützen unabhängige Block Public Access-Einstellungen für jeden Zugriffspunkt. Wenn Sie einen Zugriffspunkt erstellen, können Sie die Block Public Access-Einstellungen für diesen Zugriffspunkt festlegen. Für jede Anforderung, die über einen Zugriffspunkt eingeht, wertet Amazon S3 die Block Public Access-Einstellungen für diesen Zugriffspunkt, den

zugrunde liegenden Bucket und das Konto des Bucket-Eigentümers aus. Wenn eine dieser Einstellungen darauf hinweist, dass die Anforderung gesperrt werden soll, lehnt Amazon S3 die Anforderung ab.

Weitere Hinweise zur Funktion S3 Block Public Access finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

> **Important**
>
> - Alle Block Public Access-Einstellungen sind standardmäßig für Zugriffspunkte aktiviert. Sie müssen alle Einstellungen, die Sie nicht auf einen Zugriffspunkt anwenden möchten, explizit deaktivieren.
> - Amazon S3 unterstützt derzeit nicht das Ändern der Public Block Access-Einstellungen eines Zugriffspunkts, nachdem der Zugriffspunkt erstellt wurde.

Example

Beispiel: Erstellen eines Zugriffspunkts mit benutzerdefinierten Block-Public-Access-Einstellungen

In diesem Beispiel wird ein Zugriffspunkt mit dem Namen `example-ap` für Bucket *amzn-s3-demo-bucket* in Konto `123456789012` mit nicht standardmäßigen Block Public Access-Einstellungen erstellt. Das Beispiel ruft dann die Konfiguration des neuen Zugriffspunkts ab, um seine Block Public Access-Einstellungen zu überprüfen.

AWS CLI

```
aws s3control create-access-point --name example-ap --account-id
 123456789012 --bucket amzn-s3-demo-bucket--public-access-block-configuration
 BlockPublicAcls=false,IgnorePublicAcls=false,BlockPublicPolicy=true,RestrictPublicBuckets=t
```

```
aws s3control get-access-point --name example-ap --account-id 123456789012

{
    "Name": "example-ap",
    "Bucket": "amzn-s3-demo-bucket",
    "NetworkOrigin": "Internet",
    "PublicAccessBlockConfiguration": {
        "BlockPublicAcls": false,
        "IgnorePublicAcls": false,
```

```
        "BlockPublicPolicy": true,
        "RestrictPublicBuckets": true
    },
    "CreationDate": "2019-11-27T00:00:00Z"
}
```

Verwenden von Zugriffspunkten

Sie können auf die Objekte in einem Amazon S3-Bucket mit einem Access Point zugreifen, indem Sie die Amazon S3 S3-Konsole AWS CLI AWS SDKs,, oder Amazon S3 verwenden RESTAPI.

Access Points haben Amazon-Ressourcennamen (ARNs). Access ARNs Points ähneln BucketARNs, sind jedoch explizit typisiert und kodieren die Region des Access Points und die AWS-Konto ID des Besitzers des Access Points. Weitere Informationen zu finden Sie ARNs unter Amazon Resource Names (ARNs) in der Allgemeine AWS-Referenz.

Access Point ARNs verwendet das Format`arn:aws:s3:`*`region`*`:`*`account-id`*`:accesspoint/`*`resource`*. Beispielsweise:

- `arn:aws:s3:`*`us-west-2`*`:`*`123456789012`*`:accesspoint/`*`test`*steht für den benannten Zugangspunkt*`test`*, der dem Konto gehört *`123456789012`* in der Region *`us-west-2`*.
- `arn:aws:s3:`*`us-west-2`*`:`*`123456789012`*`:accesspoint/*`steht für alle Access Points unter dem Konto *`123456789012`* in der Region *`us-west-2`*.

ARNsVerwenden Sie für Objekte, auf die über einen Access Point zugegriffen wird, das Format`arn:aws:s3:`*`region`*`:`*`account-id`*`:accesspoint/`*`access-point-name`*`/object/`*`resource`*. Beispielsweise:

- `arn:aws:s3:`*`us-west-2`*`:`*`123456789012`*`:accesspoint/`*`test`*`/object/`*`unit-01`*steht für das Objekt *`Einheit-01`*, auf den über den genannten Zugangspunkt zugegriffen wird *`Test`*, im Besitz eines Accounts *`123456789012`* in der Region *`us-west-2`*.
- `arn:aws:s3:`*`us-west-2`*`:`*`123456789012`*`:accesspoint/`*`test`*`/object/*`steht für alle Objekte für den genannten Access Point *`Test`*, im Konto *`123456789012`* in der Region *`us-west-2`*.
- `arn:aws:s3:`*`us-west-2`*`:`*`123456789012`*`:accesspoint/`*`test`*`/object/`*`unit-01`*`/`*`finance`*`/*`steht für alle Objekte unter dem Präfix *`Einheit-01/Finanzen/`* für den genannten Access Point *`Test`*, im Konto *`123456789012`* in der Region *`us-west-2`*.

Zugreifen auf einen Bucket über einen S3-Zugriffspunkt

S3-Zugriffspunkte unterstützen nur die virtual-host-style Adressierung. Um einen Bucket über einen Zugriffspunkt zu adressieren, verwenden Sie das folgende Format.

```
https://access-point-name-account-id.s3-accesspoint.region.amazonaws.com
```

> **Note**
>
> - Wenn der Name Ihres Access Points Bindestriche (-) enthält, schließen Sie die Bindestriche in URL und fügen Sie vor der Konto-ID einen weiteren Bindestrich ein. Um beispielsweise einen Access Point mit dem Namen zu verwenden *Finanzdokumente* gehört einem Konto *123456789012* in der Region *us-west-2*, das URL Angemessene wäre`https://finance-docs-123456789012.s3-accesspoint.us-west-2.amazonaws.com`.
> - S3-Zugangspunkte unterstützen keinen Zugriff überHTTP. Access Points unterstützen nur den sicheren Zugriff überHTTPS.

Themen

- Überwachen und Protokollieren von Zugriffspunkten
- Nutzen von Amazon S3-Zugriffspunkten mit der Amazon S3-Konsole
- Verwenden eines Alias im Bucket-Stil für Ihren S3-Bucket-Zugriffspunkt
- Verwenden von Zugriffspunkten mit kompatiblen Amazon S3-Vorgänge

Wenn Sie über eine Virtual Private Cloud (VPC) verfügen, finden Sie weitere Informationen unter Verwaltung des Amazon S3 S3-Zugriffs mit VPC Endpunkten und S3-Zugriffspunkten.

Überwachen und Protokollieren von Zugriffspunkten

Amazon S3 protokolliert Anfragen, die über Access Points gestellt werden, sowie Anfragen an die API Operationen, die Access Points verwalten, wie `CreateAccessPoint` und`GetAccessPointPolicy`. Um Nutzungsmuster zu überwachen und zu verwalten, können Sie auch Amazon CloudWatch Logs-Anforderungsmetriken für Access Points konfigurieren.

Themen

- CloudWatch Metriken anfordern
- Anforderungsprotokolle

CloudWatch Metriken anfordern

Um die Leistung von Anwendungen, die Access Points verwenden, zu verstehen und zu verbessern, können Sie Amazon S3 S3-Anforderungsmetriken verwenden CloudWatch . Anforderungsmetriken helfen Ihnen beim Überwachen von Amazon S3-Anforderungen, um Probleme bei der Ausführung schnell zu identifizieren und zu beheben.

Standardmäßig stehen diese Anforderungsmetriken auf Ebene der Buckets zur Verfügung. Sie können jedoch einen Filter für Anforderungsmetriken mit einem gemeinsamen Präfix, Objekt-Tags oder einem Zugriffspunkt definieren. Wenn Sie einen Zugriffspunktfilter erstellen, enthält die Konfiguration der Anforderungsmetriken Anforderungen an den von Ihnen angegebenen Zugriffspunkt. Sie können Metriken erhalten, Alarme festlegen und auf Dashboards zugreifen, um Echtzeit-Vorgänge anzuzeigen, die über diesen Zugriffspunkt ausgeführt werden.

Sie müssen sich dafür entscheiden, Metriken anzufordern, indem Sie sie in der Konsole konfigurieren oder Amazon S3 verwendenAPI. Die Anforderungsmetriken sind in 1-Minuten-Intervallen nach einer gewissen Latenz für die Verarbeitung verfügbar. Anforderungsmetriken werden zum gleichen Tarif wie CloudWatch benutzerdefinierte Metriken in Rechnung gestellt. Weitere Informationen finden Sie unter CloudWatch Amazon-Preise.

Informationen zum Erstellen einer Anforderungsmetrikkonfiguration, die nach Zugriffspunkt filtert, finden Sie unter Erstellen einer Metrik-Konfiguration, die nach dem Präfix, Objekt-Tag oder Zugriffspunkt filtert.

Anforderungsprotokolle

Sie können Anfragen protokollieren, die über Access Points gestellt wurden, sowie Anfragen an dieAPIs, die Access Points verwalten, gestellt wurden, z. `CreateAccessPoint` B. `GetAccessPointPolicy`, mithilfe der Serverzugriffsprotokollierung und AWS CloudTrail.

CloudTrail Protokolleinträge für Anfragen, die über Access Points gestellt wurden, enthalten den Zugriffspunkt ARN im `resources` Abschnitt des Protokolls.

Angenommen, folgende Konfiguration liegt vor:

- Ein Bucket mit dem Namen *amzn-s3-demo-bucket1* in der Region *us-west-2* das enthält ein Objekt mit dem Namen *my-image.jpg*
- Ein Zugangspunkt mit dem Namen *my-bucket-ap* das ist verbunden mit *amzn-s3-demo-bucket1*
- Eine AWS-Konto ID von *123456789012*

Das folgende Beispiel zeigt den `resources` Abschnitt eines CloudTrail Protokolleintrags für die vorherige Konfiguration:

```
"resources": [
        {"type": "AWS::S3::Object",
            "ARN": "arn:aws:s3:::amzn-s3-demo-bucket1/my-image.jpg"
        },
        {"accountId": "123456789012",
            "type": "AWS::S3::Bucket",
            "ARN": "arn:aws:s3:::amzn-s3-demo-bucket1"
        },
        {"accountId": "123456789012",
            "type": "AWS::S3::AccessPoint",
            "ARN": "arn:aws:s3:us-west-2:123456789012:accesspoint/my-bucket-ap"
        }
    ]
```

Weitere Informationen zu S3 Server-Zugriffsprotokollen finden Sie unter Protokollieren von Anfragen mit Server-Zugriffsprotokollierung. Weitere Informationen zu AWS CloudTrail finden Sie unter Was ist AWS CloudTrail? im AWS CloudTrail Benutzerhandbuch.

Nutzen von Amazon S3-Zugriffspunkten mit der Amazon S3-Konsole

In diesem Abschnitt wird erläutert, wie Sie Ihre Amazon S3 Access Points mithilfe der AWS Management Console verwalten und verwenden. Bevor Sie beginnen, navigieren Sie zur Detailseite für den Zugriffspunkt, den Sie verwalten oder verwenden möchten, wie im folgenden Verfahren beschrieben.

Themen

- Auflisten von Zugriffspunkten für Ihr Konto
- Auflisten von Zugriffspunkten für einen Bucket
- Anzeigen der Konfigurationsdetails für einen Zugriffspunkt

- Verwenden eines Zugriffspunkts
- Anzeigen der Block Public Access-Einstellungen für einen Zugriffspunkt
- Bearbeiten einer Zugriffspunktrichtlinie
- Erstellen eines Zugriffspunkts

Auflisten von Zugriffspunkten für Ihr Konto

Um alle Access Points aufzulisten, die in Ihrem AWS-Konto

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen der aktuell angezeigten Datei aus AWS-Region. Wählen Sie als Nächstes die Region aus, für die Sie Access Points auflisten möchten.
3. Klicken Sie im Navigationsbereich links in der Konsole auf Access points (Zugriffspunkte).
4. Sehen Sie sich auf der Seite Access Points unter Access Points die Access Points in Ihrem an AWS-Region.
5. (Optional) Suchen Sie nach Zugriffspunkten nach Name, indem Sie einen Namen in das Textfeld neben dem Dropdown-Menü „Region“ eingeben.
6. Wählen Sie den Namen des Zugriffspunkts, den Sie verwalten oder verwenden möchten.

Auflisten von Zugriffspunkten für einen Bucket

Um alle Access Points in Ihnen AWS-Konto für einen einzelnen Bucket aufzulisten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen der aktuell angezeigten Datei und AWS-Region dann die Region aus, für die Sie Access Points auflisten möchten.
3. Klicken Sie im Navigationsbereich auf der linken Seite der Konsole auf Buckets.
4. Wählen Sie auf der Seite Buckets den Namen des Buckets aus, dessen Zugriffspunkte Sie auflisten möchten.
5. Wählen Sie auf der Bucket-Detailseite den Tab Access points (Zugriffspunkte).
6. Wählen Sie den Namen des Zugriffspunkts, den Sie verwalten oder verwenden möchten.

Anzeigen der Konfigurationsdetails für einen Zugriffspunkt

1. Navigieren Sie zur Detailseite des Zugriffspunkts für den Zugriffspunkt, dessen Details Sie anzeigen möchten, wie unter Auflisten von Zugriffspunkten für Ihr Konto beschrieben.
2. Zeigen Sie unter Access point overview (Zugriffspunkt-Übersicht) die Konfigurationsdetails und Eigenschaften für den ausgewählten Zugriffspunkt an.

Verwenden eines Zugriffspunkts

1. Navigieren Sie zur Detailseite des Zugriffspunkts für den gewünschten Zugriffspunkt, wie unter Auflisten von Zugriffspunkten für Ihr Konto beschrieben.
2. Wählen Sie im Tab Objects (Objekte) den Namen eines Objekts oder von Objekten aus, auf das/die Sie über den Zugriffspunkt zugreifen möchten. Die Konsole zeigt über dem Namen Ihres Buckets eine Beschriftung mit dem aktuell von Ihnen verwendeten Zugriffspunkt an. Während Sie den Zugriffspunkt verwenden, können Sie nur die Objekt-Vorgänge ausführen, die durch die Zugriffspunktberechtigungen gewährt werden.

> **Note**
>
> - In der Konsolenansicht werden immer alle Objekte im Bucket angezeigt. Die Verwendung eines Zugriffspunkts, wie in dieser Prozedur beschrieben, beschränkt die Vorgänge, die Sie für diese Objekte ausführen können, jedoch nicht, ob Sie sehen können, dass sie im Bucket vorhanden sind.
> - Die S3-Managementkonsole unterstützt nicht die Verwendung von Virtual Private Cloud (VPC) -Zugriffspunkten für den Zugriff auf Bucket-Ressourcen. Verwenden Sie Amazon S3, um von einem VPC Access Point aus auf AWS CLI Bucket-Ressourcen zuzugreifen RESTAPIs. AWS SDKs

Anzeigen der Block Public Access-Einstellungen für einen Zugriffspunkt

1. Navigieren Sie zur Detailseite des Zugriffspunkts für den Zugriffspunkt, dessen Einstellungen Sie anzeigen möchten, wie unter Auflisten von Zugriffspunkten für Ihr Konto beschrieben.
2. Wählen Sie Permissions (Berechtigungen).
3. Überprüfen Sie unter Access point policy (Zugriffspunktrichtlinie) die Block-Public-Access-Einstellungen für den Zugriffspunkt.

> **Note**
>
> Sie können die Block Public Access-Einstellungen für einen Zugriffspunkt nicht ändern, nachdem der Zugriffspunkt erstellt wurde.

Bearbeiten einer Zugriffspunktrichtlinie

1. Navigieren Sie zur Detailseite des Zugriffspunkts für den Zugriffspunkt, dessen Richtlinie Sie bearbeiten möchten, wie unter Auflisten von Zugriffspunkten für Ihr Konto beschrieben.
2. Wählen Sie Permissions (Berechtigungen).
3. Wählen Sie unter Access point policy (Zugriffspunktrichtlinie) Edit (Bearbeiten) aus.
4. Geben Sie die Richtlinie für den Zugriffspunkt in das Textfeld ein. Die Konsole zeigt automatisch den Amazon-Ressourcennamen (ARN) für den Access Point an, den Sie in der Richtlinie verwenden können.

Erstellen eines Zugriffspunkts

1. Navigieren Sie zur Liste der Zugriffspunkte für Ihr Konto oder für einen bestimmten Bucket, wie unter Auflisten von Zugriffspunkten für Ihr Konto beschrieben.
2. Aktivieren Sie die Optionsschaltfläche neben dem Namen des Zugriffspunkts, den Sie löschen möchten.
3. Wählen Sie Delete (Löschen).
4. Bestätigen Sie, dass Sie Ihren Zugriffspunkt löschen möchten, indem Sie den Namen in das angezeigte Textfeld eingeben und Delete (Löschen) wählen.

Verwenden eines Alias im Bucket-Stil für Ihren S3-Bucket-Zugriffspunkt

Wenn Sie einen Zugriffspunkt erstellen, generiert Amazon S3 automatisch einen Alias, den Sie anstelle eines Amazon S3-Bucket-Namens für den Datenzugriff verwenden können. Sie können diesen Access Point-Alias anstelle eines Amazon-Ressourcennamens (ARN) für den Betrieb auf der Datenebene des Access Points verwenden. Eine Liste dieser Vorgänge finden Sie unter Kompatibilität des Access Points mit Diensten AWS.

Im Folgenden finden Sie ein Beispiel ARN und einen Zugriffspunkt-Alias für einen Access Point mit dem Namen*my-access-point*.

- ARN – `arn:aws:s3:`*region*`:`*account-id*`:accesspoint/`*my-access-point*
- Zugriffspunkt-Alias – *my-access-point*`-`*hrzrlukc5m36ft7okagglf3gmwluquse1b*`-s3alias`

Weitere Informationen zu finden Sie ARNs unter Amazon Resource Names (ARNs) in der Allgemeine AWS-Referenz.

Aliasname für Zugriffspunkte

Ein Zugriffspunkt-Aliasname wird innerhalb desselben Namespace wie ein Amazon S3-Bucket erstellt. Dieser Aliasname wird automatisch generiert und kann nicht geändert werden. Ein Zugriffspunkt-Aliasname erfüllt alle Anforderungen eines gültigen Amazon S3-Bucket-Namens und besteht aus den folgenden Teilen:

access point prefix`-`*metadata*`-s3alias`

> **Note**
>
> Das Suffix `-s3alias` ist für Zugriffspunkt-Aliasnamen reserviert und kann nicht für Bucket- oder Zugriffspunkt-Namen verwendet werden. Weitere Informationen zu Amazon-S3-Bucket-Benennungsregeln finden Sie unter Allgemeine Benennungsregeln für Buckets.

Zugriffspunkt-Alias-Anwendungsfälle und -beschränkungen

Bei der Übernahme von Zugriffspunkten können Sie Zugriffspunkt-Aliasnamen verwenden, ohne dass umfangreiche Codeänderungen erforderlich sind.

Wenn Sie einen Zugriffspunkt erstellen, generiert Amazon S3 automatisch einen Zugriffspunkt-Aliasnamen, wie im folgenden Beispiel gezeigt. Zum Ausführen dieses Befehls ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control create-access-point --bucket amzn-s3-demo-bucket1 --name my-access-point
 --account-id 111122223333
{
    "AccessPointArn":
```

```
    "arn:aws:s3:region:111122223333:accesspoint/my-access-point",
    "Alias": "my-access-point-aqfqprnstn7aefdfbarligizwgyfouse1a-s3alias"
}
```

Sie können diesen Zugriffspunkt-Aliasnamen anstelle eines Amazon S3-Bucket-Namen für jeden Datenebenen-Vorgang verwenden. Eine Liste dieser Vorgänge finden Sie unter Kompatibilität des Access Points mit Diensten AWS.

Das folgende AWS CLI Beispiel für den `get-object` Befehl verwendet den Access Point-Alias des Buckets, um Informationen über das angegebene Objekt zurückzugeben. Zum Ausführen dieses Befehls ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws s3api get-object --bucket my-access-point-aqfqprnstn7aefdfbarligizwgyfouse1a-
s3alias --key dir/my_data.rtf my_data.rtf

{
    "AcceptRanges": "bytes",
    "LastModified": "2020-01-08T22:16:28+00:00",
    "ContentLength": 910,
    "ETag": "\"00751974dc146b76404bb7290f8f51bb\"",
    "VersionId": "null",
    "ContentType": "text/rtf",
    "Metadata": {}
}
```

Einschränkungen

- Aliase können nicht von Kunden konfiguriert werden.
- Aliasse können auf einem Zugriffspunkt nicht gelöscht, geändert oder deaktiviert werden.
- Sie können diesen Zugriffspunkt-Aliasnamen anstelle eines Amazon-S3-Bucket-Namen für manche Datenebenen-Operationen verwenden. Eine Liste dieser Vorgänge finden Sie unter Zugriffspunkt-Kompatibilität mit S3-Vorgänge.
- Sie können einen Aliasnamen für Amazon S3-Kontrollebenenvorgänge nicht verwenden. Eine Liste der Vorgänge auf der Amazon S3-Steuerebene finden Sie unter Amazon S3 Control in der Amazon Simple Storage Service API Reference.
- Sie können S3-Zugriffspunkt-Aliase nicht als Quelle oder Ziel für Move-Operationen in der Amazon S3 S3-Konsole verwenden.
- Aliase können nicht in Richtlinien AWS Identity and Access Management (IAM) verwendet werden.

- Aliase können nicht als Protokollierungsziel für S3-Server-Zugriffsprotokolle verwendet werden.
- Aliase können nicht als Ziel für AWS CloudTrail die Protokollierung von Protokollen verwendet werden.
- Amazon unterstützt SageMaker GroundTruth keine Access Point-Aliase.

Verwenden von Zugriffspunkten mit kompatiblen Amazon S3-Vorgänge

Die folgenden Beispiele veranschaulichen, wie Zugriffspunkte mit kompatiblen Vorgängen in Amazon S3 verwendet werden.

Themen

- Kompatibilität des Access Points mit Diensten AWS
- Zugriffspunkt-Kompatibilität mit S3-Vorgänge
- Anfordern eines Objekts über einen Zugriffspunkt
- Hochladen eines Objekts über einen Zugriffspunkt-Alias
- Löschen eines Objekts über einen Zugriffspunkt
- Auflisten von Objekten über einen Zugriffspunkt-Alias
- Hinzufügen eines Tag-Satzes zu einem Objekt über einen Zugriffspunkt
- Erteilen Sie Zugriffsberechtigungen über einen Access Point mithilfe eines ACL

Kompatibilität des Access Points mit Diensten AWS

Mit Aliasnamen von Amazon-S3-Zugriffspunkten können Anwendungen, die einen S3-Bucketnamen benötigen, problemlos einen Zugriffspunkt verwenden. Sie können S3-Zugriffspunkt-Aliasnamen überall verwenden, wo Sie S3-Bucketnamen verwenden, um auf Daten in S3 zuzugreifen. Weitere Informationen finden Sie unter Zugriffspunkt-Alias-Anwendungsfälle und -beschränkungen.

Zugriffspunkt-Kompatibilität mit S3-Vorgänge

Sie können Access Points verwenden, um mithilfe der folgenden Teilmenge von Amazon S3 APIs auf einen Bucket zuzugreifen. Alle unten aufgeführten Operationen können entweder Access Point ARNs - oder Access Point-Aliase akzeptieren:

S3-Operationen

- `AbortMultipartUpload`

- CompleteMultipartUpload
- CopyObject(Nur Kopien derselben Region)
- CreateMultipartUpload
- DeleteObject
- DeleteObjects
- DeleteObjectTagging
- GetBucketAcl
- GetBucketCors
- GetBucketLocation
- GetBucketNotificationConfiguration
- GetBucketPolicy
- GetObject
- GetObjectAcl
- GetObjectAttributes
- GetObjectLegalHold
- GetObjectRetention
- GetObjectTagging
- HeadBucket
- HeadObject
- ListMultipartUploads
- ListObjects
- ListObjectsV2
- ListObjectVersions
- ListParts
- Presign
- PutObject
- PutObjectAcl
- PutObjectLegalHold
- PutObjectRetention

- PutObjectTagging
- RestoreObject
- UploadPart
- UploadPartCopy(Nur Kopien aus derselben Region)

Anfordern eines Objekts über einen Zugriffspunkt

Im folgenden Beispiel wird das Objekt angefordert *my-image.jpg* durch den Access Point *Prod* gehört der Konto-ID *123456789012* in der Region *us-west-2*, und speichert die heruntergeladene Datei als *download.jpg*.

AWS CLI

```
aws s3api get-object --key my-image.jpg --bucket arn:aws:s3:us-
west-2:123456789012:accesspoint/prod download.jpg
```

Hochladen eines Objekts über einen Zugriffspunkt-Alias

Das folgende Beispiel lädt das Objekt hoch *my-image.jpg* über den Access Point-Alias, der der Konto-ID *my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b*-s3alias gehört *123456789012* in der Region *us-west-2*.

AWS CLI

```
aws s3api put-object --bucket my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b-
s3alias --key my-image.jpg --body my-image.jpg
```

Löschen eines Objekts über einen Zugriffspunkt

Im folgenden Beispiel wird das Objekt gelöscht *my-image.jpg* durch den Access Point *Prod* gehört der Konto-ID *123456789012* in der Region *us-west-2*.

AWS CLI

```
aws s3api delete-object --bucket arn:aws:s3:us-west-2:123456789012:accesspoint/prod
 --key my-image.jpg
```

Auflisten von Objekten über einen Zugriffspunkt-Alias

Im folgenden Beispiel werden Objekte anhand des Access Point-Alias aufgelistet, der der Konto-ID *my-access-point-hrzrlukc5m36ft7okagglf3gmwluquse1b*-s3alias gehört *123456789012* in der Region *us-west-2*.

AWS CLI

```
aws s3api list-objects-v2 --bucket my-access-point-
hrzrlukc5m36ft7okagglf3gmwluquse1b-s3alias
```

Hinzufügen eines Tag-Satzes zu einem Objekt über einen Zugriffspunkt

Im folgenden Beispiel wird dem vorhandenen Objekt ein Tag-Set hinzugefügt *my-image.jpg* durch den Access Point *Prod* gehört der Konto-ID *123456789012* in der Region *us-west-2*.

AWS CLI

```
aws s3api put-object-tagging --bucket arn:aws:s3:us-
west-2:123456789012:accesspoint/prod --key my-image.jpg --tagging
 TagSet=[{Key="finance",Value="true"}]
```

Erteilen Sie Zugriffsberechtigungen über einen Access Point mithilfe eines ACL

Im folgenden Beispiel wird ein ACL auf ein vorhandenes Objekt angewendet *my-image.jpg* durch den Access Point *Prod* gehört der Konto-ID *123456789012* in der Region *us-west-2*.

AWS CLI

```
aws s3api put-object-acl --bucket arn:aws:s3:us-west-2:123456789012:accesspoint/prod
 --key my-image.jpg --acl private
```

Verwalten des Zugriffs mit S3-Zugriffsberechtigungen

Zur Einhaltung des Grundsatzes der geringsten Berechtigung definieren Sie die Details des Zugriffs auf Ihre Amazon-S3-Daten basierend auf Anwendungen, Personas, Gruppen oder Organisationseinheiten. Abhängig von Umfang und Komplexität der Zugriffsmuster können Sie

verschiedene Ansätze zur Gewährung eines detaillierten Zugriffs auf Ihre Daten in Amazon S3 verwenden.

Der einfachste Ansatz zur Verwaltung des Zugriffs auf eine small-to-medium Anzahl von Datensätzen in Amazon S3 nach AWS Identity and Access Management (IAM) -Prinzipalen besteht darin, IAMBerechtigungsrichtlinien und S3-Bucket-Richtlinien zu definieren. Diese Strategie funktioniert, solange die erforderlichen Richtlinien innerhalb der Richtliniengrößenbeschränkungen der S3-Bucket-Richtlinien (20 KB) und IAM Richtlinien (5 KB) und innerhalb der Anzahl der pro Konto zulässigen IAM Prinzipale liegen.

Wenn die Zahl Ihrer Datensätze und Anwendungsfälle wächst, werden auch Richtlinien wichtiger. Ein Ansatz, der Richtlinienanweisungen deutlich mehr Raum gibt, besteht in der Verwendung von S3-Zugangspunkten als zusätzlichen Endpunkten für S3-Buckets, da jeder Zugangspunkt eine eigene Richtlinie haben kann. Sie können recht detaillierte Zugriffskontrollmuster definieren, da Sie Tausende von Zugriffspunkten AWS-Region pro Konto haben können, wobei für jeden Zugriffspunkt eine Richtlinie mit einer Größe von bis zu 20 KB gelten kann. S3-Zugangspunkte vergrößern zwar den verfügbaren Raum für Richtlinien, erfordern jedoch einen Mechanismus, mit dem Clients den richtigen Zugangspunkt für den richtigen Datensatz finden können.

Ein dritter Ansatz besteht darin, ein IAMSitzungsbroker-Muster zu implementieren, bei dem Sie eine Logik für die Zugriffsentscheidung implementieren und für jede Zugriffssitzung dynamisch kurzfristige IAM Sitzungsanmeldeinformationen generieren. Der IAM Sitzungsbroker-Ansatz unterstützt zwar beliebig dynamische Berechtigungsmuster und lässt sich effektiv skalieren, Sie müssen jedoch die Logik für die Zugriffsmuster erstellen.

Sie können anstelle dieser Ansätze S3 Access Grants verwenden, um den Zugriff auf Ihre Amazon-S3-Daten zu verwalten. S3 Access Grants bietet ein vereinfachtes Modell für die Definition von Zugriffsberechtigungen für Daten in Amazon S3 nach Präfix, Bucket oder Objekt. Darüber hinaus können Sie S3 Access Grants verwenden, um sowohl IAM Prinzipalen als auch direkt Benutzern oder Gruppen aus Ihrem Unternehmensverzeichnis Zugriff zu gewähren.

In der Regel definieren Sie Berechtigungen für Daten in Amazon S3, indem Sie Benutzer und Gruppen Datensätzen zuordnen. Sie können mit S3 Access Grants die direkte Zuordnung von S3-Präfixen zu Benutzern und Rollen in Amazon-S3-Buckets und -Objekten definieren. Mit dem vereinfachten Zugriffsschema in S3 Access Grants können Sie beiden IAM Prinzipalen und direkt Benutzern oder Gruppen aus einem Unternehmensverzeichnis Lese-, Schreib- oder Leserechte pro S3-Präfix gewähren. Über diese S3-Access-Grants-Funktionen können Anwendungen im Namen des aktuell authentifizierten Benutzers der Anwendung Daten von Amazon S3 anfordern.

Wenn Sie S3 Access Grants mit der Funktion zur Verbreitung vertrauenswürdiger Identitäten von integrieren AWS IAM Identity Center, können Ihre Anwendungen Anfragen AWS-Services (einschließlich S3 Access Grants) direkt im Namen eines authentifizierten Unternehmensverzeichnisbenutzers stellen. Ihre Anwendungen müssen den Benutzer nicht mehr zuerst einem IAM Prinzipal zuordnen. Darüber hinaus wird die Überprüfung vereinfacht, welcher Benutzer auf welches S3-Objekt zugegriffen hat, da Endbenutzeridentitäten bis zu Amazon S3 weitergegeben werden. Sie müssen die Beziehung zwischen verschiedenen Benutzern und IAM Sitzungen nicht mehr rekonstruieren. Wenn Sie S3 Access Grants mit IAM Identity Center Trusted Identity Propagation verwenden, enthält jedes AWS CloudTrailDatenereignis für Amazon S3 einen direkten Verweis auf den Endbenutzer, in dessen Namen auf die Daten zugegriffen wurde.

Weitere Informationen zu S3 Access Grants finden Sie in den folgenden Themen.

Themen

- Konzepte von S3 Access Grants
- S3 Access Grants und Unternehmensverzeichnisidentitäten
- Erste Schritte mit S3 Access Grants
- Arbeiten mit S3 Access Grants-Instanzen
- Mit S3 Access Grants-Standorten arbeiten
- Arbeiten mit Zuschüssen in S3 Access Grants
- Abrufen von S3-Daten mithilfe von Zugriffsberechtigungen
- Kontoübergreifender Zugriff mit S3 Access Grants
- Verwenden von AWS Tags mit S3 Access Grants
- Limits für S3 Access Grants
- S3-Access-Grants-Integrationen

Konzepte von S3 Access Grants

Arbeitsablauf bei S3 Access Grants

Der Arbeitsablauf für S3 Access Grants sieht wie folgt aus:

1. Erstellen Sie eine S3 Access Grants-Instanz. Siehe Arbeiten mit S3 Access Grants-Instanzen.

2. Registrieren Sie in Ihrer S3 Access Grants-Instance Standorte in Ihren Amazon S3 S3-Daten und ordnen Sie diese Standorte AWS Identity and Access Management (IAM) -Rollen zu. Siehe Registrieren eines Speicherorts.
3. Erstellen Sie Zuschüsse für Stipendiaten, die den Empfängern Zugriff auf Ihre S3-Ressourcen gewähren. Siehe Arbeiten mit Zuschüssen in S3 Access Grants.
4. Der Empfänger fordert temporäre Anmeldeinformationen von S3 Access Grants an. Siehe Anfordern des Zugriffs auf Amazon-S3-Daten über S3 Access Grants.
5. Der Empfänger greift mit diesen temporären Anmeldeinformationen auf die S3-Daten zu. Siehe Zugriff auf S3-Daten mithilfe von Anmeldeinformationen, die von S3 Access Grants bereitgestellt wurden.

Weitere Informationen finden Sie unter Erste Schritte mit S3 Access Grants.

S3-Access-Grants-Instances

Eine S3 Access Grants-Instanz ist ein logischer Container für einzelne Zuschüsse. Wenn Sie eine S3 Access Grants-Instanz erstellen, müssen Sie eine angeben AWS-Region. Jeder AWS-Region in Ihnen AWS-Konto kann eine S3 Access Grants-Instanz haben. Weitere Informationen finden Sie unter Arbeiten mit S3 Access Grants-Instanzen.

Wenn Sie S3 Access Grants verwenden möchten, um Zugriff auf Benutzer- und Gruppenidentitäten aus Ihrem Unternehmensverzeichnis zu gewähren, müssen Sie Ihre S3 Access Grants-Instanz auch einer AWS IAM Identity Center Instanz zuordnen. Weitere Informationen finden Sie unter S3 Access Grants und Unternehmensverzeichnisidentitäten.

Eine neu erstellte S3 Access Grants-Instanz ist leer. Sie müssen einen Speicherort in der Instance registrieren. Dabei kann es sich um den S3-Standardpfad (`s3://`), einen Bucket oder ein Präfix innerhalb eines Buckets handeln. Nachdem Sie mindestens einen Standort registriert haben, können Sie Zugriffsberechtigungen erstellen, die Zugriff auf Daten an diesem registrierten Standort gewähren.

Orte

Ein Standort von S3 Access Grants ordnet Buckets oder Präfixe einer AWS Identity and Access Management (IAM) -Rolle zu. S3 Access Grants übernimmt diese IAM Rolle, um temporäre Anmeldeinformationen an den Empfänger weiterzugeben, der auf diesen bestimmten Standort zugreift. Sie müssen zuerst mindestens einen Standort in Ihrer S3 Access Grants-Instanz registrieren, bevor Sie eine Zugriffsberechtigung erstellen können.

Wir empfehlen, den Standardstandort (`s3://`) zu registrieren und ihn einer IAM Rolle zuzuordnen. Der Standort am Standard-S3-Pfad (`s3://`) deckt den Zugriff auf all Ihre S3-Buckets in Ihrem Konto ab. AWS-Region Wenn Sie eine Zugriffsberechtigung erstellen, können Sie den Umfang der Gewährung auf einen Bucket, ein Präfix oder ein Objekt innerhalb des Standardspeicherorts einschränken.

Bei komplexeren Anwendungsfällen für die Zugriffsverwaltung müssen Sie möglicherweise mehr als den Standardspeicherort registrieren. Einige Beispiele für solche Anwendungsfälle sind:

- Nehmen wir an, dass der *amzn-s3-demo-bucket* ist ein registrierter Standort in Ihrer S3 Access Grants-Instance, dem eine IAM Rolle zugeordnet ist, aber dieser IAM Rolle wird der Zugriff auf ein bestimmtes Präfix innerhalb des Buckets verweigert. In diesem Fall können Sie das Präfix, auf das die IAM Rolle keinen Zugriff hat, als separaten Standort registrieren und diesen Standort einer anderen IAM Rolle mit dem erforderlichen Zugriff zuordnen.
- Angenommen, Sie möchten Zuweisungen einrichten, die den Zugriff nur auf Benutzer innerhalb eines Virtual Private Cloud (VPC) -Endpunkts beschränken. In diesem Fall können Sie einen Standort für einen Bucket registrieren, in dem die IAM Rolle den Zugriff auf den VPC Endpunkt einschränkt. Später, wenn ein Empfänger S3 Access Grants um Anmeldeinformationen bittet, übernimmt S3 Access Grants die IAM Rolle des Standorts, die temporären Anmeldeinformationen zu verkaufen. Diese Anmeldeinformationen verweigern den Zugriff auf den jeweiligen Bucket, sofern sich der Anrufer nicht innerhalb des Endpunkts befindet. VPC Diese Ablehnungsberechtigung wird zusätzlich zu der regulären READ oder in der Gewährung angegebenen READWRITE Berechtigung angewendet. WRITE

Wenn Ihr Anwendungsfall erfordert, dass Sie mehrere Standorte in Ihrer S3 Access Grants-Instanz registrieren, können Sie einen der folgenden Standorte registrieren:

- Der Standard-S3-Standort (`s3://`)
- Ein Bucket (zum Beispiel *amzn-s3-demo-bucket*) oder mehrere Buckets
- Ein Bucket und ein Präfix (zum Beispiel*amzn-s3-demo-bucket*`/prefix*`) oder mehrere Präfixe

Die maximale Anzahl von Standorten, die Sie in Ihrer S3 Access Grants-Instanz registrieren können, finden Sie unterLimits für S3 Access Grants. Weitere Informationen zur Registrierung eines S3 Access Grants-Standorts finden Sie unterRegistrieren eines Speicherorts.

Nachdem Sie den ersten Standort in Ihrer S3 Access Grants-Instanz registriert haben, enthält Ihre Instance immer noch keine individuellen Zugriffsberechtigungen. Es wurde also noch kein Zugriff auf Ihre S3-Daten gewährt. Sie können jetzt Zugriffsberechtigungen erstellen, um Zugriff

zu gewähren. Weitere Informationen zum Erstellen von Zuschüssen finden Sie unterArbeiten mit Zuschüssen in S3 Access Grants.

Gewährungen

Eine individuelle Gewährung in einer S3 Access Grants-Instanz ermöglicht es einer bestimmten Identität — einem IAM Prinzipal, einem Benutzer oder einer Gruppe in einem Unternehmensverzeichnis —, Zugriff von einem Standort aus zu erhalten, der in Ihrer S3 Access Grants-Instanz registriert ist.

Wenn Sie einen Grant erstellen, müssen Sie nicht den Zugriff auf den gesamten registrierten Standort gewähren. Sie können den Zugriffsumfang des Zuschusses innerhalb eines Standorts einschränken. Wenn es sich bei dem registrierten Speicherort um den Standard-S3-Pfad (`s3://`) handelt, müssen Sie den Geltungsbereich des Grants auf einen Bucket, ein Präfix innerhalb eines Buckets oder ein bestimmtes Objekt einschränken. Wenn es sich bei dem registrierten Ort des Grants um einen Bucket oder ein Präfix handelt, können Sie Zugriff auf den gesamten Bucket oder das Präfix gewähren oder optional den Geltungsbereich des Grants auf ein Präfix, ein Unterpräfix oder ein Objekt einschränken.

Im Grant legen Sie auch die Zugriffsebene des Grants aufREAD,WRITE, oder READWRITE fest. Angenommen, Sie verfügen über einen Zuschuss, der der Unternehmensverzeichnisgruppe `01234567-89ab-cdef-0123-456789abcdef` READ Zugriff auf den Bucket gewährts3://*amzn-s3-demo-bucket*/projects/items/*. Benutzer in dieser Gruppe können auf jedes Objekt READ zugreifen, das einen Objektschlüsselnamen hat, der mit dem Präfix `projects/items/` im Bucket namens beginnt *amzn-s3-demo-bucket*.

Die maximale Anzahl von Grants, die Sie in Ihrer S3 Access Grants-Instanz erstellen können, finden Sie unterLimits für S3 Access Grants. Weitere Informationen zum Erstellen von Zuschüssen finden Sie unterErstellen von Gewährungen.

Temporäre S3-Access-Grants-Anmeldeinformationen

Nachdem Sie einen Grant erstellt haben, kann eine autorisierte Anwendung, die die in der Grant angegebene Identität verwendet, just-in-time Zugangsdaten anfordern. Zu diesem Zweck ruft die Anwendung den GetDataAccessAPIS3-Vorgang auf. Empfänger können diesen API Vorgang verwenden, um Zugriff auf die S3-Daten zu beantragen, die Sie mit ihnen geteilt haben.

Die S3-Access-Grants-Instance evaluiert die `GetDataAccess`-Anforderung anhand der verfügbaren Gewährungen. Wenn es einen passenden Zuschuss für den Antragsteller gibt, übernimmt S3 Access Grants die IAM Rolle, die dem registrierten Standort des entsprechenden

Zuschusses zugeordnet ist. S3 Access Grants schränkt die Berechtigungen der temporären Anmeldeinformationen so ein, dass sie nur auf den S3-Bucket, das Präfix oder das Objekt zugreifen, die durch den Geltungsbereich der Grant angegeben sind.

Die Ablaufzeit der temporären Zugangsdaten ist standardmäßig auf 1 Stunde eingestellt, Sie können sie jedoch auf einen beliebigen Wert zwischen 15 Minuten und 12 Stunden festlegen. Die maximale Sitzungsdauer finden Sie in der AssumeRoleAPIReferenz.

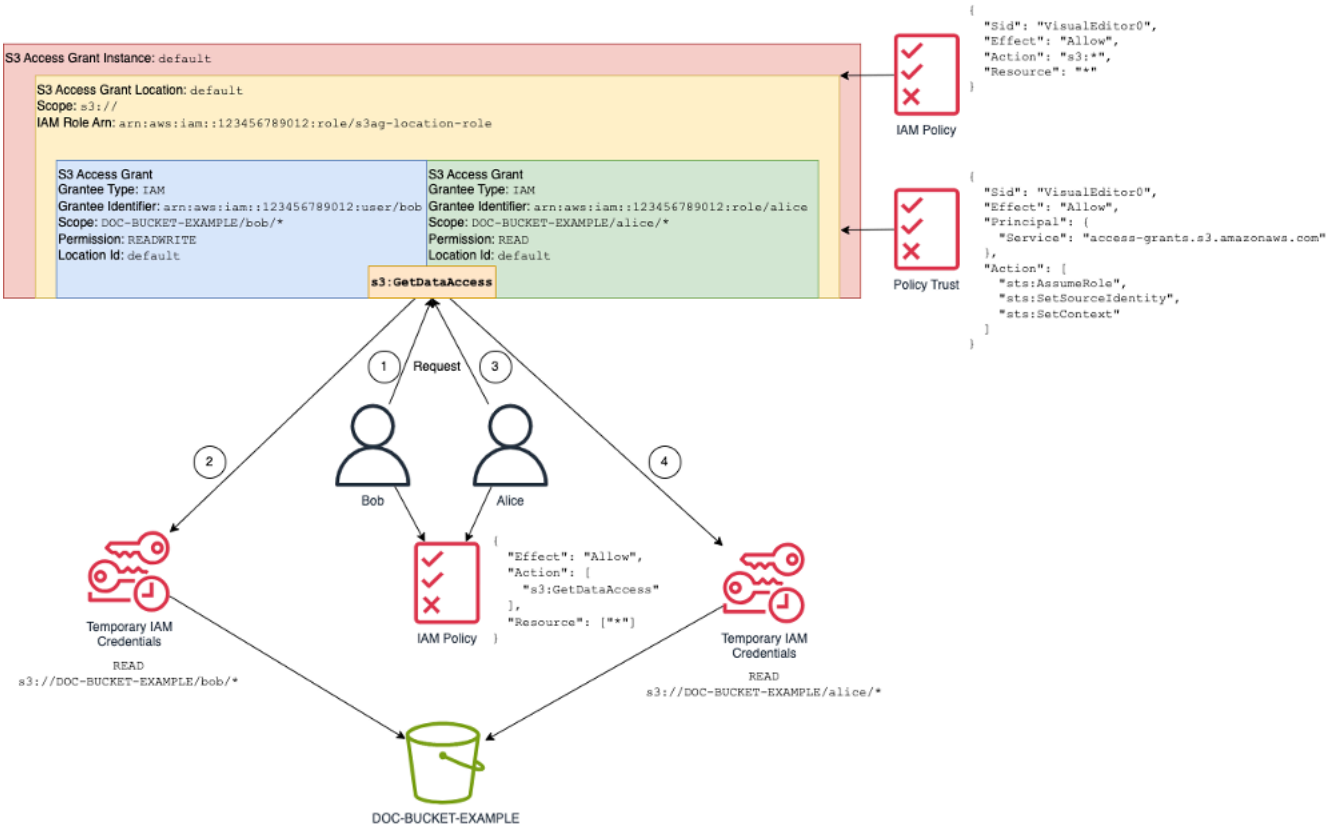
Funktionsweise

In der folgenden Abbildung `s3://` ist ein Amazon S3 S3-Standardstandort mit dem Geltungsbereich für die IAM Rolle registriert`s3ag-location-role`. Diese IAM Rolle ist berechtigt, Amazon S3 S3-Aktionen innerhalb des Kontos durchzuführen, wenn ihre Anmeldeinformationen über S3 Access Grants abgerufen wurden.

An diesem Standort werden zwei individuelle Zugriffsberechtigungen für zwei IAM Benutzer erstellt. Dem IAM Benutzer Bob werden `READ` sowohl der Zugriff als auch der `WRITE` Zugriff auf das `bob/` Präfix im `DOC-BUCKET-EXAMPLE` Bucket gewährt. Einer anderen IAM Rolle, Alice, wird nur `READ` Zugriff auf das `alice/` Präfix im `DOC-BUCKET-EXAMPLE` Bucket gewährt. Es wurde eine Gewährung (blau) definiert, damit Bob auf das Präfix `bob/` im Bucket `DOC-BUCKET-EXAMPLE` zugreifen kann. Es wurde eine Gewährung (grün) definiert, damit Alice auf das Präfix `alice/` im Bucket `DOC-BUCKET-EXAMPLE` zugreifen kann.

Wenn Bob `READ` Daten benötigt, ruft die IAM Rolle, die dem Standort zugeordnet ist, an dem sich sein Grant befindet, den GetDataAccessAPIVorgang S3 Access Grants auf. Wenn Bob versucht, `READ` mit einem S3-Präfix oder Objekt zu beginnen`s3://DOC-BUCKET-EXAMPLE/bob/*`, gibt die `GetDataAccess` Anfrage einen Satz temporärer IAM Sitzungsanmeldedaten mit der entsprechenden Genehmigung zurück`s3://DOC-BUCKET-EXAMPLE/bob/*`. Ähnlich kann Bob `WRITE`-Zugriff für alle S3-Präfixe oder Objekte erhalten, die mit `s3://DOC-BUCKET-EXAMPLE/bob/*` beginnen, da die Gewährung dies ebenfalls zulässt.

Und ähnlich kann Alice `READ`-Zugriff für alles erhalten, das mit `s3://DOC-BUCKET-EXAMPLE/alice/` beginnt. Wenn sie jedoch versucht, `WRITE`-Zugriff auf Buckets, Präfixe oder Objekte in `s3://` zu erhalten, wird ihr die Fehlermeldung „Access Denied“ (403 Forbidden) angezeigt, da es keine Gewährung gibt, die ihr einen `WRITE`-Zugriff auf Daten gewährt. Wenn Alice Zugriff (`READ` oder `WRITE`) auf Daten außerhalb von `s3://DOC-BUCKET-EXAMPLE/alice/` anfordert, wird ihr ebenfalls die Fehlermeldung „Access Denied“ angezeigt.



Dieses Muster kann auf eine große Zahl von Benutzern und Buckets skaliert werden und vereinfacht die Verwaltung dieser Berechtigungen. Anstatt jedes Mal, wenn Sie individuelle Benutzer-Präfix-Zugriffsbeziehungen hinzufügen oder entfernen möchten, potenziell umfangreiche S3-Bucket-Richtlinien zu bearbeiten, können Sie einzelne, diskrete Gewährungen hinzufügen und entfernen.

S3 Access Grants und Unternehmensverzeichnisidentitäten

Sie können Amazon S3 Access Grants verwenden, um AWS Identity and Access Management (IAM) Principals (Benutzern oder Rollen) Zugriff zu gewähren, sowohl für dieselben AWS-Konto als auch für andere. Häufig handelt es sich jedoch bei der Entität, die auf die Daten zugreift, um Endbenutzer in Ihrem Unternehmensverzeichnis. Anstatt IAM Prinzipalen Zugriff zu gewähren, können Sie S3 Access Grants verwenden, um Benutzern und Gruppen Ihres Unternehmens direkt Zugriff zu gewähren. Mit S3 Access Grants müssen Sie Ihre Unternehmensidentitäten nicht mehr zwischengeschalteten IAM Prinzipalen zuordnen, um über Ihre Unternehmensanwendungen auf Ihre S3-Daten zuzugreifen.

Diese neue Funktionalität — Unterstützung für den Zugriff auf Daten mithilfe von Endbenutzeridentitäten — wird bereitgestellt, indem Sie Ihre S3 Access Grants-Instanz mit einer Instanz verknüpfen. AWS IAM Identity Center IAMIdentity Center unterstützt standardbasierte Identitätsanbieter und ist die zentrale Anlaufstelle AWS für alle Dienste oder Funktionen,

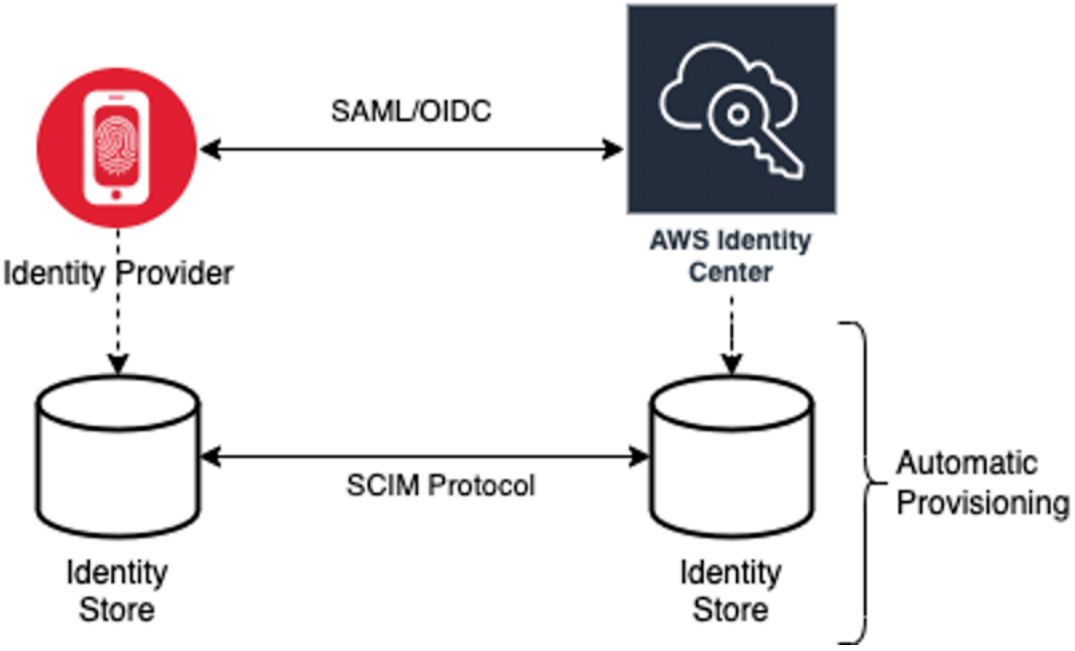
einschließlich S3 Access Grants, die Endbenutzeridentitäten unterstützen. IAMIdentity Center bietet über seine Funktion zur Weitergabe vertrauenswürdiger Identitäten Unterstützung bei der Authentifizierung von Unternehmensidentitäten. Weitere Informationen finden Sie unter Vertrauenswürdige Identitätsweitergabe über Anwendungen hinweg.

Um mit der Unterstützung von Personalidentitäten in S3 Access Grants zu beginnen, müssen Sie zunächst in IAM Identity Center die Identitätsbereitstellung zwischen Ihrem Corporate Identity Provider und IAM Identity Center konfigurieren. IAMIdentity Center unterstützt Anbieter von Unternehmensidentitäten wie Okta, Microsoft Entra ID (früher Azure Active Directory) oder einen anderen externen Identitätsanbieter (IdP), der das System for Cross-Domain Identity Management (SCIM) -Protokoll unterstützt. Wenn Sie IAM Identity Center mit Ihrem IdP verbinden und die automatische Bereitstellung aktivieren, werden die Benutzer und Gruppen Ihres IdP mit dem Identitätsspeicher in IAM Identity Center synchronisiert. Nach diesem Schritt verfügt IAM Identity Center über eine eigene Ansicht Ihrer Benutzer und Gruppen, sodass Sie mithilfe anderer AWS-Services Funktionen wie S3 Access Grants auf sie verweisen können. Weitere Informationen zur Konfiguration der automatischen Bereitstellung von IAM Identity Center finden Sie unter Automatische Bereitstellung im AWS IAM Identity Center Benutzerhandbuch.

IAMIdentity Center ist integriert, AWS Organizations sodass Sie Berechtigungen für mehrere Benutzer zentral verwalten können, AWS-Konten ohne jedes Ihrer Konten manuell konfigurieren zu müssen. In einer typischen Organisation konfiguriert Ihr Identitätsadministrator eine IAM Identity Center-Instanz für die gesamte Organisation als zentralen Punkt der Identitätssynchronisierung. Diese IAM Identity Center-Instanz wird in der Regel in einer dedizierten Instanz AWS-Konto in Ihrer Organisation ausgeführt. In dieser allgemeinen Konfiguration können Sie in S3 Access Grants auf Benutzer- und Gruppenidentitäten von jedem beliebigen Mitglied AWS-Konto der Organisation zugreifen.

Wenn Ihr AWS Organizations Administrator jedoch noch keine zentrale IAM Identity Center-Instanz konfiguriert hat, können Sie eine lokale Instanz in demselben Konto wie Ihre S3 Access Grants-Instanz erstellen. Eine solche Konfiguration ist für proof-of-concept Anwendungsfälle in der lokalen Entwicklung üblicher. In allen Fällen muss sich die IAM Identity Center-Instanz in derselben Umgebung AWS-Region befinden wie die S3 Access Grants-Instanz, der sie zugeordnet wird.

In der folgenden Abbildung einer IAM Identity Center-Konfiguration mit einem externen IdP ist der IdP so konfiguriert, dass er den Identitätsspeicher vom IdP mit SCIM dem Identitätsspeicher in IAM Identity Center synchronisiert.



Gehen Sie wie folgt vor, um Ihre Unternehmensverzeichnisidentitäten mit S3 Access Grants zu verwenden:

- Richten Sie die automatische Bereitstellung in IAM Identity Center ein, um Benutzer- und Gruppeninformationen von Ihrem IdP mit IAM Identity Center zu synchronisieren.
- Konfigurieren Sie Ihre externe Identitätsquelle in IAM Identity Center als vertrauenswürdigen Token-Aussteller. Weitere Informationen finden Sie unter Vertrauenswürdige Identitätsweitergabe über Anwendungen hinweg im AWS IAM Identity Center -Benutzerhandbuch.
- Verknüpfen Sie Ihre S3 Access Grants-Instanz mit Ihrer IAM Identity Center-Instanz. Sie können dies während der Erstellung Ihrer S3-Access-Grants-Instance durchführen. Wenn Sie Ihre S3-Access-Grants-Instance bereits erstellt haben, finden Sie weitere Informationen unter Ordnen Sie Ihre IAM Identity Center-Instanz zu oder trennen Sie sie..

Zugriff auf S3-Daten über Verzeichnisidentitäten

Angenommen, Sie haben Unternehmensverzeichnisbenutzer, die über eine Unternehmensanwendung auf Ihre S3-Daten zugreifen müssen, z. B. eine Anwendung zur Dokumentenanzeige, die in Ihren externen IdP integriert ist (z. B. Okta), um Benutzer zu authentifizieren. Die Authentifizierung des Benutzers in diesen Anwendungen erfolgt in der Regel über Weiterleitungen im Webbrowser des Benutzers. Da es sich bei den Benutzern im Verzeichnis nicht um IAM Principals handelt, benötigt Ihre Anwendung IAM Anmeldeinformationen, mit denen sie den `GetDataAccess` API Vorgang S3 Access Grants aufrufen kann, um im Namen der Benutzer

Anmeldeinformationen für S3-Daten abzurufen. Im Gegensatz zu IAM Benutzern und Rollen, die die Anmeldeinformationen selbst abrufen, muss Ihre Anwendung die Möglichkeit haben, einen Verzeichnisbenutzer darzustellen, der keiner IAM Rolle zugeordnet ist, sodass der Benutzer Datenzugriff über S3 Access Grants erhalten kann.

Dieser Übergang vom authentifizierten Verzeichnisbenutzer zu einem IAM Anrufer, der im Namen des Verzeichnisbenutzers Anfragen an S3 Access Grants stellen kann, erfolgt durch die Anwendung über die Funktion Trusted Token Issuer von Identity Center. IAM Die Anwendung hat nach der Authentifizierung des Verzeichnisbenutzers ein Identitätstoken vom IdP (z. B. Okta), das den Verzeichnisbenutzer darstellt gemäß Okta. Die Konfiguration des vertrauenswürdigen Token-Ausstellers in IAM Identity Center ermöglicht es der Anwendung, diese auszutauschen Okta Token (das Okta Der Mandant ist als „vertrauenswürdiger Aussteller“ konfiguriert) für ein anderes Identitätstoken von IAM Identity Center, das den Verzeichnisbenutzer innerhalb AWS-Services des Verzeichnisses sicher repräsentiert. Die Datenanwendung übernimmt dann eine IAM Rolle und stellt das Token des Verzeichnisbenutzers aus IAM Identity Center als zusätzlichen Kontext bereit. Die Anwendung kann die resultierende IAM Sitzung verwenden, um S3 Access Grants aufzurufen. Das Token stellt sowohl die Identität der Anwendung (den IAM Prinzipal selbst) als auch die Identität des Verzeichnisbenutzers dar.

Der Hauptschritt dieses Wechsels ist der Token-Austausch. Die Anwendung führt diesen Token-Austausch durch, indem sie den `CreateTokenWithIAM` API Vorgang in IAM Identity Center aufruft. Natürlich ist auch das ein AWS API Anruf und erfordert, dass ein IAM Schulleiter ihn unterschreibt. Der IAM Principal, der diese Anfrage stellt, ist in der Regel eine IAM Rolle, die der Anwendung zugeordnet ist. Wenn die Anwendung beispielsweise auf Amazon ausgeführt wirdEC2, wird die `CreateTokenWithIAM` Anfrage normalerweise von der IAM Rolle ausgeführt, die der EC2 Instance zugeordnet ist, auf der die Anwendung ausgeführt wird. Das Ergebnis eines erfolgreichen `CreateTokenWithIAM` Aufrufs ist ein neues Identitätstoken, das innerhalb erkannt wird AWS-Services.

Der nächste Schritt, bevor die Anwendung im Namen des Verzeichnisbenutzers aufrufen `GetDataAccess` kann, besteht darin, dass die Anwendung eine IAM Sitzung einrichtet, die die Identität des Verzeichnisbenutzers enthält. Die Anwendung tut dies mit einer AWS Security Token Service (AWS STS) `AssumeRole` -Anforderung, die auch das IAM Identity Center-Token für den Verzeichnisbenutzer als zusätzlichen Identitätskontext enthält. Dieser zusätzliche Kontext ermöglicht es IAM Identity Center, die Identität des Verzeichnisbenutzers an den nächsten Schritt weiterzugeben. Die IAM Rolle, die die Anwendung annimmt, ist die Rolle, für die IAM Berechtigungen zum Aufrufen des `GetDataAccess` Vorgangs erforderlich sind.

Nachdem die Anwendung die IAM Rolle des Identitätsträgers mit dem IAM Identity Center-Token für den Verzeichnisbenutzer als zusätzlichen Kontext übernommen hat, verfügt sie nun über alles, was sie benötigt, um im Namen des authentifizierten Verzeichnisbenutzers eine signierte Anfrage zu `GetDataAccess` stellen.

Die Token-Weitergabe basiert auf den folgenden Schritten:

Erstellen Sie eine IAM Identity Center-Anwendung

Erstellen Sie zunächst eine neue Anwendung in IAM Identity Center. Diese Anwendung verwendet eine Vorlage, anhand derer IAM Identity Center ermitteln kann, welche Art von Anwendungseinstellungen Sie verwenden können. Für den Befehl zum Erstellen der Anwendung müssen Sie den Amazon Resource Name (ARN) der IAM Identity Center-Instance, einen Anwendungsnamen und den Anwendungsanbieter angebenARN. Der Anwendungsanbieter ist der SAML oder der OAuth Anwendungsanbieter, den die Anwendung verwendet, um IAM Identity Center aufzurufen.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch eigenen Daten:

```
aws sso-admin create-application \
 --instance-arn "arn:aws:sso:::instance/ssoins-ssoins-1234567890abcdef" \
 --application-provider-arn "arn:aws:sso::aws:applicationProvider/custom" \
 --name MyDataApplication
```

Antwort:

```
{
   "ApplicationArn": "arn:aws:sso::123456789012:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d"
}
```

Erstellen eines vertrauenswürdigen Token-Ausstellers

Jetzt, wo Sie Ihre IAM Identity Center-Anwendung haben, besteht der nächste Schritt darin, einen vertrauenswürdigen Token-Aussteller zu konfigurieren, der verwendet wird, um Ihre `IdToken` Werte von Ihrem IdP gegen IAM Identity Center-Token auszutauschen. In diesem Schritt müssen Sie die folgenden Elemente bereitstellen:

- Der Emittent des Identitätsanbieters URL

- Name des vertrauenswürdigen Token-Ausstellers
- Pfad zum Anforderungsattribut
- Pfad zum Identitätsspeicherattribut
- Die Abrufoption JSON Web Key Set (JWKS)

Der Pfad zum Anforderungsattribut ist das Identitätsanbieterattribut, das für die Zuordnung zum Identitätsspeicherattribut verwendet wird. Normalerweise ist der Pfad zum Anforderungsattribut die E-Mail-Adresse des Benutzers. Sie können jedoch auch andere Attribute für die Zuordnung verwenden.

Erstellen Sie eine Datei mit dem Namen `oidc-configuration.json` und den folgenden Informationen. Wenn Sie diese Datei verwenden möchten, ersetzen Sie *`user input placeholders`* durch eigene Daten.

```
{
  "OidcJwtConfiguration":
     {
      "IssuerUrl": "https://login.microsoftonline.com/a1b2c3d4-abcd-1234-b7d5-
b154440ac123/v2.0",
      "ClaimAttributePath": "preferred_username",
      "IdentityStoreAttributePath": "userName",
      "JwksRetrievalOption": "OPEN_ID_DISCOVERY"
     }
}
```

Führen Sie den folgenden Befehl aus, um den vertrauenswürdigen Token-Aussteller zu erstellen. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws sso-admin create-trusted-token-issuer \
  --instance-arn "arn:aws:sso:::instance/ssoins-1234567890abcdef" \
  --name MyEntraIDTrustedIssuer \
  --trusted-token-issuer-type OIDC_JWT \
  --trusted-token-issuer-configuration file://./oidc-configuration.json
```

Antwort

```
{
  "TrustedTokenIssuerArn": "arn:aws:sso::123456789012:trustedTokenIssuer/
ssoins-1234567890abcdef/tti-43b4a822-1234-1234-1234-a1b2c3d41234"
```

```
}
```

Connect die IAM Identity Center-Anwendung mit dem vertrauenswürdigen Token-Aussteller

Für den vertrauenswürdigen Token-Aussteller sind einige weitere Konfigurationseinstellungen erforderlich, um zu funktionieren. Legen Sie die Zielgruppe fest, der der vertrauenswürdige Token-Aussteller vertrauen soll. Die Zielgruppe ist der Wert im `IdToken`, der durch den Schlüssel identifiziert wird und in den Einstellungen des Identitätsanbieters zu finden ist. Beispielsweise:

```
1234973b-abcd-1234-abcd-345c5a9c1234
```

Erstellen Sie eine Datei mit dem Namen `grant.json` mit dem folgenden Inhalt. Um diese Datei zu verwenden, ändern Sie die Zielgruppe so, dass sie Ihren Identitätsanbieter-Einstellungen entspricht, und geben Sie den vertrauenswürdigen Token-Aussteller anARN, der mit dem vorherigen Befehl zurückgegeben wurde.

```
{
   "JwtBearer":
     {
       "AuthorizedTokenIssuers":
         [
           {
             "TrustedTokenIssuerArn": "arn:aws:sso::123456789012:trustedTokenIssuer/
ssoins-1234567890abcdef/tti-43b4a822-1234-1234-1234-a1b2c3d41234",
               "AuthorizedAudiences":
                 [
                   "1234973b-abcd-1234-abcd-345c5a9c1234"
                 ]
            }
         ]
     }
 }
```

Führen Sie den folgenden Beispielbefehl aus. Zur Verwendung dieses Befehls ersetzen Sie *user input placeholders* durch eigene Informationen.

```
aws sso-admin put-application-grant \
  --application-arn "arn:aws:sso::123456789012:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d" \
  --grant-type "urn:ietf:params:oauth:grant-type:jwt-bearer" \
```

```
  --grant file://./grant.json \
```

Mit diesem Befehl werden die Konfigurationseinstellungen des vertrauenswürdigen Token-Ausstellers so festgelegt, dass er der Zielgruppe in der Datei `grant.json` vertraut und diese Zielgruppe mit der Anwendung verknüpft, die im ersten Schritt für den Austausch von Token des Typs `jwt-bearer` erstellt wurde. Die Zeichenfolge `urn:ietf:params:oauth:grant-type:jwt-bearer` ist keine willkürliche Zeichenfolge. Es ist ein registrierter Namespace in OAuth JSON Web Token (JWT) - Assertion-Profilen. Weitere Informationen zu diesem Namespace finden Sie in 7523. RFC

Verwenden Sie als Nächstes den folgenden Befehl, um festzulegen, welche Bereiche der vertrauenswürdige Token-Aussteller beim Austausch von `IdToken`-Werten Ihres Identitätsanbieters einbezieht. Für S3 Access Grants ist der Wert für den Parameter `--scope` `s3:access_grants:read_write`.

```
aws sso-admin put-application-access-scope \
  --application-arn "arn:aws:sso::111122223333:application/ssoins-
ssoins-111122223333abcdef/apl-abcd1234a1b2c3d" \
  --scope "s3:access_grants:read_write"
```

Der letzte Schritt besteht darin, der IAM Identity Center-Anwendung eine Ressourcenrichtlinie anzuhängen. Diese Richtlinie ermöglicht es Ihrer IAM Anwendungsrolle, Anfragen an den API Vorgang zu stellen `sso-oauth:CreateTokenWithIAM` und die `IdToken` Werte von IAM Identity Center zu empfangen.

Erstellen Sie eine Datei mit dem Namen `authentication-method.json` mit folgendem Inhalt. Ersetzen Sie *123456789012* durch Ihre Konto-ID.

```
{
   "Iam":
       {
         "ActorPolicy":
             {
                "Version": "2012-10-17",
                    "Statement":
                    [
                        {
                           "Effect": "Allow",
                            "Principal":
                            {
                              "AWS": "arn:aws:iam::123456789012:role/webapp"
                            },
```

```
                        "Action": "sso-oauth:CreateTokenWithIAM",
                         "Resource": "*"
                    }
                ]
            }
        }
    }
```

Führen Sie den folgenden Befehl aus, um die Richtlinie an die IAM Identity Center-Anwendung anzuhängen:

```
aws sso-admin put-application-authentication-method \
   --application-arn "arn:aws:sso::123456789012:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d" \
   --authentication-method-type IAM \
   --authentication-method file://./authentication-method.json
```

Damit sind die Konfigurationseinstellungen für die Verwendung von S3 Access Grants mit Verzeichnisbenutzern über eine Webanwendung abgeschlossen. Sie können dieses Setup direkt in der Anwendung testen oder den `CreateTokenWithIAM` API Vorgang aufrufen, indem Sie den folgenden Befehl von einer in der IAM Identity Center-Anwendungsrichtlinie zugelassenen IAM Rolle aus verwenden:

```
aws sso-oidc create-token-with-iam \
   --client-id "arn:aws:sso::123456789012:application/ssoins-ssoins-1234567890abcdef/
apl-abcd1234a1b2c3d"  \
   --grant-type urn:ietf:params:oauth:grant-type:jwt-bearer \
   --assertion IdToken
```

Die Antwort wird sieht ungefähr wie folgt aus:

```
{
    "accessToken": "<suppressed long string to reduce space>",
    "tokenType": "Bearer",
    "expiresIn": 3600,
    "refreshToken": "<suppressed long string to reduce space>",
    "idToken": "<suppressed long string to reduce space>",
    "issuedTokenType": "urn:ietf:params:oauth:token-type:refresh_token",
    "scope": [
      "sts:identity_context",
      "s3:access_grants:read_write",
```

```
      "openid",
      "aws"
    ]
}
```

Wenn Sie den mit Base64 codierten `IdToken` Wert dekodieren, können Sie die Schlüssel-Wert-Paare im Format sehen. JSON Der Schlüssel `sts:identity_context` enthält den Wert, den Ihre Anwendung in der `sts:AssumeRole`-Anforderung senden muss, um die Identitätsinformationen des Verzeichnisbenutzers einzuschließen. Dies ist ein Beispiel für das dekodierte `IdToken`:

```
{
    "aws:identity_store_id": "d-996773e796",
    "sts:identity_context": "AQoJb3JpZ2luX2VjEOTtl;<SUPRESSED>",
    "sub": "83d43802-00b1-7054-db02-f1d683aacba5",
    "aws:instance_account": "123456789012",
    "iss": "https://identitycenter.amazonaws.com/ssoins-1234567890abcdef",
    "sts:audit_context": "AQoJb3JpZ2luX2VjEOT<SUPRESSED>==",
    "aws:identity_store_arn": "arn:aws:identitystore::232642235904:identitystore/
d-996773e796",
    "aud": "abcd12344U0gi7n4Yyp0-WV1LWNlbnRyYWwtMQ",
    "aws:instance_arn": "arn:aws:sso:::instance/ssoins-6987d7fb04cf7a51",
    "aws:credential_id": "EXAMPLEHI5glPh40y9TpApJn8...",
    "act": {
       "sub": "arn:aws:sso::232642235904:trustedTokenIssuer/
ssoins-6987d7fb04cf7a51/43b4a822-1020-7053-3631-cb2d3e28d10e"
    },
    "auth_time": "2023-11-01T20:24:28Z",
    "exp": 1698873868,
    "iat": 1698870268
}
```

Sie können den Wert aus `sts:identity_context` abrufen und diese Informationen in einem `sts:AssumeRole`-Aufruf weitergeben. Das Folgende ist ein CLI Beispiel für die Syntax. Bei der Rolle, die übernommen werden soll, handelt es sich um eine temporäre Rolle mit Berechtigungen für den Aufruf von `s3:GetDataAccess`.

```
aws sts assume-role \
   --role-arn "arn:aws:iam::123456789012:role/temp-role" \
   --role-session-name "TempDirectoryUserRole" \
   --provided-contexts ProviderArn="arn:aws:iam::aws:contextProvider/
IdentityCenter",ContextAssertion="value from sts:identity_context"
```

Sie können nun die bei diesem Anruf erhaltenen Anmeldeinformationen verwenden, um den `s3:GetDataAccess` API Vorgang aufzurufen und die endgültigen Anmeldeinformationen für den Zugriff auf Ihre S3-Ressourcen zu erhalten.

Erste Schritte mit S3 Access Grants

Amazon S3 Access Grants ist eine Amazon-S3-Funktion, die eine skalierbare Lösung für die Zugriffssteuerung für Ihre S3-Daten bereitstellt. S3 Access Grants ist ein Anbieter von S3-Anmeldeinformationen. Das bedeutet, dass Sie Ihre Liste von Gewährungen und deren Zugriffsebenen bei S3 Access Grants registrieren. Wenn Benutzer oder Clients anschließend Zugriff auf Ihre S3-Daten benötigen, fordern sie zunächst Anmeldeinformationen von S3 Access Grants an. Wenn es eine entsprechende Gewährung gibt, die den Zugriff autorisiert, stellt S3 Access Grants temporäre Zugangsdaten mit geringsten Berechtigungen bereit. Die Benutzer oder Clients können dann mit den von S3 Access Grants bereitgestellten Anmeldeinformationen auf Ihre S3-Daten zuzugreifen. Wenn Ihre S3-Datenanforderungen eine komplexe oder umfangreiche Berechtigungskonfiguration erfordern, können Sie mit S3 Access Grants die S3-Datenberechtigungen für Benutzer, Gruppen, Rollen und Anwendungen skalieren.

In den meisten Anwendungsfällen können Sie die Zugriffskontrolle für Ihre S3-Daten verwalten, indem Sie AWS Identity and Access Management (IAM) zusammen mit Bucket-Richtlinien oder IAM identitätsbasierten Richtlinien verwenden.

Wenn es jedoch komplexe Anforderungen an die S3-Zugriffssteuerung gibt, z. B. die folgenden, könnten Sie erheblich von S3 Access Grants profitieren:

- Sie erreichen das Größenlimit von 20 KB für die Bucket-Richtlinie.
- Sie gewähren menschliche Identitäten, zum Beispiel Microsoft Entra ID (früher Azure Active Directory), Okta, oder Ping Benutzer und Gruppen, Zugriff auf S3-Daten für Analysen und Big Data.
- Sie müssen kontoübergreifenden Zugriff gewähren, ohne die IAM Richtlinien häufig aktualisieren zu müssen.
- Ihre Daten sind unstrukturiert und befinden sich eher auf Objektebene, statt ein strukturiertes Zeilen- und Spaltenformat aufzuweisen.

Der Workflow in S3 Access Grants ist wie folgt:

Schritte	Beschreibung
1	Erstellen einer S3-Access-Grants-Instance

Schritte	Beschreibung
	Initiieren Sie zunächst eine S3-Access-Grants-Instance, die Ihre individuellen Zugriffsberechtigungen enthält.
2	Registrieren eines Speicherorts Zweitens registrieren Sie einen S3-Datenstandort (z. B. den Standardspeicherort`s3://`) und geben Sie dann eine Standardrolle an, die S3 Access Grants IAM bei der Bereitstellung des Zugriffs auf den S3-Datenstandort übernimmt. Sie können bestimmten Buckets oder Präfixen auch benutzerdefinierte Speicherorte hinzufügen und diese benutzerdefinierten IAM Rollen zuordnen.
3	Erstellen von Gewährungen Erstellen Sie individuelle Berechtigungsgewährungen. Geben Sie in diesen Berechtigungsgewährungen den registrierten S3-Speicherort, den Umfang des Datenzugriffs innerhalb des Speicherorts, die Identität des Gewährungsempfängers und die Zugriffsebene (`READ`, `WRITE` oder `READWRITE`) an.
4	Anfordern von Zugriff auf S3-Daten Wenn Benutzer, Anwendungen und ich auf S3-Daten zugreifen AWS-Services möchten, stellen sie zunächst eine Zugriffsanfrage. S3 Access Grants legt fest, ob die Anforderung autorisiert werden soll. Wenn es eine entsprechende Genehmigung gibt, die den Zugriff autorisiert, verwendet S3 Access Grants die IAM Rolle des registrierten Standorts, die mit dieser Genehmigung verknüpft ist, um temporäre Anmeldeinformationen an den Antragsteller zurückzugeben.
5	Zugriff auf S3-Daten Anwendungen verwenden für den Zugriff auf S3-Daten die temporären Anmeldeinformationen, die von S3 Access Grants bereitgestellt wurden.

Arbeiten mit S3 Access Grants-Instanzen

Um mit der Verwendung von Amazon S3 Access Grants zu beginnen, erstellen Sie zunächst eine S3-Access-Grants-Instance. Sie können nur eine S3 Access Grants-Instanz AWS-Region pro Konto erstellen. Die S3-Access-Grants-Instance dient als Container für Ihre S3-Access-Grants-Ressourcen, zu denen registrierte Speicherorte und Gewährungen gehören.

Mit S3 Access Grants können Sie Berechtigungsberechtigungen für Ihre S3-Daten für AWS Identity and Access Management (IAM) Benutzer und Rollen erstellen. Wenn Sie Ihr Corporate Identity-Verzeichnis hinzugefügt haben AWS IAM Identity Center, können Sie diese IAM Identity Center-Instanz Ihres Unternehmensverzeichnisses mit Ihrer S3 Access Grants-Instanz verknüpfen. Anschließend können Sie Zugriffsgewährungen für die Benutzer und Gruppen Ihres Unternehmens erstellen. Wenn Sie Ihr Unternehmensverzeichnis noch nicht zu IAM Identity Center hinzugefügt haben, können Sie Ihre S3 Access Grants-Instanz später mit einer IAM Identity Center-Instanz verknüpfen.

Themen

- Erstellen einer S3-Access-Grants-Instance
- Rufen Sie die Details einer S3 Access Grants-Instance ab
- Listen Sie Ihre S3 Access Grants-Instanzen auf
- Ordnen Sie Ihre IAM Identity Center-Instanz zu oder trennen Sie sie.
- Löschen einer S3-Access-Grants-Instance

Erstellen einer S3-Access-Grants-Instance

Um mit der Verwendung von Amazon S3 Access Grants zu beginnen, erstellen Sie zunächst eine S3-Access-Grants-Instance. Sie können nur eine S3 Access Grants-Instanz AWS-Region pro Konto erstellen. Die S3-Access-Grants-Instance dient als Container für Ihre S3-Access-Grants-Ressourcen, zu denen registrierte Speicherorte und Gewährungen gehören.

Mit S3 Access Grants können Sie Berechtigungsberechtigungen für Ihre S3-Daten für AWS Identity and Access Management (IAM) Benutzer und Rollen erstellen. Wenn Sie Ihr Corporate Identity-Verzeichnis hinzugefügt haben AWS IAM Identity Center, können Sie diese IAM Identity Center-Instanz Ihres Unternehmensverzeichnisses mit Ihrer S3 Access Grants-Instanz verknüpfen. Anschließend können Sie Zugriffsgewährungen für die Benutzer und Gruppen Ihres Unternehmens erstellen. Wenn Sie Ihr Unternehmensverzeichnis noch nicht zu IAM Identity Center hinzugefügt

haben, können Sie Ihre S3 Access Grants-Instanz später mit einer IAM Identity Center-Instanz verknüpfen.

Sie können eine S3 Access Grants-Instance mithilfe der Amazon S3-Konsole, der AWS Command Line Interface (AWS CLI), der Amazon S3 REST API und erstellen AWS SDKs.

Verwenden der S3-Konsole

Bevor Sie mit S3 Access Grants Zugriff auf Ihre S3-Daten gewähren können, müssen Sie zunächst eine S3 Access Grants-Instance in derselben Weise AWS-Region wie Ihre S3-Daten erstellen.

Voraussetzungen

Wenn Sie mittels Identitäten aus Ihrem Unternehmensverzeichnis Zugriff auf Ihre S3-Daten gewähren möchten, müssen Sie Ihr Unternehmensidentitätsverzeichnis zu AWS IAM Identity Center hinzufügen. Wenn Sie dazu noch nicht bereit sind, können Sie Ihre S3 Access Grants-Instanz später mit einer IAM Identity Center-Instanz verknüpfen.

So erstellen Sie eine S3-Access-Grants-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Navigationsleiste den Namen der aktuell angezeigten Datei aus AWS-Region. Wählen Sie als Nächstes die Region aus, zu der Sie wechseln möchten.
3. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
4. Wählen Sie auf der Seite S3 Access Grants die Option S3-Access-Grants-Instance erstellen aus.

    a. Überprüfen Sie in Schritt 1 des Assistenten Access-Grants-Instance einrichten, ob Sie die Instance in der aktuellen AWS-Region erstellen. Stellen Sie sicher, dass sich Ihre S3-Daten dort befinden AWS-Region , wo sich Ihre S3-Daten befinden. Sie können AWS-Region pro Konto eine S3 Access Grants-Instanz erstellen.

    b. (Optional) Wenn Sie Ihr Corporate Identity-Verzeichnis hinzugefügt haben AWS IAM Identity Center, können Sie diese IAM Identity Center-Instanz Ihres Unternehmensverzeichnisses mit Ihrer S3 Access Grants-Instanz verknüpfen.

    Wählen Sie dazu IAM Identity Center-Instanz hinzufügen in ***region***. Geben Sie dann den Amazon Resource Name (ARN) der IAM Identity Center-Instanz ein.

Wenn Sie Ihr Unternehmensverzeichnis noch nicht zu IAM Identity Center hinzugefügt haben, können Sie Ihre S3 Access Grants-Instance später mit einer IAM Identity Center-Instance verknüpfen.

c. Um die S3-Access-Grants-Instance zu erstellen, wählen Sie Weiter aus. Informationen zur Registrierung eines Speicherorts finden Sie unter Schritt 2 – Registrieren eines Speicherorts.

5. Wenn Weiter oder S3-Access-Grants-Instance erstellen deaktiviert ist:

   Instance ksnn nicht erstellt werden

   - Möglicherweise gibt es bereits eine S3-Access-Grants-Instance in derselben AWS-Region. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus. Scrollen Sie auf der Seite S3 Access Grants nach unten zum Abschnitt S3-Access-Grants-Instance in Ihrem Konto, um festzustellen, ob es bereits eine Instance gibt.
   - Möglicherweise verfügen Sie nicht über die Berechtigung `s3:CreateAccessGrantsInstance`, die für die Erstellung einer S3-Access-Grants-Instance erforderlich ist. Nehmen Sie Kontakt mit Ihrem Kontoadministrator auf. Zusätzliche Berechtigungen, die erforderlich sind, wenn Sie eine IAM Identity Center-Instanz mit Ihrer S3 Access Grants-Instanz verknüpfen, finden Sie unter CreateAccessGrantsInstance .

Verwenden von AWS CLI

Informationen zur AWS CLI Installation von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre eigenen Informationen.

Example Erstellen einer S3-Access-Grants-Instance

```
aws s3control create-access-grants-instance \
--account-id 111122223333 \
--region us-east-2
```

Antwort:

```
{
    "CreatedAt": "2023-05-31T17:54:07.893000+00:00",
```

```
    "AccessGrantsInstanceId": "default",
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default"
}
```

Mit dem REST API

Sie können Amazon S3 verwenden RESTAPI, um eine S3 Access Grants-Instance zu erstellen. Informationen zur REST API Unterstützung bei der Verwaltung einer S3 Access Grants-Instance finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- AssociateAccessGrantsIdentityCenter
- CreateAccessGrantsInstance
- DeleteAccessGrantsInstance
- DissociateAccessGrantsIdentityCenter
- GetAccessGrantsInstance
- GetAccessGrantsInstanceForPrefix
- GetAccessGrantsInstanceResourcePolicy
- ListAccessGrantsInstances
- PutAccessGrantsInstanceResourcePolicy

Verwenden von AWS SDKs

Dieser Abschnitt enthält ein Beispiel für die Erstellung einer S3 Access Grants-Instanz mithilfe von AWS SDKs.

Java

In diesem Beispiel wird die S3-Access-Grants-Instance erstellt, die als Container für Ihre individuellen Zugriffsgewährungen dient. Sie können jeweils eine S3 Access Grants-Instanz AWS-Region in Ihrem Konto haben. Die Antwort enthält die Instance-ID `default` und einen Amazon-Ressourcennamen (ARN), der für Ihre S3 Access Grants-Instance generiert wurde.

Example Erstellen einer S3-Access-Grants-Instance-Anforderung

```
public void createAccessGrantsInstance() {
```

```
CreateAccessGrantsInstanceRequest createRequest =
 CreateAccessGrantsInstanceRequest.builder().accountId("111122223333").build();
CreateAccessGrantsInstanceResponse createResponse =
 s3Control.createAccessGrantsInstance(createRequest);LOGGER.info("CreateAccessGrantsInstance
 " + createResponse);
}
```

Antwort:

```
CreateAccessGrantsInstanceResponse(
CreatedAt=2023-06-07T01:46:20.507Z,
AccessGrantsInstanceId=default,
AccessGrantsInstanceArn=arn:aws:s3:us-east-2:111122223333:access-grants/default)
```

Rufen Sie die Details einer S3 Access Grants-Instance ab

Sie können die Details Ihrer Amazon S3 Access Grants-Instance in einer bestimmten Instanz abrufen AWS-Region. Sie können die Details Ihrer S3 Access Grants-Instance mithilfe der Amazon S3-Konsole, der AWS Command Line Interface (AWS CLI), der Amazon S3 REST API und der abrufen AWS SDKs.

Verwenden der S3-Konsole

Um die Details einer S3 Access Grants-Instance abzurufen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.
4. Auf der Seite S3 Access Grants werden Ihre S3-Access-Grants-Instances und alle kontoübergreifenden Instances aufgeführt, die mit Ihrem Konto geteilt wurden. Um die Details einer Instance anzuzeigen, wählen Sie Details anzeigen aus.

Verwenden Sie den AWS CLI

Informationen zur AWS CLI Installation von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example – Anzeigen der Details einer S3-Access-Grants-Instance

```
aws s3control get-access-grants-instance \
 --account-id 111122223333 \
 --region us-east-2
```

Antwort:

```
{
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2: 111122223333:access-grants/
default",
    "AccessGrantsInstanceId": "default",
    "CreatedAt": "2023-05-31T17:54:07.893000+00:00"
}
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für die Verwaltung einer S3 Access Grants-Instance finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- GetAccessGrantsInstance
- GetAccessGrantsInstanceForPrefix

Verwenden der AWS SDKs

Dieser Abschnitt enthält Beispiele dafür, wie Sie die Details einer S3 Access Grants-Instanz mithilfe von abrufen können AWS SDKs.

Um die folgenden Beispiele zu verwenden, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Java

Example – Abrufen einer S3-Access-Grants-Instance

```
public void getAccessGrantsInstance() {
```

```
GetAccessGrantsInstanceRequest getRequest = GetAccessGrantsInstanceRequest.builder()
.accountId("111122223333")
.build();
GetAccessGrantsInstanceResponse getResponse =
 s3Control.getAccessGrantsInstance(getRequest);
LOGGER.info("GetAccessGrantsInstanceResponse: " + getResponse);
}
```

Antwort:

```
GetAccessGrantsInstanceResponse(
AccessGrantsInstanceArn=arn:aws:s3:us-east-2: 111122223333:access-grants/default,
CreatedAt=2023-06-07T01:46:20.507Z)
```

Listen Sie Ihre S3 Access Grants-Instanzen auf

Sie können Ihre S3 Access Grants-Instanzen auflisten, einschließlich der Instanzen, die über AWS Resource Access Manager (AWS RAM) mit Ihnen geteilt wurden.

Sie können Ihre S3 Access Grants-Instances auflisten, indem Sie die Amazon S3-Konsole, die AWS Command Line Interface (AWS CLI), die Amazon S3 REST API und die verwenden AWS SDKs.

Verwenden der S3-Konsole

Um Ihre S3 Access Grants-Instances aufzulisten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.
4. Auf der Seite S3 Access Grants werden Ihre S3-Access-Grants-Instances und alle kontoübergreifenden Instances aufgeführt, die mit Ihrem Konto geteilt wurden. Um die Details einer Instance anzuzeigen, wählen Sie Details anzeigen aus.

Verwenden Sie den AWS CLI

Informationen zur AWS CLI Installation von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example – Auflisten aller S3-Access-Grants-Instances für ein Konto

Diese Aktion listet die S3-Access-Grants-Instances für ein Konto auf. Sie können nur eine S3 Access Grants-Instanz pro Instanz haben AWS-Region. Diese Aktion listet auch weitere kontoübergreifende S3-Access-Grants-Instances auf, auf die Ihr Konto Zugriff hat.

```
aws s3control list-access-grants-instances \
 --account-id 111122223333 \
 --region us-east-2
```

Antwort:

```
{
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2: 111122223333:access-grants/
default",
    "AccessGrantsInstanceId": "default",
    "CreatedAt": "2023-05-31T17:54:07.893000+00:00"
}
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für die Verwaltung einer S3 Access Grants-Instance finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- ListAccessGrantsInstances

Verwenden der AWS SDKs

Dieser Abschnitt enthält Beispiele dafür, wie Sie die Details einer S3 Access Grants-Instanz mithilfe von abrufen können AWS SDKs.

Um die folgenden Beispiele zu verwenden, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Java

Example – Auflisten aller S3-Access-Grants-Instances für ein Konto

Diese Aktion listet die S3-Access-Grants-Instances für ein Konto auf. Es kann pro Region nur eine S3-Access-Grants-Instance geben. Diese Aktion kann auch weitere kontoübergreifende S3-Access-Grants-Instances auflisten, auf die Ihr Konto Zugriff hat.

```
public void listAccessGrantsInstances() {
ListAccessGrantsInstancesRequest listRequest =
 ListAccessGrantsInstancesRequest.builder()
.accountId("111122223333")
.build();
ListAccessGrantsInstancesResponse listResponse =
 s3Control.listAccessGrantsInstances(listRequest);
LOGGER.info("ListAccessGrantsInstancesResponse: " + listResponse);
}
```

Antwort:

```
ListAccessGrantsInstancesResponse(
AccessGrantsInstancesList=[
ListAccessGrantsInstanceEntry(
AccessGrantsInstanceId=default,
AccessGrantsInstanceArn=arn:aws:s3:us-east-2:111122223333:access-grants/default,
CreatedAt=2023-06-07T04:28:11.728Z
)
]
)
```

Ordnen Sie Ihre IAM Identity Center-Instanz zu oder trennen Sie sie.

In Amazon S3 Access Grants können Sie die AWS IAM Identity Center Instance Ihres Corporate Identity-Verzeichnisses einer S3 Access Grants-Instance zuordnen. Nachdem Sie dies getan haben, können Sie zusätzlich zu AWS Identity and Access Management (IAM) Benutzern und Rollen Zugriffsberechtigungen für die Benutzer und Gruppen Ihres Unternehmensverzeichnisses erstellen.

Wenn Sie keine Zugriffsberechtigungen mehr für die Benutzer und Gruppen Ihres Unternehmensverzeichnisses erstellen möchten, können Sie Ihre IAM Identity Center-Instanz von Ihrer S3 Access Grants-Instanz trennen.

Sie können eine IAM Identity Center-Instance zuordnen oder die Zuordnung aufheben, indem Sie die Amazon S3-Konsole, die AWS Command Line Interface (AWS CLI), die Amazon S3 REST API und die AWS SDKs verwenden.

Verwenden der S3-Konsole

Bevor Sie Ihre IAM Identity Center-Instance mit Ihrer S3 Access Grants-Instance verknüpfen, müssen Sie Ihr Corporate Identity-Verzeichnis zu IAM Identity Center hinzufügen. Weitere Informationen finden Sie unter the section called “S3 Access Grants und Unternehmensverzeichnisidentitäten”.

Um eine IAM Identity Center-Instanz mit einer S3 Access Grants-Instanz zu verknüpfen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.
4. Wählen Sie Details anzeigen für die Instance aus.
5. Wählen Sie auf der Detailseite im Abschnitt IAMIdentity Center entweder eine Identity Center-Instance hinzufügen oder die Registrierung einer bereits verknüpften IAM IAM Identity Center-Instance aufheben.

Verwenden Sie den AWS CLI

Informationen zur AWS CLI Installation von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example — Ordnen Sie eine IAM Identity Center-Instanz einer S3 Access Grants-Instanz zu

```
aws s3control associate-access-grants-identity-center \
 --account-id 111122223333 \
 --identity-center-arn arn:aws:sso:::instance/ssoins-1234a567bb89012c \
 --profile access-grants-profile \
 --region eu-central-1
```

```
// No response body
```

Example — Trennen Sie eine IAM Identity Center-Instanz von einer S3 Access Grants-Instanz

```
aws s3control dissociate-access-grants-identity-center \
 --account-id 111122223333 \
 --profile access-grants-profile \
 --region eu-central-1

// No response body
```

Verwenden von REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für die Verwaltung der Verknüpfung zwischen einer IAM Identity Center-Instance und einer S3 Access Grants-Instance finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- AssociateAccessGrantsIdentityCenter
- DissociateAccessGrantsIdentityCenter

Löschen einer S3-Access-Grants-Instance

Sie können eine Amazon S3 Access Grants-Instance aus einer AWS-Region in Ihrem Konto löschen. Bevor Sie jedoch eine S3-Access-Grants-Instance löschen können, müssen Sie Folgendes ausführen:

- Löschen Sie alle Ressourcen innerhalb der S3-Access-Grants-Instance, einschließlich aller Gewährungen und Speicherorte. Weitere Informationen finden Sie unter Eine Gewährung löschen und Einen Speicherort löschen.
- Wenn Sie eine AWS IAM Identity Center Instance mit Ihrer S3 Access Grants-Instance verknüpft haben, müssen Sie die Zuordnung zur IAM Identity Center-Instance aufheben. Weitere Informationen finden Sie unter Ihre IAM Identity Center-Instanz zuordnen oder deren Zuordnung aufheben.

> **Important**
>
> Wenn Sie eine S3-Access-Grants-Instance löschen, wird sie dauerhaft gelöscht. Dieser Vorgang kann nicht rückgängig gemacht werden. Alle Gewährungsempfänger, die über die

Gewährungen in dieser S3-Access-Grants-Instance Zugriff erhalten haben, verlieren den Zugriff auf Ihre S3-Daten.

Sie können eine S3 Access Grants-Instance löschen, indem Sie die Amazon S3-Konsole, die AWS Command Line Interface (AWS CLI), die Amazon S3 REST API und die verwenden AWS SDKs.

Verwenden der S3-Konsole

So löschen Sie eine S3-Access-Grants-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.
4. Wählen Sie Details anzeigen für die Instance aus.
5. Wählen Sie oben rechts auf der Instance-Detailseite Instance löschen aus.
6. Wählen Sie im anschließend angezeigten Dialogfeld Löschen aus. Diese Aktion kann nicht mehr rückgängig gemacht werden.

Verwenden Sie den AWS CLI

Informationen zur AWS CLI Installation von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre eigenen Informationen.

> **Note**
>
> Bevor Sie eine S3-Access-Grants-Instance löschen können, müssen Sie zunächst alle Gewährungen und Speicherorte löschen, die in der S3-Access-Grants-Instance erstellt wurden. Wenn Sie Ihrer S3 Access Grants-Instanz eine IAM Identity Center-Instanz zugeordnet haben, müssen Sie sie zuerst trennen.

Example – Löschen einer S3-Access-Grants-Instance

```
aws s3control delete-access-grants-instance \
--account-id 111122223333 \
--profile access-grants-profile \
--region us-east-2 \
--endpoint-url https://s3-control.us-east-2.amazonaws.com \

 // No response body
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für das Löschen einer S3 Access Grants-Instance finden Sie unter DeleteAccessGrantsInstancein der Amazon Simple Storage Service API Reference.

Mit dem AWS SDKs

Dieser Abschnitt enthält Beispiele für das Löschen einer S3 Access Grants-Instanz mithilfe von AWS SDKs.

Wenn Sie das folgende Beispiel verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Java

> **Note**
>
> Bevor Sie eine S3-Access-Grants-Instance löschen können, müssen Sie zunächst alle Gewährungen und Speicherorte löschen, die in der S3-Access-Grants-Instance erstellt wurden. Wenn Sie Ihrer S3 Access Grants-Instanz eine IAM Identity Center-Instanz zugeordnet haben, müssen Sie sie zuerst trennen.

Example – Löschen einer S3-Access-Grants-Instance

```
public void deleteAccessGrantsInstance() {
DeleteAccessGrantsInstanceRequest deleteRequest =
 DeleteAccessGrantsInstanceRequest.builder()
.accountId("111122223333")
.build();
```

```
DeleteAccessGrantsInstanceResponse deleteResponse =
 s3Control.deleteAccessGrantsInstance(deleteRequest);
LOGGER.info("DeleteAccessGrantsInstanceResponse: " + deleteResponse);
}
```

Mit S3 Access Grants-Standorten arbeiten

Nachdem Sie eine Amazon S3 Access Grants-Instance AWS-Region in Ihrem Konto erstellt haben, registrieren Sie einen S3-Standort in dieser Instance. Ein S3 Access Grants-Standort ordnet den standardmäßigen S3-Standort (`s3://`), einen Bucket oder ein Präfix einer Rolle AWS Identity and Access Management (IAM) zu. S3 Access Grants übernimmt diese IAM Rolle, um temporäre Anmeldeinformationen an den Empfänger weiterzugeben, der auf diesen bestimmten Standort zugreift. Sie müssen zuerst mindestens einen Standort in Ihrer S3 Access Grants-Instanz registrieren, bevor Sie eine Zugriffsberechtigung erstellen können.

Sie können einen Standort registrieren, die Details eines Standorts anzeigen, einen Standort bearbeiten und einen Standort löschen.

> **Note**
>
> Nachdem Sie den ersten Standort in Ihrer S3 Access Grants-Instanz registriert haben, enthält Ihre Instance immer noch keine individuellen Zugriffsberechtigungen. Informationen zum Erstellen einer Zugriffsberechtigung finden Sie unterErstellen von Gewährungen.

Themen

- Registrieren eines Speicherorts
- Anzeigen der Details eines registrierten Speicherorts
- Aktualisieren eines registrierten Speicherorts
- Löschen eines registrierten Speicherorts

Registrieren eines Speicherorts

Nachdem Sie eine Amazon S3 Access Grants-Instance AWS-Region in Ihrem Konto erstellt haben, registrieren Sie einen S3-Standort in dieser Instance. Ein S3 Access Grants-Standort ordnet den standardmäßigen S3-Standort (`s3://`), einen Bucket oder ein Präfix einer Rolle AWS Identity and Access Management (IAM) zu. S3 Access Grants übernimmt diese IAM Rolle, um temporäre

Anmeldeinformationen an den Empfänger weiterzugeben, der auf diesen bestimmten Standort zugreift. Sie müssen zuerst mindestens einen Standort in Ihrer S3 Access Grants-Instanz registrieren, bevor Sie eine Zugriffsberechtigung erstellen können.

Empfohlener Anwendungsfall

Wir empfehlen, den Standardstandort (`s3://`) zu registrieren und ihn einer IAM Rolle zuzuordnen. Der Standort am Standard-S3-Pfad (`s3://`) deckt den Zugriff auf all Ihre S3-Buckets in dem AWS-Region Ihres Kontos ab. Wenn Sie eine Zugriffsberechtigung erstellen, können Sie den Umfang der Gewährung auf einen Bucket, ein Präfix oder ein Objekt innerhalb des Standardspeicherorts einschränken.

Komplexe Anwendungsfälle für die Zugriffsverwaltung

Bei komplexeren Anwendungsfällen für die Zugriffsverwaltung müssen Sie möglicherweise mehr als den Standardspeicherort registrieren. Einige Beispiele für solche Anwendungsfälle sind:

- Nehmen wir an, dass der *`amzn-s3-demo-bucket`* ist ein registrierter Standort in Ihrer S3 Access Grants-Instance, dem eine IAM Rolle zugeordnet ist, aber dieser IAM Rolle wird der Zugriff auf ein bestimmtes Präfix innerhalb des Buckets verweigert. In diesem Fall können Sie das Präfix, auf das die IAM Rolle keinen Zugriff hat, als separaten Standort registrieren und diesen Standort einer anderen IAM Rolle mit dem erforderlichen Zugriff zuordnen.
- Angenommen, Sie möchten Zuweisungen einrichten, die den Zugriff nur auf Benutzer innerhalb eines Virtual Private Cloud (VPC) -Endpunkts beschränken. In diesem Fall können Sie einen Standort für einen Bucket registrieren, in dem die IAM Rolle den Zugriff auf den VPC Endpunkt einschränkt. Später, wenn ein Empfänger S3 Access Grants um Anmeldeinformationen bittet, übernimmt S3 Access Grants die IAM Rolle des Standorts, die temporären Anmeldeinformationen zu verkaufen. Diese Anmeldeinformationen verweigern den Zugriff auf den jeweiligen Bucket, sofern sich der Anrufer nicht innerhalb des Endpunkts befindet. VPC Diese Ablehnungsberechtigung wird zusätzlich zu der regulären READ oder in der Gewährung angegebenen READWRITE Berechtigung angewendet. WRITE

Wenn Sie einen Standort registrieren, müssen Sie auch die IAM Rolle angeben, die S3 Access Grants beim Verteilen temporärer Anmeldeinformationen und beim Gültigkeitsbereich der Berechtigungen für einen bestimmten Zuschuss übernimmt.

Wenn Ihr Anwendungsfall erfordert, dass Sie mehrere Standorte in Ihrer S3 Access Grants-Instanz registrieren, können Sie einen der folgenden Standorte registrieren:

S3 URI	IAMRolle	Beschreibung
`s3://`	*Default-IAM-role*	Der Standardspeicherort (`s3://`) enthält alle Buckets in der AWS-Region.
`s3://`*amzn-s3-demo-bucket1* `/`	*IAM-role-For-bucket*	Dieser Speicherort enthält alle Objekte im angegebenen Bucket.
`s3://`*amzn-s3-demo-bucket1* `/`*prefix-name*	*IAM-role-For-prefix*	Dieser Speicherort umfasst alle Objekte im Bucket mit einem Objektschlüsselnamen, der mit diesem Präfix beginnt.

Bevor Sie einen bestimmten Bucket oder ein bestimmtes Präfix registrieren können, stellen Sie sicher, dass Sie Folgendes tun:

- Erstellen Sie einen oder mehrere Buckets mit den Daten, auf die Sie Zugriff gewähren möchten. Diese Buckets müssen sich im selben Verzeichnis befinden AWS-Region wie Ihre S3 Access Grants-Instanz. Weitere Informationen finden Sie unter Bucket erstellen.

  Das Hinzufügen eines Präfixes ist ein optionaler Schritt. Präfixe sind Zeichenketten am Anfang eines Objektschlüsselnamens. Sie können sie sowohl zur Organisation von Objekten in Ihrem Bucket als auch zur Zugriffsverwaltung verwenden. Informationen zum Hinzufügen eines Präfixes zu einem Bucket finden Sie unter Erstellen von Objektschlüsselnamen.
- Erstellen Sie eine IAM Rolle, die berechtigt ist, auf Ihre S3-Daten zuzugreifen, in der AWS-Region. Weitere Informationen finden Sie im AWS IAM Identity Center Benutzerhandbuch unter IAMRollen erstellen.
- Geben Sie in der IAM Rollenvertrauensrichtlinie dem Dienst S3 Access Grants (`access-grants.s3.amazonaws.com`) den Prinzipalzugriff auf die IAM Rolle, die Sie erstellt haben. Dazu können Sie eine JSON Datei erstellen, die die folgenden Anweisungen enthält. Informationen zum Hinzufügen der Vertrauensrichtlinie zu Ihrem Konto finden Sie unter Eine Rolle mithilfe benutzerdefinierter Vertrauensrichtlinien erstellen.

  TestRolePolicy.json

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

```
    {
      "Sid": "Stmt1234567891011",
      "Effect": "Allow",
      "Principal": {
        "Service":"access-grants.s3.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:SetSourceIdentity"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount":"accountId",
          "aws:SourceArn":"arn:aws:s3:region:accountId:access-grants/default"
        }
      }
    },
    //Optionally, for an IAM Identity Center use case, add:
    {
      "Sid": "Stmt1234567891012",
        "Effect": "Allow",
        "Principal": {
          "Service": "access-grants.s3.amazonaws.com"
        },
        "Action": "sts:SetContext",
        "Condition":{
          "StringEquals":{
            "aws:SourceAccount":"accountId",
            "aws:SourceArn":"arn:aws:s3:region:accountId:access-grants/default"
          },
          "ForAllValues:ArnEquals": {
            "sts:RequestContextProviders":"arn:aws:iam::aws:contextProvider/
IdentityCenter"
          }
        }
      }
  ]
}
```

- Erstellen Sie eine IAM Richtlinie, um der von Ihnen erstellten IAM Rolle Amazon S3 S3-Berechtigungen zuzuweisen. Schauen Sie sich die folgende Beispieldatei `iam-policy.json` an und ersetzen Sie die *`user input placeholders`* durch eigene Daten.

Note

- Wenn Sie serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln verwenden, um Ihre Daten zu verschlüsseln, enthält das folgende Beispiel die erforderlichen AWS KMS Berechtigungen für die IAM Rolle in der Richtlinie. Wenn Sie diese Funktion nicht verwenden, können Sie diese Berechtigungen aus Ihrer IAM Richtlinie entfernen.
- Sie können die IAM Rolle nur dann auf den Zugriff auf S3-Daten beschränken, wenn die Anmeldeinformationen von S3 Access Grants vergeben werden. Dieses Beispiel zeigt Ihnen, wie Sie eine `Condition` Anweisung für eine bestimmte S3 Access Grants-Instanz hinzufügen. Um dies zu verwenden`Condition`, ersetzen Sie die S3 Access Grants-Instanz ARN in der `Condition` Anweisung durch Ihre S3 Access Grants-InstanzARN, die das folgende Format hat: `arn:aws:s3:`*`region`*`:`*`accountId`*`:access-grants/default`

iam-policy.json

```
{
   "Version":"2012-10-17",
   "Statement": [
       {
         "Sid": "ObjectLevelReadPermissions",
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:GetObjectAcl",
            "s3:GetObjectVersionAcl",
            "s3:ListMultipartUploadParts"
         ],
         "Resource":[
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "accountId" },
            "ArnEquals": {
```

```
                "s3:AccessGrantsInstanceArn": ["arn:aws:s3:region:accountId:access-
grants/default"]
            }
        }
      },
      {
         "Sid": "ObjectLevelWritePermissions",
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:PutObjectVersionAcl",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion",
            "s3:AbortMultipartUpload"
         ],
         "Resource":[
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "accountId" },
            "ArnEquals": {
                "s3:AccessGrantsInstanceArn": ["arn:aws:s3:AWS-
Region:accountId:access-grants/default"]
            }
         }
      },
      {
         "Sid": "BucketLevelReadPermissions",
         "Effect":"Allow",
         "Action":[
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "accountId" },
            "ArnEquals": {
                "s3:AccessGrantsInstanceArn": ["arn:aws:s3:AWS-
Region:accountId:access-grants/default"]
            }
         }
      },
```

```
   //Optionally add the following section if you use SSE-KMS encryption
      {
         "Sid": "KMSPermissions",
         "Effect":"Allow",
         "Action":[
            "kms:Decrypt",
            "kms:GenerateDataKey"
         ],
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Sie können einen Standort in Ihrer S3 Access Grants-Instance registrieren, indem Sie die Amazon S3-Konsole, die AWS Command Line Interface (AWS CLI), die Amazon S3 REST API oder die verwenden AWS SDKs.

> **Note**
>
> Nachdem Sie den ersten Standort in Ihrer S3 Access Grants-Instance registriert haben, enthält Ihre Instance immer noch keine individuellen Zugriffsberechtigungen. Informationen zum Erstellen einer Zugriffsberechtigung finden Sie unterErstellen von Gewährungen.

Verwenden der S3-Konsole

Bevor Sie mit S3 Access Grants Zugriff auf Ihre S3-Daten gewähren können, müssen Sie über mindestens einen registrierten Speicherort verfügen.

So registrieren Sie einen Speicherort in Ihrer S3-Access-Grants-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.

Wenn Sie die S3-Access-Grants-Instance zum ersten Mal verwenden, müssen Sie Schritt 1 – Erstellen einer S3-Access-Grants-Instance abgeschlossen haben und zu Schritt 2 des Assistenten Access-Grants-Instance einrichten gewechselt sein. Wenn Sie bereits über eine S3-Access-Grants-Instance verfügen, wählen Sie Details anzeigen und dann auf der Registerkarte Speicherorte die Option Speicherort registrieren aus.

a. Wählen Sie für den Bereich Standort die Option S3 durchsuchen oder geben Sie den URI S3-Pfad zu dem Standort ein, den Sie registrieren möchten. Informationen zu URI S3-Formaten finden Sie in der Tabelle mit den Speicherformaten. Nachdem Sie einen eingegeben habenURI, können Sie Ansicht wählen, um den Speicherort zu durchsuchen.

b. Wählen Sie für die IAMRolle eine der folgenden Optionen aus:

   - Wählen Sie aus vorhandenen IAM Rollen

     Wählen Sie eine IAM Rolle aus der Drop-down-Liste aus. Wählen Sie nach der Auswahl einer Rolle Anzeigen aus, um sicherzustellen, dass diese Rolle die notwendigen Berechtigungen für die Verwaltung des Speicherorts besitzt, den Sie registrieren. Stellen Sie insbesondere sicher, dass diese Rolle S3 Access Grants die Berechtigungen `sts:AssumeRole` und `sts:SetSourceIdentity` gewährt.

   - Geben Sie die Rolle ein IAM ARN

     Navigieren Sie zur IAMKonsole. Kopieren Sie den Amazon-Ressourcennamen (ARN) der IAM Rolle und fügen Sie ihn in dieses Feld ein.

c. Wählen Sie Weiter oder Speicherort registrieren aus, um dies abzuschließen.

4. Fehlerbehebung

   Speicherort kann nicht registriert werden

   - Der Speicherort ist möglicherweise bereits registriert.

     Möglicherweise besitzen Sie nicht die Berechtigung `s3:CreateAccessGrantsLocation` für die Registrierung von Speicherorten. Nehmen Sie Kontakt mit Ihrem Kontoadministrator auf.

Verwenden Sie den AWS CLI

Informationen zur AWS CLI Installation von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Sie können in Ihrer S3-Access-Grants-Instance den Standardspeicherort (`s3://`) oder einen benutzerdefinierten Speicherort registrieren. Stellen Sie sicher, dass Sie zuerst eine IAM Rolle mit Hauptzugriff auf den Standort erstellen und dann sicherstellen, dass Sie S3 Access Grants die Berechtigung erteilen, diese Rolle anzunehmen.

Um die folgenden Beispielbefehle zu verwenden, ersetzen Sie *`user input placeholders`* durch eigene Daten.

Example Erstellen einer Ressourcenrichtlinie

Erstellen Sie eine Richtlinie, die es S3 Access Grants ermöglicht, die IAM Rolle zu übernehmen. Dazu können Sie eine JSON Datei erstellen, die die folgenden Anweisungen enthält. Informationen zum Hinzufügen der Ressourcenrichtlinie zu Ihrem Konto finden Sie unter Erstellen und Anfügen Ihrer ersten vom Kunden verwalteten Richtlinie.

TestRolePolicy.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1234567891011",
      "Action": ["sts:AssumeRole", "sts:SetSourceIdentity"],
      "Effect": "Allow",
      "Principal": {"Service":"access-grants.s3.amazonaws.com"}
    }
  ]
}
```

Example Erstellen der Rolle

Führen Sie den folgenden IAM Befehl aus, um die Rolle zu erstellen.

```
aws iam create-role --role-name accessGrantsTestRole \
 --region us-east-2 \
 --assume-role-policy-document file://TestRolePolicy.json
```

Wenn Sie den Befehl `create-role` ausführen, wird die Richtlinie zurückgegeben:

```
{
    "Role": {
```

```
        "Path": "/",
        "RoleName": "accessGrantsTestRole",
        "RoleId": "AROASRDGX4WM4GH55GIDA",
        "Arn": "arn:aws:iam::111122223333:role/accessGrantsTestRole",
        "CreateDate": "2023-05-31T18:11:06+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "Stmt1685556427189",
                    "Action": [
                        "sts:AssumeRole",
                        "sts:SetSourceIdentity"
                    ],
                    "Effect": "Allow",
                    "Principal": {
                        "Service":"access-grants.s3.amazonaws.com"
                    }
                }
            ]
        }
    }
}
```

Example

Erstellen Sie eine IAM Richtlinie, um Amazon S3 S3-Berechtigungen an die IAM Rolle anzuhängen. Schauen Sie sich die folgende Beispieldatei `iam-policy.json` an und ersetzen Sie die *`user input placeholders`* durch eigene Daten.

> **Note**
>
> Wenn Sie serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln verwenden, um Ihre Daten zu verschlüsseln, fügt das folgende Beispiel der Richtlinie die erforderlichen AWS KMS Berechtigungen für die IAM Rolle hinzu. Wenn Sie diese Funktion nicht verwenden, können Sie diese Berechtigungen aus Ihrer IAM Richtlinie entfernen.
>
> Um sicherzustellen, dass die IAM Rolle nur für den Zugriff auf Daten in S3 verwendet werden kann, wenn die Anmeldeinformationen von S3 Access Grants ausgegeben werden, zeigt Ihnen dieses Beispiel, wie Sie Ihrer IAM Richtlinie eine `Condition` Anweisung hinzufügen, die die S3 Access Grants-Instanz (`s3:AccessGrantsInstance: `*`InstanceArn`*) angibt.

Wenn Sie das folgende Beispiel verwenden, ersetzen Sie die *user input placeholders* durch eigene Daten.

iam-policy.json

```
{
   "Version":"2012-10-17",
   "Statement": [
       {
         "Sid": "ObjectLevelReadPermissions",
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:GetObjectAcl",
            "s3:GetObjectVersionAcl",
            "s3:ListMultipartUploadParts"
         ],
         "Resource":[
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "accountId" },
            "ArnEquals": {
                "s3:AccessGrantsInstanceArn": ["arn:aws:s3:region:accountId:access-
grants/default"]
            }
        }
      },
      {
         "Sid": "ObjectLevelWritePermissions",
         "Effect":"Allow",
         "Action":[
            "s3:PutObject",
            "s3:PutObjectAcl",
            "s3:PutObjectVersionAcl",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion",
            "s3:AbortMultipartUpload"
         ],
         "Resource":[
```

```
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "accountId" },
            "ArnEquals": {
                "s3:AccessGrantsInstanceArn": ["arn:aws:s3:AWS-Region:accountId:access-
grants/default"]
            }
         }
      },
      {
         "Sid": "BucketLevelReadPermissions",
         "Effect":"Allow",
         "Action":[
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::*"
         ],
         "Condition":{
            "StringEquals": { "aws:ResourceAccount": "accountId" },
            "ArnEquals": {
                "s3:AccessGrantsInstanceArn": ["arn:aws:s3:AWS-Region:accountId:access-
grants/default"]
            }
         }
      },
      {
         "Sid": "KMSPermissions",
         "Effect":"Allow",
         "Action":[
            "kms:Decrypt",
            "kms:GenerateDataKey"
         ],
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Example

Führen Sie den folgenden Befehl aus:

```
aws iam put-role-policy \
--role-name accessGrantsTestRole \
--policy-name accessGrantsTestRole \
--policy-document file://iam-policy.json
```

Example Registrieren des Standardspeicherorts

```
aws s3control create-access-grants-location \
 --account-id 111122223333 \
 --location-scope s3:// \
 --iam-role-arn arn:aws:iam::111122223333:role/accessGrantsTestRole
```

Antwort:

```
{"CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "default",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default/location/default",
    "LocationScope": "s3://”
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
}
```

Example Registrieren eines benutzerdefinierten Speicherorts

```
aws s3control create-access-grants-location \
 --account-id 111122223333 \
 --location-scope s3://DOC-BUCKET-EXAMPLE/ \
 --iam-role-arn arn:aws:iam::123456789012:role/accessGrantsTestRole
```

Antwort:

```
{"CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "635f1139-1af2-4e43-8131-a4de006eb456",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2: 111122223333:access-grants/
default/location/635f1139-1af2-4e43-8131-a4de006eb888",
    "LocationScope": "s3://DOC-BUCKET-EXAMPLE/",
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
}
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für die Verwaltung einer S3 Access Grants-Instance finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- CreateAccessGrantsLocation
- DeleteAccessGrantsLocation
- GetAccessGrantsLocation
- ListAccessGrantsLocations
- UpdateAccessGrantsLocation

Verwenden von AWS SDKs

Dieser Abschnitt enthält Beispiele für die Registrierung von Standorten mithilfe von AWS SDKs.

Um die folgenden Beispiele zu verwenden, ersetzen Sie die *`user input placeholders`* durch Ihre eigenen Informationen.

Java

Sie können in Ihrer S3-Access-Grants-Instance den Standardspeicherort (`s3://`) oder einen benutzerdefinierten Speicherort registrieren. Stellen Sie sicher, dass Sie zuerst eine IAM Rolle mit Hauptzugriff auf den Standort erstellen und dann sicherstellen, dass Sie S3 Access Grants die Berechtigung erteilen, diese Rolle anzunehmen.

Um die folgenden Beispielbefehle zu verwenden, ersetzen Sie *`user input placeholders`* durch eigene Daten.

Example Registrieren eines Standardspeicherorts

Anfrage:

```
public void createAccessGrantsLocation() {
CreateAccessGrantsLocationRequest createRequest =
 CreateAccessGrantsLocationRequest.builder()
.accountId("111122223333")
.locationScope("s3://")
.iamRoleArn("arn:aws:iam::123456789012:role/accessGrantsTestRole")
.build();
```

```
CreateAccessGrantsLocationResponse createResponse =
 s3Control.createAccessGrantsLocation(createRequest);
LOGGER.info("CreateAccessGrantsLocationResponse: " + createResponse);
}
```

Antwort:

```
CreateAccessGrantsLocationResponse(
CreatedAt=2023-06-07T04:35:11.027Z,
AccessGrantsLocationId=default,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
location/default,
LocationScope=s3://,
IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)
```

Example Registrieren eines benutzerdefinierten Speicherorts

Anfrage:

```
public void createAccessGrantsLocation() {
CreateAccessGrantsLocationRequest createRequest =
 CreateAccessGrantsLocationRequest.builder()
.accountId("111122223333")
.locationScope("s3://DOC-BUCKET-EXAMPLE/")
.iamRoleArn("arn:aws:iam::111122223333:role/accessGrantsTestRole")
.build();
CreateAccessGrantsLocationResponse createResponse =
 s3Control.createAccessGrantsLocation(createRequest);
LOGGER.info("CreateAccessGrantsLocationResponse: " + createResponse);
}
```

Antwort:

```
CreateAccessGrantsLocationResponse(
CreatedAt=2023-06-07T04:35:10.027Z,
AccessGrantsLocationId=18cfe6fb-eb5a-4ac5-aba9-8d79f04c2012,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
location/18cfe6fb-eb5a-4ac5-aba9-8d79f04c2666,
LocationScope= s3://test-bucket-access-grants-user123/,
IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)
```

Anzeigen der Details eines registrierten Speicherorts

Sie können die Details eines Standorts abrufen, der in Ihrer S3 Access Grants-Instance registriert ist, indem Sie die Amazon S3-Konsole, die AWS Command Line Interface (AWS CLI), die Amazon S3 REST API und die verwenden AWS SDKs.

Verwenden der S3-Konsole

So zeigen Sie die die in Ihrer S3-Access-Grants-Instance gespeicherten Speicherorte an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.
4. Wählen Sie Details anzeigen für die Instance aus.
5. Wählen Sie auf der Detailseite für die Instance die Registerkarte Speicherorte aus.
6. Suchen Sie den registrierten Speicherort, den Sie anzeigen möchten. Verwenden Sie das Suchfeld, um die Liste der registrierten Speicherorte zu filtern.

Verwenden Sie den AWS CLI

Informationen zur AWS CLI Installation von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre eigenen Informationen.

Example – Abrufen der Details eines registrierten Speicherorts

```
aws s3control get-access-grants-location \
--account-id 111122223333 \
--access-grants-location-id default
```

Antwort:

```
{
```

```
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "default",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default/location/default",
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
}
```

Example – Auflisten aller in einer S3-Access-Grants-Instance registrierten Speicherorte

Um die Ergebnisse auf ein S3-Präfix oder einen S3-Bucket einzuschränken, können Sie optional den Parameter `--location-scope s3://bucket-and-or-prefix` verwenden.

```
aws s3control list-access-grants-locations \
--account-id 111122223333 \
--region us-east-2
```

Antwort:

```
{"AccessGrantsLocationsList": [
  {
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "default",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default/location/default",
    "LocationScope": "s3://"
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
     },
  {
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "635f1139-1af2-4e43-8131-a4de006eb456",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default/location/635f1139-1af2-4e43-8131-a4de006eb888",
    "LocationScope": "s3://amzn-s3-demo-bucket/prefixA*",
    "IAMRoleArn": "arn:aws:iam::111122223333:role/accessGrantsTestRole"
     }
   ]
  }
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung zum Abrufen der Details eines registrierten Standorts oder zum Auflisten aller Standorte, die bei einer S3 Access Grants-Instance

registriert sind, finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- GetAccessGrantsLocation
- ListAccessGrantsLocations

Unter Verwendung der AWS SDKs

Dieser Abschnitt enthält Beispiele dafür, wie Sie die Details eines registrierten Standorts abrufen oder alle registrierten Standorte in einer S3 Access Grants-Instanz mithilfe von auflisten können AWS SDKs.

Um die folgenden Beispiele zu verwenden, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Java

Example – Abrufen der Details eines registrierten Speicherorts

```
public void getAccessGrantsLocation() {
GetAccessGrantsLocationRequest getAccessGrantsLocationRequest =
 GetAccessGrantsLocationRequest.builder()
.accountId("111122223333")
.accessGrantsLocationId("default")
.build();
GetAccessGrantsLocationResponse getAccessGrantsLocationResponse =
 s3Control.getAccessGrantsLocation(getAccessGrantsLocationRequest);
LOGGER.info("GetAccessGrantsLocationResponse: " + getAccessGrantsLocationResponse);
}
```

Antwort:

```
GetAccessGrantsLocationResponse(
CreatedAt=2023-06-07T04:35:10.027Z,
AccessGrantsLocationId=default,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
location/default,
LocationScope= s3://,
IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)
```

Example – Auflisten aller in einer S3-Access-Grants-Instance registrierten Speicherorte

Um die Ergebnisse auf ein S3-Präfix oder einen S3-Bucket zu beschränken, können Sie optional ein S3URI, z. B.`s3://bucket-and-or-prefix`, im `LocationScope` Parameter übergeben.

```
public void listAccessGrantsLocations() {

ListAccessGrantsLocationsRequest listRequest =
 ListAccessGrantsLocationsRequest.builder()
.accountId("111122223333")
.build();

ListAccessGrantsLocationsResponse listResponse =
 s3Control.listAccessGrantsLocations(listRequest);
LOGGER.info("ListAccessGrantsLocationsResponse: " + listResponse);
}
```

Antwort:

```
ListAccessGrantsLocationsResponse(
AccessGrantsLocationsList=[
ListAccessGrantsLocationsEntry(
CreatedAt=2023-06-07T04:35:11.027Z,
AccessGrantsLocationId=default,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
location/default,
LocationScope=s3://,
IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
),
ListAccessGrantsLocationsEntry(
CreatedAt=2023-06-07T04:35:10.027Z,
AccessGrantsLocationId=635f1139-1af2-4e43-8131-a4de006eb456,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
location/635f1139-1af2-4e43-8131-a4de006eb888,
LocationScope=s3://amzn-s3-demo-bucket/prefixA*,
IAMRoleArn=arn:aws:iam::111122223333:role/accessGrantsTestRole
)
]
)
```

Aktualisieren eines registrierten Speicherorts

Sie können die Rolle AWS Identity and Access Management (IAM) eines Standorts aktualisieren, der in Ihrer Amazon S3 Access Grants-Instance registriert ist. Stellen Sie sicher, dass Sie für jede neue IAM Rolle, die Sie zur Registrierung eines Standorts in S3 Access Grants verwenden, dem Dienstprinzipal (`access-grants.s3.amazonaws.com`) von S3 Access Grants Zugriff auf diese Rolle gewähren. Fügen Sie dazu einen Eintrag für die neue IAM Rolle in derselben JSON Vertrauensrichtliniendatei hinzu, die Sie bei der ersten Registrierung des Standorts verwendet haben.

Sie können einen Standort in Ihrer S3 Access Grants-Instance aktualisieren, indem Sie die Amazon S3-Konsole, die AWS Command Line Interface (AWS CLI), die Amazon S3 REST API und die verwenden AWS SDKs.

Verwenden der S3-Konsole

Um die IAM Rolle eines Standorts zu aktualisieren, der bei Ihrer S3 Access Grants-Instance registriert ist

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.
4. Wählen Sie Details anzeigen für die Instance aus.
5. Wählen Sie auf der Detailseite für die Instance die Registerkarte Speicherorte aus.
6. Suchen Sie den Speicherort aus, den Sie aktualisieren möchten. Verwenden Sie das Suchfeld, um die Liste der Speicherorte zu filtern.
7. Wählen Sie die Optionsschaltfläche neben dem registrierten Speicherort aus, den Sie aktualisieren möchten.
8. Aktualisieren Sie die IAM Rolle und wählen Sie dann Änderungen speichern.

Verwenden Sie den AWS CLI

Informationen zur AWS CLI Installation von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre eigenen Informationen.

Example — Aktualisieren Sie die IAM Rolle eines registrierten Standorts

```
aws s3control update-access-grants-location \
--account-id 111122223333 \
--access-grants-location-id 635f1139-1af2-4e43-8131-a4de006eb999 \
--iam-role-arn arn:aws:iam::777788889999:role/accessGrantsTestRole
```

Antwort:

```
{
    "CreatedAt": "2023-05-31T18:23:48.107000+00:00",
    "AccessGrantsLocationId": "635f1139-1af2-4e43-8131-a4de006eb999",
    "AccessGrantsLocationArn": "arn:aws:s3:us-east-2:777788889999:access-grants/
default/location/635f1139-1af2-4e43-8131-a4de006eb888",
    "LocationScope": "s3://amzn-s3-demo-bucket/prefixB*",
    "IAMRoleArn": "arn:aws:iam::777788889999:role/accessGrantsTestRole"
}
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für die Aktualisierung eines Standorts in einer S3 Access Grants-Instance finden Sie unter UpdateAccessGrantsLocationin der Amazon Simple Storage Service API Reference.

Mit dem AWS SDKs

Dieser Abschnitt enthält Beispiele dafür, wie Sie die IAM Rolle eines registrierten Standorts mithilfe von aktualisieren können AWS SDKs.

Wenn Sie das folgende Beispiel verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Java

Example — Aktualisieren Sie die IAM Rolle eines registrierten Standorts

```
public void updateAccessGrantsLocation() {
UpdateAccessGrantsLocationRequest updateRequest =
 UpdateAccessGrantsLocationRequest.builder()
.accountId("111122223333")
.accessGrantsLocationId("635f1139-1af2-4e43-8131-a4de006eb999")
.iamRoleArn("arn:aws:iam::777788889999:role/accessGrantsTestRole")
```

```
.build();
UpdateAccessGrantsLocationResponse updateResponse =
 s3Control.updateAccessGrantsLocation(updateRequest);
LOGGER.info("UpdateAccessGrantsLocationResponse: " + updateResponse);
}
```

Antwort:

```
UpdateAccessGrantsLocationResponse(
CreatedAt=2023-06-07T04:35:10.027Z,
AccessGrantsLocationId=635f1139-1af2-4e43-8131-a4de006eb999,
AccessGrantsLocationArn=arn:aws:s3:us-east-2:777788889999:access-grants/default/
location/635f1139-1af2-4e43-8131-a4de006eb888,
LocationScope=s3://amzn-s3-demo-bucket/prefixB*,
IAMRoleArn=arn:aws:iam::777788889999:role/accessGrantsTestRole
)
```

Löschen eines registrierten Speicherorts

Sie können eine Speicherortregistrierung aus einer Amazon-S3-Access-Grants-Instance löschen. Durch das Löschen des Speicherorts wird dessen Registrierung in der S3-Access-Grants-Instance aufgehoben.

Bevor Sie eine Speicherortregistrierung aus einer S3-Access-Grants-Instance entfernen können, müssen Sie alle mit diesem Speicherort verknüpften Gewährungen löschen. Informationen zum Löschen von Gewährungen finden Sie unter Löschen einer Gewährung.

Sie können einen Standort in Ihrer S3 Access Grants-Instance löschen, indem Sie die Amazon S3-Konsole, die AWS Command Line Interface (AWS CLI), die Amazon S3 REST API und die verwenden AWS SDKs.

Verwenden der S3-Konsole

So löschen Sie eine Speicherortregistrierung aus Ihrer Amazon-S3-Access-Grants-Instance

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.

4. Wählen Sie Details anzeigen für die Instance aus.
5. Wählen Sie auf der Detailseite für die Instance die Registerkarte Speicherorte aus.
6. Suchen Sie den Speicherort aus, den Sie aktualisieren möchten. Verwenden Sie das Suchfeld, um die Liste der Speicherorte zu filtern.
7. Wählen Sie die Optionsschaltfläche neben dem registrierten Speicherort aus, den Sie löschen möchten.
8. Wählen Sie Deregister.
9. Anschließend wird ein Dialogfeld angezeigt, in dem Sie gewarnt werden, dass diese Aktion nicht rückgängig gemacht werden kann. Um den Speicherort zu löschen, wählen Sie Registrierung aufheben aus.

Verwenden Sie den AWS CLI

Informationen zur AWS CLI Installation von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example – Löschen einer Speicherortregistrierung

```
aws s3control delete-access-grants-location \
--account-id 111122223333 \
--access-grants-location-id  a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
 // No response body
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für das Löschen eines Standorts aus einer S3 Access Grants-Instance finden Sie unter DeleteAccessGrantsLocationin der Amazon Simple Storage Service API Reference.

Mit dem AWS SDKs

Dieser Abschnitt enthält ein Beispiel dafür, wie Sie einen Standort mithilfe von löschen AWS SDKs.

Wenn Sie das folgende Beispiel verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Java

Example – Löschen einer Speicherortregistrierung

```
public void deleteAccessGrantsLocation() {
DeleteAccessGrantsLocationRequest deleteRequest =
 DeleteAccessGrantsLocationRequest.builder()
.accountId("111122223333")
.accessGrantsLocationId("a1b2c3d4-5678-90ab-cdef-EXAMPLE11111")
.build();
DeleteAccessGrantsLocationResponse deleteResponse =
 s3Control.deleteAccessGrantsLocation(deleteRequest);
LOGGER.info("DeleteAccessGrantsLocationResponse: " + deleteResponse);
}
```

Antwort:

```
DeleteAccessGrantsLocationResponse()
```

Arbeiten mit Zuschüssen in S3 Access Grants

Eine individuelle Zugriffsberechtigung in einer S3 Access Grants-Instanz ermöglicht es einer bestimmten Identität — einem AWS Identity and Access Management (IAM) Prinzipal oder einem Benutzer oder einer Gruppe in einem Unternehmensverzeichnis —, Zugriff von einem Standort aus zu erhalten, der in Ihrer S3 Access Grants-Instanz registriert ist. Ein Standort ordnet Buckets oder Präfixe einer Rolle zu. IAM S3 Access Grants übernimmt diese IAM Rolle, um temporäre Anmeldeinformationen an Empfänger zu verkaufen.

Nachdem Sie mindestens einen Standort in Ihrer S3 Access Grants-Instanz registriert haben, können Sie eine Zugriffsberechtigung erstellen.

Der Empfänger kann ein IAM Benutzer oder eine Rolle oder ein Verzeichnisbenutzer oder eine Gruppe sein. Ein Verzeichnisbenutzer ist ein Benutzer aus Ihrem Unternehmensverzeichnis oder einer externen Identitätsquelle, den Sie mit Ihrer S3 Access Grants-Instanz verknüpft haben. Weitere Informationen finden Sie unter S3 Access Grants und Unternehmensverzeichnisidentitäten. Um einen Zuschuss für einen bestimmten Verzeichnisbenutzer oder eine bestimmte Verzeichnisgruppe aus IAM Identity Center zu erstellen, suchen Sie in IAM Identity Center nach demGUID, anhand dessen IAM Identity Center diesen Benutzer identifiziert, `a1b2c3d4-5678-90ab-cdef-EXAMPLE11111` z. B. Weitere Informationen zur Verwendung von IAM Identity Center zum Anzeigen von

Benutzerinformationen finden Sie im Benutzerhandbuch unter Benutzer- und Gruppenzuweisungen anzeigen.AWS IAM Identity Center

Sie können Zugriff auf einen Bucket, ein Präfix oder ein Objekt gewähren. Ein Präfix in Amazon S3 ist eine Zeichenfolge am Anfang eines Objektschlüsselnamens, mit der die Objekte in einem Bucket organisiert werden. Dabei kann es sich um eine beliebige Zeichenfolge mit zulässigen Zeichen handeln, z. B. Objektschlüsselnamen in Ihrem Bucket, die mit dem `engineering/` Präfix beginnen.

Themen

- Erstellen von Gewährungen
- Anzeigen einer Gewährung
- Löschen einer Gewährung

Erstellen von Gewährungen

Eine individuelle Zugriffsberechtigung in einer S3 Access Grants-Instanz ermöglicht es einer bestimmten Identität — einem AWS Identity and Access Management (IAM) Prinzipal oder einem Benutzer oder einer Gruppe in einem Unternehmensverzeichnis —, Zugriff von einem Standort aus zu erhalten, der in Ihrer S3 Access Grants-Instanz registriert ist. Ein Standort ordnet Buckets oder Präfixe einer Rolle zu. IAM S3 Access Grants übernimmt diese IAM Rolle, um temporäre Anmeldeinformationen an Empfänger zu verkaufen.

Nachdem Sie mindestens einen Standort in Ihrer S3 Access Grants-Instanz registriert haben, können Sie eine Zugriffsberechtigung erstellen.

Der Empfänger kann ein IAM Benutzer oder eine Rolle oder ein Verzeichnisbenutzer oder eine Gruppe sein. Ein Verzeichnisbenutzer ist ein Benutzer aus Ihrem Unternehmensverzeichnis oder einer externen Identitätsquelle, den Sie mit Ihrer S3 Access Grants-Instanz verknüpft haben. Weitere Informationen finden Sie unter S3 Access Grants und Unternehmensverzeichnisidentitäten. Um einen Zuschuss für einen bestimmten Verzeichnisbenutzer oder eine bestimmte Verzeichnisgruppe aus IAM Identity Center zu erstellen, suchen Sie in IAM Identity Center nach demGUID, anhand dessen IAM Identity Center diesen Benutzer identifiziert, `a1b2c3d4-5678-90ab-cdef-EXAMPLE11111` z. B. Weitere Informationen zur Verwendung von IAM Identity Center zum Anzeigen von Benutzerinformationen finden Sie im Benutzerhandbuch unter Benutzer- und Gruppenzuweisungen anzeigen.AWS IAM Identity Center

Sie können Zugriff auf einen Bucket, ein Präfix oder ein Objekt gewähren. Ein Präfix in Amazon S3 ist eine Zeichenfolge am Anfang eines Objektschlüsselnamens, mit der die Objekte in einem Bucket

organisiert werden. Dabei kann es sich um eine beliebige Zeichenfolge mit zulässigen Zeichen handeln, z. B. Objektschlüsselnamen in Ihrem Bucket, die mit dem `engineering/` Präfix beginnen.

Unterpräfix

Wenn Sie Zugriff auf einen registrierten Standort gewähren, können Sie das `Subprefix` Feld verwenden, um den Zugriffsumfang auf eine Teilmenge des Standortbereichs einzuschränken. Wenn es sich bei dem registrierten Standort, den Sie für die Gewährung auswählen, um den Standard-S3-Pfad (`s3://`) handelt, müssen Sie den Geltungsbereich der Gewährung einschränken. Sie können keine Zugriffsberechtigung für den Standardspeicherort (`s3://`) erstellen, wodurch der Empfänger Zugriff auf jeden Bucket in einem AWS-Region erhalten würde. Stattdessen müssen Sie den Umfang der Gewährung auf einen der folgenden Bereiche einschränken:

- Ein Eimer: `s3://`*`bucket`*`/*`
- Ein Präfix innerhalb eines Buckets: `s3://`*`bucket`*`/`*`prefix`*`*`
- Ein Präfix innerhalb eines Präfixes: `s3://`*`bucket`*`/`*`prefixA`*`/`*`prefixB`*`*`
- Ein Objekt: `s3://`*`bucket`*`/`*`object-key-name`*

Wenn Sie eine Zugriffsberechtigung erstellen, bei der es sich bei dem registrierten Standort um einen Bucket handelt, können Sie eine der folgenden Angaben in das `Subprefix` Feld eingeben, um den Umfang der Gewährung einzuschränken:

- Ein Präfix innerhalb des Buckets: *`prefix`*`*`
- Ein Präfix innerhalb eines Präfixes: *`prefixA/prefixB`*`*`
- Ein Objekt: `/`*`object-key-name`*

Nachdem Sie den Zuschuss erstellt haben, ist der Umfang der Gewährung, der in der Amazon S3 S3-Konsole angezeigt wird oder der `GrantScope` in der Antwort API oder AWS Command Line Interface (AWS CLI) zurückgegeben wird, das Ergebnis der Verkettung des Standortpfads mit dem. `Subprefix` Dieser verkettete Pfad muss mit dem S3-Bucket, Präfix oder Objekt übereinstimmen, dem Sie Zugriff gewähren möchten.

> **Note**
>
> - Wenn Sie eine Zugriffserteilung erstellen müssen, die nur Zugriff auf ein Objekt gewährt, müssen Sie angeben, dass der Gewährungstyp für ein Objekt gilt. Um dies in einem API

Aufruf oder einem CLI Befehl zu tun, übergeben Sie den `s3PrefixType` Parameter mit dem Wert`Object`. Wenn Sie in der Amazon S3 S3-Konsole den Grant erstellen, nachdem Sie einen Standort ausgewählt haben, aktivieren Sie unter Grant Scope das Kontrollkästchen Grant scope is an object.

- Sie können keine Gewährung für einen Bucket erstellen, der noch nicht vorhanden ist. Sie können jedoch einen Zuschuss für ein Präfix erstellen, das noch nicht existiert.
- Die maximale Anzahl von Zuschüssen, die Sie in Ihrer S3 Access Grants-Instanz erstellen können, finden Sie unterLimits für S3 Access Grants.

Sie können eine Zugriffserteilung mithilfe der Amazon S3-Konsole AWS CLI, Amazon S3 REST API und erstellen AWS SDKs.

Verwenden der S3-Konsole

So erstellen Sie eine Zugriffsgewährung

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.

   Wenn Sie die S3-Access-Grants-Instance zum ersten Mal verwenden, müssen Sie Schritt 2 – Registrieren eines Speicherorts abgeschlossen haben und zu Schritt 3 des Assistenten Access-Grants-Instance einrichten gewechselt sein. Wenn Sie bereits über eine S3-Access-Grants-Instance verfügen, wählen Sie Details anzeigen und dann auf der Registerkarte Gewährungen die Option Gewährung erstellen aus.

   a. Wählen Sie im Abschnitt Gewährungsumfang einen registrierten Standort aus oder geben Sie diesen ein.

      Wenn Sie den `s3://`-Standardspeicherort ausgewählt haben, können Sie im Feld Unterpräfix den Umfang der Zugriffsgewährung einschränken. Weitere Informationen finden Sie unter Unterpräfix. Wenn Sie lediglich einem Objekt Zugriff gewähren, wählen Sie Gewährungsumfang ist ein Objekt aus.

   b. Wählen Sie unter Berechtigungen und Zugriff die Stufe der Berechtigung aus, Lesen, Schreiben oder beides.

Wählen Sie dann den Empfängertyp aus. Wenn Sie Ihr Unternehmensverzeichnis zu IAM Identity Center hinzugefügt und diese IAM Identity Center-Instance mit Ihrer S3 Access Grants-Instance verknüpft haben, können Sie Verzeichnisidentität aus IAM Identity Center auswählen. Wenn Sie diese Option wählen, rufen Sie die ID des Benutzers oder der Gruppe aus IAM Identity Center ab und geben Sie sie in diesen Abschnitt ein.

Wenn der Typ des Empfängers ein IAM Benutzer oder eine Rolle ist, wählen Sie IAMPrincipal. Wählen Sie unter IAMPrinzipaltyp die Option Benutzer oder Rolle aus. Wählen Sie dann unter IAMHauptbenutzer entweder aus der Liste aus, oder geben Sie die ID der Identität ein.

c. Um die S3-Access-Grants-Gewährung zu erstellen, wählen Sie Weiter oder Gewährung erstellen aus.

4. Wenn Weiter oder Gewährung erstellen deaktiviert ist:

Gewährung kann nicht erstellt werden

- Möglicherweise müssen Sie in Ihrer S3-Access-Grants-Instance zuerst einen Speicherort registrieren.
- Möglicherweise besitzen Sie die Berechtigung `s3:CreateAccessGrant` nicht, die für die Erstellung Zugriffsgewährung erforderlich ist. Nehmen Sie Kontakt mit Ihrem Kontoadministrator auf.

Verwenden Sie den AWS CLI

Informationen zur AWS CLI Installation von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Die folgenden Beispiele zeigen, wie Sie eine Zugriffserteilungsanforderung für einen IAM Prinzipal und eine Zugriffserteilungsanforderung für einen Benutzer oder eine Gruppe von Unternehmensverzeichnissen erstellen.

Um die folgenden Beispielbefehle zu verwenden, ersetzen Sie *`user input placeholders`* durch eigene Daten.

> **Note**
>
> Wenn Sie eine Zugriffsgewährung erstellen, die Zugriff auf ein einzelnes Objekt gewährt, geben Sie den erforderlichen Parameter `--s3-prefix-type Object` an.

Example Erstellen Sie einen Antrag auf Zugriffsgewährung für einen IAM Principal

```
aws s3control create-access-grant \
--account-id 111122223333 \
--access-grants-location-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 \
--access-grants-location-configuration S3SubPrefix=prefixB* \
--permission READ \
--grantee GranteeType=IAM,GranteeIdentifier=arn:aws:iam::123456789012:user/data-consumer-3
```

Example Erstellen einer Zugriffsgewährungsantwort

```
{"CreatedAt": "2023-05-31T18:41:34.663000+00:00",
    "AccessGrantId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/grant/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Grantee": {
        "GranteeType": "IAM",
        "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-3"
    },
    "AccessGrantsLocationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "AccessGrantsLocationConfiguration": {
        "S3SubPrefix": "prefixB*"
    },
    "GrantScope": "s3://DOC-BUCKET-EXAMPLE/prefix*",
    "Permission": "READ"
}
```

Erstellen einer Anforderung für eine Zugriffsgewährung für einen Verzeichnisbenutzer oder eine Verzeichnisgruppe

Um eine Zugriffserteilungsanforderung für einen Verzeichnisbenutzer oder eine Verzeichnisgruppe zu erstellen, müssen Sie zuerst die GUID für den Verzeichnisbenutzer oder die Verzeichnisgruppe abrufen, indem Sie einen der folgenden Befehle ausführen.

Example Rufen Sie eine GUID für einen Verzeichnisbenutzer oder eine Verzeichnisgruppe ab

Sie können den Wert GUID eines IAM Identity Center-Benutzers über die IAM Identity Center-Konsole oder mithilfe von AWS CLI oder ermitteln AWS SDKs. Der folgende Befehl listet die Benutzer in der angegebenen IAM Identity Center-Instanz mit ihren Namen und Kennungen auf.

```
aws identitystore list-users --identity-store-id d-1a2b3c4d1234
```

Dieser Befehl listet die Gruppen in der angegebenen IAM Identity Center-Instanz auf.

```
aws identitystore list-groups --identity-store-id d-1a2b3c4d1234
```

Example Erstellen einer Zugriffsgewährung für einen Verzeichnisbenutzer oder eine Verzeichnisgruppe

Dieser Befehl ähnelt der Erstellung eines Zuschusses für IAM Benutzer oder Rollen, mit der Ausnahme, dass der Typ des Empfängers `DIRECTORY_USER` oder `DIRECTORY_GROUP` ist und die Empfänger-ID die GUID für den Verzeichnisbenutzer oder die Gruppe ist.

```
aws s3control create-access-grant \
--account-id 123456789012 \
--access-grants-location-id default \
--access-grants-location-configuration S3SubPrefix="amzn-s3-demo-bucket/rafael/*" \
--permission READWRITE \
--grantee GranteeType=DIRECTORY_USER,GranteeIdentifier=83d43802-00b1-7054-db02-
f1d683aacba5 \
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für die Verwaltung von Zugriffsberechtigungen finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- CreateAccessGrant
- DeleteAccessGrant
- GetAccessGrant
- ListAccessGrants

Unter Verwendung der AWS SDKs

Dieser Abschnitt enthält Beispiele dafür, wie Sie mithilfe von eine Zugriffsberechtigung erstellen AWS SDKs.

Java

Wenn Sie das folgende Beispiel verwenden möchten, ersetzen Sie *`user input placeholders`* durch eigene Daten.

> **Note**
>
> Wenn Sie eine Zugriffsgewährung erstellen, die Zugriff auf ein einzelnes Objekt gewährt, geben Sie den erforderlichen Parameter `.s3PrefixType(S3PrefixType.Object)` an.

Example Erstellen einer Anforderung für eine Zugriffsgewährung

```
public void createAccessGrant() {
CreateAccessGrantRequest createRequest = CreateAccessGrantRequest.builder()
.accountId("111122223333")
.accessGrantsLocationId("a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa")
.permission("READ")
.accessGrantsLocationConfiguration(AccessGrantsLocationConfiguration.builder().s3SubPrefix("
.grantee(Grantee.builder().granteeType("IAM").granteeIdentifier("arn:aws:iam::111122223333:u
data-consumer-3").build())
.build();
CreateAccessGrantResponse createResponse =
 s3Control.createAccessGrant(createRequest);
LOGGER.info("CreateAccessGrantResponse: " + createResponse);
}
```

Example Erstellen einer Zugriffsgewährungsantwort

```
CreateAccessGrantResponse(
CreatedAt=2023-06-07T05:20:26.330Z,
AccessGrantId=a1b2c3d4-5678-90ab-cdef-EXAMPLE33333,
AccessGrantArn=arn:aws:s3:us-east-2:444455556666:access-grants/default/grant/
a1b2c3d4-5678-90ab-cdef-EXAMPLE33333,
Grantee=Grantee(
```

```
GranteeType=IAM,
GranteeIdentifier=arn:aws:iam::111122223333:user/data-consumer-3
),
AccessGrantsLocationId=a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa,
AccessGrantsLocationConfiguration=AccessGrantsLocationConfiguration(
S3SubPrefix=prefixB*
),
GrantScope=s3://DOC-BUCKET-EXAMPLE/prefixB,
Permission=READ
)
```

Anzeigen einer Gewährung

Sie können die Details einer Zugriffserteilung in Ihrer Amazon S3 Access Grants-Instance anzeigen, indem Sie die Amazon S3-Konsole, die AWS Command Line Interface (AWS CLI), die Amazon S3 REST API und die verwenden AWS SDKs.

Verwenden der S3-Konsole

So zeigen Sie die Details einer Zugriffsgewährung an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.
4. Wählen Sie Details anzeigen für die Instance aus.
5. Wählen Sie auf der Detailseite die Registerkarte Gewährungen aus.
6. Suchen Sie im Bereich Gewährungen nach der Zugriffsgewährung, die Sie anzeigen möchten. Sie können die Liste der Gewährungen über das Suchfeld filtern.

Verwenden Sie den AWS CLI

Informationen zur AWS CLI Installation von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Um die folgenden Beispielbefehle zu verwenden, ersetzen Sie *user input placeholders* durch eigene Daten.

Example – Abrufen der Details einer Zugriffsgewährung

```
aws s3control get-access-grant \
--account-id 111122223333 \
--access-grant-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

Antwort:

```
{
    "CreatedAt": "2023-05-31T18:41:34.663000+00:00",
    "AccessGrantId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "Grantee": {
        "GranteeType": "IAM",
        "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-3"
    },
    "Permission": "READ",
    "AccessGrantsLocationId": "12a6710f-5af8-41f5-b035-0bc795bf1a2b",
    "AccessGrantsLocationConfiguration": {
        "S3SubPrefix": "prefixB*"
    },
    "GrantScope": "s3://amzn-s3-demo-bucket/"
}
```

Example – Auflisten aller Zugriffsgewährungen in einer S3-Access-Grants-Instance

Sie können optional die folgenden Parameter verwenden, um die Ergebnisse auf ein S3-Präfix oder eine AWS Identity and Access Management (IAM) -Identität zu beschränken:

- Unterpräfix – `--grant-scope s3://bucket-name/prefix*`
- IAMIdentität — `--grantee-type IAM` und `--grantee-identifier arn:aws:iam::123456789000:role/accessGrantsConsumerRole`

```
aws s3control list-access-grants \
--account-id 111122223333
```

Antwort:

```
{
```

```
    "AccessGrantsList": [{"CreatedAt": "2023-06-14T17:54:46.542000+00:00",
            "AccessGrantId": "dd8dd089-b224-4d82-95f6-975b4185bbaa",
            "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/dd8dd089-b224-4d82-95f6-975b4185bbaa",
            "Grantee": {
                "GranteeType": "IAM",
                "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-3"
            },
            "Permission": "READ",
            "AccessGrantsLocationId": "23514a34-ea2e-4ddf-b425-d0d4bfcarda1",
            "GrantScope": "s3://amzn-s3-demo-bucket/prefixA*"
        },
        {"CreatedAt": "2023-06-24T17:54:46.542000+00:00",
            "AccessGrantId": "ee8ee089-b224-4d72-85f6-975b4185a1b2",
            "AccessGrantArn": "arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/ee8ee089-b224-4d72-85f6-975b4185a1b2",
            "Grantee": {
                "GranteeType": "IAM",
                "GranteeIdentifier": "arn:aws:iam::111122223333:user/data-consumer-9"
            },
            "Permission": "READ",
            "AccessGrantsLocationId": "12414a34-ea2e-4ddf-b425-d0d4bfcacao0",
            "GrantScope": "s3://amzn-s3-demo-bucket/prefixB*"
        },

    ]
}
```

Mit dem REST API

Sie können Amazon S3 API S3-Operationen verwenden, um die Details einer Zugriffserteilung einzusehen und alle Zugriffsberechtigungen in einer S3 Access Grants-Instance aufzulisten. Informationen zur REST API Unterstützung bei der Verwaltung von Zugriffsberechtigungen finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- GetAccessGrant
- ListAccessGrants

Verwenden von AWS SDKs

Dieser Abschnitt enthält Beispiele dafür, wie Sie die Details einer Zugriffsgewährung mithilfe von abrufen können AWS SDKs.

Um die folgenden Beispiele zu verwenden, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Java

Example – Abrufen der Details einer Zugriffsgewährung

```
public void getAccessGrant() {
GetAccessGrantRequest getRequest = GetAccessGrantRequest.builder()
.accountId("111122223333")
.accessGrantId("a1b2c3d4-5678-90ab-cdef-EXAMPLE22222")
.build();
GetAccessGrantResponse getResponse = s3Control.getAccessGrant(getRequest);
LOGGER.info("GetAccessGrantResponse: " + getResponse);
}
```

Antwort:

```
GetAccessGrantResponse(
CreatedAt=2023-06-07T05:20:26.330Z,
AccessGrantId=a1b2c3d4-5678-90ab-cdef-EXAMPLE22222,
AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant-fd3a5086-42f7-4b34-9fad-472e2942c70e,
Grantee=Grantee(
GranteeType=IAM,
GranteeIdentifier=arn:aws:iam::111122223333:user/data-consumer-3
),
Permission=READ,
AccessGrantsLocationId=12a6710f-5af8-41f5-b035-0bc795bf1a2b,
AccessGrantsLocationConfiguration=AccessGrantsLocationConfiguration(
S3SubPrefix=prefixB*
),
GrantScope=s3://amzn-s3-demo-bucket/
)
```

Example – Auflisten aller Zugriffsgewährungen in einer S3-Access-Grants-Instance

Sie können diese Parameter optional verwenden, um die Ergebnisse auf ein S3-Präfix oder eine IAM Identität zu beschränken:

- Umfang – `GrantScope=s3://bucket-name/prefix*`

- Gewährungsempfänger – `GranteeType=IAM` und `GranteeIdentifier=arn:aws:iam::111122223333:role/accessGrantsConsumerRole`

```
public void listAccessGrants() {
ListAccessGrantsRequest listRequest = ListAccessGrantsRequest.builder()
.accountId("111122223333")
.build();
ListAccessGrantsResponse listResponse = s3Control.listAccessGrants(listRequest);
LOGGER.info("ListAccessGrantsResponse: " + listResponse);
}
```

Antwort:

```
ListAccessGrantsResponse(
AccessGrantsList=[
ListAccessGrantEntry(
CreatedAt=2023-06-14T17:54:46.540z,
AccessGrantId=dd8dd089-b224-4d82-95f6-975b4185bbaa,
AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/dd8dd089-b224-4d82-95f6-975b4185bbaa,
Grantee=Grantee(
GranteeType=IAM, GranteeIdentifier= arn:aws:iam::111122223333:user/data-consumer-3
),
Permission=READ,
AccessGrantsLocationId=23514a34-ea2e-4ddf-b425-d0d4bfcarda1,
GrantScope=s3://amzn-s3-demo-bucket/prefixA
),
ListAccessGrantEntry(
CreatedAt=2023-06-24T17:54:46.540Z,
AccessGrantId=ee8ee089-b224-4d72-85f6-975b4185a1b2,
AccessGrantArn=arn:aws:s3:us-east-2:111122223333:access-grants/default/
grant/ee8ee089-b224-4d72-85f6-975b4185a1b2,
Grantee=Grantee(
GranteeType=IAM, GranteeIdentifier= arn:aws:iam::111122223333:user/data-consumer-9
),
Permission=READ,
AccessGrantsLocationId=12414a34-ea2e-4ddf-b425-d0d4bfcacao0,
GrantScope=s3://amzn-s3-demo-bucket/prefixB*
)
]
)
```

Löschen einer Gewährung

Sie können Zugriffsgewährungen aus Ihrer Amazon-S3-Access-Grants-Instance löschen. Sie können die Löschung einer Zugriffsgewährung nicht rückgängig machen. Nach dem Löschen einer Zugriffsberechtigung hat der Empfänger nicht länger Zugriff auf Ihre Amazon-S3-Daten.

Sie können eine Zugriffsgewährung löschen, indem Sie die Amazon S3-Konsole, die AWS Command Line Interface (AWS CLI), die Amazon S3 REST API und die verwenden AWS SDKs.

Verwenden der S3-Konsole

So löschen Sie eine Zugriffsgewährung

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Zugriffsgewährungen aus.
3. Wählen Sie auf der Seite S3 Access Grants die Region mit der S3-Access-Grants-Instance aus, mit der Sie arbeiten möchten.
4. Wählen Sie Details anzeigen für die Instance aus.
5. Wählen Sie auf der Detailseite die Registerkarte Gewährungen aus.
6. Suchen Sie die Gewährung, die Sie löschen möchten. Wenn Sie die Gewährung gefunden haben, wählen Sie das Optionsfeld neben der Gewährung aus.
7. Wählen Sie Löschen. Anschließend wird ein Dialogfeld angezeigt, in dem Sie gewarnt werden, dass diese Aktion nicht rückgängig gemacht werden kann. Wählen Sie erneut Löschen aus, um die Gewährung zu löschen.

Verwenden Sie den AWS CLI

Informationen zur AWS CLI Installation von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre eigenen Informationen.

Example – Löschen einer Zugriffsgewährung

```
aws s3control delete-access-grant \
--account-id 111122223333 \
--access-grant-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111

// No response body
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für die Verwaltung von Zugriffsberechtigungen finden Sie unter DeleteAccessGrantin der Amazon Simple Storage Service API Reference.

Mit dem AWS SDKs

Dieser Abschnitt enthält Beispiele dafür, wie Sie eine Zugriffsgewährung mithilfe von löschen können AWS SDKs. Wenn Sie das folgende Beispiel verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

Java

Example – Löschen einer Zugriffsgewährung

```
public void deleteAccessGrant() {
DeleteAccessGrantRequest deleteRequest = DeleteAccessGrantRequest.builder()
.accountId("111122223333")
.accessGrantId("a1b2c3d4-5678-90ab-cdef-EXAMPLE11111")
.build();
DeleteAccessGrantResponse deleteResponse =
 s3Control.deleteAccessGrant(deleteRequest);
LOGGER.info("DeleteAccessGrantResponse: " + deleteResponse);
}
```

Antwort:

```
DeleteAccessGrantResponse()
```

Abrufen von S3-Daten mithilfe von Zugriffsberechtigungen

Stipendiaten, denen über S3 Access Grants Zugriff auf S3-Daten gewährt wurde, müssen temporäre Anmeldeinformationen von S3 Access Grants anfordern, die sie für den Zugriff auf die S3-Daten verwenden. Weitere Informationen finden Sie unter Anfordern des Zugriffs auf Amazon-S3-Daten über S3 Access Grants. Die Empfänger verwenden dann die temporären Anmeldeinformationen, um zulässige S3-Aktionen mit den S3-Daten durchzuführen. Weitere Informationen finden Sie unter Zugriff auf S3-Daten mithilfe von Anmeldeinformationen, die von S3 Access Grants bereitgestellt wurden. Zuschussempfänger können optional eine Liste ihrer Zugangsberechtigungen anfordern, AWS-Konto bevor sie diese Anmeldeinformationen anfordern. Weitere Informationen finden Sie unter Listen Sie die Zugriffsberechtigungen des Anrufers auf.

Themen

- Anfordern des Zugriffs auf Amazon-S3-Daten über S3 Access Grants
- Zugriff auf S3-Daten mithilfe von Anmeldeinformationen, die von S3 Access Grants bereitgestellt wurden
- Listen Sie die Zugriffsberechtigungen des Anrufers auf

Anfordern des Zugriffs auf Amazon-S3-Daten über S3 Access Grants

Nachdem Sie Amazon S3 Access Grants verwendet haben, um eine Zugriffsberechtigung zu erstellen, die AWS Identity and Access Management (IAM) Prinzipalen, Ihren Unternehmensverzeichnisidentitäten oder autorisierten Anwendungen Zugriff auf Ihre S3-Daten gewährt, können Ihre Empfänger Anmeldeinformationen für den Zugriff auf diese Daten anfordern.

Wenn eine Anwendung oder der `GetDataAccess` API Vorgang AWS-Service verwendet, um S3 Access Grants im Namen eines Empfängers um Zugriff auf Ihre S3-Daten zu bitten, überprüft S3 Access Grants zunächst, ob Sie dieser Identität Zugriff auf die Daten gewährt haben. Anschließend verwendet S3 Access Grants AssumeRoleAPIVorgang, um ein temporäres Anmeldeinformationstoken zu erhalten und es an den Anforderer zu senden. Dieses Token für temporäre Anmeldeinformation ist ein AWS Security Token Service (AWS STS)-Token.

Die `GetDataAccess`-Anforderung muss den Parameter `target` enthalten, um den Umfang der S3-Daten anzugeben, für den die temporären Anmeldeinformationen gelten. Dieser `target`-Umfang kann mit dem Umfang der Gewährung oder einem Teil dieses Umfangs identisch sein. Der `target`-Umfang muss jedoch innerhalb des Umfangs der Gewährung für den Anforderer liegen. In der

Anforderung muss auch der Parameter `permission` angegeben werden, der die Berechtigungsstufe für die temporären Anmeldeinformationen angibt, `READ`, `WRITE` oder `READWRITE`.

Der Anforderer kann die Berechtigungsstufe des temporären Tokens in der Anforderung für Anmeldeinformationen angeben. Mit dem Parameter `privilege` kann der Anforderer den Umfang des Zugriffs für die temporären Anmeldeinformationen innerhalb der Grenzen des Gewährungsumfangs reduzieren oder erhöhen. Der Standardwert des Parameters `privilege` ist `Default`. Das bedeutet, dass der Zielumfang der zurückgegebenen Anmeldeinformationen der ursprüngliche Umfang der Gewährung ist. Der andere mögliche Wert für `privilege` ist `Minimal`. Wenn der `target`-Umfang im Vergleich zum ursprünglichen Umfang der Gewährung reduziert wird, wird der Umfang der temporären Anmeldeinformationen an den `target`-Umfang angepasst, solange der `target`-Umfang innerhalb des Gewährungsumfangs liegt.

Die folgende Tabelle zeigt die Auswirkungen des Parameters `privilege` auf zwei Gewährungen. Eine Gewährung hat den Umfang `S3://`*`amzn-s3-demo-bucket1`*`/bob/*`. Dieser Umfang umfasst das gesamte Präfix `bob/` im Bucket *`amzn-s3-demo-bucket1`*. Die andere Gewährung hat den Umfang `S3://`*`amzn-s3-demo-bucket1`*`/bob/reports/*`. Dieser Umfang umfasst nur das Präfix `bob/reports/` im Bucket *`amzn-s3-demo-bucket1`*.

Umfang der Gewährung	Angeforderter Umfang	Recht	Zurückgegebener Umfang	Auswirkung
`S3://`*`amzn-s3-demo-bucket1`*`/bob/*`	*`amzn-s3-demo-bucket1`*`/bob/*`	`Default`	*`amzn-s3-demo-bucket1`*`/bob/*`	Der Anforderer hat Zugriff auf alle Objekte im Bucket *`amzn-s3-demo-bucket1`*, deren Schlüsselnamen mit dem Präfix *`bob/`* beginnen.
`S3://`*`amzn-s3-demo-bucket1`*`/bob/*`	*`amzn-s3-demo-bucket1`*`/bob/`	`Minimal`	*`amzn-s3-demo-bucket1`*`/bob/`	Ohne das Platzhalterzeichen (*) nach dem Präfixnamen `bob/` hat der Anforderer nur Zugriff auf das Objekt mit dem Namen `bob/` im Bucket *`amzn-s3-demo-bucket1`*. Ein

Umfang der Gewährung	Angeforderter Umfang	Recht	Zurückgegebener Umfang	Auswirkung
				solches Objekt ist nicht üblich. Der Anforderer hat keinen Zugriff auf andere Objekte, auch nicht auf Objekte, deren Schlüsselnamen mit dem Präfix bob/ beginnen.
S3://*amzn-s3-demo-bucket1* /bob/*	*amzn-s3-demo-bucket1* /bob/images/*	Minimal	*amzn-s3-demo-bucket1* /bob/images/*	Der Anforderer hat Zugriff auf alle Objekte im Bucket *amzn-s3-demo-bucket1* , deren Schlüsselnamen mit dem Präfix *bob/images/** beginnen.
S3://*amzn-s3-demo-bucket1* /bob/reports/*	*amzn-s3-demo-bucket1* /bob/reports/file.txt	Default	*amzn-s3-demo-bucket1* /bob/reports/*	Der Anforderer hat Zugriff auf alle Objekte im Bucket *amzn-s3-demo-bucket1* , deren Schlüsselnamen mit dem Präfix bob/reports beginnen. Dies entspricht dem Umfang der entsprechenden Gewährung.

Umfang der Gewährung	Angeforderter Umfang	Recht	Zurückgegebener Umfang	Auswirkung
S3://*amzn-s3-demo-bucket1* /bob/reports/*	*amzn-s3-demo-bucket1* /bob/reports/file.txt	Minimal	*amzn-s3-demo-bucket1* /bob/reports/file.txt	Der Anforderer hat nur Zugriff auf das Objekt mit dem Schlüsselnamen bob/reports/file.txt im Bucket *amzn-s3-demo-bucket1* . Der Anforderer hat keinen Zugriff auf andere Objekte.

Der Parameter `durationSeconds` legt die Gültigkeitsdauer der temporären Anmeldeinformationen in Sekunden fest. Der Standardwert ist `3600` Sekunden (1 Stunde). Der Anforderer (Empfänger) kann jedoch einen Bereich von `900` Sekunden (15 Minuten) bis `43200` Sekunden (12 Stunden) angeben. Wenn der Empfänger einen höheren Wert als diesen Höchstwert anfordert, schlägt die Anforderung fehl.

> **Note**
>
> Wenn der Speicherort ein Objekt ist, legen Sie in Ihrer Anforderung eines temporären Tokens den Wert des Parameters `targetType` auf `Object` fest. Dieser Parameter ist nur erforderlich, wenn es sich der Speicherort ein Objekt ist und die Berechtigungsstufe `Minimal` ist. Wenn der Speicherort ein Bucket oder ein Präfix ist, müssen Sie diesen Parameter nicht angeben.

Weitere Informationen finden Sie GetDataAccessin der Amazon Simple Storage Service API Reference.

Sie können temporäre Anmeldeinformationen anfordern, indem Sie AWS Command Line Interface (AWS CLI), Amazon S3 REST API und den verwenden AWS SDKs.

Mit dem AWS CLI

Informationen zur AWS CLI Installation von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example Anfordern temporärer Anmeldeinformationen

Anfrage:

```
aws s3control get-data-access \
--account-id 111122223333 \
--target s3://amzn-s3-demo-bucket/prefixA* \
--permission READ \
--privilege Default \
--region us-east-2
```

Antwort:

```
{
"Credentials": {
 "AccessKeyId": "Example-key-id",
 "SecretAccessKey": "Example-access-key",
 "SessionToken": "Example-session-token",
"Expiration": "2023-06-14T18:56:45+00:00"},
    "MatchedGrantTarget": "s3://amzn-s3-demo-bucket/prefixA**"
}
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für die Anforderung temporärer Anmeldeinformationen von S3 Access Grants finden Sie GetDataAccessin der Amazon Simple Storage Service API Reference.

Mit dem AWS SDKs

Dieser Abschnitt enthält ein Beispiel dafür, wie Stipendiaten temporäre Anmeldeinformationen von S3 Access Grants anfordern, indem sie den AWS SDKs verwenden.

Java

Das folgende Codebeispiel gibt die temporären Anmeldeinformationen zurück, die der Gewährungsempfänger für den Zugriff auf Ihre S3-Daten verwendet. Wenn Sie dieses Codebeispiel verwenden möchten, ersetzen Sie *user input placeholders* durch eigene Daten.

Example Abrufen temporärer Anmeldeinformationen

Anfrage:

```
public void getDataAccess() {
GetDataAccessRequest getDataAccessRequest = GetDataAccessRequest.builder()
.accountId("111122223333")
.permission(Permission.READ)
.privilege(Privilege.MINIMAL)
.target("s3://amzn-s3-demo-bucket/prefixA*")
.build();
GetDataAccessResponse getDataAccessResponse =
 s3Control.getDataAccess(getDataAccessRequest);
LOGGER.info("GetDataAccessResponse: " + getDataAccessResponse);
}
```

Antwort:

```
GetDataAccessResponse(
Credentials=Credentials(
AccessKeyId="Example-access-key-id",
SecretAccessKey="Example-secret-access-key",
SessionToken="Example-session-token",
Expiration=2023-06-07T06:55:24Z
))
```

Zugriff auf S3-Daten mithilfe von Anmeldeinformationen, die von S3 Access Grants bereitgestellt wurden

Nachdem ein Stipendiat im Rahmen seiner Zugriffsgewährung temporäre Anmeldeinformationen erhalten hat, kann er diese temporären Anmeldeinformationen verwenden, um Amazon S3 API S3-Operationen für den Zugriff auf Ihre Daten aufzurufen.

Stipendiaten können mithilfe von AWS Command Line Interface (AWS CLI) AWS SDKs, dem und Amazon S3 REST API auf S3-Daten zugreifen. Darüber hinaus können Sie die AWS Python - und Java-Plugins verwenden, um S3 Access Grants aufzurufen

Mit dem AWS CLI

Nachdem der Gewährungsempfänger seine temporären Anmeldeinformationen von S3 Access Grants erhalten hat, kann er mit diesen Anmeldeinformationen ein Profil einrichten, um die Daten abzurufen.

Informationen zur AWS CLI Installation von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Um die folgenden Beispielbefehle zu verwenden, ersetzen Sie *user input placeholders* durch eigene Daten.

Example – Einrichten eines Profils

```
aws configure set aws_access_key_id "$accessKey" --profile access-grants-consumer-access-profile
aws configure set aws_secret_access_key "$secretKey" --profile access-grants-consumer-access-profile
aws configure set aws_session_token "$sessionToken" --profile access-grants-consumer-access-profile
```

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example – Abrufen der S3-Daten

Der Stipendiat kann das verwenden get-object AWS CLI Befehl, um auf die Daten zuzugreifen. Der Stipendiat kann auch Folgendes verwenden put-object, lsund andere AWS CLI S3-Befehle.

```
aws s3api get-object \
--bucket amzn-s3-demo-bucket1 \
--key myprefix \
--region us-east-2 \
--profile access-grants-consumer-access-profile
```

Mit dem AWS SDKs

Dieser Abschnitt enthält Beispiele dafür, wie Stipendiaten mithilfe von auf Ihre S3-Daten zugreifen können. AWS SDKs

Java

Beispiele für das Abrufen von S3-Daten mithilfe temporärer Anmeldeinformationen finden Sie unter So rufen Sie ein Objekt mithilfe des AWS SDKs und Amazon S3 S3-Codebeispiele für den AWS SDK for Java 2.x.

Unterstützte S3-Aktionen in S3 Access Grants

Ein Empfänger kann die von S3 Access Grants bereitgestellten temporären Anmeldeinformationen verwenden, um S3-Aktionen mit den S3-Daten durchzuführen, auf die er Zugriff hat. Im Folgenden finden Sie eine Liste der zulässigen S3-Aktionen, die ein Empfänger ausführen kann. Welche Aktionen zulässig sind, hängt von der in der Zugriffsgewährung gewährten Berechtigungsstufe ab, entweder`READ`, `WRITE` oder. `READWRITE`

> **Note**
>
> Zusätzlich zu den unten aufgeführten Amazon S3-Berechtigungen kann Amazon S3 die AWS Key Management Service (AWS KMS) Decrypt (`kms:decrypt`) `READ` -Berechtigung oder die AWS KMS GenerateDataKey(`kms:generateDataKey`) `WRITE` -Berechtigung aufrufen. Diese Berechtigungen ermöglichen keinen direkten Zugriff auf den AWS KMS Schlüssel.

IAMS3-Aktion	APIAktion und Dokument	S3 Access erteilt die Erlaubnis	S3-Ressource			
`s3:GetObject`	GetObject	`READ`	Object			
`s3:GetObjectVersion`	GetObject	`READ`	Object			

IAMS3-Aktion	APIAktion und Dokument	S3 Access erteilt die Erlaubnis	S3-Ressource
`s3:GetObjectAcl`	GetObjectAcl	`READ`	Object
`s3:GetObjectVersionAcl`	GetObjectAcl	`READ`	Object
`s3:ListMultipartUploads`	ListParts	`READ`	Object
`s3:PutObject`	PutObject, CreateMultipartUpload, UploadPart, UploadPartCopy, CompleteMultipartUpload	`WRITE`	Object
`s3:PutObjectAcl`	PutObjectAcl	`WRITE`	Object
`s3:PutObjectVersionAcl`	PutObjectAcl	`WRITE`	Object
`s3:DeleteObject`	DeleteObject	`WRITE`	Object
`s3:DeleteObjectVersion`	DeleteObject	`WRITE`	Object

IAMS3-Aktion	APIAktion und Dokument	S3 Access erteilt die Erlaubnis	S3-Ressource			
`s3:AbortMultipartUpload`	AbortMultipartUpload	`WRITE`	Object			
`s3:ListBucket`	HeadBucket, ListObjectsV2, ListObjects	`READ`	Bucket			
`s3:ListBucketVersions`	ListObjectVersions	`READ`	Bucket			
`s3:ListBucketMultipartUploads`	ListMultipartUploads	`READ`	Bucket			

Listen Sie die Zugriffsberechtigungen des Anrufers auf

S3-Datenbesitzer können S3 Access Grants verwenden, um Zugriffsberechtigungen für AWS Identity and Access Management (IAM) -Identitäten oder für AWS IAM Identity Center Unternehmensverzeichnisidentitäten zu erstellen. IAMIdentitäten und IAM Identity Center-Verzeichnisidentitäten können wiederum die verwenden, `ListCallerAccessGrants` API um alle Amazon S3 S3-Buckets, -Präfixe und Objekte aufzulisten, auf die sie zugreifen können, wie in ihren S3 Access Grants definiert. Verwenden Sie diese OptionAPI, um alle S3-Daten zu ermitteln, auf die eine IAM oder eine Verzeichnisidentität über S3 Access Grants zugreifen kann.

Sie können diese Funktion verwenden, um Anwendungen zu erstellen, die die Daten anzeigen, auf die bestimmte Endbenutzer zugreifen können. Beispielsweise verwendet der AWS Storage Browser for S3, eine Open-Source-UI-Komponente, die Kunden für den Zugriff auf S3-Buckets verwenden, diese Funktion, um Endbenutzern die Daten zu präsentieren, auf die sie in Amazon S3 Zugriff haben, basierend auf ihren S3 Access Grants. Ein anderes Beispiel: Wenn Sie eine Anwendung zum Durchsuchen, Hochladen oder Herunterladen von Daten in Amazon S3 erstellen, können Sie diese

Funktion verwenden, um eine Baumstruktur in Ihrer Anwendung zu erstellen, die ein Endbenutzer dann durchsuchen kann.

Der Empfänger, unabhängig davon, ob es sich um eine Identität oder eine Unternehmensverzeichnisidentität handelt, kann mithilfe von AWS Command Line Interface (AWS CLI), Amazon S3 und dem AWS SDKs eine Liste seiner REST API Zugriffsberechtigungen abrufen. IAM

Mit dem AWS CLI

Informationen zur AWS CLI Installation von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

Example Listet die Zugriffsberechtigungen eines Anrufers auf

Anfrage:

```
aws s3control list-caller-access-grants \
--account-id 111122223333 \
--region us-east-2
--max-results 5
```

Antwort:

```
{
 "NextToken": "6J9S...",
 "CallerAccessGrantsList": [
  {
   "Permission": "READWRITE",
   "GrantScope": "s3://amzn-s3-demo-bucket/prefix1/sub-prefix1/*",
   "ApplicationArn": "NA"
  },
  {
   "Permission": "READWRITE",
   "GrantScope": "s3://amzn-s3-demo-bucket/prefix1/sub-prefix2/*",
   "ApplicationArn": "ALL"
  },
```

```
  {
   "Permission": "READWRITE",
   "GrantScope": "s3://amzn-s3-demo-bucket/prefix1/sub-prefix3/*",
   "ApplicationArn": "arn:aws:sso::111122223333:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d"
  }
 ]
}
```

Example Listet die Zugriffsberechtigungen eines Anrufers für einen Bucket auf

Sie können den Umfang der Ergebnisse mithilfe des `grantscope` Parameters einschränken.

Anfrage:

```
aws s3control list-caller-access-grants \
--account-id 111122223333 \
--region us-east-2
--grant-scope "s3://amzn-s3-demo-bucket""
--max-results 1000
```

Antwort:

```
{
 "NextToken": "6J9S...",
 "CallerAccessGrantsList": [
  {
   "Permission": "READ",
   "GrantScope": "s3://amzn-s3-demo-bucket*",
   "ApplicationArn": "ALL"
  },
  {
   "Permission": "READ",
   "GrantScope": "s3://amzn-s3-demo-bucket/prefix1/*",
   "ApplicationArn": "arn:aws:sso::111122223333:application/ssoins-
ssoins-1234567890abcdef/apl-abcd1234a1b2c3d"
  }
 ]
}
```

Mit dem REST API

Informationen zur Amazon S3 REST API S3-Unterstützung für das Abrufen einer Liste der Zugriffsberechtigungen des API Anrufers finden Sie ListCallerAccessGrantsin der Amazon Simple Storage Service API Reference.

Mit dem AWS SDKs

Dieser Abschnitt enthält ein Beispiel dafür, wie Stipendiaten temporäre Anmeldeinformationen von S3 Access Grants anfordern, indem sie den AWS SDKs verwenden.

Java

Das folgende Codebeispiel gibt die Zugriffsberechtigungen des API Anrufers auf die S3-Daten einer bestimmten Person zurück. AWS-Konto Wenn Sie dieses Codebeispiel verwenden möchten, ersetzen Sie *user input placeholders* durch eigene Daten.

Example Listet die Zugriffsberechtigungen eines Anrufers auf

Anfrage:

```
Public void ListCallerAccessGrants() {
 ListCallerAccessGrantsRequest listRequest = ListCallerAccessGrantsRequest.builder()
    .withMaxResults(1000)
    .withGrantScope("s3://")
    .accountId("111122223333");
 ListCallerAccessGrantsResponse listResponse =
 s3control.listCallerAccessGrants(listRequest);
 LOGGER.info("ListCallerAccessGrantsResponse: " + listResponse);
 }
```

Antwort:

```
ListCallerAccessGrantsResponse(
CallerAccessGrantsList=[
 ListCallerAccessGrantsEntry(
  S3Prefix=s3://amzn-s3-demo-bucket/prefix1/,
  Permission=READ,
  ApplicationArn=ALL
 )
])
```

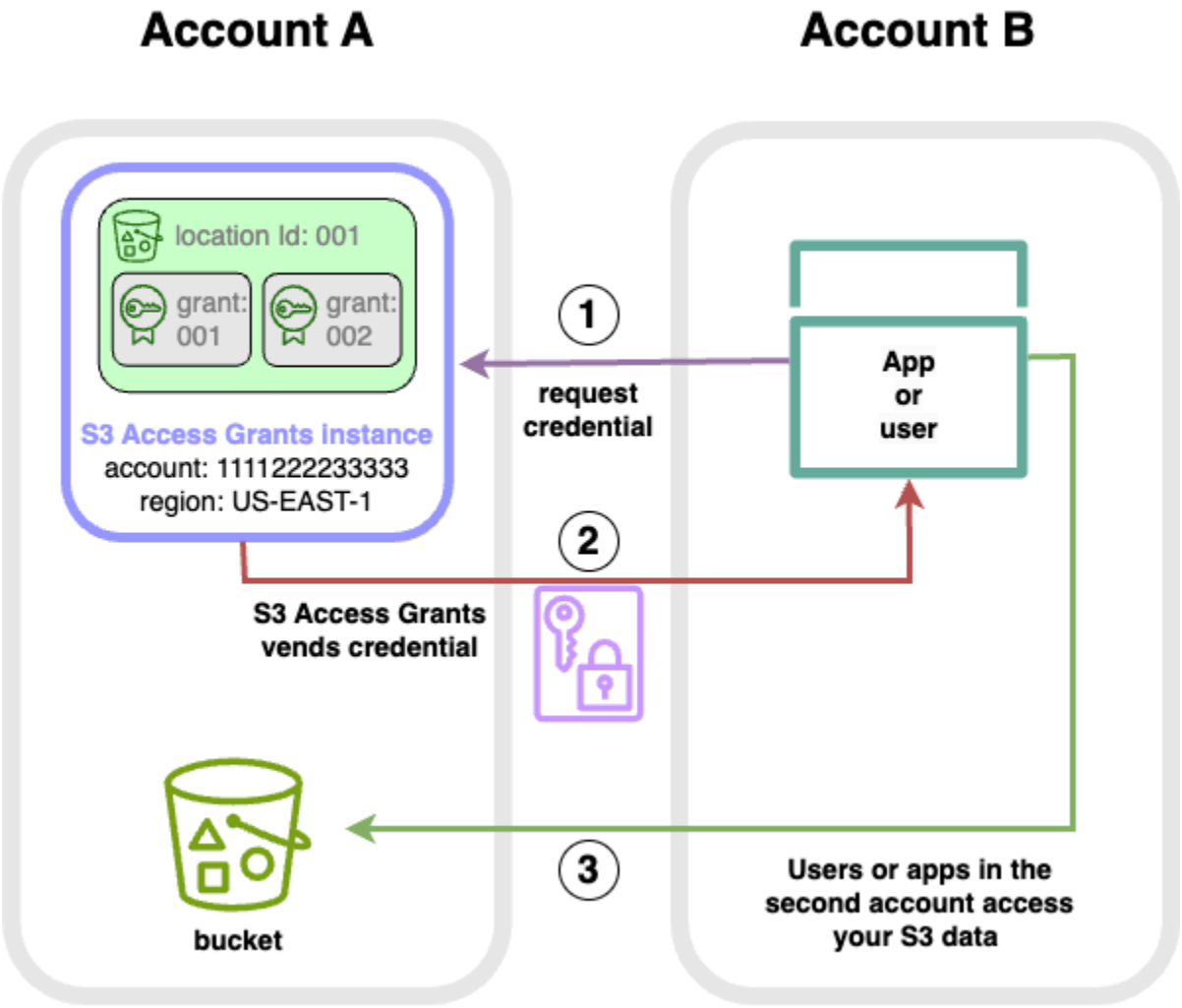
Kontoübergreifender Zugriff mit S3 Access Grants

Mit S3 Access Grants können Sie Amazon S3 S3-Datenzugriff auf Folgendes gewähren:

- AWS Identity and Access Management (IAM) Identitäten innerhalb Ihres Kontos
- IAMIdentitäten in anderen Konten AWS
- Verzeichnisbenutzer oder -gruppen in Ihrer Instanz AWS IAM Identity Center

Konfigurieren Sie zunächst den kontoübergreifenden Zugriff für das andere Konto. Dazu gehört die Gewährung des Zugriffs auf Ihre S3 Access Grants-Instanz mithilfe einer Ressourcenrichtlinie. Gewähren Sie anschließend mithilfe von Zuschüssen Zugriff auf Ihre S3-Daten (Buckets, Präfixe oder Objekte).

Nachdem Sie den kontoübergreifenden Zugriff konfiguriert haben, kann das andere Konto temporäre Zugangsdaten für Ihre Amazon S3 S3-Daten von S3 Access Grants anfordern. Die folgende Abbildung zeigt den Benutzerablauf für den kontoübergreifenden S3-Zugriff über S3 Access Grants:



1. Benutzer oder Anwendungen in einem zweiten Konto (B) fordern Anmeldeinformationen von der S3 Access Grants-Instance in Ihrem Konto (A) an, in der die Amazon S3 S3-Daten gespeichert sind. Weitere Informationen finden Sie unter Anfordern des Zugriffs auf Amazon-S3-Daten über S3 Access Grants.
2. Die S3 Access Grants-Instance in Ihrem Konto (A) gibt temporäre Anmeldeinformationen zurück, wenn es eine Zuweisung gibt, die dem zweiten Konto Zugriff auf Ihre Amazon S3 S3-Daten gewährt. Weitere Informationen zu Zugriffsberechtigungen finden Sie unterArbeiten mit Zuschüssen in S3 Access Grants.
3. Benutzer oder Anwendungen im zweiten Konto (B) verwenden die von S3 Access Grants bereitgestellten Anmeldeinformationen, um auf die S3-Daten in Ihrem Konto (A) zuzugreifen.

Die Konfiguration von S3 Access gewährt kontoübergreifenden Zugriff

Gehen Sie wie folgt vor, um kontoübergreifenden S3-Zugriff über S3 Access Grants zu gewähren:

- Schritt 1: Konfigurieren Sie eine S3 Access Grants-Instanz in Ihrem Konto, z. B. die Konto-ID111122223333, in der die S3-Daten gespeichert sind.
- Schritt 2: Konfigurieren Sie die Ressourcenrichtlinie für die S3 Access Grants-Instanz in Ihrem Konto, 111122223333 um Zugriff auf das zweite Konto zu gewähren, z. B. die Konto-ID444455556666.
- Schritt 3: Konfigurieren Sie die IAM Berechtigungen für den IAM Principal im zweiten Konto444455556666, um Anmeldeinformationen von der S3 Access Grants-Instanz in Ihrem Konto anzufordern111122223333.
- Schritt 4: Erstellen Sie in Ihrem Konto einen Grant111122223333, der dem IAM Principal im zweiten Konto 444455556666 Zugriff auf einige der S3-Daten in Ihrem Konto gewährt111122223333.

Schritt 1: Konfigurieren Sie eine S3 Access Grants-Instanz in Ihrem Konto

Zunächst benötigen Sie eine S3 Access Grants-Instance in Ihrem Konto, 111122223333 um den Zugriff auf Ihre Amazon S3 S3-Daten zu verwalten. Sie müssen jeweils eine S3 Access Grants-Instance erstellen AWS-Region , in der die S3-Daten gespeichert sind, die Sie teilen möchten. Wenn Sie Daten in mehr als einer gemeinsam nutzen AWS-Region, wiederholen Sie jeden dieser Konfigurationsschritte für jeden AWS-Region. Wenn Sie bereits über eine S3 Access Grants-Instanz in dem Bereich verfügen, in AWS-Region dem Ihre S3-Daten gespeichert sind, fahren Sie mit dem nächsten Schritt fort. Wenn Sie keine S3 Access Grants-Instanz konfiguriert haben, finden Sie hier Informationen Arbeiten mit S3 Access Grants-Instanzen zum Abschließen dieses Schritts.

Schritt 2: Konfigurieren Sie die Ressourcenrichtlinie für Ihre S3 Access Grants-Instanz, um kontoübergreifenden Zugriff zu gewähren

Nachdem Sie in Ihrem Konto eine S3 Access Grants-Instanz 111122223333 für den kontoübergreifenden Zugriff erstellt haben, konfigurieren Sie die ressourcenbasierte Richtlinie für die S3 Access Grants-Instanz in Ihrem Konto, um kontenübergreifenden Zugriff 111122223333 zu gewähren. Die S3-Access-Grants-Instance selbst unterstützt ressourcenbasierte Richtlinien. Mit der richtigen ressourcenbasierten Richtlinie können Sie AWS Identity and Access Management (IAM) Benutzern oder Rollen von anderen Benutzern Zugriff auf Ihre S3 Access Grants-Instanz AWS-Konten gewähren. Der kontoübergreifende Zugriff gewährt nur die folgenden Berechtigungen (Aktionen):

- `s3:GetAccessGrantsInstanceForPrefix`— Der Benutzer, die Rolle oder die App kann die S3 Access Grants-Instanz abrufen, die ein bestimmtes Präfix enthält.
- `s3:ListAccessGrants`
- `s3:ListAccessLocations`
- `s3:ListCallerAccessGrants`
- `s3:GetDataAccess`— Der Benutzer, die Rolle oder die App kann temporäre Anmeldeinformationen anfordern, die auf dem Zugriff basieren, den Sie über S3 Access Grants erhalten haben. Mit diesen Anmeldeinformationen können Sie auf die S3-Daten zugreifen, für die Ihnen Zugriff gewährt wurde.

Sie können auswählen, welche dieser Berechtigungen in die Ressourcenrichtlinie aufgenommen werden sollen. Diese Ressourcenrichtlinie auf der S3 Access Grants-Instanz ist eine normale ressourcenbasierte Richtlinie und unterstützt alles, was die IAMRichtliniensprache unterstützt. In derselben Richtlinie können Sie beispielsweise Zugriff auf bestimmte IAM Identitäten in Ihrem Konto gewähren`111122223333`, indem Sie die `aws:PrincipalArn` Bedingung verwenden. Bei S3 Access Grants müssen Sie das jedoch nicht tun. Stattdessen können Sie innerhalb Ihrer S3 Access Grants-Instanz Berechtigungen für einzelne IAM Identitäten von Ihrem Konto aus sowie für das andere Konto erstellen. Indem Sie die einzelnen Zugriffsberechtigungen über S3 Access Grants verwalten, können Sie Ihre Berechtigungen skalieren.

Wenn Sie AWS Resource Access Manager(AWS RAM) bereits verwenden, können Sie es verwenden, um Ihre `s3:AccessGrants`Ressourcen mit anderen Konten oder innerhalb Ihrer Organisation zu teilen. Weitere Informationen finden Sie unter Arbeiten mit gemeinsam genutzten AWS Ressourcen. Wenn Sie die Ressourcenrichtlinie nicht verwenden AWS RAM, können Sie die Ressourcenrichtlinie auch mithilfe der S3 Access API Grants-Operationen oder der Option AWS Command Line Interface (AWS CLI) hinzufügen.

Verwenden der S3-Konsole

Wir empfehlen Ihnen, die Konsole AWS Resource Access Manager (AWS RAM) zu verwenden, um Ihre `s3:AccessGrants` Ressourcen mit anderen Konten oder innerhalb Ihrer Organisation zu teilen. Gehen Sie wie folgt vor, um S3 Access Grants kontenübergreifend zu teilen:

So konfigurieren Sie die Ressourcenrichtlinie für die S3 Access Grants-Instanz:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

2. Wählen Sie das AWS-Region aus der AWS-Region Auswahl aus.
3. Wählen Sie im linken Navigationsbereich Access Grants aus.
4. Wählen Sie auf der Seite Access Grants-Instanz im Abschnitt Instanz in diesem Konto die Option Instanz teilen aus. Dadurch werden Sie zur AWS RAM Konsole weitergeleitet.
5. Wählen Sie Resource Share erstellen aus.
6. Folgen Sie den AWS RAM Schritten, um die Ressourcenfreigabe zu erstellen. Weitere Informationen finden Sie unter Eine Ressourcenfreigabe erstellen in AWS RAM.

Mit dem AWS CLI

Informationen zur AWS CLI Installation von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Sie können die Ressourcenrichtlinie mithilfe des `put-access-grants-instance-resource-policy` CLI Befehls hinzufügen.

Wenn Sie dem zweiten Konto kontoübergreifenden Zugriff für die S3 Access Grants-Instanz in Ihrem Konto `111122223333` gewähren möchten`444455556666`, `111122223333` sollte die Ressourcenrichtlinie für die S3 Access Grants-Instanz in Ihrem Konto dem zweiten Konto die `444455556666` Erlaubnis geben, die folgenden Aktionen auszuführen:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Geben Sie in der Ressourcenrichtlinie für die S3 Access Grants-Instanz das ARN Ihrer S3 Access Grants-Instanz als und das zweite Konto `444455556666` als. `Resource` `Principal` Um das folgende Beispiel zu verwenden, ersetzen Sie das *`user input placeholders`* mit Ihren eigenen Informationen.

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

```
 "AWS": "444455556666"
},
 "Action": [
  "s3:ListAccessGrants",
  "s3:ListAccessGrantsLocations",
  "s3:GetDataAccess",
  "s3:GetAccessGrantsInstanceForPrefix"
 ],
 "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
} ]
}
```

Verwenden Sie den folgenden Befehl, um die Ressourcenrichtlinie für die S3 Access Grants-Instanz hinzuzufügen oder zu aktualisieren. Wenn Sie den folgenden Beispielbefehl verwenden, ersetzen Sie den *user input placeholders* durch Ihre eigenen Informationen.

Example Fügen Sie die Ressourcenrichtlinie für die S3 Access Grants-Instanz hinzu oder aktualisieren Sie sie

```
aws s3control put-access-grants-instance-resource-policy \
--account-id 111122223333 \
--policy file://resourcePolicy.json \
--region us-east-2
{
 "Policy": "{\n
   \"Version\": \"2012-10-17\",\n
   \"Statement\": [{\n
  \"Effect\": \"Allow\",\n
  \"Principal\": {\n
    \"AWS\": \"444455556666\"\n
  },\n
  \"Action\": [\n
    \"s3:ListAccessGrants\",\n
    \"s3:ListAccessGrantsLocations\",\n
    \"s3:GetDataAccess\",\n
    \"s3:GetAccessGrantsInstanceForPrefix\",\n
    \"s3:ListCallerAccessGrants"\n
  ],\n
  \"Resource\": \"arn:aws:s3:us-east-2:111122223333:access-grants/default\"\n
    }\n
   ]\n
   }\n",
```

```
  "CreatedAt": "2023-06-16T00:07:47.473000+00:00"
 }
```

Example Abrufen einer S3-Access-Grants-Ressourcenrichtlinie

Sie können die auch verwendenCLI, um eine Ressourcenrichtlinie für eine S3 Access Grants-Instanz abzurufen oder zu löschen.

Verwenden Sie den folgenden Beispielbefehl, um eine S3 Access Grants-Ressourcenrichtlinie abzurufen. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws s3control get-access-grants-instance-resource-policy \
--account-id 111122223333 \
--region us-east-2

{
"Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":\"arn:aws:iam::111122223333:root\"},\"Action\":
[\"s3:ListAccessGrants\",\"s3:ListAccessGrantsLocations\",\"s3:GetDataAccess\",
\"s3:GetAccessGrantsInstanceForPrefix\",\"s3:ListCallerAccessGrants\"],\"Resource\":
\"arn:aws:
s3:us-east-2:111122223333:access-grants/default\"}]}",
"CreatedAt": "2023-06-16T00:07:47.473000+00:00"
}
```

Example Löschen einer S3-Access-Grants-Ressourcenrichtlinie

Verwenden Sie den folgenden Beispielbefehl, um eine S3 Access Grants-Ressourcenrichtlinie zu löschen. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws s3control delete-access-grants-instance-resource-policy \
--account-id 111122223333 \
--region us-east-2

// No response body
```

Verwenden Sie den REST API

Sie können die Ressourcenrichtlinie hinzufügen, indem Sie den verwenden PutAccessGrantsInstanceResourcePolicy API.

Wenn Sie dem zweiten Konto kontenübergreifenden Zugriff für die S3 Access Grants-Instanz in Ihrem Konto 111122223333 gewähren möchten444455556666, 111122223333 sollte die Ressourcenrichtlinie für die S3 Access Grants-Instanz in Ihrem Konto dem zweiten Konto die 444455556666 Erlaubnis geben, die folgenden Aktionen auszuführen:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Geben Sie in der Ressourcenrichtlinie für die S3 Access Grants-Instanz das ARN Ihrer S3 Access Grants-Instanz als und das zweite Konto 444455556666 als. `Resource` `Principal` Um das folgende Beispiel zu verwenden, ersetzen Sie das *`user input placeholders`* mit Ihren eigenen Informationen.

```
{
"Version": "2012-10-17",
"Statement": [
{
 "Effect": "Allow",
 "Principal": {
 "AWS": "444455556666"
},
 "Action": [
  "s3:ListAccessGrants",
  "s3:ListAccessGrantsLocations",
  "s3:GetDataAccess",
  "s3:GetAccessGrantsInstanceForPrefix"
 ],
 "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
} ]
}
```

Sie können dann den verwenden PutAccessGrantsInstanceResourcePolicy API, um die Richtlinie zu konfigurieren.

Informationen zur REST API Unterstützung beim Aktualisieren, Abrufen oder Löschen einer Ressourcenrichtlinie für eine S3 Access Grants-Instance finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- PutAccessGrantsInstanceResourcePolicy
- GetAccessGrantsInstanceResourcePolicy
- DeleteAccessGrantsInstanceResourcePolicy

Verwenden der AWS SDKs

In diesem Abschnitt finden Sie AWS SDK Beispiele dafür, wie Sie Ihre Ressourcenrichtlinie für S3 Access Grants so konfigurieren, dass einem zweiten AWS Konto Zugriff auf einige Ihrer S3-Daten gewährt wird.

Java

Sie können eine Ressourcenrichtlinie hinzufügen, aktualisieren, abrufen oder löschen, um den kontoübergreifenden Zugriff für Ihre S3-Access-Grants-Instance zu verwalten.

Example Fügen Sie eine Ressourcenrichtlinie für eine S3 Access Grants-Instanz hinzu oder aktualisieren Sie sie

Wenn Sie kontoübergreifenden Zugriff für die S3 Access Grants-Instanz in Ihrem Konto auf das zweite Konto `111122223333` gewähren möchten`444455556666`, `111122223333` sollte die Ressourcenrichtlinie für die S3 Access Grants-Instanz in Ihrem Konto dem zweiten Konto die `444455556666` Erlaubnis geben, die folgenden Aktionen auszuführen:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`

Geben Sie in der Ressourcenrichtlinie für die S3 Access Grants-Instanz das ARN Ihrer S3 Access Grants-Instanz als und das zweite Konto `444455556666` als. `Resource` `Principal` Um das folgende Beispiel zu verwenden, ersetzen Sie das *`user input placeholders`* mit Ihren eigenen Informationen.

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

```
 "Principal": {
 "AWS": "444455556666"
},
 "Action": [
  "s3:ListAccessGrants",
  "s3:ListAccessGrantsLocations",
  "s3:GetDataAccess",
  "s3:GetAccessGrantsInstanceForPrefix"
 ],
 "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
} ]
}
```

Verwenden Sie das folgende Codebeispiel, um eine Ressourcenrichtlinie für eine S3 Access Grants-Instanz hinzuzufügen oder zu aktualisieren:

```
public void putAccessGrantsInstanceResourcePolicy() {
 PutAccessGrantsInstanceResourcePolicyRequest putRequest =
 PutAccessGrantsInstanceResourcePolicyRequest.builder()
 .accountId(111122223333)
 .policy(RESOURCE_POLICY)
 .build();
 PutAccessGrantsInstanceResourcePolicyResponse putResponse =
 s3Control.putAccessGrantsInstanceResourcePolicy(putRequest);
 LOGGER.info("PutAccessGrantsInstanceResourcePolicyResponse: " + putResponse);
 }
```

Antwort:

```
PutAccessGrantsInstanceResourcePolicyResponse(
 Policy={
 "Version": "2012-10-17",
 "Statement": [{
 "Effect": "Allow",
 "Principal": {
 "AWS": "444455556666"
 },
 "Action": [
 "s3:ListAccessGrants",
 "s3:ListAccessGrantsLocations",
 "s3:GetDataAccess",
 "s3:GetAccessGrantsInstanceForPrefix",
 "s3:ListCallerAccessGrants"
```

```
 ],
 "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
 }]
 }
 )
```

Example Abrufen einer S3-Access-Grants-Ressourcenrichtlinie

Verwenden Sie das folgende Codebeispiel, um eine S3 Access Grants-Ressourcenrichtlinie abzurufen. Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

```
public void getAccessGrantsInstanceResourcePolicy() {
 GetAccessGrantsInstanceResourcePolicyRequest getRequest =
 GetAccessGrantsInstanceResourcePolicyRequest.builder()
 .accountId(111122223333)
 .build();
 GetAccessGrantsInstanceResourcePolicyResponse getResponse =
 s3Control.getAccessGrantsInstanceResourcePolicy(getRequest);
 LOGGER.info("GetAccessGrantsInstanceResourcePolicyResponse: " + getResponse);
 }
```

Antwort:

```
GetAccessGrantsInstanceResourcePolicyResponse(
 Policy={"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":
{"AWS":"arn:aws:iam::444455556666:root"},"Action":
["s3:ListAccessGrants","s3:ListAccessGrantsLocations","s3:GetDataAccess","s3:GetAccessGrants
east-2:111122223333:access-grants/default"}]},
 CreatedAt=2023-06-15T22:54:44.319Z
 )
```

Example Löschen einer S3-Access-Grants-Ressourcenrichtlinie

Verwenden Sie das folgende Codebeispiel, um eine S3 Access Grants-Ressourcenrichtlinie zu löschen. Wenn Sie den folgenden Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen.

```
public void deleteAccessGrantsInstanceResourcePolicy() {
 DeleteAccessGrantsInstanceResourcePolicyRequest deleteRequest =
 DeleteAccessGrantsInstanceResourcePolicyRequest.builder()
```

```
.accountId(111122223333)
.build();
DeleteAccessGrantsInstanceResourcePolicyResponse deleteResponse =
s3Control.putAccessGrantsInstanceResourcePolicy(deleteRequest);
LOGGER.info("DeleteAccessGrantsInstanceResourcePolicyResponse: " + deleteResponse);
}
```

Antwort:

```
DeleteAccessGrantsInstanceResourcePolicyResponse()
```

Schritt 3: Erteilen Sie IAM Identitäten in einem zweiten Konto die Erlaubnis, die S3 Access Grants-Instanz in Ihrem Konto aufzurufen

Nachdem der Eigentümer der Amazon S3 S3-Daten die kontoübergreifende Richtlinie für die S3 Access Grants-Instance im Konto konfiguriert hat`111122223333`, `444455556666` muss der Besitzer des zweiten Kontos eine identitätsbasierte Richtlinie für seine IAM Benutzer oder Rollen erstellen, und der Eigentümer muss ihnen Zugriff auf die S3 Access Grants-Instance gewähren. Nehmen Sie in die identitätsbasierte Richtlinie eine oder mehrere der folgenden Aktionen auf, je nachdem, was in der Ressourcenrichtlinie für die S3 Access Grants Instance gewährt wird und welche Berechtigungen Sie gewähren möchten:

- `s3:ListAccessGrants`
- `s3:ListAccessGrantsLocations`
- `s3:GetDataAccess`
- `s3:GetAccessGrantsInstanceForPrefix`
- `s3:ListCallerAccessGrants`

Gemäß dem AWS kontoübergreifenden Zugriffsmuster `444455556666` müssen die IAM Benutzer oder Rollen im zweiten Konto explizit über eine oder mehrere dieser Berechtigungen verfügen. Erteilen Sie beispielsweise die `s3:GetDataAccess` Berechtigung, damit der IAM Benutzer oder die Rolle die S3 Access Grants-Instanz im Konto aufrufen kann`111122223333`, um Anmeldeinformationen anzufordern.

Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
  "Effect": "Allow",
  "Action": [
   "s3:GetDataAccess",
  ],
   "Resource": "arn:aws:s3:us-east-2:111122223333:access-grants/default"
  }
 ]
}
```

Informationen zur Bearbeitung IAM identitätsbasierter Richtlinien finden Sie im Handbuch unter IAMRichtlinien bearbeiten.AWS Identity and Access Management

Schritt 4: Erstellen Sie in der S3 Access Grants-Instanz Ihres Kontos einen Zuschuss, der der IAM Identität im zweiten Konto Zugriff auf einige Ihrer S3-Daten gewährt

Für den letzten Konfigurationsschritt können Sie in der S3 Access Grants-Instanz in Ihrem Konto 111122223333 einen Grant erstellen, der Zugriff auf die IAM Identität im zweiten Konto 444455556666 auf einige der S3-Daten in Ihrem Konto gewährt. Sie können dies mithilfe der Amazon S3 S3-Konsole, CLIAPI, und tunSDKs. Weitere Informationen finden Sie unter Erstellen von Gewährungen.

Geben Sie in der Gewährung die IAM Identität AWS ARN des zweiten Kontos an und geben Sie an, für welche Position in Ihren S3-Daten (ein Bucket, ein Präfix oder ein Objekt) Sie Zugriff gewähren. Dieser Standort muss bereits bei Ihrer S3 Access Grants-Instanz registriert sein. Weitere Informationen finden Sie unter Registrieren eines Speicherorts. Sie können optional ein Unterpräfix angeben. Wenn es sich bei dem Speicherort, für den Sie Zugriff gewähren, um einen Bucket handelt und Sie den Zugriff weiter auf ein bestimmtes Objekt in diesem Bucket einschränken möchten, übergeben Sie den Objektschlüsselnamen in das `S3SubPrefix` Feld. Oder wenn Sie den Zugriff auf die Objekte im Bucket mit Schlüsselnamen einschränken möchten, die mit einem bestimmten Präfix beginnen, wie z. B.`2024-03-research-results/`, dann übergeben`S3SubPrefix=2024-03-research-results/`.

Im Folgenden finden Sie einen CLI Beispielbefehl für die Erstellung einer Zugriffsberechtigung für eine Identität im zweiten Konto. Weitere Informationen finden Sie unter Erstellen von Gewährungen. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws s3control create-access-grant \
--account-id 111122223333 \
--access-grants-location-id default \
--access-grants-location-configuration S3SubPrefix=prefixA* \
--permission READ \
--grantee GranteeType=IAM,GranteeIdentifier=arn:aws:iam::444455556666:role/data-consumer-1
```

Nach der Konfiguration des kontoübergreifenden Zugriffs kann der Benutzer oder die Rolle im zweiten Konto Folgendes tun:

- Ruft `ListAccessGrantsInstances` auf, um die S3 Access Grants-Instanzen aufzulisten, über AWS RAM die es für das Programm freigegeben wurde. Weitere Informationen finden Sie unter Rufen Sie die Details einer S3 Access Grants-Instance ab.
- Fordert temporäre Anmeldeinformationen von S3 Access Grants an. Weitere Informationen darüber, wie Sie diese Anfragen stellen können, finden Sie unterAnfordern des Zugriffs auf Amazon-S3-Daten über S3 Access Grants.

Verwenden von AWS Tags mit S3 Access Grants

Tags in Amazon S3 Access Grants besitzen ähnliche Eigenschaften wie Objekt-Tags in Amazon S3. Jeder Tag ist ein Schlüssel/Wert-Paar. Die Ressourcen in S3 Access Grants, die Sie markieren können, sind Instances, Speicherorte und Gewährungen.

> **Note**
>
> Das Taggen in S3 Access Grants verwendet andere API Operationen als das Tagging von Objekten. S3 Access Grants verwendet die TagResource, UntagResource, und ListTagsForResourceAPIOperationen, bei denen es sich bei einer Ressource entweder um eine S3 Access Grants-Instanz, einen registrierten Standort oder um eine Zugriffsberechtigung handeln kann.

Ähnlich wie bei Objekt-Tags gelten die folgenden Einschränkungen:

- Sie können Tags zu neuen S3-Access-Grants-Ressourcen hinzufügen, wenn Sie diese erstellen, oder zu vorhandenen Ressourcen hinzufügen.

- Sie können einer Ressource bis zu 10 Tags zuordnen. Wenn mehrere Tags derselben Ressource zugeordnet sind, müssen sie eindeutige Tag-Schlüssel besitzen.
- Ein Tag-Schlüssel kann maximal 128 Unicode-Zeichen lang sein, und die Tag-Werte können bis zu 256 Unicode-Zeichen lang sein. Tags werden intern in UTF -16 dargestellt. In UTF -16 verbrauchen Zeichen entweder 1 oder 2 Zeichenpositionen.
- Schlüssel und Werte unterscheiden nach Groß- und Kleinschreibung.

Weitere Informationen zu Tag-Einschränkungen finden Sie unter Einschränkungen für benutzerdefinierte Tags im AWS Billing -Benutzerhandbuch.

Sie können Ressourcen in S3 Access Grants taggen, indem Sie AWS Command Line Interface (AWS CLI), Amazon S3 REST API oder die verwenden AWS SDKs.

Mit dem AWS CLI

Informationen zur AWS CLI Installation von finden Sie unter Installation von AWS CLI im AWS Command Line Interface Benutzerhandbuch.

Sie können eine S3-Access-Grants-Ressource während oder nach der Erstellung markieren. Die folgenden Beispiele zeigen, wie Sie eine S3-Access-Grants-Instance markieren oder eine vorhandene Markierung aufheben. Sie können ähnliche Operationen für registrierte Speicherorte und Zugriffsgewährungen durchführen.

Um die folgenden Beispielbefehle zu verwenden, ersetzen Sie *user input placeholders* durch eigene Daten.

Example – Erstellen einer S3-Access-Grants-Instance mit Tags

```
aws s3control create-access-grants-instance \
 --account-id 111122223333 \
 --profile access-grants-profile \
 --region us-east-2 \
 --tags Key=tagKey1,Value=tagValue1
```

Antwort:

```
 {
    "CreatedAt": "2023-10-25T01:09:46.719000+00:00",
    "AccessGrantsInstanceId": "default",
```

```
    "AccessGrantsInstanceArn": "arn:aws:s3:us-east-2:111122223333:access-grants/
default"
}
```

Example – Markieren einer bereits erstellten S3-Access-Grants-Instance

```
aws s3control tag-resource \
--account-id 111122223333 \
--resource-arn "arn:aws:s3:us-east-2:111122223333:access-grants/default" \
--profile access-grants-profile \
--region us-east-2 \
--tags Key=tagKey2,Value=tagValue2
```

Example – Auflisten der Tags für eine S3-Access-Grants-Instance

```
aws s3control list-tags-for-resource \
--account-id 111122223333 \
--resource-arn "arn:aws:s3:us-east-2:111122223333:access-grants/default" \
--profile access-grants-profile \
--region us-east-2
```

Antwort:

```
{
    "Tags": [
        {
            "Key": "tagKey1",
            "Value": "tagValue1"
        },
        {
            "Key": "tagKey2",
            "Value": "tagValue2"
        }
    ]
}
```

Example – Aufheben der Markierung für eine S3-Access-Grants-Instance

```
aws s3control untag-resource \
 --account-id 111122223333 \
 --resource-arn "arn:aws:s3:us-east-2:111122223333:access-grants/default" \
```

```
--profile access-grants-profile \
--region us-east-2 \
--tag-keys "tagKey2"
```

Mit dem REST API

Sie können Amazon S3 verwenden, API um Tags für eine S3 Access Grants-Instance, einen registrierten Standort oder eine Zugriffserteilung zu taggen, aufzuheben oder aufzulisten. Informationen zur REST API Unterstützung der Verwaltung von S3 Access Grants-Tags finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- TagResource
- UntagResource
- ListTagsForResource

Limits für S3 Access Grants

Für S3 Access Grants gelten die folgenden Limits:

> **Note**
>
> Wenn Ihr Anwendungsfall diese Einschränkungen überschreitet, wenden Sie sich an den AWS Support, um höhere Grenzwerte anzufordern.

S3-Access-Grants-Instance

Sie können AWS-Region pro Konto 1 S3 Access Grants-Instanz erstellen. Siehe Erstellen einer S3-Access-Grants-Instance.

S3-Access-Grants Speicherort

Sie können pro S3-Access-Grants-Instance 1.000 S3-Access-Grants-Speicherorte registrieren. Siehe Registrieren eines S3-Access-Grants-Speicherorts.

Gewährung

Sie können pro S3-Access-Grants-Instance 100.000 Gewährungen erstellen. Siehe Erstellen einer Gewährung.

S3 Access Grants AWS-Regionen

S3 Access Grants ist derzeit in den folgenden Sprachen verfügbar AWS-Regionen:

AWS-Region Code	AWS-Region Name
us-east-1	USA Ost (Nord-Virginia)
us-east-2	USA Ost (Ohio)
us-west-1	USA West (Nordkalifornien)
us-west-2	USA West (Oregon)
af-south-1	Africa (Cape Town)
ap-east-1	Asia Pacific (Hongkong)
ap-northeast-1	Asien-Pazifik (Tokio)
ap-northeast-2	Asien-Pazifik (Seoul)
ap-northeast-3	Asien-Pazifik (Osaka)
ap-south-1	Asien-Pazifik (Mumbai)
ap-south-2	Asien-Pazifik (Hyderabad)
ap-southeast-1	Asien-Pazifik (Singapur)
ap-southeast-2	Asien-Pazifik (Sydney)
ap-southeast-3	Asien-Pazifik (Jakarta)
ap-southeast-4	Asien-Pazifik (Melbourne)
ca-central-1	Kanada (Zentral)
ca-west-1	Kanada West (Calgary)
eu-central-1	Europa (Frankfurt)

AWS-Region Code	AWS-Region Name
eu-central-2	Europa (Zürich)
eu-north-1	Europa (Stockholm)
eu-south-1	Europa (Milan)
eu-south-2	Europa (Spain)
eu-west-1	Europa (Irland)
eu-west-2	Europe (London)
eu-west-3	Europa (Paris)
il-central-1	Israel (Tel Aviv)
me-central-1	Naher Osten (UAE)
me-south-1	Naher Osten (Bahrain)
sa-east-1	Südamerika (São Paulo)

S3-Access-Grants-Integrationen

S3 Access Grants kann mit den folgenden AWS Diensten und Funktionen verwendet werden. Diese Seite wird aktualisiert, wenn neue Integrationen verfügbar sind.

Amazon Athena

- Verwendung von IAM Identity Center-fähigen Athena-Arbeitsgruppen

Amazon EMR

- Starten Sie einen EMR Amazon-Cluster mit S3 Access Grants

Amazon EMR auf EKS

- Starten Sie einen EMR EKS Amazon-On-Cluster mit S3 Access Grants

EMRServerless-Anwendung von Amazon

Starten Sie eine Amazon EMR Serverless-Anwendung mit S3 Access Grants

AWS IAM Identity Center

Vertrauenswürdige Identitätsverteilung zwischen Anwendungen

Amazon-Redshift

Amazon Redshift Redshift-Integration mit Amazon S3 Access Grants

Amazon SageMaker Studio

Verwenden von Amazon S3 Access Grants mit Amazon SageMaker Studio und dem SDK for Python (Boto3) -Plugin

Die Verwendung von S3 Access Grants in Amazon SageMaker Studio-Notebooks ist jetzt einfacher, wenn Sie das SDK for Python (Boto3) -Plugin verwenden. Richten Sie vorab Zugriffsberechtigungen für IAM Prinzipale und AWS IAM Identity Center Verzeichnisbenutzer ein. Amazon SageMaker Studio unterstützt zwar nicht nativ Identity Provider-Verzeichnisbenutzer, aber Sie können benutzerdefinierten Python-Code schreiben, indem Sie das Plug-in verwenden, das diesen Identitäten den Zugriff auf S3-Daten über S3 Access Grants ermöglicht. Der Datenzugriff erfolgt mit Hilfe des Plugins und nicht über Amazon SageMaker.

Open-Source-Python-Frameworks

Amazon S3 Access Grants lässt sich jetzt in Open-Source-Python-Frameworks integrieren

Verwaltung des Zugriffs mit ACLs

Zugriffskontrolllisten (ACLs) sind eine der ressourcenbasierten Optionen, mit denen Sie den Zugriff auf Ihre Buckets und Objekte verwalten können. Sie können sie verwendenACLs, um anderen grundlegende Lese-/Schreibberechtigungen zu gewähren. AWS-Konten Es gibt Einschränkungen bei der Verwaltung von Berechtigungen mithilfe von. ACLs

Sie können beispielsweise nur anderen Benutzern Berechtigungen gewähren. AWS-Konten Sie können Benutzern in Ihrem Konto keine Berechtigungen gewähren. Sie können keine bedingten Berechtigung erteilen, und Sie können nicht explizit Berechtigungen verweigern. ACLssind für bestimmte Szenarien geeignet. Wenn ein Bucket-Besitzer beispielsweise anderen AWS-Konten erlaubt, Objekte hochzuladen, können die Berechtigungen für diese Objekte nur über das Objekt ACL von demjenigen verwaltet werden AWS-Konto , dem das Objekt gehört.

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie sowohl das Eigentum an den Objekten kontrollieren können, die in Ihren Bucket hochgeladen werden, als auch um sie zu deaktivieren oder zu aktivieren. ACLs Standardmäßig ist für Object Ownership die erzwungene Einstellung des Bucket-Besitzers festgelegt, und alle sind ACLs deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff darauf ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.

Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Wenn diese ACLs Option deaktiviert ist, können Sie mithilfe von Richtlinien den Zugriff auf alle Objekte in Ihrem Bucket steuern, unabhängig davon, wer die Objekte in Ihren Bucket hochgeladen hat. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

> **Important**
>
> Wenn Ihr Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für S3 Object Ownership verwendet, müssen Sie Richtlinien verwenden, um Zugriff auf Ihren Bucket und die darin enthaltenen Objekte zu gewähren. Wenn die Einstellung „Bucket Owner erforced“ aktiviert ist, schlagen Anfragen zur Einrichtung von Zugriffskontrolllisten (ACLs) oder zur Aktualisierung ACLs fehl und geben den `AccessControlListNotSupported` Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt.

Weitere Informationen zu ACLs finden Sie in den folgenden Themen.

Themen

- Übersicht über die Zugriffskontrollliste (ACL)
- Konfiguration ACLs
- Beispiele für Richtlinien für ACLs

Übersicht über die Zugriffskontrollliste (ACL)

Mit Amazon S3 S3-Zugriffskontrolllisten (ACLs) können Sie den Zugriff auf Buckets und Objekte verwalten. Jedem Bucket und Objekt ist eine Unterressource ACL angehängt. Es definiert, welchen AWS-Konten Gruppen Zugriff gewährt wird und welche Art von Zugriff gewährt wird. Wenn eine

Anfrage für eine Ressource eingeht, überprüft Amazon S3 die entsprechende Anfrage, ACL um sicherzustellen, dass der Anforderer über die erforderlichen Zugriffsberechtigungen verfügt.

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie sowohl das Eigentum an den Objekten kontrollieren können, die in Ihren Bucket hochgeladen werden, als auch um sie zu deaktivieren oder zu aktivieren. ACLs Standardmäßig ist für Object Ownership die erzwungene Einstellung des Bucket-Besitzers festgelegt, und alle sind ACLs deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff darauf ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.

Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Wenn diese ACLs Option deaktiviert ist, können Sie mithilfe von Richtlinien den Zugriff auf alle Objekte in Ihrem Bucket steuern, unabhängig davon, wer die Objekte in Ihren Bucket hochgeladen hat. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

> **Important**
>
> Wenn Ihr Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für S3 Object Ownership verwendet, müssen Sie Richtlinien verwenden, um Zugriff auf Ihren Bucket und die darin enthaltenen Objekte zu gewähren. Wenn die Einstellung „Bucket Owner erforced“ aktiviert ist, schlagen Anfragen zur Einrichtung von Zugriffskontrolllisten (ACLs) oder zur Aktualisierung ACLs fehl und geben den `AccessControlListNotSupported` Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt.

Wenn Sie einen Bucket oder ein Objekt erstellen, erstellt Amazon S3 eine StandardeinstellungACL, die dem Eigentümer der Ressource die volle Kontrolle über die Ressource gewährt. Dies wird im folgenden Beispiel-Bucket gezeigt ACL (das Standardobjekt ACL hat dieselbe Struktur):

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>*** Owner-Canonical-User-ID ***</ID>
    <DisplayName>owner-display-name</DisplayName>
  </Owner>
  <AccessControlList>
```

```
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:type="Canonical User">
        <ID>*** Owner-Canonical-User-ID ***</ID>
        <DisplayName>display-name</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Das Beispiel ACL enthält ein `Owner` Element, das den Besitzer anhand der AWS-Konto kanonischen Benutzer-ID identifiziert. Anweisungen zum Auffinden Ihrer kanonischen Benutzer-ID finden Sie unter Eine AWS-Konto kanonische Benutzer-ID finden. Das `Grant` Element identifiziert den Empfänger (entweder eine AWS-Konto oder eine vordefinierte Gruppe) und die erteilte Berechtigung. Dieser Standard ACL hat ein `Grant` Element für den Besitzer. Sie erteilen Berechtigungen, indem Sie `Grant`-Elemente hinzufügen, wobei jedes Recht den Empfänger und die Berechtigung identifiziert.

> **Note**
>
> An ACL kann bis zu 100 Zuschüsse vergeben werden.

Themen

- Wer ist ein Empfänger?
- Welche Berechtigungen kann ich erteilen?
- aclRequired-Werte für allgemeine Amazon-S3-Anfragen
- Stichprobe ACL
- Eingemacht ACL

Wer ist ein Empfänger?

Ein Stipendiat kann eine AWS-Konto oder eine der vordefinierten Amazon S3 S3-Gruppen sein. Sie erteilen einem die Erlaubnis, AWS-Konto indem Sie die E-Mail-Adresse oder die kanonische Benutzer-ID verwenden. Wenn Sie jedoch in Ihrem Zuschussantrag eine E-Mail-Adresse angeben, findet Amazon S3 die kanonische Benutzer-ID für dieses Konto und fügt sie dem hinzu. ACL Das Ergebnis enthält ACLs immer die kanonische Benutzer-ID für den AWS-Konto, nicht die E-Mail-Adresse des. AWS-Konto

Wenn Sie Zugriffsrechte erteilen, geben Sie jeden Empfänger als *type*="*value*"-Paar an, wobei *type* einer der folgenden ist:

- `id`— Wenn der angegebene Wert die kanonische Benutzer-ID eines ist AWS-Konto
- `uri` – Wenn Sie einer vordefinierten Gruppe Berechtigungen erteilen
- `emailAddress` – Wenn der angegebene Wert die E-Mail-Adresse eines AWS-Konto ist

> **Important**
>
> Die Verwendung von E-Mail-Adressen zur Angabe eines Berechtigungsempfängers wird ausschließlich in den folgenden AWS -Regionen unterstützt:
>
> - USA Ost (Nord-Virginia)
> - USA West (Nordkalifornien)
> - USA West (Oregon)
> - Asien-Pazifik (Singapur)
> - Asien-Pazifik (Sydney)
> - Asien-Pazifik (Tokio)
> - Europa (Irland)
> - Südamerika (São Paulo)
>
> Eine Liste aller unterstützten Amazon-S3-Regionen und -Endpunkte finden Sie unter Regionen und Endpunkte in der Allgemeine Amazon Web Services-Referenz.

Example Beispiel: E-Mail-Adresse

Der folgende `x-amz-grant-read` Header gewährt beispielsweise den durch E-Mail-Adressen AWS-Konten identifizierten Adressen die Erlaubnis, Objektdaten und ihre Metadaten zu lesen:

```
x-amz-grant-read: emailAddress="xyz@example.com", emailAddress="abc@example.com"
```

> **Warning**
>
> Wenn Sie anderen Personen AWS-Konten Zugriff auf Ihre Ressourcen gewähren, sollten Sie sich bewusst sein, dass diese ihre Berechtigungen an Benutzer unter ihren Konten

delegieren AWS-Konten können. Man spricht auch von einem kontenübergreifenden Zugriff. Informationen zur Verwendung des kontoübergreifenden Zugriffs finden Sie im Benutzerhandbuch unter Erstellen einer Rolle zur Delegierung von Berechtigungen an einen IAM Benutzer. IAM

Eine AWS-Konto kanonische Benutzer-ID finden

Die kanonische Benutzer-ID ist Ihrem AWS-Konto zugeordnet. Diese ID besteht aus einer langen Zeichenfolge, wie z. B.:

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Informationen darüber, wie Sie die kanonische Benutzer-ID für Ihr Konto finden, finden Sie unter Finden Sie die kanonische Benutzer-ID für Ihr Konto AWS-Konto im Referenzhandbuch zur Kontoverwaltung.AWS

Sie können die kanonische Benutzer-ID eines auch nachschlagen, AWS-Konto indem Sie die eines Buckets oder ACL eines Objekts lesen, für das er Zugriffsberechtigungen besitzt. AWS-Konto Wenn einer Person AWS-Konto durch eine Erteilungsanfrage Berechtigungen erteilt werden, wird dem ein Eintrag ACL mit der kanonischen Benutzer-ID des Kontos hinzugefügt.

> **Note**
>
> Falls Sie Ihren Bucket öffentlich machen (nicht empfohlen), können beliebige, nicht authentifizierte Benutzer Objekte in den Bucket hochladen. Diese anonymen Benutzer haben kein AWS-Konto. Wenn ein anonymer Benutzer ein Objekt in Ihren Bucket hochlädt, fügt Amazon S3 eine spezielle kanonische Benutzer-ID (`65a011a29cdf8ec533ec3d1ccaae921c`) als Objekteigentümer in der hinzu. ACL Weitere Informationen finden Sie unter Amazon-S3-Bucket- und Objekt-Eigentümerschaft.

Vordefinierte Gruppen in Amazon S3

Amazon S3 besitzt mehrere vordefinierte Gruppen. Wenn Sie einer Gruppe Kontozugriff gewähren, geben Sie eine der Amazon S3 URIs anstelle einer kanonischen Benutzer-ID an. Amazon S3 stellt die folgenden vordefinierten Gruppen bereit:

- Gruppe „Authentifizierte Benutzer“ – Repräsentiert durch `http://acs.amazonaws.com/groups/global/AuthenticatedUsers`.

Diese Gruppe steht für alles. AWS-KontenDie Zugriffsberechtigung für diese Gruppe ermöglicht es jedem AWS-Konto , auf die Ressource zuzugreifen. Alle Anfragen müssen jedoch signiert (authentifiziert) sein.

> **Warning**
>
> Wenn Sie der Gruppe Authentifizierte Benutzer Zugriff gewähren, kann jeder AWS authentifizierte Benutzer auf der Welt auf Ihre Ressource zugreifen.

- Gruppe „Alle Benutzer“ – Repräsentiert durch `http://acs.amazonaws.com/groups/global/AllUsers`.

  Die Zugriffsberechtigung für diese Gruppe gestattet jedem, auf die Ressource zuzugreifen. Die Anfragen können signiert (authentifiziert) oder nicht signiert (anonym) sein. Nicht signierte Anfragen lassen den Authentifizierungs-Header in der Anfrage weg.

  > **Warning**
  >
  > Wir empfehlen dringend, dass Sie nie der Gruppe Alle Benutzer `WRITE`-, `WRITE_ACP`- oder `FULL_CONTROL`-Berechtigungen erteilen. Während `WRITE`-Berechtigungen es Nichtbesitzern beispielsweise nicht erlauben, vorhandene Objekte zu überschreiben oder zu löschen, erlauben `WRITE`-Berechtigungen jedem, Objekte in Ihrem Bucket zu speichern, für die Ihnen eine Rechnung gestellt wird. Weitere Informationen zu diesen Berechtigungen finden Sie im folgenden Abschnitt Welche Berechtigungen kann ich erteilen?.

- Gruppe „Protokollbereitstellung“ – Repräsentiert durch `http://acs.amazonaws.com/groups/s3/LogDelivery`.

  Die `WRITE`-Berechtigung für einen Bucket gestattet dieser Gruppe, Serverzugriff-Protokolle (siehe Protokollieren von Anfragen mit Server-Zugriffsprotokollierung) in den Bucket zu schreiben.

> **Note**
>
> Bei der Nutzung kann ACLs es sich bei einem Empfänger um eine AWS-Konto oder eine der vordefinierten Amazon S3 S3-Gruppen handeln. Der Empfänger kann jedoch kein Benutzer

sein. IAM Weitere Informationen zu den AWS Benutzern und den darin enthaltenen IAM Berechtigungen finden Sie unter Verwenden AWS Identity and Access Management.

Welche Berechtigungen kann ich erteilen?

In der folgenden Tabelle sind die Berechtigungen aufgeführt, die Amazon S3 in einem unterstütztACL. Der Satz von ACL Berechtigungen ist für ein Objekt ACL und einen Bucket identischACL. Je nach Kontext (Bucket ACL oder ObjektACL) gewähren diese ACL Berechtigungen jedoch Berechtigungen für bestimmte Buckets oder Objektoperationen. Die Tabelle listet die Berechtigungen auf und beschreibt, was sie im Kontext der Objekte und Buckets bedeuten.

Weitere Informationen zu ACL Berechtigungen in der Amazon S3 S3-Konsole finden Sie unterKonfiguration ACLs.

Berechtigung	Bei Rechteerteilung für einen Bucket	Bei Rechteerteilung für ein Objekt
`READ`	Gestattet dem Empfänger, die Objekte im Bucket aufzulisten	Gestattet dem Empfänger, die Objektedaten und seine Metadaten zu lesen
`WRITE`	Gestattet dem Empfänger, neue Objekte im Bucket zu erstellen. Für die Bucket- und Objekteigentümer vorhandener Objekte können auch Löschungen und Überschreibungen dieser Objekte ermöglicht werden.	Nicht zutreffend
`READ_ACP`	Ermöglicht dem Empfänger, den Bucket zu lesen ACL	Ermöglicht dem Empfänger, das Objekt zu lesen ACL
`WRITE_ACP`	Ermöglicht dem Empfänger, das ACL für den entsprechenden Bucket zu schreiben	Ermöglicht dem Empfänger, das ACL für das entsprechende Objekt zu schreiben
`FULL_CONTROL`	Gewährt dem Berechtigungsempfänger die `READ`-, `WRITE`-, `READ_ACP`- und	Gewährt dem Berechtigungsempfänger die `READ`-, `READ_ACP`- und `WRITE_ACP` -Berechtigungen für das Objekt

Berechtigung	Bei Rechteerteilung für einen Bucket	Bei Rechteerteilung für ein Objekt
	`WRITE_ACP` - Berechtigungen für den Bucket	

> **Warning**
>
> Seien Sie beim Gewähren von Zugriffsberechtigungen auf Ihre S3-Buckets und -Objekte vorsichtig. Beispielsweise kann der Berechtigungsempfänger nach dem Gewähren des `WRITE`-Zugriffs auf einen Bucket Objekte im Bucket erstellen. Wir empfehlen dringend, dass Sie vor dem Erteilen von Berechtigungen den gesamten Abschnitt zu Übersicht über die Zugriffskontrollliste (ACL) lesen.

Zuordnung von ACL Berechtigungen und Zugriffsrichtlinienberechtigungen

Wie in der obigen Tabelle gezeigt, ACL erlaubt an nur eine begrenzte Anzahl von Berechtigungen, verglichen mit der Anzahl von Berechtigungen, die Sie in einer Zugriffsrichtlinie festlegen können (siehePolitische Maßnahmen für Amazon S3). Jede dieser Berechtigungen erlaubt einen oder mehrere Amazon-S3-Vorgänge.

Die folgende Tabelle zeigt, wie die einzelnen ACL Berechtigungen den entsprechenden Zugriffsrichtlinienberechtigungen zugeordnet werden. Wie Sie sehen können, erlaubt die Zugriffsrichtlinie mehr Berechtigungen als ACL jede. Sie werden ACLs hauptsächlich verwendet, um grundlegende Lese-/Schreibberechtigungen zu gewähren, ähnlich wie Dateisystemberechtigungen. Weitere Informationen darüber, wann Sie eine verwenden solltenACL, finden Sie unter. Identity and Access Management für Amazon S3

Weitere Informationen zu ACL Berechtigungen in der Amazon S3 S3-Konsole finden Sie unterKonfiguration ACLs.

ACLErlaubnis	Entsprechende Zugriffsrichtlinien-Berechtigungen, wenn die ACL Erlaubnis für einen Bucket erteilt wird	Entsprechende Zugriffsrichtlinienberechtigungen, wenn die ACL Berechtigung für ein Objekt erteilt wird
READ	`s3:ListBucket` , `s3:ListBucketVersions` und `s3:ListBucketMultipartUploads`	`s3:GetObject` und `s3:GetObjectVersion`
WRITE	`s3:PutObject` Der Bucket-Eigentümer kann jedes Objekt im Bucket erstellen, überschreiben und löschen, und der Objekteigentümer hat `FULL_CONTROL` über sein Objekt. Wenn der Empfänger der Bucket-Besitzer ist, ACL ermöglicht die Erteilung von `WRITE` Berechtigungen in einem Bucket außerdem, dass die `s3:DeleteObjectVersion` Aktion für jede Version in diesem Bucket ausgeführt werden kann.	Nicht zutreffend
READ_ACP	`s3:GetBucketAcl`	`s3:GetObjectAcl` und `s3:GetObjectVersionAcl`
WRITE_ACP	`s3:PutBucketAcl`	`s3:PutObjectAcl` und `s3:PutObjectVersionAcl`
FULL_CONTROL	Entspricht der Erteilung von `READWRITE`,`READ_ACP`, und `WRITE_ACP` ACL Berechtigungen. Dementsprechend entspricht diese ACL Berechtigung einer Kombination aus entsprechenden Zugriffsrichtlinienberechtigungen.	Entspricht der Gewährung von `READREAD_ACP`, und `WRITE_ACP` ACL Berechtigungen. Dementsprechend entspricht diese ACL Berechtigung einer Kombination aus entsprechenden Zugriffsrichtlinienberechtigungen.

Bedingungsschlüssel

Wenn Sie Zugriffsrichtlinienberechtigungen gewähren, können Sie Bedingungsschlüssel verwenden, um den Wert für für ein Objekt mithilfe einer Bucket-Richtlinie einzuschränken. ACL Die folgenden Kontextschlüssel entsprechenACLs. Sie können diese Kontextschlüssel verwenden, um die Verwendung eines bestimmten Elements ACL in einer Anfrage vorzuschreiben:

- `s3:x-amz-grant-read` - Erfordert Lesezugriff.
- `s3:x-amz-grant-write` - Erfordert Schreibzugriff.
- `s3:x-amz-grant-read-acp`- Lesezugriff auf den Bucket erforderlichACL.
- `s3:x-amz-grant-write-acp`- Schreibzugriff auf den Bucket erforderlichACL.
- `s3:x-amz-grant-full-control` - Erfordert vollständige Kontrolle.
- `s3:x-amz-acl` - Erfordert eine Eingemacht ACL.

Richtlinien, die ACL -spezifische Header beinhalten, finden Sie beispielsweise unter. Gewährung von PutObject s3-Berechtigungen mit einer Bedingung, die voraussetzt, dass der Bucket-Besitzer die volle Kontrolle erhält Eine vollständige Liste der Amazon S3-spezifischen Bedingungsschlüssel finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

`aclRequired`-Werte für allgemeine Amazon-S3-Anfragen

Um Amazon S3 S3-Anfragen zu identifizieren, die ACLs für eine Autorisierung erforderlich sind, können Sie den `aclRequired` Wert in den Amazon S3 S3-Serverzugriffsprotokollen oder verwenden AWS CloudTrail. Der `aclRequired` Wert, der in CloudTrail unseren Amazon S3-Serverzugriffsprotokollen erscheint, hängt davon ab, welche Operationen aufgerufen wurden, und von bestimmten Informationen über den Anforderer, Objekteigentümer und Bucket-Besitzer. Wenn keine erforderlich ACLs waren, oder wenn Sie die Option „Gespeichert`bucket-owner-full-control`“ ACL festlegen oder wenn die Anfragen gemäß Ihrer Bucket-Richtlinie zulässig sind, lautet die `aclRequired` Wertzeichenfolge in den Amazon S3-Serverzugriffsprotokollen - "" und fehlt in CloudTrail.

In den folgenden Tabellen sind die erwarteten `aclRequired` Werte in CloudTrail unseren Amazon S3 S3-Serverzugriffsprotokollen für die verschiedenen Amazon S3 API S3-Operationen aufgeführt.

Sie können diese Informationen verwenden, um zu verstehen, von welchen Amazon S3 S3-Vorgängen ACLs die Autorisierung abhängt. In den folgenden Tabellen stellen A, B und C die verschiedenen Konten dar, die dem Anforderer, Objekteigentümer und Bucket-Besitzer zugeordnet sind. Einträge mit einem Sternchen (*) geben eines der Konten A, B oder C an.

> **Note**
>
> `PutObject`Die Operationen in der folgenden Tabelle weisen, sofern nicht anders angegeben, auf Anfragen hin, für die kein Wert festgelegt wurdeACL, sofern es sich nicht um einen ACL handelt `bucket-owner-full-control`ACL. Ein Nullwert für `aclRequired` gibt an, dass dieser `aclRequired` Wert in den AWS CloudTrail Protokollen fehlt.

Die folgende Tabelle zeigt die `aclRequired` Werte für CloudTrail.

Vorgangsname	Auftraggeber	Objekteigentümer	Bucket-Eigentümer	Bucket-Richtlinie gewährt Zugriff	**`aclRequired`** Wert	Grund
`GetObject`	A	A	A	Sie können zwischen Yes und No wählen	Null	Zugriff auf dasselbe Konto
`GetObject`	A	B	A	Sie können zwischen Yes und No wählen	Null	Zugriff auf dasselbe Konto mit erzwungenem Bucket-Eigentümer
`GetObject`	A	A	B	Ja	Null	Von Bucket-Richtlinie gewährter kontoüber

Vorgangsname	Auftraggeber	Objekteigentümer	Bucket-Eigentümer	Bucket-Richtlinie gewährt Zugriff	**`aclRequired`** Wert	Grund
						greifender Zugriff
`GetObject`	A	A	B	Nein	Ja	Der kontoüber greifende Zugriff ist abhängig von ACL
`GetObject`	A	A	B	Ja	Null	Von Bucket-Richtlinie gewährter kontoüber greifender Zugriff
`GetObject`	A	B	B	Nein	Ja	Der kontenübergreifende Zugriff ist abhängig von ACL
`GetObject`	A	B	C	Ja	Null	Von Bucket-Richtlinie gewährter kontoüber greifender Zugriff

Vorgangsname	Auftraggeber	Objekteigentümer	Bucket-Eigentümer	Bucket-Richtlinie gewährt Zugriff	**`aclRequired`** Wert	Grund
`GetObject`	A	B	C	Nein	Ja	Der kontenübergreifende Zugriff ist abhängig von ACL
`PutObject`	A	Nicht zutreffend	A	Sie können zwischen Yes und No wählen	Null	Zugriff auf dasselbe Konto
`PutObject`	A	Nicht zutreffend	B	Ja	Null	Von Bucket-Richtlinie gewährter kontoübergreifender Zugriff
`PutObject`	A	Nicht zutreffend	B	Nein	Ja	Der kontenübergreifende Zugriff ist abhängig von ACL

Vorgangsname	Auftraggeber	Objekteigentümer	Bucket-Eigentümer	Bucket-Richtlinie gewährt Zugriff	**`aclRequired`** Wert	Grund
`PutObject` mit einem ACL (außerbucke `owner-full-control`)	*	Nicht zutreffend	*	Sie können zwischen Yes und No wählen	Ja	Zuschüsse beantragen ACL
`ListObjects`	A	Nicht zutreffend	A	Sie können zwischen Yes und No wählen	Null	Zugriff auf dasselbe Konto
`ListObjects`	A	Nicht zutreffend	B	Ja	Null	Von Bucket-Richtlinie gewährter kontoüber greifender Zugriff
`ListObjects`	A	Nicht zutreffend	B	Nein	Ja	Der kontenübergreifende Zugriff ist abhängig von ACL
`DeleteObject`	A	Nicht zutreffend	A	Sie können zwischen Yes und No wählen	Null	Zugriff auf dasselbe Konto

Vorgangsname	Auftraggeber	Objekteigentümer	Bucket-Eigentümer	Bucket-Richtlinie gewährt Zugriff	**`aclRequired`** Wert	Grund
`DeleteObject`	A	Nicht zutreffend	B	Ja	Null	Von Bucket-Richtlinie gewährter kontoüber greifender Zugriff
`DeleteObject`	A	Nicht zutreffend	B	Nein	Ja	Der kontenübergreifende Zugriff ist abhängig von ACL
`PutObjectAcl`	*	*	*	Sie können zwischen Yes und No wählen	Ja	Zuschüsse beantragen ACL
`PutBucketAcl`	*	Nicht zutreffend	*	Sie können zwischen Yes und No wählen	Ja	Zuschüsse beantragen ACL

> **Note**
>
> `REST.PUT.OBJECT`Die Operationen in der folgenden Tabelle weisen, sofern nicht anders angegeben, auf Anfragen hin, die keinen Wert festlegen`ACL`, es sei denn, der ACL ist ein

`bucket-owner-full-control`ACL. Eine `aclRequired`-Wertzeichenfolge von „`-`“ gibt einen Nullwert in den Amazon-S3-Serverzugriffsprotokollen an.

Die folgende Tabelle zeigt die `aclRequired` Werte für Amazon S3 S3-Serverzugriffsprotokolle.

Vorgangsname	Auftraggeber	Objekteigentümer	Bucket-Eigentümer	Bucket-Richtlinie gewährt Zugriff	**`aclRequired`** Wert	Grund
`REST.GET.OBJECT`	A	A	A	Sie können zwischen Yes und No wählen	-	Zugriff auf dasselbe Konto
`REST.GET.OBJECT`	A	B	A	Sie können zwischen Yes und No wählen	-	Zugriff auf dasselbe Konto mit erzwungenem Bucket-Eigentümer
`REST.GET.OBJECT`	A	A	B	Ja	-	Von Bucket-Richtlinie gewährter kontoübergreifender Zugriff
`REST.GET.OBJECT`	A	A	B	Nein	Ja	Der kontoübergreifende Zugriff ist abhängig von ACL

Vorgangsname	Auftraggeber	Objekteigentümer	Bucket-Eigentümer	Bucket-Richtlinie gewährt Zugriff	**`aclRequired`** Wert	Grund
`REST.GET.OBJECT`	A	B	B	Ja	-	Von Bucket-Richtlinie gewährter kontoüber greifender Zugriff
`REST.GET.OBJECT`	A	B	B	Nein	Ja	Der kontenübergreifende Zugriff ist abhängig von ACL
`REST.GET.OBJECT`	A	B	C	Ja	-	Von Bucket-Richtlinie gewährter kontoüber greifender Zugriff
`REST.GET.OBJECT`	A	B	C	Nein	Ja	Der kontenübergreifende Zugriff ist abhängig von ACL

Vorgangsname	Auftraggeber	Objekteigentümer	Bucket-Eigentümer	Bucket-Richtlinie gewährt Zugriff	**`aclRequired`** Wert	Grund
`REST.PUT.OBJECT`	A	Nicht zutreffend	A	Sie können zwischen Yes und No wählen	-	Zugriff auf dasselbe Konto
`REST.PUT.OBJECT`	A	Nicht zutreffend	B	Ja	-	Von Bucket-Richtlinie gewährter kontoüber greifender Zugriff
`REST.PUT.OBJECT`	A	Nicht zutreffend	B	Nein	Ja	Der kontenübergreifende Zugriff ist abhängig von ACL
`REST.PUT.OBJECT` mit einem ACL (außerbucke`owner-full-control`)	*	Nicht zutreffend	*	Sie können zwischen Yes und No wählen	Ja	Zuschüsse beantragen ACL
`REST.GET.BUCKET`	A	Nicht zutreffend	A	Sie können zwischen Yes und No wählen	-	Zugriff auf dasselbe Konto

Vorgangsname	Auftraggeber	Objekteigentümer	Bucket-Eigentümer	Bucket-Richtlinie gewährt Zugriff	**`aclRequired`** Wert	Grund
`REST.GET.BUCKET`	A	Nicht zutreffend	B	Ja	-	Von Bucket-Richtlinie gewährter kontoüber greifender Zugriff
`REST.GET.BUCKET`	A	Nicht zutreffend	B	Nein	Ja	Der kontenübergreifende Zugriff ist abhängig von ACL
`REST.DELETE.OBJECT`	A	Nicht zutreffend	A	Sie können zwischen Yes und No wählen	-	Zugriff auf dasselbe Konto
`REST.DELETE.OBJECT`	A	Nicht zutreffend	B	Ja	-	Von Bucket-Richtlinie gewährter kontoüber greifender Zugriff

Vorgangsname	Auftraggeber	Objekteigentümer	Bucket-Eigentümer	Bucket-Richtlinie gewährt Zugriff	**aclRequired** Wert	Grund
REST.DELETE.OBJECT	A	Nicht zutreffend	B	Nein	Ja	Der kontenübergreifende Zugriff ist abhängig von ACL
REST.PUT.ACL	*	*	*	Sie können zwischen Yes und No wählen	Ja	Zuschüsse beantragen ACL

Stichprobe ACL

Das folgende Beispiel ACL für einen Bucket identifiziert den Eigentümer der Ressource und eine Reihe von Zuschüssen. Das Format ist die XML Darstellung von ACL in Amazon S3 RESTAPI. Der Bucket-Eigentümer hat die FULL_CONTROL über die Ressource. Darüber hinaus wird ACL gezeigt, wie zwei der vordefinierten Amazon S3 S3-Gruppen AWS-Konten, die im vorherigen Abschnitt beschrieben wurden, Berechtigungen für eine Ressource erteilt werden, die durch eine kanonische Benutzer-ID identifiziert werden.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>Owner-canonical-user-ID</ID>
    <DisplayName>display-name</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:type="CanonicalUser">
        <ID>Owner-canonical-user-ID</ID>
```

```
        <DisplayName>display-name</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>

    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:type="CanonicalUser">
        <ID>user1-canonical-user-ID</ID>
        <DisplayName>display-name</DisplayName>
      </Grantee>
      <Permission>WRITE</Permission>
    </Grant>

    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:type="CanonicalUser">
        <ID>user2-canonical-user-ID</ID>
        <DisplayName>display-name</DisplayName>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>

    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission>READ</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
      </Grantee>
      <Permission>WRITE</Permission>
    </Grant>

  </AccessControlList>
</AccessControlPolicy>
```

Eingemacht ACL

Amazon S3 unterstützt eine Reihe vordefinierter Zuschüsse, die als vorgefertigte Zuschüsse bezeichnet ACLs werden. Jeder Status ACL enthält einen vordefinierten Satz von Stipendiaten

und Berechtigungen. In der folgenden Tabelle sind die vordefinierten Zuschüsse ACLs und die zugehörigen vordefinierten Zuschüsse aufgeführt.

Eingesetzt ACL	Gilt für	Berechtigungen wurden hinzugefügt zu ACL
`private`	Bucket und Objekt	Der Eigentümer erhält `FULL_CONTROL` . Niemand anderer hat Zugriffsrechte (Standard).
`public-read`	Bucket und Objekt	Der Eigentümer erhält `FULL_CONTROL` . Die `AllUsers`-Gruppe (siehe Wer ist ein Empfänger?) erhält `READ`-Zugriff.
`public-read-write`	Bucket und Objekt	Der Eigentümer erhält `FULL_CONTROL` . Die `AllUsers`-Gruppe erhält `READ`- und `WRITE`-Zugriff. Eine solche Erteilung von Rechten für einen Bucket wird im Allgemeinen nicht empfohlen.
`aws-exec-read`	Bucket und Objekt	Der Eigentümer erhält `FULL_CONTROL` . Amazon EC2 erhält `READ` Zugriff auf `GET` ein Amazon Machine Image (AMI) -Paket von Amazon S3.
`authenticated-read`	Bucket und Objekt	Der Eigentümer erhält `FULL_CONTROL` . Die `AuthenticatedUsers` -Gruppe erhält `READ`-Zugriff.
`bucket-owner-read`	Objekt	Der Objekt-Eigentümer erhält `FULL_CONTROL` . Der Bucket-Eigentümer erhält `READ`-Zugriff. Wenn Sie ACL bei der Erstellung eines Buckets angeben, dass dieser Wert gespeichert ist, ignoriert Amazon S3 ihn.
`bucket-owner-full-control`	Object	Sowohl der Objekt-Eigentümer, als auch der Bucket-Eigentümer erhalten `FULL_CONTROL` für das Objekt. Wenn Sie ACL bei der Erstellung eines Buckets angeben, dass dieser Wert gespeichert ist, ignoriert Amazon S3 ihn.
`log-delivery-write`	Bucket	Die `LogDelivery` -Gruppe erhält `WRITE`- und `READ_ACP`-Berechtigungen für den Bucket. Weitere

Eingesetzt ACL	Gilt für	Berechtigungen wurden hinzugefügt zu ACL
		Informationen über Protokolle finden Sie unter (Protokollieren von Anfragen mit Server-Zugriffsprotokollierung).

> **Note**
>
> Sie können ACLs in Ihrer Anfrage nur eines dieser vorgefertigten Objekte angeben.

Mithilfe des Anforderungsheaders geben Sie ACL in Ihrer Anfrage einen `x-amz-acl` vorgefertigten Wert an. Wenn Amazon S3 eine Anfrage mit einer gespeicherten ACL Anfrage erhält, werden die vordefinierten Zuschüsse ACL der Ressource hinzugefügt.

Konfiguration ACLs

In diesem Abschnitt wird erklärt, wie Sie Zugriffsberechtigungen für S3-Buckets und -Objekte mithilfe von Zugriffskontrolllisten (ACLs) verwalten. ACLMithilfe von, AWS Command Line Interface (CLI) AWS Management Console, oder AWS SDKs können Sie Ihrer Ressource Zuschüsse hinzufügen. REST API

Bucket- und Objekt-Berechtigungen sind voneinander unabhängig. Ein Objekt erbt nicht die Berechtigungen von seinem Bucket. Wenn Sie beispielsweise einen Bucket erstellen und einem Benutzer Schreibzugriff erteilen, können Sie auf die Objekte dieses Benutzers nicht zugreifen, wenn Ihnen der Benutzer nicht explizit Zugriff erteilt.

Sie können anderen AWS-Konto Benutzern oder vordefinierten Gruppen Berechtigungen gewähren. Der Benutzer oder die Gruppe, dem bzw. der Sie Berechtigungen erteilen, wird als der Berechtigungsempfänger bezeichnet. Standardmäßig hat der Besitzer, der den AWS-Konto Bucket erstellt hat, volle Berechtigungen.

Jede Berechtigung, die Sie einem Benutzer oder einer Gruppe gewähren, fügt einen Eintrag in dem hinzuACL, der dem Bucket zugeordnet ist. ACLIn der Liste werden Zuschüsse aufgeführt, anhand derer der Empfänger und die erteilte Berechtigung identifiziert werden.

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie sowohl das Eigentum an den Objekten kontrollieren können, die in Ihren Bucket hochgeladen werden, als auch um sie zu deaktivieren oder zu aktivieren. ACLs Standardmäßig ist für Object Ownership die

erzwungene Einstellung des Bucket-Besitzers festgelegt, und alle sind ACLs deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff darauf ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.

Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Wenn diese ACLs Option deaktiviert ist, können Sie mithilfe von Richtlinien den Zugriff auf alle Objekte in Ihrem Bucket steuern, unabhängig davon, wer die Objekte in Ihren Bucket hochgeladen hat. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

Important

Wenn Ihr Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für S3 Object Ownership verwendet, müssen Sie Richtlinien verwenden, um Zugriff auf Ihren Bucket und die darin enthaltenen Objekte zu gewähren. Wenn die Einstellung „Bucket Owner erforced“ aktiviert ist, schlagen Anfragen zur Einrichtung von Zugriffskontrolllisten (ACLs) oder zur Aktualisierung ACLs fehl und geben den `AccessControlListNotSupported` Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt.

Warning

Es wird dringend empfohlen, den Gruppen Jeder (öffentlicher Zugriff) oder Authentifizierte Benutzer (alle AWS authentifizierten Benutzer) keinen Schreibzugriff zu gewähren. Weitere Informationen zu den Auswirkungen eines Schreibzugriffs für diese Gruppen finden Sie unter Vordefinierte Gruppen in Amazon S3.

Verwenden Sie die S3-Konsole, um ACL Berechtigungen für einen Bucket festzulegen

Die Konsole zeigt kombinierte Zugriffsberechtigungen für doppelte Berechtigungsempfänger an. Um die vollständige Liste von zu sehenACLs, verwenden Sie Amazon S3 RESTAPI, AWS CLI, oder AWS SDKs.

Die folgende Tabelle zeigt die ACL Berechtigungen, die Sie für Buckets in der Amazon S3 S3-Konsole konfigurieren können.

Amazon S3 ACL S3-Konsolenberechtigungen für Buckets

Konsolenberechtigung	ACLErlaubnis	Zugriff
Objekte – Auflisten	`READ`	Gestattet dem Empfänger, die Objekte im Bucket aufzulisten.
Objekte – Schreiben	`WRITE`	Gestattet dem Empfänger, neue Objekte im Bucket zu erstellen . Für die Bucket- und Objekteigentümer vorhandener Objekte können auch Löschungen und Überschreibungen dieser Objekte ermöglicht werden.
Eimer ACL — Lesen	`READ_ACP`	Ermöglicht dem Empfänger, den Bucket ACL zu lesen.
Eimer ACL — Schreiben	`WRITE_ACP`	Ermöglicht dem Empfänger, das ACL für den entsprechenden Bucket zu schreiben.
Jeder (öffentlicher Zugang): Objekte – Auflisten	`READ`	Gewährt öffentlichen Lesezugriff für die Objekte im Bucket. Wenn Sie jedem (öffentlichen Zugriff) Listenzugriff gewähren, kann jeder Benutzer auf der Welt auf die Objekte im Bucket zugreifen.
Jeder (öffentlich zugänglich): Bucket ACL — Lesen	`READ_ACP`	Gewährt öffentlichen Lesezugriff für den BucketACL. Wenn Sie allen Benutzern Lesezugriff gewähren (öffentlichen Zugriff), kann jeder auf der Welt auf den Bucket zugreifenACL.

Weitere Informationen zu ACL Berechtigungen finden Sie unterÜbersicht über die Zugriffskontrollliste (ACL).

> **Important**
>
> Wenn Ihr Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für S3 Object Ownership verwendet, müssen Sie Richtlinien verwenden, um Zugriff auf Ihren Bucket und die darin enthaltenen Objekte zu gewähren. Wenn die Einstellung Bucket Owner erforced

aktiviert ist, schlagen Anfragen zur Einrichtung von Zugriffskontrolllisten (ACLs) oder zur Aktualisierung ACLs fehl und geben den `AccessControlListNotSupported` Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt.

Um ACL Berechtigungen für einen Bucket festzulegen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie Berechtigungen festlegen möchten.
3. Wählen Sie Permissions (Berechtigungen).
4. Wählen Sie unter Access Control List Bearbeiten.

   Sie können die folgenden ACL Berechtigungen für den Bucket bearbeiten:

   Objekte

   - Auflisten – Gestattet einem Empfänger, die Objekte im Bucket aufzulisten.
   - Schreiben — Ermöglicht es dem Berechtigungsempfänger, neue Objekte im Bucket zu erstellen. Für die Bucket- und Objekteigentümer vorhandener Objekte können auch Löschungen und Überschreibungen dieser Objekte ermöglicht werden.

     In der S3-Konsole können Sie nur der S3-Protokollbereitstellungsgruppe und dem Bucket-Besitzer (Ihrem AWS-Konto) Schreibzugriff gewähren. Wir empfehlen dringend, anderen Empfängern keinen Schreibzugriff zu gewähren. Wenn Sie jedoch Schreibzugriff gewähren müssen, können Sie den AWS CLI AWS SDKs, oder den verwenden RESTAPI.

   Eimer ACL

   - Lesen — Ermöglicht dem Empfänger, den Bucket ACL zu lesen.
   - Schreiben — Ermöglicht dem Empfänger, das ACL für den entsprechenden Bucket zu schreiben.
5. Um die Berechtigungen des Bucket-Besitzers zu ändern, deaktivieren Sie neben Bucket-Besitzer (Ihre AWS-Konto) die folgenden ACL Berechtigungen oder wählen Sie eine aus:

   - Objekte – Auflisten oder Schreiben

- Bucket ACL — Lesen oder Schreiben

Der Besitzer bezieht sich auf den Root-Benutzer des AWS-Kontos, nicht auf einen AWS Identity and Access Management IAM Benutzer. Weitere Informationen zum Root-Benutzer finden Sie unter The Root-Benutzer des AWS-Kontos im IAMBenutzerhandbuch.

6. Wenn Sie Berechtigungen für die allgemeine Öffentlichkeit (jeder im Internet) gewähren oder rückgängig machen möchten, deaktivieren Sie neben Jeder (öffentlicher Zugriff) die folgenden Berechtigungen oder wählen Sie eine der folgenden ACL Berechtigungen aus:

   - Objekte – Auflisten
   - Bucket ACL — Lesen

   > **Warning**
   >
   > Seien Sie vorsichtig, wenn Sie der Gruppe Everyone (Jeder) öffentlichen Zugriff auf Ihren S3-Bucket gewähren. Wenn Sie dieser Gruppe anonymen Zugriff gewähren, kann jeder auf der ganzen Welt auf Ihren Bucket zugreifen. Wir empfehlen dringend, nie einen öffentlichen Schreibzugriff auf Ihren S3-Bucket zu gewähren.

7. Um einer Person, die neben der Gruppe „Authentifizierte Benutzer“ eine AWS-Konto weitere Gruppe besitzt (jede Person mit einer AWS-Konto), Berechtigungen zu erteilen oder rückgängig zu machen, deaktivieren Sie die folgenden ACL Berechtigungen oder wählen Sie eine aus:

   - Objekte – Auflisten
   - Bucket ACL — Lesen

8. Um Amazon S3 Berechtigungen zum Schreiben von Serverzugriffsprotokollen in den Bucket zu gewähren oder rückgängig zu machen, deaktivieren Sie unter S3-Protokollbereitstellungsgruppe die folgenden ACL Berechtigungen oder wählen Sie eine aus:

   - Objekte – Auflisten oder Schreiben
   - Bucket ACL — Lesen oder Schreiben

   Wenn ein Bucket als Ziel-Bucket für Zugriffsprotokolle eingerichtet ist, müssen die Bucket-Berechtigungen der Gruppe Log Delivery (Protokollbereitstellung) Schreibzugriff auf den Bucket erteilen. Wenn Sie die Server-Zugriffsprotokollierung für einen Bucket aktivieren, erteilt die Amazon-S3-Konsole der Gruppe Log Delivery (Protokollbereitstellung) Schreibzugriff

auf den Ziel-Bucket, den Sie für den Empfang der Protokolle ausgewählt haben. Weitere Informationen zu Server-Zugriffsprotokollen finden Sie unter Aktivieren Sie die Amazon-S3-Server-Zugriffsprotokollierung.

9. Gehen Sie wie folgt vor AWS-Konto, um einer anderen Person Zugriff zu gewähren:

   a. Wählen Sie Empfänger hinzufügen.

   b. Geben Sie im Feld Empfänger die kanonische ID des anderen AWS-Konto ein.

   c. Wählen Sie aus den folgenden ACL Berechtigungen aus:

      - Objekte – Auflisten oder Schreiben
      - Bucket ACL — Lesen oder Schreiben

   > ⚠ Warning
   >
   > Wenn Sie anderen Personen AWS-Konten Zugriff auf Ihre Ressourcen gewähren, sollten Sie sich bewusst sein, dass diese ihre Berechtigungen an Benutzer unter ihren Konten delegieren AWS-Konten können. Man spricht auch von einem kontenübergreifenden Zugriff. Informationen zur Verwendung des kontoübergreifenden Zugriffs finden Sie im Benutzerhandbuch unter Erstellen einer Rolle zur Delegierung von Berechtigungen an einen IAM Benutzer. IAM

10. Um einem anderen Benutzer den Zugriff zu entziehen AWS-Konto, wählen Sie unter Zugriff für andere AWS-Konten die Option Entfernen aus.

11. Klicken Sie auf Save changes (Änderungen speichern), um die Änderungen zu speichern.

Verwenden Sie die S3-Konsole, um ACL Berechtigungen für ein Objekt festzulegen

Die Konsole zeigt kombinierte Zugriffsberechtigungen für doppelte Berechtigungsempfänger an. Um die vollständige Liste von zu sehenACLs, verwenden Sie Amazon S3 RESTAPI, AWS CLI, oder AWS SDKs. Die folgende Tabelle zeigt die ACL Berechtigungen, die Sie für Objekte in der Amazon S3 S3-Konsole konfigurieren können.

Amazon S3 ACL S3-Konsolenberechtigungen für Objekte

Konsolenberechtigung	ACLErlaubnis	Zugriff
Objekt – Lesen	`READ`	Gestattet dem Empfänger, die Objektedaten und seine Metadaten zu lesen.
Objekt ACL — Lesen	`READ_ACP`	Ermöglicht dem Empfänger, das Objekt ACL zu lesen.
Objekt ACL — Schreiben	`WRITE_ACP`	Ermöglicht dem Empfänger, das ACL für das entsprechende Objekt zu schreiben

Weitere Hinweise zu ACL Berechtigungen finden Sie unterÜbersicht über die Zugriffskontrollliste (ACL).

> **Important**
>
> Wenn Ihr Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für S3 Object Ownership verwendet, müssen Sie Richtlinien verwenden, um Zugriff auf Ihren Bucket und die darin enthaltenen Objekte zu gewähren. Wenn die Einstellung Bucket Owner erforced aktiviert ist, schlagen Anfragen zur Einrichtung von Zugriffskontrolllisten (ACLs) oder zur Aktualisierung ACLs fehl und geben den `AccessControlListNotSupported` Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt.

Um ACL Berechtigungen für ein Objekt festzulegen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.
3. Wählen Sie in der Liste Objekte den Namen des Objekts aus, für das Sie Berechtigungen festlegen möchten.
4. Wählen Sie Permissions (Berechtigungen).
5. Wählen Sie unter Zugriffskontrollliste (ACL) die Option Bearbeiten aus.

   Sie können die folgenden ACL Berechtigungen für das Objekt bearbeiten:

Object

- Lesen – Gestattet dem Empfänger, die Objektedaten und seine Metadaten zu lesen.

Objekt ACL

- Lesen — Ermöglicht dem Empfänger, das Objekt ACL zu lesen.
- Schreiben — Ermöglicht dem Empfänger, das ACL für das entsprechende Objekt zu schreiben. In der S3-Konsole können Sie nur dem Bucket-Besitzer (Ihrem AWS-Konto) Schreibzugriff gewähren. Wir empfehlen dringend, anderen Empfängern keinen Schreibzugriff zu gewähren. Wenn Sie jedoch Schreibzugriff gewähren müssen, können Sie das AWS CLI AWS SDKs, oder das verwenden RESTAPI.

6. Sie können Zugriffsberechtigungen von Objekten für Folgendes verwalten:

   a. Zugriff für den Besitzer des Objekts

      Der Besitzer bezieht sich auf den Root-Benutzer des AWS-Kontos, und nicht auf einen AWS Identity and Access Management IAM Benutzer. Weitere Informationen zum Root-Benutzer finden Sie unter The Root-Benutzer des AWS-Kontos im IAMBenutzerhandbuch.

      Um die Objektzugriffsberechtigungen des Besitzers zu ändern, wählen Sie unter Zugriff für Objekteigentümer die Option Ihr AWS Konto (Eigentümer) aus.

      Aktivieren Sie die Kontrollkästchen für die Berechtigungen, die geändert werden sollen. Wählen Sie dann Save (Speichern) aus.

   b. Zugriff für andere AWS-Konten

      Um einem AWS Benutzer von einem anderen Benutzer Berechtigungen zu gewähren AWS-Konto, wählen Sie unter Zugriff für andere AWS-Konten die Option Konto hinzufügen aus. Geben Sie im Feld Geben Sie eine ID ein die kanonische ID des AWS Benutzers ein, dem Sie Objektberechtigungen erteilen möchten. Informationen zum Auffinden einer kanonischen ID finden Sie unter Ihre AWS-Konto Identifikatoren in der. Allgemeine Amazon Web Services-Referenz Sie können bis zu 99 Benutzer hinzufügen.

      Aktivieren Sie die Kontrollkästchen für die Berechtigungen, die dem Benutzer gewährt werden sollen. Wählen Sie dann Save (Speichern) aus. Um Informationen über die Berechtigungen anzuzeigen, wählen Sie die Hilfesymbole aus.

c. Öffentlicher Zugriff

Um der allgemeinen Öffentlichkeit (jedem Benutzer auf der ganzen Welt) Zugriff auf Ihr Objekt zu gewähren, wählen Sie unter Public access (Öffentlicher Zugriff) Everyone (Jeder) aus. Die Erteilung öffentlicher Zugriffsberechtigungen bedeutet, dass jeder Benutzer auf der ganzen Welt auf das Objekt zugreifen kann.

Aktivieren Sie die Kontrollkästchen für die Berechtigungen, die erteilt werden sollen. Wählen Sie dann Save (Speichern) aus.

> **Warning**
>
> - Seien Sie vorsichtig, wenn Sie der Gruppe Everyone (Jeder) anonymen Zugriff auf Ihre Amazon-S3-Objekte gewähren. Wenn Sie dieser Gruppe Zugriff gewähren, kann jeder auf der ganzen Welt auf Ihr Objekt zugreifen. Wenn Sie jedem Zugriff gewähren müssen, wird nachdrücklich empfohlen, dass Sie nur Berechtigungen für Read objects (Objekte lesen) gewähren.
> - Wir empfehlen nachdrücklich, der Gruppe Everyone (Jeder) keine Schreibberechtigungen für Objekte zu erteilen. Auf diese Weise kann jeder die ACL Berechtigungen für das Objekt überschreiben.

Mit dem AWS SDKs

Dieser Abschnitt enthält Beispiele für die Konfiguration von Zugriffskontrolllisten (ACL) für Buckets und Objekte.

> **Important**
>
> Wenn Ihr Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für S3 Object Ownership verwendet, müssen Sie Richtlinien verwenden, um Zugriff auf Ihren Bucket und die darin enthaltenen Objekte zu gewähren. Wenn die Einstellung Bucket Owner erforced aktiviert ist, schlagen Anfragen zur Einrichtung von Zugriffskontrolllisten (ACLs) oder zur Aktualisierung ACLs fehl und geben den `AccessControlListNotSupported` Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt.

Java

Dieser Abschnitt enthält Beispiele für die Konfiguration von Zugriffssteuerungslisten (ACL) für Buckets und Objekte. Im ersten Beispiel wird ein Bucket mit einem vordefinierten Wert erstellt ACL (siehe Eingemacht ACL), eine Liste mit benutzerdefinierten Berechtigungsberechtigungen erstellt und anschließend das Bucket durch ein Bucket ACL ersetzt, das die benutzerdefinierten Berechtigungen ACL enthält. Das zweite Beispiel zeigt, wie die `AccessControlList.grantPermission()` Methode geändert und ACL verwendet wird.

Example Erstellen Sie einen Bucket und geben Sie einen Bucket anACL, der der S3-Protokollzustellungsgruppe die entsprechenden Berechtigungen erteilt

Dieses Beispiel erstellt einen Bucket. In der Anfrage gibt das Beispiel einen Container an, der der Log Delivery-Gruppe die Berechtigung erteilt, Logs in den Bucket zu schreiben. ACL

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.IOException;
import java.util.ArrayList;

public class CreateBucketWithACL {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String userEmailForReadPermission = "*** user@example.com ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withRegion(clientRegion)
                    .build();

            // Create a bucket with a canned ACL. This ACL will be replaced by the
            // setBucketAcl()
            // calls below. It is included here for demonstration purposes.
            CreateBucketRequest createBucketRequest = new
 CreateBucketRequest(bucketName, clientRegion.getName())
```

```
                    .withCannedAcl(CannedAccessControlList.LogDeliveryWrite);
            s3Client.createBucket(createBucketRequest);

            // Create a collection of grants to add to the bucket.
            ArrayList<Grant> grantCollection = new ArrayList<Grant>();

            // Grant the account owner full control.
            Grant grant1 = new Grant(new
 CanonicalGrantee(s3Client.getS3AccountOwner().getId()),
                    Permission.FullControl);
            grantCollection.add(grant1);

            // Grant the LogDelivery group permission to write to the bucket.
            Grant grant2 = new Grant(GroupGrantee.LogDelivery, Permission.Write);
            grantCollection.add(grant2);

            // Save grants by replacing all current ACL grants with the two we just
 created.
            AccessControlList bucketAcl = new AccessControlList();
            bucketAcl.grantAllPermissions(grantCollection.toArray(new Grant[0]));
            s3Client.setBucketAcl(bucketName, bucketAcl);

            // Retrieve the bucket's ACL, add another grant, and then save the new
 ACL.
            AccessControlList newBucketAcl = s3Client.getBucketAcl(bucketName);
            Grant grant3 = new Grant(new
 EmailAddressGrantee(userEmailForReadPermission), Permission.Read);
            newBucketAcl.grantAllPermissions(grant3);
            s3Client.setBucketAcl(bucketName, newBucketAcl);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example Aktualisieren ACL Sie ein vorhandenes Objekt

In diesem Beispiel wird das ACL für ein Objekt aktualisiert. Das Beispiel führt die folgenden Aufgaben durch:

- Ruft die eines Objekts ab ACL
- Löscht die, ACL indem alle vorhandenen Berechtigungen entfernt werden
- Fügt zwei Berechtigungen hinzu: Vollzugriff für den Besitzer und WRITE _ ACP (siehe Welche Berechtigungen kann ich erteilen?) für einen Benutzer, der durch eine E-Mail-Adresse identifiziert wird
- Speichert ACL das im Objekt

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.AccessControlList;
import com.amazonaws.services.s3.model.CanonicalGrantee;
import com.amazonaws.services.s3.model.EmailAddressGrantee;
import com.amazonaws.services.s3.model.Permission;

import java.io.IOException;

public class ModifyACLExistingObject {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ***";
        String emailGrantee = "*** user@example.com ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Get the existing object ACL that we want to modify.
```

```
            AccessControlList acl = s3Client.getObjectAcl(bucketName, keyName);

            // Clear the existing list of grants.
            acl.getGrantsAsList().clear();

            // Grant a sample set of permissions, using the existing ACL owner for
 Full
            // Control permissions.
            acl.grantPermission(new CanonicalGrantee(acl.getOwner().getId()),
 Permission.FullControl);
            acl.grantPermission(new EmailAddressGrantee(emailGrantee),
 Permission.WriteAcp);

            // Save the modified ACL back to the object.
            s3Client.setObjectAcl(bucketName, keyName, acl);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Example Erstellen Sie einen Bucket und geben Sie einen Bucket anACL, der der S3-Protokollzustellungsgruppe Berechtigungen erteilt

Dieses C#-Beispiel erstellt einen Bucket. In der Anfrage gibt der Code auch einen Container an, der der Log Delivery-Gruppe die Berechtigung erteilt, die Logs in den Bucket zu schreiben. ACL

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem AWS SDK for. NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;
```

```
namespace Amazon.DocSamples.S3
{
    class ManagingBucketACLTest
    {
        private const string newBucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            CreateBucketUseCannedACLAsync().Wait();
        }

        private static async Task CreateBucketUseCannedACLAsync()
        {
            try
            {
                // Add bucket (specify canned ACL).
                PutBucketRequest putBucketRequest = new PutBucketRequest()
                {
                    BucketName = newBucketName,
                    BucketRegion = S3Region.EUW1, // S3Region.US,
                                                  // Add canned ACL.
                    CannedACL = S3CannedACL.LogDeliveryWrite
                };
                PutBucketResponse putBucketResponse = await
 client.PutBucketAsync(putBucketRequest);

                // Retrieve bucket ACL.
                GetACLResponse getACLResponse = await client.GetACLAsync(new
 GetACLRequest
                {
                    BucketName = newBucketName
                });
            }
            catch (AmazonS3Exception amazonS3Exception)
            {
                Console.WriteLine("S3 error occurred. Exception: " +
 amazonS3Exception.ToString());
            }
```

```
            catch (Exception e)
            {
                Console.WriteLine("Exception: " + e.ToString());
            }
        }
    }
}
```

Example Aktualisieren ACL Sie ein vorhandenes Objekt

In diesem C#-Beispiel wird das ACL für ein vorhandenes Objekt aktualisiert. Das Beispiel führt die folgenden Aufgaben durch:

- Ruft die eines Objekts ab. ACL
- Löscht die, ACL indem alle vorhandenen Berechtigungen entfernt werden.
- Fügt zwei Berechtigungen hinzu: Vollzugriff für den Besitzer und WRITE _ ACP für einen Benutzer, der anhand einer E-Mail-Adresse identifiziert wird.
- Speichert das, ACL indem es eine `PutAcl` Anfrage sendet.

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem AWS SDK for. NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class ManagingObjectACLTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** object key name ***";
        private const string emailAddress = "*** email address ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;
        public static void Main()
```

```
        {
            client = new AmazonS3Client(bucketRegion);
            TestObjectACLTestAsync().Wait();
        }
        private static async Task TestObjectACLTestAsync()
        {
            try
            {
                    // Retrieve the ACL for the object.
                    GetACLResponse aclResponse = await client.GetACLAsync(new
GetACLRequest
                    {
                        BucketName = bucketName,
                        Key = keyName
                    });

                    S3AccessControlList acl = aclResponse.AccessControlList;

                    // Retrieve the owner (we use this to re-add permissions after
we clear the ACL).
                    Owner owner = acl.Owner;

                    // Clear existing grants.
                    acl.Grants.Clear();

                    // Add a grant to reset the owner's full permission (the
previous clear statement removed all permissions).
                    S3Grant fullControlGrant = new S3Grant
                    {
                        Grantee = new S3Grantee { CanonicalUser = owner.Id },
                        Permission = S3Permission.FULL_CONTROL

                    };

                    // Describe the grant for the permission using an email address.
                    S3Grant grantUsingEmail = new S3Grant
                    {
                        Grantee = new S3Grantee { EmailAddress = emailAddress },
                        Permission = S3Permission.WRITE_ACP
                    };
                    acl.Grants.AddRange(new List<S3Grant> { fullControlGrant,
grantUsingEmail });

                    // Set a new ACL.
```

```
                    PutACLResponse response = await client.PutACLAsync(new
 PutACLRequest
                    {
                        BucketName = bucketName,
                        Key = keyName,
                        AccessControlList = acl
                    });
            }
            catch (AmazonS3Exception amazonS3Exception)
            {
                Console.WriteLine("An AmazonS3Exception was thrown. Exception: " +
 amazonS3Exception.ToString());
            }
            catch (Exception e)
            {
                Console.WriteLine("Exception: " + e.ToString());
            }
        }
    }
}
```

Mit dem REST API

Mit Amazon APIs S3 können Sie festlegen, ACL wann Sie einen Bucket oder ein Objekt erstellen. Amazon S3 bietet auch API die Möglichkeit, einen ACL auf einen vorhandenen Bucket oder ein Objekt zu setzen. Diese APIs bieten die folgenden Methoden zum Setzen einesACL:

- ACLMithilfe von Anforderungsheadern festlegen— Wenn Sie eine Anfrage zum Erstellen einer Ressource (Bucket oder Objekt) senden, legen Sie eine ACL mithilfe der Anforderungsheader fest. Mit diesen Headern können Sie entweder einen festgelegten Wert angeben ACL oder Grants explizit angeben (wobei der Empfänger und die Berechtigungen explizit identifiziert werden).
- ACLMithilfe des Anfragetextes festlegen — Wenn Sie eine Anfrage zum Einstellen eines für eine ACL vorhandene Ressource senden, können Sie dies ACL entweder im Anforderungsheader oder im Hauptteil der Anfrage festlegen.

Informationen zur REST API Unterstützung bei der Verwaltung ACLs finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- GETBucket-ACL

- PUTEimer ACL
- GETObjekt ACL
- PUTObjekt ACL
- PUTObjekt
- PUTEimer
- PUTObjekt — Kopieren
- Initiieren eines mehrteiligen Uploads

> **Important**
>
> Wenn Ihr Bucket die Einstellung „Vom Bucket-Besitzer erzwungen“ für S3 Object Ownership verwendet, müssen Sie Richtlinien für die Gewährung des Zugriffs auf Ihren Bucket und die enthaltenen Objekte verwenden. Wenn die Einstellung Bucket Owner erforced aktiviert ist, schlagen Anfragen zur Einrichtung von Zugriffskontrolllisten (ACLs) oder zur Aktualisierung ACLs fehl und geben den `AccessControlListNotSupported` Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt.

Spezifische Anforderungsheader für Access Control List (ACL)

Sie können Header verwenden, um auf Zugriffskontrolllisten (ACL) basierende Berechtigungen zu erteilen. Standardmäßig sind alle Objekte privat. Nur der Besitzer hat die vollständige Zugriffssteuerung. Wenn Sie ein neues Objekt hinzufügen, können Sie einzelnen AWS-Konten oder vordefinierten Gruppen, die von Amazon S3 definiert wurden, Berechtigungen gewähren. Diese Berechtigungen werden dann der Zugriffskontrollliste (ACL) für das Objekt hinzugefügt. Weitere Informationen finden Sie unter Übersicht über die Zugriffskontrollliste (ACL).

Mit dieser Operation können Sie Zugriffsberechtigungen mit einer der folgenden beiden Methoden erteilen:

- Canned ACL (**`x-amz-acl`**) — Amazon S3 unterstützt eine Reihe von vordefiniertenACLs, sogenannten CannedACLs. Jeder Scan ACL hat einen vordefinierten Satz von Empfängern und Berechtigungen. Weitere Informationen finden Sie unter Eingemacht ACL.
- Zugriffsberechtigungen — Um bestimmten AWS-Konten Gruppen explizit Zugriffsberechtigungen zu gewähren, verwenden Sie die folgenden Header. Jeder Header ist bestimmten Berechtigungen zugeordnet, die Amazon S3 in einem unterstütztACL. Weitere Informationen finden Sie unter

Übersicht über die Zugriffskontrollliste (ACL). Im Header geben Sie eine Liste der Empfänger an, die die jeweilige Berechtigung erhalten.

- x-amz-grant-read
- x-amz-grant-write
- x-amz-grant-read-acp
- x-amz-grant-write-acp
- x-amz-grant-full-kontrollieren

Mit dem AWS CLI

Weitere Informationen zur Verwaltung ACLs mit dem AWS CLI finden Sie put-bucket-aclin der AWS CLI Befehlsreferenz.

> **Important**
>
> Wenn Ihr Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für S3 Object Ownership verwendet, müssen Sie Richtlinien verwenden, um Zugriff auf Ihren Bucket und die darin enthaltenen Objekte zu gewähren. Wenn die Einstellung Bucket Owner erforced aktiviert ist, schlagen Anfragen zur Einrichtung von Zugriffskontrolllisten (ACLs) oder zur Aktualisierung ACLs fehl und geben den `AccessControlListNotSupported` Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt.

Beispiele für Richtlinien für ACLs

Sie können Bedingungsschlüssel in Bucket-Richtlinien verwenden, um den Zugriff auf Amazon S3 zu kontrollieren.

Themen

- Gewährung von PutObject s3-Berechtigungen mit einer Bedingung, die voraussetzt, dass der Bucket-Besitzer die volle Kontrolle erhält
- Erteilung der PutObject s3-Berechtigung mit einer Bedingung für den x-amz-acl Header

Gewährung von PutObject s3-Berechtigungen mit einer Bedingung, die voraussetzt, dass der Bucket-Besitzer die volle Kontrolle erhält

Die Operation PUTObjekt ermöglicht eine Zugriffskontrollliste (ACL), d. h. Header, die Sie verwenden können, um bestimmte Berechtigungen zu erteilenACL. Mit diesen Schlüsseln kann der Bucket-Eigentümer eine Bedingung festlegen, die bestimmte Zugriffsberechtigungen erfordert, wenn der Benutzer ein Objekt hochlädt.

Angenommen, Konto A besitzt einen Bucket und der Kontoadministrator möchte Dave, einem Benutzer in Konto B, Berechtigungen zum Hochladen von Objekten erteilen. Standardmäßig gehören Objekte, die Dave hochgeladen hat, zu Konto B, und Konto A besitzt keine Berechtigungen für diese Objekte. Da der Bucket-Eigentümer die Rechnungen bezahlt, benötigt er volle Berechtigungen für die Objekte, die Dave hochlädt. Das Konto Ein Administrator kann dies tun, indem er Dave die `s3:PutObject` entsprechende Berechtigung erteilt, allerdings mit der Bedingung, dass die Anfrage ACL spezifische Header enthält, die entweder explizit volle Rechte gewähren oder vordefinierte Header verwenden. ACL Weitere Informationen finden Sie unter PUT Objekt.

Erfordert den x-amz-full-control Header

Sie können in der Anforderung den Header `x-amz-full-control` mit vollständiger Kontrollberechtigung für den Bucket-Eigentümer anfordern. Die folgende Bucket-Richtlinie erteilt dem Benutzer Dave die Berechtigung `s3:PutObject` mit einer Bedingung, die den Bedingungsschlüssel `s3:x-amz-grant-full-control` enthält, in die Anforderung den Header `x-amz-full-control` aufzunehmen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountB-ID:user/Dave"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::awsexamplebucket1/*",
      "Condition": {
        "StringEquals": {
          "s3:x-amz-grant-full-control": "id=AccountA-CanonicalUserID"
        }
      }
```

```
    }
  ]
}
```

> **Note**
>
> In diesem Beispiel geht es um die kontoübergreifende Berechtigung. Wenn Dave (der die Erlaubnis erhält) jedoch zu demjenigen gehört AWS-Konto , dem der Bucket gehört, ist diese bedingte Berechtigung nicht erforderlich. Grund hierfür ist, dass das übergeordnete Konto, dem Dave angehört, Eigentümer von Objekten ist, die der Benutzer hochlädt.

Hinzufügen einer expliziten Zugriffsverweigerung

Die vorherige Bucket-Richtlinie erteilt dem Benutzer Dave in Konto B eine bedingte Berechtigung. Während diese Richtlinie in Kraft ist, ist es Dave möglich, die gleiche Berechtigung ohne Bedingung über eine andere Richtlinie zu erhalten. Beispiel: Dave kann einer Gruppe angehören, der Sie die Berechtigung `s3:PutObject` bedingungslos erteilen. Um solche Berechtigungslücken zu vermeiden, können Sie eine strengere Zugriffsrichtlinie schreiben, indem Sie eine explizite Zugriffsverweigerung hinzufügen. In diesem Beispiel verweigern Sie explizit die Upload-Berechtigung für Benutzer Dave, wenn er nicht die erforderlichen Header in die Anforderung einbindet, die dem Bucket-Eigentümer volle Berechtigungen gewährt. Eine explizite Verweigerung ersetzt immer eine anderswo erteilte Erlaubnis. Im Folgenden finden Sie das geänderte Zugriffsrichtlinienbeispiel mit hinzugefügter expliziter Ablehnung.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "statement1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::awsexamplebucket1/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-grant-full-control": "id=AccountA-CanonicalUserID"
                }
```

```
            }
        },
        {
            "Sid": "statement2",
            "Effect": "Deny",
            "Principal": {
                "AWS": "arn:aws:iam::AccountB-ID:user/AccountBadmin"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::awsexamplebucket1/*",
            "Condition": {
                "StringNotEquals": {
                    "s3:x-amz-grant-full-control": "id=AccountA-CanonicalUserID"
                }
            }
        }
    ]
}
```

Testen Sie die Richtlinie mit AWS CLI

Wenn Sie zwei haben AWS-Konten, können Sie die Richtlinie mit dem AWS Command Line Interface (AWS CLI) testen. Sie hängen die Richtlinie an und verwenden Daves Anmeldeinformationen, um die Berechtigung mit dem folgenden AWS CLI `put-object` Befehl zu testen. Sie stellen dem Benutzer Dave die Anmeldeinformationen bereit, indem Sie den Parameter `--profile` hinzufügen. Sie erteilen dem Bucket-Eigentümer die volle Kontrolle, indem Sie den Parameter `--grant-full-control` hinzufügen. Weitere Informationen zur Einrichtung und Verwendung von finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS CLI in der Amazon S3 API S3-Referenz. AWS CLI

```
aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body c:\HappyFace.jpg
 --grant-full-control id="AccountA-CanonicalUserID" --profile AccountBUserProfile
```

Erfordert den x-amz-acl Header

Sie können den `x-amz-acl` Header mit einem voreingestellten Header anfordern, der dem Bucket-Besitzer volle Zugriffsrechte ACL gewährt. Um den Header `x-amz-acl` in der Anforderung erforderlich zu machen, können Sie das Schlüssel-Wert-Paar im Block `Condition` ersetzen und den Bedingungsschlüssel `s3:x-amz-acl` wie im folgenden Beispiel dargestellt angeben.

```
"Condition": {
```

```
    "StringEquals": {
        "s3:x-amz-acl": "bucket-owner-full-control"
    }
}
```

Um die Berechtigung mit dem zu testen AWS CLI, geben Sie den `--acl` Parameter an. Der fügt AWS CLI dann den `x-amz-acl` Header hinzu, wenn er die Anfrage sendet.

```
aws s3api put-object --bucket examplebucket --key HappyFace.jpg --body c:\HappyFace.jpg
 --acl "bucket-owner-full-control" --profile AccountBadmin
```

Erteilung der PutObject s3-Berechtigung mit einer Bedingung für den x-amz-acl Header

Die folgende Bucket-Richtlinie gewährt die `s3:PutObject` Erlaubnis für zwei, AWS-Konten wenn die Anfrage den `x-amz-acl` Header enthält, der das Objekt öffentlich lesbar macht. Der `Condition`-Block verwendet die `StringEquals`-Bedingung und ist mit dem Schlüssel-Wert-Paar `"s3:x-amz-acl":["public-read"]` zur Auswertung versehen. Im Schlüssel-Wert-Paar ist `s3:x-amz-acl` ein Amazon S3–spezifischer Schlüssel, wie durch das Präfix `s3:` angezeigt ist.

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
         "Sid":"AddCannedAcl",
            "Effect":"Allow",
         "Principal": {
                "AWS": [
                "arn:aws:iam::Account1-ID:root",
    "arn:aws:iam::Account2-ID:root"
    ]
            },
         "Action":"s3:PutObject",
            "Resource": ["arn:aws:s3:::awsexamplebucket1/*"],
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl":["public-read"]
                }
            }
        }
    ]
```

```
}
```

> **Important**
>
> Nicht alle Bedingungen machen für alle Aktionen auch Sinn. Beispielsweise ist es sinnvoll, die Bedingung `s3:LocationConstraint` für eine Richtlinie einzufügen, die die Amazon-S3-Berechtigung `s3:CreateBucket` erteilt. Es ist jedoch nicht sinnvoll, diese Bedingung in eine Richtlinie aufzunehmen, die die `s3:GetObject`-Genehmigung erteilt. Amazon S3 kann auf semantische Fehler dieses Typs testen, die Amazon-S3-spezifische Bedingungen beinhalten. Wenn Sie jedoch eine Richtlinie für einen IAM Benutzer oder eine Rolle erstellen und eine semantisch ungültige Amazon S3 S3-Bedingung angeben, wird kein Fehler gemeldet, da die Amazon S3 S3-Bedingungen IAM nicht validiert werden können.

Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher

Die Amazon S3 Block Public Access-Funktion bietet Einstellungen für Zugriffspunkte, Buckets und Konten, mit denen Sie den öffentlichen Zugriff auf Amazon-S3-Ressourcen verwalten können. Standardmäßig erlauben neue Buckets, Zugriffspunkte und Objekte keinen öffentlichen Zugriff. Benutzer können jedoch Bucket-Richtlinien, Zugriffspunkt-Richtlinien oder Objektberechtigungen ändern, um öffentlichen Zugriff zu ermöglichen. S3 Block Public Access-Einstellungen überschreiben diese Richtlinien und Berechtigungen, damit Sie den öffentlichen Zugriff auf diese Ressourcen einschränken können.

Mit S3 Block Public Access können Administratoren und Bucket-Eigentümer problemlos zentrale Kontrollen zur Begrenzung des öffentlichen Zugriffs auf ihre Amazon-S3-Ressourcen einrichten. Diese Kontrollen werden unabhängig davon durchgesetzt, wie die Ressourcen erstellt wurden.

Anweisungen zur Konfiguration von Public Block Access finden Sie unter Verwenden von Block Public Access.

Wenn Amazon S3 eine Anforderung zum Zugriff auf einen Bucket oder ein Objekt erhält, wird ermittelt, ob für den Bucket oder das Konto des Bucket-Eigentümers eine Block Public Access-Einstellung vorliegt. Wenn die Anforderung über einen Zugriffspunkt einging, prüft Amazon S3 auch auf Block Public Access-Einstellungen für den Zugriffspunkt. Wenn eine Block Public Access-Einstellung vorhanden ist, die den angeforderten Zugriff verbietet, lehnt Amazon S3 die Anforderung ab.

Amazon S3 Block Public Access bietet vier Einstellungen. Diese Einstellungen sind voneinander unabhängig und können in beliebiger Kombination verwendet werden. Jede Einstellung kann auf einen Zugriffspunkt, einen Bucket oder ein gesamtes AWS-Konto angewendet werden. Wenn sich die Block Public Access-Einstellungen für den Zugriffspunkt, den Bucket oder das Konto unterscheiden, wendet Amazon S3 die restriktivste Kombination der Zugriffspunkt-, Bucket- und Kontoeinstellungen an.

Wenn Amazon S3 ermittelt, ob eine Operation von einer Block Public Access-Einstellung untersagt wird, werden alle Anforderungen abgelehnt, die gegen eine Zugriffspunkt-, Bucket- oder Konto-Einstellung verstoßen.

Important

Öffentlicher Zugriff auf Buckets und Objekte wird über Zugriffskontrolllisten (ACLs), Zugriffspunktrichtlinien, Bucket-Richtlinien oder alles gewährt. Um sicherzustellen, dass der öffentliche Zugriff für alle Amazon-S3-Zugriffspunkte, -Buckets und -Objekte blockiert ist, empfehlen wir, alle vier Einstellungen zur Blockierung des öffentlichen Zugriffs für das Konto zu aktivieren. Diese Einstellungen blockieren den öffentlichen Zugriff für alle aktuellen und künftigen Buckets und Zugriffspunkte.

Bevor Sie diese Einstellungen anwenden, verifizieren Sie, dass Ihre Anwendungen ohne öffentlichen Zugriff korrekt funktionieren. Wenn ein bestimmter Umfang an öffentlichem Zugriff auf Ihre Buckets oder Objekte nötig ist, z. B. zum Hosten einer statischen Website, wie unter Hosten einer statischen Website mit Amazon S3 beschrieben, können Sie die einzelnen Einstellungen an Ihre Speicheranwendungsfälle anpassen.

Die Aktivierung von Block Public Access trägt zum Schutz Ihrer Ressourcen bei, indem verhindert wird, dass öffentlicher Zugriff über die Ressourcenrichtlinien oder Zugriffskontrolllisten (ACLs) gewährt wird, die direkt mit S3-Ressourcen verknüpft sind.

Überprüfen Sie zusätzlich zur Aktivierung von Block Public Access sorgfältig die folgenden Richtlinien, um sicherzustellen, dass sie keinen öffentlichen Zugriff gewähren:

- Identitätsbasierte Richtlinien, die mit zugehörigen AWS Prinzipalen verknüpft sind (z. B. Rollen) IAM
- Ressourcenbasierte Richtlinien, die mit zugehörigen AWS Ressourcen verknüpft sind (z. B. () -Schlüssel) AWS Key Management Service KMS

> **Note**
>
> - Sie können die Einstellungen für Block Public Access nur für Zugangspunkte, Buckets und AWS-Konten aktivieren. Amazon S3 unterstützt keine Block Public Access-Einstellungen für den öffentlichen Zugriff pro Objekt.
> - Wenn Sie Einstellungen zum Sperren des öffentlichen Zugriffs auf ein Konto anwenden, gelten die Einstellungen für alle AWS-Regionen Konten weltweit. Die Einstellungen werden möglicherweise nicht in allen Regionen umgehend oder gleichzeitig wirksam, werden aber auf jeden Fall von allen Regionen übernommen.

Themen

- Block Public Access-Einstellungen
- Durchführen von Block Public Access-Vorgänge an einem Zugriffspunkt
- Die Bedeutung von „öffentlich“
- Verwenden von IAM Access Analyzer für S3 zur Überprüfung öffentlicher Buckets
- Berechtigungen
- Verwenden von Block Public Access
- Konfigurieren der Block-Public-Access-Einstellungen für Ihr Konto
- Konfigurieren von Block-Public-Access-Einstellungen für Ihre S3-Buckets

Block Public Access-Einstellungen

S3 Block Public Access bietet vier Einstellungen. Sie können diese Einstellungen in beliebiger Kombination auf einzelne Zugriffspunkte, Buckets oder auf ganze AWS-Konten anwenden. Wenn Sie eine Einstellung auf ein Konto anwenden, gilt sie für alle Buckets und Zugriffspunkte, die dem Konto gehören. Wenn Sie eine Einstellung auf einen Bucket anwenden, gilt diese auch für alle Zugriffspunkte, die diesem Bucket zugeordnet sind.

Die folgende Tabelle enthält die verfügbaren Einstellungen.

Name	Beschreibung
`BlockPublicAcls`	Das Festlegen dieser Option auf `TRUE` hat folgende Verhaltensweise zur Folge:

Name	Beschreibung
	• PUTBucket-ACL- und PUT Object-ACL-Aufrufe schlagen fehl, wenn die angegebene Zugriffskontrollliste (ACL) öffentlich ist. • PUTObjektaufrufe schlagen fehl, wenn die Anfrage eine öffentliche Komponente enthältACL. • Wenn diese Einstellung auf ein Konto angewendet wird, schlagen PUT Bucket-Aufrufe fehl, wenn die Anfrage ein öffentliches Objekt beinhalte tACL. Wenn diese Einstellung auf gesetzt istTRUE, schlagen die angegebenen Operationen fehl (unabhängig davon, ob sie über REST API AWS CLI, oder ausgeführt werden AWS SDKs). Bestehende Richtlinien sowie Richtlinien ACLs für Buckets und Objekte werden jedoch nicht geändert. Mit dieser Einstellung können Sie sich vor öffentlichem Zugriff schützen und gleichzei tig die vorhandenen Richtlinien sowie Ihre Buckets und ACLs Objekte prüfen, verfeinern oder anderweitig ändern. **Note** Access Points sind ihnen nicht ACLs zugeordnet. Wenn Sie diese Einstellung auf einen Zugriffspunkt anwenden, fungiert sie als Passthrough zum zugrunde liegenden Bucket. Wenn diese Einstellu ng für einen Zugriffspunkt aktiviert ist, verhalten sich Anforderungen, die über den Zugriffspunkt vorgenommen werden, so als ob der zugrunde liegende Bucket diese Einstellung aktiviert hat, unabhängi g davon, ob diese Einstellung für den Bucket tatsächlich aktiviert ist.

Name	Beschreibung
`IgnorePublicAcls`	Wenn Sie diese Option `TRUE` ACLs auf setzen, ignoriert Amazon S3 alle öffentlichen Informationen in einem Bucket und alle darin enthaltenen Objekte. Mit dieser Einstellung können Sie öffentlichen Zugriff, der von gewährt wird, sicher blockieren ACLs und gleichzeitig PUT Objekt-Aufrufe zulassen, die ein öffentliches Objekt enthalten ACL (im Gegensatz zu PUT ObjektaufrufenBlockPublicAcls , die ein öffentliches Objekt enthaltenACL). Die Aktivierung dieser Einstellung hat keinen Einfluss auf die Persistenz vorhandener Einstellungen ACLs und verhindert nicht, dass neue öffentliche Objekte eingerichtet ACLs werden. **Note** Access Points sind ihnen nicht ACLs zugeordnet. Wenn Sie diese Einstellung auf einen Zugriffspunkt anwenden, fungiert sie als Passthrough zum zugrunde liegenden Bucket. Wenn diese Einstellung für einen Zugriffspunkt aktiviert ist, verhalten sich Anforderungen, die über den Zugriffspunkt vorgenommen werden, so als ob der zugrunde liegende Bucket diese Einstellung aktiviert hat, unabhängig davon, ob diese Einstellung für den Bucket tatsächlich aktiviert ist.

Name	Beschreibung
`BlockPublicPolicy`	Wenn Sie diese Option auf `TRUE` für einen Bucket setzen, lehnt Amazon S3 Aufrufe der PUT Bucket-Richtlinie ab, wenn die angegebene Bucket-Richtlinie öffentlichen Zugriff zulässt. Wenn diese Option auf `TRUE` für einen Bucket gesetzt wird, lehnt Amazon S3 auch Aufrufe der PUT Access Point-Richtlinie für alle Access Points des Buckets ab, die dasselbe Konto haben, wenn die angegebene Richtlinie öffentlichen Zugriff zulässt. Wenn Sie diese Option auf `TRUE` für einen Access Point setzen, lehnt Amazon S3 Aufrufe von PUT Access Point Policy und PUT Bucket Policy ab, die über den Access Point getätigt werden, wenn die angegebene Richtlinie (entweder für den Access Point oder den zugrunde liegenden Bucket) den öffentlichen Zugriff zulässt. Mit dieser Einstellung können Sie Benutzern erlauben, Zugriffspunkt- und Bucket-Richtlinien zu verwalten, ohne ihnen die öffentliche Freigabe des Buckets oder der darin enthaltenen Objekte zu gestatten. Die Aktivierung dieser Einstellung hat keine Auswirkungen auf vorhandene Zugriffspunkt- oder Bucket-Richtlinien. **Important** Um diese Einstellung effektiv zu nutzen, empfehlen wir Ihnen, sie auf Konto-Ebene anzuwenden. Eine Bucket-Richtlinie kann Benutzern das Ändern der Block Public Access-Einstellungen eines Buckets gestatten. Daher könnten Benutzer mit der Berechtigung zum Ändern einer Bucket-Richtlinie eine Richtlinie einfügen, die es ihnen erlaubt, die Block Public Access-Einstellungen für den Bucket zu deaktivieren. Wenn diese Einstellung dann für das ganze Konto aktiviert wird statt für einen bestimmten Bucket, blockiert Amazon S3 öffentliche Richtlinien selbst dann, wenn ein Benutzer die Bucket-Richtlinie so ändert, dass diese Einstellung deaktiviert wird.

Name	Beschreibung
`RestrictPublicBuckets`	Wenn Sie diese Option so einstellen, wird der Zugriff auf einen Access Point oder Bucket mit einer öffentlichen Richtlinie auf AWS Service Principals und autorisierte Benutzer innerhalb des Kontos des Bucket-Besitzers und des Accounts des Access Point-Besitzers `TRUE` beschränkt. Mit dieser Einstellung wird der gesamte kontoübergreifende Zugriff auf den Access Point oder Bucket blockiert (außer durch AWS Service Principals), während Benutzer innerhalb des Kontos den Access Point oder Bucket weiterhin verwalten können. Die Aktivierung dieser Einstellung wirkt sich nicht auf vorhandene Zugriffspunkt- oder Bucket-Richtlinien aus – mit folgender Ausnahme: Amazon S3 blockiert den öffentlichen und kontenübergreifenden Zugriff, der von einer öffentlichen Zugriffspunkt- oder Bucket-Richtlinie abgeleitet wird, darunter auch die nicht-öffentliche Delegation auf bestimmte Konten.

Important

- Aufrufe von GET Bucket ACL und GET Object ACL geben immer die effektiven Berechtigungen zurück, die für den angegebenen Bucket oder das angegebene Objekt gelten. Nehmen wir zum Beispiel an, ein Bucket hat einen BucketACL, der öffentlichen Zugriff gewährt, aber für den Bucket ist auch die `IgnorePublicAcls` Einstellung aktiviert. In diesem Fall gibt GET Bucket acl eine ACL zurück, die die Zugriffsberechtigungen widerspiegelt, die Amazon S3 erzwingt, und nicht die tatsächlichen, ACL die dem Bucket zugeordnet sind.
- Die Einstellungen zum Blockieren des öffentlichen Zugriffs ändern keine bestehenden Richtlinien oderACLs. Daher führt das Entfernen einer Einstellung zum Blockieren des öffentlichen Zugriffs dazu, dass ein Bucket oder ein Objekt mit einer öffentlichen Richtlinie oder ACL wieder öffentlich zugänglich ist.

Durchführen von Block Public Access-Vorgänge an einem Zugriffspunkt

Verwenden Sie den AWS CLI Dienst, um Operationen zum Blockieren des öffentlichen Zugriffs an einem Access Point durchzuführen`s3control`.

> **Important**
>
> Beachten Sie, dass es derzeit nicht möglich ist, die Block Public Accesss-Einstellungen eines Zugriffspunkts nach dem Erstellen des Zugriffspunkts zu ändern. Die einzige Möglichkeit, Block Public Access-Einstellungen für einen Zugriffspunkt anzugeben, besteht darin, diese beim Erstellen des Zugriffspunkts einzuschließen.

Die Bedeutung von „öffentlich“

ACLs

Amazon S3 betrachtet einen Bucket oder ein Objekt als ACL öffentlich, wenn es Mitgliedern der vordefinierten `AuthenticatedUsers` Gruppen `AllUsers` oder Gruppen Berechtigungen gewährt. Weitere Informationen zu vordefinierten Gruppen finden Sie unter Vordefinierte Gruppen in Amazon S3.

Bucket-Richtlinien

Bei der Evaluierung einer Bucket-Richtlinie beginnt Amazon S3 mit der Annahme, dass die Richtlinie öffentlich ist. Dann evaluiert es die Richtlinie, um festzustellen, ob sie als nicht-öffentlich eingestuft werden kann. Um als nicht öffentlich zu gelten, darf eine Bucket-Richtlinie nur Zugriff auf feste Werte (Werte, die keine Platzhalter oder eine AWS Identity and Access Management -Richtlinienvariable enthalten) gewähren, für einen oder mehrere der folgenden Werte:

- Ein AWS Principal, ein Benutzer, eine Rolle oder ein Dienstprinzipal (z. B.`aws:PrincipalOrgID`)
- Eine Reihe von klassenlosen Inter-Domain Routings (CIDRs), unter Verwendung von. `aws:SourceIp` Weitere Informationen zu CIDR finden Sie unter RFC4632 auf der Editor-Website. RFC

> **Note**
>
> Bucket-Richtlinien, die abhängig vom `aws:SourceIp`-Bedingungsschlüssel Zugriff mit sehr breiten IP-Bereichen gewähren (z. B. 0.0.0.0/1), werden als „öffentlich“ bewertet. Dies schließt Werte ein, die breiter sind als /8 für IPv4 und /32 für IPv6 (ohne RFC1918 private Bereiche). Die Funktion zum Blockieren des öffentlichen Zugriffs lehnt diese „öffentlichen“ Richtlinien ab und verhindert den kontoübergreifenden Zugriff auf Buckets, die diese „öffentlichen“ Richtlinien bereits verwenden.

- `aws:SourceArn`
- `aws:SourceVpc`
- `aws:SourceVpce`
- `aws:SourceOwner`
- `aws:SourceAccount`
- `aws:userid`, außerhalb des Musters `"AROLEID:*"`
- `s3:DataAccessPointArn`

> **Note**
>
> Bei Verwendung in einer Bucket-Richtlinie kann dieser Wert einen Platzhalter für den Namen des Zugriffspunkts enthalten, ohne die Richtlinie öffentlich zu machen, solange die Konto-ID fixiert ist. Das Zulassen des Zugriffs auf `arn:aws:s3:us-west-2:123456789012:accesspoint/*` würde beispielsweise den Zugriff auf jeden Zugriffspunkt ermöglichen, der dem Konto `123456789012` in Region `us-west-2` zugeordnet ist, ohne die Bucket-Richtlinie öffentlich zu machen. Beachten Sie, dass sich dieses Verhalten von dem von Zugriffspunkt-Richtlinien unterscheidet. Weitere Informationen finden Sie unter Zugriffspunkte.

- `s3:DataAccessPointAccount`

Weitere Informationen zu Bucket-Richtlinien finden Sie unter Bucket-Richtlinien für Amazon S3.

Example : Öffentliche Bucket-Richtlinien

Nach diesen Regeln gelten die folgenden Beispielrichtlinien als öffentlich.

```
{
  "Principal": "*",
  "Resource": "*",
  "Action": "s3:PutObject",
  "Effect": "Allow"
 }
```

```
{
  "Principal": "*",
  "Resource": "*",
```

```
  "Action": "s3:PutObject",
  "Effect": "Allow",
  "Condition": { "StringLike": {"aws:SourceVpc": "vpc-*"}}
 }
```

Diese Richtlinien können zu nicht-öffentlichen Richtlinien gemacht werden, indem einer der vorgenannten Bedingungsschlüssel unter Verwendung eines festen Wertes eingefügt wird. So können Sie beispielsweise die letzte oben angeführte Richtlinie zu einer nicht-öffentlichen Richtlinie machen, indem Sie `aws:SourceVpc` wie folgt auf einen festen Wert festlegen.

```
{
  "Principal": "*",
  "Resource": "*",
  "Action": "s3:PutObject",
  "Effect": "Allow",
  "Condition": {"StringEquals": {"aws:SourceVpc": "vpc-91237329"}}
 }
```

So wertet Amazon S3 eine Bucket-Richtlinie aus, die Berechtigungen sowohl für den öffentlichen wie auch für den nicht-öffentlichen Zugriff enthält

Dieses Beispiel zeigt, wie Amazon S3 eine Bucket-Richtlinie bewertet, die Berechtigungen sowohl für den öffentlichen wie auch für den nicht-öffentlichen Zugriff enthält.

Nehmen wir an, dass ein Bucket eine Richtlinie aufweist, die den Zugriff auf einen Satz an festen Prinzipalen gewährt. Nach den zuvor beschriebenen Regeln gilt diese Richtlinie nicht als öffentlich. Wenn Sie daher die Einstellung `RestrictPublicBuckets` aktivieren, bleibt die Richtlinie wie geschrieben in Kraft, da `RestrictPublicBuckets` nur für Buckets mit öffentlichen Richtlinien gilt. Wenn Sie jedoch der Richtlinie eine öffentliche Anweisung hinzufügen, wird `RestrictPublicBuckets` für den Bucket wirksam. Es ermöglicht nur AWS Serviceprinzipalen und autorisierten Benutzern des Kontos des Bucket-Besitzers den Zugriff auf den Bucket.

Nehmen wir beispielsweise an, dass ein Bucket, der „Konto-1“ gehört, eine Richtlinie aufweist, die Folgendes enthält:

1. Eine Anweisung, die Zugriff auf gewährt AWS CloudTrail (was ein AWS Dienstprinzipal ist)
2. eine Anweisung, die Konto "Konto-2" Zugriff gewährt
3. eine Anweisung, die der Öffentlichkeit Zugriff gewährt, z. B. durch das Festlegen von `"Principal": "*"` ohne einschränkende `Condition`

Diese Richtlinie gilt wegen der dritten Anweisung als öffentlich. Wenn diese Richtlinie eingerichtet und `RestrictPublicBuckets` aktiviert ist, ermöglicht Amazon S3 den Zugriff nur von CloudTrail. Beachten Sie: Obwohl die zweite Anweisung nicht öffentlich ist, deaktiviert Amazon S3 den Zugriff durch „Account-2“. Das liegt daran, dass die dritte Anweisung die gesamte Richtlinie zu einer öffentlichen Richtlinie macht, so dass `RestrictPublicBuckets` gilt. Somit deaktiviert Amazon S3 den kontoübergreifenden Zugriff, obwohl die Richtlinie den Zugriff an ein bestimmtes Konto („Account-2“) delegiert. Wenn Sie aber die dritte Anweisung aus der Richtlinie entfernen, gilt die Richtlinie nicht als öffentlich, und `RestrictPublicBuckets` ist nicht mehr gültig. Somit erhält „Konto-2“ wieder Zugriff auf den Bucket, selbst wenn Sie `RestrictPublicBuckets` aktiviert lassen.

Zugriffspunkte

Amazon S3 wertet Block Public Access-Einstellungen für Zugriffspunkte geringfügig anders als für Buckets aus. Die Regeln, die Amazon S3 anwendet, um zu bestimmen, wann eine Zugriffspunkt-Richtlinie öffentlich ist, sind für Zugriffspunkte im Allgemeinen dieselben wie für Buckets, außer in den folgenden Situationen:

- Ein Access Point, der aus dem VPC Netzwerk stammt, gilt immer als nicht öffentlich, unabhängig vom Inhalt seiner Zugriffspunkt-Richtlinie.
- Eine Zugriffspunktrichtlinie, die Zugriff auf eine Gruppe von Zugriffspunkten unter Verwendung von `s3:DataAccessPointArn` gewährt, gilt als öffentlich. Beachten Sie, dass sich dieses Verhalten von dem von Bucket-Richtlinien unterscheidet. Beispielsweise wird eine Bucket-Richtlinie, die Zugriff auf `s3:DataAccessPointArn`-Werte gewährt, die `arn:aws:s3:us-west-2:123456789012:accesspoint/*` entsprechen, nicht als öffentlich betrachtet. Dieselbe Anweisung in einer Zugriffspunkt-Richtlinie würde jedoch den Zugriffspunkt öffentlich machen.

Verwenden von IAM Access Analyzer für S3 zur Überprüfung öffentlicher Buckets

Sie können IAM Access Analyzer for S3 verwenden, um Buckets mit BucketsACLs, Bucket-Richtlinien oder Access Point-Richtlinien zu überprüfen, die öffentlichen Zugriff gewähren. IAMAccess Analyzer for S3 warnt Sie vor Buckets, die so konfiguriert sind, dass sie allen Personen im Internet oder auf anderen Websites AWS-Konten, auch AWS-Konten außerhalb Ihrer Organisation, Zugriff gewähren. Für jeden öffentlichen oder freigegebenen Bucket erhalten Sie Ergebnisse, die die Quelle und die Ebene des öffentlichen oder freigegebenen Zugriffs melden.

In IAM Access Analyzer for S3 können Sie den gesamten öffentlichen Zugriff auf einen Bucket mit einem einzigen Klick blockieren. Sie können auch einen Drilldown in die Berechtigungseinstellungen auf Bucket-Ebene ausführen, um detaillierte Zugriffsebenen zu konfigurieren. Für bestimmte und geprüfte Anwendungsfälle, die öffentlichen oder freigegebenen Zugriff erfordern, können Sie Ihre Absicht bestätigen und aufzeichnen, dass der Bucket öffentlich oder freigegeben bleibt, indem Sie die Ergebnisse für den Bucket archivieren.

In seltenen Fällen meldet IAM Access Analyzer for S3 möglicherweise keine Ergebnisse für einen Bucket, den eine Amazon S3 S3-Block-Bewertung für öffentlichen Zugriff als öffentlich meldet. Dies ist der Fall, da Amazon S3 Block Public Access Richtlinien für aktuelle Aktionen und potenzielle Aktionen, die in Zukunft hinzugefügt werden könnten und die dazu führen könnten, dass ein Bucket öffentlich wird, überprüft. Andererseits analysiert IAM Access Analyzer for S3 nur die aktuellen Aktionen, die für den Amazon S3 S3-Service bei der Bewertung des Zugriffsstatus angegeben wurden.

Weitere Informationen zu IAM Access Analyzer for S3 finden Sie unterÜberprüfung des Bucket-Zugriffs mit IAM Access Analyzer für S3.

Berechtigungen

Um die Funktionen von Amazon S3 Block Public Access zu nutzen, benötigen Sie die folgenden Berechtigungen:

Operation	Erforderliche Berechtigungen
GETStatus der Bucket-Richtlinie	`s3:GetBucketPolicyStatus`
GETEinstellungen für den Bucket „Öffentlichen Zugriff blockieren“	`s3:GetBucketPublicAccessBlock`
PUTBucket Einstellungen für den öffentlichen Zugriff blockieren	`s3:PutBucketPublicAccessBlock`
DELETEBucket Einstellungen für den öffentlichen Zugriff blockieren	`s3:PutBucketPublicAccessBlock`
GETEinstellungen für das Konto „Öffentlichen Zugriff blockieren“	`s3:GetAccountPublicAccessBlock`

Operation	Erforderliche Berechtigungen
PUTKonto Einstellungen für öffentlichen Zugriff sperren	`s3:PutAccountPublicAccessBlock`
DELETEKonto Einstellungen für öffentlichen Zugriff sperren	`s3:PutAccountPublicAccessBlock`
PUTZugangspunkt Einstellungen für öffentlic hen Zugriff blockieren	`s3:CreateAccessPoint`

> **Note**
>
> Für die DELETE Operationen sind dieselben Berechtigungen erforderlich wie für die PUT Operationen. Für die DELETE Operationen gibt es keine separaten Berechtigungen.

Verwenden von Block Public Access

Weitere Informationen zur Konfiguration von Block Public Access für Ihre AWS-Konto, Ihre Amazon S3 S3-Buckets und Ihre Access Points finden Sie in den folgenden Themen:

- Konfigurieren der Block-Public-Access-Einstellungen für Ihr Konto
- Konfigurieren von Block-Public-Access-Einstellungen für Ihre S3-Buckets
- Durchführen von Block Public Access-Vorgänge an einem Zugriffspunkt

Konfigurieren der Block-Public-Access-Einstellungen für Ihr Konto

Die Amazon-S3-Block-Public-Access-Funktion bietet Einstellungen für Zugriffspunkte, Buckets und Konten, mit denen Sie den öffentlichen Zugriff auf Amazon-S3-Ressourcen verwalten können. Standardmäßig erlauben neue Buckets, Zugriffspunkte und Objekte keinen öffentlichen Zugriff.

Weitere Informationen finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

> **Note**
>
> Einstellungen auf Kontoebene haben Vorrang vor Einstellungen für einzelne Objekte. Wenn Sie Ihr Konto so konfigurieren, dass der öffentliche Zugriff gesperrt wird, werden alle Einstellungen für den öffentlichen Zugriff, die für einzelne Objekte in Ihrem Konto vorgenommen wurden, außer Kraft gesetzt.

Sie können die S3-Konsole,, und verwenden AWS CLI AWS SDKs, REST API um die Einstellungen für den blockierten öffentlichen Zugriff für alle Buckets in Ihrem Konto zu konfigurieren. Weitere Informationen dazu finden Sie in den folgenden Abschnitten.

Informationen zum Konfigurieren von Block Public Access-Einstellungen für Ihre Buckets finden Sie unter Konfigurieren von Block-Public-Access-Einstellungen für Ihre S3-Buckets. Weitere Hinweise zu Zugriffspunkten finden Sie unter Durchführen von Block Public Access-Vorgänge an einem Zugriffspunkt.

Verwenden der S3-Konsole

Amazon S3 Block Public Access verhindert die Anwendung von Einstellungen, die den öffentlichen Zugriff auf Daten in S3-Buckets erlauben. In diesem Abschnitt wird erläutert, wie Sie die Block-Public-Access-Einstellungen für alle S3-Buckets in Ihrem AWS-Konto bearbeiten. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

Um die Einstellungen zum Blockieren des öffentlichen Zugriffs für alle S3-Buckets in einem zu bearbeiten AWS-Konto

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie Block Public Access settings for this account (Block Public Access-Einstellung für dieses Konto).
3. Wählen Sie Edit (Bearbeiten) aus, um die Block-Public-Access-Einstellungen für alle Buckets in Ihrem AWS-Konto zu ändern.
4. Wählen Sie die Einstellung aus, die Sie ändern möchten, und wählen Sie anschließend Save changes(Änderungen speichern) aus.
5. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **`confirm`** ein. Wählen Sie anschließend Confirm (Bestätigen) aus, um Ihre Änderungen zu speichern.

Verwenden Sie den AWS CLI

Sie können Amazon S3 Block Public Access über die AWS CLI verwenden. Weitere Informationen zur Einrichtung und Verwendung von finden Sie unter Was ist der AWS Command Line Interface? AWS CLI

Account

Um Block Public Access-Vorgänge auf einem Konto durchzuführen, verwenden Sie den AWS CLI - Service `s3control`. Folgende Vorgänge auf Konto-Ebene verwenden diesen Service:

- PUT PublicAccessBlock (für ein Konto)
- GET PublicAccessBlock (für ein Konto)
- DELETE PublicAccessBlock (für ein Konto)

Weitere Informationen und Beispiele finden Sie put-public-access-blockin der AWS CLI Referenz.

Mit dem AWS SDKs

Java

Die folgenden Beispiele zeigen Ihnen, wie Sie Amazon S3 Block Public Access mit der verwenden AWS SDK for Java , um eine Konfiguration für öffentlichen Zugriff auf ein Amazon S3 S3-Konto einzurichten.

```
AWSS3ControlClientBuilder controlClientBuilder =
 AWSS3ControlClientBuilder.standard();
controlClientBuilder.setRegion(<region>);
controlClientBuilder.setCredentials(<credentials>);

AWSS3Control client = controlClientBuilder.build();
client.putPublicAccessBlock(new PutPublicAccessBlockRequest()
  .withAccountId(<account-id>)
  .withPublicAccessBlockConfiguration(new PublicAccessBlockConfiguration()
    .withIgnorePublicAcls(<value>)
    .withBlockPublicAcls(<value>)
    .withBlockPublicPolicy(<value>)
    .withRestrictPublicBuckets(<value>)));
```

> **Important**
>
> Dieses Beispiel trifft nur für Vorgänge auf Konto-Ebene zu, die die `AWSS3Control`-Client-Klasse verwenden. Sehen Sie sich für Vorgänge auf Bucket-Ebene das vorhergehende Beispiel an.

Other SDKs

Informationen zur Verwendung des anderen AWS SDKs finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS SDKs in der Amazon S3 API S3-Referenz.

Mit dem REST API

Informationen zur Verwendung von Amazon S3 Block Public Access über die REST APIs finden Sie in den folgenden Themen in der Amazon Simple Storage Service API Reference.

- Vorgänge auf Konto-Ebene
  - PUT PublicAccessBlock
  - GET PublicAccessBlock
  - DELETE PublicAccessBlock

Konfigurieren von Block-Public-Access-Einstellungen für Ihre S3-Buckets

Die Amazon-S3-Block-Public-Access-Funktion bietet Einstellungen für Zugriffspunkte, Buckets und Konten, mit denen Sie den öffentlichen Zugriff auf Amazon-S3-Ressourcen verwalten können. Standardmäßig erlauben neue Buckets, Zugriffspunkte und Objekte keinen öffentlichen Zugriff.

Weitere Informationen finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

Sie können die S3-Konsole, AWS CLI, und verwenden AWS SDKs, REST API um öffentlichen Zugriff auf einen oder mehrere Buckets zu gewähren. Sie können den öffentlichen Zugriff auf Buckets, die bereits öffentlich sind, auch blockieren. Weitere Informationen dazu finden Sie in den folgenden Abschnitten.

Informationen zum Konfigurieren von Einstellungen zum Blockieren des öffentlichen Zugriffs für jeden Bucket in Ihrem Konto finden Sie unter Konfigurieren der Block-Public-Access-Einstellungen für Ihr

Konto. Informationen zum Konfigurieren von Block Public Access für Zugriffspunkte finden Sie unter Durchführen von Block Public Access-Vorgänge an einem Zugriffspunkt.

Verwenden der S3-Konsole

Amazon S3 Block Public Access verhindert die Anwendung von Einstellungen, die den öffentlichen Zugriff auf Daten in S3-Buckets erlauben. In diesem Abschnitt wird erläutert, wie Sie die Block-Public-Access-Einstellungen für einen oder mehrere S3-Buckets bearbeiten. Informationen zum Blockieren des öffentlichen Zugriffs mithilfe von AWS CLI AWS SDKs, und Amazon S3 REST APIs finden Sie unterBlockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

In der Liste der Buckets können Sie sehen, ob Ihr Bucket öffentlich zugänglich ist. In der Spalte Zugriff kennzeichnet Amazon S3 die Berechtigungen für einen Bucket wie folgt:

- Public (Öffentlich) – Alle Benutzer haben Zugriff auf eines oder mehrere der folgenden Elemente: Objekte auflisten, Objekte schreiben, Lese- und Schreibberechtigungen.
- Objects can be public (Objekte können öffentlich sein) – Der Bucket ist nicht öffentlich, aber jeder, der über die entsprechenden Berechtigungen verfügt, kann den öffentlichen Zugriff auf Objekte gewähren.
- Buckets and objects not public (Buckets und Objekte sind nicht öffentlich) – Für den Bucket und die Objekte gilt kein öffentlicher Zugriff.
- Nur autorisierte Benutzer dieses Kontos — Der Zugriff ist auf IAM Benutzer und Rollen in diesem Konto und auf AWS Dienstprinzipale beschränkt, da es eine Richtlinie gibt, die öffentlichen Zugriff gewährt.

Sie können Bucket-Suchen auch nach Zugriffstyp filtern. Wählen Sie aus der Dropdown-Liste neben der Leiste Search for buckets (Nach Buckets suchen) einen Zugriffstyp aus.

Wenn beim Auflisten Ihrer Buckets mit ihren Einstellungen für den öffentlichen Zugriff die Fehlermeldung `Error` angezeigt wird, verfügen Sie möglicherweise nicht über die erforderlichen Berechtigungen. Stellen Sie sicher, dass Sie Ihrer Benutzer- oder Rollenrichtlinie die folgenden Berechtigungen hinzugefügt haben:

```
s3:GetAccountPublicAccessBlock
s3:GetBucketPublicAccessBlock
s3:GetBucketPolicyStatus
s3:GetBucketLocation
s3:GetBucketAcl
s3:ListAccessPoints
```

```
s3:ListAllMyBuckets
```

In einigen seltenen Fällen können Anfragen auch aufgrund des Ausfalls einer AWS-Region fehlschlagen.

Bearbeiten der Amazon-S3-Block-Public-Access-Einstellungen für einen einzelnen S3-Bucket

Führen Sie die folgenden Schritte aus, wenn Sie die Einstellungen für den öffentlichen Zugriff für einen einzelnen S3-Bucket ändern müssen.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Bucket-Name den Namen des gewünschten Buckets aus.
3. Wählen Sie Permissions (Berechtigungen).
4. Wählen Sie Edit (Bearbeiten) aus, um die Einstellungen für den öffentlichen Zugriff für den Bucket zu ändern. Weitere Informationen zu den vier Amazon-S3-Block-Public-Access-Einstellungen finden Sie unter Block Public Access-Einstellungen.
5. Wählen Sie die Einstellung aus, die Sie ändern möchten, und wählen Sie anschließend Save (Speichern) aus.
6. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **`confirm`** ein. Wählen Sie anschließend Confirm (Bestätigen) aus, um Ihre Änderungen zu speichern.

Sie können beim Erstellen eines Buckets die Amazon S3 Block Public Access-Einstellungen ändern. Weitere Informationen finden Sie unter Erstellen eines Buckets.

Verwenden Sie den AWS CLI

Verwenden Sie den AWS CLI Dienst, um den öffentlichen Zugriff auf einen Bucket zu blockieren oder den öffentlichen Zugriff zu löschen`s3api`. Folgende Vorgänge auf Bucket-Ebene verwenden diesen Service:

- PUT PublicAccessBlock (für einen Bucket)
- GET PublicAccessBlock (für einen Eimer)
- DELETE PublicAccessBlock (für einen Eimer)
- GET BucketPolicyStatus

Weitere Informationen und Beispiele finden Sie put-public-access-blockin der AWS CLI Referenz.

Mit dem AWS SDKs

Java

```
AmazonS3 client = AmazonS3ClientBuilder.standard()
   .withCredentials(<credentials>)
   .build();

client.setPublicAccessBlock(new SetPublicAccessBlockRequest()
  .withBucketName(<bucket-name>)
  .withPublicAccessBlockConfiguration(new PublicAccessBlockConfiguration()
    .withBlockPublicAcls(<value>)
    .withIgnorePublicAcls(<value>)
    .withBlockPublicPolicy(<value>)
    .withRestrictPublicBuckets(<value>)));
```

Important

Dieses Beispiel trifft nur für Vorgänge auf Bucket-Ebene zu, die die `AmazonS3`-Client-Klasse verwenden. Sehen Sie sich für Vorgänge auf Konto-Ebene das folgende Beispiel an.

Other SDKs

Informationen zur Verwendung des anderen AWS SDKs finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS SDKs in der Amazon S3 API S3-Referenz.

Mit dem REST API

Informationen zur Verwendung von Amazon S3 Block Public Access über die REST APIs finden Sie in den folgenden Themen in der Amazon Simple Storage Service API Reference.

- Vorgänge auf Bucket-Ebene
  - PUT PublicAccessBlock
  - GET PublicAccessBlock
  - DELETE PublicAccessBlock

- GET BucketPolicyStatus

Überprüfung des Bucket-Zugriffs mit IAM Access Analyzer für S3

IAMAccess Analyzer for S3 warnt Sie vor S3-Buckets, die so konfiguriert sind, dass sie allen Personen im Internet oder auf anderen Websites AWS-Konten, auch AWS-Konten außerhalb Ihres Unternehmens, Zugriff gewähren. Für jeden öffentlichen oder freigegebenen Bucket erhalten Sie Ergebnisse bezüglich der Quelle und der Ebene des öffentlichen oder freigegebenen Zugriffs. IAMAccess Analyzer for S3 kann beispielsweise anzeigen, dass ein Bucket über Lese- oder Schreibzugriff verfügt, der über eine Bucket-Zugriffskontrollliste (ACL), eine Bucket-Richtlinie, eine Multi-Region-Zugriffspunkt-Richtlinie oder eine Zugriffspunktrichtlinie gewährt wird. Mit diesem Erkenntnissen können Sie sofortige und präzise Korrekturmaßnahmen ergreifen, um den Bucket-Zugriff wie beabsichtigt wiederherzustellen.

Bei der Überprüfung eines gefährdeten Buckets in IAM Access Analyzer for S3 können Sie den gesamten öffentlichen Zugriff auf den Bucket mit einem einzigen Klick blockieren. Wir empfehlen Ihnen, den gesamten Zugriff auf Ihre Buckets zu blockieren, es sei denn, Sie benötigen öffentlichen Zugriff, um einen bestimmten Anwendungsfall zu unterstützen. Bevor Sie den gesamten öffentlichen Zugriff blockieren, stellen Sie sicher, dass Ihre Anwendungen ohne öffentlichen Zugriff weiterhin ordnungsgemäß funktionieren. Weitere Informationen finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

Sie können auch einen Drilldown in die Berechtigungseinstellungen auf Bucket-Ebene ausführen, um detaillierte Zugriffsebenen zu konfigurieren. Für bestimmte und überprüfte Anwendungsfälle, die öffentlichen Zugriff erfordern, wie statisches Website-Hosting, öffentliche Downloads oder kontenübergreifende Freigabe, können Sie Ihre Absicht bestätigen und aufzeichnen, dass der Bucket öffentlich oder freigegeben bleibt, indem Sie die Ergebnisse für den Bucket archivieren. Diese Bucket-Konfigurationen sind jederzeit aufrufbar und änderbar. Sie können Ihre Ergebnisse zu Prüfungszwecken auch als CSV Bericht herunterladen.

IAMAccess Analyzer for S3 ist ohne zusätzliche Kosten auf der Amazon S3 S3-Konsole verfügbar. IAMAccess Analyzer für S3 wird von AWS Identity and Access Management (IAM) IAM Access Analyzer unterstützt. Um IAM Access Analyzer for S3 in der Amazon S3 S3-Konsole zu verwenden, müssen Sie die IAM Konsole aufrufen und IAM Access Analyzer für jede Region aktivieren.

Weitere Informationen zu IAM Access Analyzer finden Sie unter Was ist IAM Access Analyzer? im IAMBenutzerhandbuch. Weitere Informationen zu IAM Access Analyzer for S3 finden Sie in den folgenden Abschnitten.

Important

- IAMAccess Analyzer für S3 erfordert einen Analyzer auf Kontoebene. Um IAM Access Analyzer für S3 verwenden zu können, müssen Sie IAM Access Analyzer aufrufen und einen Analyzer erstellen, der über ein Konto als Vertrauenszone verfügt. Weitere Informationen finden Sie unter Enabling IAM Access Analyzer im IAMBenutzerhandbuch.
- IAMAccess Analyzer for S3 analysiert nicht die Access Point-Richtlinie, die kontoübergreifenden Access Points zugeordnet ist. Dieses Verhalten tritt auf, weil sich der Zugriffspunkt und seine Richtlinie außerhalb der Vertrauenszone, d. h. des Kontos, befinden. Buckets, die den Zugriff an einen kontoübergreifenden Zugriffspunkt delegieren, werden unter Buckets mit öffentlichem Zugriff aufgeführt, sofern Sie die Einstellung `RestrictPublicBuckets` zum Blockieren des öffentlichen Zugriffs nicht auf den Bucket oder das Konto angewendet haben. Wenn Sie die Einstellung „Öffentlichen Zugriff `RestrictPublicBuckets` blockieren“ anwenden, wird der Bucket unter Buckets mit Zugriff durch Dritte AWS-Konten — einschließlich Drittanbieter — aufgeführt. AWS-Konten
- Wenn eine Bucket-Richtlinie oder ein Bucket hinzugefügt oder geändert ACL wird, generiert und aktualisiert IAM Access Analyzer die Ergebnisse auf der Grundlage der Änderung innerhalb von 30 Minuten. Ergebnisse im Zusammenhang mit den Einstellungen zum Blockieren des öffentlichen Zugriffs auf Kontoebene werden möglicherweise erst bis zu 6 Stunden, nachdem Sie die Einstellungen geändert haben, generiert oder aktualisiert. Ergebnisse, die sich auf Multi-Regions-Zugriffspunkte beziehen, können bis zu sechs Stunden nach dem Erstellen, Löschen oder Ändern der Richtlinie des Multi-Region Access Points nicht generiert oder aktualisiert werden.

Topics

- Welche Informationen bietet IAM Access Analyzer for S3?
- IAMAccess Analyzer für S3 aktivieren
- Blockieren des gesamten öffentlichen Zugriffs
- Überprüfen und Ändern des Bucket-Zugriffs
- Archivieren von Bucket-Ergebnissen
- Aktivieren eines archivierten Bucket-Ergebnisses
- Anzeigen von Ergebnisdetails
- Ein IAM Access Analyzer for S3-Bericht wird heruntergeladen

Welche Informationen bietet IAM Access Analyzer for S3?

IAMAccess Analyzer for S3 bietet Ergebnisse für Buckets, auf die außerhalb Ihres AWS-Konto zugegriffen werden kann. Buckets, die unter Buckets with public access (Buckets mit öffentlichem Zugriff) aufgeführt sind, können jedem im Internet zugänglich. Wenn IAM Access Analyzer for S3 öffentliche Buckets identifiziert, wird oben auf der Seite auch eine Warnung angezeigt, die Ihnen die Anzahl der öffentlichen Buckets in Ihrer Region anzeigt. Buckets, die unter Buckets mit Zugriff von anderen AWS-Konten — einschließlich Drittanbietern — aufgeführt AWS-Konten sind, werden unter bestimmten Bedingungen mit anderen geteilt AWS-Konten, auch mit Konten außerhalb Ihrer Organisation.

IAMAccess Analyzer for S3 stellt für jeden Bucket die folgenden Informationen bereit:

- Bucket-Name
- Von Access Analyzer entdeckt - Als IAM Access Analyzer for S3 den öffentlichen oder gemeinsam genutzten Bucket-Zugriff entdeckte.
- Gemeinsam genutzt — wie der Bucket gemeinsam genutzt wird — über eine Bucket-Richtlinie, einen BucketACL, eine Multi-Region-Access-Point-Richtlinie oder eine Access-Point-Richtlinie. Multi-Region Access Points und kontoübergreifende Zugriffspunkte werden unter Zugriffspunkten aufgeführt. Ein Bucket kann sowohl über Richtlinien als auch gemeinsam genutzt werden. ACLs Wenn Sie die Quelle für den Bucket-Zugriff suchen und überprüfen möchten, können die Informationen in dieser Spalte als Ausgangspunkt für sofortige und präzise Korrekturmaßnahmen dienen.
- Status - Der Status der Erkenntnisse zu dem Bucket. IAMAccess Analyzer for S3 zeigt Ergebnisse für alle öffentlichen und gemeinsam genutzten Buckets an.
  - Active (Aktiv) – Ergebnis wurde nicht überprüft.
  - Archived (Archiviert) – Das Ergebnis wurde wie vorgesehen überprüft und bestätigt.
  - Alle - Alle Ergebnisse für Buckets, die öffentlich sind oder mit anderen geteilt wurden AWS-Konten, auch AWS-Konten außerhalb Ihrer Organisation.
- Access level (Zugriffsebene) – Gewährte Zugriffsberechtigungen für den Bucket:
  - List (Liste) – Auflistung der Ressourcen.
  - Read (Lesen) – Lesen aber kein Bearbeiten von Ressourceninhalten und -attributen.
  - Write (Schreiben) – Erstellen, Löschen oder Ändern von Ressourcen.
  - Permissions (Berechtigungen) – Erteilen oder ändern der Ressourcenberechtigungen.
  - Markierungen – Aktualisieren der einer Ressource zugeordneten Markierungen.

IAMAccess Analyzer für S3 aktivieren

Um IAM Access Analyzer für S3 verwenden zu können, müssen Sie die folgenden Voraussetzungen erfüllen.

1. Sie müssen die erforderlichen Berechtigungen erteilen.

   Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Für die Verwendung von IAM Access Analyzer erforderliche Berechtigungen.

2. Besuchen SieIAM, um für jede Region, in der Sie IAM Access Analyzer verwenden möchten, einen Analyzer auf Kontoebene zu erstellen.

   IAMAccess Analyzer für S3 erfordert einen Analyzer auf Kontoebene. Um IAM Access Analyzer for S3 verwenden zu können, müssen Sie einen Analyzer erstellen, der über ein Konto als Vertrauenszone verfügt. Weitere Informationen finden Sie unter Enabling IAM Access Analyzer im IAMBenutzerhandbuch.

Blockieren des gesamten öffentlichen Zugriffs

Wenn Sie den gesamten Zugriff auf einen Bucket mit einem einzigen Klick blockieren möchten, können Sie die Schaltfläche Allen öffentlichen Zugriff blockieren in IAM Access Analyzer für S3 verwenden. Wenn Sie den gesamten öffentlichen Zugriff auf einen Bucket blockieren, wird kein öffentlicher Zugriff gewährt. Wir empfehlen Ihnen, den gesamten öffentlichen Zugriff auf Ihre Buckets zu blockieren, es sei denn, Sie benötigen öffentlichen Zugriff, um einen bestimmten und verifizierten Anwendungsfall zu unterstützen. Bevor Sie den gesamten öffentlichen Zugriff blockieren, stellen Sie sicher, dass Ihre Anwendungen ohne öffentlichen Zugriff weiterhin ordnungsgemäß funktionieren.

Wenn Sie nicht den gesamten öffentlichen Zugriff auf Ihren Bucket blockieren möchten, können Sie Ihre Block Public Access-Einstellungen in der Amazon-S3-Konsole bearbeiten, um detaillierte Zugriffsebenen für Ihre Buckets zu konfigurieren. Weitere Informationen finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

In seltenen Fällen meldet IAM Access Analyzer for S3 möglicherweise keine Ergebnisse für einen Bucket, den eine Amazon S3 S3-Block-Bewertung für öffentlichen Zugriff als öffentlich meldet. Dies ist der Fall, da Amazon S3 Block Public Access Richtlinien für aktuelle Aktionen und potenzielle Aktionen, die in Zukunft hinzugefügt werden könnten und die dazu führen könnten, dass ein Bucket öffentlich wird, überprüft. Andererseits analysiert IAM Access Analyzer for S3 nur die aktuellen

Aktionen, die für den Amazon S3 S3-Service bei der Bewertung des Zugriffsstatus angegeben wurden.

Um den gesamten öffentlichen Zugriff auf einen Bucket mithilfe von IAM Access Analyzer for S3 zu blockieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im Navigationsbereich auf der linken Seite unter Dashboards die Option Access Analyzer for S3 aus.
3. Wählen IAM Sie in Access Analyzer for S3 einen Bucket aus.
4. Wählen Sie Block all public access (Öffentlichen Zugriff blockieren) aus.
5. Um zu bestätigen, dass Sie den gesamten öffentlichen Zugriff auf den Bucket blockieren möchten, geben Sie unter Block all public access (bucket settings) (Öffentlichen Zugriff blockieren) (Bucket-Einstellungen) **`confirm`** ein.

   Amazon S3 blockiert den gesamten öffentlichen Zugriff auf Ihren Bucket. Der Status des Buckets, bei dem Aktualisierungen gefunden wurden, lautet „Gelöst“, und der Bucket verschwindet aus der Liste von IAM Access Analyzer für S3. Wenn Sie die gelösten Buckets überprüfen möchten, öffnen Sie IAM Access Analyzer in der IAMKonsole.

Überprüfen und Ändern des Bucket-Zugriffs

Wenn Sie nicht beabsichtigen, der Öffentlichkeit oder anderen Personen AWS-Konten, einschließlich Konten außerhalb Ihrer Organisation, Zugriff zu gewähren, können Sie den Bucket, die Bucket-RichtlinieACL, die Richtlinie für multiregionale Zugriffspunkte oder die Zugriffspunktrichtlinie ändern, um den Zugriff auf den Bucket zu entfernen. In der Spalte „Geteilt durch“ werden alle Quellen für den Bucket-Zugriff angezeigt: Bucket-RichtlinieACL, Bucket- und/oder Zugriffspunkt-Richtlinie. Multi-Region Access Points und kontoübergreifende Zugriffspunkte werden unter Zugriffspunkten aufgeführt.

Um eine Bucket-Richtlinie, einen Bucket, einen Access Point für mehrere Regionen oder eine Zugriffspunkt-Richtlinie zu überprüfen und zu ändern ACL

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im Navigationsbereich Access Analyzer for S3.

3. Um zu sehen, ob öffentlicher Zugriff oder gemeinsamer Zugriff über eine Bucket-Richtlinie, einen BucketACL, eine Multi-Region-Access-Point-Richtlinie oder eine Access-Point-Richtlinie gewährt wird, schauen Sie in der Spalte Shared through nach.
4. Wählen Sie unter Buckets den Namen des Buckets mit der Bucket-Richtlinie, dem BucketACL, der Multi-Region-Zugriffspunkt-Richtlinie oder der Zugriffspunktrichtlinie aus, die Sie ändern oder überprüfen möchten.
5. Wenn Sie einen Bucket ändern oder anzeigen möchten: ACL

   a. Wählen Sie Permissions (Berechtigungen).

   b. Wählen Sie Access Control List.

   c. Überprüfen Sie Ihren Bucket ACL und nehmen Sie bei Bedarf Änderungen vor.

      Weitere Informationen finden Sie unter Konfiguration ACLs.
6. Wenn Sie eine Bucket-Richtlinie ändern oder überprüfen möchten, gehen Sie folgendermaßen vor:

   a. Wählen Sie Permissions (Berechtigungen).

   b. Wählen Sie Bucket Policy aus.

   c. Überprüfen oder ändern Sie Ihre Bucket-Richtlinie nach Bedarf.

      Weitere Informationen finden Sie unter Hinzufügen einer Bucket-Richtlinie mit der Amazon-S3-Konsole.
7. Wenn Sie eine Multi-Regions-Zugriffspunkt-Richtlinie ändern oder anzeigen, gehen Sie folgendermaßen vor:

   a. Wählen Sie Multiregionaler Zugriffspunkt.

   b. Wählen Sie den multiregionalen Zugriffspunktsnamen.

   c. Überprüfen oder ändern Sie die Multi-Regions-Zugriffspunkt-Richtlinie nach Bedarf.

      Weitere Informationen finden Sie unter Berechtigungen.
8. Wenn Sie eine Zugriffspunkt-Richtlinie überprüfen oder ändern möchten, gehen Sie folgendermaßen vor:

   a. Wählen Sie Access Points (Zugriffspunkte).

   b. Wählen Sie den Namen des Zugriffspunkts aus.

   c. Überprüfen oder ändern Sie den Zugriff nach Bedarf.

Weitere Informationen finden Sie unter Nutzen von Amazon S3-Zugriffspunkten mit der Amazon S3-Konsole.

Wenn Sie einen BucketACL, eine Bucket-Richtlinie oder eine Zugriffspunkt-Richtlinie bearbeiten oder entfernen, um öffentlichen oder gemeinsamen Zugriff zu entfernen, wird der Status der Bucket-Ergebnisse auf „Gelöst“ aktualisiert. Die aufgelösten Bucket-Ergebnisse werden nicht mehr in der Liste von IAM Access Analyzer for S3 angezeigt, Sie können sie jedoch in IAM Access Analyzer anzeigen.

Archivieren von Bucket-Ergebnissen

Wenn ein Bucket der Öffentlichkeit oder anderen AWS-Konten, einschließlich Konten außerhalb Ihrer Organisation, Zugriff gewährt, um einen bestimmten Anwendungsfall zu unterstützen (z. B. eine statische Website, öffentliche Downloads oder kontoübergreifendes Teilen), können Sie die Ergebnisse für den Bucket archivieren. Wenn Sie Bucket-Ergebnisse archivieren, bestätigen und verzeichnen Sie Ihre Absicht, dass der Bucket öffentlich oder freigegeben bleiben soll. Archivierte Bucket-Ergebnisse verbleiben in Ihrer IAM Access Analyzer for S3-Liste, sodass Sie immer wissen, welche Buckets öffentlich oder gemeinsam genutzt werden.

Um Bucket-Ergebnisse in IAM Access Analyzer für S3 zu archivieren

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im Navigationsbereich Access Analyzer for S3.
3. Wählen Sie in IAM Access Analyzer for S3 einen aktiven Bucket aus.
4. Um zu bestätigen, dass Sie beabsichtigen, dass die Öffentlichkeit oder andere Personen AWS-Konten, einschließlich Konten außerhalb Ihrer Organisation, auf diesen Bucket zugreifen, wählen Sie Archivieren.
5. Geben Sie **`confirm`** ein und wählen Sie Archiv (Archivieren).

Aktivieren eines archivierten Bucket-Ergebnisses

Nachdem Sie Ergebnisse archiviert haben, können Sie sie jederzeit erneut einsehen und ihren Status wieder auf aktiv ändern, wodurch angegeben wird, dass für den Bucket eine weitere Überprüfung erforderlich ist.

Um die Suche nach einem archivierten Bucket in IAM Access Analyzer for S3 zu aktivieren

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im Navigationsbereich Access Analyzer for S3.
3. Wählen Sie die archivierten Bucket-Ergebnisse aus.
4. Wählen Sie Mark as active (Als aktiv markieren) aus.

Anzeigen von Ergebnisdetails

Wenn Sie weitere Informationen zu einem Bucket benötigen, können Sie den Bucket öffnen und Details in IAM Access Analyzer auf der IAMKonsole suchen.

Um Details zu den Ergebnissen in IAM Access Analyzer für S3 anzuzeigen

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im Navigationsbereich Access Analyzer for S3.
3. Wählen IAM Sie in Access Analyzer for S3 einen Bucket aus.
4. Wählen Sie die Option View details aus.

   Die Ergebnisdetails werden in IAM Access Analyzer auf der IAMKonsole geöffnet.

Ein IAM Access Analyzer for S3-Bericht wird heruntergeladen

Sie können Ihre Bucket-Ergebnisse als CSV Bericht herunterladen, den Sie für Prüfungszwecke verwenden können. Der Bericht enthält dieselben Informationen, die Sie in IAM Access Analyzer for S3 auf der Amazon S3 S3-Konsole sehen.

So laden Sie einen Bericht herunter

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im Navigationsbereich auf der linken Seite Access Analyzer for S3 aus.
3. Wählen Sie im Filter „Region“ die Region aus.

   IAMAccess Analyzer for S3-Updates zeigt Buckets für die gewählte Region an.
4. Wählen Sie Download Report (Bericht herunterladen) aus.

   Ein CSV Bericht wird generiert und auf Ihrem Computer gespeichert.

Überprüfen der Bucket-Eigentümerschaft mit Bucket-Eigentümer-Bedingung

Die Bedingung des Amazon S3 S3-Bucket-Besitzers stellt sicher, dass die Buckets, die Sie in Ihren S3-Vorgängen verwenden AWS-Konten , zu den von Ihnen erwarteten gehören.

Die meisten S3-Vorgänge lesen oder schreiben in bestimmte S3-Buckets. Zu diesen Vorgängen gehören das Hochladen, Kopieren und Herunterladen von Objekten, das Abrufen oder Ändern von Bucket-Konfigurationen und das Abrufen oder Ändern von Objektkonfigurationen. Wenn Sie diese Vorgänge ausführen, geben Sie den Bucket an, den Sie verwenden möchten, indem Sie seinen Namen in die Anforderung einbeziehen. Um beispielsweise ein Objekt aus S3 abzurufen, stellen Sie eine Anforderung, die den Namen eines Buckets und den Objektschlüssel angibt, der aus diesem Bucket abgerufen werden soll.

Da Amazon S3 Buckets anhand ihrer Namen identifiziert, könnte eine Anwendung, die einen falschen Bucket-Namen in einer Anforderung verwendet, versehentlich Vorgänge gegen einen anderen Bucket als erwartet ausführen. Um unbeabsichtigte Bucket-Interaktionen in solchen Situationen zu vermeiden, können Sie die Bucket-Eigentümer-Bedingung verwenden. Die Bucket-Eigentümer-Bedingung ermöglicht es Ihnen, zu überprüfen, ob der Ziel-Bucket im Besitz des erwarteten AWS-Konto ist, und bietet eine zusätzliche Gewissheit dafür, dass Ihre S3-Vorgänge die von Ihnen beabsichtigten Auswirkungen haben.

Themen

- Wann die Bucket-Eigentümer-Bedingung verwendet werden sollte
- Verifizieren eines Bucket-Eigentümers
- Beispiele
- Beschränkungen und Einschränkungen

Wann die Bucket-Eigentümer-Bedingung verwendet werden sollte

Wir empfehlen, die Bucket-Eigentümer-Bedingung zu verwenden, wenn Sie einen unterstützten S3-Vorgang ausführen und die Konto-ID des erwarteten Bucket-Eigentümers kennen. Die Bucket-Eigentümer-Bedingung ist für alle S3-Objekt-Vorgänge und die meisten S3-Bucketvorgänge verfügbar. Eine Liste der S3-Vorgänge, die die Bucket-Eigentümerbedingung nicht unterstützen, finden Sie unter Beschränkungen und Einschränkungen.

Stellen Sie sich das folgende Szenario mit AWS Kunde Bea vor, um die Vorteile der Verwendung der Bedingung „Bucket Owner“ zu erkennen:

1. Bea entwickelt eine Anwendung, die Amazon S3 verwendet. Während der Entwicklung verwendet Bea ihre reine Testfunktion, AWS-Konto um einen Bucket mit dem Namen zu erstellen`bea-data-test`, und konfiguriert ihre Anwendung so, dass sie Anfragen an sie stellt. `bea-data-test`
2. Bea stellt ihre Anwendung bereit, vergisst jedoch, die Anwendung neu zu konfigurieren, um einen Bucket in ihrem Produktions- AWS-Konto zu verwenden.
3. In der Produktion stellt Beas Anwendung Anforderungen an `bea-data-test`, die erfolgreich sind. Dies führt dazu, dass Produktionsdaten in den Bucket im Testkonto von Bea geschrieben werden.

Bea kann sich vor solchen Situationen schützen, indem sie die Bucket-Eigentümer-Bedingung verwendet. Mit der Bedingung „Bucket-Besitzer“ kann Bea die AWS-Konto ID des erwarteten Bucket-Besitzers in ihre Anfragen aufnehmen. Amazon S3 überprüft dann die Konto-ID des Bucket-Eigentümers, bevor es eine Anforderung bearbeitet. Wenn der tatsächliche Bucket-Eigentümer nicht mit dem erwarteten Bucket-Eigentümer übereinstimmt, schlägt die Anforderung fehl.

Wenn Bea die Bucket-Eigentümer-Bedingung verwendet, führt das zuvor beschriebene Szenario nicht dazu, dass die Anwendung von Bea versehentlich in einen Test-Bucket schreibt. Stattdessen schlagen die Anforderungen, die ihre Anwendung in Schritt 3 stellt, mit einer Fehlermeldung `Access Denied` fehl. Durch die Verwendung der Bucket-Eigentümer-Bedingung trägt Bea dazu bei, das Risiko einer versehentlichen Interaktion mit Buckets im falschen AWS-Konto zu eliminieren.

Verifizieren eines Bucket-Eigentümers

Um die Bucket-Eigentümer-Bedingung zu verwenden, schließen Sie in ihre Anforderung einen Parameter ein, der den erwarteten Bucket-Eigentümer angibt. Die meisten S3-Vorgänge betreffen nur einen einzelnen Bucket und erfordern nur diesen einzelnen Parameter, um die Bucket-Eigentümer-Bedingung zu verwenden. Für `CopyObject`-Vorgänge gibt dieser erste Parameter den erwarteten Eigentümer des Ziel-Buckets an, und Sie schließen einen zweiten Parameter ein, um den erwarteten Eigentümer des Quell-Buckets anzugeben.

Wenn Sie eine Anforderung stellen, die einen Bucket-Eigentümer-Bedingungsparameter enthält, überprüft S3 vor der Verarbeitung der Anforderung die Konto-ID des Bucket-Eigentümers mit dem angegebenen Parameter. Wenn der Parameter mit der Konto-ID des Bucket-Eigentümers übereinstimmt, verarbeitet S3 die Anforderung. Wenn der Parameter nicht mit der Konto-ID des Bucket-Eigentümers übereinstimmt, schlägt die Anforderung mit einer Fehlermeldung `Access Denied` fehl.

Sie können die Bucket-Owner-Bedingung mit AWS Command Line Interface (AWS CLI) AWS SDKs, und Amazon S3 verwenden RESTAPIs. Wenn Sie die Bucket Owner-Bedingung mit Amazon S3 AWS CLI und Amazon S3 verwenden RESTAPIs, verwenden Sie die folgenden Parameternamen.

Zugriffsmethode	Parameter für Nicht-Kopiervorgänge	Quellparameter für Kopiervorgang	Zielparameter für Kopiervorgang
AWS CLI	`--expected-bucket-owner`	`--expected-source-bucket-owner`	`--expected-bucket-owner`
Amazon S3 REST APIs	Header `x-amz-expected-bucket-owner`	Header `x-amz-source-expected-bucket-owner`	Header `x-amz-expected-bucket-owner`

Die Parameternamen, die für die Verwendung der Bucket-Eigentümer-Bedingung erforderlich sind, AWS SDKs variieren je nach Sprache. Informationen zur Bestimmung der erforderlichen Parameter finden Sie in der SDK Dokumentation für die gewünschte Sprache. Sie finden die SDK Dokumentation unter Tools to Build on AWS.

Beispiele

Die folgenden Beispiele zeigen, wie Sie die Bucket-Eigentümerbedingung in Amazon S3 mithilfe von AWS CLI oder der implementieren können AWS SDK for Java 2.x.

Example

Beispiel: Ein Objekt hochladen

Im folgenden Beispiel wird ein Objekt unter Verwendung der Bucket-Eigentümer-Bedingung in S3-Bucket *amzn-s3-demo-bucket1* hochgeladen, um sicherzustellen, dass sich *amzn-s3-demo-bucket1* im Besitz des AWS-Konto 111122223333 befindet

AWS CLI

```
aws s3api put-object \
                --bucket amzn-s3-demo-bucket1 --key exampleobject --
body example_file.txt \
                --expected-bucket-owner 111122223333
```

AWS SDK for Java 2.x

```
public void putObjectExample() {
    S3Client s3Client = S3Client.create();;
    PutObjectRequest request = PutObjectRequest.builder()
            .bucket("amzn-s3-demo-bucket1")
            .key("exampleobject")
            .expectedBucketOwner("111122223333")
            .build();
    Path path = Paths.get("example_file.txt");
    s3Client.putObject(request, path);
}
```

Example

Beispiel: Kopieren eines Objekts

Im folgenden Beispiel wird das Objekt `object1` vom S3-Bucket *amzn-s3-demo-bucket1* in den S3-Bucket *amzn-s3-demo-bucket2* kopiert. Dabei wird die Bucket-Eigentümer-Bedingung verwendet, um sicherzustellen, dass die Buckets den erwarteten Konten gemäß der folgenden Tabelle gehören.

Bucket	Erwarteter Eigentümer
amzn-s3-demo-bucket1	111122223333
amzn-s3-demo-bucket2	444455556666

AWS CLI

```
aws s3api copy-object --copy-source amzn-s3-demo-bucket1/object1 \
                          --bucket amzn-s3-demo-bucket2 --key object1copy \
                          --expected-source-bucket-owner 111122223333 --expected-
bucket-owner 444455556666
```

AWS SDK for Java 2.x

```
public void copyObjectExample() {
```

```
        S3Client s3Client = S3Client.create();
        CopyObjectRequest request = CopyObjectRequest.builder()
                .copySource("amzn-s3-demo-bucket1/object1")
                .destinationBucket("amzn-s3-demo-bucket2")
                .destinationKey("object1copy")
                .expectedSourceBucketOwner("111122223333")
                .expectedBucketOwner("444455556666")
                .build();
        s3Client.copyObject(request);
    }
```

Example

Beispiel: Abrufen einer Bucket-Richtlinie

Im folgenden Beispiel wird die Zugriffsrichtlinie für S3-Bucket *amzn-s3-demo-bucket1* abgerufen, wobei die Bucket-Eigentümerbedingung verwendet wird, um sicherzustellen, dass *amzn-s3-demo-bucket1* dem AWS-Konto 111122223333 gehört.

AWS CLI

```
aws s3api get-bucket-policy --bucket amzn-s3-demo-bucket1 --expected-bucket-
owner 111122223333
```

AWS SDK for Java 2.x

```
public void getBucketPolicyExample() {
    S3Client s3Client = S3Client.create();
    GetBucketPolicyRequest request = GetBucketPolicyRequest.builder()
            .bucket("amzn-s3-demo-bucket1")
            .expectedBucketOwner("111122223333")
            .build();
    try {
        GetBucketPolicyResponse response = s3Client.getBucketPolicy(request);
    }
    catch (S3Exception e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    }
}
```

Beschränkungen und Einschränkungen

Die Amazon-S3-Bucket-Eigentümer-Bedingung hat die folgenden Einschränkungen und Grenzen:

- Der Wert des Bedingungsparameters Bucket Owner muss eine AWS-Konto ID (12-stelliger numerischer Wert) sein. Service-Prinzipale werden nicht unterstützt.
- Die Bedingung „Bucket-Besitzer“ ist weder für CreateBucketnoch für andere der in AWS S3 Control enthaltenen Operationen verfügbar. ListBuckets Amazon S3 ignoriert alle Bucket-Eigentümer-Bedingungsparameter, die in Anfragen für diese Vorgänge enthalten sind
- Die Bucket-Eigentümer-Bedingung überprüft nur, dass das im Verifizierungsparameter angegebene Konto den Bucket besitzt. Die Bucket-Eigentümer-Bedingung überprüft nicht die Konfiguration des Buckets. Sie garantiert auch nicht, dass die Konfiguration des Buckets bestimmte Bedingungen erfüllt oder mit einem früheren Zustand übereinstimmt.

Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie den Besitz von Objekten kontrollieren können, die in Ihren Bucket hochgeladen wurden, und um Zugriffskontrolllisten zu deaktivieren oder zu aktivieren () ACLs. Standardmäßig ist für Object Ownership die erzwungene Einstellung Bucket Owner festgelegt, und alle sind ACLs deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff auf Daten ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.

In den meisten modernen Anwendungsfällen in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs, und wir empfehlen, diese ACLs Option weiterhin zu deaktivieren, außer in ungewöhnlichen Fällen, in denen Sie den Zugriff für jedes Objekt einzeln kontrollieren müssen. Wenn diese ACLs Option deaktiviert ist, können Sie Richtlinien verwenden, um den Zugriff auf jedes Objekt in Ihrem Bucket einfacher zu kontrollieren, unabhängig davon, wer die Objekte in Ihren Bucket hochgeladen hat.

Object Ownership verfügt über drei Einstellungen, mit denen Sie die Inhaberschaft der in Ihren Bucket hochgeladenen Objekte kontrollieren und deaktivieren oder aktivieren könnenACLs:

ACLsdeaktiviert

- Bucket-Besitzer erzwungen (Standard) — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLswirkt sich nicht mehr auf die Berechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet Richtlinien, um die Zugriffssteuerung zu definieren.

ACLsaktiviert

- Bevorzugter Bucket-Besitzer — Der Bucket-Besitzer besitzt und hat die volle Kontrolle über neue Objekte, die andere Accounts in den Bucket mit den gespeicherten Objekten `bucket-owner-full-control` schreibenACL.
- Objekt-Writer — AWS-Konto Derjenige, der ein Objekt hochlädt, besitzt das Objekt, hat die volle Kontrolle darüber und kann anderen Benutzern Zugriff darauf gewähren. ACLs

Für die meisten modernen Anwendungsfälle in S3 empfehlen wir, die ACLs Deaktivierung beizubehalten, indem Sie die Einstellung Bucket Owner erforced anwenden und Ihre Bucket-Richtlinie verwenden, um Daten nach Bedarf mit Benutzern außerhalb Ihres Kontos zu teilen. Dieser Ansatz vereinfacht die Berechtigungsverwaltung. Sie können die Option sowohl für neu erstellte als auch für bereits bestehende Buckets deaktivierenACLs. Bei neu erstellten Buckets ACLs sind diese standardmäßig deaktiviert. Bei einem vorhandenen Bucket, der bereits Objekte enthält, ACLs sind das Objekt und der Bucket nach der Deaktivierung ACLs nicht mehr Teil einer Zugriffsevaluierung, und der Zugriff wird auf der Grundlage von Richtlinien gewährt oder verweigert. Bestehende Buckets können Sie ACLs jederzeit wieder aktivieren, nachdem Sie sie deaktiviert haben. Ihr bereits vorhandener Bucket und Ihr Objekt ACLs werden wiederhergestellt.

Wir empfehlen IhnenACLs, vor der Deaktivierung Ihre Bucket-Richtlinie zu überprüfen, um sicherzustellen, dass sie alle Möglichkeiten abdeckt, mit denen Sie außerhalb Ihres Kontos Zugriff auf Ihren Bucket gewähren möchten. Nach der Deaktivierung ACLs akzeptiert Ihr Bucket nur `PUT` Anfragen, die kein ACL oder `PUT` Anfragen angeben, bei denen der Bucket-Besitzer die volle Kontrolle hatACLs, z. B. `bucket-owner-full-control` vorgefertigte ACL oder gleichwertige Formen davon, ACL ausgedrückt inXML. Bestehende Anwendungen, die die Vollzugriff ACLs des Bucket-Besitzers unterstützen, haben keine Auswirkungen. `PUT`Anfragen, die andere ACLs (z. B. benutzerdefinierte Zuschüsse für bestimmte AWS-Konten) enthalten, schlagen fehl und geben einen `400` Fehler mit dem Fehlercode zurück`AccessControlListNotSupported`.

Im Gegensatz dazu akzeptiert und berücksichtigt ein Bucket mit der bevorzugten Einstellung des Bucket-Besitzers weiterhin Bucket und ObjektACLs. Mit dieser Einstellung gehören neue Objekte, die mit der `bucket-owner-full-control` gespeicherten Option geschrieben wurden, automatisch dem Bucket-Besitzer und nicht dem Object-Writer. ACL Alle anderen ACL Verhaltensweisen bleiben bestehen. Um zu verlangen, dass alle Amazon S3 PUT S3-Operationen die gespeicherten `bucket-owner-full-control` Daten enthaltenACL, können Sie eine Bucket-Richtlinie hinzufügen, die nur Objekt-Uploads mit dieser ACL Methode zulässt.

Wenn Sie feststellen möchten, welche Einstellungen für die Objekteigentümerschaft auf Ihre Buckets angewendet werden, können Sie die Metriken von Amazon S3 Storage Lens verwenden. S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Weitere Informationen finden Sie unter Verwenden von S3 Storage Lens, um Einstellungen für die Objekteigentümerschaft zu finden.

> **Note**
>
> Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter S3 Express One Zone und Arbeiten mit Verzeichnis-Buckets.

Einstellungen für Object Ownership

Diese Tabelle zeigt die Auswirkungen, die jede Einstellung „Objekteigentum“ auf ObjekteACLs, Objekteigentum und Objekt-Uploads hat.

Einstellung	Gilt für	Auswirkung auf Object Ownership	Auswirkung auf ACLs	Hochladen akzeptiert
„Bucket-Eigentümer erzwungen“ (Standard)	Alle neuen und bestehenden Objekte	Bucket-Eigentümer besitzt jedes Objekt.	ACLssind deaktiviert und wirken sich nicht mehr auf die Zugriffsberechtigungen für Ihren Bucket aus. Anfragen	Uploads, bei denen der Bucket-Besitzer die volle Kontrolle hat, ACLs oder Uploads, bei denen kein ACL

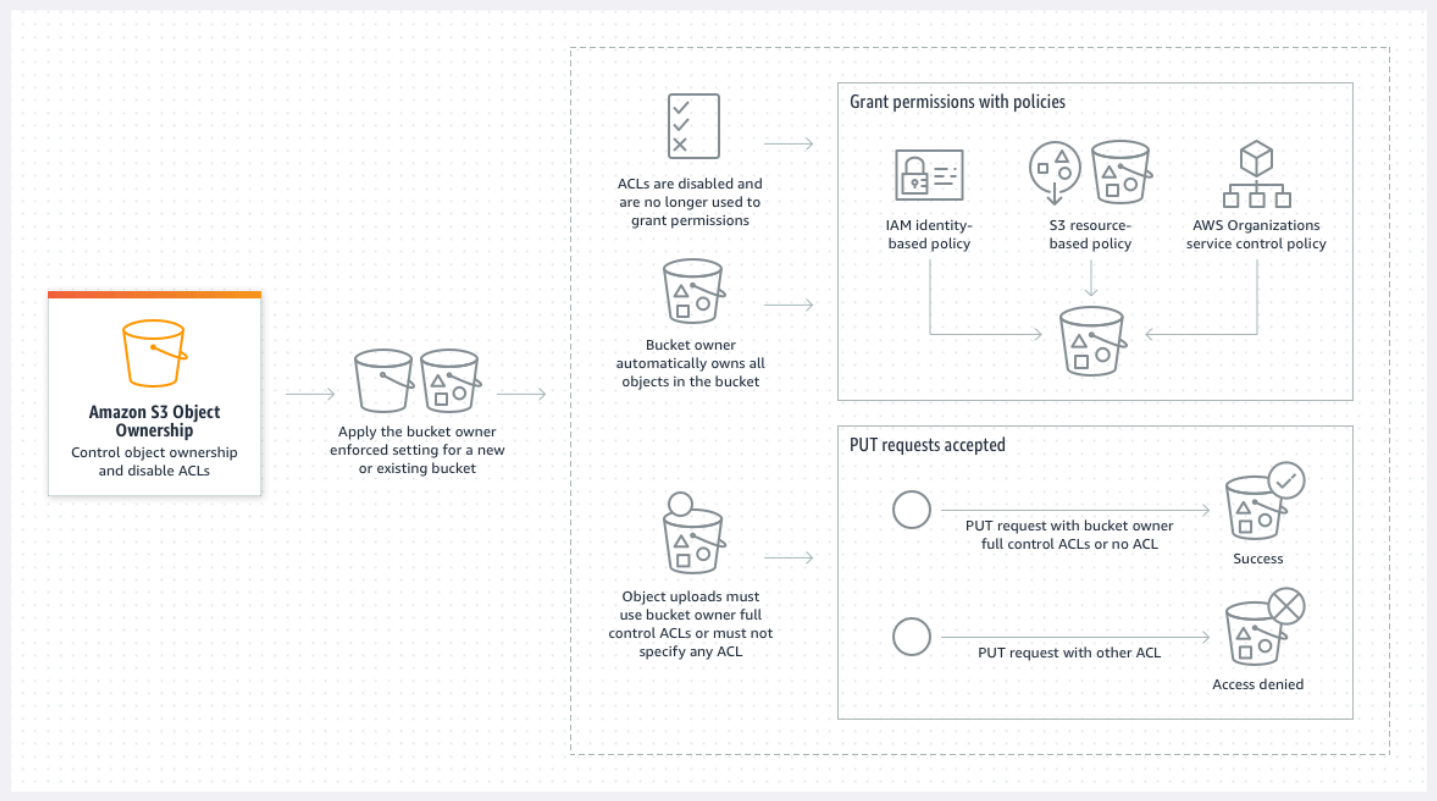
Einstellung	Gilt für	Auswirkung auf Object Ownership	Auswirkung auf ACLs	Hochladen akzeptiert
			zum Einrichten oder Aktualisieren ACLs schlagen fehl. Leseanfragen ACLs werden jedoch unterstützt. Bucket-Eigentümer hat das volle Eigentum und die volle Kontrolle. Der Objekt-Writer hat nicht mehr das volle Eigentum und die volle Kontrolle.	

Einstellung	Gilt für	Auswirkung auf Object Ownership	Auswirkung auf ACLs	Hochladen akzeptiert
Bucket-Eigentümer bevorzugt	Neue Objekte	Wenn bei einem Objekt-Upload auch das gespeicherte `bucket-owner-full-control` Objekt enthalten istACL, gehört das Objekt dem Bucket-Besitzer. Objekte, die zusammen mit anderen hochgeladen wurden, ACLs gehören dem Schreibkonto.	ACLskann aktualisiert werden und kann Berechtigungen gewähren. Wenn ein Objekt-Upload die `bucket-owner-full-control` gespeicherten Objekte enthältACL, hat der Bucket-Besitzer Vollzugriff, und der Objekt-Writer hat keinen Vollzugriff mehr.	Alle Uploads
Objektschreiber	Neue Objekte	Der Objekt-Writer besitzt das Objekt.	ACLskann aktualisiert werden und kann Berechtigungen gewähren. Der Objekt-Writer hat vollen Kontrollzugriff.	Alle Uploads

Änderungen, die durch Deaktivierung eingeführt wurden ACLs

Wenn die vom Bucket-Besitzer erzwungene Einstellung für Objekteigentum angewendet wird, ACLs sind sie deaktiviert und Sie besitzen automatisch jedes Objekt im Bucket und übernehmen die volle Kontrolle über jedes Objekt im Bucket, ohne weitere Maßnahmen ergreifen zu müssen. „Bucket-Eigentümer erzwungen“ ist die Standardeinstellung für alle neu erstellten Buckets. Nachdem die Einstellung „Bucket-Eigentümer erzwungen“ angewendet wurde, sehen Sie drei Änderungen:

- Alle Buckets ACLs und Objekte ACLs sind deaktiviert, sodass Sie als Bucket-Besitzer vollen Zugriff haben. Wenn Sie eine ACL Leseanforderung für Ihren Bucket oder Ihr Objekt ausführen, werden Sie feststellen, dass nur der Bucket-Besitzer vollen Zugriff hat.
- Sie als Bucket-Eigentümer besitzen automatisch jedes Objekt in Ihrem Bucket und haben die volle Kontrolle darüber.
- ACLswirkt sich nicht mehr auf die Zugriffsberechtigungen für Ihren Bucket aus. Daher basiert die Zugriffskontrolle für Ihre Daten auf Richtlinien wie AWS Identity and Access Management (IAM) identitätsbasierten Richtlinien, Amazon S3 S3-Bucket-Richtlinien, VPC Endpunktrichtlinien und Servicesteuerungsrichtlinien von Organizations (SCPs) oder Ressourcenkontrollrichtlinien (RCPs).



Wenn Sie S3-Versioning verwenden, besitzt der Bucket-Eigentümer alle Objektversionen in Ihrem Bucket und hat die volle Kontrolle über sie. Durch das Anwenden der Einstellung „Bucket-Eigentümer erzwungen“ wird keine neue Version eines Objekts hinzugefügt.

Neue Objekte können nur dann in Ihren Bucket hochgeladen werden, wenn für sie die Vollzugriff des Bucket-Besitzers verwendet wird ACLs oder wenn kein Objekt angegeben istACL. Objekt-Uploads schlagen fehl, wenn sie ein anderes ACL angeben. Weitere Informationen finden Sie unter Fehlerbehebung.

Da die folgende `PutObject` Beispieloperation mit der AWS Command Line Interface (AWS CLI) die Funktion „Gespeichert“ `bucket-owner-full-control` beinhaltetACL, kann das Objekt in einen Bucket mit deaktivierter Option ACLs hochgeladen werden.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key key-name --body path-to-file --
acl bucket-owner-full-control
```

Da die folgende `PutObject` Operation keine spezifiziertACL, ist sie auch für einen Bucket mit deaktivierter Option ACLs erfolgreich.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key key-name --body path-to-file
```

> **Note**
>
> Wenn andere Benutzer nach dem Hochladen Zugriff auf Objekte AWS-Konten benötigen, müssen Sie diesen Konten über Bucket-Richtlinien zusätzliche Berechtigungen gewähren. Weitere Informationen finden Sie unter Exemplarische Vorgehensweisen, die Richtlinien verwenden, um den Zugriff auf Ihre Amazon S3 S3-Ressourcen zu verwalten.

Erneut aktivieren ACLs

Sie können die Aktivierung jederzeit wieder aktivieren, ACLs indem Sie von der Einstellung „Bucket-Besitzer erzwungen“ zu einer anderen Einstellung „Objektbesitz“ wechseln. Wenn Sie ein Objekt ACLs für die Rechteverwaltung verwendet haben, bevor Sie die Einstellung „Bucket-Besitzer erzwungen“ angewendet haben, und Sie diese ACL Objektberechtigungen nicht in Ihre Bucket-Richtlinie migriert haben, werden diese Berechtigungen nach der erneuten Aktivierung wiederhergestelltACLs. Darüber hinaus gehören Objekte, die in den Bucket geschrieben wurden, während die „Einstellung „Bucket-Eigentümer erzwungen“ angewendet wurde, weiterhin dem Bucket-Eigentümer.

Wenn Sie beispielsweise von der Einstellung „Bucket-Eigentümer erzwungen“ zurück zur Einstellung „Objektschreiber“ wechseln, besitzen Sie als Bucket-Eigentümer nicht mehr die volle Kontrolle über Objekte, die zuvor anderen AWS-Konten gehörten. Stattdessen besitzen die Upload-Konten diese Objekte erneut. Objekte, die anderen Konten gehören, verwenden ACLs für Berechtigungen. Sie können also keine Richtlinien verwenden, um diesen Objekten Berechtigungen zu gewähren. Sie als Bucket-Eigentümer besitzen jedoch weiterhin alle Objekte, die in den Bucket geschrieben wurden, während die Einstellung „Bucket-Eigentümer erzwungen“ angewendet wurde. Diese Objekte gehören nicht dem Object Writer, auch wenn Sie sie erneut aktivierenACLs.

Anweisungen zur Aktivierung und Verwaltung ACLs mithilfe von AWS Management Console, AWS Command Line Interface (CLI) REST API AWS SDKs, oder finden Sie unterKonfiguration ACLs.

Voraussetzungen für die Deaktivierung ACLs

Bevor Sie die Deaktivierung ACLs für einen vorhandenen Bucket durchführen, müssen Sie die folgenden Voraussetzungen erfüllen.

Überprüfen Sie den Bucket und das Objekt ACLs und migrieren Sie die ACL Berechtigungen

Wenn Sie die Option deaktivierenACLs, wirken sich die von Bucket und Objekt erteilten Berechtigungen nicht ACLs mehr auf den Zugriff aus. Überprüfen Sie vor der Deaktivierung ACLs Ihren Bucket und Ihr ObjektACLs.

Wenn Ihr Bucket anderen Benutzern außerhalb Ihres Kontos Lese- oder Schreibberechtigungen ACLs gewährt, müssen Sie diese Berechtigungen in Ihre Bucket-Richtlinie migrieren, bevor Sie die erzwungene Einstellung für den Bucket-Besitzer anwenden können. Wenn Sie BucketsACLs, die Lese- oder Schreibzugriff außerhalb Ihres Kontos gewähren, nicht migrieren, schlägt Ihre Anfrage zur Anwendung der erzwungenen Einstellung „Bucket-Besitzer“ fehl und gibt den InvalidBucketAclWithObjectOwnershipFehlercode.

Wenn Sie die Option beispielsweise ACLs für einen Bucket deaktivieren möchten, der Serverzugriffsprotokolle empfängt, müssen Sie die ACL Bucket-Berechtigungen für die S3-Protokollbereitstellungsgruppe in einer Bucket-Richtlinie auf den Logging-Serviceprinzipal migrieren. Weitere Informationen finden Sie unter Gewähren von Zugriff auf die S3-Protokollbereitstellungsgruppe für die Protokollierung des Serverzugriffs.

Wenn Sie möchten, dass der Objektschreiber die volle Kontrolle über das hochgeladene Objekt behält, ist der Objektschreiber die beste Einstellung für die Objekteigentümerschaft in Ihrem

Anwendungsfall. Wenn Sie den Zugriff auf der Ebene einzelner Objekte steuern möchten, ist Bucket-Eigentümer bevorzugt die beste Wahl. Diese Anwendungsfälle sind ungewöhnlich.

Informationen zum Überprüfen ACLs und Migrieren von ACL Berechtigungen zu Bucket-Richtlinien finden Sie unterVoraussetzungen für die Deaktivierung ACLs.

Identifizieren Sie Anfragen, ACL für die eine Autorisierung erforderlich war

Um Amazon S3 S3-Anfragen zu identifizieren, die ACLs für eine Autorisierung erforderlich sind, können Sie den `aclRequired` Wert in den Amazon S3 S3-Serverzugriffsprotokollen oder verwenden AWS CloudTrail. Wenn ACL für die Anfrage eine Autorisierung erforderlich war oder wenn Sie PUT Anfragen haben, die eine angebenACL, lautet die ZeichenfolgeYes. Wenn keine erforderlich ACLs waren oder wenn Sie einen `bucket-owner-full-control` gespeicherten Wert festlegen oder wenn die Anfragen gemäß Ihrer Bucket-Richtlinie zulässig sind, lautet die `aclRequired` Wertzeichenfolge in den Amazon S3 S3-Serverzugriffsprotokollen - "" und fehlt in CloudTrail. ACL Weitere Informationen zu den erwarteten Werten für `aclRequired` finden Sie unter `aclRequired`-Werte für allgemeine Amazon-S3-Anfragen.

Wenn Sie `PutObjectAcl` Anfragen mit Headern haben`PutBucketAcl`, die bestimmte ACL Berechtigungen gewähren, mit Ausnahme der `bucket-owner-full-control` gespeichertenACL, müssen Sie diese Header entfernen, bevor Sie sie deaktivieren können. ACLs Andernfalls schlagen Ihre Anfragen fehl.

Für alle anderen Anfragen, ACL für die eine Autorisierung erforderlich ist, migrieren Sie diese ACL Berechtigungen in die Bucket-Richtlinien. Entfernen Sie anschließend alle Buckets, ACLs bevor Sie die Einstellung „Bucket Owner erforced“ aktivieren.

> **Note**
>
> Objekt ACLs nicht entfernen. Andernfalls verlieren Anwendungen, die ACLs für Berechtigungen auf Objekte angewiesen sind, den Zugriff.

Wenn Sie feststellen, dass keine Anfragen ACL zur Autorisierung erforderlich sind, können Sie mit der Deaktivierung fortfahrenACLs. Weitere Informationen zur Identifizierung von Anfragen finden Sie unter Verwenden von Amazon-S3-Serverzugriffsprotokollen zur Identifizierung von Anforderungen und Identifizieren von Amazon S3 S3-Anfragen mithilfe von CloudTrail.

Überprüfen und aktualisieren Sie Bucket-Richtlinien, die verwandte ACL Bedingungsschlüssel verwenden

Nachdem Sie die Einstellung Bucket Owner erforced auf Deaktivierung angewendet habenACLs, können neue Objekte nur dann in Ihren Bucket hochgeladen werden, wenn für die Anfrage die Vollzugriff auf den Bucket Owner verwendet wird ACLs oder wenn kein Bucket Owner angegeben istACL. Überprüfen Sie vor der Deaktivierung ACLs Ihre Bucket-Richtlinie auf zugehörige ACL Bedingungsschlüssel.

Wenn in Ihrer Bucket-Richtlinie ein ACL verwandter Bedingungsschlüssel verwendet wird, `bucket-owner-full-control` der ACL vordefinierte Daten verlangt (z. B.`s3:x-amz-acl`), müssen Sie Ihre Bucket-Richtlinie nicht aktualisieren. Die folgende Bucket-Richtlinie verwendet dies`s3:x-amz-acl`, um zu verlangen, dass `bucket-owner-full-control` die `PutObject` Anfragen ACL für S3 gesperrt werden. Diese Richtlinie verlangt immer noch, dass der Object Writer die gespeicherten Daten `bucket-owner-full-control` spezifiziertACL. ACLsDeaktivierte Buckets akzeptieren dies jedoch weiterhinACL, sodass Anfragen weiterhin erfolgreich sind, ohne dass clientseitige Änderungen erforderlich sind.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Only allow writes to my bucket with bucket owner full control",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:user/ExampleUser"
                ]
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        }
    ]
```

```
}
```

Wenn Ihre Bucket-Richtlinie jedoch einen verwandten ACL Bedingungsschlüssel verwendet, für den ein anderer erforderlich istACL, müssen Sie diesen Bedingungsschlüssel entfernen. Diese Beispiel-Bucket-Richtlinie erfordert die `public-read` ACL für `PutObject` S3-Anfragen und muss daher vor der Deaktivierung ACLs aktualisiert werden.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Only allow writes to my bucket with public read access",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:user/ExampleUser"                ]
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "public-read"
                }
            }
        }
    ]
}
```

Berechtigungen für Object Ownership

Um eine Einstellung für Object Ownership für einen Bucket anzuwenden, zu aktualisieren oder zu löschen, benötigen Sie die `s3:PutBucketOwnershipControls`-Berechtigung. Um die Object-Ownership-Einstellung für einen Bucket zurückzugeben, benötigen Sie die `s3:GetBucketOwnershipControls`-Berechtigung. Weitere Informationen erhalten Sie unter Festlegen von Object Ownership beim Erstellen eines Buckets und Anzeigen der Einstellung Object Ownership für einen S3-Bucket.

Deaktivierung ACLs für alle neuen Buckets

Standardmäßig werden alle neuen Buckets mit der Einstellung Bucket Owner erforced erstellt und sind deaktiviert. ACLs Wir empfehlen, diese Option weiterhin deaktiviert zu lassen. ACLs In der Regel empfehlen wir, stattdessen ressourcenbasierte S3-Richtlinien (Bucket-Richtlinien und Zugriffspunktrichtlinien) oder IAM Richtlinien für die Zugriffskontrolle zu verwenden. ACLs Richtlinien stellen eine vereinfachte und flexiblere Zugriffskontrolloption dar. Mit Bucket- und Zugriffspunktrichtlinien können Sie Regeln definieren, die allgemein für alle Anfragen an Ihre Amazon-S3-Ressourcen gelten.

Replikation und Object Ownership

Wenn Sie die S3-Replikation verwenden und die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten, können Sie deaktivieren ACLs (wobei die Einstellung Bucket-Besitzer für Objekteigentümer erzwungen wird), um den Besitz des Replikats auf den AWS-Konto Besitzer des Ziel-Buckets zu ändern. Diese Einstellung ahmt das Verhalten der bestehenden Besitzerüberschreibung nach, ohne dass eine `s3:ObjectOwnerOverrideToBucketOwner`-Berechtigung erforderlich ist. Alle Objekte, die mit der Einstellung „Bucket-Eigentümer erzwungen“ in den Ziel-Bucket repliziert werden, gehören dem Eigentümer des Ziel-Buckets. Weitere Informationen zur Besitzerüberschreibungsoption für Replikationskonfigurationen finden Sie unter Ändern des Replikat-Eigentümers.

Einstellung von Object Ownership

Sie können eine Einstellung für Objektbesitz mithilfe der Amazon S3-Konsole, AWS CLI, AWS SDKs, Amazon S3 REST API oder anwenden AWS CloudFormation. Die folgenden AWS CLI Befehle REST API und unterstützen Object Ownership:

REST API	AWS CLI	Beschreibung
PutBucketOwnershipControls	put-bucket-ownership-controls	Erstellt oder ändert die Einstellung Object Ownership für einen vorhandenen S3-Bucket.
CreateBucket	create-bucket	Erstellt einen Bucket mit dem `x-amz-object-ownership` -Anforderungsheader

REST API	AWS CLI	Beschreibung
		zur Angabe der Einstellung für Object Ownership.
GetBucketOwnershipControls	get-bucket-ownership-controls	Ruft die Einstellung Object Ownership für einen Amazon-S3-Bucket ab.
DeleteBucketOwnershipControls	delete-bucket-ownership-controls	Löscht die Einstellung für Object Ownership für einen Amazon-S3-Bucket.

Weitere Informationen zum Anwenden und Arbeiten mit Object Ownership-Einstellungen finden Sie in den folgenden Themen.

Themen

- Voraussetzungen für die Deaktivierung ACLs
- Festlegen von Object Ownership beim Erstellen eines Buckets
- Einstellung für Object Ownership für einen vorhandenen Bucket
- Anzeigen der Einstellung Object Ownership für einen S3-Bucket
- Deaktivierung ACLs für alle neuen Buckets und Durchsetzung des Objektbesitzes
- Fehlerbehebung

Voraussetzungen für die Deaktivierung ACLs

Wenn Ihr Bucket Zugriff außerhalb Ihres ACL gewährt AWS-Konto, müssen Sie vor der Deaktivierung ACLs Ihre ACL Bucket-Berechtigungen auf Ihre Bucket-Richtlinie migrieren und Ihren Bucket ACL auf die Standardeinstellung „PrivatACL“ zurücksetzen. Wenn Sie diese Buckets nicht migrierenACLs, schlägt Ihre Anfrage ACLs fehl, die Einstellung „Bucket Owner erforced“ auf Deaktivierung anzuwenden, und es wird InvalidBucketAclWithObjectOwnershipFehlercode. Wir empfehlen Ihnen außerdem, Ihre ACL Objektberechtigungen zu überprüfen und sie in Ihre Bucket-Richtlinie zu migrieren. Weitere Informationen zu anderen empfohlenen Voraussetzungen finden Sie unter Voraussetzungen für die Deaktivierung ACLs.

Für jeden Ihrer vorhandenen Buckets und Objekte ACLs gibt es ein Äquivalent in einer IAM Richtlinie. Die folgenden Beispiele für Bucket-Richtlinien zeigen Ihnen, wie `READ` und wie die `WRITE` Berechtigungen für Bucket und Objekt ACLs den IAM Berechtigungen zugeordnet werden. Weitere Informationen darüber, wie beide Begriffe in IAM Berechtigungen ACL übersetzt werden, finden Sie unterZuordnung von ACL Berechtigungen und Zugriffsrichtlinienberechtigungen.

Informationen zur Überprüfung und Migration von ACL Berechtigungen zu Bucket-Richtlinien finden Sie in den folgenden Themen.

Themen

- Beispiele für Bucket-Richtlinie
- Verwenden Sie die S3-Konsole, um ACL Berechtigungen zu überprüfen und zu migrieren
- Verwenden Sie das AWS CLI , um ACL Berechtigungen zu überprüfen und zu migrieren
- Beispielhafte Walkthroughs

Beispiele für Bucket-Richtlinie

Diese Beispiel-Bucket-Richtlinien zeigen Ihnen, wie Sie `WRITE` Bucket- und ACL Objektberechtigungen für Dritte AWS-Konto auf eine Bucket-Richtlinie migrieren`READ`. `READ_ACP`und `WRITE_ACP` ACLs sind für Richtlinien weniger relevant, da sie ACL entsprechende Berechtigungen gewähren (`s3:GetBucketAcls3:GetObjectAcl`,`s3:PutBucketAcl`, und`s3:PutObjectAcl`).

Example — **`READ`** ACL für einen Eimer

Wenn Ihr Bucket eine hatte `READ`ACL, die die AWS-Konto *`111122223333`* Erlaubnis erteilt, den Inhalt Ihres Buckets aufzulisten, können Sie eine Bucket-Richtlinie schreiben, dies`s3:ListBucket`,`s3:ListBucketVersions`, `s3:ListBucketMultipartUploads` Berechtigungen für Ihren Bucket gewährt.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Permission to list the objects in a bucket",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
```

```
                    "arn:aws:iam::111122223333:root"
                ]
            },
            "Action": [
                "s3:ListBucket",
                "s3:ListBucketVersions",
                "s3:ListBucketMultipartUploads"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket"
        }
    ]
}
```

Example — **READ** ACLs für jedes Objekt in einem Bucket

Wenn jedes Objekt in Ihrem Bucket über ein Objekt verfügt, READ ACL das Zugriff gewährt AWS-Konto *111122223333*, können Sie eine Bucket-Richtlinie schreiben, die diesem Konto für jedes Objekt in Ihrem Bucket `s3:GetObjectVersion` Berechtigungen gewährt`s3:GetObject`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Read permission for every object in a bucket",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:root"
                ]
            },
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Dieses Beispiel-Ressourcenelement gewährt Zugriff auf ein bestimmtes Objekt.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/OBJECT-KEY"
```

Example — **`WRITE`** ACL die berechtigt, Objekte in einen Bucket zu schreiben

Wenn Ihr Bucket über eine verfügt `WRITEACL`, die die AWS-Konto *`111122223333`* Erlaubnis erteilt, Objekte in Ihren Bucket zu schreiben, können Sie eine Bucket-Richtlinie schreiben, die die `s3:PutObject` Erlaubnis für Ihren Bucket erteilt.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Permission to write objects to a bucket",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:root"
                ]
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Verwenden Sie die S3-Konsole, um ACL Berechtigungen zu überprüfen und zu migrieren

Um die ACL Berechtigungen eines Buckets zu überprüfen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus.
3. Wählen Sie die Registerkarte Berechtigungen.
4. Überprüfen Sie unter Zugriffskontrollliste (ACL) Ihre ACL Bucket-Berechtigungen.

Um die ACL Berechtigungen eines Objekts zu überprüfen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der Ihr Objekt enthält.
3. Wählen Sie in der Liste Objekts (Objekte) den Objektnamen aus.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Überprüfen Sie unter Zugriffskontrollliste (ACL) Ihre ACL Objektberechtigungen.

Um ACL Berechtigungen zu migrieren und Ihren Bucket zu aktualisieren ACL

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus.
3. Wählen Sie auf der Registerkarte Berechtigungen unter Bucket-Richtlinie die Option Bearbeiten aus.
4. Fügen Sie im Feld Richtlinie Ihre Bucket-Richtlinie hinzu oder aktualisieren Sie sie.

   Beispiele für Bucket-Richtlinien finden Sie unter Beispiele für Bucket-Richtlinie und Beispielhafte Walkthroughs.
5. Wählen Sie Änderungen speichern.
6. Aktualisieren Sie Ihren Bucket ACL, um ACL Zuschüsse für andere Gruppen zu entfernen oder AWS-Konten.
7. Wenden Sie die Einstellung Bucket-Eigentümer erzwungen für die Objekteigentümerschaft an.

Verwenden Sie das AWS CLI , um ACL Berechtigungen zu überprüfen und zu migrieren

1. Um den Bucket ACL für Ihren Bucket zurückzugeben, verwenden Sie den get-bucket-acl AWS CLI Befehl:

   ```
   aws s3api get-bucket-acl --bucket amzn-s3-demo-bucket
   ```

   Dieser Bucket ACL gewährt `WRITE` beispielsweise `READ` Zugriff auf ein Drittanbieter-Konto. Dabei wird ACL das Drittanbieter-Konto anhand der kanonischen Benutzer-ID identifiziert. Um die Einstellung „Bucket-Besitzer erzwungen“ anzuwenden und zu deaktivierenACLs, müssen Sie diese Berechtigungen für das Drittanbieter-Konto auf eine Bucket-Richtlinie migrieren.

   ```
   {
   ```

```
    "Owner": {
        "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
    },
    "Grants": [
        {
            "Grantee": {
                "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
                "ID":
 "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",
                "Type": "CanonicalUser"
            },
            "Permission": "FULL_CONTROL"
        },
        {
            "Grantee": {
                "DisplayName": "THIRD-PARTY-EXAMPLE-ACCOUNT",
                "ID":
 "72806de9d1ae8b171cca9e2494a8d1335dfced4ThirdPartyAccountCanonicalUserID",
                "Type": "CanonicalUser"
            },
            "Permission": "READ"
        },
        {
            "Grantee": {
                "DisplayName": "THIRD-PARTY-EXAMPLE-ACCOUNT",
                "ID":
 "72806de9d1ae8b171cca9e2494a8d1335dfced4ThirdPartyAccountCanonicalUserID",
                "Type": "CanonicalUser"
            },
            "Permission": "WRITE"
        }
    ]
}
```

Ein anderes Beispiel finden Sie ACLs unterBeispielhafte Walkthroughs.

2. Migrieren Sie Ihre ACL Bucket-Berechtigungen zu einer Bucket-Richtlinie:

   In diesem Beispiel gewährt Bucket-Richtlinien `s3:PutObject` und `s3:ListBucket`Berechtigungen für ein Drittanbieterkonto. In der Bucket-Richtlinie wird das Drittanbieter-Konto anhand der AWS-Konto ID (*111122223333*) identifiziert.

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy file://
policy.json

policy.json:
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PolicyForCrossAccountAllowUpload",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:root"
                ]
            },
            "Action": [
                "s3:PutObject",
                "s3:ListBucket"
            ],
            "Resource": [
             "arn:aws:s3:::amzn-s3-demo-bucket",
             "arn:aws:s3:::amzn-s3-demo-bucket/*"
        }
    ]
}
```

Weitere Beispiele für Bucket-Richtlinien finden Sie unter Beispiele für Bucket-Richtlinie und Beispielhafte Walkthroughs.

3. Um das ACL für ein bestimmtes Objekt zurückzugeben, verwenden Sie den get-object-acl AWS CLI Befehl.

```
aws s3api get-object-acl --bucket amzn-s3-demo-bucket --key EXAMPLE-OBJECT-KEY
```

4. Migrieren Sie bei Bedarf ACL Objektberechtigungen zu Ihrer Bucket-Richtlinie.

   Dieses Beispiel-Ressourcenelement gewährt Zugriff auf ein bestimmtes Objekt in einer Bucket-Richtlinie.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/EXAMPLE-OBJECT-KEY"
```

5. Setzen Sie die ACL für Ihren Bucket auf die Standardeinstellung zurückACL.

```
aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --acl private
```

6. Wenden Sie die Einstellung „Bucket-Eigentümer erzwungen“ für die Objekteigentümerschaft an.

Beispielhafte Walkthroughs

Die folgenden Beispiele zeigen Ihnen, wie Sie ACL Berechtigungen für bestimmte Anwendungsfälle auf Bucket-Richtlinien migrieren.

Themen

- Gewähren von Zugriff auf die S3-Protokollbereitstellungsgruppe für die Protokollierung des Serverzugriffs
- Öffentlichen Lesezugriff auf die Objekte in einem Bucket gewähren
- Gewähren Sie Amazon ElastiCache (RedisOSS) Zugriff auf Ihren S3-Bucket

Gewähren von Zugriff auf die S3-Protokollbereitstellungsgruppe für die Protokollierung des Serverzugriffs

Wenn Sie die Einstellung Bucket Owner erforced anwenden möchten, um sie ACLs für einen Ziel-Bucket für die Serverzugriffsprotokollierung (auch als Ziel-Bucket bezeichnet) zu deaktivieren, müssen Sie die ACL Bucket-Berechtigungen für die S3-Protokollzustellungsgruppe in einer Bucket-Richtlinie auf den Logging-Serviceprinzipal (`logging.s3.amazonaws.com`) migrieren. Weitere Informationen zu Berechtigungen für die Protokollzustellung finden Sie unter Berechtigungen für Protokollbereitstellung.

Dieser Bucket ACL gewährt `WRITE` `READ_ACP` Zugriff auf die S3-Protokollbereitstellungsgruppe:

```
{
    "Owner": {
        "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
    },
    "Grants": [
        {
            "Grantee": {
                "Type": "CanonicalUser",
                "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
```

```
                "ID":
 "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
            },
            "Permission": "FULL_CONTROL"
        },
        {
            "Grantee": {
                "Type": "Group",
                "URI": "http://acs.amazonaws.com/groups/s3/LogDelivery"
            },
            "Permission": "WRITE"
        },
        {
            "Grantee": {
                "Type": "Group",
                "URI": "http://acs.amazonaws.com/groups/s3/LogDelivery"
            },
            "Permission": "READ_ACP"
        }
    ]
}
```

Um ACL Bucket-Berechtigungen für die S3-Protokollzustellungsgruppe in einer Bucket-Richtlinie auf den Logging-Serviceprinzipal zu migrieren

1. Fügen Sie Ihrem Ziel-Bucket die folgende Bucket-Richtlinie hinzu. wobei Sie die Beispielwerte ersetzen.

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy file://
policy.json

policy.json:      {
    {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "S3ServerAccessLogsPolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "logging.s3.amazonaws.com"
            },
            "Action": [
                "s3:PutObject"
```

```
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/EXAMPLE-LOGGING-PREFIX*",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::SOURCE-BUCKET-NAME"
                },
                "StringEquals": {
                    "aws:SourceAccount": "SOURCE-AWS-ACCOUNT-ID"
                }
            }
        }
    ]
}
```

2. Setzen Sie den Bucket ACL für Ihren Ziel-Bucket auf die Standardeinstellung zurückACL.

```
aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --acl private
```

3. Wenden Sie die Einstellung „Von Bucket-Besitzer erzwungen“ für den Objektbesitz auf Ihren Ziel-Bucket an.

Öffentlichen Lesezugriff auf die Objekte in einem Bucket gewähren

Wenn Ihr Objekt öffentlichen Lesezugriff auf alle Objekte in Ihrem Bucket ACLs gewährt, können Sie diese ACL Berechtigungen in eine Bucket-Richtlinie migrieren.

Dieses Objekt ACL gewährt öffentlichen Lesezugriff auf ein Objekt in einem Bucket:

```
{
    "Owner": {
        "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
    },
    "Grants": [
        {
            "Grantee": {
                "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
                "ID":
 "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",
                "Type": "CanonicalUser"
            },
            "Permission": "FULL_CONTROL"
        },
```

```
        {
            "Grantee": {
                "Type": "Group",
                "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
            },
            "Permission": "READ"
        }
    ]
}
```

Um öffentliche ACL Leseberechtigungen auf eine Bucket-Richtlinie zu migrieren

1. Um öffentlichen Lesezugriff auf alle Objekte in Ihrem Bucket zu gewähren, fügen Sie die folgende Bucket-Richtlinie hinzu und ersetzen Sie die Beispielwerte.

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy
 file://policy.json

policy.json:
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

   Um öffentlichen Zugriff auf ein bestimmtes Objekt in einer Bucket-Richtlinie zu gewähren, verwenden Sie das folgende Format für das `Resource`-Element.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/OBJECT-KEY"
```

Um öffentlichen Zugriff auf alle Objekte mit einem bestimmten Präfix zu gewähren, verwenden Sie das folgende Format für das `Resource`-Element.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/PREFIX/*"
```

2. Wenden Sie die Einstellung „Bucket-Eigentümer erzwungen“ für die Objekteigentümerschaft an.

Gewähren Sie Amazon ElastiCache (RedisOSS) Zugriff auf Ihren S3-Bucket

Sie können Ihr ElastiCache (Redis-OSS) Backup in einen S3-Bucket exportieren, sodass Sie von außen auf das Backup zugreifen können. ElastiCache Um Ihr Backup in einen S3-Bucket zu exportieren, müssen Sie ElastiCache Berechtigungen zum Kopieren eines Snapshots in den Bucket erteilen. Wenn Sie ElastiCache in einem Bucket Berechtigungen erteilt habenACL, müssen Sie diese Berechtigungen in eine Bucket-Richtlinie migrieren, bevor Sie die Einstellung Bucket owner erforced auf Deaktivierung ACLs anwenden. Weitere Informationen finden Sie unter ElastiCache Zugriff auf Ihren Amazon S3 S3-Bucket gewähren im ElastiCache Amazon-Benutzerhandbuch.

Das folgende Beispiel zeigt die ACL Bucket-Berechtigungen, denen Berechtigungen gewährt werden ElastiCache.

```
{
    "Owner": {
        "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
    },
    "Grants": [
        {
            "Grantee": {
                "DisplayName": "DOC-EXAMPLE-ACCOUNT-OWNER",
                "ID":
 "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",
                "Type": "CanonicalUser"
            },
            "Permission": "FULL_CONTROL"
        },
        {
            "Grantee": {
                "DisplayName": "aws-scs-s3-readonly",
                "ID":
 "540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353",
```

```
                "Type": "CanonicalUser"
            },
            "Permission": "READ"
        },
        {
            "Grantee": {
                "DisplayName": "aws-scs-s3-readonly",
                "ID":
 "540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353",
                "Type": "CanonicalUser"
            },
            "Permission": "WRITE"
        },
        {
            "Grantee": {
                "DisplayName": "aws-scs-s3-readonly",
                "ID":
 "540804c33a284a299d2547575ce1010f2312ef3da9b3a053c8bc45bf233e4353",
                "Type": "CanonicalUser"
            },
            "Permission": "READ_ACP"
        }
    ]
}
```

Um ACL Bucket-Berechtigungen für ElastiCache (RedisOSS) zu einer Bucket-Richtlinie zu migrieren

1. Fügen Sie Ihrem Bucket die folgende Bucket-Richtlinie hinzu und ersetzen Sie die Beispielwerte.

```
aws s3api put-bucket-policy --bucket amzn-s3-demo-bucket --policy
 file://policy.json

policy.json:
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt15399483",
            "Effect": "Allow",
            "Principal": {
                "Service": "Region.elasticache-snapshot.amazonaws.com"
            },
            "Action": [
```

```
                "s3:PutObject",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:GetBucketAcl",
                "s3:ListMultipartUploadParts",
                "s3:ListBucketMultipartUploads"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket",
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        }
    ]
}
```

2. Setzen Sie die ACL für Ihren Bucket auf die Standardeinstellung zurückACL:

```
aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --acl private
```

3. Wenden Sie die Einstellung „Bucket-Eigentümer erzwungen“ für die Objekteigentümerschaft an.

Festlegen von Object Ownership beim Erstellen eines Buckets

Wenn Sie einen Bucket erstellen, können Sie S3 Object Ownership konfigurieren. Informationen zum Festlegen von Object Ownership für einen vorhandenen Bucket finden Sie unter Einstellung für Object Ownership für einen vorhandenen Bucket.

S3 Object Ownership ist eine Einstellung auf Amazon S3-Bucket-Ebene, mit der Sie Zugriffskontrolllisten (ACLs) deaktivieren und die Verantwortung für jedes Objekt in Ihrem Bucket übernehmen können, wodurch die Zugriffsverwaltung für in Amazon S3 gespeicherte Daten vereinfacht wird. Standardmäßig ist für S3 Object Ownership die erzwungene Einstellung Bucket Owner festgelegt und ACLs für neue Buckets deaktiviert. Bei ACLs deaktivierter Option besitzt der Bucket-Besitzer jedes Objekt im Bucket und verwaltet den Zugriff auf Daten ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen.

Object Ownership verfügt über drei Einstellungen, mit denen Sie den Besitz von in Ihren Bucket hochgeladenen Objekten kontrollieren und deaktivieren oder aktivieren könnenACLs:

ACLsdeaktiviert

- Bucket-Besitzer erzwungen (Standard) — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLswirkt sich nicht mehr auf die Berechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet Richtlinien, um die Zugriffssteuerung zu definieren.

ACLsaktiviert

- Bevorzugter Bucket-Besitzer — Der Bucket-Besitzer besitzt und hat die volle Kontrolle über neue Objekte, die andere Accounts in den Bucket mit den gespeicherten Objekten `bucket-owner-full-control` schreibenACL.
- Objekt-Writer — AWS-Konto Derjenige, der ein Objekt hochlädt, besitzt das Objekt, hat die volle Kontrolle darüber und kann anderen Benutzern Zugriff darauf gewähren. ACLs

Berechtigungen: Um die Einstellung Bucket-Eigentümer erzwungen oder Bucket-Eigentümer bevorzugt anzuwenden, müssen Sie über die folgenden Berechtigungen verfügen: `s3:CreateBucket` und `s3:PutBucketOwnershipControls`. Wenn Sie einen Bucket mit aktivierter Einstellung Object writer (Objektschreiber), sind keine zusätzlichen Berechtigungen erforderlich. Weitere Informationen zu Amazon S3-Berechtigungen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

> **Important**
>
> In den meisten modernen Anwendungsfällen in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs, und wir empfehlen, diese zu deaktivieren, ACLs außer in ungewöhnlichen Fällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Mit Object Ownership können Sie Richtlinien für die Zugriffskontrolle deaktivieren ACLs und sich auf diese verlassen. Wenn Sie die Option deaktivierenACLs, können Sie ganz einfach einen Bucket mit Objekten verwalten, die von verschiedenen AWS Konten hochgeladen wurden. Sie als Bucket-Eigentümer besitzen alle Objekte im Bucket und können den Zugriff darauf mithilfe von Richtlinien verwalten.

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen der aktuell angezeigten Datei aus AWS-Region. Wählen Sie als Nächstes die Region aus, in der Sie einen Bucket erstellen möchten.

   > **Note**
   >
   > Wählen Sie eine Region in der Nähe aus, um Latenz und Kosten gering zu halten und behördliche Vorschriften zu erfüllen. In einer Region gespeicherte Objekte verbleiben so lange in der Region, bis sie explizit in eine andere Region verschoben werden. Eine Liste von Amazon S3 finden Sie AWS-Regionen unter AWS-Service Endpoints in der Allgemeine Amazon Web Services-Referenz.

3. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
4. Wählen Sie Create Bucket (Bucket erstellen) aus.

   Anschließend wird die Seite Bucket erstellen geöffnet.
5. Sehen Sie sich unter Allgemeine Konfiguration an, AWS-Region wo Ihr Bucket erstellt wird.
6. Wählen Sie unter Bucket-Typ die Option Allgemeiner Zweck aus.
7. Geben Sie unter Bucket Name (Bucket-Name) einen Namen für den Bucket ein.

   Der Bucket-Name ...:

   - Muss innerhalb einer Partition einzigartig sein. Eine Partition ist eine Gruppierung von Regionen. AWS verfügt derzeit über drei Partitionen: `aws` (Standardregionen), `aws-cn` (China-Regionen) und `aws-us-gov` (AWS GovCloud (US) Regions).
   - zwischen 3 und 63 Zeichen lang sein,
   - Darf nur aus Kleinbuchstaben, Zahlen, Punkten (.) und Bindestrichen (-) bestehen. Aus Gründen der besten Kompatibilität empfehlen wir, Punkte (.) in Bucket-Namen zu vermeiden, mit Ausnahme von Buckets, die nur für statisches Website-Hosting verwendet werden.
   - Muss mit einer Zahl oder einem Buchstaben beginnen und enden.

Der Name eines einmal erstellter Buckets kann nicht nachträglich geändert werden. AWS-Konto Derjenige, der den Bucket erstellt, besitzt ihn. Weitere Informationen zur Benennung von Buckets finden Sie unter Allgemeine Benennungsregeln für Buckets.

> **Important**
>
> Vermeiden Sie, vertrauliche Informationen, wie Kontonummern, in den Bucket-Namen einzubeziehen. Der Bucket-Name ist in dem Punkt sichtbarURLs, auf den die Objekte im Bucket verweisen.

8. AWS Management Console ermöglicht es Ihnen, die Einstellungen eines vorhandenen Buckets in Ihren neuen Bucket zu kopieren. Wenn Sie die Einstellungen eines vorhandenen Buckets nicht kopieren möchten, fahren Sie mit dem nächsten Schritt fort.

> **Note**
>
> Diese Option:
>
> - Ist in der nicht verfügbar AWS CLI und ist nur in der Konsole verfügbar
> - Ist für Verzeichnis-Buckets nicht verfügbar
> - Kopiert die Bucket-Richtlinie nicht aus dem vorhandenen Bucket in den neuen Bucket

Um die Einstellungen eines vorhandenen Buckets zu kopieren, wählen Sie unter Einstellungen aus vorhandenem Bucket kopieren die Option Bucket auswählen aus. Das Fenster „Bucket auswählen“ wird geöffnet. Suchen Sie den Bucket mit den Einstellungen, die Sie kopieren möchten, und wählen Sie Bucket auswählen aus. Das Fenster „Bucket auswählen“ wird geschlossen, und das Fenster „Bucket erstellen“ wird erneut geöffnet.

Unter Einstellungen aus vorhandenem Bucket kopieren wird nun der Name des ausgewählten Buckets angezeigt. Außerdem wird die Option Standardwerte wiederherstellen angezeigt, mit der Sie die kopierten Bucket-Einstellungen entfernen können. Überprüfen Sie die verbleibenden Bucket-Einstellungen auf der Seite Bucket erstellen. Sie werden sehen, dass sie jetzt mit den Einstellungen des von Ihnen ausgewählten Buckets übereinstimmen. Sie können mit dem letzten Schritt fortfahren.

9. Wählen Sie unter Object Ownership eine der folgenden Einstellungen aus, um die Inhaberschaft von Objekten, die in Ihren Bucket hochgeladen wurden, zu deaktivieren oder zu aktivieren ACLs und zu kontrollieren:

   ACLsdeaktiviert

   - Bucket-Besitzer erzwungen (Standard) — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLswirkt sich nicht mehr auf die Zugriffsberechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet ausschließlich Richtlinien, um die Zugriffssteuerung zu definieren.

     Standardmäßig ACLs sind sie deaktiviert. Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs. Wir empfehlen, die ACLs Option weiterhin zu deaktivieren, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

   ACLsaktiviert

   - Bevorzugter Bucket-Besitzer — Der Bucket-Besitzer besitzt und hat die volle Kontrolle über neue Objekte, die andere Accounts in den Bucket mit den gespeicherten Objekten `bucket-owner-full-control` schreibenACL.

     Wenn Sie die bevorzugte Einstellung des Bucket-Besitzers anwenden, sodass alle Amazon S3 S3-Uploads die gespeicherten Dateien enthalten müssen `bucket-owner-full-control`ACL, können Sie eine Bucket-Richtlinie hinzufügen, die nur Objekt-Uploads zulässt, die dies verwenden. ACL
   - Objekt-Writer — Derjenige AWS-Konto , der ein Objekt hochlädt, besitzt das Objekt, hat die volle Kontrolle darüber und kann anderen Benutzern Zugriff darauf gewähren. ACLs

   > **Note**
   >
   > Die Standardeinstellung ist Bucket-Eigentümer erzwungen. Um die Standardeinstellung anzuwenden und ACLs deaktiviert zu lassen, ist nur die `s3:CreateBucket` entsprechende Berechtigung erforderlich. Zum Aktivieren ACLs benötigen Sie die `s3:PutBucketOwnershipControls` entsprechende Genehmigung.

10. Wählen Sie unter Einstellungen "Öffentlichen Zugriff beschränken" für diesen Bucket die Einstellungen zum Beschränken des öffentlichen Zugriffs aus, die Sie auf den Bucket anwenden möchten.

    Alle vier Einstellungen zum Blockieren des öffentlichen Zugriffs sind standardmäßig aktiviert. Es wird empfohlen, alle Einstellungen aktiviert zu lassen, es sei denn, Sie wissen, dass Sie eine oder mehrere dieser Einstellungen für Ihren Anwendungsfall deaktivieren müssen. Weitere Informationen zum Blockieren des öffentlichen Zugriffs finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

    > **Note**
    >
    > Zum Aktivieren aller Einstellungen zum Blockieren des öffentlichen Zugriffs ist nur die `s3:CreateBucket`-Berechtigung erforderlich. Wenn Sie eine der Einstellungen zum Blockieren des öffentlichen Zugriffs deaktivieren möchten, benötigen Sie die `s3:PutBucketPublicAccessBlock`-Berechtigung.

11. (Optional) Unter Bucket Versioning (Bucket-Versionsverwaltung) können Sie auswählen, ob Sie Varianten von Objekten in Ihrem Bucket beibehalten möchten. Weitere Informationen über das Versioning finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

    Wenn Sie die Versionsverwaltung in Ihrem Bucket deaktivieren oder aktivieren möchten, wählen Sie entweder Disable (Deaktivieren) oder Enable (Aktivieren) aus.

12. (Optional) Unter Tags können Sie auswählen, ob Sie Ihrem Bucket Tags hinzufügen möchten. Tags sind Schlüssel-Wert-Paare, die zur Kategorisierung von Speicher verwendet werden.

    Wenn Sie ein Bucket-Tag hinzuzufügen, geben Sie einen Key (Schlüssel) und optional einen Value (Wert) ein. Wählen Sie dann Add Tag (Tag hinzufügen) aus.

13. Wählen Sie unter Default encryption (Standard-Verschlüsselung) Edit (Bearbeiten) aus.

14. Wählen Sie eine der folgenden Optionen unter Verschlüsselungstyp aus, um die Standardverschlüsselung zu konfigurieren:

    - Verwalteter Amazon S3 S3-Schlüssel (SSE-S3)
    - AWS Key Management Service Schlüssel (SSE-KMS)

> **Important**
>
> Wenn Sie die KMS Option SSE - für Ihre standardmäßige Verschlüsselungskonfiguration verwenden, unterliegen Sie dem Kontingent für Anfragen pro Sekunde (RPS) von AWS KMS. Weitere Informationen zu AWS KMS Kontingenten und dazu, wie Sie eine Kontingenterhöhung beantragen können, finden Sie unter Kontingente im AWS Key Management Service Entwicklerhandbuch.

Buckets und neue Objekte werden mit serverseitiger Verschlüsselung verschlüsselt. Dabei ist ein Von Amazon S3 verwalteter Schlüssel die Grundebene der Verschlüsselungskonfiguration. Weitere Informationen zur Standardverschlüsselung finden Sie unter Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets.

Weitere Informationen zur Datenverschlüsselung mit der serverseitigen Amazon-S3-Verschlüsselung finden Sie unter Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden (SSE-S3).

15. Wenn Sie die AWS Key Management Service Taste (SSE-KMS) gewählt haben, gehen Sie wie folgt vor:

    a. Geben Sie unter AWS KMS Schlüssel Ihren KMS Schlüssel auf eine der folgenden Arten an:

    - Um aus einer Liste verfügbarer KMS Schlüssel zu wählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys und wählen Sie Ihren KMSSchlüssel aus der Liste der verfügbaren Schlüssel aus.

      Sowohl der Von AWS verwalteter Schlüssel (`aws/s3`) als auch Ihr vom Kunden verwalteter Schlüssel werden in dieser Liste angezeigt. Weitere Informationen über vom Kunden verwaltete Schlüssel finden Sie unter Kundenschlüssel und AWS -Schlüssel im Entwicklerhandbuch zu AWS Key Management Service .
    - Um den KMS Schlüssel einzugebenARN, wählen Sie Enter AWS KMS key ARN und geben Sie Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
    - Um einen neuen, vom Kunden verwalteten Schlüssel in der AWS KMS Konsole zu erstellen, wählen Sie Create a KMS key aus.

      Weitere Informationen zum Erstellen eines AWS KMS key finden Sie unter Schlüssel erstellen im AWS Key Management Service Entwicklerhandbuch.

> **Important**
>
> Sie können nur KMS Schlüssel verwenden, die im selben AWS-Region Bucket verfügbar sind. Die Amazon S3 S3-Konsole listet nur die ersten 100 KMS Schlüssel in derselben Region wie der Bucket auf. Um einen KMS Schlüssel zu verwenden, der nicht aufgeführt ist, müssen Sie Ihren KMS Schlüssel eingebenARN. Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, benötigen Sie zunächst die Erlaubnis, den Schlüssel zu verwenden, und dann müssen Sie den KMS Schlüssel eingebenARN. Weitere Informationen zu kontoübergreifenden Berechtigungen für KMS Schlüssel finden Sie im AWS Key Management Service Entwicklerhandbuch unter KMSSchlüssel erstellen, die andere Konten verwenden können. Weitere Informationen zu SSE - finden KMS Sie unterGeben Sie die serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) an. Wenn Sie eine AWS KMS key für die serverseitige Verschlüsselung in Amazon S3 verwenden, müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel und keine asymmetrischen KMS Schlüssel. Weitere Informationen finden Sie unter Identifizieren symmetrischer und asymmetrischer KMS Schlüssel im Entwicklerhandbuch.AWS Key Management Service

Weitere Informationen zum Erstellen von finden Sie unter Creating Keys im AWS Key Management Service Developer Guide. AWS KMS key Weitere Informationen zur Verwendung AWS KMS mit Amazon S3 finden Sie unterServerseitige Verschlüsselung mit AWS KMS Schlüsseln verwenden (SSE-) KMS.

b. Wenn Sie Ihren Bucket so konfigurieren, dass er die Standardverschlüsselung mit SSE - verwendetKMS, können Sie auch S3 Bucket Keys aktivieren. S3-Bucket-Keys senken die Kosten für die Verschlüsselung, indem sie den Anforderungsverkehr von Amazon S3 zu verringern AWS KMS. Weitere Informationen finden Sie unter Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys.

   Um S3-Bucket-Schlüssel zu verwenden, wählen Sie unter Bucket Key (Bucket-Schlüssel) die Option Enable (Aktivieren).

16. (Optional) Wenn Sie die S3-Objektsperre aktivieren möchten, gehen Sie wie folgt vor:

a. Wählen Sie Erweiterte Einstellungen aus.

> **Important**
>
> Durch Aktivieren der Objektsperre wird auch die Versioning für den Bucket aktiviert. Nach dem Aktivieren müssen Sie die Standardeinstellungen für die Objektsperre im Hinblick auf die (rechtliche) Aufbewahrung konfigurieren, um neue Objekte vor dem Löschen oder Überschreiben zu schützen.

b. Wenn Sie die Objektsperre aktivieren möchten, wählen Sie Enable (Aktivieren) aus, lesen Sie die angezeigte Warnung und bestätigen Sie sie.

Weitere Informationen finden Sie unter Sperren von Objekten mit Object Lock.

> **Note**
>
> Wenn Sie einen Bucket mit aktivierter Objektsperre erstellen möchten, benötigen Sie die folgenden Berechtigungen: `s3:CreateBucket`, `s3:PutBucketVersioning` und `s3:PutBucketObjectLockConfiguration`.

17. Wählen Sie Bucket erstellen aus.

Mit dem AWS CLI

Um den Objekteigentum festzulegen, wenn Sie einen neuen Bucket erstellen, verwenden Sie den `create-bucket` AWS CLI Befehl mit dem `--object-ownership` Parameter.

In diesem Beispiel wird die Einstellung „Bucket-Eigentümer erzwungen“ für einen neuen Bucket mithilfe der AWS CLI angewendet:

```
aws s3api create-bucket --bucket  amzn-s3-demo-bucket --region us-east-1 --object-
ownership BucketOwnerEnforced
```

> **Important**
>
> Wenn Sie bei der Erstellung eines Buckets mithilfe von keinen Objekteigentümer festlegen AWS CLI, ist die Standardeinstellung `ObjectWriter` (ACLsaktiviert).

Verwendung des AWS SDK for Java

In diesem Beispiel wird die Einstellung „Bucket-Eigentümer erzwungen“ für einen neuen Bucket mithilfe von AWS SDK for Java festgelegt:

```
    // Build the ObjectOwnership for CreateBucket
    CreateBucketRequest createBucketRequest = CreateBucketRequest.builder()
            .bucket(bucketName)
            .objectOwnership(ObjectOwnership.BucketOwnerEnforced)
            .build()

     // Send the request to Amazon S3
     s3client.createBucket(createBucketRequest);
```

Verwenden AWS CloudFormation

Informationen dazu, wie Sie bei der Erstellung eines neuen Buckets mithilfe der `AWS::S3::Bucket` AWS CloudFormation Ressource Objekteigentümer festlegen können, finden Sie unter OwnershipControls innerhalb AWS::S3::Bucket im AWS CloudFormation -Benutzerhandbuch.

Mit dem REST API

Um die erzwungene Einstellung des Bucket-Besitzers für S3 Object Ownership anzuwenden, verwenden Sie den `CreateBucket` API Vorgang, bei dem der `x-amz-object-ownership` Anforderungsheader auf `BucketOwnerEnforced` gesetzt ist. Informationen und Beispiele finden Sie unter CreateBucketin der Amazon Simple Storage Service API Reference.

Nächste Schritte: Nach der Anwendung der Einstellungen „Von Bucket-Besitzer erzwungen“ oder „Von Bucket-Besitzer bevorzugt“ auf den Objektbesitz können Sie die folgenden Schritte ausführen:

- Bucket-Besitzer erzwungen — Erfordert, dass alle neuen Buckets mit ACLs deaktivierter Einstellung erstellt werden, indem eine Richtlinie IAM oder Organizations verwendet wird.
- Bucket-Besitzer bevorzugt — Fügen Sie eine S3-Bucket-Richtlinie hinzu, die vorschreibt, dass ACL für alle Objekt-Uploads `bucket-owner-full-control` in Ihren Bucket „gesperrt“ gilt.

Einstellung für Object Ownership für einen vorhandenen Bucket

Sie können S3 Object Ownership für einen vorhandenen S3-Bucket konfigurieren. Zum Anwenden von Objekt-Ownership beim Erstellen eines Buckets finden Sie unter Festlegen von Object Ownership beim Erstellen eines Buckets.

S3 Object Ownership ist eine Einstellung auf Amazon S3-Bucket-Ebene, mit der Sie Zugriffskontrolllisten (ACLs) deaktivieren und die Verantwortung für jedes Objekt in Ihrem Bucket übernehmen können, wodurch die Zugriffsverwaltung für in Amazon S3 gespeicherte Daten vereinfacht wird. Standardmäßig ist für S3 Object Ownership die erzwungene Einstellung Bucket Owner festgelegt und ACLs für neue Buckets deaktiviert. Bei ACLs deaktivierter Option besitzt der Bucket-Besitzer jedes Objekt im Bucket und verwaltet den Zugriff auf Daten ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen.

Object Ownership verfügt über drei Einstellungen, mit denen Sie den Besitz von in Ihren Bucket hochgeladenen Objekten kontrollieren und deaktivieren oder aktivieren könnenACLs:

ACLsdeaktiviert

- Bucket-Besitzer erzwungen (Standard) — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLswirkt sich nicht mehr auf die Berechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet Richtlinien, um die Zugriffssteuerung zu definieren.

ACLsaktiviert

- Bevorzugter Bucket-Besitzer — Der Bucket-Besitzer besitzt und hat die volle Kontrolle über neue Objekte, die andere Accounts in den Bucket mit den gespeicherten Objekten `bucket-owner-full-control` schreibenACL.
- Objekt-Writer — AWS-Konto Derjenige, der ein Objekt hochlädt, besitzt das Objekt, hat die volle Kontrolle darüber und kann anderen Benutzern Zugriff darauf gewähren. ACLs

Voraussetzungen: Bevor Sie die Deaktivierungseinstellung „Bucket Owner erforced“ anwendenACLs, müssen Sie die ACL Bucket-Berechtigungen auf Bucket-Richtlinien migrieren und Ihren Bucket ACLs auf die Standardeinstellung „Privat“ zurücksetzen. ACL Wir empfehlen außerdem, ACL Objektberechtigungen auf Bucket-Richtlinien zu migrieren und Bucket-Richtlinien zu bearbeiten, für

die ACLs außer dem Bucket-Besitzer Vollzugriff ACLs erforderlich ist. Weitere Informationen finden Sie unter Voraussetzungen für die Deaktivierung ACLs.

Berechtigungen: Um diesen Vorgang zu verwenden, müssen Sie die `s3:PutBucketOwnershipControls`-Berechtigung haben. Weitere Informationen zu Amazon S3-Berechtigungen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, auf den Sie eine Einstellung für S3 Object Ownership anwenden möchten.
3. Wählen Sie die Registerkarte Berechtigungen.
4. Wählen Sie unter Object Ownership die Option Edit (Bearbeiten).
5. Wählen Sie unter Object Ownership eine der folgenden Einstellungen aus, um die Inhaberschaft der in Ihren Bucket hochgeladenen Objekte zu deaktivieren oder zu aktivieren ACLs und zu kontrollieren:

   ACLsdeaktiviert

   - Bucket-Besitzer erzwungen — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle über jedes Objekt im Bucket. ACLswirkt sich nicht mehr auf die Berechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet Richtlinien, um die Zugriffssteuerung zu definieren.

     Informationen dazu, wie Sie vorschreiben müssen, dass alle neuen Buckets mit ACLs deaktivierten AWS Organizations Richtlinien erstellt werden, finden Sie unterDeaktivierung ACLs für alle neuen Buckets (Bucket Owner wird erzwungen). IAM

ACLsaktiviert

- Bevorzugter Bucket-Besitzer — Der Bucket-Besitzer besitzt und hat die volle Kontrolle über neue Objekte, die andere Accounts in den Bucket mit den gespeicherten Objekten `bucket-owner-full-control` schreibenACL.

  Wenn Sie die bevorzugte Einstellung des Bucket-Besitzers anwenden, sodass alle Amazon S3 S3-Uploads die gespeicherten Dateien enthalten müssen `bucket-owner-full-control`ACL, können Sie eine Bucket-Richtlinie hinzufügen, die nur Objekt-Uploads zulässt, die dies verwenden. ACL
- Objekt-Writer — Derjenige AWS-Konto , der ein Objekt hochlädt, besitzt das Objekt, hat die volle Kontrolle darüber und kann anderen Benutzern Zugriff darauf gewähren. ACLs

6. Wählen Sie Save (Speichern) aus.

Mit dem AWS CLI

Um eine Einstellungen für Object Ownership für einen vorhandenen Bucket anzuwenden, verwenden Sie den `put-bucket-ownership-controls`-Befehl mit dem `--ownership-controls`-Parameter. Gültige Werte für die Eigentümerschaft sind `BucketOwnerEnforced`, `BucketOwnerPreferred` oder `ObjectWriter`.

In diesem Beispiel wird die Einstellung „Bucket-Eigentümer erzwungen“ für einen vorhandenen Bucket mithilfe der AWS CLI angewendet:

```
aws s3api put-bucket-ownership-controls --bucket amzn-s3-demo-bucket --ownership-
controls="Rules=[{ObjectOwnership=BucketOwnerEnforced}]"
```

Weitere Informationen zu `put-bucket-ownership-controls` finden Sie unter .put-bucket-ownership-controls im AWS Command Line Interface -Benutzerhandbuch.

Verwendung des AWS SDK for Java

In diesem Beispiel gilt die `BucketOwnerEnforced`-Einstellung für Object Ownership für einen vorhandenen Bucket mit AWS SDK for Java:

```
        // Build the ObjectOwnership for BucketOwnerEnforced
        OwnershipControlsRule rule = OwnershipControlsRule.builder()
```

```
                .objectOwnership(ObjectOwnership.BucketOwnerEnforced)
                .build();

        OwnershipControls ownershipControls = OwnershipControls.builder()
                  .rules(rule)
                  .build()

         // Build the PutBucketOwnershipControlsRequest
         PutBucketOwnershipControlsRequest putBucketOwnershipControlsRequest =
               PutBucketOwnershipControlsRequest.builder()
                       .bucket(BUCKET_NAME)
                       .ownershipControls(ownershipControls)
                       .build();

         // Send the request to Amazon S3
         s3client.putBucketOwnershipControls(putBucketOwnershipControlsRequest);
```

Verwenden AWS CloudFormation

Informationen dazu, wie Sie AWS CloudFormation die Einstellung „Objekteigentum“ auf einen vorhandenen Bucket anwenden können, finden Sie unter AWS::S3::Bucket OwnershipControls im AWS CloudFormation -Benutzerhandbuch.

Verwenden von REST API

RESTAPIUm die Einstellung Objektbesitz auf einen vorhandenen S3-Bucket anzuwenden, verwenden Sie`PutBucketOwnershipControls`. Weitere Informationen finden Sie unter PutBucketOwnershipControlsin der Amazon Simple Storage Service API Reference.

Nächste Schritte: Nach der Anwendung der Einstellungen „Von Bucket-Besitzer erzwungen“ oder „Von Bucket-Besitzer bevorzugt“ auf den Objektbesitz können Sie die folgenden Schritte ausführen:

- Bucket-Besitzer erzwungen — Erfordert, dass alle neuen Buckets mit ACLs deaktivierter Einstellung erstellt werden, indem eine Richtlinie IAM oder Organizations verwendet wird.
- Bucket-Besitzer bevorzugt — Fügen Sie eine S3-Bucket-Richtlinie hinzu, die vorschreibt, dass ACL für alle Objekt-Uploads `bucket-owner-full-control` in Ihren Bucket „gesperrt“ gilt.

Anzeigen der Einstellung Object Ownership für einen S3-Bucket

S3 Object Ownership ist eine Einstellung auf Amazon S3-Bucket-Ebene, mit der Sie Zugriffskontrolllisten (ACLs) deaktivieren und die Verantwortung für jedes Objekt in Ihrem Bucket

übernehmen können, wodurch die Zugriffsverwaltung für in Amazon S3 gespeicherte Daten vereinfacht wird. Standardmäßig ist für S3 Object Ownership die erzwungene Einstellung Bucket Owner festgelegt und ACLs für neue Buckets deaktiviert. Bei ACLs deaktivierter Option besitzt der Bucket-Besitzer jedes Objekt im Bucket und verwaltet den Zugriff auf Daten ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen.

Object Ownership verfügt über drei Einstellungen, mit denen Sie den Besitz von in Ihren Bucket hochgeladenen Objekten kontrollieren und deaktivieren oder aktivieren könnenACLs:

ACLsdeaktiviert

- Bucket-Besitzer erzwungen (Standard) — ACLs sind deaktiviert, und der Bucket-Besitzer besitzt automatisch jedes Objekt im Bucket und hat die volle Kontrolle darüber. ACLswirkt sich nicht mehr auf die Berechtigungen für Daten im S3-Bucket aus. Der Bucket verwendet Richtlinien, um die Zugriffssteuerung zu definieren.

ACLsaktiviert

- Bevorzugter Bucket-Besitzer — Der Bucket-Besitzer besitzt und hat die volle Kontrolle über neue Objekte, die andere Accounts in den Bucket mit den gespeicherten Objekten `bucket-owner-full-control` schreibenACL.
- Objekt-Writer — AWS-Konto Derjenige, der ein Objekt hochlädt, besitzt das Objekt, hat die volle Kontrolle darüber und kann anderen Benutzern Zugriff darauf gewähren. ACLs

Sie können die S3 Object Ownership-Einstellungen für einen Amazon-S3-Bucket anzeigen. Informationen zum Festlegen von Object Ownership für einen neuen Bucket finden Sie unter Festlegen von Object Ownership beim Erstellen eines Buckets. Informationen zum Festlegen von Object Ownership für einen vorhandenen Bucket finden Sie unter Einstellung für Object Ownership für einen vorhandenen Bucket.

Berechtigungen Um diesen Vorgang verwenden zu können, müssen Sie die `s3:GetBucketOwnershipControls`-Berechtigung haben. Weitere Informationen zu Amazon S3-Berechtigungen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, auf den Sie eine Einstellung für Object Ownership anwenden möchten.
3. Wählen Sie die Registerkarte Berechtigungen.
4. Unter Object Ownership können Sie die Objekteigentümerschafts-Einstellungen für Ihren Bucket anzeigen.

Verwenden Sie den AWS CLI

Um die Einstellung S3-Objekteigentum für einen S3-Bucket abzurufen, verwenden Sie get-bucket-ownership-controls AWS CLI Befehl.

```
aws s3api get-bucket-ownership-controls --bucket amzn-s3-demo-bucket
```

Mit dem REST API

Verwenden Sie den `GetBucketOwnershipControls` API Vorgang, um die Einstellung Object Ownership für einen S3-Bucket abzurufen. Weitere Informationen finden Sie unter GetBucketOwnershipControls.

Deaktivierung ACLs für alle neuen Buckets und Durchsetzung des Objektbesitzes

Wir empfehlen, dass Sie die Option in Ihren ACLs Amazon S3 S3-Buckets deaktivieren. Wenden Sie dazu die Einstellung „Bucket-Eigentümer erzwungen“ für S3 Object Ownership an. Wenn Sie diese Einstellung anwenden, ACLs sind sie deaktiviert und Sie besitzen automatisch alle Objekte in Ihrem Bucket und haben die volle Kontrolle über sie. Um zu verlangen, dass alle neuen Buckets mit ACLs deaktivierter Option erstellt werden, verwenden Sie AWS Identity and Access Management (IAM) - Richtlinien oder AWS Organizations Dienststeuerungsrichtlinien (SCPs), wie im nächsten Abschnitt beschrieben.

Um den Objekteigentum für neue Objekte ohne Deaktivierung durchzusetzenACLs, können Sie die bevorzugte Einstellung des Bucket-Besitzers anwenden. Wenn Sie diese Einstellung anwenden, empfehlen wir Ihnen dringend, Ihre Bucket-Richtlinie dahingehend zu aktualisieren, dass ACL für

alle PUT Anfragen an Ihren Bucket „gesperrt“ gilt. `bucket-owner-full-control` Stellen Sie sicher, dass Sie auch Ihre Kunden so aktualisieren, dass sie die `bucket-owner-full-control` gescannten Dateien von anderen Konten ACL an Ihren Bucket senden.

Themen

- Deaktivierung ACLs für alle neuen Buckets (Bucket Owner wird erzwungen)
- Die bucket-owner-full-control ACL für Amazon S3 gespeicherten PUT Operationen werden benötigt (Bucket-Besitzer bevorzugt)

Deaktivierung ACLs für alle neuen Buckets (Bucket Owner wird erzwungen)

Die folgende IAM Beispielrichtlinie verweigert einem bestimmten IAM Benutzer oder einer bestimmten Rolle die `s3:CreateBucket` Berechtigung, sofern nicht die erzwungene Einstellung des Bucket-Besitzers für Objekteigentümer angewendet wird. Das Schlüssel-Wert-Paar im `Condition`-Block gibt `s3:x-amz-object-ownership` als Schlüssel und die `BucketOwnerEnforced`-Einstellung als seinen Wert an. Mit anderen Worten, der IAM Benutzer kann Buckets nur erstellen, wenn er die Einstellung „Bucket Owner erforced“ für „Objekteigentum“ aktiviert und deaktiviert hat. ACLs Sie können diese Richtlinie auch als Grenze SCP für Ihre AWS Organisation verwenden.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RequireBucketOwnerFullControl",
            "Action": "s3:CreateBucket",
            "Effect": "Deny",
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "s3:x-amz-object-ownership": "BucketOwnerEnforced"
                }
            }
        }
    ]
}
```

Die bucket-owner-full-control ACL für Amazon S3 gespeicherten **PUT** Operationen werden benötigt (Bucket-Besitzer bevorzugt)

Mit der bevorzugten Einstellung „Bucket Ownership“ für Object Ownership besitzen Sie als Bucket-Besitzer neue Objekte, die andere Konten zusammen mit den gespeicherten Objekten in Ihren Bucket schreiben, und haben die volle Kontrolle `bucket-owner-full-control` darüberACL. Wenn jedoch andere Konten Objekte in Ihren Bucket schreiben, ohne dass die gespeicherten Objekte gespeichert sind `bucket-owner-full-control`ACL, behält der Object Writer die volle Kontrolle über den Zugriff. Sie als Besitzer des Buckets können eine Bucket-Richtlinie implementieren, die Schreibvorgänge nur dann erlaubt, wenn der Bucket-Server spezifiziert `bucket-owner-full-control` wirdACL.

> **Note**
>
> Wenn Sie diese ACLs Option mit der Einstellung Bucket Owner erforced deaktiviert haben, besitzen Sie als Bucket-Besitzer automatisch alle Objekte in Ihrem Bucket und haben die volle Kontrolle über sie. Sie müssen diesen Abschnitt nicht verwenden, um Ihre Bucket-Richtlinie zu aktualisieren, um den Objekteigentümer für den Bucket-Eigentümer durchzusetzen.

Die folgende Bucket-Richtlinie spezifiziert dieses Konto *111122223333* kann Objekte hochladen *amzn-s3-demo-bucket* nur wenn das Objekt auf gesetzt ACL ist`bucket-owner-full-control`. Achten Sie darauf, zu ersetzen *111122223333* mit deinem Konto und *amzn-s3-demo-bucket* mit dem Namen Ihres Buckets.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "Only allow writes to my bucket with bucket owner full control",
         "Effect": "Allow",
         "Principal": {
            "AWS": [
               "arn:aws:iam::111122223333:user/ExampleUser"
            ]
         },
         "Action": [
            "s3:PutObject"
```

```
         ],
         "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
         "Condition": {
            "StringEquals": {
               "s3:x-amz-acl": "bucket-owner-full-control"
            }
         }
      }
   ]
}
```

Im Folgenden finden Sie ein Beispiel für einen Kopiervorgang, der `bucket-owner-full-control` mithilfe ACL von AWS Command Line Interface (AWS CLI) gespeicherte Dateien einschließt.

```
aws s3 cp file.txt s3://amzn-s3-demo-bucket --acl bucket-owner-full-control
```

Wenn die Bucket-Richtlinie in Kraft gesetzt wurde und der Client die gespeicherten Dateien nicht einschließt `bucket-owner-full-control`ACL, schlägt der Vorgang fehl und der Uploader erhält die folgende Fehlermeldung:

Beim Aufrufen des PutObject Vorgangs ist ein Fehler aufgetreten (AccessDenied): Zugriff verweigert.

> **Note**
>
> Wenn Clients nach dem Hochladen Zugriff auf Objekte benötigen, müssen Sie dem hochladenden Konto zusätzliche Berechtigungen erteilen. Informationen zum Erteilen von Zugriff auf Ihre Ressourcen für Konten finden Sie unter Exemplarische Vorgehensweisen, die Richtlinien verwenden, um den Zugriff auf Ihre Amazon S3 S3-Ressourcen zu verwalten.

Fehlerbehebung

Wenn Sie die erzwungene Einstellung des Bucket-Besitzers für S3 Object Ownership anwenden, sind die Zugriffskontrolllisten (ACLs) deaktiviert und Sie als Bucket-Besitzer sind automatisch Eigentümer aller Objekte in Ihrem Bucket. ACLswirkt sich nicht mehr auf die Berechtigungen für die Objekte in Ihrem Bucket aus. Sie können Richtlinien verwenden, um Berechtigungen zu erteilen. Bei allen PUT S3-Anfragen müssen entweder die `bucket-owner-full-control` gespeicherten ACL Daten angegeben werden oder nicht. Andernfalls schlagen diese Anfragen fehl. ACL Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

Wenn ein ungültiger Wert angegeben ACL ist oder ACL Bucket-Berechtigungen Zugriff außerhalb Ihrer Rechte gewähren AWS-Konto, werden möglicherweise die folgenden Fehlermeldungen angezeigt.

AccessControlListNotSupported

Nachdem Sie die erzwungene Einstellung „Bucket-Besitzer“ für „Objekteigentum“ angewendet haben, ACLs sind sie deaktiviert. Anfragen zum Setzen ACLs oder Aktualisieren ACLs schlagen mit einem `400` Fehler fehl und geben den AccessControlListNotSupportedFehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt. Leseanfragen geben ACLs immer eine Antwort zurück, aus der hervorgeht, dass der Bucket-Besitzer die volle Kontrolle hat. In Ihren PUT Vorgängen müssen Sie entweder den Bucket-Besitzer mit Vollzugriff angeben ACLs oder keinenACL. Andernfalls schlagen Ihre PUT-Operationen fehl.

Der folgende `put-object` AWS CLI Beispielbefehl beinhaltet den Befehl `public-read` cannedACL.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key object-key-name --body doc-example-body --acl public-read
```

Wenn der Bucket die Einstellung Bucket owner erforced zur Deaktivierung verwendetACLs, schlägt dieser Vorgang fehl und der Uploader erhält die folgende Fehlermeldung:

Beim Aufrufen der PutObject Operation ist ein Fehler aufgetreten (AccessControlListNotSupported): Der Bucket erlaubt es nicht ACLs

InvalidBucketAclWithObjectOwnership

Wenn Sie die Einstellung Bucket Owner erforced auf Deaktivierung anwenden möchtenACLs, ACL darf Ihr Bucket nur dem Bucket-Besitzer die volle Kontrolle geben. Ihr Bucket ACL kann keiner externen AWS-Konto oder einer anderen Gruppe Zugriff gewähren. Wenn Ihre `CreateBucket` Anfrage beispielsweise den Bucket owner enforced festlegt und einen Bucket ACL angibt, der Zugriff auf einen externen Bucket gewährt AWS-Konto, schlägt Ihre Anfrage mit einem `400` Fehler fehl und gibt den InvalidBucketAclWithObjectOwnershipFehlercode zurück. Ebenso schlägt die Anfrage fehl, wenn Sie in Ihrer `PutBucketOwnershipControls` Anfrage festlegen, dass der Bucket Owner für einen Bucket durchgesetzt wirdACL, der über einen Bucket verfügt, der anderen Berechtigungen gewährt.

Example : Ein vorhandener Bucket ACL gewährt öffentlichen Lesezugriff

Wenn ein vorhandener Bucket beispielsweise öffentlichen Lesezugriff ACL gewährt, können Sie die vom Bucket-Besitzer erzwungene Einstellung für Objekteigentum erst anwenden, wenn Sie diese ACL Berechtigungen in eine Bucket-Richtlinie migrieren und Ihren Bucket ACL auf die Standardeinstellung privat ACL zurücksetzen. Weitere Informationen finden Sie unter Voraussetzungen für die Deaktivierung ACLs.

Dieser Beispiel-Bucket ACL gewährt öffentlichen Lesezugriff:

```
{
    "Owner": {
        "ID": "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID"
    },
    "Grants": [
        {
            "Grantee": {
                "ID":
 "852b113e7a2f25102679df27bb0ae12b3f85be6BucketOwnerCanonicalUserID",
                "Type": "CanonicalUser"
            },
            "Permission": "FULL_CONTROL"
        },
        {
            "Grantee": {
                "Type": "Group",
                "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
            },
            "Permission": "READ"
        }
    ]
}
```

Der folgende `put-bucket-ownership-controls` AWS CLI Beispielbefehl wendet die erzwungene Einstellung Bucket owner für Object Ownership an:

```
aws s3api put-bucket-ownership-controls --bucket amzn-s3-demo-bucket --ownership-
controls Rules=[{ObjectOwnership=BucketOwnerEnforced}]
```

Da der Bucket öffentlichen Lesezugriff ACL gewährt, schlägt die Anfrage fehl und gibt den folgenden Fehlercode zurück:

Beim Aufrufen der PutBucketOwnershipControls Operation ist ein Fehler aufgetreten (InvalidBucketAclWithObjectOwnership): Bucket kann nicht mit ObjectOwnership der BucketOwnerEnforced Einstellung ACLs gesetzt worden sein

Sicherheit in Amazon S3

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einem Rechenzentrum und einer Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

Sicherheit der Cloud

AWS ist verantwortlich für den Schutz der Infrastruktur, auf der AWS Dienste in der ausgeführt AWS Cloud werden. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Die Wirksamkeit unserer Sicherheitsfunktionen wird regelmäßig von externen Prüfern im Rahmen des AWS - Compliance-Programms getestet und überprüft. Weitere Informationen zu den für Amazon S3 geltenden Compliance-Programmen finden Sie unter AWS Im Rahmen des Compliance-Programms zugelassene -Services.

Sicherheit in der Cloud

Ihre Verantwortung richtet sich nach dem AWS Dienst, den Sie nutzen. In Ihre Verantwortung fallen außerdem weitere Faktoren, wie z. B. die Vertraulichkeit der Daten, die Anforderungen Ihrer Organisation sowie geltende Gesetze und Vorschriften. Für Amazon S3 umfasst Ihre Verantwortlichkeit die folgenden Bereiche:

- Verwalten Ihrer Daten, einschließlich Objekt-Eigentumsrechte und Verschlüsselung.
- Klassifizierung Ihres Vermögens.
- Verwalten Sie den Zugriff auf Ihre Daten mithilfe von IAMRollen und anderen Dienstkonfigurationen, um die entsprechenden Berechtigungen anzuwenden.
- Aktivierung von Detektivkontrollen wie AWS CloudTrailAmazon GuardDuty für Amazon S3.

Diese Dokumentation beschreibt, wie Sie das Modell der übergreifenden Verantwortlichkeit bei der Verwendung von Amazon S3 anwenden können. Die folgenden Themen veranschaulichen, wie Sie Amazon S3 zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele konfigurieren können. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, mit denen Sie Ihre Amazon S3 S3-Ressourcen überwachen und sichern können.

> **Note**
>
> Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter S3 Express One Zone und Arbeiten mit Verzeichnis-Buckets.

Themen

- Bewährte Methoden für die Sicherheit in Amazon S3
- Datenschutz in Amazon S3
- Datenschutz durch Verschlüsselung
- Richtlinie für den Datenverkehr zwischen Netzwerken
- Konformitätsvalidierung für Amazon S3
- Ausfallsicherheit bei Amazon S3
- Infrastruktursicherheit in Amazon S3
- Konfigurations- und Schwachstellenanalyse in Amazon S3
- Zugriffsverwaltung

Bewährte Methoden für die Sicherheit in Amazon S3

Amazon S3 enthält eine Reihe von Sicherheitsfunktionen, die Sie bei der Entwicklung und Implementierung Ihrer eigenen Sicherheitsrichtlinien berücksichtigen sollten. Die folgenden bewährten Methoden stellen allgemeine Richtlinien und keine vollständige Sicherheitslösung dar. Da diese bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Empfehlungen und nicht als bindend ansehen.

Themen

- Bewährte Methoden für die Sicherheit in Amazon S3
- Bewährte Methoden zur Überwachung und Prüfung von Amazon S3
- Überwachung der Datensicherheit mit verwalteten AWS Sicherheitsdiensten

Bewährte Methoden für die Sicherheit in Amazon S3

Die folgenden bewährten Methoden für Amazon S3 können dazu beitragen, Sicherheitsvorfälle zu verhindern.

Deaktivieren Sie die Zugriffskontrolllisten (ACLs)

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie den Besitz von in Ihren Bucket hochgeladenen Objekten kontrollieren und deaktivieren oder aktivieren können. ACLs Standardmäßig ist für Object Ownership die Einstellung Bucket Owner erforced festgelegt, und alle sind ACLs deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff auf Daten ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.

In den meisten modernen Anwendungsfällen in Amazon S3 ist die Verwendung von Zugriffskontrolllisten (ACLs) nicht mehr erforderlich. Wir empfehlen die DeaktivierungACLs, außer in ungewöhnlichen Fällen, in denen Sie den Zugriff für jedes Objekt einzeln kontrollieren müssen. Um jedes Objekt in Ihrem Bucket zu deaktivieren ACLs und die Eigentümerschaft dafür zu übernehmen, wenden Sie für S3 Object Ownership die Einstellung Bucket Ownerforced an. Wenn Sie die Option deaktivierenACLs, können Sie ganz einfach einen Bucket mit Objekten verwalten, die von verschiedenen AWS-Konten Personen hochgeladen wurden.

Wann deaktiviert ACLs sind, basiert die Zugriffskontrolle für Ihre Daten auf Richtlinien wie den folgenden:

- AWS Identity and Access Management (IAM) Benutzerrichtlinien
- S3-Bucket-Richtlinien
- Endpunktrichtlinien für virtuelle private Clouds (VPC)
- AWS Organizations Richtlinien zur Dienststeuerung (SCPs)
- AWS Organizations Richtlinien zur Ressourcenkontrolle (RCPs)

Die Deaktivierung ACLs vereinfacht die Verwaltung und Prüfung von Berechtigungen. ACLssind für neue Buckets standardmäßig deaktiviert. Sie können sie auch ACLs für bestehende Buckets deaktivieren. Wenn Sie über einen vorhandenen Bucket verfügen, der bereits Objekte enthält, ACLs sind das Objekt und der Bucket nach der Deaktivierung ACLs nicht mehr Teil des Zugriffsevaluierungsprozesses. Stattdessen wird der Zugriff auf der Grundlage von Richtlinien gewährt oder verweigert.

Stellen Sie vor der Deaktivierung sicherACLs, dass Sie Folgendes tun:

- Überprüfen Sie Ihre Bucket-Richtlinie, um sicherzustellen, dass sie alle Möglichkeiten abdeckt, wie Sie außerhalb Ihres Kontos Zugriff auf Ihren Bucket gewähren möchten.
- Setzen Sie Ihren Bucket ACL auf die Standardeinstellung zurück (volle Kontrolle hat der Bucket-Besitzer).

Nach der Deaktivierung ACLs treten die folgenden Verhaltensweisen auf:

- Ihr Bucket akzeptiert nur PUT Anfragen, die keine PUT Oder-Anfragen angeben, für die der Bucket-Besitzer die volle Kontrolle hatACLs. ACL Dazu ACLs gehören die `bucket-owner-full-control` vorgefertigten ACL oder gleichwertigen Formen davonACL, die in ausgedrückt werdenXML.
- Bestehende Anwendungen, die die volle Kontrolle über ACLs den Bucket Owner unterstützen, haben keine Auswirkungen.
- PUTAnfragen, die andere ACLs (z. B. benutzerdefinierte Zuschüsse für bestimmte AWS-Konten) enthalten, schlagen fehl und geben einen HTTP Statuscode `400 (Bad Request)` mit dem Fehlercode zurück`AccessControlListNotSupported`.

Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

Achten Sie darauf, dass für Ihre Amazon-S3-Buckets die korrekten Richtlinien gelten und dass sie nicht öffentlich zugänglich sind.

Sofern es nicht unbedingt notwendig ist, dass jemand Ihren S3-Bucket im Internet lesen oder darin schreiben kann, stellen Sie sicher, dass der S3-Bucket nicht öffentlich ist. Im Folgenden werden einige Maßnahmen zum Blockieren des öffentlichen Zugriffs aufgeführt:

- Verwenden Sie S3 Block Public Access. Mit S3 Block Public Access können Sie problemlos zentrale Kontrollen zur Beschränkung des öffentlichen Zugriffs auf Ihre Amazon-S3-Ressourcen einrichten. Diese zentralen Kontrollen werden unabhängig davon durchgesetzt, wie die Ressourcen erstellt werden. Weitere Informationen finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.
- Identifizieren Sie Amazon-S3-Bucket-Richtlinien, die eine Platzhalteridentität zulassen, z. B. `"Principal": "*"` (was praktisch „jeder“ bedeutet). Halten Sie auch nach Richtlinien Ausschau, die die Platzhalteraktion `"*"` zulassen (was einem Benutzer erlaubt, jede Aktion im Amazon-S3-Bucket durchzuführen).
- Suchen Sie in ähnlicher Weise nach Amazon S3 S3-Bucket-Zugriffskontrolllisten (ACLs), die Lese-, Schreib- oder Vollzugriff für „Jeder“ oder „Jeder authentifizierte AWS Benutzer“ gewähren.
- Verwenden Sie den `ListBuckets` API Vorgang, um alle Ihre Amazon S3 S3-Buckets zu scannen. Ermitteln Sie dann mit `GetBucketAcl`, `GetBucketWebsite` und `GetBucketPolicy`, ob der Bucket über richtlinienkonforme Zugriffskontrollen und Konfiguration verfügt.
- Verwenden Sie AWS Trusted Advisor diese Option, um Ihre Amazon-S3-Implementierung zu überprüfen.
- Erwägen Sie die Implementierung laufender Detektivkontrollen mithilfe der s3-bucket-public-read-prohibited und s3-bucket-public-write-prohibitedverwaltet AWS-Config-Regeln.

Weitere Informationen finden Sie unter Identity and Access Management für Amazon S3.

Implementieren des Zugriffs mit geringsten Berechtigungen

Bei der Vergabe von Berechtigungen entscheiden Sie, wer welche Berechtigungen für welche Amazon-S3-Ressourcen erhält. Sie aktivieren die spezifischen Aktionen, die daraufhin für die betreffenden Ressourcen erlaubt sein sollen. Aus diesem Grund sollten Sie nur Berechtigungen gewähren, die zum Ausführen einer Aufgabe erforderlich sind. Die Implementierung der geringstmöglichen Zugriffsrechte ist eine grundlegende Voraussetzung zum Reduzieren des

Sicherheitsrisikos und der Auswirkungen, die aufgrund von Fehlern oder böswilligen Absichten entstehen könnten.

Die folgenden Tools stehen zur Implementierung der geringstmöglichen Zugriffsrechte zur Verfügung:

- Politische Maßnahmen für Amazon S3und Berechtigungsgrenzen für IAM Entitäten
- So funktioniert Amazon S3 mit IAM
- Übersicht über die Zugriffskontrollliste (ACL)

Hilfreiche Informationen zur Entscheidung, welche der vorherigen Mechanismen Sie auswählen sollten, finden Sie unter Identity and Access Management für Amazon S3.

IAMRollen für Anwendungen verwenden, für AWS-Services die Amazon S3 S3-Zugriff erforderlich ist

Damit Anwendungen, die auf Amazon EC2 oder anderen Plattformen ausgeführt werden, AWS-Services auf Amazon S3 S3-Ressourcen zugreifen können, müssen sie gültige AWS Anmeldeinformationen in ihren AWS API Anfragen angeben. Wir empfehlen, AWS Anmeldeinformationen nicht direkt in der Anwendung oder EC2 Amazon-Instance zu speichern. Dabei handelt es sich um langfristige Anmeldeinformationen, die nicht automatisch rotiert werden und bedeutende geschäftliche Auswirkungen haben könnten, wenn sie kompromittiert werden.

Verwenden Sie stattdessen eine IAM Rolle, um temporäre Anmeldeinformationen für Anwendungen oder Dienste zu verwalten, die auf Amazon S3 zugreifen müssen. Wenn Sie eine Rolle verwenden, müssen Sie keine langfristigen Anmeldeinformationen (wie einen Benutzernamen und ein Passwort oder Zugriffsschlüssel) an eine EC2 Amazon-Instance verteilen oder AWS-Service, wie AWS Lambda. Die Rolle stellt temporäre Berechtigungen bereit, die Anwendungen verwenden können, wenn sie andere AWS Ressourcen aufrufen.

Weitere Informationen finden Sie in den folgenden Themen im IAMBenutzerhandbuch:

- IAMRollen
- Gängige Szenarien für Rollen: Benutzer, Anwendungen und Services

Erwägen der Verschlüsselung von Data-at-Rest

Es gibt die folgenden Optionen, Daten im Ruhezustand in Amazon S3 zu schützen:

- Serverseitige Verschlüsselung — Für alle Amazon S3 S3-Buckets ist standardmäßig eine Verschlüsselung konfiguriert, und alle neuen Objekte, die in einen S3-Bucket hochgeladen werden, werden im Ruhezustand automatisch verschlüsselt. Die

serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Standardverschlüsselungskonfiguration für jeden Bucket in Amazon S3. Um einen anderen Verschlüsselungstyp zu verwenden, können Sie entweder die Art der serverseitigen Verschlüsselung angeben, die in Ihren S3-PUT-Anfragen verwendet werden soll, oder Sie können die Standardverschlüsselungskonfiguration im Ziel-Bucket festlegen.

Amazon S3 bietet auch diese serverseitigen Verschlüsselungsoptionen:

- Serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln (SSE-) KMS
- Zweischichtige serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln (-) DSSE KMS
- Serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE

Weitere Informationen finden Sie unter Schützen von Daten mit serverseitiger Verschlüsselung.

- Clientseitige Verschlüsselung – Sie können Daten clientseitig verschlüsseln und die verschlüsselten Daten auf Amazon S3 hochladen. In diesem Fall verwalten Sie den Verschlüsselungsprozess, die Verschlüsselungsschlüssel und die zugehörigen Tools. Genau wie die serverseitige Verschlüsselung kann die client-seitige Verschlüsselung das Risiko reduzieren, indem die Daten mit einem Schlüssel verschlüsselt werden, der durch einen anderen Mechanismus gespeichert wird als der, der die Daten selbst speichert

  Amazon S3 bietet mehrere clientseitige Verschlüsselungsoptionen. Weitere Informationen finden Sie unter Schützen von Daten mithilfe der clientseitigen Verschlüsselung.

Erzwingen der Verschlüsselung von Daten während der Übertragung

Sie können HTTPS (TLS) verwenden, um zu verhindern, dass potenzielle Angreifer den Netzwerkverkehr mit oder ähnlichen Angriffen abhören oder manipulieren. person-in-the-middle Wir empfehlen, nur verschlüsselte Verbindungen über HTTPS () TLS zuzulassen, indem Sie den aws:SecureTransportBedingung in Ihren Amazon S3 S3-Bucket-Richtlinien.

> **Important**
>
> Wir empfehlen, dass Ihre Anwendung keine Amazon S3 TLS S3-Zertifikate anheftet, da AWS das Anheften von öffentlich vertrauenswürdigen Zertifikaten nicht unterstützt wird. S3 erneuert Zertifikate automatisch und die Verlängerung kann jederzeit vor Ablauf des Zertifikats erfolgen. Durch die Erneuerung eines Zertifikats wird ein neues öffentlich-privates key pair generiert. Wenn Sie ein S3-Zertifikat angeheftet haben, das kürzlich

mit einem neuen öffentlichen Schlüssel erneuert wurde, können Sie erst dann eine Verbindung zu S3 herstellen, wenn Ihre Anwendung das neue Zertifikat verwendet.

Erwägen Sie auch die Implementierung fortlaufender Erkennungskontrollen mithilfe der s3-bucket-ssl-requests-onlyverwaltete AWS Config Regel.

Erwägen der S3-Objektsperre

Mit S3 Object Lock können Sie Objekte mithilfe eines „Write Once Read Many“ (WORM) -Modells speichern. Die S3-Objektsperre kann unbeabsichtigtes oder unsachgemäßes Löschen von Daten verhindern. Beispielsweise können Sie S3 Object Lock verwenden, um Ihre AWS CloudTrail Protokolle zu schützen.

Weitere Informationen finden Sie unter Sperren von Objekten mit Object Lock.

Aktivieren der S3-Versionsverwaltung

Die S3-Versionsverwaltung ermöglicht Ihnen, mehrere Versionen eines Objekts im selben Bucket aufzubewahren. Sie können Versioning verwenden, um sämtliche Versionen aller Objekte in Ihrem Bucket zu speichern, abzurufen oder wiederherzustellen. Daten lassen sich dank Versioning nach unbeabsichtigten Benutzeraktionen und Anwendungsfehlern leicht wiederherstellen.

Erwägen Sie auch die Implementierung fortlaufender Erkennungskontrollen mithilfe der s3-bucket-versioning-enabledverwaltete AWS Config Regel.

Weitere Informationen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

Erwägen einer regionsübergreifenden S3-Replikation

Auch wenn Amazon S3 Ihre Daten standardmäßig in mehreren entfernten Availability Zones speichert, machen es die Compliance-Anforderungen möglicherweise erforderlich, Daten in noch größeren Entfernungen zu speichern. Mit der regionsübergreifenden S3-Replikation (CRR) können Sie Daten zwischen entfernten Standorten replizieren AWS-Regionen , um diese Anforderungen zu erfüllen. CRRermöglicht das automatische, asynchrone Kopieren von Objekten zwischen verschiedenen Buckets. AWS-Regionen Weitere Informationen finden Sie unter Objekte innerhalb und zwischen Regionen replizieren.

> **Note**
>
> CRRerfordert, dass sowohl für den S3-Quell- als auch für den Ziel-S3-Bucket die Versionierung aktiviert ist.

Erwägen Sie auch die Implementierung fortlaufender Erkennungskontrollen mithilfe von s3-bucket-replication-enabledverwaltete AWS Config Regel.

Erwägen Sie die Verwendung von VPC Endpunkten für den Zugriff auf Amazon S3

Ein Virtual Private Cloud (VPC) -Endpunkt für Amazon S3 ist eine logische Einheit innerhalb einerVPC, die nur Konnektivität zu Amazon S3 ermöglicht. VPCEndpunkte können dazu beitragen, zu verhindern, dass Datenverkehr das offene Internet durchquert.

VPCEndpunkte für Amazon S3 bieten mehrere Möglichkeiten, den Zugriff auf Ihre Amazon S3 S3-Daten zu kontrollieren:

- Mithilfe von S3-Bucket-Richtlinien können Sie die Anfragen, Benutzer oder Gruppen steuern, die über einen bestimmten VPC Endpunkt zugelassen werden.
- Mithilfe von S3-Bucket-Richtlinien können Sie steuern, welche VPCs oder welche VPC Endpunkte Zugriff auf Ihre S3-Buckets haben.
- Sie können dazu beitragen, Datenexfiltration zu verhindern, indem Sie ein Gerät verwendenVPC, das kein Internet-Gateway hat.

Weitere Informationen finden Sie unter Steuern des Zugriffs von VPC Endpunkten aus mit Bucket-Richtlinien.

Verwenden Sie verwaltete AWS Sicherheitsdienste, um die Datensicherheit zu überwachen

Verschiedene verwaltete AWS Sicherheitsservices können Ihnen dabei helfen, Sicherheits- und Compliance-Risiken für Ihre Amazon S3 S3-Daten zu identifizieren, zu bewerten und zu überwachen. Diese Services können Ihnen auch dabei helfen, Ihre Daten vor diesen Risiken zu schützen. Diese Services umfassen automatisierte Erkennungs-, Überwachungs- und Schutzfunktionen, die so konzipiert sind, dass sie von Amazon S3 S3-Ressourcen für einzelne AWS-Konto bis hin zu Ressourcen für Organisationen mit Tausenden von Konten skaliert werden können.

Weitere Informationen finden Sie unter Überwachung der Datensicherheit mit verwalteten AWS Sicherheitsdiensten.

Bewährte Methoden zur Überwachung und Prüfung von Amazon S3

Die folgenden bewährten Methoden für Amazon S3 können dabei helfen, potenzielle Sicherheitsschwächen und Vorfälle zu erkennen.

Identifizieren und Prüfen aller Ihrer Amazon-S3-Buckets

Die Identifikation Ihrer IT-Assets ist ein wichtiger Aspekt von Governance und Sicherheit. Es ist erforderlich, dass Sie alle Ihre Amazon-S3-Ressourcen überblicken, um ihren Sicherheitsstatus beurteilen und Maßnahmen gegen potenzielle Schwachstellen ergreifen zu können. Zum Prüfen Ihrer Ressourcen empfehlen wir Folgendes:

- Verwenden Sie den Tag-Editor, um sicherheits- und prüfungsrelevante Ressourcen zu identifizieren und zu markieren. Nutzen Sie dann diese Markierungen für die Suche nach den entsprechenden Ressourcen. Weitere Informationen finden Sie unter Suchen nach zu taggenden Ressourcen im Tagging AWS Resources User Guide.
- Verwenden Sie S3 Inventory für die Prüfung und Meldung des Replikations- und Verschlüsselungsstatus Ihrer Objekte für Unternehmens-, Compliance- und regulatorische Anforderungen. Weitere Informationen finden Sie unter Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory.
- Erstellen Sie Ressourcengruppen für Ihre Amazon-S3-Ressourcen. Weitere Informationen finden Sie unter Was sind Ressourcengruppen? im AWS Resource Groups -Benutzerhandbuch.

Implementieren Sie die Überwachung mithilfe von AWS Überwachungstools

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Sicherheit, Verfügbarkeit und Leistung von Amazon S3 und Ihren AWS Lösungen. AWS bietet verschiedene Tools und Dienste, mit denen Sie Amazon S3 und Ihre anderen überwachen können AWS-Services. Sie können beispielsweise CloudWatch Amazon-Metriken für Amazon S3 überwachen, insbesondere die `DeleteRequests` Kennzahlen `PutRequests` `GetRequests4xxErrors`,, und. Weitere Informationen erhalten Sie unter Metriken mit Amazon überwachen CloudWatch und Protokollierung und Überwachung in Amazon S3.

Ein zweites Beispiel finden Sie unter Beispiel: Amazon-S3-Bucket-Aktivität. In diesem Beispiel wird beschrieben, wie ein CloudWatch Alarm erstellt wird, der ausgelöst wird, wenn ein Amazon S3 API S3-Aufruf an `PUT` `DELETE` eine Bucket-Richtlinie, einen Bucket-Lebenszyklus oder eine Bucket-Replikationskonfiguration oder an `PUT` einen Bucket erfolgtACL.

Aktivieren Sie die Amazon-S3-Server-Zugriffsprotokollierung

Die Server-Zugriffsprotokollierung bietet detaillierte Aufzeichnungen über die Anfragen, die an einen Bucket gestellt wurden. Server-Zugriffsprotokolle können Sie bei Sicherheits- und Zugriffsprüfungen unterstützen; sie geben Informationen über Ihren Kundenstamm und helfen beim Verständnis der Amazon-S3-Rechnung. Weitere Informationen zur Aktivierung der Server-Zugriffsprotokollierung finden Sie unter Protokollieren von Anfragen mit Server-Zugriffsprotokollierung.

Erwägen Sie auch die Implementierung fortlaufender detektiver Kontrollen mithilfe von s3-bucket-logging-enabled AWS Config verwaltete Regel.

Verwenden Sie AWS CloudTrail

AWS CloudTrail bietet eine Aufzeichnung der Aktionen, die von einem Benutzer, einer Rolle oder einem AWS-Service in Amazon S3 ausgeführt wurden. Sie können die von gesammelten Informationen verwenden CloudTrail , um Folgendes zu ermitteln:

- Die Anforderung, die an Amazon S3 gestellt wurde
- Die IP-Adresse, von der die Anforderung erfolgt ist
- Wer die Anforderung gestellt hat
- Wann die Anforderung gestellt wurde
- Zusätzliche Details zur Anforderung

Sie können beispielsweise CloudTrail Einträge für PUT Aktionen identifizieren, die sich auf den Datenzugriff auswirken`PutBucketAcl`, insbesondere `PutObjectAcl``PutBucketPolicy`, und`PutBucketWebsite`.

Wenn Sie Ihre einrichten AWS-Konto, CloudTrail ist standardmäßig aktiviert. Sie können die letzten Ereignisse in der CloudTrail Konsole einsehen. Um eine fortlaufende Aufzeichnung der Aktivitäten und Ereignisse für Ihre Amazon S3 S3-Buckets zu erstellen, können Sie in der CloudTrail Konsole einen Trail erstellen. Weitere Informationen finden Sie unter Protokollieren von Datenereignissen im Benutzerhandbuch für AWS CloudTrail .

Wenn Sie einen Trail erstellen, können Sie ihn so konfigurieren, CloudTrail dass Datenereignisse protokolliert werden. Datenereignisse sind Datensätze von Ressourcen-Vorgänge, die auf oder innerhalb einer Ressource ausgeführt werden. In Amazon S3 zeichnen Datenereignisse API Aktivitäten auf Objektebene für einzelne Buckets auf. CloudTrail unterstützt eine Teilmenge

von Amazon S3 API S3-Operationen auf Objektebene, wie`GetObject`, und`DeleteObject`. `PutObject` Weitere Informationen zur CloudTrail Funktionsweise mit Amazon S3 finden Sie unterProtokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail. In der Amazon-S3-Konsole können Sie Ihre S3-Buckets auch dafür konfigurieren, Aktivieren der CloudTrail Ereignisprotokollierung für S3-Buckets und -Objekte.

AWS Config stellt eine verwaltete Regel (`cloudtrail-s3-dataevents-enabled`) bereit, mit der Sie bestätigen können, dass mindestens ein CloudTrail Trail Datenereignisse für Ihre S3-Buckets protokolliert. Weitere Informationen finden Sie unter cloudtrail-s3-dataevents-enabled im Entwicklerhandbuch für AWS Config .

Aktivieren AWS Config

Einige der in diesem Thema aufgeführten bewährten Methoden schlagen vor, AWS Config Regeln zu erstellen. AWS Config hilft Ihnen dabei, die Konfigurationen Ihrer AWS Ressourcen zu beurteilen, zu prüfen und zu bewerten. AWS Config überwacht die Ressourcenkonfigurationen, sodass Sie die aufgezeichneten Konfigurationen mit den gewünschten sicheren Konfigurationen vergleichen können. Mit AWS Config können Sie Folgendes tun:

- Prüfen von Änderungen der Konfigurationen und Beziehungen zwischen AWS -Ressourcen
- Untersuchen detaillierter Ressourcenkonfigurationsverläufe
- Bestimmen der allgemeinen Compliance im Hinblick auf die Konfigurationen, die in Ihren internen Leitlinien angegeben sind

Mithilfe AWS Config können Sie Compliance-Prüfungen, Sicherheitsanalysen, das Änderungsmanagement und die betriebliche Fehlerbehebung vereinfachen. Weitere Informationen finden Sie unter Einrichtung AWS Config mit der Konsole im AWS Config Entwicklerhandbuch. Wenn Sie die aufzuzeichnenden Ressourcentypen angeben, stellen Sie sicher, dass Sie Amazon-S3-Ressourcen mit einbeziehen.

> **Important**
>
> AWS Config managed rules unterstützt bei der Evaluierung von Amazon S3 S3-Ressourcen nur Buckets für allgemeine Zwecke. AWS Config zeichnet keine Konfigurationsänderungen für Verzeichnis-Buckets auf. Weitere Informationen finden Sie unter AWS Config Verwaltete Regeln und Liste der AWS Config verwalteten Regeln im AWS Config Entwicklerhandbuch.

Ein AWS Config Anwendungsbeispiel finden Sie im AWS Sicherheits-Blog unter How to Use AWS Config to Monitor for and Respond to Amazon S3 Buckets Allowing Public Access.

Verwenden von S3 Storage Lens

S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. S3 Storage Lens analysiert Metriken, um kontextbezogene Empfehlungen zur Optimierung der Speicherkosten und zur Anwendung bewährter Datenschutzmethoden zu geben.

Mit S3 Storage Lens können Sie Metriken verwenden, um zusammenfassende Erkenntnisse zu gewinnen und z. B. herauszufinden, wie viel Speicher Sie in Ihrer gesamten Organisation haben oder welche Buckets und Präfixe am schnellsten wachsen. Außerdem können Sie anhand der Metriken von S3 Storage Lens umfassende Möglichkeiten zur Kostenoptimierung aufdecken, bewährte Methoden für den Datenschutz und die Zugriffsverwaltung implementieren und die Leistung von Anwendungs-Workloads verbessern.

Sie können beispielsweise Buckets identifizieren, für die keine S3-Lebenszyklusregeln gelten, damit unvollständige mehrteilige Uploads, die älter als 7 Tage sind, abgebrochen werden. Sie können auch Buckets identifizieren, die nicht den bewährten Datenschutzmethoden entsprechen, z. B. die Verwendung von S3 Replication oder S3 Versionierung. Weitere Informationen finden Sie unter Grundlegendes zu Amazon S3 Storage Lens.

Überwachen von AWS -Sicherheitsempfehlungen

Wir raten Ihnen, die in Trusted Advisor für Ihr AWS-Konto geposteten Sicherheitsempfehlungen regelmäßig zu überprüfen. Achten Sie dabei besonders auf Warnungen zu Amazon-S3-Buckets mit „offenen Zugriffsberechtigungen“. Sie können dies programmgesteuert tun, indem Sie describe-trusted-advisor-checks.

Überwachen Sie außerdem aktiv die primäre E-Mail-Adresse, die für jede Ihrer E-Mails registriert ist. AWS-Konten AWS verwendet diese E-Mail-Adresse, um Sie über neu auftretende Sicherheitsprobleme zu informieren, die Sie betreffen könnten.

AWS betriebliche Probleme mit weitreichenden Auswirkungen werden im AWS Health Dashboard Servicestatus veröffentlicht. Operative Probleme werden über AWS Health Dashboard auch in den einzelnen Konten veröffentlicht. Weitere Informationen finden Sie in der AWS Health - Dokumentation.

Überwachung der Datensicherheit mit verwalteten AWS Sicherheitsdiensten

Verschiedene verwaltete AWS Sicherheitsservices können Ihnen dabei helfen, Sicherheits- und Compliance-Risiken für Ihre Amazon S3 S3-Daten zu identifizieren, zu bewerten und zu überwachen. Sie können Sie auch dabei unterstützen, Ihre Daten vor diesen Risiken zu schützen. Diese Services umfassen automatisierte Erkennungs-, Überwachungs- und Schutzfunktionen, die so konzipiert sind, dass sie von Amazon S3 S3-Ressourcen für eine einzelne Person bis hin AWS-Konto zu Ressourcen für Organisationen mit Tausenden von AWS-Konten Unternehmen skaliert werden können.

AWS Erkennungs- und Reaktionsdienste können Ihnen helfen, potenzielle Sicherheitsfehlkonfigurationen, Bedrohungen oder unerwartetes Verhalten zu identifizieren, sodass Sie schnell auf potenziell unautorisierte oder böswillige Aktivitäten in Ihrer Umgebung reagieren können. AWS Datenschutzdienste können Ihnen helfen, Ihre Daten, Konten und Workloads zu überwachen und vor unbefugtem Zugriff zu schützen. Sie können Ihnen auch dabei helfen, sensible Daten, wie z. B. persönlich identifizierbare Informationen (PII), in Ihrem Amazon S3 S3-Datenbestand zu entdecken.

Um Sie bei der Identifizierung und Bewertung von Datensicherheits- und Compliance-Risiken zu unterstützen, generieren verwaltete AWS -Sicherheitsservices Erkenntnisse, um Sie über potenzielle Sicherheitsereignisse oder -probleme mit Ihren Amazon-S3-Daten zu informieren. Die Erkenntnisse enthalten relevante Informationen, anhand derer Sie diese Risiken untersuchen, bewerten und entsprechend Ihren Vorfallreaktions-Workflows und -richtlinien handeln können. Über die einzelnen Services können Sie direkt auf die Erkenntnisdaten zugreifen. Sie können die Daten auch an andere Anwendungen, Dienste und Systeme senden, z. B. an Ihr System zur Verwaltung von Sicherheitsvorfällen und Ereignissen (SIEM).

Um die Sicherheit Ihrer Amazon S3 S3-Daten zu überwachen, sollten Sie die Nutzung dieser verwalteten AWS Sicherheitsservices in Betracht ziehen.

Amazon GuardDuty

Amazon GuardDuty ist ein Service zur Bedrohungserkennung, der Ihre Workloads AWS-Konten und Ihre Workloads kontinuierlich auf böswillige Aktivitäten überwacht und detaillierte Sicherheitsergebnisse zur besseren Übersicht und Problembehebung liefert.

Wenn die S3-Schutzfunktion aktiviert ist GuardDuty, können Sie die Konfiguration so konfigurieren, GuardDuty dass AWS CloudTrail Management- und Datenereignisse für Ihre Amazon S3 S3-Ressourcen analysiert werden. GuardDuty überwacht diese Ereignisse dann auf böswillige und verdächtige Aktivitäten. GuardDuty Nutzt Threat-Intelligence-Feeds und

maschinelles Lernen, um fundierte Analysen durchzuführen und potenzielle Sicherheitsrisiken zu identifizieren.

GuardDuty kann verschiedene Arten von Aktivitäten für Ihre Amazon S3 S3-Ressourcen überwachen. Zu den CloudTrail Verwaltungsereignissen für Amazon S3 gehören beispielsweise Operationen auf Bucket-Ebene, wie `ListBucketsDeleteBucket`, und. `PutBucketReplication` CloudTrail Datenereignisse für Amazon S3 umfassen Operationen auf Objektebene, wie `GetObjectListObjects`, und. `PutObject` Wenn ungewöhnliche oder potenziell bösartige Aktivitäten GuardDuty erkannt werden, generiert es einen Befund, der Sie benachrichtigt.

Weitere Informationen finden Sie unter Amazon S3 Protection in Amazon GuardDuty im GuardDuty Amazon-Benutzerhandbuch.

Amazon Detective

Amazon Detective vereinfacht den Ermittlungsprozess und hilft Ihnen, schnellere und effektivere Sicherheitsuntersuchungen durchzuführen. Detective bietet vordefinierte Datenaggregationen, Zusammenfassungen und Kontexte, mit denen Sie Art und Ausmaß möglicher Sicherheitsprobleme analysieren und bewerten können.

Detective extrahiert automatisch zeitbasierte Ereignisse wie API Anrufe von AWS CloudTrail und Amazon VPC Flow Logs für Ihre AWS Ressourcen. Es nimmt auch von Amazon GuardDuty generierte Ergebnisse auf. Detective verwendet dann Machine Learning, statistische Analysen und die Graphentheorie, um Visualisierungen zu erstellen, mit denen Sie effektive Sicherheitsuntersuchungen schneller durchführen können.

Diese Visualisierungen bieten eine einheitliche, interaktive Ansicht des Ressourcenverhaltens und der Interaktionen zwischen ihnen im Zeitverlauf. Anhand dieses Verhaltensdiagramms können Sie potenziell böswillige Aktionen untersuchen, wie z. B. fehlgeschlagene Anmeldeversuche oder verdächtige API Anrufe. Sie können auch sehen, wie sich diese Aktionen auf Ressourcen wie S3-Buckets und -Objekte auswirken.

Weitere Informationen finden Sie im Administratorhandbuch für Amazon Detective.

IAM Access Analyzer

AWS Identity and Access Management Access Analyzer (IAMAccess Analyzer) kann Ihnen helfen, Ressourcen zu identifizieren, die mit einer externen Entität gemeinsam genutzt werden. Sie können IAM Access Analyzer auch verwenden, um IAM Richtlinien anhand der

Richtliniengrammatik und der bewährten Methoden zu überprüfen und IAM Richtlinien auf der Grundlage der Zugriffsaktivitäten in Ihren AWS CloudTrail Protokollen zu generieren.

IAMAccess Analyzer analysiert Ressourcenrichtlinien in Ihrer AWS Umgebung mithilfe von logischer Argumentation, z. B. Bucket-Richtlinien. Mit IAM Access Analyzer for S3 werden Sie benachrichtigt, wenn ein S3-Bucket so konfiguriert ist, dass er jedem Zugriff auf das Internet oder andere Personen gewährt AWS-Konten, auch Konten außerhalb Ihrer Organisation. IAMAccess Analyzer for S3 kann beispielsweise melden, dass ein Bucket Lese- oder Schreibzugriff hat, der über eine Bucket-Zugriffskontrollliste (ACL), eine Bucket-Richtlinie, eine Multi-Region-Access-Point-Richtlinie oder eine Access-Point-Richtlinie gewährt wird. Für jeden öffentlichen oder freigegebenen Bucket erhalten Sie Erkenntnisse, die die Quelle und die Ebene des öffentlichen oder freigegebenen Zugriffs angeben. Anhand dieser Erkenntnisse können Sie sofortige und präzise Korrekturmaßnahmen ergreifen, um den Bucket-Zugriff wie beabsichtigt wiederherzustellen.

Weitere Informationen finden Sie unter Überprüfung des Bucket-Zugriffs mit IAM Access Analyzer für S3.

Amazon Macie

Amazon Macie ist ein Sicherheitsservice, der sensible Daten mithilfe von Machine Learning und Musterabgleich erkennt. Macie bietet Einblick in Datensicherheitsrisiken und ermöglicht automatischen Schutz vor diesen Risiken. Mit Macie können Sie die Erkennung und Meldung sensibler Daten in Ihrem Amazon-S3-Datenbestand automatisieren, um die Daten besser zu verstehen, die Ihre Organisation in S3 speichert.

Um sensible Daten mit Macie zu erkennen, können Sie integrierte Kriterien und Techniken verwenden, die darauf ausgelegt sind, eine lange und wachsende Liste vertraulicher Datentypen für viele Länder und Regionen zu erkennen. Zu diesen sensiblen Datentypen gehören mehrere Arten von personenbezogenen Daten (PII), Finanzdaten und Anmeldedaten. Sie können auch benutzerdefinierte Kriterien verwenden, die Sie definieren – reguläre Ausdrücke, die Textmuster zum Abgleichen definieren, und optional Zeichenfolgen und Näherungsregeln, um die Ergebnisse zu verfeinern.

Wenn Macie sensible Daten in einem S3-Objekt entdeckt, generiert es eine Sicherheitserkenntnis, um Sie zu benachrichtigen. Diese Erkenntnis enthält Informationen über das betroffene Objekt, die Art und Häufigkeit der Vorkommen der sensiblen Daten, die Macie gefunden hat, sowie zusätzliche Details, die Ihnen bei der Untersuchung des betroffenen S3-Buckets und -Objekts helfen. Weitere Informationen finden Sie im Benutzerhandbuch für Amazon Macie.

AWS Security Hub

AWS Security Hub ist ein Service zur Verwaltung des Sicherheitsstatus, der bewährte Sicherheitsverfahren überprüft, Warnmeldungen und Ergebnisse aus mehreren Quellen in einem einzigen Format zusammenfasst und automatische Problembehebungen ermöglicht.

Security Hub sammelt und stellt Daten zu Sicherheitsergebnissen aus integrierten AWS Partner Network Sicherheitslösungen wie Amazon Detective, Amazon GuardDuty, IAM Access Analyzer und Amazon Macie bereit. AWS-Services Es generiert auch seine eigenen Ergebnisse, indem es kontinuierliche, automatisierte Sicherheitsprüfungen durchführt, die auf AWS bewährten Verfahren und unterstützten Industriestandards basieren.

Security Hub korreliert und konsolidiert die Erkenntnisse dann anbieterübergreifend, um Ihnen dabei zu helfen, die wichtigsten Erkenntnisse zu priorisieren und zu verarbeiten. Es bietet auch Unterstützung für benutzerdefinierte Aktionen, mit denen Sie Antworten oder Abhilfemaßnahmen für bestimmte Klassen von Erkenntnissen aufrufen können.

Mit Security Hub können Sie den Sicherheits- und Compliance-Status Ihrer Amazon S3 S3-Ressourcen bewerten, und zwar im Rahmen einer umfassenderen Analyse der Sicherheitslage Ihres Unternehmens in einzelnen AWS-Regionen und in mehreren Regionen. Dazu gehören die Analyse von Sicherheitstrends und die Identifizierung der Sicherheitsprobleme mit der höchsten Priorität. Sie können auch Erkenntnisse aus mehreren AWS-Regionen zusammenfassen und aggregierte Erkenntnisdaten aus einer einzelnen Region überwachen und verarbeiten.

Weitere Informationen finden Sie unter Kontrollen von Amazon Simple Storage Service im Benutzerhandbuch für AWS Security Hub .

Datenschutz in Amazon S3

Amazon S3 bietet eine sehr robuste Speicherinfrastruktur, die für geschäftskritische und primäre Speicheranwendungen entwickelt wurde. S3-Standard, S3 Intelligent-Tiering, S3-Standard-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive speichern Objekte redundant auf mehreren Geräten in mindestens drei Availability Zones in einem AWS-Region. Eine Availability Zone ist eines oder mehrere diskrete Rechenzentren mit redundanter Stromversorgung, Vernetzung und Konnektivität in einem AWS-Region. Availability Zones sind physisch durch eine bedeutende Entfernung von vielen Kilometern von jeder anderen Availability Zone getrennt, obwohl alle innerhalb von 100 km (60 Meilen) voneinander liegen. Die Speicherklasse S3 One Zone-IA speichert Daten redundant auf mehreren Geräten innerhalb einer einzigen

Availability Zone. Diese Services wurden entwickelt, um mit gleichzeitigen Geräteausfällen umzugehen, indem sie verlorene Redundanz schnell erkennen und reparieren, und auch regelmäßig die Integrität Ihrer Daten mithilfe von Prüfsummen überprüfen.

Der Amazon-S3-Standardspeicher bietet folgende Funktionen:

- Unterstützung durch Service Level Agreement von Amazon S3.
- Auf 99,999999999 %-ige Zuverlässigkeit und 99,99 %-ige Verfügbarkeit von Objekten über einen Zeitraum von einem Jahr ausgelegt.
- S3-Standard, S3 Intelligent-Tiering, S3-Standard-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive sind alle darauf ausgelegt, Daten im Falle des Verlustes einer gesamten Amazon-S3-Availability-Zone zu erhalten.

Darüber hinaus schützt Amazon S3 Ihre Daten durch Versioning. Sie können Versioning verwenden, um sämtliche Versionen aller Objekte in Ihrem Amazon-S3-Bucket zu speichern, abzurufen oder wiederherzustellen. Daten lassen sich dank Versioning nach unbeabsichtigten Benutzeraktionen und Anwendungsfehlern leicht wiederherstellen. Standardmäßig wird durch Anforderungen die zuletzt geschriebene Version abgerufen. Ältere Versionen eines Objekts können abgerufen werden, indem die entsprechende Version des Objekts in der Anforderung angegeben wird.

Neben S3 Versioning können Sie auch Amazon S3 Object Lock und S3 Replication verwenden, um Ihre Daten zu schützen. Weitere Informationen finden Sie im Tutorial: Schutz von Daten in Amazon S3 vor versehentlichem Löschen oder Anwendungsfehlern mithilfe von S3 Versionierung, S3 Object Lock und S3 Replication.

Aus Datenschutzgründen empfehlen wir, dass Sie Ihre AWS-Konto Anmeldeinformationen schützen und individuelle Benutzerkonten mit einrichten AWS Identity and Access Management, sodass jeder Benutzer nur die Berechtigungen erhält, die für die Erfüllung seiner Aufgaben erforderlich sind.

Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine andere FIPS 140-2 validierte kryptografische Module benötigenAPI, verwenden Sie einen Endpunkt. FIPS Weitere Informationen zu den verfügbaren FIPS Endpunkten finden Sie unter Federal Information Processing Standard () 140-2. FIPS

Die folgenden bewährten Sicherheitsmethoden umfassen ebenfalls den Adressdatenschutz in Amazon S3:

- Implement server-side encryption
- Enforce encryption of data in transit

- Consider using Macie with Amazon S3
- Identify and audit all your Amazon S3 buckets
- Monitor Amazon Web Services security advisories

Datenschutz durch Verschlüsselung

> **Important**
>
> Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist in den AWS CloudTrail Protokollen, im S3-Inventar, in der S3-Speicherlinse, in der Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header im AWS Command Line Interface und AWS SDKs verfügbar. Weitere Informationen finden Sie unter Standardverschlüsselung FAQ.

Dieser Datenschutz bezieht sich auf Daten bei der Übertragung (wenn sie zu Amazon S3 oder von diesem geschickt werden), ebenso wie auf ruhende Daten (die in Amazon-S3-Rechenzentren auf Datenträgern gespeichert sind). Sie können Daten während der Übertragung mithilfe von Secure Socket Layer/Transport Layer Security (SSL/TLS () oder clientseitiger Verschlüsselung schützen. Sie haben folgende Optionen, um Daten im Ruhezustand in Amazon S3 zu schützen:

- Serverseitige Verschlüsselung — Amazon S3 verschlüsselt Ihre Objekte, bevor sie auf Festplatten in AWS Rechenzentren gespeichert werden, und entschlüsselt die Objekte dann, wenn Sie sie herunterladen.

  Für alle Amazon-S3-Buckets ist die Verschlüsselung standardmäßig konfiguriert und alle neuen Objekte, die in einen S3-Bucket hochgeladen werden, werden im Ruhezustand automatisch verschlüsselt. Die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Standardverschlüsselungskonfiguration für jeden Bucket in Amazon S3. Um einen anderen Verschlüsselungstyp zu verwenden, können Sie entweder die Art der serverseitigen Verschlüsselung angeben, die in Ihren S3-PUT-Anfragen verwendet werden soll, oder Sie können die Standardverschlüsselungskonfiguration im Ziel-Bucket festlegen.

Wenn Sie in Ihren PUT Anfragen einen anderen Verschlüsselungstyp angeben möchten, können Sie serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln (-), zweischichtige serverseitige Verschlüsselung mit Schlüsseln (SSE-KMS) oder serverseitige Verschlüsselung mit vom Kunden bereitgestellten AWS KMS Schlüsseln (DSSE-CKMS) verwenden. SSE Wenn Sie im Ziel-Bucket eine andere Standardverschlüsselungskonfiguration festlegen möchten, können Sie - oder - verwenden. SSE KMS DSSE KMS

Weitere Informationen zu den einzelnen Optionen für die serverseitige Verschlüsselung finden Sie unter Schützen von Daten mit serverseitiger Verschlüsselung.

Informationen zum Konfigurieren der serverseitigen Verschlüsselung finden Sie unter:

- Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) angeben
- Geben Sie die serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) an
- the section called “Spezifizierung von - DSSE KMS”
- Angeben der serverseitigen Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE

- Clientseitige Verschlüsselung – Sie können Daten clientseitig verschlüsseln und die verschlüsselten Daten auf Amazon S3 hochladen. In diesem Fall verwalten Sie den Verschlüsselungsprozess, die Verschlüsselungsschlüssel und die zugehörigen Tools.

  Informationen zum Konfigurieren der clientseitigen Verschlüsselung finden Sie unter Schützen von Daten mithilfe der clientseitigen Verschlüsselung.

Wenn Sie feststellen möchten, welcher Prozentsatz Ihrer Speicherbytes verschlüsselt ist, können Sie die Metriken von Amazon S3 Storage Lens verwenden. S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Weitere Informationen finden Sie unter Bewertung Ihrer Speicheraktivität und -nutzung mit S3 Storage Lens. Eine vollständige Liste der Metriken finden Sie im Glossar der S3-Storage-Lens-Metriken.

Weitere Informationen zur serverseitigen Verschlüsselung und zur clientseitigen Verschlüsselung finden Sie in den unten aufgeführten Themen.

Themen

- Schützen von Daten mit serverseitiger Verschlüsselung

- Schützen von Daten mithilfe der clientseitigen Verschlüsselung

Schützen von Daten mit serverseitiger Verschlüsselung

> **Important**
>
> Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist in den AWS CloudTrail Protokollen, im S3-Inventar, in der S3-Speicherlinse, in der Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header im AWS Command Line Interface und AWS SDKs verfügbar. Weitere Informationen finden Sie unter Standardverschlüsselung FAQ.

Serverseitige Verschlüsselung ist die Verschlüsselung von Daten am Zielort durch die Anwendung oder den Service, der sie erhält. Amazon S3 verschlüsselt Ihre Daten auf Objektebene, wenn es sie auf Festplatten in AWS Rechenzentren schreibt, und entschlüsselt sie für Sie, wenn Sie darauf zugreifen. Wenn Sie Ihre Anforderung authentifizieren und Zugriffsberechtigungen besitzen, gibt es in Bezug auf die Art und Weise, wie Sie auf verschlüsselte oder nicht verschlüsselte Objekte zugreifen, keinen Unterschied. Wenn Sie Ihre Objekte beispielsweise mit einem vorsignierten Objekt teilenURL, URL funktioniert das sowohl für verschlüsselte als auch für unverschlüsselte Objekte auf die gleiche Weise. Wenn Sie Objekte in Ihrem Bucket auflisten, geben die API Listenoperationen außerdem eine Liste aller Objekte zurück, unabhängig davon, ob sie verschlüsselt sind.

Für alle Amazon-S3-Buckets ist die Verschlüsselung standardmäßig konfiguriert und alle neuen Objekte, die in einen S3-Bucket hochgeladen werden, werden im Ruhezustand automatisch verschlüsselt. Die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Standardverschlüsselungskonfiguration für jeden Bucket in Amazon S3. Um einen anderen Verschlüsselungstyp zu verwenden, können Sie entweder die Art der serverseitigen Verschlüsselung angeben, die in Ihren S3-PUT-Anfragen verwendet werden soll, oder Sie können die Standardverschlüsselungskonfiguration im Ziel-Bucket festlegen.

Wenn Sie in Ihren PUT Anfragen einen anderen Verschlüsselungstyp angeben möchten, können Sie serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln

(-), zweischichtige serverseitige Verschlüsselung mit Schlüsseln (SSE-KMS) oder serverseitige Verschlüsselung mit vom Kunden bereitgestellten AWS KMS Schlüsseln (DSSE-CKMS) verwenden. SSE Wenn Sie im Ziel-Bucket eine andere Standardverschlüsselungskonfiguration festlegen möchten, können Sie - oder - verwenden. SSE KMS DSSE KMS

> **Note**
>
> Sie können nicht gleichzeitig unterschiedliche Arten serverseitiger Verschlüsselung auf dasselbe Objekt anwenden.

Wenn Sie Ihre vorhandenen Objekte verschlüsseln müssen, verwenden Sie S3 Batch Operations und S3 Inventory. Weitere Informationen finden Sie unter Verschlüsseln von Objekten mit Amazon S3 Batch Operations) und Massenausführung von Objektoperationen mit Batch Operations.

Sie haben vier sich gegenseitig ausschließende Optionen für die serverseitige Verschlüsselung, abhängig davon, wie Sie die Verschlüsselungsschlüssel verwalten und wie viele Verschlüsselungsebenen Sie anwenden möchten.

Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3)

Für alle Amazon-S3-Buckets ist die Verschlüsselung standardmäßig konfiguriert. Die Standardoption für serverseitige Verschlüsselung ist mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3). Jedes Objekt wird mit einem eindeutigen Schlüssel verschlüsselt. Als zusätzliche Sicherheitsmaßnahme verschlüsselt SSE -S3 den Schlüssel selbst mit einem Stammschlüssel, den es regelmäßig wechselt. SSE-S3 verwendet eine der stärksten verfügbaren Blockchiffren, den 256-Bit-Advanced Encryption Standard (AES-256), um Ihre Daten zu verschlüsseln. Weitere Informationen finden Sie unter Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden (SSE-S3).

Serverseitige Verschlüsselung mit () Schlüsseln (-) AWS Key Management ServiceAWS KMS SSE KMS

Die serverseitige Verschlüsselung mit AWS KMS keys (SSE-KMS) wird durch eine Integration des AWS KMS Dienstes mit Amazon S3 bereitgestellt. Mit AWS KMS haben Sie mehr Kontrolle über Ihre Schlüssel. Sie können beispielsweise separate Schlüssel anzeigen, Kontrollrichtlinien bearbeiten und den Schlüsseln in AWS CloudTrail folgen. Darüber hinaus können Sie vom Kunden verwaltete Schlüssel erstellen und verwalten oder von Von AWS verwaltete Schlüssel verwaltete Schlüssel verwenden, die für Sie, Ihren Service und Ihre Region einzigartig sind. Weitere Informationen finden Sie unter Serverseitige Verschlüsselung mit AWS KMS Schlüsseln verwenden (SSE-) KMS.

Zweischichtige serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln (DSSE-) KMS

Die zweischichtige serverseitige Verschlüsselung mit AWS KMS keys (DSSE-KMS) ähnelt SSE - KMS, verwendet jedoch DSSE zwei einzelne Verschlüsselungsebenen auf Objektebene anstelle einer Ebene. KMS Da beide Verschlüsselungsebenen serverseitig auf ein Objekt angewendet werden, können Sie eine Vielzahl von Tools verwenden, um Daten in S3 zu analysieren AWS-Services und gleichzeitig eine Verschlüsselungsmethode zu verwenden, die Ihren Compliance-Anforderungen gerecht wird. Weitere Informationen finden Sie unter Verwendung einer zweischichtigen serverseitigen Verschlüsselung mit AWS KMS Schlüsseln (DSSE-) KMS.

Serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE

Bei serverseitiger Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) verwalten Sie die Verschlüsselungsschlüssel, und Amazon S3 verwaltet die Verschlüsselung beim Schreiben auf Festplatten und die Entschlüsselung beim Zugriff auf Ihre Objekte. Weitere Informationen finden Sie unter Verwenden Sie serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE.

Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets

> **Important**
>
> Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist in den AWS CloudTrail Protokollen, im S3-Inventar, in der S3-Speicherlinse, in der Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header im AWS Command Line Interface und AWS SDKs verfügbar. Weitere Informationen finden Sie unter Standardverschlüsselung FAQ.

Für alle Amazon S3 S3-Buckets ist die Verschlüsselung standardmäßig konfiguriert, und Objekte werden automatisch mithilfe serverseitiger Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt. Diese Verschlüsselungseinstellung gilt für alle Objekte in Ihren Amazon-S3-Buckets.

Wenn Sie mehr Kontrolle über Ihre Schlüssel benötigen, z. B. bei der Verwaltung der Schlüsselrotation und der Gewährung von Zugriffsrichtlinien, können Sie zwischen serverseitiger Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln (SSE-KMS) oder zweischichtiger serverseitiger Verschlüsselung mit Schlüsseln (-) wählen. AWS KMS DSSE KMS Weitere Informationen zum Bearbeiten von KMS Schlüsseln finden Sie unter Bearbeiten von Schlüsseln im AWS Key Management Service Entwicklerhandbuch.

> **Note**
>
> Wir haben die Buckets geändert, sodass neue Objekt-Uploads automatisch verschlüsselt werden. Wenn Sie zuvor einen Bucket ohne Standardverschlüsselung erstellt haben, aktiviert Amazon S3 standardmäßig die Verschlüsselung für den Bucket mit SSE -S3. Für einen vorhandenen Bucket, für den bereits SSE -S3 oder SSE - KMS konfiguriert ist, werden keine Änderungen an der Standardverschlüsselungskonfiguration vorgenommen. Wenn Sie Ihre Objekte mit SSE - verschlüsseln möchtenKMS, müssen Sie den Verschlüsselungstyp in Ihren Bucket-Einstellungen ändern. Weitere Informationen finden Sie unter Serverseitige Verschlüsselung mit AWS KMS Schlüsseln verwenden (SSE-) KMS.

Wenn Sie Ihren Bucket so konfigurieren, dass er die Standardverschlüsselung mit SSE - verwendetKMS, können Sie auch S3 Bucket Keys aktivieren, um den Anforderungsverkehr von Amazon S3 zu reduzieren AWS KMS und die Kosten für die Verschlüsselung zu senken. Weitere Informationen finden Sie unter Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys.

Um Buckets zu identifizieren, für die SSE die Standardverschlüsselung KMS aktiviert ist, können Sie Amazon S3 Storage Lens-Metriken verwenden. S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Weitere Informationen finden Sie unter Verwenden von S3 Storage Lens zum Schutz Ihrer Daten.

Wenn Sie die serverseitige Verschlüsselung verwenden, verschlüsselt Amazon S3 ein Objekt vor dem Speichern auf der Festplatte und entschlüsselt es beim Herunterladen des Objekts. Weitere Informationen zum Schutz von Daten mithilfe der serverseitigen Verschlüsselung und der Verwaltung der Verschlüsselungsschlüssel finden Sie unter Schützen von Daten mit serverseitiger Verschlüsselung.

Weitere Informationen zu den für die Standardverschlüsselung erforderlichen Berechtigungen finden Sie unter PutBucketEncryptionin der Amazon Simple Storage Service API Reference.

Sie können das standardmäßige Verschlüsselungsverhalten von Amazon S3 für einen S3-Bucket konfigurieren, indem Sie die AWS SDKs Amazon S3-Konsole RESTAPI, Amazon S3 und die AWS Befehlszeilenschnittstelle (AWS CLI) verwenden.

Verschlüsseln vorhandener Objekte

Zum Verschlüsseln Ihrer vorhandenen nicht verschlüsselten Amazon-S3-Objekte können Sie Amazon S3 Batch Operations verwenden. Sie stellen S3 Batch Operations eine Liste von Objekten zur Verfügung, mit denen gearbeitet werden soll, und Batch Operations ruft die entsprechenden Objekte aufAPI, um die angegebene Operation auszuführen. Mit der Operation Batch Operations Copy können Sie vorhandenen nicht verschlüsselte Objekte kopieren und die neuen verschlüsselten Objekte in denselben Bucket schreiben. Ein einzelner Batchoperations-Auftrag kann die angegebene Operation für Milliarden von Objekten ausführen. Weitere Informationen finden Sie unter Massenausführung von Objektoperationen mit Batch Operations und im Beitrag des AWS Storage Blog Encrypting objects with Amazon S3 Batch Operations.

Sie können auch vorhandene Objekte verschlüsseln, indem Sie die `CopyObject` API Operation oder den `copy-object` AWS CLI Befehl verwenden. Weitere Informationen finden Sie unter AWS CLI und im Beitrag des AWS Storage Blog Encrypting existing objects with Amazon S3 Batch Operations.

> **Note**
>
> Amazon S3 S3-Buckets, bei denen die Standard-Bucket-Verschlüsselung auf SSE — eingestellt ist, KMS können nicht als Ziel-Buckets für verwendet werden. the section called “Protokollierungs-Serverzugriff” Für SSE Ziel-Buckets des Serverzugriffsprotokolls wird nur die Standardverschlüsselung -S3 unterstützt.

Verwendung von SSE — KMS Verschlüsselung für kontoübergreifende Operationen

Beachten Sie Folgendes, wenn Sie kontoübergreifende Operationen verschlüsseln:

- Wenn bei der Anfrage oder über die Standardverschlüsselungskonfiguration des Buckets kein AWS KMS key Amazon-Ressourcenname (ARN) oder Alias angegeben wird, wird Von AWS verwalteter Schlüssel (`aws/s3`) verwendet.
- Wenn Sie S3-Objekte hochladen oder auf sie zugreifen, indem Sie AWS Identity and Access Management (IAM) -Prinzipale verwenden, die mit Ihrem KMS Schlüssel AWS-Konto identisch sind, können Sie Von AWS verwalteter Schlüssel (`aws/s3`) verwenden.

- Verwenden Sie einen vom Kunden verwalteten Schlüssel, wenn Sie kontoübergreifenden Zugriff auf Ihre S3-Objekte gewähren möchten. Sie können die Richtlinie eines vom Kunden verwalteten Schlüssel so konfigurieren, dass der Zugriff von einem anderen Konto aus möglich ist.
- Wenn Sie einen vom Kunden verwalteten KMS Schlüssel angeben, empfehlen wir die Verwendung eines vollqualifizierten KMS Schlüssels. ARN Wenn Sie stattdessen einen KMS Schlüsselalias verwenden, AWS KMS wird der Schlüssel im Konto des Anfragenden aufgelöst. Dieses Verhalten kann dazu führen, dass Daten mit einem KMS Schlüssel verschlüsselt werden, der dem Anforderer und nicht dem Bucket-Besitzer gehört.
- Sie müssen einen Schlüssel angeben, für den Ihnen (dem Anforderer) die Berechtigung `Encrypt` erteilt wurde. Weitere Informationen finden Sie im AWS Key Management Service Entwicklerhandbuch unter Erlauben Sie Schlüsselbenutzern, einen KMS Schlüssel für kryptografische Operationen zu verwenden.

Weitere Informationen darüber, wann vom Kunden verwaltete Schlüssel und AWS KMS verwaltete Schlüssel verwendet werden sollten, finden Sie unter Sollte ich einen Von AWS verwalteter Schlüssel oder einen vom Kunden verwalteten Schlüssel verwenden, um meine Objekte in Amazon S3 zu verschlüsseln?

Verwenden der Standard-Verschlüsselung mit der Replikation

Wenn Sie die Standard-Verschlüsselung für einen Replikations-Ziel-Bucket aktivieren, gilt das folgende Verschlüsselungsverhalten:

- Wenn Objekte im Quell-Bucket nicht verschlüsselt sind, werden die Replikatobjekte im Ziel-Bucket mithilfe der Einstellungen der Standard-Verschlüsselung des Ziel-Buckets verschlüsselt. Infolgedessen unterscheiden sich die Entity-Tags (ETags) der Quellobjekte von denen ETags der Replikatobjekte. Wenn Sie Anwendungen haben, die dies verwendenETags, müssen Sie diese Anwendungen aktualisieren, um diesem Unterschied Rechnung zu tragen.
- Wenn Objekte im Quell-Bucket mit serverseitiger Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3), serverseitiger Verschlüsselung mit AWS Key Management Service () -Schlüsseln (SSE-AWS KMS KMS) oder zweischichtiger serverseitiger Verschlüsselung mit AWS KMS Schlüsseln (DSSE-KMS) verschlüsselt werden, verwenden die Replikatobjekte im Ziel-Bucket denselben Verschlüsselungstyp wie die Quellobjekte. Die Einstellungen der Standard-Verschlüsselung des Ziel-Buckets werden nicht verwendet.

Weitere Informationen zur Verwendung der Standardverschlüsselung mit - finden Sie unter. SSE KMS Replizieren verschlüsselter Objekte

Verwenden von Amazon S3-Bucket-Schlüsseln mit Standard-Verschlüsselung

Wenn Sie Ihren Bucket so konfigurieren, dass er SSE - KMS als Standardverschlüsselungsverhalten für neue Objekte verwendet, können Sie auch S3-Bucket-Keys konfigurieren. S3-Bucket-Keys reduzieren die Anzahl der Transaktionen von Amazon S3, AWS KMS um die Kosten von SSE - zu senkenKMS.

Wenn Sie Ihren Bucket so konfigurieren, dass S3-Bucket-Keys für SSE — für neue Objekte verwendet werden, wird ein Schlüssel KMS auf Bucket-Ebene AWS KMS generiert, der verwendet wird, um einen eindeutigen Datenschlüssel für Objekte im Bucket zu erstellen. Dieser S3-Bucket-Key wird für einen begrenzten Zeitraum innerhalb von Amazon S3 verwendet, sodass Amazon S3 weniger Anfragen AWS KMS zum Abschluss von Verschlüsselungsvorgängen stellen muss.

Weitere Informationen zur Verwendung von S3-Bucket-Schlüsseln finden Sie unter Verwenden von Amazon-S3-Bucket-Schlüssel.

Konfigurieren der Standardverschlüsselung

> **Important**
>
> Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist in den AWS CloudTrail Protokollen, im S3-Inventar, in der S3-Speicherlinse, in der Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header im AWS Command Line Interface und AWS SDKs verfügbar. Weitere Informationen finden Sie unter Standardverschlüsselung FAQ.

In Amazon S3 S3-Buckets ist die Bucket-Verschlüsselung standardmäßig aktiviert, und neue Objekte werden automatisch mithilfe serverseitiger Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt. Diese Verschlüsselung gilt für alle neuen Objekte in Ihren Amazon-S3-Buckets und es fallen keine Kosten für Sie an.

Wenn Sie mehr Kontrolle über Ihre Verschlüsselungsschlüssel benötigen, z. B. die Verwaltung der Schlüsselrotation und die Gewährung von Zugriffsrichtlinien, können Sie wählen, ob Sie serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-) oder eine zweischichtige serverseitige Verschlüsselung mit Schlüsseln (-KMS) verwenden möchten. AWS KMS DSSE KMS Weitere Hinweise zu SSE - KMS finden Sie unter. Geben Sie die serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) an Weitere Informationen zu DSSE - finden KMS Sie unterthe section called “Serverseitige Dual-Layer-Verschlüsselung (-) DSSE KMS”.

Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, müssen Sie über die entsprechende Berechtigung verfügen. Weitere Informationen zu kontoübergreifenden Berechtigungen für KMS Schlüssel finden Sie im AWS Key Management Service Entwicklerhandbuch unter Erstellen von KMS Schlüsseln, die andere Konten verwenden können.

Wenn Sie die Standard-Bucket-Verschlüsselung auf SSE - setzenKMS, können Sie auch einen S3-Bucket-Key konfigurieren, um Ihre AWS KMS Anforderungskosten zu senken. Weitere Informationen finden Sie unter Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys.

> **Note**
>
> Wenn Sie Ihre Standard-Bucket-Verschlüsselung auf SSE - setzenKMS, sollten Sie überprüfen, ob Ihre KMS Schlüssel-ID korrekt ist. PutBucketEncryption Amazon S3 validiert die in PutBucketEncryption Anfragen angegebene KMS Schlüssel-ID nicht.

Es gibt keine zusätzlichen Gebühren für die Nutzung von Standard-Verschlüsselung für S3-Buckets. Für Anforderungen zum Konfigurieren des Standardverschlüsselungsverhaltens werden Standardgebühren für Amazon-S3-Anforderungen berechnet. Informationen zu Preisen finden Sie unter Amazon S3 – Preise. Für SSE - KMS und DSSE - AWS KMS fallen Gebühren anKMS, die in den AWS KMS Preisen aufgeführt sind.

Serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) wird für die Standardverschlüsselung nicht unterstützt.

Sie können die Amazon S3-Standardverschlüsselung für einen S3-Bucket konfigurieren, indem Sie die AWS SDKs Amazon S3 S3-Konsole RESTAPI, Amazon S3 und AWS Command Line Interface (AWS CLI) verwenden.

Änderungen, die Sie vor dem Aktivieren der Standardverschlüsselung beachten sollten

Nachdem Sie die Standard-Verschlüsselung für einen Bucket aktiviert haben, gilt das folgende Verschlüsselungsverhalten:

- Es gibt keine Änderung der Verschlüsselung der Objekte, die vor der Aktivierung der Standard-Verschlüsselung im Bucket vorhanden waren.
- Wenn Sie Objekte nach der Aktivierung der Standard-Verschlüsselung hochladen:
  - Wenn Ihre PUT-Abfrage-Header keine Verschlüsselungsinformationen mit einschließen, verwendet Amazon S3 die Standardverschlüsselungseinstellungen des Buckets, um die Objekte zu verschlüsseln.
  - Wenn Ihre PUT-Anfrage-Header Verschlüsselungsinformationen mit einschließen, verwendet Amazon S3 die Verschlüsselungsinformationen der PUT-Anfrage, um Objekte zu verschlüsseln, bevor sie in Amazon S3 gespeichert werden.
- Wenn Sie die KMS Option SSE - KMS oder DSSE - für Ihre Standardverschlüsselungskonfiguration verwenden, unterliegen Sie den Quoten für Anfragen pro Sekunde (RPS) von AWS KMS. Weitere Informationen zu AWS KMS -Kontingenten und zum Anfordern einer Kontingenterhöhung finden Sie unter Kontingente im Entwicklerhandbuch zu AWS Key Management Service .

> **Note**
>
> Objekte, die hochgeladen wurden, bevor die Standardverschlüsselung aktiviert wurde, werden nicht verschlüsselt. Weitere Informationen zum Verschlüsseln vorhandener Objekte finden Sie unter the section called “Festlegen der Standard-Bucket-Verschlüsselung”.

Verwenden der S3-Konsole

Konfigurieren der Standardverschlüsselung für einen Amazon-S3-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des von Ihnen erstellten Buckets aus.
4. Wählen Sie die Registerkarte Eigenschaften aus.
5. Wählen Sie unter Default encryption (Standard-Verschlüsselung) Edit (Bearbeiten) aus.

6. Wählen Sie eine der folgenden Optionen unter Verschlüsselungstyp aus, um die Verschlüsselung zu konfigurieren:

   - Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3)
   - Serverseitige Verschlüsselung mit AWS Key Management Service Schlüsseln (-) SSE KMS
   - Zweischichtige serverseitige Verschlüsselung mit AWS Key Management Service Schlüsseln (-) DSSE KMS

   > **Important**
   >
   > Wenn Sie die KMS Optionen SSE - KMS oder DSSE - für Ihre Standardverschlüsselungskonfiguration verwenden, gelten für Sie die Kontingente für Anfragen pro Sekunde (RPS) von. AWS KMS Weitere Informationen zu AWS KMS Kontingenten und dazu, wie Sie eine Kontingenterhöhung beantragen können, finden Sie unter Kontingente im AWS Key Management Service Entwicklerhandbuch.

   Buckets und neue Objekte werden standardmäßig mit SSE -S3 verschlüsselt, es sei denn, Sie geben einen anderen Standardverschlüsselungstyp für Ihre Buckets an. Weitere Informationen zur Standardverschlüsselung finden Sie unter Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets.

   Weitere Informationen zur Datenverschlüsselung mit der serverseitigen Amazon-S3-Verschlüsselung finden Sie unter Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden (SSE-S3).

7. Wenn Sie sich für serverseitige Verschlüsselung mit AWS Key Management Service Schlüsseln (SSE-KMS) oder Serverseitige Dual-Layer-Verschlüsselung mit AWS Key Management Service Schlüsseln (DSSE- KMS) entschieden haben, gehen Sie wie folgt vor:

   a. Geben Sie unter AWS KMS Schlüssel Ihren KMS Schlüssel auf eine der folgenden Arten an:

      - Um aus einer Liste verfügbarer KMS Schlüssel zu wählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys und wählen Sie Ihren KMSSchlüssel aus der Liste der verfügbaren Schlüssel aus.

        Sowohl der Von AWS verwalteter Schlüssel (`aws/s3`) als auch Ihr vom Kunden verwalteter Schlüssel werden in dieser Liste angezeigt. Weitere Informationen zu vom

Kunden verwalteten Schlüsseln finden Sie unter Kundenschlüssel und AWS Schlüssel im AWS Key Management Service Entwicklerhandbuch.

- Um den KMS Schlüssel einzugebenARN, wählen Sie Enter AWS KMS key ARN und geben Sie Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
- Um einen neuen, vom Kunden verwalteten Schlüssel in der AWS KMS Konsole zu erstellen, wählen Sie Create a KMS key aus.

  Weitere Informationen zum Erstellen eines AWS KMS key finden Sie unter Schlüssel erstellen im AWS Key Management Service Entwicklerhandbuch.

> **Important**
>
> Sie können nur KMS Schlüssel verwenden, die im selben AWS-Region Bucket aktiviert sind. Wenn Sie „Aus Ihren KMS Schlüsseln auswählen“ wählen, listet die S3-Konsole nur 100 KMS Schlüssel pro Region auf. Wenn Sie mehr als 100 KMS Schlüssel in derselben Region haben, können Sie nur die ersten 100 KMS Schlüssel in der S3-Konsole sehen. Um einen KMS Schlüssel zu verwenden, der nicht in der Konsole aufgeführt ist, wählen Sie Enter AWS KMS key ARN und geben Sie den KMS Schlüssel einARN.
> Wenn Sie eine AWS KMS key für die serverseitige Verschlüsselung in Amazon S3 verwenden, müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel. Weitere Informationen zu diesen Schlüsseln finden Sie unter Symmetrische KMS Verschlüsselungsschlüssel im AWS Key Management Service Entwicklerhandbuch.

Weitere Informationen zur Verwendung von SSE - KMS mit Amazon S3 finden Sie unterServerseitige Verschlüsselung mit AWS KMS Schlüsseln verwenden (SSE-) KMS. Weitere Informationen zur Verwendung von DSSE - KMS finden Sie unterthe section called “Serverseitige Dual-Layer-Verschlüsselung (-) DSSE KMS”.

b. Wenn Sie Ihren Bucket so konfigurieren, dass er die Standardverschlüsselung mit SSE - verwendetKMS, können Sie auch einen S3-Bucket-Key aktivieren. S3-Bucket-Keys senken die Kosten für die Verschlüsselung, indem sie den Anforderungsverkehr von Amazon S3 zu verringern AWS KMS. Weitere Informationen finden Sie unter Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys.

Um S3-Bucket-Schlüssel zu verwenden, wählen Sie unter Bucket Key (Bucket-Schlüssel) die Option Enable (Aktivieren).

> **Note**
>
> S3-Bucket-Keys werden für DSSE - nicht unterstütztKMS.

8. Wählen Sie Änderungen speichern.

Mit dem AWS CLI

Diese Beispiele zeigen Ihnen, wie Sie die Standardverschlüsselung mithilfe von SSE -S3 oder mithilfe von SSE - KMS mit einem S3-Bucket-Key konfigurieren.

Weitere Informationen zur Standardverschlüsselung finden Sie unter Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets. Weitere Informationen AWS CLI zur Konfiguration der Standardverschlüsselung finden Sie unter put-bucket-encryption.

Example — Standardverschlüsselung mit SSE -S3

In diesem Beispiel wird die Standardverschlüsselung von Buckets mit von Amazon S3 verwalteten Schlüsseln konfiguriert.

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-encryption-
configuration '{
    "Rules": [
        {
            "ApplyServerSideEncryptionByDefault": {
                "SSEAlgorithm": "AES256"
            }
        }
    ]
}'
```

Example — Standardverschlüsselung mit SSE — KMS unter Verwendung eines S3-Bucket-Keys

In diesem Beispiel wird die Standard-Bucket-Verschlüsselung mit SSE - KMS unter Verwendung eines S3-Bucket-Keys konfiguriert.

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-encryption-
configuration '{
    "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "aws:kms",
                    "KMSMasterKeyID": "KMS-Key-ARN"
                },
                "BucketKeyEnabled": true
            }
        ]
    }'
```

Mit dem REST API

Verwenden Sie den REST API `PutBucketEncryption` Vorgang, um die Standardverschlüsselung zu aktivieren und den Typ der serverseitigen Verschlüsselung auf— SSE -S3KMS, SSE - oder DSSE - festzulegen. KMS

Weitere Informationen finden Sie unter PutBucketEncryptionin der Amazon Simple Storage Service API Reference.

Überwachung der Standardverschlüsselung mit AWS CloudTrail und Amazon EventBridge

> **Important**
>
> Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist in den AWS CloudTrail Protokollen, im S3-Inventar, in der S3-Speicherlinse, in der Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header im AWS Command Line Interface und AWS SDKs verfügbar. Weitere Informationen finden Sie unter Standardverschlüsselung FAQ.

Sie können Konfigurationsanforderungen zur Standardverschlüsselung für Amazon-S3-Buckets mithilfe von AWS CloudTrail -Ereignissen verfolgen. Die folgenden API Ereignisnamen werden in CloudTrail Protokollen verwendet:

- `PutBucketEncryption`
- `GetBucketEncryption`
- `DeleteBucketEncryption`

Sie können auch EventBridge Regeln erstellen, die den CloudTrail Ereignissen für diese API Aufrufe entsprechen. Weitere Informationen zu CloudTrail Ereignissen finden Sie unterAktivieren der Protokollierung für Objekte in einem Bucket mit der Konsole. Weitere Informationen zu EventBridge Ereignissen finden Sie unter Ereignisse von AWS-Services.

Sie können CloudTrail Protokolle für Amazon S3-Aktionen auf Objektebene verwenden, um `POST` Anfragen an Amazon S3 nachzuverfolgen`PUT`. Sie können diese Aktionen verwenden, um zu überprüfen, ob die Standardverschlüsselung zum Verschlüsseln von Objekten verwendet wird, wenn eingehende Anfragen `PUT` keine Verschlüsselungs-Header haben.

Wenn Amazon S3 ein Objekt mit den Einstellungen der Standardverschlüsselung verschlüsselt, enthält das Protokoll eins der folgenden Felder als Name-Wert-Paar: `"SSEApplied":"Default_SSE_S3"`, `"SSEApplied":"Default_SSE_KMS"` oder `"SSEApplied":"Default_DSSE_KMS"`.

Wenn Amazon S3 ein Objekt mit den `PUT`-Verschlüsselungs-Headern verschlüsselt, enthält das Protokoll eins der folgenden Felder als Name-Wert-Paar: `"SSEApplied":"SSE_S3"`, `"SSEApplied":"SSE_KMS"`, `"SSEApplied":"DSSE_KMS"` oder `"SSEApplied":"SSE_C"`.

Bei mehrteiligen Uploads sind diese Informationen in Ihren Betriebsanfragen enthalten. `InitiateMultipartUpload` API Weitere Hinweise zur Verwendung von CloudTrail und finden Sie CloudWatch unter. Protokollierung und Überwachung in Amazon S3

Standardverschlüsselung FAQ

Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. SSE-S3, das den 256-Bit-Advanced Encryption Standard (AES-256) verwendet, wird automatisch auf alle neuen Buckets und auf alle vorhandenen S3-Buckets angewendet, für die noch keine Standardverschlüsselung konfiguriert ist. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist in den AWS CloudTrail Protokollen, im S3-

Inventar, in der S3-Speicherlinse, in der Amazon S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwortheader in den AWS Command Line Interface (AWS CLI) und im verfügbar. AWS SDKs

In den folgenden Abschnitten werden Fragen zu diesem Update beantwortet.

Ändert Amazon S3 die Standardverschlüsselungseinstellungen für meine vorhandenen Buckets, für die bereits die Standardverschlüsselung konfiguriert ist?

Nein. Es gibt keine Änderungen an der Standardverschlüsselungskonfiguration für einen vorhandenen Bucket, für den bereits SSE -S3 oder serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-KMS) konfiguriert ist. Weitere Informationen zum Festlegen des Standardverschlüsselungsverhalten für Buckets finden Sie unter Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets. Weitere Hinweise zu den KMS Verschlüsselungseinstellungen SSE -S3 und SSE - finden Sie unter. Schützen von Daten mit serverseitiger Verschlüsselung

Wird die Standardverschlüsselung für meine vorhandenen Buckets aktiviert, für die keine Standardverschlüsselung konfiguriert ist?

Ja. Amazon S3 konfiguriert jetzt die Standardverschlüsselung für alle vorhandenen unverschlüsselten Buckets, um serverseitige Verschlüsselung mit verwalteten S3-Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für neue Objekte anzuwenden, die in diese Buckets hochgeladen werden. Objekte, die sich bereits in einem vorhandenen nicht verschlüsselten Bucket befinden, werden nicht automatisch verschlüsselt.

Wie kann ich den Standardverschlüsselungsstatus neuer Objekt-Uploads einsehen?

Derzeit können Sie den Standardverschlüsselungsstatus neuer Objekt-Uploads in den AWS CloudTrail Protokollen, S3 Inventory und S3 Storage Lens, der Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header in den AWS Command Line Interface (AWS CLI) und dem AWS SDKs anzeigen.

- Informationen zum Anzeigen Ihrer CloudTrail Ereignisse finden Sie im AWS CloudTrail Benutzerhandbuch unter CloudTrail Ereignisse in der CloudTrail Konsole anzeigen. CloudTrail Protokolle ermöglichen die API Nachverfolgung von `PUT` und `POST` Anfragen an Amazon S3. Wenn die Standardverschlüsselung zur Verschlüsselung von Objekten in Ihren Buckets verwendet wird, enthalten die CloudTrail Protokolle für `PUT` und `POST` API Anfragen das folgende Feld als Name-Wert-Paar:. `"SSEApplied":"Default_SSE_S3"`
- Um den automatischen Verschlüsselungsstatus neuer Objektuploads in S3 Inventar anzuzeigen, konfigurieren Sie einen S3-Inventarbericht, der das Feld Verschlüsselungsmetadaten enthält, und

sehen Sie sich dann den Verschlüsselungsstatus jedes neuen Objekts im Bericht an. Weitere Informationen finden Sie unter Einrichtung von Amazon S3 Inventory.

- Um den automatischen Verschlüsselungsstatus für neue Objektuploads in S3 Storage Lens anzuzeigen, konfigurieren Sie ein S3-Storage-Lens-Dashboard und sehen Sie sich die Metriken Verschlüsselte Bytes und Anzahl verschlüsselter Objekte in der Kategorie Datenschutz des Dashboards an. Weitere Informationen erhalten Sie unter Verwenden der S3-Konsole und Anzeigen von S3-Storage-Lens-Metriken in den Dashboards.
- Wenn Sie den Status der automatischen Verschlüsselung auf Bucket-Ebene in der Amazon-S3-Konsole anzeigen möchten, überprüfen Sie die Standardverschlüsselung Ihrer Amazon-S3-Buckets in der Amazon-S3-Konsole. Weitere Informationen finden Sie unter Konfigurieren der Standardverschlüsselung.
- Um den Status der automatischen Verschlüsselung als zusätzlichen Amazon S3 API S3-Antwort-Header in den AWS Command Line Interface (AWS CLI) und dem anzuzeigen AWS SDKs, überprüfen Sie den Antwort-Header, `x-amz-server-side-encryption` wenn Sie Objektaktionen verwendenAPIs, z. B. PutObjectund GetObject.

Was muss ich tun, um von dieser Änderung zu profitieren?

Sie müssen keine Änderungen an Ihren bestehenden Anwendungen vornehmen. Da die Standardverschlüsselung für alle Ihre Buckets aktiviert ist, werden alle neuen Objekte, die auf Amazon S3 hochgeladen werden, automatisch verschlüsselt.

Kann ich die Verschlüsselung für die neuen Objekte deaktivieren, die in meinen Bucket geschrieben werden?

Nein. SSE-S3 ist die neue Basisverschlüsselungsebene, die auf alle neuen Objekte angewendet wird, die in Ihren Bucket hochgeladen werden. Sie können die Verschlüsselung für das Hochladen neuer Objekte nicht mehr deaktivieren.

Werden sich meine Gebühren ändern?

Nein. Die Standardverschlüsselung mit SSE -S3 ist ohne zusätzliche Kosten verfügbar. Speicherplatz, Anfragen und andere S3-Funktionen werden Ihnen wie gewohnt in Rechnung gestellt. Preisinformationen finden Sie unter Amazon S3 – Preise.

Verschlüsselt Amazon S3 meine vorhandenen unverschlüsselten Objekte?

Nein. Ab dem 5. Januar 2023 verschlüsselt Amazon S3 nur neue Objekt-Uploads automatisch. Zum Verschlüsseln bestehender Objekte können Sie S3-Batch-Operationen verwenden, um

verschlüsselte Kopien Ihrer Objekte zu erstellen. Diese verschlüsselten Kopien behalten die vorhandenen Objektdaten und den Namen bei und werden mit den von Ihnen angegebenen Verschlüsselungsschlüsseln verschlüsselt. Weitere Informationen finden Sie unter Encrypting objects with Amazon S3 Batch Operations im AWS Storage Blog.

Ich habe vor dieser Version die Verschlüsselung für meine Buckets nicht aktiviert. Muss ich die Art und Weise, wie ich auf Objekte zugreife, ändern?

Nein. Die Standardverschlüsselung mit SSE -S3 verschlüsselt Ihre Daten automatisch, während sie in Amazon S3 geschrieben werden, und entschlüsselt sie für Sie, wenn Sie darauf zugreifen. Die Art und Weise, wie Sie auf automatisch verschlüsselte Objekte zugreifen, ändert sich nicht.

Muss ich die Art und Weise, wie ich auf meine clientseitig verschlüsselten Objekte zugreife, ändern?

Nein. Alle clientseitig verschlüsselten Objekte, die vor dem Hochladen in Amazon S3 verschlüsselt werden, kommen als verschlüsselte Geheimtextobjekte in Amazon S3 an. Diese Objekte verfügen jetzt über eine zusätzliche SSE -S3-Verschlüsselungsebene. Ihre Workloads, die clientseitig verschlüsselte Objekte verwenden, erfordern keine Änderungen an Ihren Client-Services oder Autorisierungseinstellungen.

> **Note**
>
> HashiCorp Terraform-Benutzer, die keine aktualisierte Version des AWS Anbieters verwenden, stellen möglicherweise eine unerwartete Abweichung fest, nachdem sie neue S3-Buckets ohne benutzerdefinierte Verschlüsselungskonfiguration erstellt haben. Um diese Abweichung zu vermeiden, aktualisieren Sie Ihre Terraform AWS Provider-Version auf eine der folgenden Versionen: beliebig 4.x veröffentlichen, 3.76.1, oder 2.70.4.

Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden (SSE-S3)

> **Important**
>
> Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-

Buckets und für das Hochladen neuer Objekte ist in den AWS CloudTrail Protokollen, im S3-Inventar, in der S3-Speicherlinse, in der Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header im AWS Command Line Interface und AWS SDKs verfügbar. Weitere Informationen finden Sie unter Standardverschlüsselung FAQ.

Alle neuen Objekt-Uploads in Amazon S3 S3-Buckets werden standardmäßig mit serverseitiger Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verschlüsselt.

Die serverseitige Verschlüsselung schützt Daten im Ruhezustand. Amazon S3 verschlüsselt jedes Objekt mit einem eindeutigen Schlüssel. Als zusätzliche Sicherheit verschlüsselt es den Schlüssel selbst mit einem Schlüssel, der regelmäßig rotiert. Die serverseitige Amazon S3 S3-Verschlüsselung verwendet den 256-Bit-Advanced Encryption Standard Galois/Counter Mode (AES-GCM), um alle hochgeladenen Objekte zu verschlüsseln.

Für die Verwendung der serverseitigen Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) fallen keine zusätzlichen Gebühren an. Für Anfragen zum Konfigurieren der Standard-Verschlüsselungsfunktion werden jedoch Standardgebühren für Amazon-S3-Anfragen berechnet. Informationen zu Preisen finden Sie unter Amazon S3 – Preise.

Wenn Sie möchten, dass Ihre Daten-Uploads ausschließlich mit von Amazon S3 verwalteten Schlüsseln verschlüsselt werden, können Sie die folgende Bucket-Richtlinie verwenden. Beispielsweise verweigert die folgende Bucket-Richtlinie Berechtigungen zum Hochladen von Objekten, wenn die Anforderung nicht den `x-amz-server-side-encryption`-Header enthält, der eine serverseitige Verschlüsselung anfordert:

```
{
  "Version": "2012-10-17",
  "Id": "PutObjectPolicy",
  "Statement": [
    {
      "Sid": "DenyObjectsThatAreNotSSES3",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition": {
        "StringNotEquals": {
          "s3:x-amz-server-side-encryption": "AES256"
        }
```

```
      }
    }
   ]
}
```

> **Note**
>
> Die serverseitige Verschlüsselung verschlüsselt nur die Objektdaten, nicht die Metadaten des Objekts.

APIUnterstützung für serverseitige Verschlüsselung

Für alle Amazon-S3-Buckets ist die Verschlüsselung standardmäßig konfiguriert und alle neuen Objekte, die in einen S3-Bucket hochgeladen werden, werden im Ruhezustand automatisch verschlüsselt. Die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Standardverschlüsselungskonfiguration für jeden Bucket in Amazon S3. Um einen anderen Verschlüsselungstyp zu verwenden, können Sie entweder die Art der serverseitigen Verschlüsselung angeben, die in Ihren S3-PUT-Anfragen verwendet werden soll, oder Sie können die Standardverschlüsselungskonfiguration im Ziel-Bucket festlegen.

Wenn Sie in Ihren PUT Anfragen einen anderen Verschlüsselungstyp angeben möchten, können Sie serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln (-), zweischichtige serverseitige Verschlüsselung mit Schlüsseln (SSE-KMS) oder serverseitige Verschlüsselung mit vom Kunden bereitgestellten AWS KMS Schlüsseln (DSSE-CKMS) verwenden. SSE Wenn Sie im Ziel-Bucket eine andere Standardverschlüsselungskonfiguration festlegen möchten, können Sie - oder - verwenden. SSE KMS DSSE KMS

Um die serverseitige Verschlüsselung mithilfe der Objekterstellung zu konfigurieren RESTAPIs, müssen Sie den `x-amz-server-side-encryption` Anforderungsheader angeben. Hinweise zu dem finden Sie REST APIs unterUnter Verwendung der REST API.

Die folgenden Amazon S3 APIs unterstützen diesen Header:

- PUTOperationen — Geben Sie den Anforderungsheader an, wenn Sie Daten mit dem PUT API hochladen. Weitere Informationen finden Sie unter PUTObjekt.
- Mehrteiligen Upload initiieren — Geben Sie den Header in der Initialisierungsanforderung an, wenn Sie große Objekte mithilfe des mehrteiligen Upload-Vorgangs hochladen. API Weitere Informationen finden Sie unter Mehrteiligen Upload initiieren.

- COPYOperationen — Wenn Sie ein Objekt kopieren, haben Sie sowohl ein Quell- als auch ein Zielobjekt. Weitere Informationen finden Sie unter PUTObjekt — Kopieren.

> **Note**
>
> Wenn Sie eine POST-Operation für das Hochladen eines Objekts verwenden, anstatt den Anforderungs-Header anzugeben, stellen Sie die gleichen Informationen in den Formularfeldern bereit. Weitere Informationen finden Sie unter POSTObjekt.

Sie stellen AWS SDKs auch einen Wrapper bereitAPIs, mit dem Sie serverseitige Verschlüsselung anfordern können. Sie können den auch verwenden, AWS Management Console um Objekte hochzuladen und serverseitige Verschlüsselung anzufordern.

Weitere Informationen finden Sie unter AWS KMS -Konzepte im Entwicklerhandbuch zu AWS Key Management Service .

Themen

- Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) angeben

Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) angeben

Für alle Amazon-S3-Buckets ist die Verschlüsselung standardmäßig konfiguriert und alle neuen Objekte, die in einen S3-Bucket hochgeladen werden, werden im Ruhezustand automatisch verschlüsselt. Die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Standardverschlüsselungskonfiguration für jeden Bucket in Amazon S3. Um einen anderen Verschlüsselungstyp zu verwenden, können Sie entweder die Art der serverseitigen Verschlüsselung angeben, die in Ihren S3-PUT-Anfragen verwendet werden soll, oder Sie können die Standardverschlüsselungskonfiguration im Ziel-Bucket festlegen.

Wenn Sie in Ihren PUT Anfragen einen anderen Verschlüsselungstyp angeben möchten, können Sie serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln (-), zweischichtige serverseitige Verschlüsselung mit Schlüsseln (SSE-KMS) oder serverseitige Verschlüsselung mit vom Kunden bereitgestellten AWS KMS Schlüsseln (DSSE-CKMS) verwenden. SSE Wenn Sie im Ziel-Bucket eine andere Standardverschlüsselungskonfiguration festlegen möchten, können Sie - oder - verwenden. SSE KMS DSSE KMS

Sie können SSE -S3 mithilfe der S3-Konsole, REST APIs AWS SDKs, und AWS Command Line Interface (AWS CLI) angeben. Weitere Informationen finden Sie unter Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets.

Verwenden der S3-Konsole

In diesem Thema wird beschrieben, wie der Verschlüsselungstyp eines Objekts mit der AWS Management Console festgelegt oder geändert wird. Wenn Sie ein Objekt unter Verwendung der Konsole kopieren, kopiert Amazon S3 das Objekt unverändert. Wenn das Quellobjekt verschlüsselt ist, wird das Zielobjekt demnach ebenfalls verschlüsselt. Sie können mit der Konsole die Verschlüsselung für ein Objekt hinzufügen oder ändern.

> **Note**
>
> - Sie können die Verschlüsselung eines Objekts ändern, wenn Ihr Objekt weniger als 5 GB groß ist. Wenn Ihr Objekt größer als 5 GB ist, müssen Sie AWS CLIoder verwenden, AWS SDKsum die Verschlüsselung eines Objekts zu ändern.
> - Eine Liste der zusätzlichen Berechtigungen, die zum Ändern der Verschlüsselung eines Objekts erforderlich sind, finden Sie unterthe section called “Erforderliche Berechtigungen für API S3-Operationen”. Richtlinien, die diese Berechtigung gewähren, finden Sie beispielsweise unterthe section called “Beispiele für identitätsbasierte Richtlinien”.
> - Wenn Sie die Verschlüsselung eines Objekts ändern, wird ein neues Objekt erstellt, um das alte zu ersetzen. Wenn S3-Versioning aktiviert ist, wird eine neue Version des Objekts erstellt, und das vorhandene Objekt wird zu einer älteren Version. Die Rolle, die die Eigenschaft ändert, wird auch Besitzer des neuen Objekts (oder der neuen Objektversion).

So ändern Sie die Verschlüsselung für ein Objekt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im Navigationsbereich Buckets und dann den Tab Allzweck-Buckets aus. Navigieren Sie zu dem Amazon S3 S3-Bucket oder -Ordner, der die Objekte enthält, die Sie ändern möchten.
3. Aktivieren Sie das Kontrollkästchen für die Objekte, die Sie ändern möchten.
4. Wählen Sie im Menü Aktionen aus der daraufhin angezeigten Optionsliste die Option Serverseitige Verschlüsselung bearbeiten aus.

5. Scrollen Sie zum Abschnitt Serverseitige Verschlüsselung.
6. Wählen Sie unter Verschlüsselungseinstellungen die Option Verwenden von Bucket-Einstellungen für die Standardverschlüsselung oder Überschreiben der Bucket-Einstellungen für die Standardverschlüsselung aus.
7. Wenn Sie Überschreiben der Bucket-Einstellungen für die Standardverschlüsselung ausgewählt haben, konfigurieren Sie die folgenden Verschlüsselungseinstellungen.

   - Wählen Sie unter Verschlüsselungstyp die Option Serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) aus. SSE-S3 verwendet eine der stärksten Blockchiffren — den 256-Bit-Advanced Encryption Standard (-256), um jedes Objekt zu verschlüsseln. AES Weitere Informationen finden Sie unter Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden (SSE-S3).
8. Wählen Sie unter Zusätzliche Kopiereinstellungen aus, ob Sie Quelleinstellungen kopieren, Keine Einstellungen angeben oder Einstellungen angeben möchten. Quelleinstellungen kopieren ist die Standardoption. Wenn Sie das Objekt nur ohne die Quelleinstellungsattribute kopieren möchten, wählen Sie „Keine Einstellungen angeben“. Wählen Sie „Einstellungen angeben“, um Einstellungen für SpeicherklasseACLs, Objekt-Tags, Metadaten, serverseitige Verschlüsselung und zusätzliche Prüfsummen anzugeben.
9. Wählen Sie Änderungen speichern.

> **Note**
>
> Diese Aktion wendet auf alle angegebenen Objekte Verschlüsselung an. Warten Sie beim Verschlüsseln von Ordnern, bis die Speicheroperation abgeschlossen ist, bevor Sie dem Ordner neue Objekte hinzufügen.

Unter Verwendung der REST API

Zum Zeitpunkt der Objekterstellung, d. h. wenn Sie ein neues Objekt hochladen oder eine Kopie eines vorhandenen Objekts erstellen, können Sie angeben, ob Amazon S3 Ihre Daten mit Amazon S3-verwalteten Schlüsseln (SSE-S3) verschlüsseln soll, indem Sie der Anfrage den Header hinzufügen. `x-amz-server-side-encryption` Setzen Sie den Wert des Headers auf den Verschlüsselungsalgorithmus `AES256`, den Amazon S3 unterstützt. Amazon S3 bestätigt, dass Ihr Objekt mit SSE -S3 gespeichert ist, indem es den Antwort-Header `x-amz-server-side-encryption` zurückgibt.

Die folgenden REST API Upload-Operationen akzeptieren den `x-amz-server-side-encryption` Anforderungsheader.

- PUTObjekt
- PUTObjekt — Kopieren
- POSTObjekt
- Initiieren eines mehrteiligen Uploads

Wenn Sie große Objekte mithilfe des mehrteiligen API Upload-Vorgangs hochladen, können Sie die serverseitige Verschlüsselung angeben, indem Sie den `x-amz-server-side-encryption` Header zur Anforderung Multipart Upload initiieren hinzufügen. Beim Kopieren eines vorhandenen Objekts wird das Zielobjekt unabhängig davon, ob das Quellobjekt verschlüsselt ist, nur dann verschlüsselt, wenn Sie die serverseitige Verschlüsselung explizit anfordern.

Die Antwortheader der folgenden REST API Operationen geben den Header zurück, wenn ein Objekt mit `x-amz-server-side-encryption` -S3 gespeichert wird. SSE

- PUTObjekt
- PUTObjekt — Kopieren
- POSTObjekt
- Initiieren eines mehrteiligen Uploads
- Upload Part
- Hochladen eines Teiluploads – Kopieren
- Abschließen eines mehrteiligen Uploads
- Get Object
- Head Object

> **Note**
>
> Senden Sie keine Header für `GET` Verschlüsselungsanforderungen für Anfragen und `HEAD` Anfragen, wenn Ihr Objekt SSE -S3 verwendet. Andernfalls erhalten Sie HTTP den Statuscode 400 (Bad Request).

Verwenden Sie den AWS SDKs

Bei der Verwendung können Sie Amazon S3 auffordern AWS SDKs, serverseitige Verschlüsselung mit von Amazon S3 verwalteten Verschlüsselungsschlüsseln (SSE-S3) zu verwenden. Dieser Abschnitt enthält Beispiele für die Verwendung von AWS SDKs in mehreren Sprachen. Informationen zu anderen SDKs finden Sie unter Beispielcode und Bibliotheken.

Java

Wenn Sie ein Objekt AWS SDK for Java mit dem hochladen, können Sie es mit SSE - S3 verschlüsseln. Um eine serverseitige Verschlüsselung anzufordern, legen Sie mit der `ObjectMetadata`-Eigenschaft der `PutObjectRequest` den Anforderungs-Header `x-amz-server-side-encryption` fest. Wenn Sie die Methode `putObject()` des `AmazonS3Client`-Clients aufrufen, verschlüsselt und speichert Amazon S3 die Daten.

Sie können auch die SSE -S3-Verschlüsselung anfordern, wenn Sie Objekte mit dem mehrteiligen Upload-Vorgang hochladen: API

- Wenn Sie den mehrteiligen API Upload-Vorgang auf hoher Ebene verwenden, verwenden Sie die `TransferManager` Methoden, um Objekte beim Hochladen serverseitig zu verschlüsseln. Sie können eine beliebige der Upload-Methoden nutzen, die `ObjectMetadata` als Parameter entgegennehmen. Weitere Informationen finden Sie unter Hochladen eines Objekts mit Multipart-Upload.
- Wenn Sie den mehrteiligen API Upload-Vorgang auf niedriger Ebene verwenden, geben Sie die serverseitige Verschlüsselung an, wenn Sie den mehrteiligen Upload starten. Sie fügen die Eigenschaft `ObjectMetadata` beim Aufruf der Methode `InitiateMultipartUploadRequest.setObjectMetadata()` hinzu. Weitere Informationen finden Sie unter Verwenden von AWS SDKs (APILow-Level).

Sie können den Verschlüsselungsstatus eines Objekts (Verschlüsseln eines unverschlüsselten Objekts oder Entschlüsseln eines verschlüsselten Objekts) nicht direkt ändern. Um den Verschlüsselungsstatus eines Objekts zu ändern, erstellen Sie eine Kopie des Objekts, geben dabei den gewünschten Verschlüsselungsstatus der Kopie an und löschen dann das Originalobjekt. Amazon S3 verschlüsselt das kopierte Objekt nur, wenn Sie explizit eine serverseitige Verschlüsselung anfordern. Um die Verschlüsselung des kopierten Objekts über Java anzufordernAPI, verwenden Sie die `ObjectMetadata` Eigenschaft, um die serverseitige Verschlüsselung in der anzugeben. `CopyObjectRequest`

Example Beispiel

Das folgende Beispiel veranschaulicht, wie Sie die serverseitige Verschlüsselung unter Verwendung des AWS SDK for Java festlegen. Es veranschaulicht, wie Sie die folgenden Aufgaben ausführen:

- Laden Sie ein neues Objekt hoch, indem Sie SSE -S3 verwenden.
- Ändern des Verschlüsselungsstatus eines Objekts (in diesem Beispiel Verschlüsseln eines zuvor unverschlüsselten Objekts) durch Anfertigen einer Kopie des Objekts
- Überprüfen des Verschlüsselungsstatus des Objekts

Weitere Informationen zur serverseitigen Verschlüsselung finden Sie unter Unter Verwendung der REST API. Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.internal.SSEResultBase;
import com.amazonaws.services.s3.model.*;

import java.io.ByteArrayInputStream;

public class SpecifyServerSideEncryption {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyNameToEncrypt = "*** Key name for an object to upload and encrypt
 ***";
        String keyNameToCopyAndEncrypt = "*** Key name for an unencrypted object to
 be encrypted by copying ***";
        String copiedObjectKeyName = "*** Key name for the encrypted copy of the
 unencrypted object ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
```

```
                    .withRegion(clientRegion)
                    .withCredentials(new ProfileCredentialsProvider())
                    .build();

            // Upload an object and encrypt it with SSE.
            uploadObjectWithSSEEncryption(s3Client, bucketName, keyNameToEncrypt);

            // Upload a new unencrypted object, then change its encryption state
            // to encrypted by making a copy.
            changeSSEEncryptionStatusByCopying(s3Client,
                    bucketName,
                    keyNameToCopyAndEncrypt,
                    copiedObjectKeyName);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }

    private static void uploadObjectWithSSEEncryption(AmazonS3 s3Client, String
bucketName, String keyName) {
        String objectContent = "Test object encrypted with SSE";
        byte[] objectBytes = objectContent.getBytes();

        // Specify server-side encryption.
        ObjectMetadata objectMetadata = new ObjectMetadata();
        objectMetadata.setContentLength(objectBytes.length);

objectMetadata.setSSEAlgorithm(ObjectMetadata.AES_256_SERVER_SIDE_ENCRYPTION);
        PutObjectRequest putRequest = new PutObjectRequest(bucketName,
                keyName,
                new ByteArrayInputStream(objectBytes),
                objectMetadata);

        // Upload the object and check its encryption status.
        PutObjectResult putResult = s3Client.putObject(putRequest);
        System.out.println("Object \"" + keyName + "\" uploaded with SSE.");
        printEncryptionStatus(putResult);
    }
```

```
    private static void changeSSEEncryptionStatusByCopying(AmazonS3 s3Client,
            String bucketName,
            String sourceKey,
            String destKey) {
        // Upload a new, unencrypted object.
        PutObjectResult putResult = s3Client.putObject(bucketName, sourceKey,
 "Object example to encrypt by copying");
        System.out.println("Unencrypted object \"" + sourceKey + "\" uploaded.");
        printEncryptionStatus(putResult);

        // Make a copy of the object and use server-side encryption when storing the
        // copy.
        CopyObjectRequest request = new CopyObjectRequest(bucketName,
                sourceKey,
                bucketName,
                destKey);
        ObjectMetadata objectMetadata = new ObjectMetadata();

 objectMetadata.setSSEAlgorithm(ObjectMetadata.AES_256_SERVER_SIDE_ENCRYPTION);
        request.setNewObjectMetadata(objectMetadata);

        // Perform the copy operation and display the copy's encryption status.
        CopyObjectResult response = s3Client.copyObject(request);
        System.out.println("Object \"" + destKey + "\" uploaded with SSE.");
        printEncryptionStatus(response);

        // Delete the original, unencrypted object, leaving only the encrypted copy
 in
        // Amazon S3.
        s3Client.deleteObject(bucketName, sourceKey);
        System.out.println("Unencrypted object \"" + sourceKey + "\" deleted.");
    }

    private static void printEncryptionStatus(SSEResultBase response) {
        String encryptionStatus = response.getSSEAlgorithm();
        if (encryptionStatus == null) {
            encryptionStatus = "Not encrypted with SSE";
        }
        System.out.println("Object encryption status is: " + encryptionStatus);
    }
}
```

.NET

Wenn Sie ein Objekt hochladen, können Sie Amazon S3 dazu anweisen, es zu verschlüsseln. Um den Verschlüsselungsstatus eines vorhandenen Objekts zu ändern, erstellen Sie eine Kopie des Objekts und löschen dann das Quellobjekt. Beachten Sie, dass die Kopieroperation das Ziel nur verschlüsselt, wenn Sie auf dem Zielobjekt ausdrücklich eine serverseitige Verschlüsselung anfordern. Um SSE -S3 in der anzugeben`CopyObjectRequest`, fügen Sie Folgendes hinzu:

```
ServerSideEncryptionMethod = ServerSideEncryptionMethod.AES256
```

Ein funktionierendes Beispiel, das zeigt, wie ein Objekt kopiert wird, finden Sie unter Unter Verwendung der AWS SDKs.

Das folgende Beispiel lädt ein Objekt hoch. In der Anfrage weist das Beispiel Amazon S3 dazu an, das Objekt zu verschlüsseln. Das Beispiel ruft dann Objekt-Metadaten ab und überprüft die verwendete Verschlüsselungsmethode. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem AWS SDK for. NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class SpecifyServerSideEncryptionTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** key name for object created ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            WritingAnObjectAsync().Wait();
        }
```

```
        static async Task WritingAnObjectAsync()
        {
            try
            {
                var putRequest = new PutObjectRequest
                {
                    BucketName = bucketName,
                    Key = keyName,
                    ContentBody = "sample text",
                    ServerSideEncryptionMethod = ServerSideEncryptionMethod.AES256
                };

                var putResponse = await client.PutObjectAsync(putRequest);

                // Determine the encryption state of an object.
                GetObjectMetadataRequest metadataRequest = new
 GetObjectMetadataRequest
                {
                    BucketName = bucketName,
                    Key = keyName
                };
                GetObjectMetadataResponse response = await
 client.GetObjectMetadataAsync(metadataRequest);
                ServerSideEncryptionMethod objectEncryption =
 response.ServerSideEncryptionMethod;

                Console.WriteLine("Encryption method used: {0}",
 objectEncryption.ToString());
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered ***. Message:'{0}' when writing
 an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
    }
}
```

PHP

In diesem Thema wird gezeigt, wie Sie Klassen aus Version 3 von verwenden AWS SDK for PHP , um SSE -S3 zu Objekten hinzuzufügen, die Sie auf Amazon S3 hochladen. Weitere Informationen über die AWS SDK für Ruby API finden Sie unter AWS SDKfür Ruby — Version 2.

Verwenden Sie die Methode Aws\ S3\ S3Client:: putObject (), um ein Objekt auf Amazon S3 hochzuladen. Um Ihrer Upload-Anfrage den Anfrage-Header `x-amz-server-side-encryption` hinzuzufügen, geben Sie den Parameter `ServerSideEncryption` mit dem Wert `AES256` an, wie im folgenden Codebeispiel veranschaulicht. Weitere Informationen zur serverseitigen Verschlüsselungsanforderungen finden Sie unter Unter Verwendung der REST API.

```
 require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

// $filepath should be an absolute path to a file on disk.
$filepath = '*** Your File Path ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Upload a file with server-side encryption.
$result = $s3->putObject([
    'Bucket'               => $bucket,
    'Key'                  => $keyname,
    'SourceFile'           => $filepath,
    'ServerSideEncryption' => 'AES256',
]);
```

Amazon S3 gibt als Antwort den `x-amz-server-side-encryption`-Header mit dem Wert zurück, den der Verschlüsselungsalgorithmus für die Verschlüsselung Ihrer Objektdaten verwendet hat.

Wenn Sie große Objekte mithilfe des mehrteiligen API Upload-Vorgangs hochladen, können Sie SSE -S3 für die Objekte, die Sie hochladen, wie folgt angeben:

- Wenn Sie den mehrteiligen API Upload-Vorgang auf niedriger Ebene verwenden, geben Sie beim Aufrufen der Methode Aws\ S3\ S3Client:: () die serverseitige Verschlüsselung an. createMultipartUpload Um Ihrer Upload-Anfrage den Anfrage-Header `x-amz-server-side-encryption` hinzuzufügen, geben Sie für den Parameter `array` den `ServerSideEncryption`-Schlüssel mit dem Wert `AES256` an. Weitere Hinweise zum mehrteiligen Upload-Vorgang auf niedriger Ebene finden Sie unter. API Verwenden von AWS SDKs (APILow-Level)
- Wenn Sie den mehrteiligen API Upload-Vorgang auf hoher Ebene verwenden, geben Sie die serverseitige Verschlüsselung mithilfe des `ServerSideEncryption` Vorgangsparameters an. CreateMultipartUploadAPI Ein Beispiel für die Verwendung der `setOption()` Methode mit dem mehrteiligen API Upload-Vorgang auf hoher Ebene finden Sie unter. Hochladen eines Objekts mit Multipart-Upload

Um den Verschlüsselungsstatus eines vorhandenen Objekts zu ermitteln, rufen Sie die Objektmetadaten ab, indem Sie die Methode Aws\ S3\ S3Client:: headObject () aufrufen, wie im folgenden PHP Codebeispiel gezeigt.

```
 require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';
$keyname = '*** Your Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Check which server-side encryption algorithm is used.
$result = $s3->headObject([
    'Bucket' => $bucket,
    'Key'    => $keyname,
]);
echo $result['ServerSideEncryption'];
```

Um den Verschlüsselungsstatus eines vorhandenen Objekts zu ändern, erstellen Sie mithilfe der Methode Aws\ S3\ S3Client:: copyObject () eine Kopie des Objekts und löschen Sie das Quellobjekt. Standardmäßig verschlüsselt `copyObject()` das Ziel nicht, es sei denn, Sie fordern explizit die serverseitige Verschlüsselung des Zielobjekts an, indem Sie den Parameter `ServerSideEncryption` mit dem Wert `AES256` angeben. Das folgende PHP Codebeispiel erstellt eine Kopie eines Objekts und fügt dem kopierten Objekt serverseitige Verschlüsselung hinzu.

```
 require 'vendor/autoload.php';

use Aws\S3\S3Client;

$sourceBucket = '*** Your Source Bucket Name ***';
$sourceKeyname = '*** Your Source Object Key ***';

$targetBucket = '*** Your Target Bucket Name ***';
$targetKeyname = '*** Your Target Object Key ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);

// Copy an object and add server-side encryption.
$s3->copyObject([
    'Bucket'               => $targetBucket,
    'Key'                  => $targetKeyname,
    'CopySource'           => "$sourceBucket/$sourceKeyname",
    'ServerSideEncryption' => 'AES256',
]);
```

Weitere Informationen finden Sie unter den folgenden Themen:

- AWS SDK for PHP für Amazon S3 Aws\ S3\ S3Client-Klasse
- AWS SDK for PHP -Dokumentation:

Ruby

Wenn Sie die AWS SDK for Ruby zum Hochladen eines Objekts verwenden, können Sie mit SSE-S3 angeben, dass das Objekt im Ruhezustand verschlüsselt gespeichert wird. Wenn Sie das Objekt zurücklesen, wird es automatisch entschlüsselt.

Das folgende Beispiel für AWS SDK for Ruby Version 3 zeigt, wie angegeben wird, dass eine auf Amazon S3 hochgeladene Datei im Ruhezustand verschlüsselt wird.

```
require 'aws-sdk-s3'

# Wraps Amazon S3 object actions.
class ObjectPutSseWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  def put_object_encrypted(object_content, encryption)
    @object.put(body: object_content, server_side_encryption: encryption)
    true
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't put your content to #{object.key}. Here's why: #{e.message}"
    false
  end
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-encrypted-content"
  object_content = "This is my super-secret content."
  encryption = "AES256"

  wrapper = ObjectPutSseWrapper.new(Aws::S3::Object.new(bucket_name,
 object_content))
  return unless wrapper.put_object_encrypted(object_content, encryption)

  puts "Put your content into #{bucket_name}:#{object_key} and encrypted it with
 #{encryption}."
end
```

```
run_demo if $PROGRAM_NAME == __FILE__
```

Das folgende Beispiel zeigt, wie Sie den Verschlüsselungsstatus eines vorhandenen Objekts bestimmen.

```
require 'aws-sdk-s3'

# Wraps Amazon S3 object actions.
class ObjectGetEncryptionWrapper
  attr_reader :object

  # @param object [Aws::S3::Object] An existing Amazon S3 object.
  def initialize(object)
    @object = object
  end

  # Gets the object into memory.
  #
  # @return [Aws::S3::Types::GetObjectOutput, nil] The retrieved object data if
 successful; otherwise nil.
  def object
    @object.get
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't get object #{@object.key}. Here's why: #{e.message}"
  end
end

# Example usage:
def run_demo
  bucket_name = "amzn-s3-demo-bucket"
  object_key = "my-object.txt"

  wrapper = ObjectGetEncryptionWrapper.new(Aws::S3::Object.new(bucket_name,
 object_key))
  obj_data = wrapper.get_object
  return unless obj_data

  encryption = obj_data.server_side_encryption.nil? ? 'no' :
 obj_data.server_side_encryption
  puts "Object #{object_key} uses #{encryption} encryption."
end
```

```
run_demo if $PROGRAM_NAME == __FILE__
```

Wenn die serverseitige Verschlüsselung für das in Amazon S3 gespeicherte Objekt nicht verwendet wird, gibt die Methode `null` zurück.

Um den Verschlüsselungsstatus eines vorhandenen Objekts zu ändern, erstellen Sie eine Kopie des Objekts und löschen dann das Quellobjekt. Standardmäßig verschlüsseln die Methoden zum Kopieren das Ziel nicht, es sei denn, Sie fordern explizit die serverseitige Verschlüsselung an. Sie können die Verschlüsselung des Zielobjekts anfordern, indem Sie den Wert `server_side_encryption` im Hash-Argument der Option angeben, wie im folgenden Ruby-Codebeispiel gezeigt. Das Codebeispiel zeigt, wie ein Objekt kopiert und die Kopie mit SSE-S3 verschlüsselt wird.

```
require 'aws-sdk-s3'

# Wraps Amazon S3 object actions.
class ObjectCopyEncryptWrapper
  attr_reader :source_object

  # @param source_object [Aws::S3::Object] An existing Amazon S3 object. This is
 used as the source object for
  #                                        copy actions.
  def initialize(source_object)
    @source_object = source_object
  end

  # Copy the source object to the specified target bucket, rename it with the target
 key, and encrypt it.
  #
  # @param target_bucket [Aws::S3::Bucket] An existing Amazon S3 bucket where the
 object is copied.
  # @param target_object_key [String] The key to give the copy of the object.
  # @return [Aws::S3::Object, nil] The copied object when successful; otherwise,
 nil.
  def copy_object(target_bucket, target_object_key, encryption)
    @source_object.copy_to(bucket: target_bucket.name, key: target_object_key,
 server_side_encryption: encryption)
    target_bucket.object(target_object_key)
  rescue Aws::Errors::ServiceError => e
    puts "Couldn't copy #{@source_object.key} to #{target_object_key}. Here's why:
 #{e.message}"
  end
```

```
end

# Example usage:
def run_demo
  source_bucket_name = "amzn-s3-demo-bucket1"
  source_key = "my-source-file.txt"
  target_bucket_name = "amzn-s3-demo-bucket2"
  target_key = "my-target-file.txt"
  target_encryption = "AES256"

  source_bucket = Aws::S3::Bucket.new(source_bucket_name)
  wrapper = ObjectCopyEncryptWrapper.new(source_bucket.object(source_key))
  target_bucket = Aws::S3::Bucket.new(target_bucket_name)
  target_object = wrapper.copy_object(target_bucket, target_key, target_encryption)
  return unless target_object

  puts "Copied #{source_key} from #{source_bucket_name} to
 #{target_object.bucket_name}:#{target_object.key} and "\
       "encrypted the target with #{target_object.server_side_encryption}
 encryption."
end

run_demo if $PROGRAM_NAME == __FILE__
```

Mit dem AWS CLI

Verwenden Sie das folgende Beispiel, um SSE -S3 anzugeben AWS CLI, wenn Sie ein Objekt mithilfe von hochladen.

```
aws s3api put-object --bucket amzn-s3-demo-bucket1 --key object-key-name --server-side-
encryption AES256  --body file path
```

Weitere Informationen finden Sie unter put-object in der AWS CLI -Referenz. Informationen zur Angabe von SSE -S3 beim Kopieren eines Objekts mithilfe von finden Sie unter AWS CLI copy-object.

Verwenden AWS CloudFormation

Beispiele für das Einrichten der Verschlüsselung mit finden Sie unter Erstellen eines Buckets mit Standardverschlüsselung und unter Erstellen eines Buckets mithilfe AWS KMS serverseitiger

Verschlüsselung mit einem S3-Bucket-Key im `AWS::S3::Bucket ServerSideEncryptionRule` Thema im AWS CloudFormation Benutzerhandbuch. AWS CloudFormation

Serverseitige Verschlüsselung mit AWS KMS Schlüsseln verwenden (SSE-) KMS

> **Important**
>
> Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist in den AWS CloudTrail Protokollen, im S3-Inventar, in der S3-Speicherlinse, in der Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header im AWS Command Line Interface und AWS SDKs verfügbar. Weitere Informationen finden Sie unter Standardverschlüsselung FAQ.

Serverseitige Verschlüsselung ist die Verschlüsselung von Daten am Zielort durch die Anwendung oder den Service, der sie erhält.

Amazon S3 aktiviert automatisch die serverseitige Verschlüsselung mit Amazon S3 S3-verwalteten Schlüsseln (SSE-S3) für das Hochladen neuer Objekte.

Sofern Sie nichts anderes angeben, verwenden Buckets standardmäßig SSE -S3, um Objekte zu verschlüsseln. Sie können Buckets jedoch so konfigurieren, dass sie stattdessen serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (-) verwenden. SSE KMS Weitere Informationen finden Sie unter Geben Sie die serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) an.

AWS KMS ist ein Dienst, der sichere, hochverfügbare Hardware und Software kombiniert, um ein für die Cloud skaliertes Schlüsselverwaltungssystem bereitzustellen. Amazon S3 verwendet serverseitige Verschlüsselung mit AWS KMS (SSE-KMS), um Ihre S3-Objektdaten zu verschlüsseln. Wenn SSE - für das Objekt angefordert KMS wird, wird außerdem die S3-Prüfsumme (als Teil der Metadaten des Objekts) in verschlüsselter Form gespeichert. Weitere Informationen zur Prüfsumme finden Sie unter Überprüfung der Objektintegrität.

Wenn Sie KMS Schlüssel verwenden, können Sie das AWS Management Consoleoder AWS KMS das verwenden AWS KMS API, um Folgendes zu tun:

- Schlüssel können zentral erstellt, angezeigt, bearbeitet, überwacht, aktiviert oder deaktiviert, rotiert und das Löschen von KMS Schlüsseln geplant werden.
- Definieren Sie die Richtlinien, die steuern, wie und von wem KMS Schlüssel verwendet werden können.
- Prüfen Sie deren Verwendung, um zu beweisen, dass sie korrekt verwendet werden. Das Auditing wird von der unterstützt AWS KMS API, aber nicht von der AWS KMSAWS Management Console.

Die darin enthaltenen Sicherheitskontrollen AWS KMS können Ihnen helfen, die Compliance-Anforderungen im Zusammenhang mit der Verschlüsselung zu erfüllen. Sie können diese KMS Schlüssel verwenden, um Ihre Daten in Amazon S3 S3-Buckets zu schützen. Wenn Sie die KMS Verschlüsselung SSE mit einem S3-Bucket verwenden, AWS KMS keys muss sich dieser in derselben Region wie der Bucket befinden.

Für die Nutzung fallen zusätzliche Gebühren an AWS KMS keys. Weitere Informationen finden Sie unter AWS KMS key -Konzepte im AWS Key Management Service -Entwicklerhandbuch und in den AWS KMS -Preisen.

Berechtigungen

Um ein mit einem verschlüsseltes Objekt AWS KMS key auf Amazon S3 hochzuladen, benötigen Sie `kms:GenerateDataKey` Berechtigungen für den Schlüssel. Um ein mit einem verschlüsseltes Objekt herunterzuladen AWS KMS key, benötigen Sie `kms:Decrypt` Berechtigungen. Informationen zu den AWS KMS Berechtigungen, die für mehrteilige Uploads erforderlich sind, finden Sie unter. Mehrteiliger Upload API und Berechtigungen

> **Important**
>
> Prüfen Sie sorgfältig die Berechtigungen, die Ihnen in Ihren KMS wichtigsten Richtlinien gewährt werden. Beschränken Sie vom Kunden verwaltete KMS Schlüsselrichtlinienberechtigungen immer nur auf die IAM Prinzipale und AWS Dienste, die Zugriff auf die entsprechende AWS KMS Schlüsselaktion haben müssen. Weitere Informationen finden Sie unter Wichtige Richtlinien unter. AWS KMS

Themen

- AWS KMS keys

- Amazon-S3-Bucket-Schlüssel
- Erzwingen serverseitiger Verschlüsselung
- Verschlüsselungskontext
- Senden von Anfragen für AWS KMS verschlüsselte Objekte
- Geben Sie die serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) an
- Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys

AWS KMS keys

Wenn Sie serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) verwenden, können Sie den AWS verwalteten Standardschlüssel verwenden oder einen vom Kunden verwalteten Schlüssel angeben, den Sie bereits erstellt haben. AWS KMS unterstützt die Envelope-Verschlüsselung. S3 verwendet die AWS KMS Funktionen für die Umschlagverschlüsselung, um Ihre Daten weiter zu schützen. Bei der Umschlagverschlüsselung werden Ihre Klartextdaten mit einem Datenschlüssel und anschließend dieser Datenschlüssel mit einem Schlüssel verschlüsselt. KMS Weitere Informationen zur Envelope-Verschlüsselung finden Sie unter Envelope-Verschlüsselung im AWS Key Management Service -Entwicklerhandbuch.

Wenn Sie keinen vom Kunden verwalteten Schlüssel angeben, erstellt Amazon S3 automatisch einen Von AWS verwalteter Schlüssel , wenn Sie AWS-Konto zum ersten Mal ein mit SSE — verschlüsseltes Objekt KMS zu einem Bucket hinzufügen. Standardmäßig verwendet Amazon S3 diesen KMS Schlüssel für SSE -KMS.

> **Note**
>
> Objekte, die mit SSE - KMS mit verschlüsselt wurden, Von AWS verwaltete Schlüsselkönnen nicht kontoübergreifend geteilt werden. Wenn Sie KMS Daten kontoübergreifend SSE teilen möchten, müssen Sie einen vom Kunden verwalteten Schlüssel von verwenden. AWS KMS

Wenn Sie einen vom Kunden verwalteten Schlüssel für SSE - verwenden möchtenKMS, erstellen Sie vor der Konfiguration SSE von - einen vom Kunden verwalteten Schlüssel für symmetrische Verschlüsselung. KMS Wenn Sie dann SSE - KMS für Ihren Bucket konfigurieren, geben Sie den vorhandenen, vom Kunden verwalteten Schlüssel an. Weitere Informationen zu symmetrischen Verschlüsselungsschlüsseln finden Sie unter Symmetrische KMS Verschlüsselungsschlüssel im AWS Key Management Service Entwicklerhandbuch.

Durch die Erstellung eines vom Kunden verwalteten Schlüssels erhalten Sie mehr Flexibilität und Kontrolle. Beispielsweise können Sie kundenverwaltete Schlüssel erstellen, drehen und deaktivieren. Sie können auch Zugriffskontrollen definieren und die vom Kunden verwalteten Schlüssel prüfen, mit denen Sie Ihre Daten schützen. Weitere Informationen zu vom Kunden verwalteten und AWS verwalteten Schlüsseln finden Sie unter Kundenschlüssel und AWS Schlüssel im AWS Key Management Service Entwicklerhandbuch.

> **Note**
>
> Wenn Sie serverseitige Verschlüsselung mit einem vom Kunden verwalteten Schlüssel verwenden, der in einem externen Schlüsselspeicher gespeichert ist, sind Sie im Gegensatz zu KMS Standardschlüsseln dafür verantwortlich, die Verfügbarkeit und Haltbarkeit Ihres Schlüsselmaterials sicherzustellen. Weitere Informationen zu externen Schlüsselspeichern und dazu, wie sie das Modell der geteilten Verantwortung verändern, finden Sie unter Externe Schlüsselspeicher im Entwicklerhandbuch für AWS Key Management Service .

Verwendung SSE — KMS Verschlüsselung für kontoübergreifende Operationen

Beachten Sie Folgendes, wenn Sie kontoübergreifende Operationen verschlüsseln:

- Wenn bei der Anfrage oder über die Standardverschlüsselungskonfiguration des Buckets kein AWS KMS key Amazon-Ressourcenname (ARN) oder Alias angegeben wird, wird Von AWS verwalteter Schlüssel (`aws/s3`) verwendet.
- Wenn Sie S3-Objekte hochladen oder auf sie zugreifen, indem Sie AWS Identity and Access Management (IAM) -Prinzipale verwenden, die mit Ihrem KMS Schlüssel AWS-Konto identisch sind, können Sie Von AWS verwalteter Schlüssel (`aws/s3`) verwenden.
- Verwenden Sie einen vom Kunden verwalteten Schlüssel, wenn Sie kontoübergreifenden Zugriff auf Ihre S3-Objekte gewähren möchten. Sie können die Richtlinie eines vom Kunden verwalteten Schlüssel so konfigurieren, dass der Zugriff von einem anderen Konto aus möglich ist.
- Wenn Sie einen vom Kunden verwalteten KMS Schlüssel angeben, empfehlen wir die Verwendung eines vollqualifizierten KMS Schlüssels. ARN Wenn Sie stattdessen einen KMS Schlüsselalias verwenden, AWS KMS wird der Schlüssel im Konto des Anfragenden aufgelöst. Dieses Verhalten kann dazu führen, dass Daten mit einem KMS Schlüssel verschlüsselt werden, der dem Anforderer und nicht dem Bucket-Besitzer gehört.
- Sie müssen einen Schlüssel angeben, für den Ihnen (dem Anforderer) die Berechtigung `Encrypt` erteilt wurde. Weitere Informationen finden Sie im AWS Key Management Service

Entwicklerhandbuch unter Erlauben Sie Schlüsselbenutzern, einen KMS Schlüssel für kryptografische Operationen zu verwenden.

Weitere Informationen darüber, wann vom Kunden verwaltete Schlüssel und AWS KMS verwaltete Schlüssel verwendet werden sollten, finden Sie unter Sollte ich einen Von AWS verwalteter Schlüssel oder einen vom Kunden verwalteten Schlüssel verwenden, um meine Objekte in Amazon S3 zu verschlüsseln?

SSE- KMS Verschlüsselungsablauf

Wenn Sie sich dafür entscheiden, Ihre Daten mit einem Von AWS verwalteter Schlüssel oder einem vom Kunden verwalteten Schlüssel zu verschlüsseln, AWS KMS und Amazon S3 die folgenden Envelope-Verschlüsselungsaktionen durchführen:

1. Amazon S3 fordert einen Klartext-Datenschlüssel und eine Kopie des unter dem angegebenen Schlüssel verschlüsselten KMS Schlüssels an.
2. AWS KMS generiert einen Datenschlüssel, verschlüsselt ihn unter dem KMS Schlüssel und sendet sowohl den Klartext-Datenschlüssel als auch den verschlüsselten Datenschlüssel an Amazon S3.
3. Amazon S3 verschlüsselt die Daten mit dem Datenschlüssel und entfernt anschließend den Klartextschlüssel schnellstmöglich aus dem Arbeitsspeicher.
4. Amazon S3 speichert den verschlüsselten Datenschlüssel im Metadatenformat zusammen mit den verschlüsselten Daten.

Wenn Sie die Entschlüsselung Ihrer Daten beantragen, nutzen Sie Amazon S3 und AWS KMS führen Sie die folgenden Aktionen aus:

1. Amazon S3 sendet den verschlüsselten Datenschlüssel AWS KMS in einer `Decrypt` Anfrage an.
2. AWS KMS entschlüsselt den verschlüsselten Datenschlüssel mit demselben KMS Schlüssel und gibt den Klartext-Datenschlüssel an Amazon S3 zurück.
3. Amazon S3 entschlüsselte die verschlüsselten Daten mit dem Klartext-Datenschlüssel und entfernt den Klartext-Datenschlüssel anschließend schnellstmöglich aus dem Arbeitsspeicher.

> **Important**
>
> Wenn Sie eine AWS KMS key für die serverseitige Verschlüsselung in Amazon S3 verwenden, müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel

wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel. Weitere Informationen zu diesen Schlüsseln finden Sie unter Symmetrische KMS Verschlüsselungsschlüssel im AWS Key Management Service Entwicklerhandbuch.

Prüfung SSE — Verschlüsselung KMS

Um Anfragen zu identifizieren, die SSE - angebenKMS, können Sie die Metriken All SSE - KMS requests und% all SSE - KMS requests in Amazon S3 Storage Lens Metriken verwenden. S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Sie können auch die Buckets SSE - KMS enabled bucket count und% SSE - KMS enabled verwenden, um die Anzahl der Buckets zu ermitteln, die (SSE-KMS) für die Standard-Bucket-Verschlüsselung verwendet werden. Weitere Informationen finden Sie unter Bewertung Ihrer Speicheraktivität und -nutzung mit S3 Storage Lens. Eine vollständige Liste der Metriken finden Sie im Glossar der S3-Storage-Lens-Metriken.

Um die Verwendung Ihrer AWS KMS Schlüssel für Ihre SSE - KMS verschlüsselten Daten zu überprüfen, können Sie Protokolle verwenden AWS CloudTrail . Sie können Einblick in Ihre kryptografischen Operationen erhalten, wie zum Beispiel GenerateDataKey und Decrypt. CloudTrail unterstützt zahlreiche Attributwerte zum Filtern Ihrer Suche, einschließlich Ereignisname, Benutzername und Ereignisquelle.

Amazon-S3-Bucket-Schlüssel

Wenn Sie die serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) konfigurieren, können Sie Ihre Buckets so konfigurieren, dass S3-Bucket-Keys für SSE - verwendet werden. KMS Die Verwendung eines Schlüssels auf Bucket-Ebene für SSE - KMS kann Ihre AWS KMS Anforderungskosten um bis zu 99 Prozent senken, indem der Anforderungsverkehr von Amazon S3 zu reduziert wird. AWS KMS

Wenn Sie einen Bucket so konfigurieren, dass bei neuen Objekten ein S3-Bucket Key verwendet wird, wird ein Schlüssel KMS auf Bucket-Ebene AWS KMS generiert, der verwendet wird, um eindeutige Datenschlüssel für Objekte im Bucket zu erstellen. SSE Dieser S3-Bucket-Key wird für einen begrenzten Zeitraum innerhalb von Amazon S3 verwendet, sodass Amazon S3 keine Anfragen mehr stellen muss, AWS KMS um Verschlüsselungsvorgänge abzuschließen. Weitere Informationen zur Verwendung von S3-Bucket-Schlüsseln finden Sie unter Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys.

Erzwingen serverseitiger Verschlüsselung

Wenn Sie die serverseitige Verschlüsselung aller Objekte in einem bestimmten Amazon-S3-Bucket anfordern möchten, können Sie eine Bucket-Richtlinie verwenden. Die folgende Bucket-Richtlinie verweigert beispielsweise allen Benutzern die Berechtigung zum Hochladen von Objekten (`s3:PutObject`), wenn die Anfrage keinen `x-amz-server-side-encryption-aws-kms-key-id` Header enthält, der eine serverseitige Verschlüsselung mit - anfordert. SSE KMS

```
{
   "Version":"2012-10-17",
   "Id":"PutObjectPolicy",
   "Statement":[{
         "Sid":"DenyObjectsThatAreNotSSEKMS",
         "Effect":"Deny",
         "Principal":"*",
         "Action":"s3:PutObject",
         "Resource":"arn:aws:s3:::amzn-s3-demo-bucket1/*",
         "Condition":{
            "Null":{
               "s3:x-amz-server-side-encryption-aws-kms-key-id":"true"
            }
         }
      }
   ]
}
```

Um zu verlangen, dass ein bestimmter AWS KMS key Code zum Verschlüsseln der Objekte in einem Bucket verwendet wird, können Sie den `s3:x-amz-server-side-encryption-aws-kms-key-id` Bedingungsschlüssel verwenden. Um den KMS Schlüssel anzugeben, müssen Sie einen Schlüssel Amazon Resource Name (ARN) im folgenden `arn:aws:kms:`*`region`*`:`*`acct-id`*`:key/`*`key-id`* Format verwenden. AWS Identity and Access Management überprüft nicht, ob die Zeichenfolge für `s3:x-amz-server-side-encryption-aws-kms-key-id` existiert.

> **Note**
>
> Wenn Sie ein Objekt hochladen, können Sie den KMS Schlüssel mithilfe des `x-amz-server-side-encryption-aws-kms-key-id` Headers angeben oder sich auf Ihre Standardkonfiguration für die Bucket-Verschlüsselung verlassen. Wenn Ihre PutObject Anfrage `aws:kms` im `x-amz-server-side-encryption` Header angegeben ist, den `x-`

`amz-server-side-encryption-aws-kms-key-id` Header jedoch nicht spezifiziert, geht Amazon S3 davon aus, dass Sie den verwenden möchten Von AWS verwalteter Schlüssel. Unabhängig davon muss die AWS KMS Schlüssel-ID, die Amazon S3 für die Objektverschlüsselung verwendet, mit der AWS KMS Schlüssel-ID in der Richtlinie übereinstimmen, andernfalls lehnt Amazon S3 die Anfrage ab.

Eine vollständige Liste der Amazon S3-spezifischen Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference.

Verschlüsselungskontext

Ein Verschlüsselungskontext ist ein Satz von Schlüssel-Wert-Paaren, die zusätzliche kontextbezogene Informationen zu den Daten enthalten können. Der Verschlüsselungskontext ist nicht verschlüsselt. Wenn für eine Verschlüsselungsoperation ein Verschlüsselungskontext angegeben wird, muss Amazon S3 denselben Verschlüsselungskontext auch für die Entschlüsselungsoperation angeben. Andernfalls schlägt die Entschlüsselung fehl. AWS KMS verwendet den Verschlüsselungskontext als zusätzliche authentifizierte Daten (AAD), um die authentifizierte Verschlüsselung zu unterstützen. Weitere Informationen zum Verschlüsselungs-Kontext finden Sie unter Encryption Context (Verschlüsselungs-Kontext) im AWS Key Management Service -Entwicklerhandbuch.

Standardmäßig verwendet Amazon S3 das Objekt oder den Bucket Amazon Resource Name (ARN) als Verschlüsselungskontextpaar:

- Wenn Sie SSE - verwenden, KMS ohne einen S3-Bucket Key zu aktivieren, ARN wird das Objekt als Verschlüsselungskontext verwendet.

  ```
  arn:aws:s3:::object_ARN
  ```

- Wenn Sie SSE - verwenden KMS und einen S3-Bucket-Key aktivieren, ARN wird der Bucket als Verschlüsselungskontext verwendet. Weitere Informationen zu S3-Bucket-Schlüsseln finden Sie unter Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys.

  ```
  arn:aws:s3:::bucket_ARN
  ```

Sie können optional ein zusätzliches Verschlüsselungskontextpaar bereitstellen, indem Sie den `x-amz-server-side-encryption-context` Header in einer s3: PutObject -Anforderung

verwenden. Da der Verschlüsselungskontext jedoch nicht verschlüsselt ist, sollte er keine sensiblen Informationen enthalten. Amazon S3 speichert dieses zusätzliche Schlüsselpaar zusammen mit dem Standardverschlüsselungskontext. Wenn Ihre PUT-Anforderung verarbeitet wird, hängt Amazon S3 den Standardverschlüsselungskontext `aws:s3:arn` an den von Ihnen bereitgestellten an.

Sie können den Verschlüsselungskontext verwenden, um Ihre kryptografischen Vorgänge zu identifizieren und zu kategorisieren. Sie können auch den ARN Standardwert für den Verschlüsselungskontext verwenden, um relevante Anfragen nachzuverfolgen, AWS CloudTrail indem Sie sich ansehen, welcher Amazon S3 mit welchem Verschlüsselungsschlüssel verwendet ARN wurde.

Im `requestParameters` Feld einer CloudTrail Protokolldatei sieht der Verschlüsselungskontext dem folgenden ähnlich.

```
"encryptionContext": {
    "aws:s3:arn": "arn:aws:s3:::amzn-s3-demo-bucket1/file_name"
}
```

Wenn Sie SSE - KMS mit der optionalen Funktion S3 Bucket Keys verwenden, entspricht der Wert des Verschlüsselungskontextes dem Wert ARN des Buckets.

```
"encryptionContext": {
    "aws:s3:arn": "arn:aws:s3:::amzn-s3-demo-bucket1"
}
```

Senden von Anfragen für AWS KMS verschlüsselte Objekte

> **Important**
>
> Alle GET PUT Anfragen für AWS KMS verschlüsselte Objekte müssen mit Secure Sockets Layer (SSL) oder Transport Layer Security (TLS) gestellt werden. Anfragen müssen außerdem mit gültigen Anmeldeinformationen signiert werden, z. B. mit AWS Signature Version 4 (oder AWS Signature Version 2).

AWS Bei Signature Version 4 werden Authentifizierungsinformationen zu AWS Anfragen hinzugefügt, die von gesendet wurdenHTTP. Aus Sicherheitsgründen AWS müssen die meisten Anfragen mit einem Zugriffsschlüssel signiert werden, der aus einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel besteht. Diese beiden Schlüssel werden in der Regel als

Sicherheitsanmeldeinformationen bezeichnet. Weitere Informationen finden Sie unter Authenticating Requests (Authentifizierung von Anforderungen) (AWS Signature Version 4) und Signature Version 4 signing process (Signaturprozess).

> **Important**
>
> Wenn Ihr Objekt SSE - verwendetKMS, senden Sie keine Header für `GET` Verschlüsselungsanfragen für Anfragen und `HEAD` Anfragen. Andernfalls erhalten Sie den Fehler HTTP400 Bad Request.

Themen

- Geben Sie die serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) an
- Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys

Geben Sie die serverseitige Verschlüsselung mit AWS KMS (SSE-KMS) an

Für alle Amazon-S3-Buckets ist die Verschlüsselung standardmäßig konfiguriert und alle neuen Objekte, die in einen S3-Bucket hochgeladen werden, werden im Ruhezustand automatisch verschlüsselt. Die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) ist die Standardverschlüsselungskonfiguration für jeden Bucket in Amazon S3. Um einen anderen Verschlüsselungstyp zu verwenden, können Sie entweder die Art der serverseitigen Verschlüsselung angeben, die in Ihren S3-PUT-Anfragen verwendet werden soll, oder Sie können die Standardverschlüsselungskonfiguration im Ziel-Bucket festlegen.

Wenn Sie in Ihren `PUT` Anfragen einen anderen Verschlüsselungstyp angeben möchten, können Sie serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln (-), zweischichtige serverseitige Verschlüsselung mit Schlüsseln (SSE-KMS) oder serverseitige Verschlüsselung mit vom Kunden bereitgestellten AWS KMS Schlüsseln (DSSE-CKMS) verwenden. SSE Wenn Sie im Ziel-Bucket eine andere Standardverschlüsselungskonfiguration festlegen möchten, können Sie - oder - verwenden. SSE KMS DSSE KMS

Sie können die Verschlüsselung anwenden, wenn Sie entweder ein neues Objekt hochladen oder ein vorhandenes Objekt kopieren.

Sie können SSE - KMS mithilfe der Amazon S3 S3-Konsole, der REST API Operationen und der AWS Command Line Interface (AWS CLI) angeben. AWS SDKs Weitere Informationen finden Sie unter den folgenden Themen.

> **Note**
>
> Sie können Multiregion AWS KMS keys in Amazon S3 verwenden. Amazon S3 behandelt jedoch derzeit Multi-Regions-Schlüssel wie Einzel-Regions-Schlüssel und verwendet nicht die Multi-Regions-Funktionen des Schlüssels. Weitere Informationen finden Sie unter Verwenden von Schlüsseln für mehrere Regionen im AWS Key Management Service Entwicklerhandbuch.

> **Note**
>
> Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, benötigen Sie die Erlaubnis, den Schlüssel zu verwenden. Weitere Informationen zu kontoübergreifenden Berechtigungen für KMS Schlüssel finden Sie im AWS Key Management Service Entwicklerhandbuch unter Erstellen von KMS Schlüsseln, die andere Konten verwenden können.

Verwenden der S3-Konsole

In diesem Thema wird beschrieben, wie Sie mithilfe der Amazon S3 S3-Konsole den Verschlüsselungstyp eines Objekts so einstellen oder ändern, dass serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS KMS) -Schlüsseln (SSE-) verwendet wird.

> **Note**
>
> - Sie können die Verschlüsselung eines Objekts ändern, wenn Ihr Objekt weniger als 5 GB groß ist. Wenn Ihr Objekt größer als 5 GB ist, müssen Sie AWS CLIoder verwenden, AWS SDKsum die Verschlüsselung eines Objekts zu ändern.
> - Eine Liste der zusätzlichen Berechtigungen, die zum Ändern der Verschlüsselung eines Objekts erforderlich sind, finden Sie unterthe section called “Erforderliche Berechtigungen für API S3-Operationen”. Richtlinien, die diese Berechtigung gewähren, finden Sie beispielsweise unterthe section called “Beispiele für identitätsbasierte Richtlinien”.
> - Wenn Sie die Verschlüsselung eines Objekts ändern, wird ein neues Objekt erstellt, um das alte zu ersetzen. Wenn S3-Versioning aktiviert ist, wird eine neue Version des

Objekts erstellt, und das vorhandene Objekt wird zu einer älteren Version. Die Rolle, die die Eigenschaft ändert, wird auch Besitzer des neuen Objekts (oder der neuen Objektversion).

So fügen Sie die Verschlüsselung für ein Objekt hinzu oder ändern Sie sie

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im Navigationsbereich Buckets und dann den Tab Allzweck-Buckets aus. Navigieren Sie zu dem Amazon S3 S3-Bucket oder -Ordner, der die Objekte enthält, die Sie ändern möchten.
3. Aktivieren Sie das Kontrollkästchen für die Objekte, die Sie ändern möchten.
4. Wählen Sie im Menü Aktionen aus der daraufhin angezeigten Optionsliste die Option Serverseitige Verschlüsselung bearbeiten aus.
5. Scrollen Sie zum Abschnitt Serverseitige Verschlüsselung.
6. Wählen Sie unter Verschlüsselungseinstellungen die Option Verwenden von Bucket-Einstellungen für die Standardverschlüsselung oder Überschreiben der Bucket-Einstellungen für die Standardverschlüsselung aus.

   > **Important**
   >
   > Wenn Sie die KMS Option SSE - für Ihre standardmäßige Verschlüsselungskonfiguration verwenden, gelten für Sie die Kontingente für Anfragen pro Sekunde (RPS) von AWS KMS. Weitere Informationen zu AWS KMS -Kontingenten und zum Anfordern einer Kontingenterhöhung finden Sie unter Kontingente im Entwicklerhandbuch zu AWS Key Management Service .

7. Wenn Sie Überschreiben der Bucket-Einstellungen für die Standardverschlüsselung ausgewählt haben, konfigurieren Sie die folgenden Verschlüsselungseinstellungen.

   a. Wählen Sie unter Verschlüsselungstyp die Option Serverseitige Verschlüsselung mit AWS Key Management Service Schlüsseln (SSE-KMS) aus.

   b. Führen Sie unter AWS KMS Schlüssel einen der folgenden Schritte aus, um Ihren KMS Schlüssel auszuwählen:

- Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys und wählen Sie dann Ihren KMSSchlüssel aus der Liste der verfügbaren Schlüssel aus.

  Sowohl der Von AWS verwalteter Schlüssel (`aws/s3`) als auch Ihr vom Kunden verwalteter Schlüssel werden in dieser Liste angezeigt. Weitere Informationen über vom Kunden verwaltete Schlüssel finden Sie unter Kundenschlüssel und AWS -Schlüssel im Entwicklerhandbuch zu AWS Key Management Service .

- Um den KMS Schlüssel einzugebenARN, wählen Sie Enter AWS KMS key ARN und geben Sie dann Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
- Um einen neuen, vom Kunden verwalteten Schlüssel in der AWS KMS Konsole zu erstellen, wählen Sie Create a KMS key aus.

  Weitere Informationen zum Erstellen eines AWS KMS key finden Sie unter Schlüssel erstellen im AWS Key Management Service Entwicklerhandbuch.

> **Important**
>
> Sie können nur KMS Schlüssel verwenden, die im selben AWS-Region Bucket verfügbar sind. Die Amazon S3 S3-Konsole listet nur die ersten 100 KMS Schlüssel in derselben Region wie der Bucket auf. Um einen KMS Schlüssel zu verwenden, der nicht aufgeführt ist, müssen Sie Ihren KMS Schlüssel eingebenARN. Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, benötigen Sie zunächst die Erlaubnis, den Schlüssel zu verwenden, und dann müssen Sie den KMS Schlüssel eingebenARN.
> Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel und keine asymmetrischen KMS Schlüssel. Weitere Informationen finden Sie unter Identifizieren symmetrischer und asymmetrischer KMS Schlüssel im Entwicklerhandbuch.AWS Key Management Service

8. Wählen Sie unter Zusätzliche Kopiereinstellungen aus, ob Sie Quelleinstellungen kopieren, Keine Einstellungen angeben oder Einstellungen angeben möchten. Quelleinstellungen kopieren ist die Standardoption. Wenn Sie das Objekt nur ohne die Quelleinstellungsattribute kopieren möchten, wählen Sie „Keine Einstellungen angeben“. Wählen Sie „Einstellungen angeben“, um Einstellungen für SpeicherklasseACLs, Objekt-Tags, Metadaten, serverseitige Verschlüsselung und zusätzliche Prüfsummen anzugeben.

9. Wählen Sie Änderungen speichern.

> **Note**
>
> Diese Aktion wendet auf alle angegebenen Objekte Verschlüsselung an. Warten Sie beim Verschlüsseln von Ordnern, bis die Speicheroperation abgeschlossen ist, bevor Sie dem Ordner neue Objekte hinzufügen.

Unter Verwendung der REST API

Wenn Sie ein Objekt erstellen, d. h. wenn Sie ein neues Objekt hochladen oder ein vorhandenes Objekt kopieren, können Sie die serverseitige Verschlüsselung mit AWS KMS keys (SSE-) angeben, um Ihre Daten zu verschlüsseln. KMS Fügen Sie hierzu der Anforderung den Header `x-amz-server-side-encryption` hinzu. Setzen Sie den Wert des Headers auf den `aws:kms`-Verschlüsselungsalgorithmus. Amazon S3 bestätigt mit SSE —, dass Ihr Objekt gespeichert wurde, KMS indem es den Antwort-Header zurückgibt`x-amz-server-side-encryption`.

Wenn Sie den Header `x-amz-server-side-encryption` mit dem Wert `aws:kms` angeben, können Sie auch die folgenden Anforderungs-Header verwenden:

- `x-amz-server-side-encryption-aws-kms-key-id`
- `x-amz-server-side-encryption-context`
- `x-amz-server-side-encryption-bucket-key-enabled`

Themen

- Amazon S3 REST API S3-Operationen, die unterstützen SSE - KMS
- Verschlüsselungskontext (x-amz-server-side-encryption-context)
- AWS KMS Schlüssel-ID (x-amz-server-side-encryption-aws-kms-key-id)
- S3-Bucket-Schlüssel (x-amz-server-side-encryption-aws-bucket-key-enabled)

Amazon S3 REST API S3-Operationen, die unterstützen SSE - KMS

Die folgenden REST API Operationen akzeptieren die Header `x-amz-server-side-encryption`x-amz-server-side-encryption-aws-kms-key-id, und `x-amz-server-side-encryption-context` Request-Header.

- PutObject— Wenn Sie Daten mithilfe der PUT API Operation hochladen, können Sie diese Anforderungsheader angeben.
- CopyObject – Wenn Sie ein Objekt kopieren, erhalten Sie ein Quell- und ein Zielobjekt. Wenn Sie bei der `CopyObject` Operation SSE KMS -Header übergeben, werden sie nur auf das Zielobjekt angewendet. Wenn Sie ein vorhandenes Objekt kopieren, unabhängig davon, ob das Quellobjekt verschlüsselt ist oder nicht, wird das Zielobjekt nicht verschlüsselt, es sei denn, Sie fordern ausdrücklich serverseitige Verschlüsselung an.
- POST Object— Wenn Sie einen `POST` Vorgang verwenden, um ein Objekt hochzuladen, geben Sie anstelle der Anforderungsheader dieselben Informationen in die Formularfelder ein.
- CreateMultipartUpload— Wenn Sie große Objekte mithilfe des mehrteiligen API Upload-Vorgangs hochladen, können Sie diese Header angeben. Sie geben diese Header in der Anfrage an. `CreateMultipartUpload`

Die Antwortheader der folgenden REST API Operationen geben den `x-amz-server-side-encryption` Header zurück, wenn ein Objekt mithilfe serverseitiger Verschlüsselung gespeichert wird.

- PutObject
- CopyObject
- POST Object
- CreateMultipartUpload
- UploadPart
- UploadPartCopy
- CompleteMultipartUpload
- GetObject
- HeadObject

> **Important**
>
> - Alle `GET` `PUT` Anfragen für ein Objekt, das durch geschützt ist, AWS KMS schlagen fehl, wenn Sie diese Anfragen nicht mithilfe von Secure Sockets Layer (SSL), Transport Layer Security (TLS) oder Signature Version 4 stellen.

- Wenn Ihr Objekt SSE - verwendetKMS, senden Sie keine Header für `GET` Verschlüsselungsanfragen für Anfragen und `HEAD` Anfragen, da Sie sonst die BadRequest Fehlermeldung HTTP400 erhalten.

Verschlüsselungskontext (**`x-amz-server-side-encryption-context`**)

Wenn Sie angeben`x-amz-server-side-encryption:aws:kms`, API unterstützt Amazon S3 einen Verschlüsselungskontext mit dem `x-amz-server-side-encryption-context` Header. Ein Verschlüsselungskontext ist ein Satz von Schlüssel-Wert-Paaren, die zusätzliche kontextbezogene Informationen zu den Daten enthalten können.

Amazon S3 verwendet automatisch das Objekt oder den Bucket Amazon Resource Name (ARN) als Verschlüsselungskontextpaar. Wenn Sie SSE - verwenden, KMS ohne einen S3-Bucket Key zu aktivieren, verwenden Sie das Objekt ARN als Ihren Verschlüsselungskontext; zum Beispiel`arn:aws:s3:::`*`object_ARN`*. Wenn Sie jedoch SSE - verwenden KMS und einen S3-Bucket-Key aktivieren, verwenden Sie den Bucket ARN für Ihren Verschlüsselungskontext; zum Beispiel`arn:aws:s3:::`*`bucket_ARN`*.

Sie können optional ein zusätzliches Verschlüsselungskontextpaar bereitstellen, indem Sie den Header `x-amz-server-side-encryption-context` verwenden. Da der Verschlüsselungskontext jedoch nicht verschlüsselt ist, stellen Sie sicher, dass er keine vertraulichen Informationen enthält. Amazon S3 speichert dieses zusätzliche Schlüsselpaar zusammen mit dem Standardverschlüsselungskontext.

Weitere Informationen zum Verschlüsselungskontext in Amazon S3 finden Sie unter Verschlüsselungskontext. Allgemeine Informationen zum Verschlüsselungs-Kontext finden Sie unter AWS Key Management Service Concepts – Encryption Context (Konzepte – Verschlüsselungs-Kontext) im AWS Key Management Service -Entwicklerhandbuch.

AWS KMS Schlüssel-ID (**`x-amz-server-side-encryption-aws-kms-key-id`**)

Sie können den Header `x-amz-server-side-encryption-aws-kms-key-id` verwenden, um die ID des vom Kunden verwalteten Schlüssels anzugeben, der zum Schutz der Daten verwendet wird. Wenn Sie den Header `x-amz-server-side-encryption:aws:kms`, jedoch nicht den Header `x-amz-server-side-encryption-aws-kms-key-id` angeben, verwendet Amazon S3 Von AWS verwalteter Schlüssel (`aws/s3`), um die Daten zu schützen. Wenn Sie einen vom Kunden verwalteten Schlüssel verwenden möchten, müssen Sie den `x-amz-server-side-encryption-aws-kms-key-id`-Header des vom Kunden verwalteten Schlüssels angeben.

> **Important**
>
> Wenn Sie eine AWS KMS key für die serverseitige Verschlüsselung in Amazon S3 verwenden, müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel. Weitere Informationen zu diesen Schlüsseln finden Sie unter Symmetrische KMS Verschlüsselungsschlüssel im AWS Key Management Service Entwicklerhandbuch.

S3-Bucket-Schlüssel (**`x-amz-server-side-encryption-aws-bucket-key-enabled`**)

Sie können den `x-amz-server-side-encryption-aws-bucket-key-enabled` Anforderungsheader verwenden, um einen S3-Bucket-Key auf Objektebene zu aktivieren oder zu deaktivieren. S3 Bucket Keys reduzieren Ihre AWS KMS Anforderungskosten, indem sie den Anforderungsverkehr von Amazon S3 zu reduzieren AWS KMS. Weitere Informationen finden Sie unter Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys.

Wenn Sie den Header `x-amz-server-side-encryption:aws:kms`, jedoch nicht den Header `x-amz-server-side-encryption-aws-bucket-key-enabled` angeben, werden die Einstellung des S3-Bucket-Schlüssels für den Ziel-Bucket verwendet, um Ihr Objekt zu verschlüsseln. Weitere Informationen finden Sie unter Konfigurieren eines S3-Bucket-Schlüssels auf Objektebene .

Mit dem AWS CLI

Um die folgenden AWS CLI Beispielbefehle zu verwenden, ersetzen Sie die *`user input placeholders`* durch Ihre eigenen Informationen.

Wenn Sie ein neues Objekt hochladen oder ein vorhandenes Objekt kopieren, können Sie angeben, dass serverseitige Verschlüsselung mit AWS KMS Schlüsseln zum Verschlüsseln Ihrer Daten verwendet werden soll. Fügen Sie hierzu der Anforderung den Header `--server-side-encryption aws:kms` hinzu. Verwenden Sie den`--ssekms-key-id` *`example-key-id`*, um Ihren vom Kunden verwalteten AWS KMS Schlüssel hinzuzufügen, den Sie erstellt haben. Wenn Sie eine AWS KMS Schlüssel-ID angeben`--server-side-encryption aws:kms`, aber keine angeben, verwendet Amazon S3 einen AWS verwalteten Schlüssel.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key example-object-key --server-side-encryption aws:kms --ssekms-key-id example-key-id --body filepath
```

Sie können Amazon S3 Bucket Keys zusätzlich für Ihre COPY Operationen aktivieren PUT oder deaktivieren, indem Sie `--bucket-key-enabled` oder hinzufügen`--no-bucket-key-enabled`. Amazon S3 Bucket Keys kann Ihre AWS KMS Anforderungskosten senken, indem der Anforderungsverkehr von Amazon S3 zu reduziert wird AWS KMS. Weitere Informationen finden Sie unter Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key example-object-key --server-
side-encryption aws:kms --bucket-key-enabled --body filepath
```

Sie können ein unverschlüsseltes Objekt verschlüsseln, um es zu verwenden, KMS indem SSE Sie das Objekt wieder an seinen ursprünglichen Platz kopieren.

```
aws s3api copy-object --bucket amzn-s3-demo-bucket --key example-object-key --
body filepath --bucket amzn-s3-demo-bucket --key example-object-key --sse aws:kms --
sse-kms-key-id example-key-id --body filepath
```

Mit dem AWS SDKs

Bei der Verwendung AWS SDKs können Sie Amazon S3 AWS KMS keys für die serverseitige Verschlüsselung anfordern. Die folgenden Beispiele zeigen die Verwendung von SSE - KMS mit dem AWS SDKs für Java und. NET. Informationen zu anderen SDKs finden Sie im AWS Developer Center unter Beispielcode und Bibliotheken.

> **Important**
>
> Wenn Sie eine AWS KMS key für die serverseitige Verschlüsselung in Amazon S3 verwenden, müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel. Weitere Informationen zu diesen Schlüsseln finden Sie unter Symmetrische KMS Verschlüsselungsschlüssel im AWS Key Management Service Entwicklerhandbuch.

`CopyObject`-Operation

Wenn Sie Objekte kopieren, fügen Sie dieselben Anfrageeigenschaften (`ServerSideEncryptionMethod` und `ServerSideEncryptionKeyManagementServiceKeyId`) hinzu, um Amazon S3 aufzufordern, ein AWS KMS key zu verwenden. Weitere Informationen über das Kopieren von Objekten finden Sie unter Objekte kopieren, verschieben und umbenennen.

PUT-Operation

Java

Wenn Sie ein Objekt mithilfe von hochladen AWS SDK for Java, können Sie Amazon S3 auffordern, ein zu verwenden, AWS KMS key indem Sie die `SSEAwsKeyManagementParams` Eigenschaft hinzufügen, wie in der folgenden Anfrage gezeigt:

```
PutObjectRequest putRequest = new PutObjectRequest(bucketName,
   keyName, file).withSSEAwsKeyManagementParams(new SSEAwsKeyManagementParams());
```

In diesem Fall verwendet Amazon S3 die Von AWS verwalteter Schlüssel (`aws/s3`). Weitere Informationen finden Sie unter Serverseitige Verschlüsselung mit AWS KMS Schlüsseln verwenden (SSE-) KMS. Sie können optional einen symmetrischen KMS Verschlüsselungsschlüssel erstellen und diesen in der Anfrage angeben, wie im folgenden Beispiel gezeigt:

```
PutObjectRequest putRequest = new PutObjectRequest(bucketName,
   keyName, file).withSSEAwsKeyManagementParams(new
 SSEAwsKeyManagementParams(keyID));
```

Weitere Informationen zur Erstellung von kundenverwalteten Schlüsseln finden Sie unter Programmieren von AWS KMS API im AWS Key Management Service Entwicklerhandbuch.

Funktionierende Codebeispiele zum Hochladen eines Objekts finden Sie unter den folgenden Themen. Um diese Beispiele zu verwenden, müssen Sie die Codebeispiele aktualisieren und Verschlüsselungsinformationen wie im vorigen Codefragment gezeigt bereitstellen.

- Weitere Informationen zum Hochladen eines Objekts in einem einzigen Vorgang finden Sie unter Objekte hochladen.
- Informationen zu mehrteiligen Uploads, bei denen mehrteilige Uploads auf hoher oder niedriger Ebene verwendet werden, finden Sie API unter. Hochladen eines Objekts mit Multipart-Upload

.NET

Wenn Sie ein Objekt mithilfe von hochladen AWS SDK for .NET, können Sie Amazon S3 auffordern, ein zu verwenden, AWS KMS key indem Sie die `ServerSideEncryptionMethod` Eigenschaft hinzufügen, wie in der folgenden Anfrage gezeigt:

```
PutObjectRequest putRequest = new PutObjectRequest
 {
     BucketName = amzn-s3-demo-bucket,
     Key = keyName,
     // other properties
     ServerSideEncryptionMethod = ServerSideEncryptionMethod.AWSKMS
 };
```

In diesem Fall verwendet Amazon S3 die Von AWS verwalteter Schlüssel. Weitere Informationen finden Sie unter Serverseitige Verschlüsselung mit AWS KMS Schlüsseln verwenden (SSE-) KMS. Sie können optional Ihren eigenen, vom Kunden verwalteten Schlüssel für symmetrische Verschlüsselung erstellen und diesen in der Anfrage angeben, wie im folgenden Beispiel gezeigt:

```
PutObjectRequest putRequest1 = new PutObjectRequest
{
  BucketName = amzn-s3-demo-bucket,
  Key = keyName,
  // other properties
  ServerSideEncryptionMethod = ServerSideEncryptionMethod.AWSKMS,
  ServerSideEncryptionKeyManagementServiceKeyId = keyId
};
```

Weitere Informationen zur Erstellung von kundenverwalteten Schlüsseln finden Sie unter Programmieren von AWS KMS API im AWS Key Management Service Entwicklerhandbuch.

Funktionierende Codebeispiele zum Hochladen eines Objekts finden Sie unter den folgenden Themen. Um diese Beispiele zu verwenden, müssen Sie die Codebeispiele aktualisieren und Verschlüsselungsinformationen wie im vorigen Codefragment gezeigt bereitstellen.

- Weitere Informationen zum Hochladen eines Objekts in einem einzigen Vorgang finden Sie unter Objekte hochladen.
- Informationen zu mehrteiligen Uploads, bei denen mehrteilige Uploads auf hoher oder niedriger Ebene verwendet werden, finden Sie API unter. Hochladen eines Objekts mit Multipart-Upload

Vorsigniert URLs

Java

Wenn Sie ein vorsigniertes Objekt URL für ein Objekt erstellen, das mit einem verschlüsselt ist AWS KMS key, müssen Sie Signature Version 4 explizit angeben, wie im folgenden Beispiel gezeigt:

```
ClientConfiguration clientConfiguration = new ClientConfiguration();
clientConfiguration.setSignerOverride("AWSS3V4SignerType");
AmazonS3Client s3client = new AmazonS3Client(
        new ProfileCredentialsProvider(), clientConfiguration);
...
```

Ein Codebeispiel finden Sie unter Objekte mit vorsignierten teilen URLs.

.NET

Wenn Sie eine Vorsignatur URL für ein Objekt erstellen, das mit einer verschlüsselt ist AWS KMS key, müssen Sie Signature Version 4 explizit angeben, wie im folgenden Beispiel gezeigt:

```
AWSConfigs.S3Config.UseSignatureVersion4 = true;
```

Ein Codebeispiel finden Sie unter Objekte mit vorsignierten teilen URLs.

Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys

Amazon S3 Bucket Keys reduzieren die Kosten für die serverseitige Amazon S3 S3-Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-KMS). Durch die Verwendung eines Schlüssels auf Bucket-Ebene für SSE - KMS können die AWS KMS Anforderungskosten um bis zu 99 Prozent gesenkt werden, da der Anforderungsverkehr von Amazon S3 zu reduziert wird. AWS KMS Mit wenigen Klicks und ohne Änderungen an Ihren Client-Anwendungen können Sie Ihren Bucket so konfigurieren, dass er einen S3-Bucket-Key für SSE die KMS Verschlüsselung neuer Objekte verwendet. AWS Management Console

> **Note**
>
> S3-Bucket-Keys werden für die zweischichtige serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (DSSE-KMS) nicht unterstützt.

S3-Bucket-Keys für - SSE KMS

Workloads, die auf Millionen oder Milliarden von Objekten zugreifen, die mit SSE - verschlüsselt sind, KMS können große Mengen an Anfragen an AWS KMS generieren. Wenn Sie SSE — KMS zum Schutz Ihrer Daten ohne einen S3-Bucket Key verwenden, verwendet Amazon S3 für jedes Objekt einen individuellen AWS KMS Datenschlüssel. In diesem Fall ruft Amazon S3 AWS KMS jedes Mal, wenn eine Anfrage für ein KMS -verschlüsseltes Objekt gestellt wird, auf. Informationen darüber, wie SSE - KMS funktioniert, finden Sie unterServerseitige Verschlüsselung mit AWS KMS Schlüsseln verwenden (SSE-) KMS.

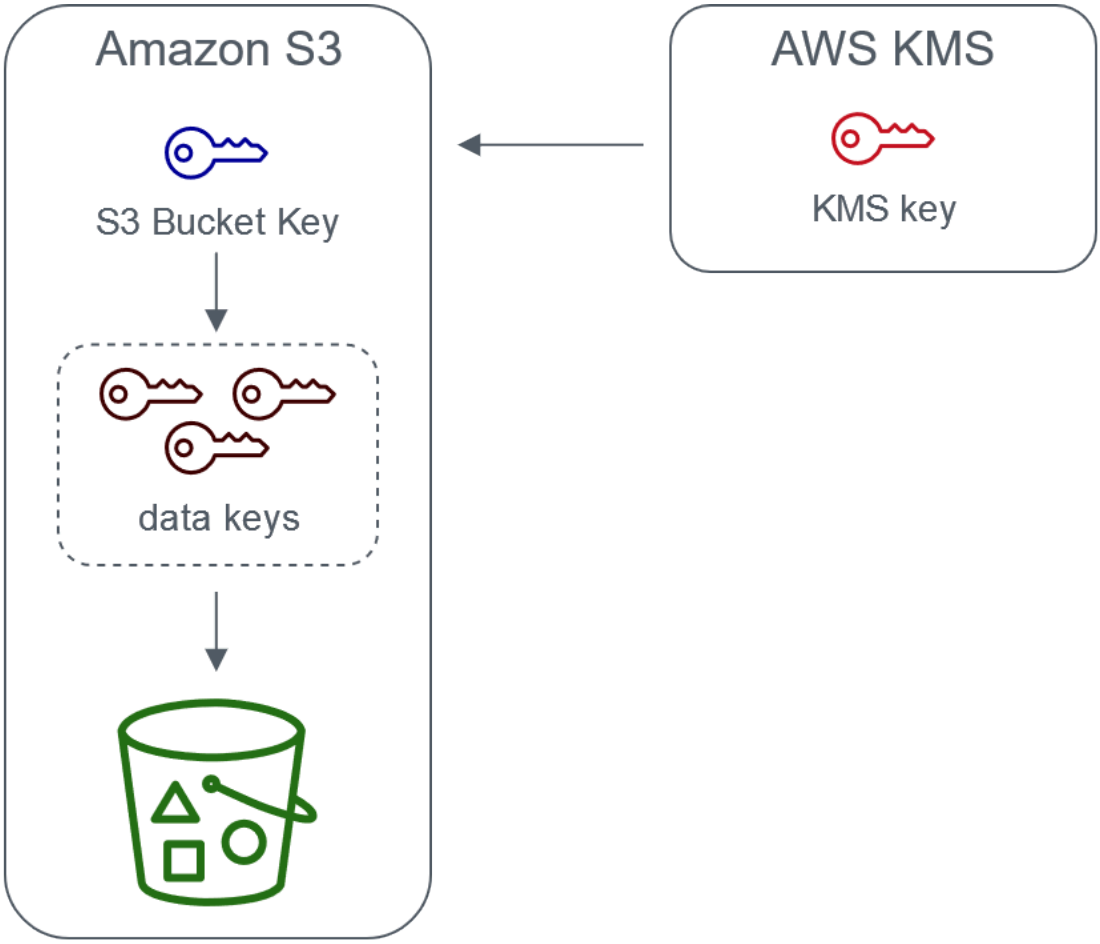
Wenn Sie Ihren Bucket so konfigurieren, dass er einen S3-Bucket-Key für SSE - verwendetKMS, AWS generiert es einen kurzlebigen Schlüssel auf Bucket-Ebene von AWS KMS und speichert ihn dann vorübergehend in S3. Dieser Schlüssel auf Bucket-Ebene erstellt während seines Lebenszyklus Datenschlüssel für neue Objekte. S3-Bucket-Keys werden für einen begrenzten Zeitraum in Amazon S3 verwendet, sodass S3 weniger Anfragen stellen muss, AWS KMS um Verschlüsselungsvorgänge abzuschließen. Dadurch wird der Datenverkehr von S3 zu reduziert AWS KMS, sodass Sie zu einem Bruchteil der bisherigen Kosten auf AWS KMS-verschlüsselte Objekte in Amazon S3 zugreifen können.

Eindeutige Schlüssel auf Bucket-Ebene werden mindestens einmal pro Anforderer abgerufen, um sicherzustellen, dass der Zugriff des Anforderers auf den Schlüssel bei einem Ereignis erfasst wird. AWS KMS CloudTrail Amazon S3 behandelt Anrufer als unterschiedliche Anforderer, wenn sie unterschiedliche Rollen oder Konten oder dieselbe Rolle mit unterschiedlichen Bereichsrichtlinien verwenden. AWS KMS Die Einsparungen bei Anfragen spiegeln die Anzahl der Anfragenden, das Anforderungsmuster und das relative Alter der angeforderten Objekte wider. Beispielsweise führt eine geringere Anzahl von Anforderern, die mehrere mit demselben Schlüssel auf Bucket-Ebene verschlüsselte Objekte in einem begrenzten Zeitfenster anfordern, zu größeren Einsparungen.

> **Note**
>
> Durch die Verwendung von S3 Bucket Keys können Sie AWS KMS Anforderungskosten sparen `EncryptGenerateDataKey`, indem Sie Ihre Anfragen mithilfe eines Schlüssels auf Bucketebene auf Anfragen und `Decrypt` Operationen reduzieren. AWS KMS Standardmäßig führen nachfolgende Anfragen, die diesen Schlüssel auf Bucket-Ebene nutzen, nicht zu AWS KMS API Anfragen oder zur Überprüfung des Zugriffs anhand der Schlüsselrichtlinie. AWS KMS

Wenn Sie einen S3-Bucket-Schlüssel konfigurieren, verwenden Objekte, die sich bereits im Bucket befinden, nicht den S3-Bucket-Schlüssel. Zum Konfigurieren eines S3-Bucket-Schlüssels für vorhandene Objekte können Sie eine `CopyObject`-Operation verwenden. Weitere Informationen finden Sie unter Konfigurieren eines S3-Bucket-Schlüssels auf Objektebene .

Amazon S3 gibt einen S3-Bucket-Schlüssel nur für Objekte frei, die mit demselben AWS KMS key verschlüsselt werden. S3-Bucket-Keys sind kompatibel mit KMS Schlüsseln AWS KMS, die von importiertem Schlüsselmaterial erstellt wurden, und Schlüsselmaterial, das von benutzerdefinierten Schlüsselspeichern unterstützt wird.



Server-side encryption with AWS Key Management service using an S3 Bucket Key

Konfigurieren von S3-Bucket-Schlüsseln

Sie können Ihren Bucket so konfigurieren, dass er einen S3-Bucket Key für SSE — KMS für neue Objekte über die Amazon S3 S3-Konsole AWS SDKs, AWS CLI, oder verwendet RESTAPI. Wenn S3-Bucket Keys in Ihrem Bucket aktiviert sind, verwenden Objekte, die mit einem anderen angegebenen SSE KMS Schlüssel hochgeladen wurden, ihre eigenen S3-Bucket-Keys. Unabhängig

von Ihrer S3-Bucket-Schlüsseleinstellung können Sie den Header `x-amz-server-side-encryption-bucket-key-enabled` mit einem `true`- oder `false`-Wert in Ihre Anforderung aufnehmen, um die Bucket-Einstellung zu überschreiben.

Bevor Sie Ihren Bucket für die Verwendung eines S3-Bucket-Schlüssels konfigurieren, lesen Sie Änderungen, die Sie vor dem Aktivieren eines S3-Bucket-Schlüssels beachten sollten.

Konfigurieren eines S3-Bucket-Schlüssels mit der Amazon-S3-Konsole

Wenn Sie einen neuen Bucket erstellen, können Sie Ihren Bucket so konfigurieren, dass er einen S3-Bucket-Key für SSE — KMS für neue Objekte verwendet. Sie können einen vorhandenen Bucket auch so konfigurieren, dass er einen S3-Bucket-Key für SSE — KMS für neue Objekte verwendet, indem Sie Ihre Bucket-Eigenschaften aktualisieren.

Weitere Informationen finden Sie unter Konfigurieren Sie Ihren Bucket so, dass er einen S3-Bucket-Key mit SSE - KMS für neue Objekte verwendet.

RESTAPI AWS CLI, und AWS SDK Unterstützung für S3-Bucket Keys

Sie können das RESTAPI, oder verwenden AWS CLI, um Ihren Bucket so AWS SDK zu konfigurieren, dass er einen S3-Bucket-Key für SSE - KMS für neue Objekte verwendet. Sie können einen S3-Bucket-Schlüssel auch auf Objektebene aktivieren.

Weitere Informationen finden Sie hier:

- Konfigurieren eines S3-Bucket-Schlüssels auf Objektebene
- Konfigurieren Sie Ihren Bucket so, dass er einen S3-Bucket-Key mit SSE - KMS für neue Objekte verwendet

Die folgenden API Operationen unterstützen S3-Bucket-Keys für SSE -KMS:

- PutBucketEncryption
  - `ServerSideEncryptionRule` akzeptiert den Parameter `BucketKeyEnabled` zum Aktivieren und Deaktivieren eines S3-Bucket-Schlüssels.
- GetBucketEncryption
  - `ServerSideEncryptionRule` gibt die Einstellungen für `BucketKeyEnabled` zurück.
- PutObject, CopyObjectCreateMultipartUpload, und POSTObjekt
  - Der Anforderungs-Header `x-amz-server-side-encryption-bucket-key-enabled` aktiviert oder deaktiviert einen S3-Bucket-Schlüssel auf Objektebene.

- HeadObject, GetObjectUploadPartCopy, UploadPart, und CompleteMultipartUpload
  - Der Antwort-Header `x-amz-server-side-encryption-bucket-key-enabled` zeigt an, ob ein S3-Bucket-Schlüssel für ein Objekt aktiviert oder deaktiviert ist.

Arbeitet mit AWS CloudFormation

In AWS CloudFormation enthält die `AWS::S3::Bucket` Ressource eine Verschlüsselungseigenschaft namens`BucketKeyEnabled`, mit der Sie einen S3-Bucket-Key aktivieren oder deaktivieren können.

Weitere Informationen finden Sie unter Verwenden AWS CloudFormation.

Änderungen, die Sie vor dem Aktivieren eines S3-Bucket-Schlüssels beachten sollten

Bevor Sie einen S3-Bucket-Schlüssel aktivieren, beachten Sie bitte die folgenden damit verbundenen Änderungen:

IAModer AWS KMS wichtige Richtlinien

Wenn Ihre bestehenden AWS Identity and Access Management (IAM) Richtlinien oder AWS KMS Schlüsselrichtlinien Ihr Objekt Amazon Resource Name (ARN) als Verschlüsselungskontext verwenden, um den Zugriff auf Ihren KMS Schlüssel zu verfeinern oder einzuschränken, funktionieren diese Richtlinien nicht mit einem S3-Bucket Key. S3-Bucket-Keys verwenden den Bucket ARN als Verschlüsselungskontext. Bevor Sie einen S3-Bucket Key aktivieren, aktualisieren Sie Ihre IAM Richtlinien oder AWS KMS Schlüsselrichtlinien, um Ihren Bucket ARN als Verschlüsselungskontext zu verwenden.

Weitere Informationen zum Verschlüsselungskontext und zu S3-Bucket-Schlüsseln finden Sie unter Verschlüsselungskontext.

CloudTrail Ereignisse für AWS KMS

Nachdem Sie einen S3-Bucket Key aktiviert haben, protokollieren Ihre AWS KMS CloudTrail Ereignisse Ihren ARN Bucket und nicht Ihr ObjektARN. Darüber hinaus werden in Ihren Protokollen weniger KMS CloudTrail Ereignisse für SSE KMS -Objekte angezeigt. Da wichtige Materialien in Amazon S3 zeitlich begrenzt sind, werden weniger Anfragen gestellt. AWS KMS

Verwenden eines S3-Bucket-Schlüssels mit Replikation

Sie können S3-Bucket-Keys mit Replikation in derselben Region (SRR) und regionsübergreifender Replikation () verwenden. CRR

Wenn Amazon S3 ein verschlüsseltes Objekt repliziert, werden im Allgemeinen die Verschlüsselungseinstellungen des Replikatobjekts im Ziel-Bucket beibehalten. Wenn das Quellobjekt jedoch nicht verschlüsselt ist und Ihr Ziel-Bucket eine Standard-Verschlüsselung oder einen S3-Bucket-Schlüssel verwendet, verschlüsselt Amazon S3 das Objekt mit der Konfiguration des Ziel-Buckets.

Die folgenden Beispiele veranschaulichen, wie ein S3-Bucket-Schlüssel mit der Replikation funktioniert. Weitere Informationen finden Sie unter Replikation verschlüsselter Objekte (SSE-S3, SSE -, DSSE - KMSKMS, SSE -C).

Example Beispiel 1 – Quellobjekt verwendet S3-Bucket-Schlüssel; Ziel-Bucket verwendet Standardverschlüsselung

Wenn Ihr Quellobjekt einen S3-Bucket Key verwendet, Ihr Ziel-Bucket jedoch die Standardverschlüsselung mit SSE - verwendetKMS, behält das Replikatobjekt seine S3-Bucket-Key-Verschlüsselungseinstellungen im Ziel-Bucket bei. Der Ziel-Bucket verwendet weiterhin die Standardverschlüsselung mit SSE -KMS.

Example Beispiel 2 — Das Quellobjekt ist nicht verschlüsselt; der Ziel-Bucket verwendet einen S3-Bucket-Schlüssel mit SSE - KMS

Wenn Ihr Quellobjekt nicht verschlüsselt ist und der Ziel-Bucket einen S3-Bucket-Schlüssel mit SSE - verwendetKMS, wird das Replikat-Objekt mithilfe eines S3-Bucket-Schlüssels mit SSE — KMS im Ziel-Bucket verschlüsselt. Daher unterscheidet sich das `ETag` des Quell-Objekts von dem `ETag` des Replikatobjekts. Sie müssen die Anwendungen, die das `ETag` verwenden, zur Anpassung an diesen Unterschied aktualisieren.

Arbeiten mit S3-Bucket-Schlüsseln

Weitere Informationen zum Aktivieren und Arbeiten mit S3-Bucket-Schlüsseln finden Sie in den folgenden Abschnitten:

- Konfigurieren Sie Ihren Bucket so, dass er einen S3-Bucket-Key mit SSE - KMS für neue Objekte verwendet
- Konfigurieren eines S3-Bucket-Schlüssels auf Objektebene
- Anzeigen der Einstellungen für einen S3-Bucket-Schlüssel

Konfigurieren Sie Ihren Bucket so, dass er einen S3-Bucket-Key mit SSE - KMS für neue Objekte verwendet

Wenn Sie die serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) - Schlüsseln (SSE-KMS) konfigurieren, können Sie Ihren Bucket so konfigurieren, dass für neue Objekte ein S3-Bucket-Key für SSE - KMS verwendet wird. S3-Bucket-Keys reduzieren den Anforderungsverkehr von Amazon S3 zu AWS KMS und reduzieren die Kosten für SSE -KMS. Weitere Informationen finden Sie unter Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys.

Sie können Ihren Bucket so konfigurieren, dass er einen S3-Bucket Key für SSE — KMS für neue Objekte verwendet, indem Sie die Amazon S3 S3-Konsole RESTAPI, AWS SDKs,, AWS Command Line Interface (AWS CLI) oder verwenden AWS CloudFormation. Wenn Sie einen S3-Bucket-Schlüssel für vorhandene Objekte aktivieren oder deaktivieren möchten, können Sie eine `CopyObject`-Operation verwenden. Weitere Informationen finden Sie unter Konfigurieren eines S3-Bucket-Schlüssels auf Objektebene und Verwenden von Batch-Operationen zur Aktivierung von S3-Bucket-Keys für SSE - KMS.

Wenn ein S3-Bucket Key für den Quell- oder Ziel-Bucket aktiviert ist, ist der Verschlüsselungskontext beispielsweise der Bucket Amazon Resource Name (ARN) und nicht das Objekt ARN`arn:aws:s3:::`*`bucket_ARN`*. Sie müssen Ihre IAM Richtlinien aktualisieren, um den Bucket ARN für den Verschlüsselungskontext verwenden zu können. Weitere Informationen finden Sie unter S3 Bucket-Schlüssel und Replikation.

Die folgenden Beispiele veranschaulichen, wie ein S3-Bucket-Schlüssel mit der Replikation funktioniert. Weitere Informationen finden Sie unter Replikation verschlüsselter Objekte (SSE-S3, SSE -, DSSE - KMSKMS, SSE -C).

Voraussetzungen

Bevor Sie Ihren Bucket für die Verwendung eines S3-Bucket-Schlüssels konfigurieren, lesen Sie Änderungen, die Sie vor dem Aktivieren eines S3-Bucket-Schlüssels beachten sollten.

Verwenden der S3-Konsole

In der S3-Konsole können Sie einen S3-Bucket-Schlüssel für einen neuen oder vorhandenen Bucket aktivieren oder deaktivieren. Objekte in der S3-Konsole übernehmen ihre S3-Bucket-Schlüssel-Einstellung aus der Bucket-Konfiguration. Wenn Sie einen S3-Bucket Key für Ihren Bucket aktivieren, verwenden neue Objekte, die Sie in den Bucket hochladen, einen S3-Bucket Key für SSE -KMS.

Hochladen, Kopieren oder Ändern von Objekten in Buckets, für die ein S3-Bucket-Schlüssel aktiviert ist

Wenn Sie ein Objekt in einen Bucket hochladen oder kopieren, für den ein S3-Bucket-Schlüssel aktiviert ist, oder ein Objekt darin ändern, werden die Einstellung des S3-Bucket-Schlüssels für dieses Objekt möglicherweise aktualisiert, um sie an die Bucket-Konfiguration anzupassen.

Wenn für ein Objekt bereits ein S3-Bucket-Schlüssel aktiviert ist, ändern sich die S3-Bucket-Schlüssel-Einstellungen für dieses Objekt nicht, wenn Sie das Objekt kopieren oder ändern. Wenn Sie jedoch ein Objekt ändern oder kopieren, für das kein S3-Bucket-Schlüssel aktiviert ist, und der Ziel-Bucket eine S3-Bucket-Schlüssel-Konfiguration hat, übernimmt das Objekt die S3-Bucket-Schlüssel-Einstellungen des Ziel-Buckets. Wenn beispielsweise für den Ziel-Bucket ein S3-Bucket-Schlüssel aktiviert ist, für das Quellobjekt jedoch nicht, wird ein S3-Bucket-Schlüssel für das Objekt aktiviert.

So aktivieren Sie einen S3-Bucket-Schlüssel beim Erstellen eines neuen Buckets

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie Create Bucket (Bucket erstellen) aus.
4. Geben Sie Ihren Bucket-Namen ein und wählen Sie Ihre AWS-Region aus.
5. Wählen Sie unter Standardverschlüsselung für Verschlüsselungsschlüsseltyp die Option AWS Key Management Service Schlüssel (SSE-KMS) aus.
6. Führen Sie unter AWS KMS Schlüssel einen der folgenden Schritte aus, um Ihren KMS Schlüssel auszuwählen:

   - Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys und wählen Sie dann Ihren KMSSchlüssel aus der Liste der verfügbaren Schlüssel aus.

     Sowohl der Von AWS verwalteter Schlüssel (`aws/s3`) als auch Ihr vom Kunden verwalteter Schlüssel werden in dieser Liste angezeigt. Weitere Informationen zu vom Kunden verwalteten Schlüsseln finden Sie unter Kundenschlüssel und AWS Schlüssel im AWS Key Management Service Entwicklerhandbuch.
   - Um den KMS Schlüssel einzugebenARN, wählen Sie Enter AWS KMS key ARN und geben Sie Ihren KMS Schlüssel ARN in das angezeigte Feld ein.

- Um einen neuen, vom Kunden verwalteten Schlüssel in der AWS KMS Konsole zu erstellen, wählen Sie Create a KMS key aus.

  Weitere Informationen zum Erstellen eines finden Sie AWS KMS key unter Creating Keys im AWS Key Management Service Developer Guide.

7. Wählen Sie unter Bucket Key (Bucket-Schlüssel) die Option Enable (Aktivieren).
8. Wählen Sie Create Bucket (Bucket erstellen) aus.

   Amazon S3 erstellt Ihren Bucket mit einem aktivierten S3-Bucket-Schlüssel. Neue Objekte, die Sie in den Bucket hochladen, verwenden einen S3-Bucket-Schlüssel.

   Um einen S3-Bucket-Schlüssel zu deaktivieren, führen Sie die vorherigen Schritte aus und wählen Deaktivieren.

So aktivieren Sie einen S3-Bucket-Schlüssel für einen vorhandenen Bucket

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Bucket, für den Sie einen S3-Bucket-Schlüssel aktivieren möchten.
4. Wählen Sie die Registerkarte Eigenschaften aus.
5. Wählen Sie unter Default encryption (Standard-Verschlüsselung) Edit (Bearbeiten) aus.
6. Wählen Sie unter Standardverschlüsselung für Verschlüsselungsschlüsseltyp die Option AWS Key Management Service Schlüssel (SSE-KMS) aus.
7. Führen Sie unter AWS KMS Schlüssel einen der folgenden Schritte aus, um Ihren KMS Schlüssel auszuwählen:

   - Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys und wählen Sie dann Ihren KMSSchlüssel aus der Liste der verfügbaren Schlüssel aus.

     Sowohl der Von AWS verwalteter Schlüssel (`aws/s3`) als auch Ihr vom Kunden verwalteter Schlüssel werden in dieser Liste angezeigt. Weitere Informationen zu vom Kunden verwalteten Schlüsseln finden Sie unter Kundenschlüssel und AWS Schlüssel im AWS Key Management Service Entwicklerhandbuch.

- Um den KMS Schlüssel einzugebenARN, wählen Sie Enter AWS KMS key ARN und geben Sie Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
- Um einen neuen, vom Kunden verwalteten Schlüssel in der AWS KMS Konsole zu erstellen, wählen Sie Create a KMS key aus.

  Weitere Informationen zum Erstellen eines finden Sie AWS KMS key unter Creating Keys im AWS Key Management Service Developer Guide.

8. Wählen Sie unter Bucket-Schlüssel die Option Aktivieren.
9. Wählen Sie Save Changes (Änderungen speichern).

   Amazon S3 aktiviert einen S3-Bucket-Schlüssel für neue Objekte, die zu Ihrem Bucket hinzugefügt werden. Bestehende Objekte verwenden den S3-Bucket-Schlüssel nicht. Zum Konfigurieren eines S3-Bucket-Schlüssels für vorhandene Objekte können Sie eine `CopyObject`-Operation verwenden. Weitere Informationen finden Sie unter Konfigurieren eines S3-Bucket-Schlüssels auf Objektebene .

   Um einen S3-Bucket-Schlüssel zu deaktivieren, führen Sie die vorherigen Schritte aus und wählen Deaktivieren.

Mit dem REST API

Sie können PutBucketEncryptioneinen S3-Bucket-Key für Ihren Bucket aktivieren oder deaktivieren. Um einen S3-Bucket Key mit zu konfigurieren`PutBucketEncryption`, verwenden Sie den ServerSideEncryptionRuleDatentyp, der die Standardverschlüsselung mit SSE - beinhaltetKMS. Sie können optional auch einen vom Kunden verwalteten Schlüssel verwenden, indem Sie die KMS Schlüssel-ID für den vom Kunden verwalteten Schlüssel angeben.

Weitere Informationen und eine Beispielsyntax finden Sie unter PutBucketEncryption.

Verwendung des AWS SDK for Java

Das folgende Beispiel aktiviert die Standard-Bucket-Verschlüsselung mit SSE - KMS und einem S3-Bucket-Schlüssel mithilfe von AWS SDK for Java.

Java

```
AmazonS3 s3client = AmazonS3ClientBuilder.standard()
    .withRegion(Regions.DEFAULT_REGION)
```

```
    .build();

ServerSideEncryptionByDefault serverSideEncryptionByDefault = new
 ServerSideEncryptionByDefault()
    .withSSEAlgorithm(SSEAlgorithm.KMS);
ServerSideEncryptionRule rule = new ServerSideEncryptionRule()
    .withApplyServerSideEncryptionByDefault(serverSideEncryptionByDefault)
    .withBucketKeyEnabled(true);
ServerSideEncryptionConfiguration serverSideEncryptionConfiguration =
    new ServerSideEncryptionConfiguration().withRules(Collections.singleton(rule));

SetBucketEncryptionRequest setBucketEncryptionRequest = new
 SetBucketEncryptionRequest()
    .withServerSideEncryptionConfiguration(serverSideEncryptionConfiguration)
    .withBucketName(bucketName);

s3client.setBucketEncryption(setBucketEncryptionRequest);
```

Mit dem AWS CLI

Das folgende Beispiel aktiviert die Standard-Bucket-Verschlüsselung mit SSE - KMS und einem S3-Bucket-Schlüssel mithilfe von AWS CLI. Ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws s3api put-bucket-encryption --bucket amzn-s3-demo-bucket --server-side-encryption-
configuration '{
        "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "aws:kms",
                    "KMSMasterKeyID": "KMS-Key-ARN"
                },
                "BucketKeyEnabled": true
            }
        ]
    }'
```

Verwenden AWS CloudFormation

Weitere Informationen zur Konfiguration eines S3-Bucket Keys mit AWS CloudFormation finden Sie unter AWS: :S3: :Bucket ServerSideEncryptionRule im AWS CloudFormation Benutzerhandbuch.

Konfigurieren eines S3-Bucket-Schlüssels auf Objektebene

Wenn Sie eine PUT COPY Oder-Operation mit dem RESTAPI,, oder AWS CLI ausführen AWS SDKs, können Sie einen S3-Bucket-Key auf Objektebene aktivieren oder deaktivieren, indem Sie den `x-amz-server-side-encryption-bucket-key-enabled` Anforderungsheader mit einem `true` `false` Oder-Wert hinzufügen. S3-Bucket-Keys reduzieren die Kosten für serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) (SSE-KMS), indem sie den Anforderungsverkehr von Amazon S3 zu AWS KMS verringern. Weitere Informationen finden Sie unter Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys.

Wenn Sie mithilfe einer COPY Operation PUT oder einen S3-Bucket Key für ein Objekt konfigurieren, aktualisiert Amazon S3 nur die Einstellungen für dieses Objekt. Die S3-Bucket-Schlüssel-Einstellungen für den Ziel-Bucket ändern sich nicht. Wenn Sie eine COPY Oder-Anfrage für ein PUT KMS -verschlüsseltes Objekt an einen Bucket mit aktivierten S3-Bucket-Keys senden, verwendet Ihr Vorgang auf Objektebene automatisch S3-Bucket-Keys, sofern Sie die Schlüssel im Anforderungsheader nicht deaktivieren. Wenn Sie keinen S3-Bucket-Schlüssel für Ihr Objekt angeben, wendet Amazon S3 die S3-Bucket-Schlüssel-Einstellungen für den Ziel-Bucket auf das Objekt an.

Voraussetzung:

Bevor Sie Ihr Objekt für die Verwendung eines S3-Bucket-Schlüssels konfigurieren, lesen Sie Änderungen, die Sie vor dem Aktivieren eines S3-Bucket-Schlüssels beachten sollten.

Themen

- Amazon S3 Batchvorgänge
- Verwenden von REST API
- Verwenden von AWS SDK for Java (PutObject)
- Verwenden von AWS CLI (PutObject)

Amazon S3 Batchvorgänge

Um Ihre vorhandenen Amazon-S3-Objekte zu verschlüsseln, können Sie Amazon-S3-Batchvorgänge verwenden. Sie stellen S3 Batch Operations eine Liste von Objekten zur Verfügung, mit denen gearbeitet werden soll, und Batch Operations ruft die entsprechenden Objekte aufAPI, um die angegebene Operation auszuführen.

Mit dem S3-Batch–Vorgangs-Kopiervorgang können Sie vorhandene nicht verschlüsselte Objekte kopieren und sie in denselben Bucket zurückschreiben, wie verschlüsselte Objekte. Ein einzelner Batch-Vorgangsauftrag kann die angegebene Operation für Milliarden von Objekten ausführen. Weitere Informationen finden Sie unter Massenausführung von Objektoperationen mit Batch Operations und Encrypting Objects with Amazon S3 Batch Operations (Verschlüsseln von Objekten mit Amazon S3 Batch Operations).

Verwenden von REST API

Wenn Sie SSE - verwendenKMS, können Sie einen S3-Bucket-Key für ein Objekt aktivieren, indem Sie die folgenden API Operationen verwenden:

- PutObject— Wenn Sie ein Objekt hochladen, können Sie den `x-amz-server-side-encryption-bucket-key-enabled` Anforderungsheader angeben, um einen S3-Bucket-Key auf Objektebene zu aktivieren oder zu deaktivieren.
- CopyObject— Wenn Sie ein Objekt kopieren und SSE - konfigurierenKMS, können Sie den `x-amz-server-side-encryption-bucket-key-enabled` Anforderungsheader angeben, um einen S3-Bucket-Key für Ihr Objekt zu aktivieren oder zu deaktivieren.
- POSTObjekt — Wenn Sie einen `POST` Vorgang verwenden, um ein Objekt hochzuladen und SSE - zu konfigurierenKMS, können Sie das `x-amz-server-side-encryption-bucket-key-enabled` Formularfeld verwenden, um einen S3-Bucket-Key für Ihr Objekt zu aktivieren oder zu deaktivieren.
- CreateMultipartUpload— Wenn Sie große Objekte mithilfe der `CreateMultipartUpload` API Operation hochladen und SSE - konfigurierenKMS, können Sie den `x-amz-server-side-encryption-bucket-key-enabled` Anforderungsheader verwenden, um einen S3-Bucket-Key für Ihr Objekt zu aktivieren oder zu deaktivieren.

Um einen S3-Bucket-Schlüssel auf Objektebene zu aktivieren, schließen Sie den Anforderungs-Header `x-amz-server-side-encryption-bucket-key-enabled` ein. Weitere Informationen zu SSE — KMS und dem REST API finden Sie unterUnter Verwendung der REST API.

Verwenden von AWS SDK for Java (PutObject)

Sie können das folgende Beispiel verwenden, um einen S3-Bucket-Schlüssel auf Objektebene mit AWS SDK for Java zu konfigurieren.

Java

```
AmazonS3 s3client = AmazonS3ClientBuilder.standard()
    .withRegion(Regions.DEFAULT_REGION)
    .build();

String bucketName = "amzn-s3-demo-bucket1";
String keyName = "key name for object";
String contents = "file contents";

PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName, keyName,
 contents)
    .withBucketKeyEnabled(true);

s3client.putObject(putObjectRequest);
```

Verwenden von AWS CLI (PutObject)

Sie können das folgende AWS CLI Beispiel verwenden, um einen S3-Bucket-Key auf Objektebene als Teil einer `PutObject` Anfrage zu konfigurieren.

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key object key name --server-side-
encryption aws:kms --bucket-key-enabled --body filepath
```

Anzeigen der Einstellungen für einen S3-Bucket-Schlüssel

Sie können die Einstellungen für einen S3-Bucket Key auf Bucket- oder Objektebene mithilfe der Amazon S3 S3-Konsole RESTAPI,, AWS Command Line Interface (AWS CLI) oder anzeigen AWS SDKs.

S3-Bucket-Keys reduzieren den Anforderungsverkehr von Amazon S3 zu AWS KMS und reduzieren die Kosten für serverseitige Verschlüsselung mithilfe von AWS Key Management Service (SSE-KMS). Weitere Informationen finden Sie unter Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys.

Zum Anzeigen der Einstellungen des S3-Bucket-Schlüssels für einen Bucket oder ein Objekt, der/das die Einstellungen des S3-Bucket-Schlüssels von der Bucket-Konfiguration übernommen hat, benötigen Sie die Berechtigung zum Ausführen der Aktion `s3:GetEncryptionConfiguration`.

Weitere Informationen finden Sie GetBucketEncryptionin der Amazon Simple Storage Service API Reference.

Verwenden der S3-Konsole

In der S3-Konsole können Sie die S3-Bucket-Schlüssel-Einstellungen für Ihren Bucket oder Ihr Objekt anzeigen. S3-Bucket-Schlüssel-Einstellungen werden aus der Bucket-Konfiguration übernommen, es sei denn, für die Quellobjekte ist bereits ein S3-Bucket-Schlüssel konfiguriert.

Objekte und Ordner im selben Bucket können unterschiedliche S3-Bucket-Schlüssel-Einstellungen haben. Wenn Sie beispielsweise ein Objekt mit dem hochladen REST API und einen S3-Bucket-Key für das Objekt aktivieren, behält das Objekt seine S3-Bucket-Key-Einstellung im Ziel-Bucket bei, auch wenn S3 Bucket Key im Ziel-Bucket deaktiviert ist. Ein weiteres Beispiel: Wenn Sie einen S3-Bucket-Schlüssel für einen vorhandenen Bucket aktivieren, verwenden Objekte, die sich bereits im Bucket befinden, keinen S3-Bucket-Schlüssel. Für neue Objekte ist jedoch ein S3-Bucket-Schlüssel aktiviert.

So zeigen Sie die Einstellungen des S3-Bucket-Schlüssels für Ihren Bucket an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Bucket, für den Sie einen S3-Bucket-Schlüssel aktivieren möchten.
4. Wählen Sie Properties (Eigenschaften).
5. Im Bereich Standard-Verschlüsselung sehen Sie unter Bucket-Schlüssel die S3-Bucket-Schlüssel-Einstellung für Ihren Bucket.

   Wenn Sie die S3-Bucket-Schlüssel-Einstellung nicht sehen, sind Sie möglicherweise nicht berechtigt, die `s3:GetEncryptionConfiguration`-Aktion durchzuführen. Weitere Informationen finden Sie GetBucketEncryptionin der Amazon Simple Storage Service API Reference.

So zeigen Sie die S3-Bucket-Schlüssel-Einstellung für Ihr Objekt an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Bucket, für den Sie einen S3-Bucket-Schlüssel aktivieren möchten.

3. Wählen Sie in der Liste Objekte den Objektnamen aus.
4. Wählen Sie im Tab Details unter Serverseitige Verschlüsselungseinstellungen die Option Bearbeiten.

   Unter Bucket-Schlüssel sehen Sie die Einstellungen des S3-Bucket-Schlüssels für Ihr Objekt. Sie können diese Einstellungen nicht bearbeiten.

Verwenden Sie den AWS CLI

So zeigen Sie die S3-Bucket-Schlüssel-Einstellungen auf Bucket-Ebene an

Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie jeden *user input placeholder* durch Ihre Informationen.

```
aws s3api get-bucket-encryption --bucket amzn-s3-demo-bucket1
```

Weitere Informationen finden Sie get-bucket-encryptionin der AWS CLI Befehlsreferenz.

So zeigen Sie die Einstellungen für einen S3-Bucket-Schlüssel auf Objektebene an

Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie jeden *user input placeholder* durch Ihre Informationen.

```
aws s3api head-object --bucket amzn-s3-demo-bucket1 --key my_images.tar.bz2
```

Weitere Informationen finden Sie unter head-object in der AWS CLI -Befehlsreferenz.

Mit dem REST API

So zeigen Sie die S3-Bucket-Schlüssel-Einstellungen auf Bucket-Ebene an

Verwenden Sie die Operation `GetBucketEncryption`, um Verschlüsselungsinformationen für einen Bucket anzuzeigen, einschließlich der Einstellungen für einen S3-Bucket-Schlüssel. Die Einstellungen des S3-Bucket-Schlüssels werden im Antworttext im Element `ServerSideEncryptionConfiguration` mit der Einstellung `BucketKeyEnabled` angezeigt. Weitere Informationen finden Sie GetBucketEncryptionin der Amazon S3 API S3-Referenz.

So zeigen Sie die Einstellungen für einen S3-Bucket-Schlüssel auf Objektebene an

Verwenden Sie den Vorgang `HeadObject`, um den S3-Bucket-Schlüssel-Status für ein Objekt anzuzeigen. `HeadObject` liefert den Antwort-Header `x-amz-server-side-encryption-`

`bucket-key-enabled`, der zeigt, ob ein S3-Bucket-Schlüssel für das Objekt aktiviert oder deaktiviert ist. Weitere Informationen finden Sie HeadObjectin der Amazon S3 API S3-Referenz.

Die folgenden API Operationen geben auch den `x-amz-server-side-encryption-bucket-key-enabled` Antwort-Header zurück, wenn ein S3-Bucket-Key für ein Objekt konfiguriert ist:

- PutObject
- PostObject
- CopyObject
- CreateMultipartUpload
- UploadPartCopy
- UploadPart
- CompleteMultipartUpload
- GetObject

Verwendung einer zweischichtigen serverseitigen Verschlüsselung mit AWS KMS Schlüsseln (DSSE-) KMS

Bei Verwendung der serverseitigen Dual-Layer-Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (DSSE-KMS) werden Objekte beim Upload auf Amazon S3 mit zwei Verschlüsselungsebenen versehen. DSSE— KMS hilft Ihnen, Compliance-Standards, die eine mehrschichtige Verschlüsselung Ihrer Daten erfordern, einfacher zu erfüllen und die volle Kontrolle über Ihre Verschlüsselungsschlüssel zu haben.

Wenn Sie DSSE - KMS mit einem Amazon S3 S3-Bucket verwenden, müssen sich die AWS KMS Schlüssel in derselben Region wie der Bucket befinden. Wenn DSSE - für das Objekt angefordert KMS wird, wird außerdem die S3-Prüfsumme, die Teil der Metadaten des Objekts ist, in verschlüsselter Form gespeichert. Weitere Informationen zu Prüfsummen finden Sie unter Überprüfung der Objektintegrität.

Für die Nutzung von DSSE - KMS und AWS KMS keys fallen zusätzliche Gebühren an. Weitere Informationen zur DSSE KMS Preisgestaltung finden Sie unter AWS KMS key Konzepte im AWS Key Management Service Entwicklerhandbuch und unter AWS KMS Preise.

> **Note**
>
> S3-Bucket-Keys werden für DSSE - nicht unterstütztKMS.

Serverseitige Dual-Layer-Verschlüsselung mit AWS KMS keys (DSSE-) erforderlich KMS

Wenn Sie die serverseitige Dual-Layer-Verschlüsselung aller Objekte in einem bestimmten Amazon-S3-Bucket anfordern möchten, können Sie eine Bucket-Richtlinie verwenden. Die folgende Bucket-Richtlinie verweigert beispielsweise allen Benutzern die Berechtigung zum Hochladen von Objekten (`s3:PutObject`), wenn die Anfrage keinen `x-amz-server-side-encryption` Header enthält, der eine serverseitige Verschlüsselung mit - anfordert. DSSE KMS

```
{
            "Version":"2012-10-17",
            "Id":"PutObjectPolicy",
            "Statement":[{
                  "Sid":"DenyUnEncryptedObjectUploads",
                  "Effect":"Deny",
                  "Principal":"*",
                  "Action":"s3:PutObject",
                  "Resource":"arn:aws:s3:::amzn-s3-demo-bucket/*",
                  "Condition":{
                     "StringNotEquals":{
                        "s3:x-amz-server-side-encryption":"aws:kms:dsse"
                     }
                  }
               }
            ]
         }
```

Themen

- Angabe einer zweischichtigen serverseitigen Verschlüsselung mit AWS KMS Schlüsseln (-) DSSE KMS

Angabe einer zweischichtigen serverseitigen Verschlüsselung mit AWS KMS Schlüsseln (-) DSSE KMS

Sie können die Verschlüsselung anwenden, wenn Sie entweder ein neues Objekt hochladen oder ein vorhandenes Objekt kopieren.

Sie können DSSE — KMS mithilfe der Amazon S3-Konsole, Amazon S3 REST API und der AWS Command Line Interface (AWS CLI) angeben. Weitere Informationen finden Sie unter den folgenden Themen.

> **Note**
>
> Sie können Multiregion AWS KMS keys in Amazon S3 verwenden. Amazon S3 behandelt jedoch derzeit Multi-Regions-Schlüssel wie Einzel-Regions-Schlüssel und verwendet nicht die Multi-Regions-Funktionen des Schlüssels. Weitere Informationen finden Sie unter Verwenden von Schlüsseln für mehrere Regionen im AWS Key Management Service Entwicklerhandbuch.

> **Note**
>
> Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, müssen Sie über die entsprechende Berechtigung verfügen. Weitere Informationen zu kontoübergreifenden Berechtigungen für KMS Schlüssel finden Sie im AWS Key Management Service Entwicklerhandbuch unter Erstellen von KMS Schlüsseln, die andere Konten verwenden können.

Verwenden der S3-Konsole

In diesem Abschnitt wird beschrieben, wie Sie mithilfe der Amazon S3 S3-Konsole den Verschlüsselungstyp eines Objekts so einstellen oder ändern können, dass serverseitige Dual-Layer-Verschlüsselung mit AWS Key Management Service (AWS KMS KMS) -Schlüsseln (DSSE-) verwendet wird.

> **Note**
>
> - Sie können die Verschlüsselung eines Objekts ändern, wenn Ihr Objekt weniger als 5 GB groß ist. Wenn Ihr Objekt größer als 5 GB ist, müssen Sie AWS CLIoder verwenden, AWS SDKsum die Verschlüsselung eines Objekts zu ändern.
> - Eine Liste der zusätzlichen Berechtigungen, die zum Ändern der Verschlüsselung eines Objekts erforderlich sind, finden Sie unterthe section called “Erforderliche Berechtigungen für API S3-Operationen”. Richtlinien, die diese Berechtigung gewähren, finden Sie beispielsweise unterthe section called “Beispiele für identitätsbasierte Richtlinien”.
> - Wenn Sie die Verschlüsselung eines Objekts ändern, wird ein neues Objekt erstellt, um das alte zu ersetzen. Wenn S3-Versioning aktiviert ist, wird eine neue Version des

Objekts erstellt, und das vorhandene Objekt wird zu einer älteren Version. Die Rolle, die die Eigenschaft ändert, wird auch Besitzer des neuen Objekts (oder der neuen Objektversion).

So fügen Sie die Verschlüsselung für ein Objekt hinzu oder ändern Sie sie

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im Navigationsbereich Buckets und dann den Tab Allzweck-Buckets aus. Navigieren Sie zu dem Amazon S3 S3-Bucket oder -Ordner, der die Objekte enthält, die Sie ändern möchten.
3. Aktivieren Sie das Kontrollkästchen für die Objekte, die Sie ändern möchten.
4. Wählen Sie im Menü Aktionen aus der daraufhin angezeigten Optionsliste die Option Serverseitige Verschlüsselung bearbeiten aus.
5. Scrollen Sie zum Abschnitt Serverseitige Verschlüsselung.
6. Wählen Sie unter Verschlüsselungseinstellungen die Option Verwenden von Bucket-Einstellungen für die Standardverschlüsselung oder Überschreiben der Bucket-Einstellungen für die Standardverschlüsselung aus.
7. Wenn Sie Überschreiben der Bucket-Einstellungen für die Standardverschlüsselung ausgewählt haben, konfigurieren Sie die folgenden Verschlüsselungseinstellungen.

   a. Wählen Sie unter Verschlüsselungstyp die Option Serverseitige Dual-Layer-Verschlüsselung mit AWS Key Management Service Schlüsseln (DSSE-) aus. KMS

   b. Führen Sie unter AWS KMS Schlüssel einen der folgenden Schritte aus, um Ihren KMS Schlüssel auszuwählen:

      - Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys und wählen Sie dann Ihren KMSSchlüssel aus der Liste der verfügbaren Schlüssel aus.

        Sowohl der Von AWS verwalteter Schlüssel (`aws/s3`) als auch Ihr vom Kunden verwalteter Schlüssel werden in dieser Liste angezeigt. Weitere Informationen über vom Kunden verwaltete Schlüssel finden Sie unter Kundenschlüssel und AWS -Schlüssel im Entwicklerhandbuch zu AWS Key Management Service .

      - Um den KMS Schlüssel einzugebenARN, wählen Sie Enter AWS KMS key ARN und geben Sie dann Ihren KMS Schlüssel ARN in das angezeigte Feld ein.

- Um einen neuen, vom Kunden verwalteten Schlüssel in der AWS KMS Konsole zu erstellen, wählen Sie Create a KMS key aus.

  Weitere Informationen zum Erstellen eines AWS KMS key finden Sie unter Schlüssel erstellen im AWS Key Management Service Entwicklerhandbuch.

> **Important**
>
> Sie können nur KMS Schlüssel verwenden, die im selben AWS-Region Bucket verfügbar sind. Die Amazon S3 S3-Konsole listet nur die ersten 100 KMS Schlüssel in derselben Region wie der Bucket auf. Um einen KMS Schlüssel zu verwenden, der nicht aufgeführt ist, müssen Sie Ihren KMS Schlüssel eingebenARN. Wenn Sie einen KMS Schlüssel verwenden möchten, der einem anderen Konto gehört, benötigen Sie zunächst die Erlaubnis, den Schlüssel zu verwenden, und dann müssen Sie den KMS Schlüssel eingebenARN.
> Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel und keine asymmetrischen KMS Schlüssel. Weitere Informationen finden Sie unter Identifizieren asymmetrischer KMS Schlüssel im Entwicklerhandbuch.AWS Key Management Service

8. Wählen Sie unter Bucket-Schlüssel die Option Deaktivieren aus. S3-Bucket-Keys werden für DSSE - KMS nicht unterstützt.
9. Wählen Sie unter Zusätzliche Kopiereinstellungen aus, ob Sie Quelleinstellungen kopieren, Keine Einstellungen angeben oder Einstellungen angeben möchten. Quelleinstellungen kopieren ist die Standardoption. Wenn Sie das Objekt nur ohne die Quelleinstellungsattribute kopieren möchten, wählen Sie „Keine Einstellungen angeben“. Wählen Sie „Einstellungen angeben“, um Einstellungen für SpeicherklasseACLs, Objekt-Tags, Metadaten, serverseitige Verschlüsselung und zusätzliche Prüfsummen anzugeben.
10. Wählen Sie Änderungen speichern.

> **Note**
>
> Diese Aktion wendet auf alle angegebenen Objekte Verschlüsselung an. Warten Sie beim Verschlüsseln von Ordnern, bis die Speicheroperation abgeschlossen ist, bevor Sie dem Ordner neue Objekte hinzufügen.

Unter Verwendung der REST API

Wenn Sie ein Objekt erstellen, d. h. wenn Sie ein neues Objekt hochladen oder ein vorhandenes Objekt kopieren, können Sie die Verwendung einer zweischichtigen serverseitigen Verschlüsselung mit AWS KMS keys (-) angeben, um Ihre Daten zu verschlüsseln. DSSE KMS Fügen Sie hierzu der Anforderung den Header `x-amz-server-side-encryption` hinzu. Setzen Sie den Wert des Headers auf den `aws:kms:dsse`-Verschlüsselungsalgorithmus. Amazon S3 bestätigt, dass Ihr Objekt KMS verschlüsselt gespeichert wurde, indem es den Antwort-Header zurückgibt`x-amz-server-side-encryption`. DSSE

Wenn Sie den Header `x-amz-server-side-encryption` mit dem Wert `aws:kms:dsse` angeben, können Sie auch die folgenden Anforderungs-Header verwenden:

- `x-amz-server-side-encryption-aws-kms-key-id: SSEKMSKeyId`
- `x-amz-server-side-encryption-context: SSEKMSEncryptionContext`

Themen

- Amazon S3 REST API S3-Operationen, die unterstützen DSSE - KMS
- Verschlüsselungskontext (x-amz-server-side-encryption-context)
- AWS KMS Schlüssel-ID (x-amz-server-side-encryption-aws-kms-key-id)

Amazon S3 REST API S3-Operationen, die unterstützen DSSE - KMS

Die folgenden REST API Operationen akzeptieren die Header `x-amz-server-side-encryption`x-amz-server-side-encryption-aws-kms-key-id, und `x-amz-server-side-encryption-context` Request-Header.

- PutObject— Wenn Sie Daten mithilfe der PUT API Operation hochladen, können Sie diese Anforderungsheader angeben.
- CopyObject – Wenn Sie ein Objekt kopieren, erhalten Sie ein Quell- und ein Zielobjekt. Wenn Sie bei der `CopyObject` Operation DSSE KMS -Header übergeben, werden sie nur auf das Zielobjekt angewendet. Beim Kopieren eines vorhandenen Objekts wird das Zielobjekt unabhängig davon, ob das Quellobjekt verschlüsselt ist, nur dann verschlüsselt, wenn Sie die serverseitige Verschlüsselung explizit anfordern.
- POST Objekt — Wenn Sie eine `POST` Operation verwenden, um ein Objekt hochzuladen, geben Sie anstelle der Anforderungsheader dieselben Informationen in die Formularfelder ein.

- CreateMultipartUpload – Wenn Sie große Objekte über einen mehrteiligen Upload hochladen, können Sie diese Header in der `CreateMultipartUpload`-Anforderung angeben.

Die Antwortheader der folgenden REST API Operationen geben den `x-amz-server-side-encryption` Header zurück, wenn ein Objekt mit serverseitiger Verschlüsselung gespeichert wird.

- PutObject
- CopyObject
- POST Objekt
- CreateMultipartUpload
- UploadPart
- UploadPartCopy
- CompleteMultipartUpload
- GetObject
- HeadObject

> **Important**
>
> - Alle `GET` `PUT` Anfragen für ein Objekt, das durch geschützt ist, AWS KMS schlagen fehl, wenn Sie sie nicht mithilfe von Secure Sockets Layer (SSL), Transport Layer Security (TLS) oder Signature Version 4 stellen.
> - Wenn Ihr Objekt DSSE - verwendetKMS, senden Sie keine Header für `GET` Verschlüsselungsanfragen für Anfragen und `HEAD` Anfragen, da Sie sonst die Fehlermeldung HTTP400 (Bad Request) erhalten.

Verschlüsselungskontext (**`x-amz-server-side-encryption-context`**)

Wenn Sie angeben`x-amz-server-side-encryption:aws:kms:dsse`, API unterstützt Amazon S3 einen Verschlüsselungskontext mit dem `x-amz-server-side-encryption-context` Header. Ein Verschlüsselungskontext ist ein Satz von Schlüssel-Wert-Paaren, die zusätzliche kontextbezogene Informationen zu den Daten enthalten können.

Amazon S3 verwendet automatisch den Amazon-Ressourcennamen (ARN) des Objekts als Verschlüsselungskontextpaar; zum Beispiel`arn:aws:s3:::`*`object_ARN`*.

Sie können optional ein zusätzliches Verschlüsselungskontextpaar bereitstellen, indem Sie den Header `x-amz-server-side-encryption-context` verwenden. Da der Verschlüsselungskontext jedoch nicht verschlüsselt ist, sollte er keine sensiblen Informationen enthalten. Amazon S3 speichert dieses zusätzliche Schlüsselpaar zusammen mit dem Standardverschlüsselungskontext.

Weitere Informationen zum Verschlüsselungskontext in Amazon S3 finden Sie unter Verschlüsselungskontext. Allgemeine Informationen zum Verschlüsselungs-Kontext finden Sie unter AWS Key Management Service Concepts – Encryption Context (Konzepte – Verschlüsselungs-Kontext) im AWS Key Management Service -Entwicklerhandbuch.

AWS KMS Schlüssel-ID (**`x-amz-server-side-encryption-aws-kms-key-id`**)

Sie können den Header `x-amz-server-side-encryption-aws-kms-key-id` verwenden, um die ID des vom Kunden verwalteten Schlüssels anzugeben, der zum Schutz der Daten verwendet wird. Wenn Sie den `x-amz-server-side-encryption:aws:kms:dsse` Header angeben, den `x-amz-server-side-encryption-aws-kms-key-id` Header aber nicht angeben, verwendet Amazon S3 die Von AWS verwalteter Schlüssel (`aws/s3`), um die Daten zu schützen. Wenn Sie einen vom Kunden verwalteten Schlüssel verwenden möchten, müssen Sie den `x-amz-server-side-encryption-aws-kms-key-id`-Header des vom Kunden verwalteten Schlüssels angeben.

> **Important**
>
> Wenn Sie eine AWS KMS key für die serverseitige Verschlüsselung in Amazon S3 verwenden, müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel. Weitere Informationen zu diesen Schlüsseln finden Sie unter Symmetrische KMS Verschlüsselungsschlüssel im AWS Key Management Service Entwicklerhandbuch.

Mit dem AWS CLI

Wenn Sie ein neues Objekt hochladen oder ein vorhandenes Objekt kopieren, können Sie die Verwendung von DSSE - angeben, KMS um Ihre Daten zu verschlüsseln. Fügen Sie hierzu der Anforderung den Parameter `--server-side-encryption aws:kms:dsse` hinzu. Verwenden Sie den Parameter `--ssekms-key-id` *`example-key-id`*, um Ihren kundenverwalteten AWS KMS -Schlüssel hinzuzufügen, den Sie erstellt haben. Wenn Sie eine AWS KMS Schlüssel-ID angeben`--server-side-encryption aws:kms:dsse`, aber keine angeben, verwendet Amazon S3 den AWS verwalteten Schlüssel (`aws/s3`).

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key example-object-key --server-
side-encryption aws:kms:dsse --ssekms-key-id example-key-id --body filepath
```

Sie können ein unverschlüsseltes Objekt verschlüsseln, um es zu verwenden, DSSE KMS indem Sie das Objekt wieder an seinen Platz kopieren.

```
aws s3api copy-object --bucket amzn-s3-demo-bucket --key example-object-key --
body filepath --bucket amzn-s3-demo-bucket --key example-object-key --sse aws:kms:dsse
 --sse-kms-key-id example-key-id --body filepath
```

Verwenden Sie serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE

Die serverseitige Verschlüsselung dient zum Schutz ruhender Daten. Die serverseitige Verschlüsselung verschlüsselt nur die Objektdaten, nicht die Metadaten des Objekts. Durch die serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) können Sie Ihre Daten mit Ihren eigenen Verschlüsselungsschlüsseln verschlüsselt speichern. Mit dem Verschlüsselungsschlüssel, den Sie als Teil Ihrer Anforderung bereitstellen, verwaltet Amazon S3 die Datenverschlüsselung, wenn es auf Datenträger schreibt, und die Entschlüsselung, wenn Sie auf Ihre Objekte zugreifen. Sie müssen also für die Datenverschlüsselung und -entschlüsselung keinen Code mehr verwalten. Sie müssen nur noch die von Ihnen bereitgestellten Verschlüsselungsschlüssel verwalten.

Wenn Sie ein Objekt hochladen, verwendet Amazon S3 den von Ihnen angegebenen Verschlüsselungsschlüssel, um die AES 256-Verschlüsselung auf Ihre Daten anzuwenden. Amazon S3 entfernt dann den Verschlüsselungsschlüssel aus dem Speicher. Wenn Sie ein Objekt abrufen, müssen Sie denselben Verschlüsselungsschlüssel als Teil Ihrer Anfrage angeben. Amazon S3 überprüft zuerst, ob der von Ihnen bereitgestellte Verschlüsselungsschlüssel übereinstimmt, und entschlüsselt das Objekt, bevor Objektdaten zurückgegeben werden.

Für die Verwendung von SSE -C fallen keine zusätzlichen Gebühren an. Für Anfragen zur Konfiguration und Verwendung von SSE -C fallen jedoch Standardgebühren für Amazon S3 S3-Anfragen an. Informationen zu Preisen finden Sie unter Amazon S3 – Preise.

> **Note**
>
> Amazon S3 speichert den von Ihnen bereitgestellten Verschlüsselungsschlüssel nicht. Stattdessen wird ein zufällig gesalzener Hash-basierter Message Authentication Code (HMAC) -Wert des Verschlüsselungsschlüssels gespeichert, um future Anfragen zu

validieren. Der gesalzene HMAC Wert kann nicht verwendet werden, um den Wert des Verschlüsselungsschlüssels abzuleiten oder den Inhalt des verschlüsselten Objekts zu entschlüsseln. Das bedeutet, wenn Sie den Verschlüsselungsschlüssel verlieren, verlieren Sie das Objekt.

Die S3-Replikation unterstützt Objekte, die mit SSE -C verschlüsselt sind. Weitere Hinweise zur Replikation verschlüsselter Objekte finden Sie unter. the section called “Replizieren verschlüsselter Objekte”

Weitere Informationen zu SSE -C finden Sie in den folgenden Themen.

Themen

- SSE-C im Überblick
- -C erforderlich und einschränkend SSE
- Vorsigniert und URLs -C SSE
- Angeben der serverseitigen Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE

SSE-C im Überblick

Dieser Abschnitt bietet einen Überblick über SSE -C. Beachten Sie bei der Verwendung von SSE -C die folgenden Überlegungen.

- Sie müssen HTTPS verwenden.

> **Important**
>
> Amazon S3 lehnt alle Anfragen ab, die HTTP bei Verwendung von SSE -C gestellt werden. Aus Sicherheitsgründen empfehlen wir, dass Sie jeden Schlüssel, den Sie irrtümlich versenden, als HTTP kompromittiert betrachten. Verwerfen Sie den Schlüssel und rotieren Sie ihn wie erforderlich.

- Das Entity-Tag (ETag) in der Antwort ist nicht der MD5 Hash der Objektdaten.
- Sie Verwalten ein Mapping, welcher Verschlüsselungsschlüssel für die Verschlüsselung welches Objekts verwendet wurde. Amazon S3 speichert keine Verschlüsselungsschlüssel. Sie sind dafür verantwortlich, zu verwalten, welchen Verschlüsselungsschlüssel Sie für welches Objekt angegeben haben.

- Wenn für Ihren Bucket Versioning aktiviert ist, kann jede Objektversion, die Sie mit dieser Funktion hochladen, einen eigenen Verschlüsselungsschlüssel haben. Sie sind dafür verantwortlich, zu verwalten, welcher Verschlüsselungsschlüssel für welche Objektversion verwendet wurde.
- Sie verwalten die Verschlüsselungsschlüssel auf der Clientseite, deshalb verwalten Sie auch alle zusätzlichen Sicherungsmechanismen auf der Clientseite, wie beispielsweise die Schlüsselrotation.

> **Warning**
>
> Wenn Sie den Verschlüsselungsschlüssel verlieren, schlagen alle GET-Anfragen für ein Objekt ohne seinen Verschlüsselungsschlüssel fehl und Sie verlieren das Objekt.

-C erforderlich und einschränkend SSE

Um SSE -C für alle Objekte in einem bestimmten Amazon S3 S3-Bucket zu verlangen, können Sie eine Bucket-Richtlinie verwenden.

Die folgende Bucket-Richtlinie verweigert beispielsweise Upload object (`s3:PutObject`) -Berechtigungen für alle Anfragen, die nicht den `x-amz-server-side-encryption-customer-algorithm` Header enthalten, der -C anfordertSSE.

```
{
    "Version": "2012-10-17",
    "Id": "PutObjectPolicy",
    "Statement": [
        {
            "Sid": "RequireSSECObjectUploads",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "Null": {
                    "s3:x-amz-server-side-encryption-customer-algorithm": "true"
                }
            }
        }
    ]
```

```
}
```

Sie können auch eine Richtlinie verwenden, um die serverseitige Verschlüsselung aller Objekte in einem bestimmten Amazon-S3-Bucket einzuschränken. Die folgende Bucket-Richtlinie verweigert beispielsweise allen Benutzern die Berechtigung zum Hochladen von Objekten (`s3:PutObject`), wenn die Anfrage den `x-amz-server-side-encryption-customer-algorithm` Header enthält, der -C anfordert. SSE

```
{
    "Version": "2012-10-17",
    "Id": "PutObjectPolicy",
    "Statement": [
        {
            "Sid": "RestrictSSECObjectUploads",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "Null": {
                    "s3:x-amz-server-side-encryption-customer-algorithm": "false"
                }
            }
        }
    ]
}
```

> **Important**
>
> Wenn Sie eine Bucket-Richtlinie verwenden, bei der die Option SSE -C aktiviert werden muss`s3:PutObject`, müssen Sie den `x-amz-server-side-encryption-customer-algorithm` Header in alle mehrteiligen Upload-Anfragen (CreateMultipartUpload, UploadPart und) aufnehmen. CompleteMultipartUpload

Vorsigniert und URLs -C SSE

Sie können ein vorsigniertes Objekt generierenURL, das für Operationen wie das Hochladen eines neuen Objekts, das Abrufen eines vorhandenen Objekts oder das Abrufen von Objektmetadaten verwendet werden kann. Vorsignierte unterstützen URLs -C wie folgt: SSE

- Wenn Sie eine vorsignierte Datei erstellenURL, müssen Sie den Algorithmus angeben, indem Sie den `x-amz-server-side-encryption-customer-algorithm` Header in der Signaturberechnung verwenden.
- Wenn Sie Presigned verwenden, URL um ein neues Objekt hochzuladen, ein vorhandenes Objekt abzurufen oder nur Objektmetadaten abzurufen, müssen Sie alle Verschlüsselungsheader in der Anfrage Ihrer Client-Anwendung angeben.

> **Note**
>
> Für non-SSE-C Objekte können Sie ein vorsigniertes Objekt generieren URL und dieses direkt URL in einen Browser einfügen, um auf die Daten zuzugreifen.
> Bei SSE -C-Objekten ist dies jedoch nicht möglich, da Sie zusätzlich zu den vorsignierten Objekten auch HTTP Header angeben müssenURL, die für -C-Objekte spezifisch sind. SSE Daher können Sie vorsignierte Objekte nur programmgesteuert URLs für SSE -C-Objekte verwenden.

Weitere Hinweise zu presigned finden Sie unter. URLs the section called “Verwenden von presigned URLs zum Herunterladen und Hochladen von Objekten”

Angeben der serverseitigen Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE

Bei der Objekterstellung mit dem können Sie die REST API serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) angeben. Wenn Sie SSE -C verwenden, müssen Sie die Informationen zum Verschlüsselungsschlüssel mithilfe der folgenden Anforderungsheader angeben.

Name	Beschreibung
`x-amz-server-side-encryption-customer-algorithm`	Verwenden Sie diesen Header, um den Verschlüsselungsalgorithmus anzugeben. Der Header-Wert muss `AES256` sein.
`x-amz-server-side-encryption-customer-key`	Verwenden Sie diesen Header, um den base64-codierten 256-bit-V erschlüsselungsschlüssel für Amazon S3 bereitzustellen, mit dem Ihre Daten verschlüsselt und entschlüsselt werden.

Name	Beschreibung
`x-amz-server-side-encryption-customer-key-MD5`	Verwenden Sie diesen Header, um den Base64-codierten MD5 128-Bit-Digest des Verschlüsselungsschlüssels gemäß 1321 bereitzustellen. RFC Amazon S3 verwendet diesen Header für eine Überprüfung der Nachrichtenintegrität, um sicherzustellen, dass der Verschlüsselungsschlüssel fehlerfrei übertragen wurde.

Sie können AWS SDK Wrapper-Bibliotheken verwenden, um diese Header zu Ihrer Anfrage hinzuzufügen. Bei Bedarf können Sie die Amazon S3 REST API S3-Aufrufe direkt in Ihrer Anwendung tätigen.

> **Note**
>
> Sie können die Amazon S3 S3-Konsole nicht verwenden, um ein Objekt hochzuladen und SSE -C anzufordern. Sie können die Konsole auch nicht verwenden, um ein vorhandenes Objekt, das mit SSE -C gespeichert wurde, zu aktualisieren (z. B. die Speicherklasse zu ändern oder Metadaten hinzuzufügen).

Verwenden von REST API

Amazon S3 S3-RestAPIs, der SSE -C unterstützt

Die folgenden Amazon S3 APIs unterstützen serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C).

- GETVorgang — Beim Abrufen von Objekten mithilfe von GET API (siehe GETObjekt) können Sie die Anforderungsheader angeben.
- HEADOperation — Um Objektmetadaten mithilfe von HEAD API (siehe HEADObjekt) abzurufen, können Sie diese Anforderungsheader angeben.
- PUTOperation — Beim Hochladen von Daten mithilfe des PUT Objekts API (siehe PUTObjekt) können Sie diese Anforderungsheader angeben.
- Mehrteiliger Upload — Wenn Sie große Objekte mithilfe des mehrteiligen Uploads hochladenAPI, können Sie diese Header angeben. Sie geben diese Header in der Initiierungsanforderung (siehe Initiieren mehrteiliger Uploads) und in jeder nachfolgenden Anforderung für einen Teil-Upload an (siehe Hochladen von Teilen oder Hochladen von Teilen – Kopieren). Für jede teilweise Upload-

Anfrage muss dieselbe Verschlüsslungsinformation angegeben werden, wie diejenige, die Sie in der Initiierungsanfrage für den mehrteiligen Upload angegeben haben.

- POSTOperation — Wenn Sie einen POST Vorgang zum Hochladen eines Objekts (siehe POSTObjekt) verwenden, geben Sie anstelle der Anforderungsheader dieselben Informationen in die Formularfelder ein.
- Kopiervorgang — Wenn Sie ein Objekt kopieren (siehe PUTObjekt — Kopieren), haben Sie sowohl ein Quell- als auch ein Zielobjekt:
  - Wenn Sie möchten, dass das Zielobjekt serverseitig mit AWS verwalteten Schlüsseln verschlüsselt wird, müssen Sie den `x-amz-server-side-encryption` Anforderungsheader angeben.
  - Wenn Sie möchten, dass das Zielobjekt mit SSE -C verschlüsselt wird, müssen Sie Verschlüsselungsinformationen mithilfe der drei in der vorherigen Tabelle beschriebenen Header angeben.
  - Wenn das Quellobjekt mit SSE -C verschlüsselt ist, müssen Sie die Informationen zum Verschlüsselungsschlüssel mithilfe der folgenden Header angeben, damit Amazon S3 das Objekt zum Kopieren entschlüsseln kann.

Name	Beschreibung
`x-amz-copy-source-server-side-encryption-customer-algorithm`	Verwenden Sie diesen Header, um den Verschlüsselungsal gorithmus anzugeben, den Amazon S3 für die Entschlüsselung des Quellobjekts verwenden soll. Dieser Wert muss `AES256` lauten.
`x-amz-copy-source-server-side-encryption-customer-key`	Verwenden Sie diesen Header, um den base64-codierten 256-bit-Verschlüsselungsschlüssel für Amazon S3 bereitzustellen, mit dem das Quellobjekt entschlüsselt werden soll. Dieser Verschlüs selungsschlüssel muss derjenige sein, den Sie Amazon S3 beim Erstellen des Quellobjekts bereitgestellt haben. Andernfalls kann Amazon S3 das Objekt nicht entschlüsseln.
`x-amz-copy-source-server-side-encryption`	Fügen Sie diesen Header hinzu, um den Base64-codierten MD5 128-Bit-Digest des Verschlüsselungsschlüssels gemäß 1321 bereitzustellen. RFC

Name	Beschreibung
`-customer-key-MD5`	

Verwenden Sie die Option AWS SDKs, um SSE -C für PUT die Operationen, Head und Copy anzugeben GET

Die folgenden Beispiele zeigen, wie eine serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) für Objekte angefordert wird. Dabei werden folgende Vorgänge ausgeführt. Jeder Vorgang zeigt, wie SSE-C-related Header in der Anfrage angegeben werden:

- Put object – Lädt ein Objekt hoch und fordert die serverseitige Verschlüsselung mit einem vom Kunden bereitgestellten Verschlüsselungsschlüssel an.
- Get object – Lädt das im vorigen Schritt hochgeladene Objekt herunter. Sie stellen in der Anforderung dieselben Verschlüsselungs-Informationen bereit, die Sie beim Hochladen des Objekts angegeben haben. Amazon S3 benötigt diese Informationen, um das Objekt zu entschlüsseln und an Sie zurückzugeben.
- Get object metadata – Ruft die Metadaten des Objekts ab. Sie stellen dieselben Verschlüsselungs-Informationen bereit, die beim Erstellen des Objekts verwendet wurden.
- Copy object – Erstellt eine Kopie des zuvor hochgeladenen Objekts. Da das Quellobjekt mit SSE -C gespeichert wird, müssen Sie seine Verschlüsselungsinformationen in Ihrer Kopieranforderung angeben. Standardmäßig verschlüsselt Amazon S3 die Kopie des Objekts nur, wenn Sie dies ausdrücklich anfordern. In diesem Beispiel wird Amazon S3 dazu angewiesen, eine verschlüsselte Kopie des Objekts zu speichern.

Java

> **Note**
>
> Dieses Beispiel zeigt, wie ein Objekt in einer einzigen Operation hochgeladen wird. Wenn Sie den Multipart Upload verwenden, API um große Objekte hochzuladen, geben Sie die Verschlüsselungsinformationen auf die gleiche Weise ein, wie in diesem Beispiel gezeigt. Beispiele für mehrteilige Uploads, bei denen das verwendet wird, finden Sie unter AWS SDK for Java. Hochladen eines Objekts mit Multipart-Upload

Um die erforderlichen Verschlüsselungsinformationen hinzuzufügen, schließen Sie in Ihre Anfrage einen `SSECustomerKey` ein. Weitere Informationen zur `SSECustomerKey` Klasse finden Sie im REST API Abschnitt.

Hinweise zu SSE -C finden Sie unterVerwenden Sie serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE. Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

Example

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import javax.crypto.KeyGenerator;
import java.io.BufferedReader;
import java.io.File;
import java.io.IOException;
import java.io.InputStreamReader;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;

public class ServerSideEncryptionUsingClientSideEncryptionKey {
    private static SSECustomerKey SSE_KEY;
    private static AmazonS3 S3_CLIENT;
    private static KeyGenerator KEY_GENERATOR;

    public static void main(String[] args) throws IOException,
 NoSuchAlgorithmException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String keyName = "*** Key name ***";
        String uploadFileName = "*** File path ***";
        String targetKeyName = "*** Target key name ***";

        // Create an encryption key.
        KEY_GENERATOR = KeyGenerator.getInstance("AES");
```

```
        KEY_GENERATOR.init(256, new SecureRandom());
        SSE_KEY = new SSECustomerKey(KEY_GENERATOR.generateKey());

        try {
            S3_CLIENT = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Upload an object.
            uploadObject(bucketName, keyName, new File(uploadFileName));

            // Download the object.
            downloadObject(bucketName, keyName);

            // Verify that the object is properly encrypted by attempting to
retrieve it
            // using the encryption key.
            retrieveObjectMetadata(bucketName, keyName);

            // Copy the object into a new object that also uses SSE-C.
            copyObject(bucketName, keyName, targetKeyName);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }

    private static void uploadObject(String bucketName, String keyName, File file) {
        PutObjectRequest putRequest = new PutObjectRequest(bucketName, keyName,
file).withSSECustomerKey(SSE_KEY);
        S3_CLIENT.putObject(putRequest);
        System.out.println("Object uploaded");
    }

    private static void downloadObject(String bucketName, String keyName) throws
IOException {
        GetObjectRequest getObjectRequest = new GetObjectRequest(bucketName,
keyName).withSSECustomerKey(SSE_KEY);
```

```
        S3Object object = S3_CLIENT.getObject(getObjectRequest);

        System.out.println("Object content: ");
        displayTextInputStream(object.getObjectContent());
    }

    private static void retrieveObjectMetadata(String bucketName, String keyName) {
        GetObjectMetadataRequest getMetadataRequest = new
 GetObjectMetadataRequest(bucketName, keyName)
                .withSSECustomerKey(SSE_KEY);
        ObjectMetadata objectMetadata =
 S3_CLIENT.getObjectMetadata(getMetadataRequest);
        System.out.println("Metadata retrieved. Object size: " +
 objectMetadata.getContentLength());
    }

    private static void copyObject(String bucketName, String keyName, String
 targetKeyName)
            throws NoSuchAlgorithmException {
        // Create a new encryption key for target so that the target is saved using
        // SSE-C.
        SSECustomerKey newSSEKey = new SSECustomerKey(KEY_GENERATOR.generateKey());

        CopyObjectRequest copyRequest = new CopyObjectRequest(bucketName, keyName,
 bucketName, targetKeyName)
                .withSourceSSECustomerKey(SSE_KEY)
                .withDestinationSSECustomerKey(newSSEKey);

        S3_CLIENT.copyObject(copyRequest);
        System.out.println("Object copied");
    }

    private static void displayTextInputStream(S3ObjectInputStream input) throws
 IOException {
        // Read one line at a time from the input stream and display each line.
        BufferedReader reader = new BufferedReader(new InputStreamReader(input));
        String line;
        while ((line = reader.readLine()) != null) {
            System.out.println(line);
        }
        System.out.println();
    }
}
```

.NET

> **Note**
>
> Beispiele für das Hochladen großer Objekte mithilfe des mehrteiligen Uploads finden Sie unter API Hochladen eines Objekts mit Multipart-Upload und. Verwenden von AWS SDKs (APILow-Level)

Hinweise zu SSE -C finden Sie unter. Verwenden Sie serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem AWS SDK for. NETim AWS SDKfür. NETLeitfaden für Entwickler.

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.IO;
using System.Security.Cryptography;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class SSEClientEncryptionKeyObjectOperationsTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string keyName = "*** key name for new object created ***";
        private const string copyTargetKeyName = "*** key name for object copy ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            ObjectOpsUsingClientEncryptionKeyAsync().Wait();
```

```
        }
        private static async Task ObjectOpsUsingClientEncryptionKeyAsync()
        {
            try
            {
                // Create an encryption key.
                Aes aesEncryption = Aes.Create();
                aesEncryption.KeySize = 256;
                aesEncryption.GenerateKey();
                string base64Key = Convert.ToBase64String(aesEncryption.Key);

                // 1. Upload the object.
                PutObjectRequest putObjectRequest = await
UploadObjectAsync(base64Key);
                // 2. Download the object and verify that its contents matches what
you uploaded.
                await DownloadObjectAsync(base64Key, putObjectRequest);
                // 3. Get object metadata and verify that the object uses AES-256
encryption.
                await GetObjectMetadataAsync(base64Key);
                // 4. Copy both the source and target objects using server-side
encryption with
                //    a customer-provided encryption key.
                await CopyObjectAsync(aesEncryption, base64Key);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered ***. Message:'{0}' when writing
an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }

        private static async Task<PutObjectRequest> UploadObjectAsync(string
base64Key)
        {
            PutObjectRequest putObjectRequest = new PutObjectRequest
            {
                BucketName = bucketName,
                Key = keyName,
```

```
                ContentBody = "sample text",
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key
            };
            PutObjectResponse putObjectResponse = await
client.PutObjectAsync(putObjectRequest);
            return putObjectRequest;
        }
        private static async Task DownloadObjectAsync(string base64Key,
PutObjectRequest putObjectRequest)
        {
            GetObjectRequest getObjectRequest = new GetObjectRequest
            {
                BucketName = bucketName,
                Key = keyName,
                // Provide encryption information for the object stored in Amazon
S3.
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key
            };

            using (GetObjectResponse getResponse = await
client.GetObjectAsync(getObjectRequest))
            using (StreamReader reader = new
StreamReader(getResponse.ResponseStream))
            {
                string content = reader.ReadToEnd();
                if (String.Compare(putObjectRequest.ContentBody, content) == 0)
                    Console.WriteLine("Object content is same as we uploaded");
                else
                    Console.WriteLine("Error...Object content is not same.");

                if (getResponse.ServerSideEncryptionCustomerMethod ==
ServerSideEncryptionCustomerMethod.AES256)
                    Console.WriteLine("Object encryption method is AES256, same as
we set");
                else
                    Console.WriteLine("Error...Object encryption method is not the
same as AES256 we set");

                // Assert.AreEqual(putObjectRequest.ContentBody, content);
```

```
                // Assert.AreEqual(ServerSideEncryptionCustomerMethod.AES256,
getResponse.ServerSideEncryptionCustomerMethod);
            }
        }
        private static async Task GetObjectMetadataAsync(string base64Key)
        {
            GetObjectMetadataRequest getObjectMetadataRequest = new
GetObjectMetadataRequest
            {
                BucketName = bucketName,
                Key = keyName,

                // The object stored in Amazon S3 is encrypted, so provide the
necessary encryption information.
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key
            };

            GetObjectMetadataResponse getObjectMetadataResponse = await
client.GetObjectMetadataAsync(getObjectMetadataRequest);
            Console.WriteLine("The object metadata show encryption method used is:
{0}", getObjectMetadataResponse.ServerSideEncryptionCustomerMethod);
            // Assert.AreEqual(ServerSideEncryptionCustomerMethod.AES256,
getObjectMetadataResponse.ServerSideEncryptionCustomerMethod);
        }
        private static async Task CopyObjectAsync(Aes aesEncryption, string
base64Key)
        {
            aesEncryption.GenerateKey();
            string copyBase64Key = Convert.ToBase64String(aesEncryption.Key);

            CopyObjectRequest copyRequest = new CopyObjectRequest
            {
                SourceBucket = bucketName,
                SourceKey = keyName,
                DestinationBucket = bucketName,
                DestinationKey = copyTargetKeyName,
                // Information about the source object's encryption.
                CopySourceServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                CopySourceServerSideEncryptionCustomerProvidedKey = base64Key,
                // Information about the target object's encryption.
```

```
                ServerSideEncryptionCustomerMethod =
 ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = copyBase64Key
            };
            await client.CopyObjectAsync(copyRequest);
        }
    }
}
```

Verwenden von AWS SDKs, um SSE -C für mehrteilige Uploads anzugeben

Das Beispiel im vorherigen Abschnitt zeigt, wie Sie bei den Vorgängen,, Head und Copy eine serverseitige Verschlüsselung mit einem vom Kunden bereitgestellten Schlüssel (SSE-C) anfordern. PUT GET In diesem Abschnitt werden andere Amazon S3 beschriebenAPIs, die SSE -C unterstützen.

Java

Um große Objekte hochzuladen, können Sie den mehrteiligen Upload verwenden API (siehe[Hochladen und Kopieren von Objekten mit mehrteiligen Uploads](#)). APIsUm große Objekte hochzuladen, können Sie entweder High-Level oder Low-Level verwenden. Diese APIs unterstützen verschlüsselungsbezogene Header in der Anfrage.

- Wenn Sie den High-Level verwenden `TransferManager`API, geben Sie die verschlüsselungsspezifischen Header in der an (siehe). `PutObjectRequest` [Hochladen eines Objekts mit Multipart-Upload](#)
- Wenn Sie den Low-Level-Modus verwendenAPI, geben Sie verschlüsselungsbezogene Informationen in den ein`InitiateMultipartUploadRequest`, gefolgt von identischen Verschlüsselungsinformationen in jedem. `UploadPartRequest` In Ihrer `CompleteMultipartUploadRequest` müssen Sie keine verschlüsselungsspezifischen Header angeben. Beispiele finden Sie unter [Verwenden von AWS SDKs (APILow-Level)](#).

Das folgende Beispiel dient der Erstellung von Objekten und zeigt`TransferManager`, wie Informationen im Zusammenhang mit SSE -C bereitgestellt werden. Das Beispiel erledigt Folgendes:

- Erstellt ein Objekts mit der Methode `TransferManager.upload()`. In der Instance `PutObjectRequest` geben Sie anzufordernde Verschlüsselungsschlüssel-Informationen an. Amazon S3 verschlüsselt das Objekt mit dem vom Kunden bereitgestellten Schüssel.
- Erstellt eine Kopie des Objekts durch Aufrufen der `TransferManager.copy()`-Methode. Das Beispiel weist Amazon S3 dazu an, eine Objektkopie unter Verwendung eines neuen `SSECustomerKey` zu verschlüsseln. Da das Quellobjekt mit SSE -C verschlüsselt ist, stellt der `CopyObjectRequest` auch den Verschlüsselungsschlüssel des Quellobjekts bereit, sodass Amazon S3 das Objekt vor dem Kopieren entschlüsseln kann.

Example

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CopyObjectRequest;
import com.amazonaws.services.s3.model.PutObjectRequest;
import com.amazonaws.services.s3.model.SSECustomerKey;
import com.amazonaws.services.s3.transfer.Copy;
import com.amazonaws.services.s3.transfer.TransferManager;
import com.amazonaws.services.s3.transfer.TransferManagerBuilder;
import com.amazonaws.services.s3.transfer.Upload;

import javax.crypto.KeyGenerator;
import java.io.File;
import java.security.SecureRandom;

public class ServerSideEncryptionCopyObjectUsingHLwithSSEC {

    public static void main(String[] args) throws Exception {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";
        String fileToUpload = "*** File path ***";
        String keyName = "*** New object key name ***";
        String targetKeyName = "*** Key name for object copy ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
```

```
                    .withRegion(clientRegion)
                    .withCredentials(new ProfileCredentialsProvider())
                    .build();
            TransferManager tm = TransferManagerBuilder.standard()
                    .withS3Client(s3Client)
                    .build();

            // Create an object from a file.
            PutObjectRequest putObjectRequest = new PutObjectRequest(bucketName,
keyName, new File(fileToUpload));

            // Create an encryption key.
            KeyGenerator keyGenerator = KeyGenerator.getInstance("AES");
            keyGenerator.init(256, new SecureRandom());
            SSECustomerKey sseCustomerEncryptionKey = new
SSECustomerKey(keyGenerator.generateKey());

            // Upload the object. TransferManager uploads asynchronously, so this
call
            // returns immediately.
            putObjectRequest.setSSECustomerKey(sseCustomerEncryptionKey);
            Upload upload = tm.upload(putObjectRequest);

            // Optionally, wait for the upload to finish before continuing.
            upload.waitForCompletion();
            System.out.println("Object created.");

            // Copy the object and store the copy using SSE-C with a new key.
            CopyObjectRequest copyObjectRequest = new CopyObjectRequest(bucketName,
keyName, bucketName, targetKeyName);
            SSECustomerKey sseTargetObjectEncryptionKey = new
SSECustomerKey(keyGenerator.generateKey());
            copyObjectRequest.setSourceSSECustomerKey(sseCustomerEncryptionKey);

copyObjectRequest.setDestinationSSECustomerKey(sseTargetObjectEncryptionKey);

            // Copy the object. TransferManager copies asynchronously, so this call
returns
            // immediately.
            Copy copy = tm.copy(copyObjectRequest);

            // Optionally, wait for the upload to finish before continuing.
            copy.waitForCompletion();
            System.out.println("Copy complete.");
```

```
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Um große Objekte hochzuladen, können Sie den mehrteiligen Upload verwenden API (siehe). Hochladen und Kopieren von Objekten mit mehrteiligen Uploads AWS SDKfür. NETermöglicht das Hochladen großer Objekte sowohl auf hoher als auch APIs auf niedriger Ebene. Diese APIs unterstützen verschlüsselungsbezogene Header in der Anfrage.

- Wenn Sie High-Level verwenden `Transfer-Utility` API, geben Sie die verschlüsselungsspezifischen Header in der Abbildung an. `TransferUtilityUploadRequest` Codebeispiele finden Sie unter Hochladen eines Objekts mit Multipart-Upload.

```
TransferUtilityUploadRequest request = new TransferUtilityUploadRequest()
{
    FilePath = filePath,
    BucketName = existingBucketName,
    Key = keyName,
    // Provide encryption information.
    ServerSideEncryptionCustomerMethod =
 ServerSideEncryptionCustomerMethod.AES256,
    ServerSideEncryptionCustomerProvidedKey = base64Key,
};
```

- Bei Verwendung der Low-Level-Methode API geben Sie Informationen zur Verschlüsselung in der Anfrage zur Initiierung eines mehrteiligen Uploads ein, gefolgt von identischen Verschlüsselungsinformationen in den nachfolgenden Upload-Teilanforderungen. In der vollständigen Anforderung zum mehrteiligen Upload müssen Sie keine

verschlüsselungsspezifischen Header angeben. Beispiele finden Sie unter Verwenden von AWS SDKs (APILow-Level).

Nachfolgend finden Sie ein Beispiel für einen mehrteiligen Low-Level-Upload, der eine Kopie eines vorhandenen großen Objekts erstellt. In dem Beispiel wird das zu kopierende Objekt mit SSE -C in Amazon S3 gespeichert, und Sie möchten das Zielobjekt auch mit SSE -C speichern. In dem Beispiel gehen Sie wie folgt vor:

- Initiieren Sie eine mehrteilige Upload-Anfrage, indem Sie einen Verschlüsselungsschlüssel und zugehörige Informationen bereitstellen.
- Stellen Sie Verschlüsselungsschlüssel für das Quell- und das Zielobjekt sowie die zugehörigen Informationen in der `CopyPartRequest` bereit.
- Ermitteln Sie die Größe des zu kopierenden Quellobjekts, indem Sie die Objekt-Metadaten abrufen.
- Laden Sie die Objekte in Teilen mit je 5 MB hoch.

Example

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.IO;
using System.Security.Cryptography;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class SSECLowLevelMPUcopyObjectTest
    {
        private const string existingBucketName = "*** bucket name ***";
        private const string sourceKeyName      = "*** source object key name
 ***";
        private const string targetKeyName      = "*** key name for the target
 object ***";
        private const string filePath           = @"*** file path ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
```

```
        static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            CopyObjClientEncryptionKeyAsync().Wait();
        }

        private static async Task CopyObjClientEncryptionKeyAsync()
        {
            Aes aesEncryption = Aes.Create();
            aesEncryption.KeySize = 256;
            aesEncryption.GenerateKey();
            string base64Key = Convert.ToBase64String(aesEncryption.Key);

            await CreateSampleObjUsingClientEncryptionKeyAsync(base64Key,
s3Client);

            await CopyObjectAsync(s3Client, base64Key);
        }
        private static async Task CopyObjectAsync(IAmazonS3 s3Client, string
base64Key)
        {
            List<CopyPartResponse> uploadResponses = new List<CopyPartResponse>();

            // 1. Initialize.
            InitiateMultipartUploadRequest initiateRequest = new
InitiateMultipartUploadRequest
            {
                BucketName = existingBucketName,
                Key = targetKeyName,
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key,
            };

            InitiateMultipartUploadResponse initResponse =
                await s3Client.InitiateMultipartUploadAsync(initiateRequest);

            // 2. Upload Parts.
            long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB
            long firstByte = 0;
            long lastByte = partSize;

            try
            {
```

```
                // First find source object size. Because object is stored
encrypted with
                // customer provided key you need to provide encryption
information in your request.
                GetObjectMetadataRequest getObjectMetadataRequest = new
GetObjectMetadataRequest()
                {
                    BucketName = existingBucketName,
                    Key = sourceKeyName,
                    ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                    ServerSideEncryptionCustomerProvidedKey = base64Key // " *
**source object encryption key ***"
                };

                GetObjectMetadataResponse getObjectMetadataResponse = await
s3Client.GetObjectMetadataAsync(getObjectMetadataRequest);

                long filePosition = 0;
                for (int i = 1; filePosition <
getObjectMetadataResponse.ContentLength; i++)
                {
                    CopyPartRequest copyPartRequest = new CopyPartRequest
                    {
                        UploadId = initResponse.UploadId,
                        // Source.
                        SourceBucket = existingBucketName,
                        SourceKey = sourceKeyName,
                        // Source object is stored using SSE-C. Provide encryption
information.
                        CopySourceServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                        CopySourceServerSideEncryptionCustomerProvidedKey =
base64Key, //"***source object encryption key ***",
                        FirstByte = firstByte,
                        // If the last part is smaller then our normal part size
then use the remaining size.
                        LastByte = lastByte >
getObjectMetadataResponse.ContentLength ?
                            getObjectMetadataResponse.ContentLength - 1 :
lastByte,

                        // Target.
                        DestinationBucket = existingBucketName,
```

```
                        DestinationKey = targetKeyName,
                        PartNumber = i,
                        // Encryption information for the target object.
                        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                        ServerSideEncryptionCustomerProvidedKey = base64Key
                    };
                    uploadResponses.Add(await
s3Client.CopyPartAsync(copyPartRequest));
                    filePosition += partSize;
                    firstByte += partSize;
                    lastByte += partSize;
                }

                // Step 3: complete.
                CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest
                {
                    BucketName = existingBucketName,
                    Key = targetKeyName,
                    UploadId = initResponse.UploadId,
                };
                completeRequest.AddPartETags(uploadResponses);

                CompleteMultipartUploadResponse completeUploadResponse =
                    await s3Client.CompleteMultipartUploadAsync(completeRequest);
            }
            catch (Exception exception)
            {
                Console.WriteLine("Exception occurred: {0}", exception.Message);
                AbortMultipartUploadRequest abortMPURequest = new
AbortMultipartUploadRequest
                {
                    BucketName = existingBucketName,
                    Key = targetKeyName,
                    UploadId = initResponse.UploadId
                };
                s3Client.AbortMultipartUpload(abortMPURequest);
            }
        }
        private static async Task
CreateSampleObjUsingClientEncryptionKeyAsync(string base64Key, IAmazonS3
s3Client)
        {
```

```
            // List to store upload part responses.
            List<UploadPartResponse> uploadResponses = new
List<UploadPartResponse>();

            // 1. Initialize.
            InitiateMultipartUploadRequest initiateRequest = new
InitiateMultipartUploadRequest
            {
                BucketName = existingBucketName,
                Key = sourceKeyName,
                ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                ServerSideEncryptionCustomerProvidedKey = base64Key
            };

            InitiateMultipartUploadResponse initResponse =
               await s3Client.InitiateMultipartUploadAsync(initiateRequest);

            // 2. Upload Parts.
            long contentLength = new FileInfo(filePath).Length;
            long partSize = 5 * (long)Math.Pow(2, 20); // 5 MB

            try
            {
                long filePosition = 0;
                for (int i = 1; filePosition < contentLength; i++)
                {
                    UploadPartRequest uploadRequest = new UploadPartRequest
                    {
                        BucketName = existingBucketName,
                        Key = sourceKeyName,
                        UploadId = initResponse.UploadId,
                        PartNumber = i,
                        PartSize = partSize,
                        FilePosition = filePosition,
                        FilePath = filePath,
                        ServerSideEncryptionCustomerMethod =
ServerSideEncryptionCustomerMethod.AES256,
                        ServerSideEncryptionCustomerProvidedKey = base64Key
                    };

                    // Upload part and add response to our list.
                    uploadResponses.Add(await
s3Client.UploadPartAsync(uploadRequest));
```

```
                filePosition += partSize;
            }

            // Step 3: complete.
            CompleteMultipartUploadRequest completeRequest = new
 CompleteMultipartUploadRequest
            {
                BucketName = existingBucketName,
                Key = sourceKeyName,
                UploadId = initResponse.UploadId,
                //PartETags = new List<PartETag>(uploadResponses)

            };
            completeRequest.AddPartETags(uploadResponses);

            CompleteMultipartUploadResponse completeUploadResponse =
                await s3Client.CompleteMultipartUploadAsync(completeRequest);

        }
        catch (Exception exception)
        {
            Console.WriteLine("Exception occurred: {0}", exception.Message);
            AbortMultipartUploadRequest abortMPURequest = new
 AbortMultipartUploadRequest
            {
                BucketName = existingBucketName,
                Key = sourceKeyName,
                UploadId = initResponse.UploadId
            };
            await s3Client.AbortMultipartUploadAsync(abortMPURequest);
        }
    }
  }
}
```

Schützen von Daten mithilfe der clientseitigen Verschlüsselung

Clientseitige Verschlüsselung bezeichnet die lokale Verschlüsselung Ihrer Daten, um ihre Sicherheit bei der Übertragung und im Ruhezustand zu gewährleisten. Verwenden Sie den Amazon S3 Encryption Client, um Ihre Objekte zu verschlüsseln, bevor Sie sie an Amazon S3 senden. Wenn Ihre Objekte auf diese Weise verschlüsselt werden, werden Ihre Objekte keinen Dritten zugänglich

gemacht, auch AWS nicht. Amazon S3 empfängt Ihre Objekte bereits verschlüsselt. Amazon S3 ist an der Verschlüsselung oder Entschlüsselung Ihrer Objekte nicht beteiligt. Sie können sowohl den Amazon S3 Encryption Client als auch die serverseitige Verschlüsselung verwenden, um Ihre Daten zu verschlüsseln. Wenn Sie verschlüsselte Objekte an Amazon S3 senden, erkennt Amazon S3 die Objekte nicht als verschlüsselt, sondern nur typische Objekte.

Der Amazon S3 Encryption Client fungiert als Vermittler zwischen Ihnen und Amazon S3. Nachdem Sie den Amazon S3 Encryption Client instanziiert haben, werden Ihre Objekte als Teil Ihrer `PutObject`- und `GetObject`-Anforderungen in Amazon S3 automatisch verschlüsselt. Ihre Objekte werden alle mit einem eindeutigen Datenschlüssel verschlüsselt. Der Amazon S3 Encryption Client verwendet oder interagiert nicht mit Bucket-Schlüsseln, auch wenn Sie einen KMS Schlüssel als Ihren Wrapping-Schlüssel angeben.

Das Entwicklerhandbuch zum Amazon S3 Encryption Client konzentriert sich auf die Versionen 3.0 und höher des Amazon S3 Encryption Client. Weitere Informationen finden Sie unter Was ist der Amazon S3 Encryption Client? im Entwicklerhandbuch zum Amazon S3 Encryption Client.

Weitere Informationen zu früheren Versionen des Amazon S3 Encryption Clients finden Sie im AWS SDK Entwicklerhandbuch für Ihre Programmiersprache.

- AWS SDK for Java
- AWS SDK for .NET
- AWS SDK for Go
- AWS SDK for PHP
- AWS SDK for Ruby
- AWS SDK for C++

Richtlinie für den Datenverkehr zwischen Netzwerken

In diesem Thema wird beschrieben, wie Amazon S3 Verbindungen vom Service zu anderen Speicherorten sichert.

Datenverkehr zwischen Service und lokalen Clients und Anwendungen

Die folgenden Verbindungen können kombiniert werden AWS PrivateLink , um Konnektivität zwischen Ihrem privaten Netzwerk und bereitzustellen AWS:

- Eine AWS Site-to-Site VPN Verbindung. Weitere Informationen finden Sie unter Was ist AWS Site-to-Site VPN?
- Eine AWS Direct Connect Verbindung. Weitere Informationen finden Sie unter Was ist AWS Direct Connect?

Der Zugriff auf Amazon S3 über das Netzwerk erfolgt über AWS PublishedAPIs. Kunden müssen Transport Layer Security (TLS) 1.2 unterstützen. Wir empfehlen TLS 1.3. Kunden müssen auch Cipher Suites mit Perfect Forward Secrecy (PFS) unterstützen, wie Ephemeral Diffie-Hellman () oder Elliptic Curve Diffie-Hellman Ephemeral (DHE). ECDHE Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi. Darüber hinaus müssen Sie Anfragen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signieren, die einem IAM Prinzipal zugeordnet sind, oder Sie können das () verwenden, um temporäre Sicherheitsanmeldeinformationen zum Signieren von Anfragen zu generieren.AWS Security Token Service STS

Verkehr zwischen AWS Ressourcen in derselben Region

Ein Virtual Private Cloud (VPC) -Endpunkt für Amazon S3 ist eine logische Einheit innerhalb einerVPC, die nur Konnektivität zu Amazon S3 ermöglicht. Der VPC leitet Anfragen an Amazon S3 weiter und leitet Antworten zurück an denVPC. Weitere Informationen finden Sie unter VPCEndpoints im VPCBenutzerhandbuch. Beispiele für Bucket-Richtlinien, mit denen Sie den Zugriff auf S3-Buckets von VPC Endpunkten aus steuern können, finden Sie unter. Steuern des Zugriffs von VPC Endpunkten aus mit Bucket-Richtlinien

AWS PrivateLink für Amazon S3

Mit AWS PrivateLink for Amazon S3 können Sie VPCSchnittstellenendpunkte (Schnittstellenendpunkte) in Ihrer virtuellen privaten Cloud (VPC) bereitstellen. Auf diese Endpunkte kann direkt von Anwendungen aus zugegriffen werden, die sich vor Ort über VPN und AWS Direct Connect oder in einem anderen AWS-Region Over-Peering befinden. VPC

Schnittstellenendpunkte werden durch eine oder mehrere Elastic Network-Schnittstellen (ENIs) repräsentiert, denen private IP-Adressen aus Subnetzen in Ihrem zugewiesen werden. VPC Anforderungen an Amazon S3 über Schnittstellenendpunkte bleiben im Amazon-Netzwerk. Sie können auch über oder () auf Schnittstellenendpunkte in Ihren VPC lokalen Anwendungen zugreifen AWS Direct Connect . AWS Virtual Private Network AWS VPNWeitere Informationen dazu, wie Sie Ihr Netzwerk VPC mit Ihrem lokalen Netzwerk verbinden, finden Sie im AWS Direct Connect Benutzerhandbuch und im AWS Site-to-Site VPN Benutzerhandbuch.

Allgemeine Informationen zu Schnittstellenendpunkten finden Sie im Handbuch unter VPCSchnittstellenendpunkte (AWS PrivateLink).AWS PrivateLink

Themen

- Arten von VPC Endpunkten für Amazon S3
- Einschränkungen und Einschränkungen von AWS PrivateLink für Amazon S3
- Einen VPC Endpunkt erstellen
- Zugriff auf Amazon-S3-Schnittstellen-Endpunkte
- Privat DNS
- Zugriff auf Buckets, Access Points und Amazon S3 API Control-Operationen von Endpunkten mit S3-Schnittstelle
- Aktualisierung einer lokalen Konfiguration DNS
- Erstellen einer VPC Endpunktrichtlinie für Amazon S3

Arten von VPC Endpunkten für Amazon S3

Sie können zwei Arten von VPC Endpunkten für den Zugriff auf Amazon S3 verwenden: Gateway-Endpunkte und Schnittstellenendpunkte (mit). AWS PrivateLink Ein Gateway-Endpunkt ist ein Gateway, das Sie in Ihrer Routing-Tabelle angeben, um VPC über das AWS Netzwerk auf Amazon S3 zuzugreifen. Schnittstellen-Endpunkte erweitern die Funktionalität von Gateway-Endpunkten, indem sie private IP-Adressen verwenden, um Anfragen von innerhalb Ihres UnternehmensVPC, vor Ort oder von einem VPC anderen Standort aus mithilfe AWS-Region von VPC Peering oder an Amazon S3 weiterzuleiten. AWS Transit Gateway Weitere Informationen finden Sie unter Was ist Peering? VPC und Transit Gateway im Vergleich zu VPC Peering.

Schnittstellenendpunkte sind mit Gateway-Endpunkten kompatibel. Wenn Sie bereits einen Gateway-Endpunkt in der habenVPC, können Sie beide Arten von Endpunkten gleichzeitig verwenden. VPC

Gateway-Endpunkte für Amazon S3	Schnittstellenendpunkte für Amazon S3
In beiden Fällen verbleibt Ihr Netzwerkverkehr im AWS Netzwerk.	
Verwenden Sie öffentliche IP-Adressen von Amazon S3	Verwenden Sie private IP-Adressen von IhremVPC, um auf Amazon S3 zuzugreifen

Gateway-Endpunkte für Amazon S3	Schnittstellenendpunkte für Amazon S3
Verwenden Sie dieselben Amazon S3 DNS S3-Namen	Endpunktspezifische Amazon S3 S3-Namen erforderlich DNS
Erlauben keinen On-Premises-Zugriff	Erlaubt On-Premises-Zugriff
Erlauben Sie keinen Zugriff von einem anderen AWS-Region	Erlauben Sie den Zugriff von einem VPC in einem anderen, AWS-Region indem Sie VPC Peering verwenden oder AWS Transit Gateway
Nicht berechnet	Berechnet

Weitere Informationen zu Gateway-Endpunkten finden Sie im Handbuch unter VPCGateway-Endpunkte.AWS PrivateLink

Einschränkungen und Einschränkungen von AWS PrivateLink für Amazon S3

VPCEinschränkungen gelten AWS PrivateLink für Amazon S3. Weitere Informationen finden Sie unter Überlegungen zu Schnittstellenendpunkten und AWS PrivateLink -Kontingente im Handbuch zu AWS PrivateLink . Darüber hinaus gelten die folgenden Einschränkungen.

AWS PrivateLink für Amazon S3 unterstützt Folgendes nicht:

- Endpunkte nach dem Federal Information Processing Standard (FIPS)
- Website-Endpunkte
- Globale Legacy-Endpunkte
- Endpunkte der S3-Dash-Region
- Dual-Stack-Endpunkte
- Verwenden von CopyObject oder UploadPartCopyzwischen Buckets in verschiedenen AWS-Regionen
- Sicherheit auf Transportschicht (TLS) 1.1

Einen VPC Endpunkt erstellen

Informationen zum Erstellen eines VPC Schnittstellenendpunkts finden Sie unter Erstellen eines VPC Endpunkts im AWS PrivateLink Handbuch.

Zugriff auf Amazon-S3-Schnittstellen-Endpunkte

Wenn Sie einen Schnittstellenendpunkt erstellen, generiert Amazon S3 zwei Arten von endpunktspezifischen DNS S3-Namen: Regional und Zonal.

- Ein regionaler DNS Name beinhaltet eine eindeutige VPC Endpunkt-ID, eine Service-ID AWS-Region, das und `vpce.amazonaws.com` in seinem Namen. Für die VPC Endpunkt-ID *`vpce-1a2b3c4d`* könnte der generierte DNS Name beispielsweise ähnlich sein wie*`vpce-1a2b3c4d-5e6f`*`.s3.us-east-1.vpce.amazonaws.com`.
- Ein zonaler DNS Name beinhaltet die Availability Zone — zum Beispiel. *`vpce-1a2b3c4d-5e6f`***`-us-east-1a.`**`s3.us-east-1.vpce.amazonaws.com` Sie können diese Option verwenden, wenn Ihre Architektur Availability Zones isoliert. Sie könnten sie beispielsweise zur Fehlereingrenzung oder zur Senkung der regionalen Datenübertragungskosten verwenden.

Endpunktspezifische DNS S3-Namen können von der öffentlichen S3-Domäne aus aufgelöst werden. DNS

Privat DNS

Private DNS Optionen für VPC Schnittstellenendpunkte vereinfachen die Weiterleitung von S3-Verkehr über VPC Endpunkte und helfen Ihnen, den kostengünstigsten Netzwerkpfad zu nutzen, der für Ihre Anwendung verfügbar ist. Sie können private DNS Optionen verwenden, um regionalen S3-Verkehr weiterzuleiten, ohne Ihre S3-Clients so zu aktualisieren, dass sie die endpunktspezifischen DNS Namen Ihrer Schnittstellen-Endpunkte verwenden oder die Infrastruktur verwalten müssen. DNS Wenn private DNS Namen aktiviert sind, werden regionale DNS S3-Abfragen zu den privaten IP-Adressen der AWS PrivateLink folgenden Endpunkte aufgelöst:

- Regionale Bucket-Endpunkte (z. B. `s3.us-east-1.amazonaws.com`)
- Kontrollendpunkte (z. B. `s3-control.us-east-1.amazonaws.com`)
- Zugriffspunkt-Endpunkte (z. B. `s3-accesspoint.us-east-1.amazonaws.com`)

AWS PrivateLink für Amazon S3 unterstützt nicht die Verwendung von Amazon S3 S3-Dual-Stack-Endpunkten. Wenn Sie einen DNS S3-Dual-Stack-Namen als privaten DNS Namen verwenden, wird Ihr IPv6 Datenverkehr entweder gelöscht oder, falls Ihre virtuelle private Cloud (VPC) über ein Internet-Gateway verfügt, wird Ihr IPv6 Datenverkehr über das Internet-Gateway in Ihrem weitergeleitet. VPC

Wenn Sie einen Gateway-Endpunkt in Ihrem habenVPC, können Sie eingehende VPC Anfragen automatisch über Ihren vorhandenen S3-Gateway-Endpunkt und lokale Anfragen über Ihren Schnittstellenendpunkt weiterleiten. Mit diesem Ansatz können Sie Ihre Netzwerkkosten optimieren, indem Sie Gateway-Endpunkte verwenden, für die keine Gebühren anfallen. VPC Ihre lokalen Anwendungen können AWS PrivateLink mithilfe des Resolver-Endpunkts für eingehende Anrufe verwendet werden. Amazon stellt Ihnen einen DNS Server namens Route 53 Resolver zur VerfügungVPC. Ein eingehender Resolver-Endpunkt leitet DNS Anfragen aus dem lokalen Netzwerk an den Route 53 Resolver weiter.

> **Important**
>
> Um bei der Verwendung von Enable Private DNS nur für eingehende Endpunkte den kostengünstigsten Netzwerkpfad nutzen zu können, muss in Ihrem System ein Gateway-Endpunkt vorhanden sein. VPC Das Vorhandensein eines Gateway-Endpunkts trägt dazu bei, dass eingehender VPC Datenverkehr immer über das AWS private Netzwerk geleitet wird, wenn die Option DNS Nur für eingehende Endgeräte aktivieren aktiviert ist. Sie müssen diesen Gateway-Endpunkt beibehalten, solange Sie die Option DNSNur für eingehende Endpunkte aktivieren aktiviert haben. Wenn Sie Ihren Gateway-Endpunkt löschen möchten, müssen Sie zunächst die Option Private Option DNS nur für eingehende Endpunkte aktivieren deaktivieren.
>
> Wenn Sie einen vorhandenen Schnittstellenendpunkt auf Enable Private DNS only for Inbound Endpoints aktualisieren möchten, vergewissern Sie sich zunächst, dass Sie VPC über einen S3-Gateway-Endpunkt verfügen. Weitere Informationen zu Gateway-Endpunkten und zur Verwaltung privater DNS Namen finden Sie unter VPCGateway-Endpunkte bzw. DNSNamen verwalten im Handbuch.AWS PrivateLink

Die Option DNSNur privat für eingehende Endpunkte aktivieren ist nur für Dienste verfügbar, die Gateway-Endpunkte unterstützen.

Weitere Informationen zum Erstellen eines VPC Endpunkts, der Enable Private DNS nur für eingehende Endpunkte verwendet, finden Sie im Handbuch unter Erstellen eines Schnittstellenendpunkts.AWS PrivateLink

Verwenden der Konsole VPC

In der Konsole haben Sie zwei Optionen: DNSName aktivieren und DNS Nur privat für eingehende Endpunkte aktivieren. DNSName aktivieren ist eine Option, die von unterstützt wird. AWS PrivateLink

Mit der Option DNSNamen aktivieren können Sie die private Konnektivität von Amazon zu Amazon S3 nutzen und gleichzeitig Anfragen an die standardmäßigen öffentlichen DNS Endpunktnamen stellen. Wenn diese Option aktiviert ist, können Kunden den für Ihre Anwendung kostengünstigsten Netzwerkpfad nutzen.

Wenn Sie private DNS Namen auf einem vorhandenen oder neuen VPC Schnittstellenendpunkt für Amazon S3 aktivieren, ist die Option Private DNS nur für eingehende Endpunkte aktivieren standardmäßig ausgewählt. Wenn diese Option ausgewählt ist, verwenden Ihre Anwendungen nur Schnittstellenendpunkte für Ihren On-Premises-Datenverkehr. Dieser eingehende VPC Datenverkehr verwendet automatisch die kostengünstigeren Gateway-Endpunkte. Alternativ können Sie die Option Private DNS nur für eingehende Endpunkte aktivieren deaktivieren, um alle S3-Anfragen über Ihren Schnittstellenendpunkt weiterzuleiten.

Verwenden Sie AWS CLI

Wenn Sie keinen Wert für `PrivateDnsOnlyForInboundResolverEndpoint` angeben, wird standardmäßig `true` verwendet. Bevor Sie Ihre VPC Einstellungen übernehmen, wird jedoch überprüft, ob in der ein Gateway-Endpunkt vorhanden istVPC. Wenn in der ein Gateway-Endpunkt vorhanden istVPC, ist der Anruf erfolgreich. Wenn nicht, wird die folgende Fehlermeldung angezeigt:

Um den Wert PrivateDnsOnlyForInboundResolverEndpoint auf true zu setzen, VPC *`vpce_id`* muss über einen Gateway-Endpunkt für den Dienst verfügen.

Für einen neuen VPC Interface-Endpunkt

Verwenden Sie die `dns-options` Attribute `private-dns-enabled` und, um Private DNS über die Befehlszeile zu aktivieren. Die Option `PrivateDnsOnlyForInboundResolverEndpoint` im Attribut `dns-options` muss auf `true` gesetzt sein. Ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws ec2 create-vpc-endpoint \
--region us-east-1 \
--service-name s3-service-name \
--vpc-id client-vpc-id \
--subnet-ids client-subnet-id \
--vpc-endpoint-type Interface  \
--private-dns-enabled  \
--ip-address-type ip-address-type \
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=true \
--security-group-ids client-sg-id
```

Für einen vorhandenen VPC Endpunkt

Wenn Sie private DNS für einen vorhandenen VPC Endpunkt verwenden möchten, verwenden Sie den folgenden Beispielbefehl und ersetzen Sie den *user input placeholders* durch Ihre eigenen Informationen.

```
aws ec2 modify-vpc-endpoint \
--region us-east-1 \
--vpc-endpoint-id client-vpc-id \
--private-dns-enabled \
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=false
```

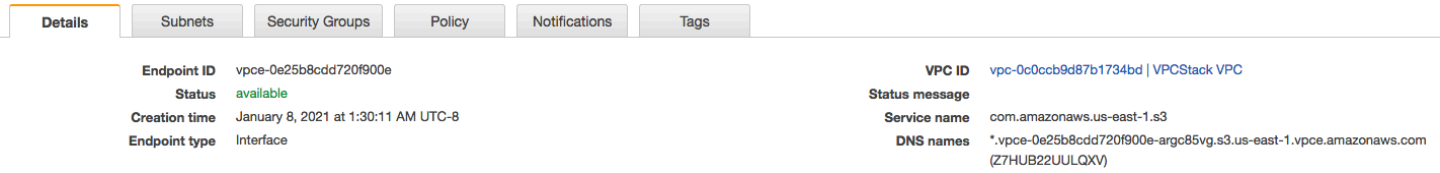
Wenn Sie einen vorhandenen VPC Endpunkt aktualisieren möchten, sodass Private DNS nur für den Inbound Resolver aktiviert wird, verwenden Sie das folgende Beispiel und ersetzen Sie die Beispielwerte durch Ihre eigenen.

```
aws ec2 modify-vpc-endpoint \
--region us-east-1 \
--vpc-endpoint-id client-vpc-id \
--private-dns-enabled \
--dns-options PrivateDnsOnlyForInboundResolverEndpoint=true
```

Zugriff auf Buckets, Access Points und Amazon S3 API Control-Operationen von Endpunkten mit S3-Schnittstelle

Sie können das AWS CLI oder verwenden, AWS SDKs um über S3-Schnittstellenendpunkte auf Buckets, S3-Zugriffspunkte und Amazon S3 API Control-Operationen zuzugreifen.

Die folgende Abbildung zeigt die Registerkarte „Details“ der VPC Konsole, auf der Sie den DNS Namen eines VPC Endpunkts finden. In diesem Beispiel lautet die VPCEndpunkt-ID (vpce-id) *vpce-0e25b8cdd720f900e* und der DNS Name lautet. `*.vpce-0e25b8cdd720f900e-argc85vg.s3.us-east-1.vpce.amazonaws.com`



Wenn Sie den DNS Namen für den Zugriff auf eine Ressource verwenden, ersetzen Sie * mit dem entsprechenden Wert. Die entsprechenden Werte, die anstelle von * verwendet werden können, lauten wie folgt:

- `bucket`
- `accesspoint`
- `control`

Um beispielsweise auf einen Bucket zuzugreifen, verwenden Sie einen DNSNamen wie diesen:

`bucket.vpce-0e25b8cdd720f900e-argc85vg.s3.us-east-1.vpce.amazonaws.com`

Beispiele für die Verwendung von DNS Namen für den Zugriff auf Buckets, Access Points und Amazon S3 API Control-Operationen finden Sie in den folgenden Abschnitten von AWS CLI Beispiele und AWS SDKBeispiele.

Weitere Informationen zum Anzeigen Ihrer endpunktspezifischen DNS Namen finden Sie unter Konfiguration der privaten DNS Namen von Endpoint Services anzeigen im VPC Benutzerhandbuch.

AWS CLI Beispiele

Verwenden Sie die Parameter und, um über S3-Schnittstellenendpunkte in AWS CLI Befehlen auf S3-Buckets, S3-Zugriffspunkte oder Amazon S3 API `--region` `--endpoint-url` Control-Operationen zuzugreifen.

Beispiel: Verwenden Sie einen EndpunktURL, um Objekte in Ihrem Bucket aufzulisten

Ersetzen Sie im folgenden Beispiel den Bucket-Namen*my-bucket*, die Region *us-east-1* und den DNS Namen der VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* durch Ihre eigenen Informationen.

```
aws s3 ls s3://my-bucket/ --region us-east-1 --endpoint-url
 https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com
```

Beispiel: Verwenden Sie einen EndpunktURL, um Objekte von einem Access Point aufzulisten

- Methode 1 — Verwendung des Amazon-Ressourcennamens (ARN) des Access Points mit dem Access Point-Endpunkt

  Ersetzen Sie die ARN*us-east-1:123456789012:accesspoint/accesspointexamplename*, die Region *us-east-1* und die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* durch Ihre eigenen Informationen.

  ```
  aws s3api list-objects-v2 --bucket arn:aws:s3:us-east-1:123456789012:accesspoint/accesspointexamplename --region us-east-1 --endpoint-url
   https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com
  ```

  Wenn Sie den Befehl nicht erfolgreich ausführen können, aktualisieren Sie Ihre Version AWS CLI auf die neueste Version und versuchen Sie es erneut. Weitere Informationen zur Aktualisierung finden Sie unter Installation oder Aktualisierung der aktuellen Version der AWS CLI im Benutzerhandbuch zu AWS Command Line Interface .

- Methode 2 – Verwenden des Alias des Zugriffspunkts mit dem regionalen Bucket-Endpunkt

  Ersetzen Sie im folgenden Beispiel den Access Point-Alias*accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias*, die Region *us-east-1* und die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* durch Ihre eigenen Informationen.

  ```
  aws s3api list-objects-v2 --
  bucket accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias
   --region us-east-1 --endpoint-url https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com
  ```

- Methode 3 – Verwenden des Alias des Zugriffspunkts mit dem Zugriffspunkt-Endpunkt

  Um einen S3-Endpunkt zu erstellen, bei dem der Bucket als Teil des Hostnamens enthalten ist, legen Sie zunächst den Adressierungsstil auf `virtual` fest, damit `aws s3api` ihn verwenden kann. Weitere Informationen zu `AWS configure` finden Sie unter Einstellungen der Konfigurations- und Anmeldeinformationsdatei im Benutzerhandbuch zu AWS Command Line Interface .

  ```
  aws configure set default.s3.addressing_style virtual
  ```

Ersetzen Sie dann im folgenden Beispiel den Access Point-Alias *accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias*, die Region *us-east-1* und die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* durch Ihre eigenen Informationen. Weitere Informationen zum Access Point-Alias finden Sie unter Verwenden eines Alias im Bucket-Stil für Ihren S3-Bucket-Zugriffspunkt.

```
aws s3api list-objects-v2 --
bucket accesspointexamplename-8tyekmigicmhun8n9kwpfur39dnw4use1a-s3alias --
region us-east-1 --endpoint-url https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com
```

Beispiel: Verwenden Sie einen EndpunktURL, um Jobs mit einem API S3-Steuerungsvorgang aufzulisten

Ersetzen Sie im folgenden Beispiel die Region *us-east-1*, die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* und die Konto-ID *12345678* durch Ihre eigenen Informationen.

```
aws s3control --region us-east-1 --endpoint-url
 https://control.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com list-jobs --
account-id 12345678
```

AWS SDKBeispiele

Um auf S3-Buckets, S3-Zugriffspunkte oder Amazon S3 API Control-Operationen über S3-Schnittstellenendpunkte zuzugreifen, wenn Sie die verwenden AWS SDKs, aktualisieren Sie Ihre Version SDKs auf die neueste Version. Konfigurieren Sie dann Ihre Clients so, dass sie einen Endpunkt URL für den Zugriff auf einen Bucket, Access Point oder Amazon S3 API Control-Operationen über S3-Schnittstellenendpunkte verwenden.

SDK for Python (Boto3)

Beispiel: Verwenden Sie einen EndpunktURL, um auf einen S3-Bucket zuzugreifen

Ersetzen Sie im folgenden Beispiel die Region *us-east-1* und die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* durch Ihre eigenen Informationen.

```
s3_client = session.client(
service_name='s3',
region_name='us-east-1',
endpoint_url='https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com'
)
```

Beispiel: Verwenden Sie einen EndpunktURL, um auf einen S3-Zugriffspunkt zuzugreifen

Ersetzen Sie im folgenden Beispiel die Region *us-east-1* und die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* durch Ihre eigenen Informationen.

```
ap_client = session.client(
service_name='s3',
region_name='us-east-1',
endpoint_url='https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com'
)
```

Beispiel: Verwenden Sie einen EndpunktURL, um auf Amazon S3 Control zuzugreifen API

Ersetzen Sie im folgenden Beispiel die Region *us-east-1* und die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* durch Ihre eigenen Informationen.

```
control_client = session.client(
service_name='s3control',
region_name='us-east-1',
endpoint_url='https://control.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com'
)
```

SDK for Java 1.x

Beispiel: Verwenden Sie einen EndpunktURL, um auf einen S3-Bucket zuzugreifen

Ersetzen Sie im folgenden Beispiel die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* durch Ihre eigenen Informationen.

```
// bucket client
final AmazonS3 s3 = AmazonS3ClientBuilder.standard().withEndpointConfiguration(
```

```
        new AwsClientBuilder.EndpointConfiguration(
                "https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com",
                Regions.DEFAULT_REGION.getName()
        )
).build();
List<Bucket> buckets = s3.listBuckets();
```

Beispiel: Verwenden Sie einen EndpunktURL, um auf einen S3-Zugriffspunkt zuzugreifen

Ersetzen Sie im folgenden Beispiel die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* und ARN *us-east-1:123456789012:accesspoint/prod* durch Ihre eigenen Informationen.

```
// accesspoint client
final AmazonS3 s3accesspoint =
 AmazonS3ClientBuilder.standard().withEndpointConfiguration(
        new AwsClientBuilder.EndpointConfiguration(
                "https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com",
                Regions.DEFAULT_REGION.getName()
        )
).build();
ObjectListing objects = s3accesspoint.listObjects("arn:aws:s3:us-
east-1:123456789012:accesspoint/prod");
```

Beispiel: Verwenden Sie einen EndpunktURL, um auf einen Amazon S3 API Control-Vorgang zuzugreifen

Ersetzen Sie im folgenden Beispiel die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* durch Ihre eigenen Informationen.

```
// control client
final AWSS3Control s3control =
 AWSS3ControlClient.builder().withEndpointConfiguration(
        new AwsClientBuilder.EndpointConfiguration(
                "https://control.vpce-1a2b3c4d-5e6f.s3.us-
east-1.vpce.amazonaws.com",
                Regions.DEFAULT_REGION.getName()
        )
).build();
final ListJobsResult jobs = s3control.listJobs(new ListJobsRequest());
```

SDK for Java 2.x

Beispiel: Verwenden Sie einen EndpunktURL, um auf einen S3-Bucket zuzugreifen

Ersetzen Sie im folgenden Beispiel die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* und die Region *Region.US_EAST_1* durch Ihre eigenen Informationen.

```
// bucket client
Region region = Region.US_EAST_1;
s3Client = S3Client.builder().region(region)

 .endpointOverride(URI.create("https://bucket.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com"))
                   .build()
```

Beispiel: Verwenden Sie einen EndpunktURL, um auf einen S3-Zugangspunkt zuzugreifen

Ersetzen Sie im folgenden Beispiel die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* und die Region *Region.US_EAST_1* durch Ihre eigenen Informationen.

```
// accesspoint client
Region region = Region.US_EAST_1;
s3Client = S3Client.builder().region(region)

 .endpointOverride(URI.create("https://accesspoint.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com"))
                   .build()
```

Beispiel: Verwenden Sie einen EndpunktURL, um auf Amazon S3 Control zuzugreifen API

Ersetzen Sie im folgenden Beispiel die VPC Endpunkt-ID *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* und die Region *Region.US_EAST_1* durch Ihre eigenen Informationen.

```
// control client
Region region = Region.US_EAST_1;
s3ControlClient = S3ControlClient.builder().region(region)

 .endpointOverride(URI.create("https://control.vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com"))
```

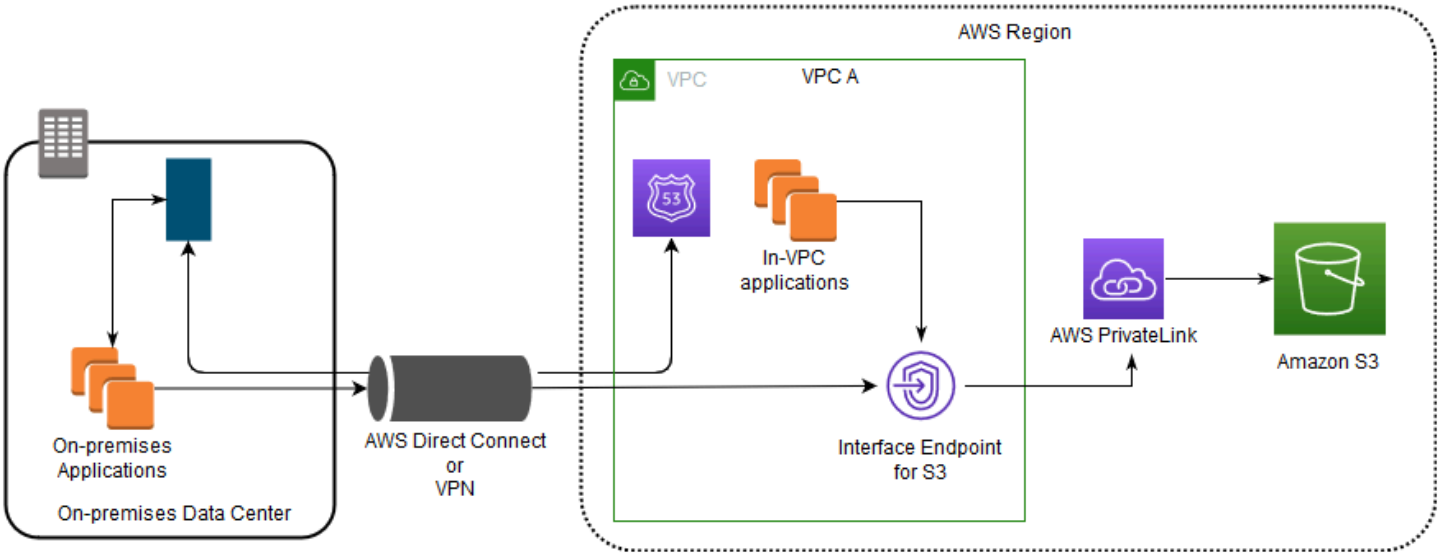
```
                .build()
```

Aktualisierung einer lokalen Konfiguration DNS

Wenn Sie endpunktspezifische DNS Namen für den Zugriff auf die Schnittstellenendpunkte für Amazon S3 verwenden, müssen Sie Ihren lokalen Resolver nicht aktualisieren. DNS Sie können den endpunktspezifischen DNS Namen mit der privaten IP-Adresse des Schnittstellenendpunkts aus der öffentlichen Amazon S3 DNS S3-Domain auflösen.

Verwenden von Schnittstellenendpunkten für den Zugriff auf Amazon S3 ohne einen Gateway-Endpunkt oder ein Internet-Gateway in der VPC

Schnittstellenendpunkte in Ihrem VPC System können sowohl interne als auch lokale VPC Anwendungen über das Amazon-Netzwerk an Amazon S3 weiterleiten, wie in der folgenden Abbildung dargestellt.



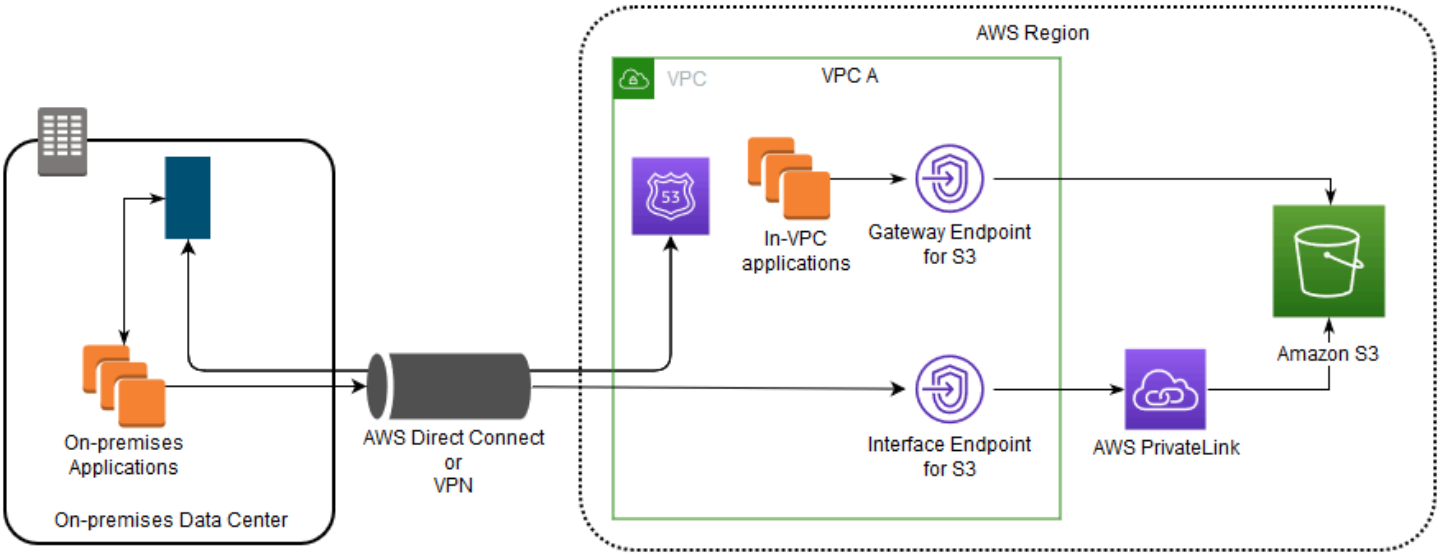
Das Diagramm veranschaulicht folgende Vorgänge:

- Ihr lokales Netzwerk verwendet AWS Direct Connect oder, um eine Verbindung AWS VPN zu A herzustellen. VPC
- Ihre Anwendungen vor Ort und in VPC A verwenden endpunktspezifische DNS Namen, um über den S3-Schnittstellenendpunkt auf Amazon S3 zuzugreifen.
- Lokale Anwendungen senden Daten VPC über AWS Direct Connect (oder) an den Schnittstellenendpunkt. AWS VPN AWS PrivateLink verschiebt die Daten vom Schnittstellenendpunkt über das AWS Netzwerk zu Amazon S3.

- Interne VPC Anwendungen senden auch Datenverkehr an den Schnittstellenendpunkt. AWS PrivateLink verschiebt die Daten vom Schnittstellenendpunkt über das AWS Netzwerk zu Amazon S3.

Gemeinsame Nutzung von Gateway-Endpunkten und Schnittstellen-Endpunkten für den VPC Zugriff auf Amazon S3

Sie können Schnittstellen-Endpunkte erstellen und den vorhandenen Gateway-Endpunkt darin beibehaltenVPC, wie das folgende Diagramm zeigt. Mit diesem Ansatz ermöglichen Sie internen VPC Anwendungen, weiterhin über den Gateway-Endpunkt auf Amazon S3 zuzugreifen, was nicht in Rechnung gestellt wird. Dann würden nur Ihre lokalen Anwendungen Schnittstellenendpunkte für den Zugriff auf Amazon S3 verwenden. Um auf diese Weise auf Amazon S3 zuzugreifen, müssen Sie Ihre lokalen Anwendungen so aktualisieren, dass sie endpunktspezifische DNS Namen für Amazon S3 verwenden.



Das Diagramm veranschaulicht folgende Vorgänge:

- Lokale Anwendungen verwenden endpunktspezifische DNS Namen, um Daten innerhalb des Durchgangs (oder) an den Schnittstellenendpunkt zu senden. VPC AWS Direct Connect AWS VPN AWS PrivateLink verschiebt die Daten vom Schnittstellenendpunkt über das AWS Netzwerk zu Amazon S3.
- Mithilfe der standardmäßigen regionalen Amazon S3-Namen senden VPC In-Anwendungen Daten an den Gateway-Endpunkt, der über das AWS Netzwerk eine Verbindung zu Amazon S3 herstellt.

Weitere Informationen zu Gateway-Endpunkten finden Sie unter VPCGateway-Endpunkte im VPCBenutzerhandbuch.

Erstellen einer VPC Endpunktrichtlinie für Amazon S3

Sie können Ihrem VPC Endpunkt eine Endpunktrichtlinie hinzufügen, die den Zugriff auf Amazon S3 steuert. Die Richtlinie gibt die folgenden Informationen an:

- Der AWS Identity and Access Management (IAM) Principal, der Aktionen ausführen kann
- Aktionen, die ausgeführt werden können
- Ressourcen, für die Aktionen ausgeführt werden können

Sie können auch Amazon S3 S3-Bucket-Richtlinien verwenden, um den Zugriff auf bestimmte Buckets von einem bestimmten VPC Endpunkt aus zu beschränken, indem Sie die `aws:sourceVpce` Bedingung in Ihrer Bucket-Richtlinie verwenden. Die folgenden Beispiele zeigen Richtlinien, die den Zugriff auf einen Bucket oder einen Endpunkt einschränken.

Themen

- Beispiel: Beschränken des Zugriffs auf einen bestimmten Bucket von einem Endpunkt aus VPC
- Beispiel: Beschränken des Zugriffs auf Buckets in einem bestimmten Konto von einem Endpunkt aus VPC
- Beispiel: Beschränkung des Zugriffs auf einen bestimmten VPC Endpunkt in der S3-Bucket-Richtlinie

Beispiel: Beschränken des Zugriffs auf einen bestimmten Bucket von einem Endpunkt aus VPC

Sie können eine Endpunktrichtlinie erstellen, die den Zugriff auf spezifische Amazon-S3-Buckets beschränkt. Diese Art von Richtlinie ist nützlich, wenn Sie andere AWS-Services Richtlinien habenVPC, die Buckets verwenden. Die folgende Bucket-Richtlinie schränkt den Zugriff auf ausschließlich *`amzn-s3-demo-bucket1`* ein. Um diese Endpunktrichtlinie zu verwenden, ersetzen Sie *`amzn-s3-demo-bucket1`* im Beispiel durch den Namen Ihres Buckets.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909151",
  "Statement": [
    { "Sid": "Access-to-specific-bucket-only",
      "Principal": "*",
```

```
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket1",
                   "arn:aws:s3:::amzn-s3-demo-bucket1/*"]
    }
  ]
}
```

Beispiel: Beschränken des Zugriffs auf Buckets in einem bestimmten Konto von einem Endpunkt aus VPC

Sie können eine Endpunktrichtlinie erstellen, die den Zugriff nur auf die S3-Buckets in einem bestimmten Bereich beschränkt. AWS-Konto Um zu verhindern, dass Ihre VPC Clients auf Buckets zugreifen, die Sie nicht besitzen, verwenden Sie die folgende Anweisung in Ihrer Endpunktrichtlinie. Die folgende Beispielanweisung erstellt eine Richtlinie, die den Zugriff auf Ressourcen einschränkt, die einer einzigen AWS-Konto ID gehören: *111122223333*.

```
{
  "Statement": [
    {
      "Sid": "Access-to-bucket-in-specific-account-only",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:s3:::*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": "111122223333"
        }
      }
    }
  ]
}
```

> **Note**
>
> Um die AWS-Konto ID der Ressource anzugeben, auf die zugegriffen wird, können Sie entweder den `aws:ResourceAccount` oder den `s3:ResourceAccount` Schlüssel in Ihrer IAM Richtlinie verwenden. Beachten Sie jedoch, dass einige auf den Zugriff auf AWS verwaltete Buckets AWS-Services angewiesen sind. Daher kann sich die Verwendung der `s3:ResourceAccount` Taste `aws:ResourceAccount` oder in Ihrer IAM Richtlinie auch auf den Zugriff auf diese Ressourcen auswirken.

Beispiel: Beschränkung des Zugriffs auf einen bestimmten VPC Endpunkt in der S3-Bucket-Richtlinie

Beispiel: Beschränkung des Zugriffs auf einen bestimmten VPC Endpunkt in der S3-Bucket-Richtlinie

Die folgende Amazon S3 S3-Bucket-Richtlinie ermöglicht den Zugriff auf einen bestimmten Bucket nur vom VPC Endpunkt aus*vpce-1a2b3c4d*. *amzn-s3-demo-bucket2* Die Richtlinie lehnt sämtlichen Zugriff auf den Bucket ab, der nicht über den angegebenen Endpunkt erfolgt. Die `aws:sourceVpce` Bedingung spezifiziert den Endpunkt und erfordert keinen Amazon-Ressourcennamen (ARN) für die VPC Endpunktressource, sondern nur die Endpunkt-ID. Um diese Bucket-Richtlinie zu verwenden, ersetzen Sie *amzn-s3-demo-bucket2* und *vpce-1a2b3c4d* durch den Namen Ihres Buckets und Ihren Endpunkt.

> **Important**
>
> - Wenn Sie die folgende Amazon S3 S3-Bucket-Richtlinie anwenden, um den Zugriff nur auf bestimmte VPC Endpunkte zu beschränken, blockieren Sie möglicherweise Ihren Zugriff auf den Bucket, ohne dies zu beabsichtigen. Bucket-Richtlinien, mit denen der Bucket-Zugriff speziell auf Verbindungen beschränkt werden soll, die von Ihrem VPC Endpunkt ausgehen, können alle Verbindungen zum Bucket blockieren. Informationen zur Behebung dieses Problems finden Sie unter Meine Bucket-Richtlinie hat die falsche VPC oder die VPC Endpunkt-ID. Wie kann ich die Richtlinie so ändern, dass ich auf den Bucket zugreifen kann? im AWS Support Knowledge Center.
> - Bevor Sie die folgende Beispielrichtlinie verwenden, ersetzen Sie die VPC Endpunkt-ID durch einen Wert, der Ihrem Anwendungsfall entspricht. Andernfalls können Sie nicht auf Ihren Bucket zugreifen.

- Diese Richtlinie deaktiviert den Konsolenzugriff auf den angegebenen Bucket, da Konsolenanfragen nicht vom angegebenen VPC Endpunkt stammen.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909152",
  "Statement": [
    { "Sid": "Access-to-specific-VPCE-only",
      "Principal": "*",
      "Action": "s3:*",
      "Effect": "Deny",
      "Resource": ["arn:aws:s3:::amzn-s3-demo-bucket2",
                   "arn:aws:s3:::amzn-s3-demo-bucket2/*"],
      "Condition": {"StringNotEquals": {"aws:sourceVpce": "vpce-1a2b3c4d"}}
    }
  ]
}
```

Weitere Richtlinienbeispiele finden Sie unter Endpoints for Amazon S3 im VPCBenutzerhandbuch.

Weitere Informationen zur VPC Konnektivität finden Sie unter Network-to-VPC Konnektivitätsoptionen im AWS Whitepaper Amazon Virtual Private Cloud Connectivity Options.

Konformitätsvalidierung für Amazon S3

Die Sicherheit und Konformität von Amazon S3 wird von externen Prüfern im Rahmen mehrerer AWS Compliance-Programme bewertet, darunter die folgenden:

- System- und Organisationskontrollen (SOC)
- Datensicherheitsstandard der Zahlungskartenbranche (PCIDSS)
- Föderales Risiko- und Autorisierungsmanagementprogramm (FedRAMP)
- Gesetz über die Portabilität und Rechenschaftspflicht von Krankenversicherungen () HIPAA

AWS bietet unter AWS Services in Scope by Compliance Program eine häufig aktualisierte Liste der AWS Services im Rahmen bestimmter Compliance-Programme.

Prüfberichte von Drittanbietern stehen Ihnen zum Herunterladen zur Verfügung unter AWS Artifact. Weitere Informationen finden Sie unter Berichte in AWS Artifact herunterladen.

Weitere Informationen zu AWS Compliance-Programmen finden Sie unter AWS Compliance-Programme.

Welche Compliance-Verpflichtungen Sie bei der Verwendung von Amazon S3 haben, hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihrer Organisation und den geltenden Gesetzen und Vorschriften ab. Wenn Ihre Nutzung von Amazon S3 der Einhaltung von Standards wieHIPAA, oder Fed unterliegtPCI, AWS bietet Fed RessourcenRAMP, die Ihnen helfen:

- Schnellstartanleitungen zu Sicherheit und Compliance, in denen architektonische Überlegungen und Schritte zur Implementierung von sicherheits- und Compliance-orientierten Basisumgebungen erörtert werden. AWS
- Architecting for HIPAA Security and Compliance beschreibt, wie Unternehmen dabei helfen AWS , ihre Anforderungen zu erfüllen. HIPAA
- AWS Die Compliance-Ressourcen bieten verschiedene Arbeitsmappen und Leitfäden, die möglicherweise auf Ihre Branche und Ihren Standort zutreffen.
- AWS Config Mit können Sie bewerten, zu welchem Grad die Konfiguration Ihrer Ressourcen den internen Vorgehensweisen, Branchenrichtlinien und Vorschriften entspricht.
- AWS Security Hub bietet Ihnen einen umfassenden Überblick über Ihren Sicherheitsstatus AWS und hilft Ihnen dabei, Ihre Einhaltung der Sicherheitsstandards und Best Practices der Sicherheitsbranche zu überprüfen.

- Sperren von Objekten mit Object Lockkann Ihnen helfen, die technischen Anforderungen der Aufsichtsbehörden für Finanzdienstleistungen zu erfüllen (wie z. B. die SECFINRA, undCFTC), die für bestimmte Arten von Buch- und Datensatzinformationen Datenspeicher vorschreiben, die einmal schreiben und viele (WORM) lesen müssen.
- Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory können Sie bei der Prüfung und Meldung des Replikations- und Verschlüsselungsstatus Ihrer Objekte für Unternehmens-, Compliance- und regulatorische Anforderungen verwenden.

Ausfallsicherheit bei Amazon S3

Die AWS globale Infrastruktur basiert auf Regionen und Availability Zones. AWS-Regionen bieten mehrere, physisch getrennte und isolierte Availability Zones, die über geringe Latenz, hohen Durchsatz und hochredundante Netzwerke miteinander verbunden sind. Diese Availability Zones bieten Ihnen eine effektive Methode zum Entwerfen und Betreiben von Anwendungen und Datenbanken. Availability Zones sind in noch größeren Ausmaß hochverfügbar und sie sind fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren. Wenn Sie Ihre Daten speziell über größere geografische Entfernungen replizieren müssen, können Sie diese Option verwenden, die das automatischeObjekte innerhalb und zwischen Regionen replizieren, asynchrone Kopieren von Objekten zwischen Buckets in verschiedenen Buckets ermöglicht. AWS-Regionen

Jede Zone AWS-Region hat mehrere Availability Zones. Sie können Ihre Anwendungen über mehrere Availability Zones in derselben Region bereitstellen, um eine bessere Fehlertoleranz und niedriger Latenz zu erzielen. Availability Zones sind mit schnellen, privaten Glasfasernetzwerken verbunden. Dies ermöglicht Ihnen die Nutzung von Anwendungen, für die ein automatischer, unterbrechungsfreier Failover zwischen den Availability Zones eingerichtet ist.

Weitere Informationen zu Availability Zones AWS-Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Zusätzlich zur AWS globalen Infrastruktur bietet Amazon S3 mehrere Funktionen, um Ihre Anforderungen an Datenstabilität und Datensicherung zu erfüllen.

Lebenszyklus-Konfiguration

> Eine Lebenszyklus-Konfiguration besteht aus einer Reihe von Regeln, mit denen Aktionen definiert werden, die Amazon S3 auf eine Gruppe von Objekten anwendet. Mithilfe der Konfigurationsregeln für den Lebenszyklus können Sie Amazon S3 anweisen, Objekte in kostengünstigere Speicherklassen zu übergeben bzw. zu archivieren oder zu löschen. Weitere Informationen finden Sie unter Verwaltung des Lebenszyklus von Objekten.

Versioning

> Das Versioning ermöglicht Ihnen, mehrere Versionen eines Objekts im selben Bucket aufzubewahren. Sie können Versioning verwenden, um sämtliche Versionen aller Objekte in Ihrem Amazon S3 Bucket zu speichern, abzurufen oder wiederherzustellen. Daten lassen sich dank Versioning nach unbeabsichtigten Benutzeraktionen und Anwendungsfehlern leicht

wiederherstellen. Weitere Informationen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

S3-Objektsperre

Sie können S3 Object Lock verwenden, um Objekte mit einem Write-Once-Read-Many (WORM) -Modell zu speichern. Mit der S3-Objektsperre können Sie für einen festen Zeitraum oder auf unbegrenzte Zeit verhindern, dass ein Objekt gelöscht oder überschrieben wird. Mit S3 Object Lock können Sie gesetzliche Anforderungen erfüllen, die WORM Speicherplatz erfordern, oder einfach eine zusätzliche Schutzebene vor Objektänderungen und -löschungen hinzufügen. Weitere Informationen finden Sie unter Sperren von Objekten mit Object Lock.

Speicherklassen

Amazon S3 bietet je nach den Anforderungen Ihrer Workload eine Reihe von Speicherklassen an. Die Speicherklassen S3 Standard-IA und S3 One Zone-IA sind für Daten konzipiert, auf die Sie etwa einmal im Monat zugreifen und auf Millisekunden zugreifen müssen. Die Speicherklasse S3 Glacier Instant Retrieval ist für langlebige Archivdaten konzipiert, auf die Sie mit Millisekunden-Zugriff zugreifen, auf den Sie etwa einmal pro Quartal zugreifen. Für Archivdaten, die keinen sofortigen Zugriff erfordern, wie zum Beispiel Backups, können Sie die Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive verwenden. Weitere Informationen finden Sie unter Amazon S3 S3-Speicherklassen verstehen und verwalten.

Die folgenden bewährten Sicherheitsmethoden umfassen ebenfalls die Ausfallsicherheit:

- Enable versioning
- Consider Amazon S3 cross-region replication
- Identify and audit all your Amazon S3 buckets

Verschlüsselung von Amazon-S3-Sicherungen

Wenn Sie Backups mit Amazon S3 speichern, hängt die Verschlüsselung Ihrer Backups von der Konfiguration dieser Buckets ab. Die Amazon-S3-Standard-Verschlüsselung bietet eine Methode zum Festlegen des Verhaltens der Standard-Verschlüsselung für einen S3-Bucket. Sie können die Standard-Verschlüsselung in einem Bucket festlegen, sodass alle Objekte beim Speichern im Bucket verschlüsselt werden. Die Standardverschlüsselung unterstützt Schlüssel, die in AWS KMS (SSE-KMS) gespeichert sind. Weitere Informationen finden Sie unter Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets.

Weitere Informationen zu Versioning und Objektsperre finden Sie in den folgenden Themen: Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning Sperren von Objekten mit Object Lock

Infrastruktursicherheit in Amazon S3

Als verwalteter Service ist Amazon S3 durch die AWS globalen Netzwerksicherheitsverfahren geschützt, die in der Sicherheitssäule des AWS Well-Architected Framework beschrieben sind.

Der Zugriff auf Amazon S3 über das Netzwerk erfolgt über AWS PublishedAPIs. Kunden müssen Transport Layer Security (TLS) 1.2 unterstützen. Wir empfehlen, auch TLS 1.3 zu unterstützen. (Weitere Informationen zu dieser Empfehlung finden Sie unter Schnellere AWS Cloud-Verbindungen mit TLS 1.3 im AWS Sicherheitsblog.) Kunden müssen außerdem Cipher Suites mit Perfect Forward Secrecy (PFS) wie Ephemeral Diffie-Hellman () oder Elliptic Curve Diffie-Hellman Ephemeral (DHE) unterstützen. ECDHE Darüber hinaus müssen Anfragen mit Signature V4 oder Signature V2 signiert werden, wobei gültige Anmeldeinformationen erforderlich sind. AWS AWS

Diese APIs können von jedem Netzwerkstandort aus aufgerufen werden. Da Amazon S3 jedoch ressourcenbasierte Zugriffsrichtlinien unterstützt, kann es zu Einschränkungen bezüglich der Quell-IP-Adresse kommen. Sie können auch Amazon S3 S3-Bucket-Richtlinien verwenden, um den Zugriff auf Buckets von bestimmten Virtual Private Cloud (VPC) -Endpunkten oder bestimmten zu kontrollieren. VPCs Dadurch wird der Netzwerkzugriff auf einen bestimmten Amazon S3 S3-Bucket effektiv nur von den spezifischen VPC innerhalb des AWS Netzwerks isoliert. Weitere Informationen finden Sie unter Steuern des Zugriffs von VPC Endpunkten aus mit Bucket-Richtlinien.

Die folgenden bewährten Sicherheitsmethoden beziehen sich ebenfalls auf die Sicherheit der Infrastruktur von Amazon S3:

- Consider VPC endpoints for Amazon S3 access
- Identify and audit all your Amazon S3 buckets

Konfigurations- und Schwachstellenanalyse in Amazon S3

AWS erledigt grundlegende Sicherheitsaufgaben wie das Patchen von Gastbetriebssystemen (OS) und Datenbanken, die Firewall-Konfiguration und die Notfallwiederherstellung. Diese Verfahren wurden von qualifizierten Dritten überprüft und zertifiziert. Weitere Informationen finden Sie in den folgenden Ressourcen:

- Konformitätsvalidierung für Amazon S3
- Modell der übergreifenden Verantwortlichkeit
- Amazon Web Services – Übersicht über Sicherheitsverfahren

Die folgenden bewährten Sicherheitsmethoden beinhalten auch die Adresskonfiguration und Schwachstellenanalyse in Amazon S3:

- Identify and audit all your Amazon S3 buckets
- Aktivieren AWS Config

Zugriffsverwaltung

Amazon S3 bietet eine Vielzahl von Tools zur Zugriffsverwaltung. Im Folgenden finden Sie eine Liste dieser Funktionen und Tools. Sie benötigen nicht alle diese Zugriffsverwaltungstools, aber Sie müssen eines oder mehrere verwenden, um Zugriff auf Ihre Amazon S3 S3-Buckets, Objekte und andere S3-Ressourcen zu gewähren. Durch die richtige Anwendung dieser Tools kann sichergestellt werden, dass Ihre Ressourcen nur den vorgesehenen Benutzern zugänglich sind.

Das am häufigsten verwendete Zugriffsverwaltungstool ist eine Zugriffsrichtlinie. Eine Zugriffsrichtlinie kann eine ressourcenbasierte Richtlinie sein, die an eine AWS Ressource angehängt ist, z. B. eine Bucket-Richtlinie für einen Bucket. Eine Zugriffsrichtlinie kann auch eine identitätsbasierte Richtlinie sein, die mit einer AWS Identity and Access Management (IAM) Identität verknüpft ist, z. B. einem IAM Benutzer, einer Gruppe oder einer Rolle. Eine Zugriffsrichtlinie beschreibt, wer Zugriff auf welche Dinge hat. Schreiben Sie eine Zugriffsrichtlinie, um IAM Benutzern, Gruppen und Rollen die Erlaubnis zu erteilen, Operationen an einer Ressource auszuführen. AWS-Konten Sie können beispielsweise einem anderen Konto die `PUT Object` Erlaubnis erteilen, AWS-Konto sodass das andere Konto Objekte in Ihren Bucket hochladen kann.

Im Folgenden sind die in Amazon S3 verfügbaren Zugriffsverwaltungstools aufgeführt. Eine umfassendere Anleitung zur Amazon S3 S3-Zugriffskontrolle finden Sie unterZugriffskontrolle in Amazon S3.

Bucket-Richtlinie

Eine Amazon S3 S3-Bucket-Richtlinie ist eine JSON -formatierte AWS Identity and Access Management (IAM) ressourcenbasierte Richtlinie, die an einen bestimmten Bucket angehängt ist. Verwenden Sie Bucket-Richtlinien, um anderen Benutzern AWS-Konten oder IAM Identitäten Berechtigungen für den Bucket und die darin enthaltenen Objekte zu gewähren. Viele Anwendungsfälle für die S3-Zugriffsverwaltung können durch die Verwendung einer Bucket-Richtlinie erfüllt werden. Mit Bucket-Richtlinien können Sie den Bucket-Zugriff personalisieren, um sicherzustellen, dass nur die von Ihnen genehmigten Identitäten auf Ressourcen zugreifen und darin Aktionen ausführen können. Weitere Informationen finden Sie unter Bucket-Richtlinien für Amazon S3.

Identitätsbasierte Richtlinien

Eine identitätsbasierte Richtlinie oder IAM Benutzerrichtlinie ist eine Art von AWS Identity and Access Management Richtlinie (). IAM Eine identitätsbasierte Richtlinie ist eine JSON -formatierte Richtlinie, die IAM Benutzern, Gruppen oder Rollen in Ihrem Konto zugeordnet ist. AWS Sie können identitätsbasierte Richtlinien verwenden, um einer IAM Identität Zugriff auf Ihre Buckets oder Objekte zu gewähren. Sie können IAM Benutzer, Gruppen und Rollen in Ihrem Konto erstellen und ihnen Zugriffsrichtlinien zuordnen. Sie können dann Zugriff auf AWS Ressourcen gewähren, einschließlich Amazon S3 S3-Ressourcen. Weitere Informationen finden Sie unter Identitätsbasierte Richtlinien für Amazon S3.

S3 Access Grants

Verwenden Sie S3 Access Grants, um Zugriffsberechtigungen für Ihre Amazon S3 S3-Daten für beide Identitäten in Unternehmensidentitätsverzeichnissen zu erstellen, z. B. Active Directory, und zu AWS Identity and Access Management (IAM) Identitäten. S3 Access Grants unterstützt Sie bei der Verwaltung von Datenberechtigungen in großem Umfang. Darüber hinaus protokolliert S3 Access Grants die Identität des Endbenutzers und die Anwendung, die für den Zugriff auf die S3-Daten verwendet wurde. AWS CloudTrail Dies bietet einen detaillierten Auditverlauf bis hin zur Endbenutzeridentität für den gesamten Zugriff auf die Daten in Ihren S3-Buckets. Weitere Informationen finden Sie unter Verwalten des Zugriffs mit S3-Zugriffsberechtigungen.

Zugriffspunkte

Amazon S3 Access Points vereinfacht die Verwaltung des Datenzugriffs in großem Umfang für Anwendungen, die gemeinsam genutzte Datensätze auf S3 verwenden. Access Points werden als Netzwerkendpunkte bezeichnet, die an einen Bucket angeschlossen sind. Sie können Access Points verwenden, um S3-Objektoperationen in großem Umfang durchzuführen, wie z. B. das Hochladen und Abrufen von Objekten. An einen Bucket können bis zu 10.000 Access Points angeschlossen sein, und für jeden Access Point können Sie eigene Berechtigungen und Netzwerkkontrollen durchsetzen, um den Zugriff auf Ihre S3-Objekte detailliert zu kontrollieren. S3 Access Points können Buckets im selben Konto oder in einem anderen vertrauenswürdigen Konto zugeordnet werden. Access Point-Richtlinien sind ressourcenbasierte Richtlinien, die in Verbindung mit der zugrunde liegenden Bucket-Richtlinie bewertet werden. Weitere Informationen finden Sie unter Verwaltung des Zugriffs auf gemeinsam genutzte Datensätze mit Access Points.

Liste der Zugriffskontrollen () ACL

An ACL ist eine Liste von Zuschüssen, in der der Empfänger und die erteilte Genehmigung identifiziert werden. ACLsanderen AWS-Konten Personen grundlegende Lese- oder Schreibberechtigungen gewähren. ACLsverwenden Sie ein Amazon S3 S3-spezifisches SchemaXML. An ACL ist eine Art von Richtlinie AWS Identity and Access Management () IAM. Ein Objekt ACL wird verwendet, um den Zugriff auf ein Objekt zu verwalten, und ein Bucket ACL wird verwendet, um den Zugriff auf einen Bucket zu verwalten. Bei Bucket-Richtlinien gibt es eine einzige Richtlinie für den gesamten Bucket, aber Objekte ACLs werden für jedes Objekt angegeben. Es wird empfohlen, die Option ACLs ausgeschaltet zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Weitere Informationen zur Verwendung finden ACLs Sie unterKontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

> ⚠ Warning
>
> Die meisten modernen Anwendungsfälle in Amazon S3 erfordern nicht die Verwendung vonACLs.

Eigentum an Objekten

Um den Zugriff auf Ihre Objekte zu verwalten, müssen Sie der Eigentümer des Objekts sein. Sie können die Einstellung „Objekteigentum“ auf Bucketebene verwenden, um den Besitz von Objekten zu kontrollieren, die in Ihren Bucket hochgeladen wurden. Verwenden Sie auch „Objekteigentum“, um die Option zu aktivieren. ACLs Standardmäßig ist für Object Ownership die Einstellung Bucket Owner erforced festgelegt und alle Optionen ACLs sind deaktiviert. Wenn ACLs diese Option deaktiviert

ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet ausschließlich den Zugriff auf Daten. Um den Zugriff zu verwalten, verwendet der Bucket-Besitzer Richtlinien oder ein anderes Zugriffsverwaltungstool, ausgenommenACLs. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

Einen umfassenderen Leitfaden zur Amazon S3 S3-Zugriffskontrolle und weitere bewährte Methoden finden Sie unterZugriffskontrolle in Amazon S3.

Datenschutz in Amazon S3

Zusätzlich zu der Stabilität, die die AWS globale Infrastruktur bietet, bietet Amazon S3 eine Reihe von Funktionen, mit denen Sie Ihre Daten vor versehentlichem Löschen oder regionalen Ausfällen schützen können.

S3-Replikation

Sie können die Live-Replikation verwenden, um das automatische, asynchrone Kopieren von Objekten zwischen Amazon S3 S3-Buckets zu ermöglichen. Buckets, die für die Objektreplikation konfiguriert sind, können demselben AWS-Konto oder unterschiedlichen Konten gehören. Sie können Objekte in einen einzelnen Ziel-Bucket oder in mehrere Ziel-Buckets replizieren. Die Ziel-Buckets können sich in einer anderen Region AWS-Regionen oder in derselben Region wie der Quell-Bucket befinden. Um die Failover-Steuerung zu aktivieren, können Sie die Replikation so konfigurieren, dass die Replikation bidirektional (bidirektional) erfolgt, sodass Ihre Quell- und Ziel-Buckets bei einem regionalen Ausfall synchron gehalten werden können. Weitere Informationen finden Sie unter the section called “Objekte innerhalb und zwischen Regionen replizieren”.

Access Points und Failover-Steuerungen für mehrere Regionen

Amazon S3 Multi-Region Access Points stellen einen globalen Endpunkt bereit, mit dem Anwendungen Anforderungen von S3-Buckets ausführen können, die sich in mehreren AWS-Regionen befinden. Sie können Multi-Regions Access Points verwenden, um Multi-Regions-Anwendungen mit derselben Architektur zu erstellen, die in einer einzelnen Region verwendet wird, und diese Anwendungen dann überall auf der Welt ausführen. Anstatt Anforderungen über das überlastete öffentliche Internet zu senden, bieten Multi-Regions-Zugriffspunkte integrierte Netzwerkausfallsicherheit mit Beschleunigung internetbasierter Anforderungen an Amazon S3. Anwendungsanforderungen an einen globalen Endpunkt von Multi-Region Access Points verwenden AWS Global Accelerator für die automatische Weiterleitung über das globale AWS -Netzwerk an den nächstgelegenen S3-Bucket mit einem aktiven Routing-Status. Weitere Informationen zu Access Points für mehrere Regionen finden Sie unter. the section called “Verwaltung des Datenverkehrs in mehreren Regionen ”

Mit den Failover-Kontrollen für Amazon S3 Multi-Region Access Points können Sie die Geschäftskontinuität bei regionalen Verkehrsunterbrechungen aufrechterhalten und Ihren Anwendungen gleichzeitig eine regionsübergreifende Architektur zur Erfüllung von Compliance- und Redundanzanforderungen bieten. Wenn Ihr regionaler Datenverkehr unterbrochen wird, können Sie mithilfe der Failover-Steuerelemente für mehrere Regionen auswählen, welcher AWS-

Regionen hinter einem Amazon S3-Access Point mit mehreren Regionen Datenzugriffs- und Speicheranfragen verarbeitet werden soll.

Zur Failover-Unterstützung können Sie Ihren Multi-Region Access Point in einer Aktiv-Passiv-Konfiguration einrichten, wobei der Datenverkehr unter normalen Bedingungen in die aktive Region fließt und sich eine passive Region für den Failover im Standby-Modus befindet. Wenn Sie die regionsübergreifende S3-Replikation (CRR) mit bidirektionalen Replikationsregeln aktiviert haben, können Sie Ihre Buckets während eines Failovers synchronisieren. Weitere Informationen zu Failover-Steuerelementen finden Sie unter. the section called “Failover-Konfiguration”

S3-Versioning

Das Versioning ermöglicht Ihnen, mehrere Versionen eines Objekts im selben Bucket aufzubewahren. Sie können Versioning verwenden, um sämtliche Versionen aller Objekte in Ihrem Amazon S3 Bucket zu speichern, abzurufen oder wiederherzustellen. Daten lassen sich dank Versioning nach unbeabsichtigten Benutzeraktionen und Anwendungsfehlern leicht wiederherstellen. Weitere Informationen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

S3-Objektsperre

Sie können S3 Object Lock verwenden, um Objekte mit einem Write-Once-Read-Many (WORM) -Modell zu speichern. Mit der S3-Objektsperre können Sie für einen festen Zeitraum oder auf unbegrenzte Zeit verhindern, dass ein Objekt gelöscht oder überschrieben wird. Mit S3 Object Lock können Sie gesetzliche Anforderungen erfüllen, die WORM Speicherplatz erfordern, oder einfach eine zusätzliche Schutzebene vor Objektänderungen und -löschungen hinzufügen. Weitere Informationen finden Sie unter Sperren von Objekten mit Object Lock.

AWS Backup

Amazon S3 ist nativ integriert AWS Backup, ein vollständig verwalteter, richtlinienbasierter Service, mit dem Sie Backup-Richtlinien zum Schutz Ihrer Daten in Amazon S3 zentral definieren können. Nachdem Sie Ihre Backup-Richtlinien definiert und Amazon S3 S3-Ressourcen den Richtlinien zugewiesen haben, AWS Backup automatisiert die Erstellung von Amazon S3 S3-Backups und speichert die Backups sicher in einem verschlüsselten Backup-Tresor, den Sie in Ihrem Backup-Plan angeben. Weitere Informationen finden Sie unter the section called “Sichern von Daten”.

Ein Tutorial zur gemeinsamen Verwendung einiger dieser Funktionen zum Schutz Ihrer Daten finden Sie unter Tutorial: Schützen von Daten auf Amazon S3 vor versehentlichem Löschen oder Anwendungsfehlern mithilfe von S3 Versioning, S3 Object Lock und S3 Replication.

> **Important**
>
> Zusätzlich zur Verwendung der oben genannten Funktionen zum Schutz Ihrer Daten empfehlen wir, die Empfehlungen unter zu lesen. the section called “Bewährte Methoden für die Gewährleistung der Sicherheit”

Themen

- Objekte innerhalb und zwischen Regionen replizieren
- Verwaltung von regionsübergreifendem Verkehr mit multiregionalen Access Points
- Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning
- Sperren von Objekten mit Object Lock
- Sicherung Ihrer Amazon S3 S3-Daten

Objekte innerhalb und zwischen Regionen replizieren

Sie können die Replikation verwenden, um das automatische, asynchrone Kopieren von Objekten zwischen Amazon S3 S3-Buckets zu ermöglichen. Buckets, die für die Objektreplikation konfiguriert sind, können sich im Besitz desselben AWS-Konto oder unterschiedlicher Konten befinden. Sie können Objekte in einen einzelnen Ziel-Bucket oder in mehrere Ziel-Buckets replizieren. Die Ziel-Buckets können sich in einer anderen Region AWS-Regionen oder in derselben Region wie der Quell-Bucket befinden.

Es gibt zwei Arten der Replikation: Live-Replikation und On-Demand-Replikation.

- Live-Replikation — Verwenden Sie die Live-Replikation, um neue und aktualisierte Objekte automatisch zu replizieren, während sie in den Quell-Bucket geschrieben werden. Bei der Live-Replikation werden keine Objekte repliziert, die vor der Einrichtung der Replikation im Bucket vorhanden waren. Um Objekte zu replizieren, die vor der Einrichtung der Replikation vorhanden waren, verwenden Sie die On-Demand-Replikation.
- Replikation auf Abruf — Verwenden Sie S3 Batch Replication, um bestehende Objekte bei Bedarf aus dem Quell-Bucket in einen oder mehrere Ziel-Buckets zu replizieren. Weitere Informationen

zum Replizieren vorhandener Objekte finden Sie unter Wann die S3-Batch-Replikation verwendet wird.

Es gibt zwei Formen der Live-Replikation: Regionsübergreifende Replikation (CRR) und Replikation in derselben Region (). SRR

- Regionsübergreifende Replikation (CRR) — Sie können CRR damit Objekte in verschiedenen Amazon S3 S3-Buckets replizieren. AWS-Regionen Weitere Informationen CRR zu finden Sie unter. the section called “Verwenden der regionsübergreifenden Replikation”
- Replikation in derselben Region (SRR) — Sie können sie verwendenSRR, um Objekte zwischen Amazon S3 S3-Buckets im selben zu kopieren. AWS-Region Weitere Informationen SRR zu finden Sie unter. the section called “Verwenden von Replikation innerhalb derselben Region”

Themen

- Warum sollten Sie Replikation verwenden?
- Verwenden der regionsübergreifenden Replikation
- Verwenden von Replikation innerhalb derselben Region
- Wann sollte die bidirektionale Replikation verwendet werden
- Wann die S3-Batch-Replikation verwendet wird
- Workload-Anforderungen und Live-Replikation
- Was repliziert Amazon S3?
- Anforderungen und Überlegungen für die Replikation
- Überblick über die Einrichtung der Live-Replikation
- Live-Replikation verwalten oder pausieren
- Replizieren vorhandener Objekte mit Batch Replication
- Fehlerbehebung bei einer Replikation
- Überwachung der Replikation mit Metriken, Ereignisbenachrichtigungen und Status

Warum sollten Sie Replikation verwenden?

Die Replikation unterstützt Sie bei Folgendem:

- Objekte replizieren und dabei Metadaten beibehalten — Sie können die Replikation verwenden, um Kopien Ihrer Objekte zu erstellen, die alle Metadaten, wie z. B. die Erstellungszeit und Version IDs des ursprünglichen Objekts, beibehalten. Diese Funktion ist wichtig, wenn Sie sicherstellen müssen, dass Ihr Replikat mit dem Quellobjekt identisch ist.
- Replikation von Objekten in verschiedene Speicherklassen – Sie können mit der Replikation Objekte direkt in S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive oder eine andere Speicherklasse in den Ziel-Buckets platzieren. Sie können Ihre Daten auch in dieselbe Speicherklasse replizieren und Lebenszyklus-Konfigurationen für die Ziel-Buckets verwenden, um Ihre Objekte bei zunehmendem Alter in eine Speicherklasse für seltener genutzte Objekte zu verschieben.
- Objektkopien unter unterschiedlichen Eigentümern verwalten — Unabhängig davon, wem das Quellobjekt gehört, können Sie Amazon S3 anweisen, den Besitz des Replikats auf das Objekt zu ändern AWS-Konto , dem der Ziel-Bucket gehört. Diese Instance wird als Eigentümer-Überschreibungs-Option bezeichnet. Sie können diese Option nutzen, um den Zugriff auf Objektreplikate einzuschränken.
- Objekte über mehrere speichern lassen AWS-Regionen — Um geografische Unterschiede in Bezug auf den Speicherort Ihrer Daten zu gewährleisten, können Sie mehrere Ziel-Buckets für verschiedene festlegen. AWS-Regionen Diese Funktion kann Ihnen helfen, bestimmte Compliance-Anforderungen zu erfüllen.
- Objekte innerhalb von 15 Minuten replizieren — Um Ihre Daten innerhalb eines vorhersehbaren Zeitraums in derselben AWS-Region oder in verschiedenen Regionen zu replizieren, können Sie S3 Replication Time Control (S3) verwenden. RTC S3 RTC repliziert 99,99 Prozent der in Amazon S3 gespeicherten neuen Objekte innerhalb von 15 Minuten (unterstützt durch ein Service Level Agreement). Weitere Informationen finden Sie unter the section called “Verwenden der S3-Replikationszeitkontrolle”.

  > **Note**
  >
  > S3 gilt RTC nicht für die Batch-Replikation. Die Batch-Replikation ist ein On-Demand-Replikationsauftrag und kann mit S3-Batchvorgängen verfolgt werden. Weitere Informationen finden Sie unter Verfolgen von Auftragsstatus- und Abschluss.

- Synchronisieren Sie Buckets, replizieren Sie vorhandene Objekte und replizieren Sie zuvor fehlgeschlagene oder replizierte Objekte – Verwenden Sie die Batch-Replikation als On-Demand-Replikationsaktion, um Buckets zu synchronisieren und vorhandene Objekte zu replizieren. Weitere

Informationen dazu, wann Sie die Batch-Replikation verwenden sollten, finden Sie unter Wann die S3-Batch-Replikation verwendet wird.

- Objekte replizieren und zu einem Bucket in einer anderen AWS-Region wechseln – Damit während der Datenreplikation alle Metadaten und Objekte zwischen Buckets synchron bleiben, verwenden Sie Regeln für die bidirektionale Replikation, bevor Sie die Failover-Kontrollen für Amazon S3 Multi-Region Access Points konfigurieren. Bidirektionale Replikationsregeln tragen dazu bei, sicherzustellen, dass wenn Daten in den S3-Bucket geschrieben werden, auf den der Datenverkehr bei einem Failover zurückgreift, diese Daten dann zurück in den Quell-Bucket repliziert werden.

Verwenden der regionsübergreifenden Replikation

S3 Cross-Region Replication (CRR) wird verwendet, um Objekte zwischen Amazon S3 S3-Buckets in verschiedene zu kopieren. AWS-Regionen CRRkann Ihnen dabei helfen, Folgendes zu tun:

- Einhalten der Compliance-Anforderungen – Auch wenn Amazon S3 Ihre Daten standardmäßig in mehreren geografisch entfernten Availability Zones speichert, machen es die Compliance-Anforderungen möglicherweise erforderlich, Daten in noch größeren Entfernungen zu speichern. Um diese Anforderungen zu erfüllen, verwenden Sie die Replikation in mehreren Regionen, um Daten zwischen entfernten AWS-Regionen zu replizieren.
- Latenz minimieren — Wenn sich Ihre Kunden an zwei geografischen Standorten befinden, können Sie die Latenz beim Zugriff auf Objekte minimieren, indem Sie Objektkopien speichern AWS-Regionen , die sich geografisch näher an Ihren Benutzern befinden.
- Steigern Sie die betriebliche Effizienz — Wenn Sie Rechencluster in zwei verschiedenen Bereichen haben AWS-Regionen , die dieselbe Gruppe von Objekten analysieren, können Sie sich dafür entscheiden, Objektkopien in diesen Regionen zu verwalten.

Verwenden von Replikation innerhalb derselben Region

Same-Region Replication (SRR) wird verwendet, um Objekte zwischen Amazon S3 S3-Buckets in demselben zu kopieren. AWS-Region SRRkann Ihnen dabei helfen, Folgendes zu tun:

- Aggregieren von Protokollen in einen einzelnen Bucket – Wenn Sie Protokolle in mehrere Buckets oder kontoübergreifend speichern, können Sie Protokolle ganz einfach in einen einzelnen Bucket innerhalb derselben Region replizieren. Dies ermöglicht eine einfachere Protokollverarbeitung an einem einzelnen Standort.

- Konfigurieren von Live-Replikation zwischen Produktions- und Testkonten – Wenn Sie oder Ihre Kunden Produktions- und Testkonten haben, die dieselben Daten nutzen, können Sie Objekte kontoübergreifend replizieren und dabei die Objektmetadaten erhalten.
- Halten Sie sich an die Gesetze zur Datenhoheit — Möglicherweise müssen Sie AWS-Konten innerhalb einer bestimmten Region mehrere Kopien Ihrer Daten getrennt speichern. Mit der Replikation innerhalb derselben Region können Sie automatisch kritische Daten replizieren, wenn Ihre Daten aufgrund von Compliance-Regelungen im Land bleiben müssen.

Wann sollte die bidirektionale Replikation verwendet werden

- Erstellen Sie gemeinsam genutzte Datensätze für mehrere AWS-Regionen — Mit der Replikatmodifikationssynchronisierung können Sie Metadatenänderungen wie Objektzugriffskontrolllisten (ACLs), Objekt-Tags oder Objektsperren auf einfache Weise auf Replikationsobjekte replizieren. Diese bidirektionale Replikation ist wichtig, wenn Sie alle Objekte und Objektmetadatenänderungen synchron halten möchten. Sie können die Synchronisierung von Replikatänderungen für eine neue oder bestehende Replikationsregel aktivieren, wenn Sie eine bidirektionale Replikation zwischen zwei oder mehr Buckets in derselben oder verschiedenen AWS-Regionen durchführen.
- Sorgen Sie dafür, dass Daten während eines Failovers regionsübergreifend synchronisiert werden — Sie können Daten in Buckets zwischen den einzelnen Regionen synchronisieren, AWS-Regionen indem Sie Regeln für die bidirektionale Replikation mit der regionsübergreifenden S3-Replikation (CRR) direkt von einem Multi-Region-Access Point aus konfigurieren. Um eine fundierte Entscheidung darüber zu treffen, wann ein Failover eingeleitet werden soll, können Sie auch S3-Replikationsmetriken aktivieren, sodass Sie die Replikation in Amazon CloudWatch, in S3 Replication Time Control (S3RTC) oder vom Multi-Region Access Point aus überwachen können.
- Sorgen Sie für hohe Verfügbarkeit Ihrer Anwendung – Selbst im Fall einer regionalen Datenverkehrsunterbrechung können Sie bidirektionale Replikationsregeln verwenden, um alle Metadaten und Objekte während der Datenreplikation bucketübergreifend synchron zu halten.

Wann die S3-Batch-Replikation verwendet wird

Die Batch-Replikation repliziert vorhandene Objekte als On-Demand-Option in verschiedene Buckets. Im Gegensatz zur Live-Replikation können diese Aufträge nach Bedarf ausgeführt werden. Die Batch-Replikation unterstützt Sie bei Folgendem:

- Replizieren vorhandener Objekte – Sie können die Batch-Replikation verwenden, um Objekte zu replizieren, die dem Bucket hinzugefügt wurden, bevor die Replikation innerhalb derselben Region oder die Replikation in mehreren Regionen konfiguriert wurden.
- Objekte replizieren, die zuvor nicht repliziert werden konnten — Sie können einen Batch-Replikationsauftrag filtern, um zu versuchen, Objekte mit dem Replikationsstatus von zu replizieren. FAILED
- Replizieren von Objekten, die bereits repliziert wurden – Möglicherweise müssen Sie mehrere Kopien Ihrer Daten in separaten AWS-Konten oder AWS-Regionen speichern. Die Batch-Replikation kann vorhandene Objekte an neu hinzugefügte Ziele replizieren.
- Replizieren von Replikaten von Objekten, die aus einer Replikationsregel erstellt wurden – Replikationskonfigurationen erstellen Replikate von Objekten in Ziel-Buckets. Replikate von Objekten können nur mit Batch-Replikation repliziert werden.

Workload-Anforderungen und Live-Replikation

Abhängig von Ihren Workload-Anforderungen eignen sich einige Arten der Live-Replikation besser für Ihren Anwendungsfall als andere. Ermitteln Sie anhand der folgenden Tabelle, welche Art von Replikation Sie für Ihre Situation verwenden sollten und ob Sie S3 Replication Time Control (S3RTC) für Ihren Workload verwenden sollten. S3 RTC repliziert 99,99 Prozent der in Amazon S3 gespeicherten neuen Objekte innerhalb von 15 Minuten (unterstützt durch ein Service Level Agreement oder). SLA Weitere Informationen finden Sie unter the section called “Verwenden der S3-Replikationszeitkontrolle”.

Arbeitslastanforderung	S3 RTC (15 MinutenSLA)	Regionsübergreifende Replikation () CRR	Replikation in derselben Region () SRR
Replizieren Sie Objekte zwischen verschiedenen AWS-Konten	Ja	Ja	Ja
Replizieren Sie Objekte innerhalb derselben AWS-Region innerhalb von	Nein	Nein	Ja

Arbeitslastanforderung	S3 RTC (15 MinutenSLA)	Regionsübergreifende Replikation () CRR	Replikation in derselben Region () SRR
24-48 Stunden (nicht gesichert) SLA			
Objekte zwischen verschiedenen Objekten AWS-Regionen innerhalb von 24-48 Stunden replizieren (nicht gesichert) SLA	Nein	Ja	Nein
Vorhersehbare Replikationszeit: Unterstützt durchSLA, dass 99,9 Prozent der Objekte innerhalb von 15 Minuten repliziert werden	Ja	Nein	Nein

Was repliziert Amazon S3?

Amazon S3 repliziert nur bestimmte Elemente in Buckets, die für die Replikation konfiguriert sind.

Themen

- Was wird mit Replikationskonfigurationen repliziert?
- Was wird mit Replikationskonfigurationen nicht repliziert?

Was wird mit Replikationskonfigurationen repliziert?

Standardmäßig repliziert Amazon S3 Folgendes:

- Objekte, die nach dem Hinzufügen einer Replikations-Konfiguration erstellt wurden.
- Unverschlüsselte Objekte.

- Objekte, die mit vom Kunden bereitgestellten Schlüsseln (SSE-C) verschlüsselt wurden, Objekte, die im Ruhezustand unter einem von Amazon S3 verwalteten Schlüssel (SSE-S3) oder einem in AWS Key Management Service (SSE-KMS) gespeicherten KMS Schlüssel verschlüsselt wurden. Weitere Informationen finden Sie unter the section called “Replizieren verschlüsselter Objekte”.
- Objektmetadaten von den Quellobjekten zu den Replikaten Informationen zum Replizieren von Metadaten aus den Replikaten zu den Quellobjekten finden Sie unter Replizieren von Metadatenänderungen mit der Replikatmodifikationssynchronisierung.
- Nur Objekte im Quell-Bucket, für die der Bucket-Besitzer berechtigt ist, Objekte und Zugriffskontrolllisten () ACLs zu lesen.

  Weitere Informationen zum Eigentum an Ressourcen finden Sie unter Amazon-S3-Bucket- und Objekt-Eigentümerschaft.
- ACLObjektaktualisierungen, sofern Sie Amazon S3 nicht anweisen, den Besitz des Replikats zu ändern, wenn Quell- und Ziel-Buckets nicht denselben Konten gehören.

  Weitere Informationen finden Sie unter Ändern des Replikat-Eigentümers.

  Es kann eine Weile dauern, bis Amazon S3 die beiden ACLs synchronisieren kann. Diese Änderung der Eigentümerschaft gilt nur für Objekte, die erstellt wurden, nachdem Sie dem Bucket eine Replikationskonfiguration hinzugefügt haben.
- Objekt-Markierungen, sofern vorhanden.
- Informationen zur Beibehaltung der S3-Objektsperre, sofern vorhanden.

  Wenn Amazon S3 Objekte mit angewandten Aufbewahrungsinformationen repliziert, wendet es dieselben Aufbewahrungssteuerungen auf Ihre Replikate an, wodurch der für Ihre Ziel-Buckets konfigurierte standardmäßige Aufbewahrungszeitraum überschrieben wird. Wenn auf die Objekte in Ihrem Quell-Bucket keine Aufbewahrungssteuerungen angewandt sind und Sie sie in Ziel-Buckets mit festgelegtem Standard-Aufbewahrungszeitraum replizieren, wird der Standard-Aufbewahrungszeitraum der Ziel-Buckets auf Ihre Objektreplikate angewandt. Weitere Informationen finden Sie unter Sperren von Objekten mit Object Lock.

Auswirkungen von LöschVorgänge auf die Replikation

Wenn Sie ein Objekt aus dem Quell-Bucket löschen, werden standardmäßig die folgenden Aktionen ausgeführt:

- Wenn Sie eine DELETE Anfrage stellen, ohne eine Objektversions-ID anzugeben, fügt Amazon S3 eine Löschmarkierung hinzu. Amazon S3 befasst sich wie folgt mit der Löschmarkierung:
  - Wenn Sie die aktuelle Version der Replikations-Konfiguration verwenden, d. h. das Element `Filter` in einer Replikations-Konfigurations-Regel angeben, repliziert Amazon S3 die Löschmarkierung nicht standardmäßig. Sie können den non-tag-based Regeln jedoch die Replikation von Löschmarkierungen hinzufügen. Weitere Informationen finden Sie unter Replizieren von Löschmarkierungen auf Buckets.
  - Wenn Sie das Element `Filter` nicht angeben, geht Amazon S3 davon aus, dass die Replikations-Konfiguration Version V1 ist, und repliziert Löschmarkierungen, die aus Benutzeraktionen resultierten. Wenn Amazon S3 jedoch ein Objekt aufgrund einer Lebenszyklus-Aktion löscht, wird die Löschmarkierung nicht auf die Ziel-Buckets repliziert.
- Wenn Sie angeben, dass eine Objektversions-ID in einer `DELETE`-Anforderung gelöscht werden soll, löscht Amazon S3 diese Objektversion im Quell-Bucket. Die Löschung wird jedoch nicht in den Ziel-Buckets repliziert. Anders ausgedrückt: Dieselbe Objektversion wird aus den Ziel-Buckets nicht gelöscht. Dies schützt Daten vor missbräuchlichen Löschungen.

Was wird mit Replikationskonfigurationen nicht repliziert?

Standardmäßig repliziert Amazon S3 Folgendes nicht:

- Objekte im Quell-Bucket, bei denen es sich um Replikate handelt, die von einer anderen Replikationsregel erstellt wurden. Zum Beispiel: Angenommen Sie konfigurieren eine Replikation, bei der Bucket A die Quelle und Bucket B das Ziel ist. Nehmen wir jetzt an, Sie fügen eine weitere Replikations-Konfiguration hinzu, bei der Bucket B die Quelle und Bucket C das Ziel ist. In diesem Fall werden Objekte in Bucket B, die Replikate von Objekten in Bucket A sind, nicht in Bucket C repliziert.

  Verwenden Sie die Batch-Replikation, um Objekte zu replizieren, die Replikate sind. Weitere Informationen zum Konfigurieren der Batch-Replikation finden Sie unter Replizieren vorhandener Objekte.
- Objekte im Quell-Bucket, die bereits auf ein anderes Ziel repliziert wurden. Wenn Sie beispielsweise den Ziel-Bucket in einer vorhandenen Replikationskonfiguration ändern, repliziert Amazon S3 diese Objekte nicht erneut.

  Verwenden Sie die Batch-Replikation, um zuvor replizierte Objekte zu replizieren. Weitere Informationen zum Konfigurieren der Batch-Replikation finden Sie unter Replizieren vorhandener Objekte.

- Die Batch-Replikation unterstützt nicht das erneute Replizieren von Objekten, die mit der Versions-ID des Objekts aus dem Ziel-Bucket gelöscht wurden. Wenn Sie diese Objekte erneut replizieren möchten, können Sie die Quellobjekte mit einem Batch–Kopierauftrag kopieren. Beim Kopieren dieser Objekte werden neue Versionen des Objekts im Quell-Bucket erstellt und die Replikation zum Ziel wird automatisch initiiert. Weitere Informationen zur Verwendung von Batch Copy finden Sie unter Beispiele, die Batch-Vorgänge zum Kopieren von Objekten verwenden.
- Standardmäßig werden beim Replizieren aus einem anderen Bereich Löschmarken AWS-Konto, die dem Quell-Bucket hinzugefügt wurden, nicht repliziert.

  Informationen zum Replizieren von Löschmarkierungen finden Sie unter Replizieren von Löschmarkierungen auf Buckets.
- Objekte, die in den Speicherklassen oder Stufen S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive, S3 Intelligent-Tiering Archive Access oder S3 Intelligent-Tiering Deep Archive Access gespeichert sind. Sie können diese Objekte erst replizieren, wenn Sie sie wiederhergestellt und in eine andere Speicherklasse kopiert haben.

  Weitere Informationen zu S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive finden Sie unterSpeicherklassen für Objekte, auf die selten zugegriffen wird.

  Weitere Informationen über S3 Intelligent-Tiering finden Sie unter. Verwaltung der Speicherkosten mit Amazon S3 Intelligent-Tiering
- Objekte im Quell-Bucket, für die der Bucket-Eigentümer nicht über ausreichende Berechtigungen zur Replikation verfügt.

  Weitere Informationen darüber, wie ein Objekt-Eigentümer einem Bucket-Eigentümer Berechtigungen erteilen kann, finden Sie unter Erteilung von kontoübergreifenden Berechtigungen für das Hochladen von Objekten, wobei sichergestellt wird, dass der Bucket-Eigentümer volle Kontrolle besitzt.
- Aktualisierungen von Unterressourcen auf Bucket-Ebene.

  Wenn Sie beispielsweise die Lebenszyklus-Konfiguration ändern oder eine Benachrichtigungskonfiguration zu Ihrem Quell-Bucket hinzufügen, werden diese Änderungen nicht auf den Ziel-Bucket angewendet. Durch diese Funktion ist es möglich, für den Quell- und den Ziel-Bucket verschiedene Konfigurationen zu nutzen.
- Aktionen, die von der Lebenszyklus-Konfiguration durchgeführt werden.

Wenn eine Lebenszyklus-Konfiguration beispielsweise nur auf Ihrem Quell-Bucket aktiviert ist, erstellt Amazon S3 Löschmarkierungen für abgelaufene Objekte, repliziert diese Markierungen jedoch nicht. Wenn Sie dieselbe Lebenszyklus-Konfiguration sowohl auf den Quell- als auch auf den Ziel-Bucket anwenden möchten, aktivieren Sie für beide Buckets dieselbe Lebenszyklus-Konfiguration. Weitere Informationen zur Lebenszyklus-Konfiguration finden Sie unter Verwaltung des Lebenszyklus von Objekten.

- Wenn Sie tagbasierte Replikationsregeln mit Live-Replikation verwenden, müssen neue Objekte während des Vorgangs mit dem entsprechenden Tag für die Replikationsregel gekennzeichnet werden. `PutObject` Andernfalls werden die Objekte nicht repliziert. Wenn Objekte nach dem `PutObject` Vorgang markiert werden, werden diese Objekte ebenfalls nicht repliziert.

  Um Objekte zu replizieren, die nach dem `PutObject` Vorgang markiert wurden, müssen Sie S3 Batch Replication verwenden. Weitere Informationen zur Batch-Replikationen finden Sie unter Replizieren vorhandener Objekte.

Anforderungen und Überlegungen für die Replikation

Für die Amazon S3 S3-Replikation ist Folgendes erforderlich:

- Der Besitzer des Quell-Buckets muss die Quelle und das Ziel für AWS-Regionen sein Konto aktiviert haben. Für das Konto des Ziel-Bucket-Eigentümers muss die Zielregion aktiviert sein.

  Weitere Informationen zur Aktivierung oder Deaktivierung von finden Sie im AWS Account Management Referenzhandbuch unter Geben Sie an AWS-Region, welche AWS-Regionen Konten verwendet werden können.
- Für Quell- und Ziel-Buckets muss die Versioning aktiviert sein. Weitere Informationen über das Versioning finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.
- Amazon S3 muss berechtigt sein, Objekte aus dem Quell-Bucket in Ihrem Namen in den Ziel-Bucket/die Ziel-Buckets zu replizieren. Weitere Informationen zu diesen Berechtigungen finden Sie unter Berechtigungen für die Live-Replikation einrichten.
- Wenn der Besitzer des Quell-Buckets nicht Eigentümer des Objekts im Bucket ist, muss der Objekteigentümer dem Bucket-Besitzer `READ` die `READ_ACP` entsprechenden Berechtigungen mit der Objektzugriffskontrollliste (ACL) gewähren. Weitere Informationen finden Sie unter Übersicht über die Zugriffskontrollliste (ACL).
- Wenn für den Quell-Bucket die S3-Objektsperre aktiviert ist, muss sie für die Ziel-Buckets ebenfalls aktiviert sein.

Um die Replikation für einen Bucket zu aktivieren, für den die Objektsperre aktiviert ist, müssen Sie AWS Command Line Interface RESTAPI, oder verwenden AWS SDKs. Weitere allgemeine Informationen finden Sie unter Sperren von Objekten mit Object Lock.

> **Note**
>
> Sie müssen zwei neue Berechtigungen für den S3-Quell-Bucket in der Rolle AWS Identity and Access Management (IAM) gewähren, mit der Sie die Replikation einrichten. Die zwei neuen Berechtigungen lauten `s3:GetObjectRetention` und `s3:GetObjectLegalHold`. Wenn die Rolle über eine `s3:Get*`-Berechtigung verfügt, ist die Anforderung erfüllt. Weitere Informationen finden Sie unter Berechtigungen für die Live-Replikation einrichten.

Weitere Informationen finden Sie unter Überblick über die Einrichtung der Live-Replikation.

Wenn Sie die Replikationskonfiguration in einem kontoübergreifenden Szenario festlegen, in dem sich die Quell- und Ziel-Buckets im Besitz von verschiedenen AWS-Konten befinden, gilt die folgenden zusätzliche Anforderung:

- Der Eigentümer der Ziel-Buckets muss dem Eigentümer des Quell-Buckets mit einer Bucket-Richtlinie die Berechtigungen zum Replizieren von Objekten erteilen. Weitere Informationen finden Sie unter Erteilen von Berechtigungen, wenn die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten.
- Die Ziel-Buckets dürfen nicht als Buckets des Typs "Zahlung durch den Anforderer" konfiguriert sein. Weitere Informationen finden Sie unter Nutzen von Buckets mit Zahlung durch den Anforderer für Speicherübertragungen und Nutzung.

Überlegungen zur Replikation

Bevor Sie eine Replikationskonfiguration erstellen, sollten Sie die folgenden Überlegungen beachten.

Themen

- Lebenszyklus-Konfiguration und Objektreplikate
- Versioning-Konfiguration und Replikations-Konfiguration
- Verwenden der S3-Replikation mit S3-Intelligent-Tiering

- Protokollierungskonfiguration und Replikations-Konfiguration
- CRRund die Zielregion
- S3-Batch-Replikation
- Steuerung der S3-Replikationszeit

Lebenszyklus-Konfiguration und Objektreplikate

Die von Amazon S3 für die Replikation eines Objekts benötigte Zeit hängt von der Größe des Objekts ab. Bei großen Objekten kann dies mehrere Stunden dauern. Auch wenn es möglicherweise etwas dauert, bis ein Replikat im Ziel verfügbar ist, entspricht die Erstellungszeit des Replikats der Erstellungszeit des entsprechenden Objekts im Quell-Bucket. Wenn eine Lebenszyklus-Konfiguration für einen Ziel-Bucket aktiviert ist, berücksichtigen die Lebenszyklusregeln die ursprüngliche Erstellungszeit des Objekts, nicht den Zeitpunkt, zu dem das Replikat im Ziel-Bucket verfügbar wurde.

Die Replikations-Konfiguration erfordert einen Bucket mit aktiviertem Versioning. Berücksichtigen Sie bei der Aktivierung des Versioning in einem Bucket Folgendes:

- Wenn eine Lebenszyklus-Konfiguration für den Ablauf von Objekten vorhanden ist, nachdem Sie die Versionsverwaltung aktiviert haben, fügen Sie eine `NonCurrentVersionExpiration`-Richtlinie hinzu, damit dasselbe Verhalten hinsichtlich einer dauerhaften Löschung beibehalten wird wie vor der Aktivierung der Versionsverwaltung.
- Wenn eine Übergangs-Lebenszyklus-Konfiguration vorhanden ist, nachdem Sie die Versionsverwaltung aktiviert haben, sollten Sie in Betracht ziehen, eine `NonCurrentVersionTransition`-Richtlinie hinzuzufügen.

Versioning-Konfiguration und Replikations-Konfiguration

Wenn Sie die Replikation auf einem Bucket konfigurieren, muss sowohl für den Quell- als auch für den Ziel-Bucket das Versioning aktiviert sein. Nachdem Sie sowohl im Quell- als auch im Ziel-Bucket das Versioning aktiviert und die Replikation im Quell-Bucket konfiguriert haben, werden Sie folgende Schwierigkeiten beobachten:

- Wenn Sie versuchen, das Versioning auf dem Quell-Bucket zu deaktivieren, gibt Amazon S3 einen Fehler zurück. Sie müssen die Replikations-Konfiguration entfernen, damit Sie das Versioning auf dem Quell-Bucket deaktivieren können.

- Wenn Sie das Versioning auf dem Ziel-Bucket deaktivieren, schlägt die Replikation fehl. Das Quellobjekt weist den Replikationsstatus `FAILED` auf.

Verwenden der S3-Replikation mit S3-Intelligent-Tiering

S3-Intelligent-Tiering ist eine Speicherklasse, die entworfen wurde, um die Speicherkosten zu optimieren, indem sie Daten automatisch zur kostengünstigsten Zugriffsstufe verschiebt. Gegen eine geringe monatliche Gebühr für Objektüberwachung und Automatisierung überwacht S3 Intelligent-Tiering die Zugriffsmuster und verschiebt die Objekte, auf die nicht zugegriffen wurde, automatisch in kostengünstigere Zugriffsebenen.

Replizieren von in S3 Intelligent-Tiering gespeicherten Objekten mit S3 Batch Replication oder Aufrufen CopyObject oder UploadPartCopystellt Zugriff dar. In diesen Fällen werden die Quellobjekte der Kopier- oder Replikationsvorgänge hochgestuft.

Weitere Informationen zu S3-Intelligent-Tiering finden Sie unter Verwaltung der Speicherkosten mit Amazon S3 Intelligent-Tiering.

Protokollierungskonfiguration und Replikations-Konfiguration

Wenn Amazon S3 Protokolle an einen Bucket übermittelt, für den die Replikation aktiviert ist, repliziert der Service die Protokollobjekte.

Wenn Serverzugriffsprotokolle oder AWS CloudTrail -protokolle in Ihrem Quell- oder Ziel-Bucket aktiviert sind, nimmt Amazon S3 replikationsbezogene Anfragen in die Protokolle auf. Beispiel: Amazon S3 protokolliert jedes Objekt, das es repliziert.

CRRund die Zielregion

Amazon S3 Cross-Region Replication (CRR) wird verwendet, um Objekte zwischen verschiedenen S3-Buckets zu kopieren. AWS-Regionen Sie können die Region für Ihren Ziel-Bucket entweder auf der Basis Ihrer geschäftlichen Anforderungen oder von Kostenaspekten auswählen. Beispielsweise variieren die Gebühren für eine regionsübergreifende Datenübertragung je nach gewählten Regionen.

Angenommen, Sie haben USA Ost (Nord-Virginia) (`us-east-1`) als Region für Ihren Quell-Bucket ausgewählt. Wenn Sie USA West (Oregon) (`us-west-2`) als Region für Ihre Ziel-Buckets wählen, zahlen Sie mehr als wenn Sie die Region USA Ost (Ohio) (`us-east-2`) wählen. Weitere Informationen zu Preisen finden Sie im Abschnitt „Datenübertragungspreise“ unter Amazon S3 – Preise.

Im Zusammenhang mit der Replikation in derselben Region () fallen keine Datenübertragungsgebühren an. SRR

S3-Batch-Replikation

Informationen zu Überlegungen zur Batch-Replikation finden Sie unterÜberlegungen zur S3-Batch-Replikation.

Steuerung der S3-Replikationszeit

Informationen zu bewährten Methoden und Überlegungen zu S3 Replication Time Control (S3RTC) finden Sie unterBewährte Methoden und Richtlinien für S3 RTC.

Überblick über die Einrichtung der Live-Replikation

> **Note**
>
> Objekte, die vor dem Einrichten der Replikation vorhanden waren, werden nicht automatisch repliziert. Anders ausgedrückt: Amazon S3 repliziert Objekte nicht rückwirkend. Verwenden Sie die S3-Batch-Replikation, um Objekte zu replizieren, die vor Ihrer Replikationskonfiguration erstellt wurden. Weitere Informationen zur Konfiguration der Batch-Replikation finden Sie unterReplizieren vorhandener Objekte.

Um die Live-Replikation — Replikation in derselben Region (SRR) oder Regionsübergreifende Replikation () — zu aktivieren, fügen Sie Ihrem CRR Quell-Bucket eine Replikationskonfiguration hinzu. Diese Konfiguration weist Amazon S3 an, Objekte wie angegeben zu replizieren. In der Replikations-Konfiguration müssen Sie Folgendes angeben:

- Die Ziel-Buckets – Der Bucket oder die Buckets, in den/die Amazon S3 die Objekte replizieren soll.
- Die Objekte, die Sie replizieren möchten — Sie können alle Objekte im Quell-Bucket oder eine Teilmenge von Objekten replizieren. Teilmengen identifizieren Sie, indem Sie ein Schlüsselnamenpräfix, mindestens ein Objekt-Tag oder beides in der Konfiguration angeben.

  Wenn Sie beispielsweise eine Replikationsregel konfigurieren, um nur Objekte mit dem Schlüsselnamenpräfix `Tax/` zu replizieren, repliziert Amazon S3 Objekte mit Schlüsseln wie `Tax/doc1` oder `Tax/doc2`. Es repliziert aber keine Objekte mit dem Schlüssel `Legal/doc3`. Wenn Sie sowohl ein Präfix als auch mindestens ein Tag angeben, repliziert Amazon S3 nur Objekte, die dieses Schlüsselpräfix und diese Tags aufweisen.

- Eine AWS Identity and Access Management (IAM) Rolle — Amazon S3 übernimmt diese IAM Rolle, um Objekte in Ihrem Namen zu replizieren. Weitere Informationen zur Erstellung dieser IAM Rolle und zur Verwaltung von Berechtigungen finden Sie unterBerechtigungen für die Live-Replikation einrichten.

Zusätzlich zu diesen Mindestanforderungen können Sie die folgenden Optionen auswählen:

- Replikat-Speicherklasse – Standardmäßig speichert Amazon S3 Objektreplikate in derselben Speicherklasse wie das Quellobjekt. Sie können eine andere Speicherklasse für die Replikate festlegen.
- Replikat-Eigentümerschaft – Amazon S3 geht davon aus, dass ein Objektreplikat weiterhin das Eigentum des Eigentümers des Quellobjekts bleibt. Wenn sie also Objekte repliziert, repliziert sie auch die entsprechende Objektzugriffskontrollliste (ACL) oder die Einstellung für S3-Objekteigentümer. Wenn sich der Quell- und der Ziel-Bucket im Besitz verschiedener AWS-Konten befinden, können Sie die Replikation so konfigurieren, dass der Besitzer eines Replikats auf das AWS-Konto geändert wird, dem der Ziel-Bucket gehört. Weitere Informationen finden Sie unter the section called “Ändern des Replikat-Eigentümers”.

Sie können die Replikation mithilfe der Amazon S3-Konsole AWS Command Line Interface (AWS CLI) oder Amazon S3 konfigurieren RESTAPI. AWS SDKs Eine ausführliche Anleitung zur Einrichtung der Replikation finden Sie unter. the section called “Replikations-Walkthroughs”

Amazon S3 bietet REST API Funktionen zur Unterstützung der Einrichtung von Replikationsregeln. Weitere Informationen finden Sie unter den folgenden Themen in der Amazon Simple Storage Service API Reference:

- PutBucketReplication
- GetBucketReplication
- DeleteBucketReplication

Themen

- Elemente der Replikationskonfigurationsdatei
- Berechtigungen für die Live-Replikation einrichten
- Beispiele für die Konfiguration der Live-Replikation

Elemente der Replikationskonfigurationsdatei

Amazon S3 speichert eine Replikationskonfiguration alsXML. Wenn Sie die Replikation programmgesteuert über Amazon S3 konfigurieren RESTAPI, geben Sie die verschiedenen Elemente Ihrer Replikationskonfiguration in dieser XML Datei an. Wenn Sie die Replikation über AWS Command Line Interface (AWS CLI) konfigurieren, geben Sie Ihre Replikationskonfiguration JSON im Format an. JSONBeispiele finden Sie in den exemplarischen Vorgehensweisen unter. the section called “Replikations-Walkthroughs”

> **Note**
>
> Die neueste Version des XML Replikationskonfigurationsformats ist V2. XMLV2-Replikationskonfigurationen enthalten das `<Filter>` Element für Regeln und Regeln, die S3 Replication Time Control (S3RTC) spezifizieren.
> Um Ihre Version der Replikationskonfiguration zu sehen, können Sie den `GetBucketReplication` API Vorgang verwenden. Weitere Informationen finden Sie unter GetBucketReplicationin der Amazon Simple Storage Service API Reference.
> Aus Gründen der Abwärtskompatibilität unterstützt Amazon S3 weiterhin das XML V1-Replikationskonfigurationsformat. Wenn Sie das XML V1-Replikationskonfigurationsformat verwendet haben, finden Sie Hinweise Überlegungen zur Abwärtskompatibilität zur Abwärtskompatibilität unter.

In der XML Replikationskonfigurationsdatei müssen Sie eine Rolle AWS Identity and Access Management (IAM) und eine oder mehrere Regeln angeben, wie im folgenden Beispiel gezeigt:

```
<ReplicationConfiguration>
    <Role>IAM-role-ARN</Role>
    <Rule>
        ...
    </Rule>
    <Rule>
         ...
    </Rule>
     ...
</ReplicationConfiguration>
```

Amazon S3 kann ohne Berechtigung durch Sie keine Objekte replizieren. Sie gewähren Amazon S3 Berechtigungen mit der IAM Rolle, die Sie in der Replikationskonfiguration angeben. Amazon

S3 übernimmt diese IAM Rolle, um Objekte in Ihrem Namen zu replizieren. Sie müssen der IAM Rolle zuerst die erforderlichen Berechtigungen erteilen. Weitere Informationen zum Verwalten von Berechtigungen finden Sie unter Berechtigungen für die Live-Replikation einrichten.

In den folgenden Szenarien fügen Sie einer Replikationskonfiguration nur eine Regel hinzu:

- Sie möchten alle Objekte replizieren.
- Sie möchten nur eine Teilmenge von Objekten replizieren. Sie identifizieren die Teilmenge der Objekte, indem Sie einen Filter zur Regel hinzufügen. Im Filter geben Sie ein Objektschlüsselpräfix, Tags oder eine Kombination aus beidem an, um die Teilmenge der Objekte zu identifizieren, für die die Regel gilt. Die Filter zielen auf Objekte ab, die genau den von Ihnen angegebenen Werten entsprechen.

Wenn Sie verschiedene Teilmengen von Objekten replizieren möchten, fügen Sie einer Replikationskonfiguration mehrere Regeln hinzu. In jeder Regel geben Sie einen Filter an, der eine andere Teilmenge auswählt. Beispiel: Sie möchten Objekte mit dem Schlüsselpräfix `tax/` oder `document/` replizieren. Dazu fügen Sie zwei Regeln hinzu, eine, die den `tax/`-Schlüsselpräfix-Filter angibt und eine andere, die das `document/`-Schlüsselpräfix angibt. Weitere Hinweise zu Objektschlüsselpräfixen finden Sie unter. Organisieren von Objekten mit Präfixen

In den folgenden Abschnitten finden Sie zusätzliche Informationen.

Themen

- Basisregelkonfiguration
- Optional: Festlegen eines Filters
- Zusätzliche Zielkonfigurationen
- Beispiele für Replikations-Konfigurationen
- Überlegungen zur Abwärtskompatibilität

Basisregelkonfiguration

Jede Regel muss den Status und die Priorität der Regel enthalten. Die Regel muss auch angeben, ob Löschmarkierungen repliziert werden sollen.

- Das `<Status>` Element gibt anhand der Werte oder an, ob die Regel aktiviert `Enabled` oder `Disabled` deaktiviert ist. Wenn eine Regel deaktiviert ist, führt Amazon S3 die in der Regel angegebenen Aktionen nicht durch.

- Das `<Priority>` Element gibt an, welche Regel Vorrang hat, wenn zwei oder mehr Replikationsregeln kollidieren. Amazon S3 versucht, Objekte gemäß allen Replikationsregeln zu replizieren. Wenn es jedoch zwei oder mehr Regeln mit demselben Ziel-Bucket gibt, werden Objekte gemäß der Regel mit der höchsten Priorität repliziert. Je höher die Zahl, desto höher die Priorität.
- Das `<DeleteMarkerReplication>` Element gibt an, ob Löschmarkierungen mithilfe der Werte `Enabled` oder repliziert werden sollen. `Disabled`

In der `<Destination>` Elementkonfiguration müssen Sie den Namen des Ziel-Buckets oder der Ziel-Buckets angeben, in den Amazon S3 Objekte replizieren soll.

Das folgende Beispiel zeigt die Mindestanforderungen für eine V2-Regel. Aus Gründen der Abwärtskompatibilität unterstützt Amazon S3 weiterhin das XML V1-Format. Weitere Informationen finden Sie unter Überlegungen zur Abwärtskompatibilität.

```
...
    <Rule>
        <ID>Rule-1</ID>
        <Status>Enabled-or-Disabled</Status>
        <Filter>
            <Prefix></Prefix>
        </Filter>
        <Priority>integer</Priority>
        <DeleteMarkerReplication>
           <Status>Enabled-or-Disabled</Status>
        </DeleteMarkerReplication>
        <Destination>
           <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
        </Destination>
    </Rule>
    <Rule>
         ...
    </Rule>
     ...
...
```

Sie können auch andere Konfigurationsoptionen festlegen. Beispiel: Sie möchten für Objektreplikate eine andere Speicherklasse verwenden als für das Quellobjekt.

Optional: Festlegen eines Filters

Um eine Teilmenge von Objekten auszuwählen, für die die Regel gilt, fügen Sie einen optionalen Filter hinzu. Sie können nach Objekt-Schlüsselpräfix, Objekt-Tags oder einer Kombination aus beidem filtern. Wenn Sie sowohl nach Schlüsselpräfix als auch nach Objekt-Tags filtern, kombiniert Amazon S3 die Filter mit einem logischen AND-Operator. Anders ausgedrückt: Die Regel wird auf eine Teilmenge von Objekten sowohl mit einem bestimmten Schlüsselpräfix als auch mit bestimmten Tags angewendet.

Filter basierend auf Objektschlüsselpräfix

Verwenden Sie Folgendes, um eine Regel mit einem Filter anzugeben, der auf einem Objektschlüsselpräfix basiertXML. Sie können nur ein Präfix pro Regel angeben.

```
<Rule>
    ...
    <Filter>
        <Prefix>key-prefix</Prefix>
    </Filter>
    ...
</Rule>
...
```

Filter basierend auf Objekt-Markierungen

Gehen Sie wie folgt vor, um eine Regel mit einem auf Objekt-Tags basierenden Filter anzugebenXML. Sie können einen oder mehrere Objekt-Markierungen festlegen.

```
<Rule>
    ...
    <Filter>
        <And>
            <Tag>
                <Key>key1</Key>
                <Value>value1</Value>
            </Tag>
            <Tag>
                <Key>key2</Key>
                <Value>value2</Value>
            </Tag>
             ...
```

```
        </And>
    </Filter>
    ...
</Rule>
...
```

Filtern mit einem Schlüsselpräfix und Objekt-Markierungen

Verwenden Sie Folgendes, um einen Regelfilter mit einer Kombination aus einem key prefix und Objekt-Tags anzugebenXML. Sie wickeln diese Filter in ein übergeordnetes <And>-Element ein. Amazon S3 führt eine logische AND-Operation aus, um diese Filter zu kombinieren. Anders ausgedrückt: Die Regel wird auf eine Teilmenge von Objekten sowohl mit einem bestimmten Schlüsselpräfix als auch mit bestimmten Tags angewendet.

```
<Rule>
    ...
    <Filter>
        <And>
            <Prefix>key-prefix</Prefix>
            <Tag>
                <Key>key1</Key>
                <Value>value1</Value>
            </Tag>
            <Tag>
                <Key>key2</Key>
                <Value>value2</Value>
            </Tag>
             ...
    </Filter>
    ...
</Rule>
...
```

> **Note**
>
> - Wenn Sie eine Regel mit einem leeren <Filter> Element angeben, gilt Ihre Regel für alle Objekte in Ihrem Bucket.
> - Wenn Sie tagbasierte Replikationsregeln mit Live-Replikation verwenden, müssen neue Objekte während des PutObject Vorgangs mit dem entsprechenden Tag für die Replikationsregel gekennzeichnet werden. Andernfalls werden die Objekte nicht repliziert.

Wenn Objekte nach dem `PutObject` Vorgang markiert werden, werden diese Objekte ebenfalls nicht repliziert.

Um Objekte zu replizieren, die nach dem `PutObject` Vorgang markiert wurden, müssen Sie S3 Batch Replication verwenden. Weitere Informationen zur Batch-Replikationen finden Sie unter Replizieren vorhandener Objekte.

Zusätzliche Zielkonfigurationen

In der Zielkonfiguration geben Sie den/die Bucket(s) an, in den/die Amazon S3 Objekte replizieren soll. Sie können Konfigurationen einrichten, um Objekte aus einem Quell-Bucket in einen oder mehrere Ziel-Buckets zu replizieren.

```
...
<Destination>
    <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
</Destination>
...
```

Sie können die folgenden Optionen im Element `<Destination>` hinzufügen.

Themen

- Festlegen der Speicherklasse
- Hinzufügen mehrerer Ziel-Buckets
- Angeben verschiedener Parameter für jede Replikationsregel mit mehreren Ziel-Buckets
- Ändern der Replikat-Eigentümerschaft
- Aktivieren der S3 Replication Time Control
- Replizieren Sie Objekte, die mit serverseitiger Verschlüsselung erstellt wurden, mithilfe von AWS KMS

Festlegen der Speicherklasse

Sie können die Speicherklasse für die Objektreplikate festlegen. Standardmäßig verwendet Amazon S3 wie im folgenden Beispiel die Speicherklasse des Quellobjekts zum Erstellen von Objektreplikaten.

```
...
```

```
<Destination>
       <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
       <StorageClass>storage-class</StorageClass>
</Destination>
...
```

Hinzufügen mehrerer Ziel-Buckets

Sie können wie folgt mehrere Ziel-Buckets in einer einzigen Replikations-Konfiguration hinzufügen.

```
...
<Rule>
    <ID>Rule-1</ID>
    <Status>Enabled-or-Disabled</Status>
    <Priority>integer</Priority>
    <DeleteMarkerReplication>
       <Status>Enabled-or-Disabled</Status>
    </DeleteMarkerReplication>
    <Destination>
       <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket1</Bucket>
    </Destination>
</Rule>
<Rule>
    <ID>Rule-2</ID>
    <Status>Enabled-or-Disabled</Status>
    <Priority>integer</Priority>
    <DeleteMarkerReplication>
       <Status>Enabled-or-Disabled</Status>
    </DeleteMarkerReplication>
    <Destination>
       <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket2</Bucket>
    </Destination>
</Rule>
...
```

Angeben verschiedener Parameter für jede Replikationsregel mit mehreren Ziel-Buckets

Wenn Sie mehrere Ziel-Buckets in einer einzigen Replikations-Konfiguration hinzufügen, können Sie wie folgt verschiedene Parameter für jede Replizierungsregel angeben.

```
...
<Rule>
    <ID>Rule-1</ID>
```

```
    <Status>Enabled-or-Disabled</Status>
    <Priority>integer</Priority>
    <DeleteMarkerReplication>
       <Status>Disabled</Status>
    </DeleteMarkerReplication>
      <Metrics>
    <Status>Enabled</Status>
    <EventThreshold>
      <Minutes>15</Minutes>
    </EventThreshold>
  </Metrics>
    <Destination>
       <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket1</Bucket>
    </Destination>
</Rule>
<Rule>
    <ID>Rule-2</ID>
    <Status>Enabled-or-Disabled</Status>
    <Priority>integer</Priority>
    <DeleteMarkerReplication>
       <Status>Enabled</Status>
    </DeleteMarkerReplication>
      <Metrics>
    <Status>Enabled</Status>
    <EventThreshold>
      <Minutes>15</Minutes>
    </EventThreshold>
  </Metrics>
  <ReplicationTime>
    <Status>Enabled</Status>
    <Time>
      <Minutes>15</Minutes>
    </Time>
  </ReplicationTime>
    <Destination>
       <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket2</Bucket>
    </Destination>
</Rule>
...
```

Ändern der Replikat-Eigentümerschaft

Wenn die Quell- und Ziel-Buckets nicht denselben Konten gehören, können Sie den Besitz des Replikats auf das Konto ändern, dem der AWS-Konto Ziel-Bucket gehört. Fügen Sie dazu das `<AccessControlTranslation>`-Element hinzu. Dieses Element übernimmt den Wert `Destination`.

```
...
<Destination>
   <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
   <Account>destination-bucket-owner-account-id</Account>
   <AccessControlTranslation>
       <Owner>Destination</Owner>
   </AccessControlTranslation>
</Destination>
...
```

Wenn Sie das `<AccessControlTranslation>` Element nicht zur Replikationskonfiguration hinzufügen, gehören die Replikate demjenigen, der das AWS-Konto Quellobjekt besitzt. Weitere Informationen finden Sie unter Ändern des Replikat-Eigentümers.

Aktivieren der S3 Replication Time Control

Sie können S3 Replication Time Control (S3RTC) in Ihrer Replikationskonfiguration aktivieren. S3 RTC repliziert die meisten Objekte innerhalb von Sekunden und 99,99 Prozent der Objekte innerhalb von 15 Minuten (unterstützt durch ein Service Level Agreement).

> **Note**
>
> Für die Elemente und `<Minutes>15</Minutes>` wird nur der Wert von akzeptiert. `<EventThreshold> <Time>`

```
...
<Destination>
  <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
  <Metrics>
    <Status>Enabled</Status>
    <EventThreshold>
      <Minutes>15</Minutes>
```

```
    </EventThreshold>
  </Metrics>
  <ReplicationTime>
    <Status>Enabled</Status>
    <Time>
      <Minutes>15</Minutes>
    </Time>
  </ReplicationTime>
</Destination>
...
```

Weitere Informationen finden Sie unter Erfüllung von Compliance-Anforderungen mit S3 Replication Time Control. APIBeispiele finden Sie unter PutBucketReplicationin der Amazon Simple Storage Service API Reference.

Replizieren Sie Objekte, die mit serverseitiger Verschlüsselung erstellt wurden, mithilfe von AWS KMS

Ihr Quell-Bucket enthält möglicherweise Objekte, die mit serverseitiger Verschlüsselung mithilfe von () -Schlüsseln AWS Key Management Service (SSE-AWS KMS) erstellt wurden. KMS Standardmäßig repliziert Amazon S3 diese Objekte nicht. Optional können Sie Amazon S3 zum Replizieren dieser Objekte anweisen. Um dies zu tun, entscheiden Sie sich zunächst explizit für diese Funktion, indem Sie das `<SourceSelectionCriteria>`-Element hinzufügen. Geben Sie dann den AWS KMS key (für den Ziel-Bucket) an, AWS-Region der zum Verschlüsseln von Objektreplikaten verwendet werden soll. Das folgende Beispiel zeigt, wie diese Elemente angegeben werden.

```
...
<SourceSelectionCriteria>
  <SseKmsEncryptedObjects>
    <Status>Enabled</Status>
  </SseKmsEncryptedObjects>
</SourceSelectionCriteria>
<Destination>
  <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
  <EncryptionConfiguration>
    <ReplicaKmsKeyID>AWS KMS key ID to use for encrypting object replicas</
ReplicaKmsKeyID>
  </EncryptionConfiguration>
</Destination>
...
```

Weitere Informationen finden Sie unter Replikation verschlüsselter Objekte (SSE-S3, SSE -, DSSE - KMSKMS, SSE -C).

Beispiele für Replikations-Konfigurationen

Fügen Sie zunächst die folgenden Beispiel-Replikations-Konfigurationen zu Ihrem Bucket hinzu.

> **Important**
>
> Um eine Replikationskonfiguration zu einem Bucket hinzuzufügen, benötigen Sie die `iam:PassRole`-Berechtigung. Mit dieser Berechtigung können Sie die IAM Rolle weitergeben, die Amazon S3 S3-Replikationsberechtigungen gewährt. Sie geben die IAM Rolle an, indem Sie den Amazon-Ressourcennamen (ARN) angeben, der in dem `<Role>` Element in der Replikationskonfiguration verwendet wirdXML. Weitere Informationen finden Sie unter Erteilen von Benutzerberechtigungen zur Übergabe einer Rolle an einen AWS-Service im IAMBenutzerhandbuch.

Example 1: Replikations-Konfiguration mit einer Regel

Die folgende grundlegende Replikations-Konfiguration legt eine Regel fest. Die Regel spezifiziert eine IAM Rolle, die Amazon S3 übernehmen kann, und einen einzelnen Ziel-Bucket für Objektreplikate. Der `<Status>` Elementwert von `Enabled` gibt an, dass die Regel in Kraft ist.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>

    <Destination>
      <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Um eine Teilmenge von Objekten für die Replikation auszuwählen, können Sie einen Filter hinzufügen. In der folgenden Konfiguration gibt der Filter ein Objektschlüsselpräfix an. Diese Regel gilt für Objekte mit dem Schlüsselnamenpräfix *Tax/*.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
    </DeleteMarkerReplication>

    <Filter>
       <Prefix>Tax/</Prefix>
    </Filter>

    <Destination>
       <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
    </Destination>

  </Rule>
</ReplicationConfiguration>
```

Wenn Sie das <Filter>-Element angeben, müssen Sie auch die Elemente <Priority> und <DeleteMarkerReplication> einschließen. In diesem Beispiel ist der Wert, den Sie für das <Priority> Element festlegen, irrelevant, da es nur eine Regel gibt.

In der folgenden Konfiguration gibt der Filter ein Präfix und zwei Markierungen an. Die Regel gilt für eine Untermenge der Objekte, die das angegebene Schlüsselpräfix und die angegebenen Markierungen aufweisen. Insbesondere gilt sie für Objekte, deren Schlüsselnamen das *Tax/* Präfix enthalten, und für die beiden angegebenen Objekt-Tags. In diesem Beispiel ist der Wert, den Sie für das <Priority> Element festlegen, irrelevant, da es nur eine Regel gibt.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
    </DeleteMarkerReplication>

    <Filter>
        <And>
```

```
          <Prefix>Tax/</Prefix>
          <Tag>
             <Tag>
                <Key>tagA</Key>
                <Value>valueA</Value>
             </Tag>
          </Tag>
          <Tag>
             <Tag>
                <Key>tagB</Key>
                <Value>valueB</Value>
             </Tag>
          </Tag>
       </And>

    </Filter>

    <Destination>
        <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
    </Destination>

  </Rule>
</ReplicationConfiguration>
```

Sie können eine Speicherklasse für die Objektreplikate wie folgt angeben:

```
<?xml version="1.0" encoding="UTF-8"?>

<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Destination>
       <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
       <StorageClass>storage-class</StorageClass>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Sie können jede Speicherklasse festlegen, die Amazon S3 unterstützt.

Example 2: Replikations-Konfiguration mit zwei Regeln

Example

In der folgenden Replikationskonfiguration geben die Regeln Folgendes an:

- Jede Regel filtert nach einem anderen Schlüsselpräfix, sodass jede Regel auf eine bestimmte Teilmenge von Objekten angewendet wird. In diesem Beispiel repliziert Amazon S3 Objekte mit den Schlüsselnamen *Tax/doc1.pdf* and *Project/project1.txt*, aber es repliziert keine Objekte mit dem Schlüsselnamen *PersonalDoc/DocumentA*.
- Obwohl beide Regeln einen Wert für das <Priority> Element angeben, ist die Regelpriorität irrelevant, da die Regeln für zwei unterschiedliche Gruppen von Objekten gelten. Das nächste Beispiel zeigt, was geschieht, wenn die Regelpriorität angewendet wird.
- Die zweite Regel gibt eine Speicherklasse S3 Standard-IA für Objektreplikate an. Amazon S3 verwendet die angegebene Speicherklasse für diese Objektreplikate.

```
<?xml version="1.0" encoding="UTF-8"?>

<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
    </DeleteMarkerReplication>
    <Filter>
        <Prefix>Tax</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Destination>
      <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
    </Destination>
     ...
  </Rule>
 <Rule>
    <Status>Enabled</Status>
    <Priority>2</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
```

```
    </DeleteMarkerReplication>
    <Filter>
        <Prefix>Project</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Destination>
      <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
     <StorageClass>STANDARD_IA</StorageClass>
    </Destination>
     ...
  </Rule>


</ReplicationConfiguration>
```

Example 3: Replikations-Konfiguration mit zwei Regeln mit einander überlappenden Präfixen

In dieser Konfiguration spezifizieren die beiden Regeln Filter mit überlappenden Schlüsselpräfixen, *Stern/* and *Raumschiff/*Beide Regeln gelten für Objekte mit dem Schlüsselnamen . *Raumschiff-X*. In diesem Fall bestimmt Amazon S3 anhand der Regelpriorität, welche Regel angewendet werden soll. Je höher die Zahl, desto höher die Priorität.

```
<ReplicationConfiguration>

  <Role>arn:aws:iam::account-id:role/role-name</Role>

  <Rule>
    <Status>Enabled</Status>
    <Priority>1</Priority>
    <DeleteMarkerReplication>
       <Status>string</Status>
    </DeleteMarkerReplication>
    <Filter>
        <Prefix>star</Prefix>
    </Filter>
    <Destination>
      <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
    </Destination>
  </Rule>
  <Rule>
    <Status>Enabled</Status>
    <Priority>2</Priority>
    <DeleteMarkerReplication>
```

```
        <Status>string</Status>
    </DeleteMarkerReplication>
    <Filter>
        <Prefix>starship</Prefix>
    </Filter>
    <Destination>
      <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Example 4. Beispielhafte Walkthroughs

Beispielanleitungen finden Sie unter Beispiele für die Konfiguration der Live-Replikation.

Weitere Informationen zur XML Struktur der Replikationskonfiguration finden Sie PutBucketReplicationin der Amazon Simple Storage Service API Reference.

Überlegungen zur Abwärtskompatibilität

Die neueste Version des XML Replikationskonfigurationsformats ist V2. XMLV2-Replikationskonfigurationen enthalten das `<Filter>` Element für Regeln und Regeln, die S3 Replication Time Control (S3RTC) spezifizieren.

Um Ihre Version der Replikationskonfiguration zu sehen, können Sie den `GetBucketReplication` API Vorgang verwenden. Weitere Informationen finden Sie unter GetBucketReplicationin der Amazon Simple Storage Service API Reference.

Aus Gründen der Abwärtskompatibilität unterstützt Amazon S3 weiterhin das XML V1-Replikationskonfigurationsformat. Wenn Sie das XML V1-Replikationskonfigurationsformat verwendet haben, sollten Sie die folgenden Probleme berücksichtigen, die sich auf die Abwärtskompatibilität auswirken:

- Das XML V2-Format der Replikationskonfiguration enthält das `<Filter>` Element für Regeln. Mit dem `<Filter>`-Element können Sie Objektfilter basierend auf dem Objektschlüsselpräfix, Tags oder einer Kombination aus beidem angeben, um die Objekte festzulegen, für die die Regel gilt. Das XML V1-Format der Replikationskonfiguration unterstützt das Filtern nur auf der Grundlage des key prefix. In diesem Fall fügen Sie das `<Prefix>` Element direkt als untergeordnetes Element des `<Rule>` Elements hinzu, wie im folgenden Beispiel:

  ```
  <?xml version="1.0" encoding="UTF-8"?>
  <ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  ```

```
  <Role>arn:aws:iam::account-id:role/role-name</Role>
  <Rule>
    <Status>Enabled</Status>
    <Prefix>key-prefix</Prefix>
    <Destination>
       <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
    </Destination>

  </Rule>
</ReplicationConfiguration>
```

- Wenn Sie ein Objekt aus Ihrem Quell-Bucket löschen, ohne eine Objektversions-ID anzugeben, fügt Amazon S3 eine Löschmarkierung hinzu. Wenn Sie das XML V1-Format für die Replikationskonfiguration verwenden, repliziert Amazon S3 nur Löschmarkierungen, die sich aus Benutzeraktionen ergeben. Anders ausgedrückt: Amazon S3 repliziert die Löschmarkierung nur, wenn ein Benutzer ein Objekt löscht. Wenn ein abgelaufenes Objekt von Amazon S3 entfernt wird (als Teil einer Lebenszyklusaktion), repliziert Amazon S3 die Löschmarkierung nicht.

  Im XML V2-Format der Replikationskonfiguration können Sie die Replikation von Löschmarkierungen für non-tag-based Regeln aktivieren. Weitere Informationen finden Sie unter Replizieren von Löschmarkierungen auf Buckets.

Berechtigungen für die Live-Replikation einrichten

Wenn Sie die Live-Replikation in Amazon S3 einrichten, müssen Sie die erforderlichen Berechtigungen wie folgt erwerben:

- Amazon S3 benötigt Berechtigungen, um Objekte in Ihrem Namen zu replizieren. Sie gewähren diese Berechtigungen, indem Sie eine Rolle AWS Identity and Access Management (IAM) erstellen und diese Rolle dann in Ihrer Replikationskonfiguration angeben.
- Wenn die Quell- und Ziel-Buckets nicht denselben Konten gehören, muss der Besitzer des Ziel-Buckets dem Besitzer des Quell-Buckets auch Berechtigungen zum Speichern der Replikate gewähren.

Themen

- Eine Rolle erstellen IAM
- Erteilen von Berechtigungen, wenn die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten

- Ändern des Replikatbesitzers
- Gewähren von Berechtigungen für S3-Batch-Operationen

Eine Rolle erstellen IAM

Standardmäßig sind alle Amazon-S3-Ressourcen – Buckets, Objekte und zugehörige Unterressourcen – privat, sodass nur der Ressourcenbesitzer auf die Ressource zugreifen kann. Amazon S3 benötigt Berechtigungen zum Lesen und Replizieren von Objekten aus dem Quell-Bucket. Sie gewähren diese Berechtigungen, indem Sie eine IAM Rolle erstellen und diese Rolle in Ihrer Replikationskonfiguration angeben.

In diesem Abschnitt werden die Vertrauensrichtlinie und die Richtlinie für die Mindestberechtigungen erläutert, die dieser IAM Rolle zugeordnet sind. Die exemplarischen Vorgehensweisen enthalten step-by-step Anweisungen zum Erstellen einer IAM Rolle. Weitere Informationen finden Sie unter Beispiele für die Konfiguration der Live-Replikation.

Die Vertrauensrichtlinie legt fest, welche Hauptidentitäten die Rolle übernehmen können. IAM Die Berechtigungsrichtlinie legt fest, welche Aktionen die IAM Rolle auf welchen Ressourcen und unter welchen Bedingungen ausführen kann.

- Das folgende Beispiel zeigt eine Vertrauensrichtlinie, bei der Sie Amazon S3 als AWS-Service Principal identifizieren, der die Rolle übernehmen kann:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

- Das folgende Beispiel zeigt eine Vertrauensrichtlinie, bei der Sie Amazon S3 und S3 Batch Operations als Service Principals identifizieren, die diese Rolle übernehmen können. Verwenden Sie diesen Ansatz, wenn Sie einen Batch-Replikationsauftrag erstellen. Weitere Informationen

finden Sie unter Erstellen Sie einen Batch-Replikationsauftrag für neue Replikationsregeln oder Ziele.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service": [
              "s3.amazonaws.com",
              "batchoperations.s3.amazonaws.com"
           ]
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

Weitere Informationen zu IAM IAMRollen finden Sie im IAMBenutzerhandbuch unter Rollen.

- Das folgende Beispiel zeigt die Berechtigungsrichtlinie, mit der Sie der IAM Rolle Berechtigungen zur Ausführung von Replikationsaufgaben in Ihrem Namen gewähren. Wenn Amazon S3 die Rolle annimmt, verfügt es über die Berechtigungen, die Sie in dieser Richtlinie angeben. In dieser Richtlinie ist *amzn-s3-demo-source-bucket* der Quell-Bucket und *amzn-s3-demo-destination-bucket* der Ziel-Bucket.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
         ]
      },
      {
         "Effect":"Allow",
```

```
         "Action":[
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl",
            "s3:GetObjectVersionTagging"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ReplicateObject",
            "s3:ReplicateDelete",
            "s3:ReplicateTags"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
      }
   ]
}
```

Die Berechtigungsrichtlinie gewährt Berechtigungen für die folgenden Aktionen:

- `s3:GetReplicationConfiguration`und `s3:ListBucket` — Berechtigungen für diese Aktionen im *amzn-s3-demo-source-bucket* Bucket ermöglichen es Amazon S3, die Replikationskonfiguration abzurufen und den Bucket-Inhalt aufzulisten. (Das aktuelle Berechtigungsmodell erfordert die `s3:ListBucket`-Berechtigung für den Zugriff auf Löschmarkierungen.)
- `s3:GetObjectVersionForReplication`und `s3:GetObjectVersionAcl` — Berechtigungen für diese Aktionen werden für alle Objekte erteilt, sodass Amazon S3 eine bestimmte Objektversion und Zugriffskontrollliste (ACL) abrufen kann, die den Objekten zugeordnet sind.

- `s3:ReplicateObject` und `s3:ReplicateDelete` – Berechtigungen für diese Aktionen für alle Objekte im *amzn-s3-demo-destination-bucket*-Bucket erlauben es Amazon S3, Objekte oder Löschmarkierungen in den Ziel-Bucket zu replizieren. Informationen zu Löschmarkierungen finden Sie unter Auswirkungen von LöschVorgänge auf die Replikation.

> **Note**
>
> Die Berechtigungen für die `s3:ReplicateObject` Aktion im *amzn-s3-demo-destination-bucket* Bucket ermöglichen auch die Replikation von Metadaten wie Objekt-Tags undACLs. Daher müssen Sie für die `s3:ReplicateTags`-Aktion keine explizite Berechtigung erteilen.

- `s3:GetObjectVersionTagging`— Berechtigungen für diese Aktion für Objekte im *amzn-s3-demo-source-bucket* Bucket ermöglichen Amazon S3, Objekt-Tags für die Replikation zu lesen. Weitere Informationen über Objekt-Markierungen finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen. Wenn Amazon S3 nicht über die `s3:GetObjectVersionTagging` entsprechende Berechtigung verfügt, repliziert es die Objekte, aber nicht die Objekt-Tags.

Eine Liste der Amazon S3-Aktionen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

> **Important**
>
> AWS-Konto Derjenige, dem die IAM Rolle gehört, muss über Berechtigungen für die Aktionen verfügen, die er der IAM Rolle gewährt.
> Angenommen, der Quell-Bucket enthält beispielsweise Objekte, die im Besitz eines anderen AWS-Konto sind. Der Besitzer der Objekte muss demjenigen AWS-Konto , dem die IAM Rolle gehört, ausdrücklich die erforderlichen Berechtigungen über die Zugriffskontrolllisten der Objekte (ACLs) gewähren. Andernfalls kann Amazon S3 nicht auf die Objekte zugreifen, und die Replikation dieser Objekte schlägt fehl. Hinweise zu ACL Berechtigungen finden Sie unterÜbersicht über die Zugriffskontrollliste (ACL).
> Die hier beschriebenen Berechtigungen gehören zur Mindest-Replikationskonfiguration. Wenn Sie optionale Replikations-Konfigurationen hinzufügen möchten, müssen Sie Amazon S3 zusätzliche Berechtigungen erteilen. Wenn Sie beispielsweise verschlüsselte Objekte replizieren möchten, müssen Sie auch die erforderlichen Schlüsselberechtigungen AWS Key Management Service (AWS KMS) gewähren. Weitere Informationen finden Sie unter the section called “Replizieren verschlüsselter Objekte”.

Erteilen von Berechtigungen, wenn die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten

Wenn die Quell- und Ziel-Buckets nicht denselben Konten gehören, muss der Besitzer des Ziel-Buckets auch eine Bucket-Richtlinie hinzufügen, um dem Besitzer des Quell-Buckets Berechtigungen zur Durchführung von Replikationsaktionen zu gewähren, wie im folgenden Beispiel gezeigt. In diesem Beispiel *amzn-s3-demo-destination-bucket* ist die Richtlinie der Ziel-Bucket.

Sie können die Amazon S3 S3-Konsole auch verwenden, um diese Bucket-Richtlinie automatisch für Sie zu generieren. Weitere Informationen finden Sie unter Empfangen replizierter Objekte aus einem Quell-Bucket aktivieren.

> **Note**
>
> Das ARN Format der Rolle sieht möglicherweise anders aus. Wenn die Rolle mithilfe der Konsole erstellt wurde, lautet das ARN Format`arn:aws:iam::`*account-ID*`:role/service-role/`*role-name*. Wenn die Rolle mithilfe von erstellt wurde AWS CLI, lautet das ARN Format`arn:aws:iam::`*account-ID*`:role/`*role-name*. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter IAMRollen.

```
{
   "Version":"2012-10-17",
   "Id":"PolicyForDestinationBucket",
   "Statement":[
      {
         "Sid":"Permissions on objects",
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::source-bucket-account-ID:role/service-role/source-account-IAM-role"
         },
         "Action":[
            "s3:ReplicateDelete",
            "s3:ReplicateObject"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
      },
      {
         "Sid":"Permissions on bucket",
         "Effect":"Allow",
```

```
         "Principal":{
            "AWS":"arn:aws:iam::source-bucket-account-ID:role/service-role/source-
account-IAM-role"
         },
         "Action": [
            "s3:List*",
            "s3:GetBucketVersioning",
            "s3:PutBucketVersioning"
         ],
         "Resource":"arn:aws:s3::amzn-s3-demo-destination-bucket"
      }
   ]
}
```

Ein Beispiel finden Sie unter Konfiguration der Replikation für Buckets in verschiedenen Konten.

Wenn Objekte im Quell-Bucket mit einem Tag versehen sind, beachten Sie Folgendes:

- Wenn der Besitzer des Quell-Buckets Amazon S3 die Erlaubnis für die `s3:GetObjectVersionTagging` und `s3:ReplicateTags` Aktionen zur Replikation von Objekt-Tags (über die IAM Rolle) erteilt, repliziert Amazon S3 die Tags zusammen mit den Objekten. Informationen zur IAM Rolle finden Sie unter. Eine Rolle erstellen IAM
- Wenn der Eigentümer des Ziel-Buckets die Tags nicht replizieren will, kann er die folgende Anweisung zur Richtlinie für den Ziel-Bucket hinzufügen, um die Berechtigung für die Aktion `s3:ReplicateTags` explizit zu verweigern. In dieser Richtlinie ist *amzn-s3-demo-destination-bucket* der Ziel-Bucket.

```
...
   "Statement":[
      {
         "Effect":"Deny",
         "Principal":{
            "AWS":"arn:aws:iam::source-bucket-account-id:role/service-role/source-
account-IAM-role"
         },
         "Action":"s3:ReplicateTags",
         "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
      }
   ]
...
```

> **Note**
>
> Wenn Sie verschlüsselte Objekte replizieren möchten, müssen Sie auch die erforderlichen Schlüsselberechtigungen AWS Key Management Service (AWS KMS) gewähren. Weitere Informationen finden Sie unter the section called “Replizieren verschlüsselter Objekte”.

Aktivieren Sie den Empfang replizierter Objekte aus einem Quell-Bucket

Anstatt die vorherige Richtlinie manuell zu Ihrem Ziel-Bucket hinzuzufügen, können Sie schnell die Richtlinien generieren, die erforderlich sind, um den Empfang replizierter Objekte aus einem Quell-Bucket über die Amazon S3 S3-Konsole zu ermöglichen.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Bucket aus, den Sie als Ziel-Bucket verwenden möchten.
4. Wählen Sie die Registerkarte Management (Verwaltung) aus und scrollen Sie nach unten zu Replication rules (Replikationsregeln).
5. Wählen Sie für Actions (Aktionen) die Option Receive replicated objects (Replizierte Objekte empfangen) aus.

   Folgen Sie den Anweisungen, geben Sie die AWS-Konto ID des Quell-Bucket-Kontos ein und wählen Sie dann Richtlinien generieren aus. Die Konsole generiert eine Amazon S3 S3-Bucket-Richtlinie und eine KMS Schlüsselrichtlinie.
6. Wenn Sie diese Richtlinie Ihrer bestehenden Bucket-Richtlinie hinzufügen möchten, wählen Sie entweder Apply settings (Einstellungen anwenden) oder Copy (Kopieren) aus, um die Änderungen manuell zu kopieren.
7. (Optional) Kopieren Sie die AWS KMS Richtlinie in die gewünschte KMS Schlüsselrichtlinie in der AWS Key Management Service Konsole.

Ändern des Replikatbesitzers

Wenn unterschiedliche AWS-Konten Eigentümer des Quell- und des Ziel-Buckets sind, können Sie Amazon S3 anweisen, den Eigentümer des Replikats auf den Eigentümer des Ziel-Buckets zu ändern. AWS-Konto Weitere Informationen zum Außerkraftsetzen des Besitzers finden Sie unter Ändern des Replikat-Eigentümers.

Gewähren von Berechtigungen für S3-Batch-Operationen

S3 Batch Replication bietet Ihnen die Möglichkeit, die folgenden Objekte zu replizieren:

- Objekte, die vor der Einrichtung einer Replikationskonfiguration existierten
- Objekte, die zuvor repliziert wurden
- Objekte, deren Replikation fehlgeschlagen ist

Sie können einen einmaligen Batch-Replikationsauftrag erstellen, wenn Sie die erste Regel in einer neuen Replikationskonfiguration erstellen oder wenn Sie einer bestehenden Konfiguration über die Amazon S3 S3-Konsole ein neues Ziel hinzufügen. Sie können die Batch-Replikation auch für eine bestehende Replikationskonfiguration initiieren, indem Sie einen Batch Operations-Job erstellen.

Eine IAM Batch-Replikationsrolle und Richtlinienbeispiele finden Sie unterKonfiguration einer IAM Rolle für die S3-Batch-Replikation.

Beispiele für die Konfiguration der Live-Replikation

Die folgenden Beispiele bieten step-by-step exemplarische Vorgehensweisen, die zeigen, wie die Live-Replikation für allgemeine Anwendungsfälle konfiguriert wird.

> **Note**
>
> Die Live-Replikation bezieht sich auf die Replikation in derselben Region (SRR) und die regionsübergreifende Replikation (). CRR Bei der Live-Replikation werden keine Objekte repliziert, die vor der Einrichtung der Replikation im Bucket vorhanden waren. Um Objekte zu replizieren, die vor der Einrichtung der Replikation vorhanden waren, verwenden Sie die On-Demand-Replikation. Informationen zum Synchronisieren von Buckets und zum Replizieren vorhandener Objekte bei Bedarf finden Sie unter. Replizieren vorhandener Objekte

Diese Beispiele zeigen, wie eine Replikationskonfiguration mithilfe der Amazon S3 S3-Konsole, AWS Command Line Interface (AWS CLI) und AWS SDKs (AWS SDK for Java und es werden AWS SDK for .NET Beispiele gezeigt) erstellt.

Informationen zur Installation und Konfiguration von finden Sie in den folgenden Themen im AWS Command Line Interface Benutzerhandbuch: AWS CLI

- Beginnen Sie mit dem AWS CLI

- Konfigurieren Sie das AWS CLI — Sie müssen mindestens ein Profil einrichten. Richten Sie bei kontoübergreifenden Szenarien zwei Profile ein.

Informationen über das AWS SDKs finden Sie unter AWS SDK for Javaund AWS SDK for .NET.

> **Tip**
>
> Ein step-by-step Tutorial, das zeigt, wie Sie Daten mithilfe der Live-Replikation replizieren können, finden Sie unter Praktische Einführung: Replizieren von Daten innerhalb und zwischen Daten AWS-Regionen mithilfe der S3-Replikation.

Themen

- Konfiguration der Replikation für Buckets im selben Konto
- Konfiguration der Replikation für Buckets in verschiedenen Konten
- Erfüllung von Compliance-Anforderungen mit S3 Replication Time Control
- Replikation verschlüsselter Objekte (SSE-S3, SSE -, DSSE - KMSKMS, SSE -C)
- Replizieren von Metadatenänderungen mit der Replikatmodifikationssynchronisierung
- Replizieren von Löschmarkierungen auf Buckets

Konfiguration der Replikation für Buckets im selben Konto

Bei der Live-Replikation handelt es sich um das automatische, asynchrone Kopieren von Objekten über Buckets hinweg, die sich im selben oder in unterschiedlichen Buckets befinden. AWS-Regionen Bei der Live-Replikation werden neu erstellte Objekte und Objektaktualisierungen von einem Quell-Bucket in einen oder mehrere Ziel-Buckets kopiert. Weitere Informationen finden Sie unter Objekte innerhalb und zwischen Regionen replizieren.

Wenn Sie die Replikation konfigurieren, fügen Sie dem Quell-Bucket Replikationsregeln hinzu. Replikationsregeln definieren, welche Quell-Bucket-Objekte repliziert werden sollen, und den Ziel-Bucket/die Ziel-Buckets, in dem/denen die replizierten Objekte gespeichert werden sollen. Sie können eine Regel erstellen, um alle Objekte in einem Bucket oder eine Untermenge von Objekten mit einem spezifischen Schlüsselnamenpräfixen, einem oder mehreren Objekt-Markierungen oder beidem zu replizieren. Ein Ziel-Bucket kann sich im selben AWS-Konto wie der Quell-Bucket oder in einem anderen Konto befinden.

Wenn Sie angeben, dass eine Objektversions-ID gelöscht werden soll, löscht Amazon S3 diese Objektversion im Quell-Bucket. Es repliziert die Löschung aber nicht im Ziel-Bucket. Anders ausgedrückt: Dieselbe Objektversion wird im Ziel-Bucket nicht gelöscht. Dies schützt Daten vor missbräuchlichen Löschungen.

Wenn Sie einem Bucket eine Replikationsregel hinzufügen, ist diese standardmäßig aktiviert, sodass sie ausgeführt wird, sobald Sie sie speichern.

In diesem Beispiel richten Sie die Live-Replikation für Quell- und Ziel-Buckets ein, die demselben AWS-Konto gehören. Es werden Beispiele für die Verwendung der Amazon S3 S3-Konsole, der AWS Command Line Interface (AWS CLI) und der AWS SDK for Java und bereitgestellt AWS SDK for .NET.

Verwenden der S3-Konsole

Gehen Sie wie folgt vor, um eine Replikationsregel zu konfigurieren, wenn sich der Ziel-Bucket im selben AWS-Konto wie der Quell-Bucket befindet.

Wenn sich der Ziel-Bucket in einem anderen Konto als der Quell-Bucket befindet, müssen Sie eine Bucket-Richtlinie für den Ziel-Bucket hinzufügen, um dem Eigentümer des Quell-Bucket-Kontos die Berechtigung zum Replizieren von Objekten in der Ziel-Bucket zu erteilen. Weitere Informationen finden Sie unter Erteilen von Berechtigungen, wenn die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des von Ihnen erstellten Buckets aus.
4. Wählen Sie die Registerkarte Verwaltung aus, scrollen Sie nach unten zu Replikationsregeln und wählen Sie dann Replikationsregel erstellen aus.
5. Geben Sie im Abschnitt Replikationsregelkonfiguration unter Name der Replikationsregel einen Namen für Ihre Regel ein, um die Regel später leichter identifizieren zu können. Der Name ist erforderlich und muss innerhalb des Buckets eindeutig sein.
6. In Status (Status)ist Enabled (Aktiviert) standardmäßig ausgewählt. Eine aktivierte Regel wird ausgeführt, sobald Sie speichern. Wenn Sie die Regel später aktivieren möchten, wählen Sie Deaktiviert aus.

7. Wenn es bereits Replikationsregeln für den Bucket gibt, werden Sie angewiesen, eine Priorität für die Regel festzulegen. Sie müssen eine Priorität für die Regel festlegen, um Konflikte zu vermeiden, die durch Objekte verursacht werden, die mehreren Regeln unterliegen. Wenn sich Regeln überschneiden, verwendet Amazon S3 die Regelpriorität, um die Regel zu ermitteln, die angewendet werden muss. Je höher die Zahl, desto höher die Priorität. Weitere Informationen zur Priorität von Regeln finden Sie unter Elemente der Replikationskonfigurationsdatei.
8. Unter Quell-Bucket stehen Ihnen folgende Optionen zum Festlegen der Replikationsquelle zur Verfügung:

   - Um den gesamten Bucket zu replizieren, wählen Sie Apply to all objects in the bucket (Auf alle Objekte im Bucket anwenden).
   - Um alle Objekte zu replizieren, die dasselbe Präfix haben, wählen Sie Limit the scope of this rule using one or more filters (Geltungsbereich dieser Regel mit einem oder mehreren Filtern einschränken). Dies beschränkt die Replikation auf alle Objekte, deren Namen mit dem von Ihnen angegebenen Präfix beginnen (z. B. `pictures`). Geben Sie ein Präfix in das Feld Präfix ein.

     > **Note**
     >
     > Wenn Sie ein Präfix eingeben, das den Namen eines Ordners darstellt, müssen Sie / (Schrägstrich) als letztes Zeichen eingeben (z. B. `pictures/`).

   - Um alle Objekte mit einem oder mehreren Objekt-Tags zu replizieren, wählen Sie Tag hinzufügen aus und geben Sie das Schlüssel-Wert-Paar in die Felder ein. Wiederholen Sie den Vorgang, um ein weiteres Tag hinzuzufügen. Sie können ein Präfix und Markierungen kombinieren. Weitere Informationen über Objekt-Markierungen finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen.

   Das neue XML Replikationskonfigurationsschema unterstützt Präfix- und Tag-Filterung sowie die Priorisierung von Regeln. Weitere Informationen über das neue Schema finden Sie unter Überlegungen zur Abwärtskompatibilität. Weitere Informationen zur XML Verwendung mit Amazon S3API, die hinter der Benutzeroberfläche steckt, finden Sie unterElemente der Replikationskonfigurationsdatei. Das neue Schema wird als Replikationskonfiguration XML V2 beschrieben.
9. Wählen Sie unter Ziel den Bucket aus, in den Amazon S3 Objekte replizieren soll.

> **Note**
>
> Die Anzahl der Ziel-Buckets ist auf die Anzahl der AWS-Regionen in einer bestimmten Partition beschränkt. Eine Partition ist eine Gruppierung von Regionen. AWS hat derzeit drei Partitionen: `aws` (Standardregionen), `aws-cn` (China-Regionen) und `aws-us-gov` (AWS GovCloud (US) Regionen). Sie können Service Quotas verwenden, um eine Erhöhung Ihres Ziel-Bucket-Kontingents anzufordern.

- Um in einen Bucket oder mehrere Buckets in Ihrem Konto zu replizieren, wählen Sie Bucket in diesem Konto wählen aus und geben Sie die Ziel-Bucket-Namen ein oder suchen Sie danach.
- Um in einen oder mehrere Buckets in einem anderen Bucket zu replizieren AWS-Konto, wählen Sie Spezify a bucket in another account aus und geben Sie die Konto-ID und den Bucket-Namen des Ziel-Buckets ein.

  Wenn sich das Ziel in einem anderen Konto als der Quell-Bucket befindet, müssen Sie eine Bucket-Richtlinie für die Ziel-Buckets hinzufügen, um dem Eigentümer des Quell-Bucket-Kontos die Berechtigung zum Replizieren von Objekten zu erteilen. Weitere Informationen finden Sie unter Erteilen von Berechtigungen, wenn die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten.

  Wenn Sie eine leichtere Standardisierung der Eigentümerschaft für neue Objekte im Ziel-Bucket ermöglichen möchten, wählen Sie Objekteigentümerschaft in Eigentümer des Ziel-Buckets ändern aus. Weitere Informationen zu dieser Option finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

> **Note**
>
> Wenn die Versionssteuerung für den Ziel-Bucket nicht aktiviert ist, erhalten Sie eine Warnmeldung, die die Schaltfläche Versionierung aktivieren enthält. Wählen Sie diese Schaltfläche, um das Versioning für den Bucket zu aktivieren.

10. Richten Sie eine AWS Identity and Access Management (IAM) -Rolle ein, die Amazon S3 übernehmen kann, um Objekte in Ihrem Namen zu replizieren.

Um eine IAM Rolle einzurichten, wählen Sie im Abschnitt IAMRolle eine der folgenden Optionen aus der IAMRollen-Dropdownliste aus:

- Wir empfehlen dringend, dass Sie Neue Rolle erstellen wählen, damit Amazon S3 eine neue IAM Rolle für Sie erstellt. Wenn Sie die Regel speichern, wird eine neue Richtlinie für die IAM Rolle generiert, die den von Ihnen ausgewählten Quell- und Ziel-Buckets entspricht.
- Sie können wählen, ob Sie eine vorhandene IAM Rolle verwenden möchten. In diesem Fall müssen Sie eine Rolle auswählen, die Amazon S3 die erforderlichen Berechtigungen für die Replikation gewährt. Die Replikation schlägt fehl, wenn diese Rolle Amazon S3 keine ausreichenden Berechtigungen gewährt, um Ihre Replikationsregel zu befolgen.

> **Important**
>
> Wenn Sie einem Bucket eine Replikationsregel hinzufügen, benötigen Sie die `iam:PassRole` Erlaubnis, die IAM Rolle zu übergeben, die Amazon S3 S3-Replikationsberechtigungen gewährt. Weitere Informationen finden Sie unter Erteilen von Benutzerberechtigungen zur Übergabe einer Rolle an einen AWS-Service im IAMBenutzerhandbuch.

11. Um Objekte im Quell-Bucket zu replizieren, die mit serverseitiger Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-KMS) verschlüsselt wurden, wählen Sie unter Verschlüsselung die Option Objekte replizieren mit aus. AWS KMS Unter AWS KMS -Schlüssel für die Verschlüsselung von Zielobjekten befinden sich die Quellschlüssel, deren Verwendung Sie der Replikation erlauben. Alle KMS Quellschlüssel sind standardmäßig enthalten. Um die KMS Schlüsselauswahl einzuschränken, können Sie einen Alias oder eine Schlüssel-ID wählen.

    Objekte AWS KMS keys , mit denen Sie nicht verschlüsselt sind, werden nicht repliziert. Ein KMS Schlüssel oder eine Gruppe von KMS Schlüsseln wird für Sie ausgewählt, aber Sie können die KMS Schlüssel wählen, wenn Sie möchten. Hinweise zur Verwendung AWS KMS mit Replikation finden Sie unterReplikation verschlüsselter Objekte (SSE-S3, SSE -, DSSE - KMSKMS, SSE -C).

    > **Important**
    >
    > Wenn Sie Objekte replizieren, die mit verschlüsselt sind AWS KMS, verdoppelt sich die AWS KMS Anforderungsrate in der Quellregion und steigt in der Zielregion

um den gleichen Betrag. Diese erhöhten Anrufraten AWS KMS sind auf die Art und Weise zurückzuführen, wie Daten mithilfe des KMS Schlüssels, den Sie für die Zielregion der Replikation definieren, erneut verschlüsselt werden. AWS KMS hat ein Kontingent für die Anforderungsrate, das pro Anrufkonto und Region gilt. Informationen zu den standardmäßigen Kontingenten finden Sie unter AWS KMS - Kontingente – Anforderungen pro Sekunde: variabel im AWS Key Management Service Entwicklerhandbuch.
Wenn Ihre aktuelle Amazon S3 PUT S3-Objektanforderungsrate während der Replikation mehr als die Hälfte der AWS KMS Standardratenbegrenzung für Ihr Konto beträgt, empfehlen wir Ihnen, eine Erhöhung Ihres Anforderungsratenkontingents zu AWS KMS beantragen. Wenn Sie eine Erhöhung anfordern möchten, erstellen Sie einen Fall im AWS Support -Center unter Contact Us (Kontakt). Nehmen wir zum Beispiel an, dass Ihre aktuelle PUT Objektanforderungsrate 1.000 Anfragen pro Sekunde beträgt und AWS KMS Sie Ihre Objekte verschlüsseln. In diesem Fall empfehlen wir Ihnen, Ihr AWS KMS Ratenlimit sowohl in Ihrer Quell- als auch in Ihrer Zielregion (falls unterschiedlich) auf 2.500 Anfragen pro Sekunde zu erhöhen, um sicherzustellen, dass es nicht zu einer Drosselung durch kommt. AWS Support AWS KMS
Ihre PUT Objektanforderungsrate im Quell-Bucket finden Sie `PutRequests` in den CloudWatch Amazon-Anforderungsmetriken für Amazon S3. Informationen zum Anzeigen von CloudWatch Metriken finden Sie unterVerwenden der S3-Konsole.

Wenn Sie Objekte replizieren möchten, die mit verschlüsselt sind AWS KMS, gehen Sie wie folgt vor:

- Geben Sie unter AWS KMS key Für die Verschlüsselung von Zielobjekten Ihren KMS Schlüssel auf eine der folgenden Arten an:

  - Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys und wählen Sie Ihren KMSSchlüssel aus der Liste der verfügbaren Schlüssel aus.

    Sowohl der Von AWS verwalteter Schlüssel (`aws/s3`) als auch Ihr vom Kunden verwalteter Schlüssel werden in dieser Liste angezeigt. Weitere Informationen über vom Kunden verwaltete Schlüssel finden Sie unter Kundenschlüssel und AWS -Schlüssel im Entwicklerhandbuch zu AWS Key Management Service .

- Um den KMS Schlüssel Amazon Resource Name (ARN) einzugeben, wählen Sie Enter AWS KMS key ARN und geben Sie Ihren KMS Schlüssel ARN in das angezeigte Feld ein. Dadurch werden die Replikate im Ziel-Bucket verschlüsselt. Sie finden den Schlüssel ARN für Ihren KMS Schlüssel in der IAMKonsole unter Verschlüsselungsschlüssel.
- Um einen neuen, vom Kunden verwalteten Schlüssel in der AWS KMS Konsole zu erstellen, wählen Sie Create a KMS key aus.

  Weitere Informationen zum Erstellen eines AWS KMS key finden Sie unter Schlüssel erstellen im AWS Key Management Service Entwicklerhandbuch.

> **Important**
>
> Sie können nur KMS Schlüssel verwenden, die im selben AWS-Region Bucket aktiviert sind. Wenn Sie „Aus Ihren KMS Schlüsseln auswählen“ wählen, listet die S3-Konsole nur 100 KMS Schlüssel pro Region auf. Wenn Sie mehr als 100 KMS Schlüssel in derselben Region haben, können Sie nur die ersten 100 KMS Schlüssel in der S3-Konsole sehen. Um einen KMS Schlüssel zu verwenden, der nicht in der Konsole aufgeführt ist, wählen Sie Enter AWS KMS key ARN und geben Sie Ihren KMS Schlüssel einARN.
> Wenn Sie eine AWS KMS key für die serverseitige Verschlüsselung in Amazon S3 verwenden, müssen Sie einen symmetrischen KMS Verschlüsselungsschlüssel wählen. Amazon S3 unterstützt nur symmetrische KMS Verschlüsselungsschlüssel und keine asymmetrischen KMS Schlüssel. Weitere Informationen finden Sie unter Identifizieren symmetrischer und asymmetrischer KMS Schlüssel im Entwicklerhandbuch.AWS Key Management Service

Weitere Informationen zum Erstellen von finden Sie unter Creating Keys im AWS Key Management Service Developer Guide. AWS KMS key Weitere Informationen zur Verwendung AWS KMS mit Amazon S3 finden Sie unterServerseitige Verschlüsselung mit AWS KMS Schlüsseln verwenden (SSE-) KMS.

12. Wenn Sie Ihre Daten in eine bestimmte Speicherklasse im Ziel replizieren wollen, wählen Sie unter Zielspeicherklasse die Option Speicherklasse für die replizierten Objekte ändern aus. Anschließend wählen Sie die Speicherklasse, die Sie für die replizierten Objekte im Ziel verwenden wollen. Wenn Sie diese Option nicht auswählen, ist die Speicherklasse für replizierte Objekte dieselbe Klasse wie für die ursprünglichen Objekte.

13. Beim Festlegen der zusätzlichen Replikationsoptionen haben Sie folgende zusätzliche Optionen:

    - Wenn Sie S3 Replication Time Control (S3RTC) in Ihrer Replikationskonfiguration aktivieren möchten, wählen Sie Replication Time Control (RTC) aus. Weitere Informationen zu dieser Option finden Sie unter Erfüllung von Compliance-Anforderungen mit S3 Replication Time Control.
    - Wenn Sie S3-Replikationsmetriken in Ihrer Replikationskonfiguration aktivieren möchten, wählen Sie Replication metrics and events (Replikationsmetriken und -ereignisse) aus. Weitere Informationen finden Sie unter Überwachung der Replikation mit Metriken, Ereignisbenachrichtigungen und Status.
    - Wenn Sie die Löschmarkierungs-Replikation in Ihrer Replikations-Konfiguration aktivieren möchten, wählen Sie Markierungsreplikation löschen aus. Weitere Informationen finden Sie unter Replizieren von Löschmarkierungen auf Buckets.
    - Wenn Sie die Synchronisierung der Amazon-S3-Replikatänderung in Ihrer Replikations-Konfiguration aktivieren möchten, wählen Sie Synchronisierung der Replikatänderung aus. Weitere Informationen finden Sie unter Replizieren von Metadatenänderungen mit der Replikatmodifikationssynchronisierung.

    > **Note**
    >
    > Wenn Sie S3 RTC - oder S3-Replikationsmetriken verwenden, fallen zusätzliche Gebühren an.

14. Zum Abschluss wählen Sie Save (Speichern).
15. Nachdem Sie Ihre Regel gespeichert haben, können Sie Ihre Regel bearbeiten, aktivieren, deaktivieren oder löschen, indem Sie Ihre Regel auswählen und Edit rule (Regel bearbeiten) wählen.

Mit dem AWS CLI

Gehen Sie wie folgt vor AWS CLI , um die Replikation einzurichten, wenn Quell- und Ziel-Buckets demselben AWS-Konto gehören:

- Erstellen Sie Quell- und Ziel-Buckets.
- Aktivieren Sie die Versionierung für die Buckets.

- Erstellen Sie eine AWS Identity and Access Management (IAM) -Rolle, die Amazon S3 die Berechtigung erteilt, Objekte zu replizieren.
- Fügen Sie die Replikationskonfiguration zum Quell-Bucket hinzu.

Um die Einrichtung zu prüfen, testen Sie sie.

Um die Replikation einzurichten, wenn der Quell- und der Ziel-Bucket demselben gehören AWS-Konto

1. Richten Sie das Anmeldeinformationsprofil für die AWS CLI ein. Dieses Beispiel verwendet den Profilnamen `acctA`. Informationen zum Einrichten von Anmeldeinformationsprofilen und zur Verwendung benannter Profile finden Sie im Benutzerhandbuch unter Konfiguration und Einstellungen für Anmeldeinformationsdateien.AWS Command Line Interface

   > **Important**
   >
   > Das Profil, das Sie für dieses Beispiel verwenden, muss über die erforderlichen Berechtigungen verfügen. In der Replikationskonfiguration geben Sie beispielsweise die IAM Rolle an, die Amazon S3 übernehmen kann. Sie können dies nur tun, wenn das von Ihnen verwendete Profil über die `iam:PassRole` entsprechende Berechtigung verfügt. Weitere Informationen finden Sie im Benutzerhandbuch unter Erteilen von Berechtigungen zum Übergeben einer Rolle AWS-Service an einen. IAM Wenn Sie zur Erstellung eines benannten Profils die Anmeldeinformationen eines Administrators verwenden, können Sie alle Aufgaben durchführen.

2. Erstellen Sie einen Quell-Bucket und aktivieren Sie die Versionsverwaltung für ihn, indem Sie die folgenden AWS CLI Befehle verwenden. Um diese Befehle zu verwenden, ersetzen Sie den *`Platzhalter für Benutzereingaben`* mit Ihren eigenen Informationen.

   Mit dem folgenden `create-bucket` Befehl wird ein Quell-Bucket mit *`amzn-s3-demo-source-bucket`* dem Namen der Region USA Ost (Nord-Virginia) (`us-east-1`) erstellt:

   ```
   aws s3api create-bucket \
   --bucket amzn-s3-demo-source-bucket \
   --region us-east-1 \
   --profile acctA
   ```

Der folgende `put-bucket-versioning` Befehl aktiviert die S3-Versionierung für den *amzn-s3-demo-source-bucket* Bucket:

```
aws s3api put-bucket-versioning \
--bucket amzn-s3-demo-source-bucket \
--versioning-configuration Status=Enabled \
--profile acctA
```

3. Erstellen Sie einen Ziel-Bucket und aktivieren Sie die Versionsverwaltung für ihn, indem Sie die folgenden AWS CLI Befehle verwenden. Um diese Befehle zu verwenden, ersetzen Sie den *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

> **Note**
>
> Um eine Replikationskonfiguration einzurichten, wenn sich Quell- und Ziel-Bucket in demselben befinden AWS-Konto, verwenden Sie dasselbe Profil für die Quell- und Ziel-Buckets. Dieses Beispiel verwendet `acctA`.
> Um eine Replikationskonfiguration zu testen, wenn die Buckets unterschiedlichen Besitzern gehören AWS-Konten, geben Sie für jedes Konto unterschiedliche Profile an. Verwenden Sie beispielsweise ein `acctB` Profil für den Ziel-Bucket.

Mit dem folgenden `create-bucket` Befehl wird ein Ziel-Bucket mit *amzn-s3-demo-destination-bucket* dem Namen der Region USA West (Oregon) (`us-west-2`) erstellt:

```
aws s3api create-bucket \
--bucket amzn-s3-demo-destination-bucket \
--region us-west-2 \
--create-bucket-configuration LocationConstraint=us-west-2 \
--profile acctA
```

Der folgende `put-bucket-versioning` Befehl aktiviert die S3-Versionierung für den *amzn-s3-demo-destination-bucket* Bucket:

```
aws s3api put-bucket-versioning \
--bucket amzn-s3-demo-destination-bucket \
--versioning-configuration Status=Enabled \
```

```
--profile acctA
```

4. Erstellen Sie eine IAM Rolle. Sie geben diese Rolle in der Replizierungskonfiguration an, die Sie später zum *Quell-Bucket* hinzufügen. Amazon S3 übernimmt diese Rolle, um Objekte in Ihrem Namen zu replizieren. Sie erstellen eine IAM Rolle in zwei Schritten:

   - Erstellen Sie eine Rolle.
   - Fügen Sie eine Berechtigungsrichtlinie zur Rolle hinzu.

   a. Erstellen Sie die IAM Rolle.

      i. Kopieren Sie die folgende Vertrauensrichtlinie und speichern Sie sie in einer Datei mit dem Namen `s3-role-trust-policy.json` im aktuellen Verzeichnis auf Ihrem lokalen Computer. Diese Richtlinie gewährt dem Amazon S3 S3-Serviceprinzipal die Erlaubnis, die Rolle zu übernehmen.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

      ii. Führen Sie den folgenden Befehl aus, um eine Rolle zu erstellen.

```
$ aws iam create-role \
--role-name replicationRole \
--assume-role-policy-document file://s3-role-trust-policy.json  \
--profile acctA
```

   b. Fügen Sie eine Berechtigungsrichtlinie zur Rolle hinzu.

      i. Kopieren Sie die folgende Berechtigungsrichtlinie und speichern Sie sie in einer Datei mit dem Namen `s3-role-permissions-policy.json` im aktuellen Verzeichnis auf

Ihrem lokalen Computer. Diese Zugriffsrichtlinie erteilt Berechtigungen für verschiedene Amazon-S3-Bucket- und -Objektaktionen.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl",
            "s3:GetObjectVersionTagging"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ListBucket",
            "s3:GetReplicationConfiguration"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ReplicateObject",
            "s3:ReplicateDelete",
            "s3:ReplicateTags"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
      }
   ]
}
```

ii. Führen Sie den folgenden Befehl aus, um eine Richtlinie zu erstellen und sie der Rolle anzufügen. Ersetzen Sie das *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
$ aws iam put-role-policy \
--role-name replicationRole \
--policy-document file://s3-role-permissions-policy.json \
--policy-name replicationRolePolicy \
--profile acctA
```

5. Fügen Sie eine Replikationskonfiguration zum Quell-Bucket hinzu.

   a. Amazon S3 API erfordert zwar, dass Sie die Replikationskonfiguration als angebenXML, AWS CLI erfordert jedoch, dass Sie die Replikationskonfiguration als angebenJSON. Speichern Sie Folgendes JSON in einer Datei namens `replication.json` im lokalen Verzeichnis auf Ihrem Computer.

```
{
  "Role": "IAM-role-ARN",
  "Rules": [
    {
      "Status": "Enabled",
      "Priority": 1,
      "DeleteMarkerReplication": { "Status": "Disabled" },
      "Filter" : { "Prefix": "Tax"},
      "Destination": {
        "Bucket": "arn:aws:s3:::amzn-s3-demo-destination-bucket"
      }
    }
  ]
}
```

   b. Aktualisieren Sie die, JSON indem Sie die Werte für *amzn-s3-demo-destination-bucket* und *IAM-role-ARN* durch Ihre eigenen Informationen ersetzen. Speichern Sie die Änderungen.

   c. Führen Sie den folgenden `put-bucket-replication` Befehl aus, um die Replikationskonfiguration zu Ihrem Quell-Bucket hinzuzufügen. Stellen Sie sicher, dass Sie den Quell-Bucket-Namen angeben:

```
$ aws s3api put-bucket-replication \
--replication-configuration file://replication.json \
--bucket amzn-s3-demo-source-bucket \
--profile acctA
```

Verwenden Sie den `get-bucket-replication` folgenden Befehl, um die Replikationskonfiguration abzurufen:

```
$ aws s3api get-bucket-replication \
--bucket amzn-s3-demo-source-bucket \
--profile acctA
```

6. Testen Sie das Setup in der Amazon S3 S3-Konsole, indem Sie die folgenden Schritte ausführen:

   a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

   b. Wählen Sie im linken Navigationsbereich Buckets aus. Wählen Sie in der Liste der Buckets für allgemeine Zwecke den Quell-Bucket aus.

   c. Erstellen Sie im Quell-Bucket einen Ordner mit dem Namen *Tax*.

   d. Fügen Sie dem *Tax* Ordner im Quell-Bucket Beispielobjekte hinzu.

   > **Note**
   >
   > Die Zeit, die Amazon S3 benötigt, um ein Objekt zu replizieren, hängt von der Größe des Objekts ab. Weitere Informationen zum Anzeigen des Replikationsstatus finden Sie unter Abrufen von Replikationsstatusinformationen.

   Überprüfen Sie im *Ziel*-Bucket Folgendes:

   - Dass Amazon S3 die Objekte repliziert hat.
   - Dass es sich bei den Objekten um Replikate handelt. Scrollen Sie auf der Registerkarte Eigenschaften für Ihre Objekte nach unten zum Abschnitt Übersicht über die Objektverwaltung. Unter Verwaltungskonfigurationen finden Sie den Wert unter Replikationsstatus. Stellen Sie sicher, dass dieser Wert auf eingestellt ist`REPLICA`.
   - Dass die Replikate dem Quell-Bucket-Konto gehören. Sie können den Objekteigentum auf der Registerkarte Berechtigungen für Ihre Objekte überprüfen.

     Wenn die Quell- und Ziel-Buckets unterschiedlichen Konten gehören, können Sie eine optionale Konfiguration hinzufügen, um Amazon S3 anzuweisen, den Besitz des Replikats

auf das Zielkonto zu ändern. Ein Beispiel finden Sie unter Wie ändere ich den Besitzer des Replikats.

Mit dem AWS SDKs

Verwenden Sie die folgenden Codebeispiele, um einem Bucket eine Replikationskonfiguration mit dem AWS SDK for Java bzw. AWS SDK for .NET hinzuzufügen.

Java

Das folgende Beispiel fügt eine Replikations-Konfiguration einem Bucket hinzu und ruft die Konfiguration anschließend ab und überprüft sie. Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.identitymanagement.AmazonIdentityManagement;
import
 com.amazonaws.services.identitymanagement.AmazonIdentityManagementClientBuilder;
import com.amazonaws.services.identitymanagement.model.CreateRoleRequest;
import com.amazonaws.services.identitymanagement.model.PutRolePolicyRequest;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.BucketReplicationConfiguration;
import com.amazonaws.services.s3.model.BucketVersioningConfiguration;
import com.amazonaws.services.s3.model.CreateBucketRequest;
import com.amazonaws.services.s3.model.DeleteMarkerReplication;
import com.amazonaws.services.s3.model.DeleteMarkerReplicationStatus;
import com.amazonaws.services.s3.model.ReplicationDestinationConfig;
import com.amazonaws.services.s3.model.ReplicationRule;
import com.amazonaws.services.s3.model.ReplicationRuleStatus;
import com.amazonaws.services.s3.model.SetBucketVersioningConfigurationRequest;
import com.amazonaws.services.s3.model.StorageClass;
import com.amazonaws.services.s3.model.replication.ReplicationFilter;
import com.amazonaws.services.s3.model.replication.ReplicationFilterPredicate;
import com.amazonaws.services.s3.model.replication.ReplicationPrefixPredicate;

import java.io.IOException;
```

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class CrossRegionReplication {

        public static void main(String[] args) throws IOException {
                Regions clientRegion = Regions.DEFAULT_REGION;
                String accountId = "*** Account ID ***";
                String roleName = "*** Role name ***";
                String sourceBucketName = "*** Source bucket name ***";
                String destBucketName = "*** Destination bucket name ***";
                String prefix = "Tax/";

                String roleARN = String.format("arn:aws:iam::%s:%s", accountId,
 roleName);
                String destinationBucketARN = "arn:aws:s3:::" + destBucketName;

                AmazonS3 s3Client = AmazonS3Client.builder()
                                .withCredentials(new ProfileCredentialsProvider())
                                .withRegion(clientRegion)
                                .build();

                createBucket(s3Client, clientRegion, sourceBucketName);
                createBucket(s3Client, clientRegion, destBucketName);
                assignRole(roleName, clientRegion, sourceBucketName,
 destBucketName);

                try {

                        // Create the replication rule.
                        List<ReplicationFilterPredicate> andOperands = new
 ArrayList<ReplicationFilterPredicate>();
                        andOperands.add(new ReplicationPrefixPredicate(prefix));

                        Map<String, ReplicationRule> replicationRules = new
 HashMap<String, ReplicationRule>();
                        replicationRules.put("ReplicationRule1",
                                        new ReplicationRule()
                                                        .withPriority(0)

 .withStatus(ReplicationRuleStatus.Enabled)
```

```
.withDeleteMarkerReplication(
                                                                    new
DeleteMarkerReplication().withStatus(

   DeleteMarkerReplicationStatus.DISABLED))
                                                        .withFilter(new
ReplicationFilter().withPredicate(
                                                                    new
ReplicationPrefixPredicate(prefix)))
                                                        .withDestinationConfig(new
ReplicationDestinationConfig()

.withBucketARN(destinationBucketARN)

.withStorageClass(StorageClass.Standard)));

                        // Save the replication rule to the source bucket.
                        s3Client.setBucketReplicationConfiguration(sourceBucketName,
                                        new BucketReplicationConfiguration()
                                                        .withRoleARN(roleARN)

.withRules(replicationRules));

                        // Retrieve the replication configuration and verify that
the configuration
                        // matches the rule we just set.
                        BucketReplicationConfiguration replicationConfig = s3Client

.getBucketReplicationConfiguration(sourceBucketName);
                        ReplicationRule rule =
replicationConfig.getRule("ReplicationRule1");
                        System.out.println("Retrieved destination bucket ARN: "
                                        +
rule.getDestinationConfig().getBucketARN());
                        System.out.println("Retrieved priority: " +
rule.getPriority());
                        System.out.println("Retrieved source-bucket replication rule
status: " + rule.getStatus());
                } catch (AmazonServiceException e) {
                        // The call was transmitted successfully, but Amazon S3
couldn't process
                        // it, so it returned an error response.
                        e.printStackTrace();
```

```
                } catch (SdkClientException e) {
                        // Amazon S3 couldn't be contacted for a response, or the
 client
                        // couldn't parse the response from Amazon S3.
                        e.printStackTrace();
                }
        }

        private static void createBucket(AmazonS3 s3Client, Regions region, String
 bucketName) {
                CreateBucketRequest request = new CreateBucketRequest(bucketName,
 region.getName());
                s3Client.createBucket(request);
                BucketVersioningConfiguration configuration = new
 BucketVersioningConfiguration()
                                .withStatus(BucketVersioningConfiguration.ENABLED);

                SetBucketVersioningConfigurationRequest enableVersioningRequest =
 new SetBucketVersioningConfigurationRequest(
                                bucketName, configuration);
                s3Client.setBucketVersioningConfiguration(enableVersioningRequest);

        }

        private static void assignRole(String roleName, Regions region, String
 sourceBucket, String destinationBucket) {
                AmazonIdentityManagement iamClient =
 AmazonIdentityManagementClientBuilder.standard()
                                .withRegion(region)
                                .withCredentials(new ProfileCredentialsProvider())
                                .build();
                StringBuilder trustPolicy = new StringBuilder();
                trustPolicy.append("{\\r\\n   ");
                trustPolicy.append("\\\"Version\\\":\\\"2012-10-17\\\",\\r\\n   ");
                trustPolicy.append("\\\"Statement\\\":[\\r\\n      {\\r\\n
 ");
                trustPolicy.append("\\\"Effect\\\":\\\"Allow\\\",\\r\\n         \\
\"Principal\\\":{\\r\\n            ");
                trustPolicy.append("\\\"Service\\\":\\\"s3.amazonaws.com\\\"\\r\\n
      },\\r\\n         ");
                trustPolicy.append("\\\"Action\\\":\\\"sts:AssumeRole\\\"\\r\\n
  }\\r\\n   ]\\r\\n}");

                CreateRoleRequest createRoleRequest = new CreateRoleRequest()
```

```
                                .withRoleName(roleName)

 .withAssumeRolePolicyDocument(trustPolicy.toString());

                iamClient.createRole(createRoleRequest);

                StringBuilder permissionPolicy = new StringBuilder();
                permissionPolicy.append(
                                "{\\r\\n   \\\"Version\\\":\\\"2012-10-17\\\",\\r\\n
   \\\"Statement\\\":[\\r\\n      {\\r\\n         ");
                permissionPolicy.append(
                                "\\\"Effect\\\":\\\"Allow\\\",\\r\\n        \\
\"Action\\\":[\\r\\n             ");
                permissionPolicy.append("\\\"s3:GetObjectVersionForReplication\\\",\
\r\\n            ");
                permissionPolicy.append(
                                "\\\"s3:GetObjectVersionAcl\\\"\\r\\n         ],\\r\
\n         \\\"Resource\\\":[\\r\\n            ");
                permissionPolicy.append("\\\"arn:aws:s3:::");
                permissionPolicy.append(sourceBucket);
                permissionPolicy.append("/*\\\"\\r\\n         ]\\r\\n      },\\r\\n
     {\\r\\n         ");
                permissionPolicy.append(
                                "\\\"Effect\\\":\\\"Allow\\\",\\r\\n        \\
\"Action\\\":[\\r\\n            ");
                permissionPolicy.append(
                                "\\\"s3:ListBucket\\\",\\r\\n            \\
\"s3:GetReplicationConfiguration\\\"\\r\\n         ");
                permissionPolicy.append("],\\r\\n         \\\"Resource\\\":[\\r\\n
          \\\"arn:aws:s3:::");
                permissionPolicy.append(sourceBucket);
                permissionPolicy.append("\\r\\n         ");
                permissionPolicy
                                .append("]\\r\\n      },\\r\\n      {\\r\\n
 \\\"Effect\\\":\\\"Allow\\\",\\r\\n         ");
                permissionPolicy.append(
                                "\\\"Action\\\":[\\r\\n            \\
\"s3:ReplicateObject\\\",\\r\\n            ");
                permissionPolicy
                                .append("\\\"s3:ReplicateDelete\\\",\\r\\n
  \\\"s3:ReplicateTags\\\",\\r\\n            ");
                permissionPolicy.append("\\\"s3:GetObjectVersionTagging\\\"\\r\\n\\r
\\n         ],\\r\\n         ");
                permissionPolicy.append("\\\"Resource\\\":\\\"arn:aws:s3:::");
```

```
                permissionPolicy.append(destinationBucket);
                permissionPolicy.append("/*\\\"\\r\\n      }\\r\\n   ]\\r\\n}");

                PutRolePolicyRequest putRolePolicyRequest = new
 PutRolePolicyRequest()
                                .withRoleName(roleName)
                                .withPolicyDocument(permissionPolicy.toString())
                                .withPolicyName("crrRolePolicy");

                iamClient.putRolePolicy(putRolePolicyRequest);

        }
}
```

C#

Das folgende AWS SDK for .NET Codebeispiel fügt einem Bucket eine Replikationskonfiguration hinzu und ruft sie dann ab. Um diesen Code zu verwenden, geben Sie die Namen für Ihre Buckets und den Amazon-Ressourcennamen (ARN) für Ihre IAM Rolle an. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Getting Started with the AWS SDK for .NET im AWS SDK for .NET Entwicklerhandbuch.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CrossRegionReplicationTest
    {
        private const string sourceBucket = "*** source bucket ***";
        // Bucket ARN example - arn:aws:s3:::destinationbucket
        private const string destinationBucketArn = "*** destination bucket ARN
 ***";
        private const string roleArn = "*** IAM Role ARN ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint sourceBucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;
        public static void Main()
```

```
        {
            s3Client = new AmazonS3Client(sourceBucketRegion);
            EnableReplicationAsync().Wait();
        }
        static async Task EnableReplicationAsync()
        {
            try
            {
                ReplicationConfiguration replConfig = new ReplicationConfiguration
                {
                    Role = roleArn,
                    Rules =
                        {
                            new ReplicationRule
                            {
                                Prefix = "Tax",
                                Status = ReplicationRuleStatus.Enabled,
                                Destination = new ReplicationDestination
                                {
                                    BucketArn = destinationBucketArn
                                }
                            }
                        }
                };

                PutBucketReplicationRequest putRequest = new
PutBucketReplicationRequest
                {
                    BucketName = sourceBucket,
                    Configuration = replConfig
                };

                PutBucketReplicationResponse putResponse = await
s3Client.PutBucketReplicationAsync(putRequest);

                // Verify configuration by retrieving it.
                await RetrieveReplicationConfigurationAsync(s3Client);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
            catch (Exception e)
```

```
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
        private static async Task RetrieveReplicationConfigurationAsync(IAmazonS3
 client)
        {
            // Retrieve the configuration.
            GetBucketReplicationRequest getRequest = new GetBucketReplicationRequest
            {
                BucketName = sourceBucket
            };
            GetBucketReplicationResponse getResponse = await
 client.GetBucketReplicationAsync(getRequest);
            // Print.
            Console.WriteLine("Printing replication configuration information...");
            Console.WriteLine("Role ARN: {0}", getResponse.Configuration.Role);
            foreach (var rule in getResponse.Configuration.Rules)
            {
                Console.WriteLine("ID: {0}", rule.Id);
                Console.WriteLine("Prefix: {0}", rule.Prefix);
                Console.WriteLine("Status: {0}", rule.Status);
            }
        }
    }
}
```

Konfiguration der Replikation für Buckets in verschiedenen Konten

Bei der Live-Replikation handelt es sich um das automatische, asynchrone Kopieren von Objekten zwischen Buckets in demselben oder unterschiedlichen Buckets. AWS-Regionen Bei der Live-Replikation werden neu erstellte Objekte und Objektaktualisierungen von einem Quell-Bucket in einen oder mehrere Ziel-Buckets kopiert. Weitere Informationen finden Sie unter Objekte innerhalb und zwischen Regionen replizieren.

Wenn Sie die Replikation konfigurieren, fügen Sie dem Quell-Bucket Replikationsregeln hinzu. Replikationsregeln definieren, welche Quell-Bucket-Objekte repliziert werden sollen, und den Ziel-Bucket/die Ziel-Buckets, in dem/denen die replizierten Objekte gespeichert werden sollen. Sie können eine Regel erstellen, um alle Objekte in einem Bucket oder eine Untermenge von Objekten mit einem spezifischen Schlüsselnamenpräfixen, einem oder mehreren Objekt-Markierungen oder beidem zu

replizieren. Ein Ziel-Bucket kann sich im selben AWS-Konto wie der Quell-Bucket oder in einem anderen Konto befinden.

Wenn Sie angeben, dass eine Objektversions-ID gelöscht werden soll, löscht Amazon S3 diese Objektversion im Quell-Bucket. Es repliziert die Löschung aber nicht im Ziel-Bucket. Anders ausgedrückt: Dieselbe Objektversion wird im Ziel-Bucket nicht gelöscht. Dies schützt Daten vor missbräuchlichen Löschungen.

Wenn Sie einem Bucket eine Replikationsregel hinzufügen, ist diese standardmäßig aktiviert, sodass sie ausgeführt wird, sobald Sie sie speichern.

Das Einrichten der Live-Replikation, wenn Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören, AWS-Konten ähnelt dem Einrichten der Replikation, wenn beide Buckets demselben Konto gehören. Es gibt jedoch mehrere Unterschiede, wenn Sie die Replikation in einem kontoübergreifenden Szenario konfigurieren:

- Der Besitzer des Ziel-Buckets muss dem Besitzer des Quell-Buckets die Erlaubnis erteilen, Objekte in der Ziel-Bucket-Richtlinie zu replizieren.
- Wenn Sie Objekte replizieren, die mit serverseitiger Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-KMS) verschlüsselt sind, muss der Besitzer des Schlüssels dem Besitzer des Quell-Buckets die Erlaubnis zur Verwendung des KMS Schlüssels erteilen. KMS Weitere Informationen finden Sie unter Erteilen von zusätzlichen Berechtigungen für kontenübergreifende Szenarien.
- Standardmäßig gehören replizierte Objekte dem Besitzer des Quell-Buckets. In einem kontoübergreifenden Szenario möchten Sie die Replikation möglicherweise so konfigurieren, dass der Eigentümer der replizierten Objekte dem Besitzer des Ziel-Buckets zugewiesen wird. Weitere Informationen finden Sie unter Ändern des Replikat-Eigentümers.

Um die Replikation zu konfigurieren, wenn die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten

1. In diesem Beispiel erstellen Sie Quell- und Ziel-Buckets in zwei verschiedenen Buckets. AWS-Konten Sie müssen zwei Anmeldeinformationsprofile für eingerichtet haben. AWS CLI In diesem Beispiel werden `acctA` und `acctB` für diese Profilnamen verwendet. Informationen zum Einrichten von Anmeldeinformationsprofilen und zur Verwendung benannter Profile finden Sie im AWS Command Line Interface Benutzerhandbuch unter Konfiguration und Einstellungen für Anmeldeinformationsdateien.

2. Folgen Sie den step-by-step Anweisungen unter Konfiguration der Replikation für Buckets im selben Konto mit den folgenden Änderungen:

   - Verwenden Sie für alle AWS CLI Befehle, die sich auf Aktivitäten im Quell-Bucket beziehen (z. B. das Erstellen des Quell-Buckets, das Aktivieren der Versionierung und das Erstellen der IAM Rolle), das `acctA` Profil. Verwenden Sie das `acctB` Profil, um den Ziel-Bucket zu erstellen.
   - Stellen Sie sicher, dass die Berechtigungsrichtlinie für die IAM Rolle die Quell- und Ziel-Buckets angibt, die Sie für dieses Beispiel erstellt haben.

3. Fügen Sie in der Konsole die folgende Bucket-Richtlinie für den *`Ziel`*-Bucket hinzu, um dem Eigentümer des *`Quell`*-Buckets die Berechtigung zum Replizieren von Objekten zu erteilen. Detaillierte Anweisungen finden Sie unter Hinzufügen einer Bucket-Richtlinie mit der Amazon-S3-Konsole. Achten Sie darauf, die Richtlinie zu bearbeiten, indem Sie die AWS-Konto ID des Besitzers des Quell-Buckets, den IAM Rollennamen und den Ziel-Bucket-Namen angeben.

   > **Note**
   >
   > Wenn Sie das folgende Beispiel verwenden möchten, ersetzen Sie die *`user input placeholders`* durch Ihre eigenen Informationen. Ersetzen Sie es *`amzn-s3-demo-destination-bucket`* durch den Namen Ihres Ziel-Buckets. Ersetzen Sie *`source-bucket-account-ID`*`:role/service-role/`*`source-account-IAM-role`* den IAM Amazon-Ressourcennamen (ARN) durch die IAM Rolle, die Sie für diese Replikationskonfiguration verwenden.
   > Wenn Sie die IAM Servicerolle manuell erstellt haben, legen Sie den Rollenpfad im IAM ARN as fest`role/service-role/`, wie im folgenden Richtlinienbeispiel gezeigt. Weitere Informationen finden Sie IAMARNsim IAMBenutzerhandbuch.

```
{
   "Version":"2012-10-17",
   "Id":"",
   "Statement":[
      {
         "Sid":"Set-permissions-for-objects",
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::source-bucket-account-ID:role/service-role/source-account-IAM-role"
```

```
         },
         "Action":["s3:ReplicateObject", "s3:ReplicateDelete"],
         "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
      },
      {
         "Sid":"Set permissions on bucket",
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::source-bucket-account-ID:role/service-role/source-
account-IAM-role"
         },
         "Action":["s3:GetBucketVersioning", "s3:PutBucketVersioning"],
         "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket"
      }
   ]
}
```

4. (Optional) Wenn Sie Objekte replizieren, die mit SSE - verschlüsselt sindKMS, muss der Besitzer des KMS Schlüssels dem Besitzer des Quell-Buckets die Erlaubnis zur Verwendung des KMS Schlüssels erteilen. Weitere Informationen finden Sie unter Erteilen von zusätzlichen Berechtigungen für kontenübergreifende Szenarien.
5. (Optional) Bei der Replikation ist der Besitzer des Quellobjekts standardmäßig Eigentümer des Replikats. Wenn Quell- und Ziel-Bucket unterschiedlichen Besitzern gehören AWS-Konten, können Sie optionale Konfigurationseinstellungen hinzufügen, um den Besitz des Replikats auf das zu übertragen AWS-Konto , dem die Ziel-Buckets gehören. Dazu gehört auch die Gewährung der `ObjectOwnerOverrideToBucketOwner`-Berechtigung. Weitere Informationen finden Sie unter Ändern des Replikat-Eigentümers.

Ändern des Replikat-Eigentümers

Bei der Replikation besitzt der Eigentümer des Quellobjekts standardmäßig auch das Replikat. Wenn die Quell- und Ziel-Buckets jedoch unterschiedlichen Besitzern gehören AWS-Konten, sollten Sie möglicherweise den Besitz des Replikats ändern. Möglicherweise möchten Sie die Eigentümerschaft ändern, um den Zugriff auf Objektreplikate einzuschränken. In Ihrer Replikationskonfiguration können Sie optionale Konfigurationseinstellungen hinzufügen, um den Besitz des Replikats auf das Replikat zu übertragen AWS-Konto , dem die Ziel-Buckets gehören.

Gehen Sie wie folgt vor, um den Besitzer des Replikats zu ändern:

- Fügen Sie der Replikationskonfiguration die Option Owner Override hinzu, um Amazon S3 anzuweisen, den Besitz des Replikats zu ändern.
- Erteilen Sie Amazon S3 die `s3:ObjectOwnerOverrideToBucketOwner` Erlaubnis, den Besitz des Replikats zu ändern.
- Fügen Sie die `s3:ObjectOwnerOverrideToBucketOwner` Berechtigung zur Ziel-Bucket-Richtlinie hinzu, um die Änderung des Replikatbesitzes zu ermöglichen. Die `s3:ObjectOwnerOverrideToBucketOwner` Berechtigung ermöglicht es dem Besitzer der Ziel-Buckets, die Inhaberschaft von Objektreplikaten zu akzeptieren.

Weitere Informationen erhalten Sie unter the section called “Überlegungen zur Option zum Außerkraftsetzen von Besitzrechten” und Hinzufügen der Eigentümer-Überschreibungs-Option zur Replikations-Konfiguration. Ein funktionierendes Beispiel mit step-by-step Anweisungen finden Sie unter. Wie ändere ich den Besitzer des Replikats

> **Important**
>
> Anstatt die Option zum Überschreiben des Besitzers zu verwenden, können Sie die erzwungene Einstellung des Bucket-Besitzers für Objekteigentum verwenden. Wenn Sie die Replikation verwenden und Quell- und Ziel-Bucket unterschiedlichen Besitzern gehören AWS-Konten, kann der Bucket-Besitzer des Ziel-Buckets die erzwungene Einstellung des Bucket-Besitzers für Objekteigentümer verwenden, um den Besitz des Replikats auf den AWS-Konto Besitzer des Ziel-Buckets zu ändern. Diese Einstellung deaktiviert Objektzugriffskontrolllisten (). ACLs
> Die Einstellung „Bucket Owner erforced“ ahmt das bestehende Verhalten beim Überschreiben des Besitzers nach, ohne dass eine entsprechende Genehmigung erforderlich ist. `s3:ObjectOwnerOverrideToBucketOwner` Alle Objekte, die mit der erzwungenen Einstellung des Bucket-Eigentümers in den Ziel-Bucket repliziert werden, gehören dem Eigentümer des Ziel-Buckets. Informationen zu Object Ownership finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

Überlegungen zur Option zum Außerkraftsetzen von Besitzrechten

Wenn Sie die Eigentümer-Überschreibungs-Option konfigurieren, berücksichtigen Sie die folgenden Überlegungen:

- Standardmäßig besitzt der Eigentümer des Quellobjekts auch das Replikat. Amazon S3 repliziert die Objektversion und die ACL damit verbundene Version.

  Wenn Sie Ihrer Replikationskonfiguration die Option Owner Override hinzufügen, repliziert Amazon S3 nur die Objektversion, nicht dieACL. Darüber hinaus repliziert Amazon S3 keine nachfolgenden Änderungen am QuellobjektACL. Amazon S3 legt das ACL auf dem Replikat fest, das dem Besitzer des Ziel-Buckets die volle Kontrolle gewährt.
- Wenn Sie eine Replikationskonfiguration aktualisieren, um die Eigentümerüberschreibung zu aktivieren oder zu deaktivieren, tritt das folgende Verhalten auf:
  - Wenn Sie die Eigentümerüberschreibungs-Option zur Replikations-Konfiguration hinzufügen

    Wenn Amazon S3 eine Objektversion repliziert, verwirft es die Version, ACL die dem Quellobjekt zugeordnet ist. Stattdessen legt Amazon S3 das ACL auf dem Replikat fest und gibt dem Besitzer des Ziel-Buckets die volle Kontrolle. Amazon S3 repliziert keine nachfolgenden Änderungen am QuellobjektACL. Diese ACL Änderung gilt jedoch nicht für Objektversionen, die repliziert wurden, bevor Sie die Option zum Überschreiben des Besitzers festgelegt haben. ACLAktualisierungen von Quellobjekten, die repliziert wurden, bevor die Besitzerüberschreibung festgelegt wurde, werden weiterhin repliziert (weil das Objekt und seine Replikate weiterhin denselben Besitzer haben).
  - Wenn Sie die Eigentümer-Überschreibungs-Option aus der Replikations-Konfiguration entfernen

    Amazon S3 repliziert neue Objekte, die im Quell-Bucket und den mit den Ziel-Buckets verknüpften ACLs Buckets erscheinen. Für Objekte, die repliziert wurden, bevor Sie die Eigentümerüberschreibung entfernt haben, repliziert Amazon S3 das nicht, ACLs da die von Amazon S3 vorgenommene Änderung des Objekteigentums weiterhin in Kraft ist. Das heißt, ACLs die Objektversion, die repliziert wurde, als die Eigentümerüberschreibung festgelegt wurde, wird weiterhin nicht repliziert.

Hinzufügen der Eigentümer-Überschreibungs-Option zur Replikations-Konfiguration

> **Warning**
>
> Fügen Sie die Option zum Überschreiben des Besitzers nur hinzu, wenn die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören. AWS-Konten Amazon S3 überprüft nicht, ob die Buckets denselben oder unterschiedlichen Konten gehören. Wenn Sie die Besitzer-Override hinzufügen, obwohl beide Buckets demselben gehören AWS-Konto, wendet Amazon S3 die Besitzer-Override an. Diese Option gewährt dem Besitzer des Ziel-Buckets

volle Rechte und repliziert keine nachfolgenden Aktualisierungen der Zugriffskontrolllisten der Quellobjekte (). ACLs Der Replikatbesitzer kann die mit einem Replikat ACL verknüpften Daten mit einer `PutObjectAcl` Anfrage direkt ändern, jedoch nicht durch Replikation.

Um die Option zur Eigentümer-Überschreibung festzulegen, fügen Sie Folgendes zu jedem `Destination`-Element hinzu:

- Das Element `AccessControlTranslation`, das Amazon S3 anweist, die Replikateigentümerschaft zu ändern
- Das `Account` Element, das den Besitzer AWS-Konto des Ziel-Buckets angibt

```
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    ...
    <Destination>
      ...
      <AccessControlTranslation>
           <Owner>Destination</Owner>
       </AccessControlTranslation>
      <Account>destination-bucket-owner-account-id</Account>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Die folgende Beispiel-Replikationskonfiguration weist Amazon S3 an, Objekte zu replizieren, die *`Steuer`* Geben Sie dem *`amzn-s3-demo-destination-bucket`* Ziel-Bucket ein Präfix ein und ändern Sie den Besitz der Replikate. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
   <Role>arn:aws:iam::account-id:role/role-name</Role>
   <Rule>
      <ID>Rule-1</ID>
      <Priority>1</Priority>
      <Status>Enabled</Status>
      <DeleteMarkerReplication>
         <Status>Disabled</Status>
      </DeleteMarkerReplication>
      <Filter>
```

```
         <Prefix>Tax</Prefix>
      </Filter>
      <Destination>
         <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
         <Account>destination-bucket-owner-account-id</Account>
         <AccessControlTranslation>
            <Owner>Destination</Owner>
         </AccessControlTranslation>
      </Destination>
   </Rule>
</ReplicationConfiguration>
```

Erteilen der Berechtigung zur Änderung der Replikat-Eigentümerschaft an Amazon S3

Gewähren Sie Amazon S3 S3-Berechtigungen zum Ändern des Replikatbesitzes, indem Sie in der Berechtigungsrichtlinie, die der Rolle AWS Identity and Access Management (IAM) zugeordnet ist, die Berechtigung für die `s3:ObjectOwnerOverrideToBucketOwner` Aktion hinzufügen. Diese Rolle haben Sie in der IAM Replikationskonfiguration angegeben, sodass Amazon S3 Objekte in Ihrem Namen übernehmen und replizieren kann. Um das folgende Beispiel zu verwenden, *amzn-s3-demo-destination-bucket* ersetzen Sie es durch den Namen des Ziel-Buckets.

```
...
{
    "Effect":"Allow",
         "Action":[
       "s3:ObjectOwnerOverrideToBucketOwner"
    ],
    "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
}
...
```

Hinzufügen der Berechtigung zur Ziel-Bucket-Richtlinie, um das Ändern der Replikat-Eigentümerschaft zuzulassen

Der Eigentümer des Ziel-Buckets muss dem Eigentümer des Quell-Buckets die Berechtigung zum Ändern der Replikat-Eigentümerschaft erteilen. Der Eigentümer des Ziel-Buckets erteilt dem Eigentümer des Quell-Buckets die Berechtigung für die Aktion `s3:ObjectOwnerOverrideToBucketOwner`. Diese Berechtigung ermöglicht es dem Besitzer des Ziel-Buckets, das Eigentum an den Objektreplikaten zu akzeptieren. Die folgende Beispielanweisung einer Bucket-Richtlinie zeigt, wie dies funktioniert: Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
...
{
    "Sid":"1",
    "Effect":"Allow",
    "Principal":{"AWS":"source-bucket-account-id"},
    "Action":["s3:ObjectOwnerOverrideToBucketOwner"],
    "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
}
...
```

Wie ändere ich den Besitzer des Replikats

Wenn die Quell- und Ziel-Buckets in einer Replikationskonfiguration unterschiedlichen Besitzern gehören AWS-Konten, können Sie Amazon S3 anweisen, den Besitz des Replikats in den Besitz des AWS-Konto Ziel-Buckets zu ändern. Die folgenden Beispiele zeigen, wie Sie die Amazon S3 S3-Konsole, die AWS Command Line Interface (AWS CLI) und die verwenden, AWS SDKs um den Besitz eines Replikats zu ändern.

Verwenden der S3-Konsole

step-by-stepEine Anleitung finden Sie unterKonfiguration der Replikation für Buckets im selben Konto. Dieses Thema enthält Anweisungen zum Einrichten einer Replikationskonfiguration, wenn Quell- und Ziel-Buckets denselben und unterschiedlichen AWS-Konten Besitzern gehören.

Verwenden von AWS CLI

Das folgende Verfahren zeigt, wie Sie den Besitz eines Replikats mithilfe von ändern können. AWS CLI In diesem Verfahren gehen Sie wie folgt vor:

- Erstellen Sie die Quell- und Ziel-Buckets.
- Aktivieren Sie die Versionierung für die Buckets.
- Erstellen Sie eine AWS Identity and Access Management (IAM) -Rolle, die Amazon S3 die Berechtigung erteilt, Objekte zu replizieren.
- Fügen Sie die Replikationskonfiguration zum Quell-Bucket hinzu.
- In der Replikationskonfiguration weisen Sie Amazon S3 an, den Besitz des Replikats zu ändern.
- Sie testen Ihre Replikationskonfiguration.

Um den Besitz des Replikats zu ändern, wenn der Quell- und der Ziel-Bucket different AWS-Konten () gehören AWS CLI

Um die AWS CLI Beispielbefehle in diesem Verfahren zu verwenden, ersetzen Sie die *`user input placeholders`* durch Ihre eigenen Informationen.

1. In diesem Beispiel erstellen Sie die Quell- und Ziel-Buckets in zwei verschiedenen AWS-Konten Buckets. Um mit diesen beiden Konten zu arbeiten, konfigurieren Sie sie AWS CLI mit zwei benannten Profilen. In diesem Beispiel werden Profile mit dem Namen verwendet *`ACCTA`* and *`ACCTB`*, jeweils. Informationen zum Einrichten von Anmeldeinformationsprofilen und zur Verwendung benannter Profile finden Sie unter Konfiguration und Einstellungen für Anmeldeinformationsdateien im AWS Command Line Interface Benutzerhandbuch.

   > **Important**
   >
   > Die Profile, die Sie für dieses Verfahren verwenden, müssen über die erforderlichen Berechtigungen verfügen. In der Replikationskonfiguration geben Sie beispielsweise die IAM Rolle an, die Amazon S3 übernehmen kann. Sie können dies nur tun, wenn das von Ihnen verwendete Profil über die `iam:PassRole` entsprechende Berechtigung verfügt. Wenn Sie Administratoranmeldedaten verwenden, um ein benanntes Profil zu erstellen, können Sie alle Aufgaben in diesem Verfahren ausführen. Weitere Informationen finden Sie im Benutzerhandbuch unter Erteilen von Benutzerberechtigungen zur Übergabe einer Rolle AWS-Service an einen. IAM

2. Erstellen Sie den *`Quell`*-Bucket und aktivieren Sie das Versioning für ihn. In diesem Beispiel wird ein Quell-Bucket mit dem Namen *`amzn-s3-demo-source-bucket`* der Region USA Ost (Nord-Virginia) (`us-east-1`) erstellt.

   ```
   aws s3api create-bucket \
   --bucket amzn-s3-demo-source-bucket \
   --region us-east-1 \
   --profile acctA
   ```

   ```
   aws s3api put-bucket-versioning \
   --bucket amzn-s3-demo-source-bucket \
   --versioning-configuration Status=Enabled \
   --profile acctA
   ```

3. Erstellen Sie einen *Ziel*-Bucket und aktivieren Sie das Versioning für ihn. In diesem Beispiel wird ein Ziel-Bucket mit dem Namen *amzn-s3-demo-destination-bucket* der Region USA West (Oregon) (`us-west-2`) erstellt. Verwenden Sie ein AWS-Konto Profil, das sich von dem unterscheidet, das Sie für den Quell-Bucket verwendet haben.

```
aws s3api create-bucket \
--bucket amzn-s3-demo-destination-bucket \
--region us-west-2 \
--create-bucket-configuration LocationConstraint=us-west-2 \
--profile acctB
```

```
aws s3api put-bucket-versioning \
--bucket amzn-s3-demo-destination-bucket \
--versioning-configuration Status=Enabled \
--profile acctB
```

4. Sie müssen Ihrer *Ziel*-Bucket-Richtlinie Berechtigungen hinzufügen, um eine Änderung der Replikat-Eigentümerschaft zuzulassen.

   a. Speichern Sie die folgende Richtlinie in einer Datei mit dem Namen*destination-bucket-policy*`.json`. Stellen Sie sicher, dass Sie das ersetzen *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "destination_bucket_policy_sid",
      "Principal": {
        "AWS": "source-bucket-owner-account-id"
      },
      "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateDelete",
        "s3:ObjectOwnerOverrideToBucketOwner",
        "s3:ReplicateTags",
        "s3:GetObjectVersionTagging"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
```

```
      ]
    }
  ]
}
```

b. Fügen Sie die vorherige Richtlinie mithilfe des folgenden `put-bucket-policy` Befehls zum Ziel-Bucket hinzu:

```
aws s3api put-bucket-policy --region $ {destination-region} --bucket $ {amzn-s3-demo-destination-bucket} --policy file://destination_bucket_policy.json
```

5. Erstellen Sie eine IAM Rolle. Sie geben diese Rolle in der Replizierungskonfiguration an, die Sie später zum *Quell-Bucket* hinzufügen. Amazon S3 übernimmt diese Rolle, um Objekte in Ihrem Namen zu replizieren. Sie erstellen eine IAM Rolle in zwei Schritten:

   - Erstellen Sie die -Rolle.
   - Fügen Sie eine Berechtigungsrichtlinie zur Rolle hinzu.

   a. Erstellen Sie die IAM Rolle.

      i. Kopieren Sie die folgende Vertrauensrichtlinie und speichern Sie sie in einer Datei mit dem Namen *s3-role-trust-policy*`.json` im aktuellen Verzeichnis auf Ihrem lokalen Computer. Diese Richtlinie erteilt Amazon S3 Berechtigungen für die Übernahme der Rolle.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

      ii. Führen Sie den folgenden AWS CLI `create-role` Befehl aus, um die IAM Rolle zu erstellen:

```
$ aws iam create-role \
--role-name replicationRole \
--assume-role-policy-document file://s3-role-trust-policy.json  \
--profile acctA
```

Notieren Sie sich den Amazon-Ressourcennamen (ARN) der IAM Rolle, die Sie erstellt haben. Sie werden ihn ARN in einem späteren Schritt benötigen.

b. Fügen Sie eine Berechtigungsrichtlinie zur Rolle hinzu.

i. Kopieren Sie die folgende Berechtigungsrichtlinie und speichern Sie sie in einer Datei mit dem Namen *s3-role-perm-pol-changeowner*.json im aktuellen Verzeichnis auf Ihrem lokalen Computer. Diese Zugriffsrichtlinie erteilt Berechtigungen für verschiedene Amazon-S3-Bucket- und -Objektaktionen. In den folgenden Schritten fügen Sie diese Richtlinie der IAM Rolle hinzu, die Sie zuvor erstellt haben.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ListBucket",
            "s3:GetReplicationConfiguration"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
         ]
      },
      {
         "Effect":"Allow",
```

```
            "Action":[
               "s3:ReplicateObject",
               "s3:ReplicateDelete",
               "s3:ObjectOwnerOverrideToBucketOwner",
               "s3:ReplicateTags",
               "s3:GetObjectVersionTagging"
            ],
            "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
         }
      ]
}
```

ii. Führen Sie den folgenden `put-role-policy` Befehl aus, um die vorherige Berechtigungsrichtlinie an die Rolle anzuhängen:

```
$ aws iam put-role-policy \
--role-name replicationRole \
--policy-document file://s3-role-perm-pol-changeowner.json \
--policy-name replicationRolechangeownerPolicy \
--profile acctA
```

6. Fügen Sie Ihrem Quell-Bucket eine Replikations-Konfiguration hinzu.

   a. Das AWS CLI erfordert die Angabe der Replikationskonfiguration alsJSON. Speichern Sie Folgendes JSON in einer Datei mit dem Namen *`replication`*`.json` im aktuellen Verzeichnis auf Ihrem lokalen Computer. In der Konfiguration `AccessControlTranslation` gibt der die Änderung des Besitzers des Replikats vom Besitzer des Quell-Buckets zum Besitzer des Ziel-Buckets an.

```
{
   "Role":"IAM-role-ARN",
   "Rules":[
      {
         "Status":"Enabled",
         "Priority":1,
         "DeleteMarkerReplication":{
            "Status":"Disabled"
         },
         "Filter":{
         },
         "Status":"Enabled",
         "Destination":{
```

```
            "Bucket":"arn:aws:s3:::amzn-s3-demo-destination-bucket",
            "Account":"destination-bucket-owner-account-id",
            "AccessControlTranslation":{
               "Owner":"Destination"
            }
         }
      }
   ]
}
```

b. Bearbeiten Sie das, JSON indem Sie Werte für den Namen des Ziel-Buckets, die Konto-ID des Besitzers des Ziel-Buckets und den *IAM-role-ARN* angeben. Ersetzen *IAM-Rolle-ARN* mit ARN der IAM Rolle, die Sie zuvor erstellt haben. Speichern Sie die Änderungen.

c. Führen Sie den folgenden Befehl aus, um die Replikationskonfiguration zum Quell-Bucket hinzuzufügen:

```
$ aws s3api put-bucket-replication \
--replication-configuration file://replication.json \
--bucket amzn-s3-demo-source-bucket \
--profile acctA
```

7. Testen Sie Ihre Replikationskonfiguration, indem Sie in der Amazon S3 S3-Konsole den Besitz des Replikats überprüfen.

   a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

   b. Fügen Sie Objekte zum *Quell*-Bucket hinzu. Stellen Sie sicher, dass der Ziel-Bucket die Objektreplikate enthält und dass der Eigentümer der Replikate auf den Eigentümer des Ziel-Buckets übergegangen ist. AWS-Konto

Verwenden von AWS SDKs

Ein Codebeispiel zum Hinzufügen einer Replikationskonfiguration finden Sie unterMit dem AWS SDKs. Sie müssen die Replikationskonfiguration entsprechend ändern. Weitere konzeptuelle Informationen finden Sie unter Ändern des Replikat-Eigentümers.

Erfüllung von Compliance-Anforderungen mit S3 Replication Time Control

S3 Replication Time Control (S3RTC) unterstützt Sie bei der Erfüllung von Compliance- oder Geschäftsanforderungen für die Datenreplikation und bietet Einblick in die Amazon S3 S3-

Replikationszeiten. S3 RTC repliziert die meisten Objekte, die Sie auf Amazon S3 hochladen, innerhalb von Sekunden und 99,99 Prozent dieser Objekte innerhalb von 15 Minuten.

Standardmäßig RTC bietet S3 zwei Möglichkeiten, den Fortschritt der Replikation zu verfolgen:

- S3-Replikationsmetriken — Sie können S3-Replikationsmetriken verwenden, um die Gesamtzahl der API S3-Operationen, deren Replikation aussteht, die Gesamtgröße der zu replizierenden Objekte, die maximale Replikationszeit in die Zielregion und die Gesamtzahl der Operationen, bei denen die Replikation fehlgeschlagen ist, zu überwachen. Anschließend können Sie jedes Dataset, das Sie separat replizieren, überwachen. Sie können S3-Replikationsmetriken auch unabhängig von S3 aktivierenRTC. Weitere Informationen finden Sie unter the section called “Verwendung von S3-Replikationsmetriken”.

  Replikationsregeln, bei denen S3 Replication Time Control (S3RTC) aktiviert ist, veröffentlichen S3-Replikationsmetriken. Replikationsmetriken sind innerhalb von 15 Minuten nach Aktivierung von S3 verfügbarRTC. Replikationsmetriken sind über die Amazon S3-Konsole, Amazon S3API, the AWS SDKs, the AWS Command Line Interface (AWS CLI) und Amazon verfügbar CloudWatch. Weitere Informationen zu CloudWatch Metriken finden Sie unterMetriken mit Amazon überwachen CloudWatch. Weitere Informationen zum Anzeigen von Replikationsmetriken über die Amazon S3 S3-Konsole finden Sie unterAnzeigen von Replikationsmetriken.

  S3-Replikationsmetriken werden zum gleichen Tarif wie CloudWatch benutzerdefinierte Amazon-Metriken abgerechnet. Weitere Informationen finden Sie unter CloudWatchAmazon-Preise.

- Amazon S3 S3-Ereignisbenachrichtigungen — S3 RTC stellt `OperationMissedThreshold` `OperationReplicatedAfterThreshold` Ereignisse bereit, die den Bucket-Besitzer benachrichtigen, wenn die Objektreplikation den Schwellenwert von 15 Minuten überschreitet oder überschreitet. Mit S3 können Sie Amazon S3 Event Notifications in seltenen Fällen benachrichtigenRTC, wenn Objekte nicht innerhalb von 15 Minuten repliziert werden und wenn diese Objekte nach dem 15-minütigen Schwellenwert repliziert werden.

  Replikationsereignisse sind innerhalb von 15 Minuten nach der Aktivierung von S3 verfügbar. RTC Amazon S3 S3-Ereignisbenachrichtigungen sind über Amazon SQSSNS, Amazon oder erhältlich AWS Lambda. Weitere Informationen finden Sie unter the section called “Erhalten von Benachrichtigungen über Replikations-Fehlerereignisse”.

Bewährte Methoden und Richtlinien für S3 RTC

Wenn Sie Daten in Amazon S3 mit aktiviertem S3 Replication Time Control (S3RTC) replizieren, befolgen Sie diese Best-Practice-Richtlinien, um die Replikationsleistung für Ihre Workloads zu optimieren.

Themen

- Leistungsrichtlinien für Amazon-S3-Replikation and -Anforderungsraten
- Schätzen der Replikationsanforderungsraten
- Überschreitung der RTC S3-Datenübertragungsraten
- AWS KMS Anforderungsraten für die Replikation verschlüsselter Objekte

Leistungsrichtlinien für Amazon-S3-Replikation and -Anforderungsraten

Wenn Speicherinhalte zu Amazon S3 hochgeladen oder von dort abgerufen werden, können Ihre Anwendungen Tausende von Transaktionen pro Sekunde bei der Anforderungsleistung erhalten. Beispielsweise kann eine Anwendung mindestens 3.500`PUT`/`DELETE`oder 5.500 `COPY` `POST`/`GET`/`HEAD`Anfragen pro Sekunde pro Präfix in einem S3-Bucket erreichen, einschließlich der Anfragen, die S3 Replication in Ihrem Namen stellt. Es gibt keine Einschränkungen für die Anzahl der Präfixe in einem Bucket. Sie können Ihre Lese- und Schreibleistung steigern, indem Sie Lesevorgänge parallelisieren. Wenn Sie beispielsweise 10 Präfixe in einem S3-Bucket erstellen, um Lesevorgänge zu parallelisieren, können Sie Ihre Leseleistung auf 55.000 Leseanfragen pro Sekunde skalieren.

Amazon S3 skaliert automatisch als Reaktion auf anhaltende Anforderungsraten, die über diesen Richtlinien liegen, oder auf anhaltende Anforderungsraten, die gleichzeitig mit `LIST` Anfragen einhergehen. Während Amazon S3 intern für die neue Anforderungsrate optimiert, erhalten Sie möglicherweise vorübergehend HTTP 503 Antworten auf Anfragen, bis die Optimierung abgeschlossen ist. Dieses Verhalten kann auftreten, wenn die Anzahl der Anfragen pro Sekunde steigt oder wenn Sie S3 zum ersten Mal aktivierenRTC. Während dieser Zeiträume kann sich die Replikationslatenz erhöhen. Das RTC S3-Service Level Agreement (SLA) gilt nicht für Zeiträume, in denen die Amazon S3 S3-Leistungsrichtlinien für Anfragen pro Sekunde überschritten werden.

S3 gilt RTC SLA auch nicht in Zeiträumen, in denen Ihre Replikationsdatenübertragungsrate das Standardkontingent von 1 Gigabit pro Sekunde (Gbit/s) überschreitet. Wenn Sie davon ausgehen, dass Ihre Replikationsübertragungsrate 1 Gbit/s überschreiten wird, können Sie sich an das AWS Support Center wenden oder mithilfe von Service Quotas eine Erhöhung Ihres Kontingents für die Replikationsübertragungsrate beantragen.

Schätzen der Replikationsanforderungsraten

Ihre gesamte Anforderungsrate, einschließlich der Anfragen, die Amazon S3 S3-Replikation in Ihrem Namen stellt, muss den Amazon S3 S3-Anforderungsratenrichtlinien sowohl für die Replikationsquelle als auch für die Ziel-Buckets entsprechen. Für jedes replizierte Objekt stellt die Amazon S3 S3-Replikation bis zu fünf `GET` `HEAD` /-Anfragen und eine `PUT` Anfrage an den Quell-Bucket und eine `PUT` Anfrage an jeden Ziel-Bucket.

Wenn Sie beispielsweise erwarten, 100 Objekte pro Sekunde zu replizieren, kann die Amazon S3 S3-Replikation weitere 100 `PUT` Anfragen in Ihrem Namen ausführen, was insgesamt 200 `PUT` Anfragen pro Sekunde an den S3-Quell-Bucket ergibt. Die Amazon S3 S3-Replikation kann auch bis zu 500`GET`//`HEAD`Anfragen ausführen (5`GET`//`HEAD`Anfragen für jedes Objekt, das repliziert wird).

> **Note**
>
> Es fallen Kosten für nur eine `PUT` Anfrage pro repliziertem Objekt an. Weitere Informationen finden Sie in den Preisinformationen zur Replikation in Amazon S3 FAQs.

Überschreitung der RTC S3-Datenübertragungsraten

Wenn Sie davon ausgehen, dass Ihre RTC S3-Datenübertragungsrate das Standardkontingent von 1 Gbit/s überschreitet, wenden Sie sich an das AWS Support Center oder verwenden Sie Service Quotas, um eine Erhöhung Ihres Kontingents für die Replikationsübertragungsrate zu beantragen.

AWS KMS Anforderungsraten für die Replikation verschlüsselter Objekte

Wenn Sie Objekte replizieren, die mit serverseitiger Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-KMS) verschlüsselt sind, gelten Kontingente für AWS KMS Anfragen pro Sekunde. AWS KMS lehnt möglicherweise eine ansonsten gültige Anfrage ab, weil Ihre Anforderungsrate das Kontingent für die Anzahl der Anfragen pro Sekunde überschreitet. Wenn eine Anfrage gedrosselt wird, wird ein AWS KMS `ThrottlingException` Fehler zurückgegeben. Das AWS KMS Anforderungsratenkontingent gilt für Anfragen, die Sie direkt stellen, und für Anfragen, die von der Amazon S3 S3-Replikation in Ihrem Namen gestellt werden.

Wenn Sie beispielsweise damit rechnen, 1.000 Objekte pro Sekunde zu replizieren, können Sie 2.000 Anfragen von Ihrer AWS KMS Quote für Anfragen abziehen. Die daraus resultierende Anforderungsrate pro Sekunde ist für Ihre AWS KMS Workloads ohne Replikation verfügbar. Sie können die AWS KMS Anforderungsmetriken in Amazon verwenden CloudWatch, um die gesamte AWS KMS Anforderungsrate auf Ihrem zu überwachen AWS-Konto.

Um eine Erhöhung Ihres Kontingents für AWS KMS Anfragen pro Sekunde zu beantragen, wenden Sie sich an das AWS Support Center oder nutzen Sie Service Quotas.

Aktivieren der S3 Replication Time Control

Sie können beginnen, S3 Replication Time Control (S3RTC) mit einer neuen oder vorhandenen Replikationsregel zu verwenden. Sie können wählen, ob Sie Ihre Replikationsregel auf einen ganzen Bucket oder auf Objekte mit einem bestimmten Präfix oder Tag anwenden möchten. Wenn Sie S3 aktivierenRTC, werden S3-Replikationsmetriken auch für Ihre Replikationsregel aktiviert.

Sie können S3 RTC mithilfe der Amazon S3-Konsole, Amazon S3 API AWS SDKs, der und der AWS Command Line Interface (AWS CLI) konfigurieren.

Verwenden der S3-Konsole

step-by-stepAnweisungen finden Sie unterKonfiguration der Replikation für Buckets im selben Konto. Dieses Thema enthält Anweisungen zur Aktivierung von S3 RTC in Ihrer Replikationskonfiguration, wenn die Quell- und Ziel-Buckets denselben und unterschiedlichen AWS-Konten Besitzern gehören.

Verwenden von AWS CLI

Um Objekte mit RTC aktiviertem S3 AWS CLI zu replizieren, erstellen Sie Buckets, aktivieren die Versionierung für die Buckets, erstellen eine IAM Rolle, die Amazon S3 die Berechtigung erteilt, Objekte zu replizieren, und fügen die Replikationskonfiguration zum Quell-Bucket hinzu. Für die Replikationskonfiguration muss S3 RTC aktiviert sein, wie im folgenden Beispiel gezeigt.

step-by-stepAnweisungen zum Einrichten Ihrer Replikationskonfiguration mithilfe von finden Sie unterKonfiguration der Replikation für Buckets im selben Konto. AWS CLI

Das folgende Beispiel für eine Replikationskonfiguration aktiviert und legt die `EventThreshold` Werte `ReplicationTime` und für eine Replikationsregel fest. Wenn Sie diese Werte aktivieren und festlegen, wird S3 RTC für die Regel aktiviert.

```
{
    "Rules": [
        {
            "Status": "Enabled",
            "Filter": {
                "Prefix": "Tax"
            },
            "DeleteMarkerReplication": {
                "Status": "Disabled"
```

```
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::amzn-s3-demo-destination-bucket",
                "Metrics": {
                    "Status": "Enabled",
                    "EventThreshold": {
                        "Minutes": 15
                    }
                },
                "ReplicationTime": {
                    "Status": "Enabled",
                    "Time": {
                        "Minutes": 15
                    }
                }
            },
            "Priority": 1
        }
    ],
    "Role": "IAM-Role-ARN"
}
```

> **Important**
>
> `Metrics:EventThreshold:Minutes`und `ReplicationTime:Time:Minutes` kann nur einen gültigen Wert haben`15`.

Verwendung des AWS SDK for Java

Das folgende Java-Beispiel fügt eine Replikationskonfiguration hinzu, bei der S3 Replication Time Control (S3RTC) aktiviert ist.

```
import software.amazon.awssdk.auth.credentials.AwsBasicCredentials;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.model.DeleteMarkerReplication;
import software.amazon.awssdk.services.s3.model.Destination;
import software.amazon.awssdk.services.s3.model.Metrics;
import software.amazon.awssdk.services.s3.model.MetricsStatus;
import software.amazon.awssdk.services.s3.model.PutBucketReplicationRequest;
import software.amazon.awssdk.services.s3.model.ReplicationConfiguration;
import software.amazon.awssdk.services.s3.model.ReplicationRule;
```

```
import software.amazon.awssdk.services.s3.model.ReplicationRuleFilter;
import software.amazon.awssdk.services.s3.model.ReplicationTime;
import software.amazon.awssdk.services.s3.model.ReplicationTimeStatus;
import software.amazon.awssdk.services.s3.model.ReplicationTimeValue;

public class Main {

  public static void main(String[] args) {
    S3Client s3 = S3Client.builder()
      .region(Region.US_EAST_1)
      .credentialsProvider(() -> AwsBasicCredentials.create(
          "AWS_ACCESS_KEY_ID",
          "AWS_SECRET_ACCESS_KEY")
      )
      .build();

    ReplicationConfiguration replicationConfig = ReplicationConfiguration
      .builder()
      .rules(
          ReplicationRule
            .builder()
            .status("Enabled")
            .priority(1)
            .deleteMarkerReplication(
                DeleteMarkerReplication
                    .builder()
                    .status("Disabled")
                    .build()
            )
            .destination(
                Destination
                    .builder()
                    .bucket("destination_bucket_arn")
                    .replicationTime(
                        ReplicationTime.builder().time(
                            ReplicationTimeValue.builder().minutes(15).build()
                        ).status(
                            ReplicationTimeStatus.ENABLED
                        ).build()
                    )
                    .metrics(
                        Metrics.builder().eventThreshold(
                            ReplicationTimeValue.builder().minutes(15).build()
                        ).status(
```

```
                            MetricsStatus.ENABLED
                        ).build()
                    )
                    .build()
            )
            .filter(
                ReplicationRuleFilter
                    .builder()
                    .prefix("testtest")
                    .build()
            )
        .build())
        .role("role_arn")
        .build();

    // Put replication configuration
    PutBucketReplicationRequest putBucketReplicationRequest =
 PutBucketReplicationRequest
      .builder()
      .bucket("source_bucket")
      .replicationConfiguration(replicationConfig)
      .build();

    s3.putBucketReplication(putBucketReplicationRequest);
  }
}
```

Replikation verschlüsselter Objekte (SSE-S3, SSE -, DSSE - KMSKMS, SSE -C)

> **Important**
>
> Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist in den AWS CloudTrail Protokollen, im S3-Inventar, in der S3-Speicherlinse, in der Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header im AWS Command Line Interface und AWS SDKs verfügbar. Weitere Informationen finden Sie unter Standardverschlüsselung FAQ.

Bei der Replikation von Objekten, die mit der serverseitigen Verschlüsselung verschlüsselt wurden, sind einige besondere Punkte zu beachten. Amazon S3 unterstützt die folgenden Arten von serverseitiger Verschlüsselung:

- Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3)
- Serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln (-) SSE KMS
- Zweischichtige serverseitige Verschlüsselung mit AWS KMS Schlüsseln (-) DSSE KMS
- Serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE

Weitere Informationen zur serverseitigen Verschlüsselung finden Sie unter the section called “Server-side encryption”.

In diesem Thema werden die Berechtigungen erläutert, die Sie benötigen, um Amazon S3 anzuweisen, Objekte zu replizieren, die mithilfe serverseitiger Verschlüsselung verschlüsselt wurden. Dieses Thema enthält auch zusätzliche Konfigurationselemente, die Sie hinzufügen können, sowie Beispielrichtlinien AWS Identity and Access Management (IAM), die die erforderlichen Berechtigungen für die Replikation verschlüsselter Objekte gewähren.

Ein Beispiel mit step-by-step Anweisungen finden Sie unterReplikation für verschlüsselte Objekte aktivieren. Weitere Informationen zum Erstellen einer Replikationskonfiguration finden Sie unter Objekte innerhalb und zwischen Regionen replizieren.

> **Note**
>
> Sie können Multiregion AWS KMS keys in Amazon S3 verwenden. Amazon S3 behandelt jedoch derzeit Multi-Regions-Schlüssel wie Einzel-Regions-Schlüssel und verwendet nicht die Multi-Regions-Funktionen des Schlüssels. Weitere Informationen finden Sie unter Verwenden von Schlüsseln für mehrere Regionen im AWS Key Management Service Entwicklerhandbuch.

Themen

- So wirkt sich die Standard-Bucket-Verschlüsselung auf die Replikation aus
- Replizieren von Objekten, die mit -C verschlüsselt wurden SSE
- Replizieren von Objekten, die mit SSE -S3KMS, SSE - oder - verschlüsselt wurden DSSE KMS

- Replikation für verschlüsselte Objekte aktivieren

So wirkt sich die Standard-Bucket-Verschlüsselung auf die Replikation aus

Wenn Sie die Standard-Verschlüsselung für einen Replikations-Ziel-Bucket aktivieren, gilt das folgende Verschlüsselungsverhalten:

- Wenn Objekte im Quell-Bucket nicht verschlüsselt sind, werden die Replikatobjekte im Ziel-Bucket mithilfe der Einstellungen der Standard-Verschlüsselung des Ziel-Buckets verschlüsselt. Infolgedessen unterscheiden sich die Entity-Tags (ETags) der Quellobjekte von denen ETags der Replikatobjekte. Wenn Sie Anwendungen haben, die dies verwendenETags, müssen Sie diese Anwendungen aktualisieren, um diesem Unterschied Rechnung zu tragen.
- Wenn Objekte im Quell-Bucket mit serverseitiger Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3), serverseitiger Verschlüsselung mit AWS Key Management Service () -Schlüsseln (SSE-AWS KMS KMS) oder zweischichtiger serverseitiger Verschlüsselung mit AWS KMS Schlüsseln (DSSE-KMS) verschlüsselt werden, verwenden die Replikatobjekte im Ziel-Bucket denselben Verschlüsselungstyp wie die Quellobjekte. Die Einstellungen der Standard-Verschlüsselung des Ziel-Buckets werden nicht verwendet.

Replizieren von Objekten, die mit -C verschlüsselt wurden SSE

Durch die serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (SSE-C) können Sie Ihre eigenen proprietären Verschlüsselungsschlüssel verwalten. Mit SSE -C verwalten Sie die Schlüssel, während Amazon S3 den Verschlüsselungs- und Entschlüsselungsprozess verwaltet. Sie müssen einen Verschlüsselungsschlüssel als Teil Ihrer Anfrage angeben, brauchen aber keinen Code zu schreiben, um die Objektverschlüsselung oder -entschlüsselung durchzuführen. Wenn Sie ein Objekt hochladen, verschlüsselt Amazon S3 das Objekt mithilfe des von Ihnen angegebenen Schlüssels. Amazon S3 löscht diesen Schlüssel dann aus dem Speicher. Wenn Sie ein Objekt abrufen, müssen Sie denselben Verschlüsselungsschlüssel als Teil Ihrer Anfrage angeben. Weitere Informationen finden Sie unter the section called “Vom Kunden bereitgestellte Verschlüsselungsschlüssel (-C) SSE”.

Die S3-Replikation unterstützt Objekte, die mit SSE -C verschlüsselt sind. Sie können die SSE -C-Objektreplikation in der Amazon S3 S3-Konsole oder mit dem auf die AWS SDKs gleiche Weise konfigurieren, wie Sie die Replikation für unverschlüsselte Objekte konfigurieren. Es gibt keine zusätzlichen SSE -C-Berechtigungen, die über das hinausgehen, was derzeit für die Replikation erforderlich ist.

S3 Replication repliziert automatisch neu hochgeladene SSE -C-verschlüsselte Objekte, sofern sie geeignet sind, wie in Ihrer S3-Replikationskonfiguration angegeben. Verwenden Sie die S3-Batchreplikation, um vorhandene Objekte in Ihren Buckets zu replizieren. Weitere Informationen zum Replizieren von Objekten finden Sie unter the section called “Einrichtung der Live-Replikation” und the section called “Replizieren vorhandener Objekte”.

Für die Replikation SSE von -C-Objekten fallen keine zusätzlichen Gebühren an. Einzelheiten zu den Replikationspreisen finden Sie unter Amazon S3 S3-Preise.

Replizieren von Objekten, die mit SSE -S3KMS, SSE - oder - verschlüsselt wurden DSSE KMS

Standardmäßig repliziert Amazon S3 keine Objekte, die mit SSE - KMS oder DSSE - KMS verschlüsselt sind. In diesem Abschnitt werden die zusätzlichen Konfigurationselemente erörtert, die Sie hinzufügen können, um Amazon S3 anzuweisen, diese Objekte zu replizieren.

Ein Beispiel mit step-by-step Anweisungen finden Sie unterReplikation für verschlüsselte Objekte aktivieren. Weitere Informationen zum Erstellen einer Replikationskonfiguration finden Sie unter Objekte innerhalb und zwischen Regionen replizieren.

Angeben zusätzlicher Informationen in der Replikations-Konfiguration

In der Replikations-Konfiguration machen Sie Folgendes:

- Fügen Sie in dem `Destination` Element in Ihrer Replikationskonfiguration die ID des symmetrischen, vom AWS KMS Kunden verwalteten Schlüssels hinzu, den Amazon S3 zur Verschlüsselung von Objektreplikaten verwenden soll, wie in der folgenden Beispiel-Replikationskonfiguration gezeigt.
- Entscheiden Sie sich ausdrücklich dafür, indem Sie die Replikation von Objekten aktivieren, die mithilfe von KMS Schlüsseln (SSE- KMS oder DSSE -) verschlüsselt wurden. KMS Fügen Sie dazu das `SourceSelectionCriteria`-Element hinzu, wie im folgenden Beispiel einer Replikationskonfiguration dargestellt.

```
<ReplicationConfiguration>
   <Rule>
      ...
      <SourceSelectionCriteria>
         <SseKmsEncryptedObjects>
           <Status>Enabled</Status>
         </SseKmsEncryptedObjects>
```

```
      </SourceSelectionCriteria>

      <Destination>
          ...
          <EncryptionConfiguration>
             <ReplicaKmsKeyID>AWS KMS key ARN or Key Alias ARN that's in the same AWS-
Region as the destination bucket.</ReplicaKmsKeyID>
          </EncryptionConfiguration>
       </Destination>
      ...
   </Rule>
</ReplicationConfiguration>
```

> **Important**
>
> - Der KMS Schlüssel muss im selben Ordner AWS-Region wie der Ziel-Bucket erstellt worden sein.
> - Der KMS Schlüssel muss gültig sein. Die `PutBucketReplication` API Operation überprüft nicht die Gültigkeit von KMS Schlüsseln. Wenn Sie einen ungültigen KMS Schlüssel verwenden, erhalten Sie als Antwort den HTTP `200 OK` Statuscode, aber die Replikation schlägt fehl.

Das folgende Beispiel zeigt eine Replikationskonfiguration, die optionale Konfigurationselemente enthält. Diese Replikations-Konfiguration weist eine Regel auf. Die Regel wird auf alle Objekte mit dem Schlüsselpräfix *Tax* angewendet. Amazon S3 verwendet die angegebene AWS KMS key ID, um diese Objektreplikate zu verschlüsseln.

```
<?xml version="1.0" encoding="UTF-8"?>
<ReplicationConfiguration>
   <Role>arn:aws:iam::account-id:role/role-name</Role>
   <Rule>
      <ID>Rule-1</ID>
      <Priority>1</Priority>
      <Status>Enabled</Status>
      <DeleteMarkerReplication>
         <Status>Disabled</Status>
      </DeleteMarkerReplication>
      <Filter>
         <Prefix>Tax</Prefix>
```

```
      </Filter>
      <Destination>
         <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
         <EncryptionConfiguration>
            <ReplicaKmsKeyID>AWS KMS key ARN or Key Alias ARN that's in the same AWS-
Region as the destination bucket.</ReplicaKmsKeyID>
         </EncryptionConfiguration>
      </Destination>
      <SourceSelectionCriteria>
         <SseKmsEncryptedObjects>
            <Status>Enabled</Status>
         </SseKmsEncryptedObjects>
      </SourceSelectionCriteria>
   </Rule>
</ReplicationConfiguration>
```

Erteilen zusätzlicher Berechtigungen für die Rolle IAM

Um Objekte zu replizieren, die im Ruhezustand mit SSE -S3, - oder SSE DSSE - verschlüsselt wurden KMSKMS, gewähren Sie der Rolle AWS Identity and Access Management (IAM), die Sie in der Replikationskonfiguration angeben, die folgenden zusätzlichen Berechtigungen. Sie gewähren diese Berechtigungen, indem Sie die der Rolle zugeordnete Berechtigungsrichtlinie aktualisieren. IAM

- **`s3:GetObjectVersionForReplication`**Aktion für Quellobjekte — Diese Aktion ermöglicht Amazon S3, sowohl unverschlüsselte Objekte als auch Objekte zu replizieren, die mit serverseitiger Verschlüsselung mithilfe von SSE -S3, SSE - oder - KMS erstellt wurden. DSSE KMS

  > **Note**
  >
  > Wir empfehlen, dass Sie die Aktion `s3:GetObjectVersionForReplication` statt der Aktion `s3:GetObjectVersion` verwenden, da `s3:GetObjectVersionForReplication` Amazon S3 nur die minimalen Berechtigungen bereitstellt, die für eine Replikation nötig sind. Darüber hinaus ermöglicht die `s3:GetObjectVersion` Aktion die Replikation von unverschlüsselten und SSE -S3-verschlüsselten Objekten, jedoch nicht von Objekten, die mithilfe von Schlüsseln (- oder -) verschlüsselt wurden. KMS SSE KMS DSSE KMS

- **`kms:Decrypt`**und **`kms:Encrypt`** AWS KMS Aktionen für die Schlüssel KMS
  - Sie müssen `kms:Decrypt`-Berechtigungen für den AWS KMS key gewähren, der zum Entschlüsseln des Quellobjekts verwendet wird.

  - Sie müssen `kms:Encrypt`-Berechtigungen für den AWS KMS key gewähren, der zum Verschlüsseln des Objektreplikats verwendet wird.
- **`kms:GenerateDataKey`**Aktion zum Replizieren von Klartextobjekten — Wenn Sie Klartextobjekte in einen Bucket replizieren, bei dem SSE die KMS Verschlüsselung standardmäßig aktiviert KMS oder DSSE aktiviert ist, müssen Sie die `kms:GenerateDataKey` Berechtigung für den Zielverschlüsselungskontext und den KMS Schlüssel in die Richtlinie aufnehmen. IAM

Wir empfehlen, diese Berechtigungen mithilfe von Bedingungsschlüsseln nur auf die Ziel-Buckets und -Objekte zu beschränken. AWS KMS Derjenige, dem die IAM Rolle gehört AWS-Konto , muss über Berechtigungen für die Schlüssel `kms:Encrypt` und `kms:Decrypt` Aktionen für die KMS Schlüssel verfügen, die in der Richtlinie aufgeführt sind. Wenn die KMS Schlüssel einer anderen Person gehören AWS-Konto, muss der Besitzer der KMS Schlüssel diese Berechtigungen demjenigen gewähren AWS-Konto , dem die IAM Rolle gehört. Weitere Informationen zur Verwaltung des Zugriffs auf diese KMS Schlüssel finden Sie unter IAM Richtlinien verwenden mit AWS KMS im AWS Key Management Service Entwicklerhandbuch.

S3 Bucket-Schlüssel und Replikation

Um die Replikation mit einem S3-Bucket-Schlüssel zu verwenden, muss die AWS KMS key Richtlinie für den KMS Schlüssel, der zum Verschlüsseln des Objektreplikats verwendet wird, die `kms:Decrypt` Berechtigung für den aufrufenden Prinzipal enthalten. Der Aufruf zur `kms:Decrypt` Überprüfung der Integrität des S3-Bucket-Schlüssels, bevor er verwendet wird. Weitere Informationen finden Sie unter Verwenden eines S3-Bucket-Schlüssels mit Replikation.

Wenn ein S3-Bucket Key für den Quell- oder Ziel-Bucket aktiviert ist, ist der Verschlüsselungskontext der Amazon-Ressourcenname (ARN) des Buckets, nicht der des Objekts ARN (zum Beispiel`arn:aws:s3:::`*`bucket_ARN`*). Sie müssen Ihre IAM Richtlinien aktualisieren, um den Bucket ARN für den Verschlüsselungskontext verwenden zu können:

```
"kms:EncryptionContext:aws:s3:arn": [
"arn:aws:s3:::bucket_ARN"
]
```

Weitere Informationen finden Sie unter Verschlüsselungskontext (`x-amz-server-side-encryption-context`) (im Abschnitt „Verwenden von RESTAPI“) undÄnderungen, die Sie vor dem Aktivieren eines S3-Bucket-Schlüssels beachten sollten.

Beispielrichtlinien: Verwendung von SSE -S3 und SSE - KMS mit Replikation

Die folgenden IAM Beispielrichtlinien zeigen Anweisungen für die Verwendung von SSE -S3 und SSE - KMS bei der Replikation.

Example — Verwenden von SSE - KMS mit separaten Ziel-Buckets

Die folgende Beispielrichtlinie zeigt Anweisungen zur Verwendung von SSE - KMS mit separaten Ziel-Buckets.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action": ["kms:Decrypt"],
         "Effect": "Allow",
         "Condition": {
             "StringLike": {
                  "kms:ViaService": "s3.source-bucket-region.amazonaws.com",
                  "kms:EncryptionContext:aws:s3:arn": [
                      "arn:aws:s3:::amzn-s3-demo-source-bucket/key-prefix1*"
                  ]
             }
          },
          "Resource": [
             "List of AWS KMS key ARNs that are used to encrypt source objects."
          ]
      },
      {
         "Action": ["kms:Encrypt"],
         "Effect": "Allow",
         "Condition": {
            "StringLike": {
               "kms:ViaService": "s3.destination-bucket-1-region.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn": [
                  "arn:aws:s3:::amzn-s3-demo-destination-bucket1/key-prefix1*"
               ]
         }
         },
         "Resource": [
            "AWS KMS key ARNs (in the same AWS-Region as destination bucket 1). Used to
 encrypt object replicas created in destination bucket 1."
         ]
      },
```

```
      {
         "Action": ["kms:Encrypt"],
         "Effect": "Allow",
         "Condition": {
            "StringLike": {
               "kms:ViaService": "s3.destination-bucket-2-region.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn": [
                  "arn:aws:s3:::amzn-s3-demo-destination-bucket2/key-prefix1*"
               ]
         }
         },
         "Resource": [
            "AWS KMS key ARNs (in the same AWS-Region as destination bucket 2). Used to
 encrypt object replicas created in destination bucket 2."
         ]
      }
   ]
}
```

Example — Replizieren von Objekten, die mit SSE -S3 und - erstellt wurden SSE KMS

Im Folgenden finden Sie eine vollständige IAM Richtlinie, die die erforderlichen Berechtigungen zum Replizieren von unverschlüsselten Objekten, Objekten, die mit SSE -S3 erstellt wurden, und Objekten, die mit - erstellt wurden, gewährt. SSE KMS

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl"
```

```
        ],
        "Resource":[
           "arn:aws:s3:::amzn-s3-demo-source-bucket/key-prefix1*"
        ]
     },
     {
        "Effect":"Allow",
        "Action":[
           "s3:ReplicateObject",
           "s3:ReplicateDelete"
        ],
        "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/key-prefix1*"
     },
     {
        "Action":[
           "kms:Decrypt"
        ],
        "Effect":"Allow",
        "Condition":{
           "StringLike":{
              "kms:ViaService":"s3.source-bucket-region.amazonaws.com",
              "kms:EncryptionContext:aws:s3:arn":[
                 "arn:aws:s3:::amzn-s3-demo-source-bucket/key-prefix1*"
              ]
           }
        },
        "Resource":[
          "List of the AWS KMS key ARNs that are used to encrypt source objects."
        ]
     },
     {
        "Action":[
           "kms:Encrypt"
        ],
        "Effect":"Allow",
        "Condition":{
           "StringLike":{
              "kms:ViaService":"s3.destination-bucket-region.amazonaws.com",
              "kms:EncryptionContext:aws:s3:arn":[
                 "arn:aws:s3:::amzn-s3-demo-destination-bucket/prefix1*"
              ]
           }
        },
        "Resource":[
```

```
            "AWS KMS key ARNs (in the same AWS-Region as the destination bucket) to use
 for encrypting object replicas"
         ]
      }
   ]
}
```

Example – Replizieren von Objekten mit S3-Bucket-Schlüsseln

Im Folgenden finden Sie eine vollständige IAM Richtlinie, die die erforderlichen Berechtigungen zum Replizieren von Objekten mit S3-Bucket-Keys gewährt.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:ListBucket"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket/key-prefix1*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:ReplicateObject",
            "s3:ReplicateDelete"
         ],
         "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/key-prefix1*"
      },
```

```
      {
         "Action":[
            "kms:Decrypt"
         ],
         "Effect":"Allow",
         "Condition":{
            "StringLike":{
               "kms:ViaService":"s3.source-bucket-region.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn":[
                  "arn:aws:s3:::amzn-s3-demo-source-bucket"
               ]
            }
         },
         "Resource":[
           "List of the AWS KMS key ARNs that are used to encrypt source objects."
         ]
      },
      {
         "Action":[
            "kms:Encrypt"
         ],
         "Effect":"Allow",
         "Condition":{
            "StringLike":{
               "kms:ViaService":"s3.destination-bucket-region.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn":[
                  "arn:aws:s3:::amzn-s3-demo-destination-bucket"
               ]
            }
         },
         "Resource":[
            "AWS KMS key ARNs (in the same AWS-Region as the destination bucket) to use
 for encrypting object replicas"
         ]
      }
   ]
}
```

Erteilen von zusätzlichen Berechtigungen für kontenübergreifende Szenarien

In einem kontoübergreifenden Szenario, in dem die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören, können Sie einen KMS Schlüssel verwenden AWS-Konten, um Objektreplikate

zu verschlüsseln. Der KMS Schlüsselbesitzer muss dem Besitzer des Quell-Buckets jedoch die Erlaubnis zur Verwendung des Schlüssels erteilen. KMS

> **Note**
>
> Wenn Sie KMS Daten kontoübergreifend SSE replizieren müssen, muss Ihre Replikationsregel einen vom Kunden verwalteten Schlüssel AWS KMS für das Zielkonto angeben. Von AWS verwaltete Schlüsselerlauben keine kontoübergreifende Verwendung und können daher nicht für die kontenübergreifende Replikation verwendet werden.

Um dem Besitzer des Quell-Buckets die Erlaubnis zur Verwendung des KMS Schlüssels zu erteilen (Konsole)AWS KMS

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS KMS Konsole unter https://console.aws.amazon.com/kms.
2. Um das zu ändern AWS-Region, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
3. Zum Anzeigen der Schlüssel in Ihrem Konto, die Sie erstellen und verwalten, wählen Sie im Navigationsbereich Customer managed keys (Vom Kunden verwaltete Schlüssel) aus.
4. Wählen Sie den KMS-Schlüssel aus.
5. Wählen Sie unter Allgemeine Konfiguration den Tab Schlüsselrichtlinie aus.
6. Scrollen Sie nach unten zu Andere. AWS-Konten
7. Wählen Sie Andere hinzufügen AWS-Konten.

   Das AWS-Konten Dialogfeld „Andere“ wird angezeigt.
8. Wählen Sie im Dialogfeld „Weitere hinzufügen“ aus AWS-Konto. Für arn:aws:iam::In geben Sie die Konto-ID des Quell-Buckets ein.
9. Wählen Sie Änderungen speichern.

Um dem Besitzer des Quell-Buckets die Erlaubnis zu erteilen, den KMS Schlüssel (AWS CLI) zu verwenden

- Hinweise zum Befehl `put-key-policy` AWS Command Line Interface (AWS CLI) finden Sie unter put-key-policyin der AWS CLI Befehlsreferenz. Hinweise zur zugrunde liegenden

`PutKeyPolicy` API Operation finden Sie unter PutKeyPolicyin der AWS Key Management Service APIReferenz.

AWS KMS Überlegungen zu Transaktionsquoten

Wenn Sie nach der Aktivierung der regionsübergreifenden Replikation (CRR) viele neue Objekte mit AWS KMS Verschlüsselung hinzufügen, kann es zu Drosselungen (HTTP`503 Service Unavailable`Fehlern) kommen. Die Drosselung erfolgt, wenn die Anzahl an AWS KMS - Transaktionen pro Sekunde das aktuelle Kontingent überschreitet. Weitere Informationen finden Sie unter Kontingente im AWS Key Management Service -Entwicklerhandbuch.

Zum Anfordern einer Erhöhung für ein Kontingent verwenden Sie Service-Quotas verwenden. Weitere Informationen finden Sie unter Anfordern einer Kontingenterhöhung. Wenn Service Quotas in Ihrer Region nicht unterstützt werden, öffnen Sie einen AWS Support Fall.

Replikation für verschlüsselte Objekte aktivieren

Standardmäßig repliziert Amazon S3 keine Objekte, die mit serverseitiger Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln (SSE-) oder zweischichtiger serverseitiger Verschlüsselung mit AWS KMS Schlüsseln (-KMS) verschlüsselt wurden. DSSE KMS Um Objekte zu replizieren, die mit SSE - KMS oder DSS - verschlüsselt wurdenKMS, müssen Sie die Bucket-Replikationskonfiguration so ändern, dass Amazon S3 diese Objekte replizieren soll. In diesem Beispiel wird erklärt, wie Sie die Amazon S3 S3-Konsole und die AWS Command Line Interface (AWS CLI) verwenden, um die Bucket-Replikationskonfiguration zu ändern, um die Replikation verschlüsselter Objekte zu ermöglichen.

> **Note**
>
> Wenn ein S3-Bucket Key für den Quell- oder Ziel-Bucket aktiviert ist, ist der Verschlüsselungskontext der Amazon-Ressourcenname (ARN) des Buckets, nicht der des ObjektsARN. Sie müssen Ihre IAM Richtlinien aktualisieren, um den Bucket ARN für den Verschlüsselungskontext verwenden zu können. Weitere Informationen finden Sie unter S3 Bucket-Schlüssel und Replikation.

> **Note**
>
> Sie können Multiregion AWS KMS keys in Amazon S3 verwenden. Amazon S3 behandelt jedoch derzeit Multi-Regions-Schlüssel wie Einzel-Regions-Schlüssel und verwendet nicht

die Multi-Regions-Funktionen des Schlüssels. Weitere Informationen finden Sie unter Verwenden von Schlüsseln für mehrere Regionen im AWS Key Management Service Entwicklerhandbuch.

Verwenden der S3-Konsole

step-by-stepAnweisungen finden Sie unterKonfiguration der Replikation für Buckets im selben Konto. Dieses Thema enthält Anweisungen zum Einrichten einer Replikationskonfiguration, wenn Quell- und Ziel-Buckets denselben und unterschiedlichen AWS-Konten Besitzern gehören.

Verwenden von AWS CLI

Um verschlüsselte Objekte mit dem zu replizieren AWS CLI, gehen Sie wie folgt vor:

- Erstellen Sie Quell- und Ziel-Buckets und aktivieren Sie die Versionsverwaltung für diese Buckets.
- Erstellen Sie eine AWS Identity and Access Management (IAM) -Servicerolle, die Amazon S3 die Berechtigung erteilt, Objekte zu replizieren. Die Berechtigungen der IAM Rolle beinhalten die erforderlichen Berechtigungen, um die verschlüsselten Objekte zu replizieren.
- Fügen Sie eine Replikationskonfiguration zum Quell-Bucket hinzu. Die Replikationskonfiguration enthält Informationen zur Replikation von Objekten, die mithilfe KMS von Schlüsseln verschlüsselt wurden.
- Fügen Sie verschlüsselte Objekte zum Quell-Bucket hinzu.
- Testen Sie die Einrichtung, um sicherzustellen, dass Ihre verschlüsselten Objekte in den Ziel-Bucket repliziert werden.

Die folgenden Verfahren führen Sie durch diesen Prozess.

Replizieren Sie serverseitig verschlüsselte Objekte (AWS CLI) wie folgt:

Wenn Sie die Beispiele in diesem Verfahren verwenden möchten, ersetzen Sie sie durch Ihre eigenen Informationen. *`user input placeholders`*

1. In diesem Beispiel erstellen Sie sowohl die Quelle (*`amzn-s3-demo-source-bucket`*) und Ziel (*`amzn-s3-demo-destination-bucket`*) Eimer im selben AWS-Konto. Sie richten auch das Anmeldeinformationsprofil für die AWS CLI ein. Dieses Beispiel verwendet den Profilnamen *`acctA`*.

Informationen zum Einrichten von Anmeldeinformationsprofilen und zur Verwendung benannter Profile finden Sie im Benutzerhandbuch unter Konfiguration und Einstellungen für Anmeldeinformationsdateien.AWS Command Line Interface

2. Verwenden Sie die folgenden Befehle, um den *amzn-s3-demo-source-bucket*-Bucket zu erstellen und die Versionsverwaltung für diesen zu aktivieren. Mit den folgenden Beispielbefehlen wird der *amzn-s3-demo-source-bucket*-Bucket in der Region USA Ost (Nord-Virginia) (`us-east-1`) erstellt.

```
aws s3api create-bucket \
--bucket amzn-s3-demo-source-bucket \
--region us-east-1 \
--profile acctA
```

```
aws s3api put-bucket-versioning \
--bucket amzn-s3-demo-source-bucket \
--versioning-configuration Status=Enabled \
--profile acctA
```

3. Verwenden Sie die folgenden Befehle, um den *amzn-s3-demo-destination-bucket*-Bucket zu erstellen und die Versionsverwaltung für diesen zu aktivieren. Mit den folgenden Beispielbefehlen wird der *amzn-s3-demo-destination-bucket*-Bucket in der Region USA West (Oregon) (`us-west-2`) erstellt.

> **Note**
>
> Um eine Replikationskonfiguration einzurichten, wenn sich *amzn-s3-demo-source-bucket*- und *amzn-s3-demo-destination-bucket*-Bucket im selben AWS-Konto befinden, verwenden Sie dasselbe Profil. Dieses Beispiel verwendet *acctA*. Um die Replikation zu konfigurieren, wenn die Buckets unterschiedlichen Besitzern gehören AWS-Konten, geben Sie für jeden Bucket unterschiedliche Profile an.

```
aws s3api create-bucket \
--bucket amzn-s3-demo-destination-bucket \
--region us-west-2 \
--create-bucket-configuration LocationConstraint=us-west-2 \
```

```
--profile acctA
```

```
aws s3api put-bucket-versioning \
--bucket amzn-s3-demo-destination-bucket \
--versioning-configuration Status=Enabled \
--profile acctA
```

4. Als Nächstes erstellen Sie eine IAM Servicerolle. Sie geben diese Rolle in der Replikationskonfiguration an, die Sie dem *amzn-s3-demo-source-bucket* Bucket später hinzufügen. Amazon S3 übernimmt diese Rolle, um Objekte in Ihrem Namen zu replizieren. Sie erstellen eine IAM Rolle in zwei Schritten:

   - Erstellen Sie eine Servicerolle.
   - Fügen Sie eine Berechtigungsrichtlinie zur Rolle hinzu.

   a. Gehen Sie wie folgt vor, um eine IAM Servicerolle zu erstellen:

      i. Kopieren Sie die folgende Vertrauensrichtlinie und speichern Sie sie in einer Datei mit dem Namen *s3-role-trust-policy-kmsobj*`.json` im aktuellen Verzeichnis auf Ihrem lokalen Computer. Diese Richtlinie gewährt dem Amazon S3-Serviceprinzipal die Erlaubnis, die Rolle zu übernehmen, sodass Amazon S3 Aufgaben in Ihrem Namen ausführen kann.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

      ii. Verwenden Sie den folgenden Befehl, um die Rolle zu erstellen:

```
$ aws iam create-role \
--role-name replicationRolekmsobj \
```

```
--assume-role-policy-document file://s3-role-trust-policy-kmsobj.json  \
--profile acctA
```

b. Als Nächstes fügen Sie eine Berechtigungsrichtlinie zur Rolle hinzu. Diese Zugriffsrichtlinie erteilt Berechtigungen für verschiedene Amazon-S3-Bucket- und -Objektaktionen.

   i. Kopieren Sie die folgende Berechtigungsrichtlinie und speichern Sie sie in einer Datei mit dem Namen *s3-role-permissions-policykmsobj*`.json` im aktuellen Verzeichnis auf Ihrem lokalen Computer. Sie werden eine IAM Rolle erstellen und ihr die Richtlinie später hinzufügen.

> **Important**
>
> In der Berechtigungsrichtlinie geben Sie den AWS KMS Schlüssel anIDs, der für die Verschlüsselung der *amzn-s3-demo-destination-bucket* Buckets *amzn-s3-demo-source-bucket* und verwendet wird. Sie müssen zwei separate KMS Schlüssel für die *amzn-s3-demo-destination-bucket* Buckets *amzn-s3-demo-source-bucket* und erstellen. AWS KMS keys werden nicht außerhalb des Bereichs geteilt, AWS-Region in dem sie erstellt wurden.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action":[
            "s3:ListBucket",
            "s3:GetReplicationConfiguration",
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl",
            "s3:GetObjectVersionTagging"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket",
            "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
         ]
      },
      {
```

```
        "Action":[
           "s3:ReplicateObject",
           "s3:ReplicateDelete",
           "s3:ReplicateTags"
        ],
        "Effect":"Allow",
        "Condition":{
           "StringLikeIfExists":{
              "s3:x-amz-server-side-encryption":[
                 "aws:kms",
                 "AES256",
                 "aws:kms:dsse"
              ],
              "s3:x-amz-server-side-encryption-aws-kms-key-id":[
                 "AWS KMS key IDs(in ARN format) to use for encrypting
object replicas"
              ]
           }
        },
        "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
     },
     {
        "Action":[
           "kms:Decrypt"
        ],
        "Effect":"Allow",
        "Condition":{
           "StringLike":{
              "kms:ViaService":"s3.us-east-1.amazonaws.com",
              "kms:EncryptionContext:aws:s3:arn":[
                 "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
              ]
           }
        },
        "Resource":[
           "AWS KMS key IDs(in ARN format) used to encrypt source
objects."
        ]
     },
     {
        "Action":[
           "kms:Encrypt"
        ],
        "Effect":"Allow",
```

```
         "Condition":{
            "StringLike":{
               "kms:ViaService":"s3.us-west-2.amazonaws.com",
               "kms:EncryptionContext:aws:s3:arn":[
                  "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
               ]
            }
         },
         "Resource":[
            "AWS KMS key IDs (in ARN format) to use for encrypting object
 replicas"
         ]
      }
   ]
}
```

ii. Erstellen Sie eine Richtlinie und hängen Sie sie an die Rolle an.

```
$ aws iam put-role-policy \
--role-name replicationRolekmsobj \
--policy-document file://s3-role-permissions-policykmsobj.json \
--policy-name replicationRolechangeownerPolicy \
--profile acctA
```

5. Als Nächstes fügen Sie die folgende Replikationskonfiguration zum *amzn-s3-demo-source-bucket*-Bucket hinzu. Sie weist Amazon S3 an, Objekte mit dem Präfix *Tax/* in den *amzn-s3-demo-destination-bucket*-Bucket zu replizieren.

> **Important**
>
> In der Replikationskonfiguration geben Sie die IAM Rolle an, die Amazon S3 übernehmen kann. Dies können Sie nur tun, wenn Sie über die `iam:PassRole`-Berechtigung verfügen. Das Profil, das Sie im CLI Befehl angeben, muss über diese Berechtigung verfügen. Weitere Informationen finden Sie im Benutzerhandbuch unter Erteilen von Benutzerberechtigungen zur Übergabe einer Rolle AWS-Service an einen. IAM

```
<ReplicationConfiguration>
 <Role>IAM-Role-ARN</Role>
 <Rule>
```

```
    <Priority>1</Priority>
    <DeleteMarkerReplication>
       <Status>Disabled</Status>
    </DeleteMarkerReplication>
    <Filter>
       <Prefix>Tax</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <SourceSelectionCriteria>
      <SseKmsEncryptedObjects>
        <Status>Enabled</Status>
      </SseKmsEncryptedObjects>
    </SourceSelectionCriteria>
    <Destination>
      <Bucket>arn:aws:s3:::amzn-s3-demo-destination-bucket</Bucket>
      <EncryptionConfiguration>
        <ReplicaKmsKeyID>AWS KMS key IDs to use for encrypting object replicas</
ReplicaKmsKeyID>
      </EncryptionConfiguration>
    </Destination>
  </Rule>
</ReplicationConfiguration>
```

Um eine Replikationskonfiguration zum *amzn-s3-demo-source-bucket*-Bucket hinzuzufügen, gehen Sie wie folgt vor:

a. AWS CLI Für das müssen Sie die Replikationskonfiguration als angebenJSON. Speichern Sie Folgendes JSON in einer Datei (*replication*.json) im aktuellen Verzeichnis auf Ihrem lokalen Computer.

```
{
   "Role":"IAM-Role-ARN",
   "Rules":[
      {
         "Status":"Enabled",
         "Priority":1,
         "DeleteMarkerReplication":{
            "Status":"Disabled"
         },
         "Filter":{
            "Prefix":"Tax"
         },
```

```
         "Destination":{
            "Bucket":"arn:aws:s3:::amzn-s3-demo-destination-bucket",
            "EncryptionConfiguration":{
               "ReplicaKmsKeyID":"AWS KMS key IDs (in ARN format) to use for
 encrypting object replicas"
            }
         },
         "SourceSelectionCriteria":{
            "SseKmsEncryptedObjects":{
               "Status":"Enabled"
            }
         }
      }
   ]
}
```

b. Bearbeiten Sie denJSON, um Werte für den *amzn-s3-demo-destination-bucket* Bucket*AWS KMS key IDs (in ARN format)*, und anzugeben*IAM-role-ARN*. Speichern Sie die Änderungen.

c. Verwenden Sie den folgenden Befehl, um die Replikationskonfiguration zu Ihrem *amzn-s3-demo-source-bucket*-Bucket hinzuzufügen. Stellen Sie sicher, dass Sie den Namen für den *amzn-s3-demo-source-bucket*-Bucket angeben.

```
$ aws s3api put-bucket-replication \
--replication-configuration file://replication.json \
--bucket amzn-s3-demo-source-bucket \
--profile acctA
```

6. Testen Sie die Konfiguration, um zu überprüfen, ob die verschlüsselten Objekte repliziert werden. Führen Sie in der Amazon-S3-Konsole die folgenden Schritte aus:

a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

b. Erstellen Sie im *amzn-s3-demo-source-bucket*-Bucket einen Ordner mit dem Namen *Tax*.

c. Fügen Sie Beispielobjekte zum Ordner hinzu. Achten Sie darauf, die Verschlüsselungsoption auszuwählen und Ihren KMS Schlüssel zur Verschlüsselung der Objekte anzugeben.

d. Stellen Sie sicher, dass der *amzn-s3-demo-destination-bucket* Bucket die Objektreplikate enthält und dass sie mit dem KMS Schlüssel verschlüsselt sind, den Sie

in der Konfiguration angegeben haben. Weitere Informationen finden Sie unter the section called “Abrufen des Replikationsstatus”.

Verwenden von AWS SDKs

Ein Code-Beispiel, das zeigt, wie eine Replikationskonfiguration hinzugefügt wird, finden Sie unter Mit dem AWS SDKs. Sie müssen die Replikationskonfiguration entsprechend ändern.

Replizieren von Metadatenänderungen mit der Replikatmodifikationssynchronisierung

Amazon S3 Replica Modification Sync kann Ihnen dabei helfen, Objektmetadaten wie Tags, Zugriffskontrolllisten (ACLs) und Object Lock-Einstellungen zwischen Replikaten und Quellobjekten replizieren zu lassen. Standardmäßig repliziert Amazon S3 Metadaten aus den Quellobjekten nur auf die Replikate. Wenn die Synchronisierung von Replikatänderungen aktiviert ist, repliziert Amazon S3 Metadatenänderungen, die an den Replikatkopien vorgenommen wurden, zurück in das Quellobjekt, wodurch die Replikation bidirektional erfolgt (bidirektionale Replikation).

Aktivieren der Synchronisierung von Replikatänderungen

Sie können die Synchronisierung von Amazon-S3-Replikatänderungen mit neuen oder vorhandenen Replikationsregeln verwenden. Sie können es auf einen ganzen Bucket oder auf Objekte mit einem bestimmten Präfix anwenden.

Informationen zum Aktivieren der Synchronisation von Replikatänderungen mithilfe der Amazon S3 S3-Konsole finden Sie unterBeispiele für die Konfiguration der Live-Replikation. Dieses Thema enthält Anweisungen zur Aktivierung der Synchronisation von Replikatänderungen in Ihrer Replikationskonfiguration, wenn die Quell- und Ziel-Buckets demselben oder verschiedenen gehören. AWS-Konten

Um die Synchronisierung von Replikatänderungen mithilfe von AWS Command Line Interface (AWS CLI) zu aktivieren, müssen Sie dem Bucket, der die Replikate enthält, eine Replikationskonfiguration hinzufügen, wenn diese Option aktiviert ist. `ReplicaModifications` Um die bidirektionale Replikation einzurichten, erstellen Sie eine Replikationsregel vom Quell-Bucket (*`amzn-s3-demo-source-bucket`*) zum Bucket, der die Replikate () enthält. *`amzn-s3-demo-destination-bucket`* Erstellen Sie dann eine zweite Replikationsregel für den Bucket, der die Replikate (*`amzn-s3-demo-destination-bucket`*) enthält, zum Quell-Bucket (). *`amzn-s3-demo-source-bucket`* Die Quell- und Ziel-Buckets können sich im selben oder in unterschiedlichen Buckets befinden. AWS-Regionen

> **Note**
>
> Sie müssen die Replikaténderungssynchronisierung sowohl für den Quell- als auch für den Ziel-Bucket aktivieren, um Änderungen an den Replikatmetadaten wie Objektzugriffskontrolllisten (ACLs), Objekttags oder Objektsperreinstellungen für die replizierten Objekte zu replizieren. Wie alle Replikationsregeln können Sie diese Regeln auf den gesamten Bucket oder auf eine Teilmenge von Objekten anwenden, die nach Präfix- oder Objekt-Tags gefiltert sind.

In der folgenden Beispielkonfiguration repliziert Amazon S3 Metadatenänderungen unter dem Präfix *Tax* in den Bucket*amzn-s3-demo-source-bucket*, der die Quellobjekte enthält.

```
{
    "Rules": [
        {
            "Status": "Enabled",
            "Filter": {
                "Prefix": "Tax"
            },
            "SourceSelectionCriteria": {
                "ReplicaModifications":{
                    "Status": "Enabled"
                }
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::amzn-s3-demo-source-bucket"
            },
            "Priority": 1
        }
    ],
    "Role": "IAM-Role-ARN"
}
```

Vollständige Anweisungen zur Erstellung von Replikationsregeln mithilfe von finden Sie AWS CLI unterKonfiguration der Replikation für Buckets im selben Konto.

Replizieren von Löschmarkierungen auf Buckets

Wenn S3-Replikation aktiviert ist und ein Objekt im Quell-Bucket gelöscht wird, fügt Amazon S3 standardmäßig nur im Quell-Bucket eine Löschmarkierung hinzu. Diese Aktion trägt dazu bei,

Daten in den Ziel-Buckets vor versehentlichem oder böswilligem Löschen zu schützen. Wenn Sie die Replikation von Löschmarkern aktiviert haben, werden diese Markierungen in die Ziel-Buckets kopiert, und Amazon S3 verhält sich so, als ob das Objekt sowohl im Quell- als auch im Ziel-Bucket gelöscht worden wäre. Weitere Informationen zur Funktionsweise von Löschmarkierungen finden Sie unter Arbeiten mit Löschmarkierungen.

> **Note**
>
> - Die Replikation von Löschmarkern wird für tagbasierte Replikationsregeln nicht unterstützt. Die Replikation mit Löschmarkern entspricht auch nicht dem 15-minütigen Service Level Agreement (SLA), das gewährt wird, wenn Sie S3 Replication Time Control (S3RTC) verwenden.
> - Wenn Sie nicht die neueste XML Version der Replikationskonfiguration verwenden, wirken sich Löschvorgänge unterschiedlich auf die Replikation aus. Weitere Informationen finden Sie unter Auswirkungen von LöschVorgänge auf die Replikation.
> - Wenn Sie die Replikation mit Löschmarkierungen aktivieren und Ihr Quell-Bucket über eine S3-Lifecycle-Ablaufregel verfügt, werden die durch die S3-Lifecycle-Ablaufregel hinzugefügten Löschmarkierungen nicht in den Ziel-Bucket repliziert.

Aktivieren der Löschmarkierungs-Replikation

Sie können die Löschmarkierungs-Replikation mit einer neuen oder einer vorhandenen Replikationsregel verwenden. Sie können die Replikation von Löschmarkierungen auf einen ganzen Bucket oder auf Objekte mit einem bestimmten Präfix anwenden.

Informationen zum Aktivieren der Replikation von Löschmarkern mithilfe der Amazon S3 S3-Konsole finden Sie unterVerwenden der S3-Konsole. Dieses Thema enthält Anweisungen zur Aktivierung der Replikation von Löschmarken in Ihrer Replikationskonfiguration, wenn die Quell- und Ziel-Buckets demselben oder verschiedenen AWS-Konten gehören.

Um die Replikation mit Löschmarkern mithilfe von AWS Command Line Interface (AWS CLI) zu aktivieren, müssen Sie dem Quell-Bucket eine Replikationskonfiguration mit `DeleteMarkerReplication` aktivierter Option hinzufügen, wie in der folgenden Beispielkonfiguration gezeigt.

In der folgenden Beispielreplikationskonfiguration werden Löschmarkierungen *`amzn-s3-demo-destination-bucket`* für Objekte unter dem Präfix *`Tax`* in den Ziel-Bucket repliziert.

```
{
    "Rules": [
        {
            "Status": "Enabled",
            "Filter": {
                "Prefix": "Tax"
            },
            "DeleteMarkerReplication": {
                "Status": "Enabled"
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::amzn-s3-demo-destination-bucket"
            },
            "Priority": 1
        }
    ],
    "Role": "IAM-Role-ARN"
}
```

Vollständige Anweisungen zum Erstellen von Replikationsregeln mit dem finden Sie AWS CLI unterKonfiguration der Replikation für Buckets im selben Konto.

Live-Replikation verwalten oder pausieren

Bei der Live-Replikation handelt es sich um das automatische, asynchrone Kopieren von Objekten über Buckets hinweg, die sich im selben oder in unterschiedlichen Buckets befinden. AWS-Regionen Nachdem Sie Ihre Replikationskonfiguration eingerichtet haben, repliziert Amazon S3 neu erstellte Objekte und Objektaktualisierungen von einem Quell-Bucket in einen oder mehrere angegebene Ziel-Buckets.

Sie verwenden die Amazon-S3-Konsole, um dem Quell-Bucket Replikationsregeln hinzuzufügen. Replikationsregeln definieren, welche Quell-Bucket-Objekte repliziert werden sollen, und den Ziel-Bucket/die Ziel-Buckets, in dem/denen die replizierten Objekte gespeichert werden sollen. Weitere Informationen zur Replikation finden Sie unter Objekte innerhalb und zwischen Regionen replizieren.

Sie können Replikationsregeln auf der Seite Replikation in der Amazon S3 S3-Konsole verwalten. Sie können Replikationsregeln hinzufügen, anzeigen, bearbeiten, aktivieren, deaktivieren oder löschen. Sie können auch die Priorität Ihrer Replikationsregeln ändern. Weitere Informationen zum Hinzufügen von Replikationsregeln zu einem Bucket finden Sie unter Verwenden der S3-Konsole.

So verwalten Sie die Replikationsregeln für einen Bucket mithilfe der Amazon S3 S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie auf der Registerkarte Allgemeine Buckets den Namen des gewünschten Buckets aus.
4. Wählen Sie die Registerkarte Verwaltung und scrollen Sie dann nach unten zu Replikationsregeln.
5. Sie können Ihre Replikationsregeln auf folgende Weise ändern:
   - Um eine Replikationsregel zu aktivieren oder zu deaktivieren, wählen Sie das Optionsfeld links neben der Regel. Wählen Sie im Menü Aktionen die Option Regel aktivieren oder Regel deaktivieren aus. Sie können auch alle Regeln im Bucket im Menü Aktionen deaktivieren, aktivieren oder löschen.

     > **Note**
     >
     > Wenn Sie eine Replikationsregel deaktivieren und die Regel später wieder aktivieren, werden alle neuen oder geänderten Objekte, die während der Deaktivierung der Regel nicht repliziert wurden, nicht automatisch repliziert, wenn die Regel wieder aktiviert wird. Um diese Objekte zu replizieren, müssen Sie S3 Batch Replication verwenden. Weitere Informationen finden Sie unter the section called “Replizieren vorhandener Objekte”.

   - Um die Priorität einer Regel zu ändern, wählen Sie das Optionsfeld links neben der Regel und wählen Sie dann Regel bearbeiten aus.

     Regelprioritäten werden festgelegt, um Konflikte zu vermeiden, die durch Objekte verursacht werden, die mehreren Regeln unterliegen. Wenn sich Regeln überschneiden, verwendet Amazon S3 die Regelpriorität, um die Regel zu ermitteln, die angewendet werden muss. Je höher die Zahl, desto höher die Priorität. Weitere Informationen zur Priorität von Regeln finden Sie unter Elemente der Replikationskonfigurationsdatei.

Die Replikation wird angehalten oder beendet

Um die Replikation vorübergehend anzuhalten und sie später automatisch wieder aufzunehmen, können Sie die `aws:s3:bucket-pause-replication` Aktion unter verwenden. AWS Fault

Injection Service Weitere Informationen finden Sie unter aws:s3:bucket-pause-replicationund Unterbrechen Sie die S3-Replikation im AWS Fault Injection Service Benutzerhandbuch.

Um die Replikation in Amazon S3 zu beenden, empfehlen wir, Ihre Replikationsregeln zu deaktivieren. Wenn Sie eine Replikationsregel deaktivieren und die Regel später wieder aktivieren, werden alle neuen oder geänderten Objekte, die nicht repliziert wurden, während die Regel deaktiviert war, nicht automatisch repliziert, wenn die Regel erneut aktiviert wird. Um diese Objekte zu replizieren, müssen Sie S3 Batch Replication verwenden. Weitere Informationen finden Sie unter the section called “Replizieren vorhandener Objekte”.

Die Replikation wird auch beendet, wenn Sie die Rolle AWS Identity and Access Management (IAM), die AWS Key Management Service (AWS KMS) -Berechtigungen oder die Bucket-Richtlinienberechtigungen entfernen, die Amazon S3 die erforderlichen Berechtigungen gewähren. Wir empfehlen diese Ansätze jedoch nicht, da sie dazu führen, dass die Replikation fehlschlägt. Amazon S3 meldet den Replikationsstatus für betroffene Objekte als `FAILED`. Wenn Berechtigungen später wiederhergestellt werden, werden Objekte, die als markiert `FAILED` sind, nicht automatisch repliziert. Um diese Objekte zu replizieren, müssen Sie S3 Batch Replication verwenden.

Replizieren vorhandener Objekte mit Batch Replication

Die S3-Batch-Replikation unterscheidet sich von der Live-Replikation, bei der kontinuierlich und automatisch neue Objekte in Amazon S3 S3-Buckets repliziert werden. Stattdessen erfolgt die S3-Batchreplikation bei Bedarf für bestehende Objekte. Sie können S3 Batch Replication verwenden, um die folgenden Objekttypen zu replizieren:

- Objekte, die vor der Einrichtung einer Replikationskonfiguration existierten
- Objekte, die zuvor repliziert wurden
- Objekte, deren Replikation fehlgeschlagen ist

Sie können diese Objekte bei Bedarf replizieren, indem Sie einen Batch Operations-Job verwenden.

Um mit Batch Replication zu beginnen, können Sie:

- Batch-Replikation für eine neue Replikationsregel oder ein neues Ziel initiieren — Sie können einen einmaligen Batch-Replikationsauftrag erstellen, wenn Sie die erste Regel in einer neuen Replikationskonfiguration erstellen oder wenn Sie über die Amazon S3 S3-Konsole einen neuen Ziel-Bucket zu einer bestehenden Konfiguration hinzufügen.

- Batch-Replikation für eine bestehende Replikationskonfiguration initiieren — Sie können einen neuen Batch-Replikationsauftrag erstellen, indem Sie S3 Batch Operations über die Amazon S3-Konsole, die AWS Command Line Interface (AWS CLI) AWS SDKs, die oder Amazon S3 verwenden RESTAPI.

Wenn der Batch-Replikationsauftrag abgeschlossen ist, erhalten Sie einen Abschlussbericht. Weitere Informationen darüber, wie Sie diesen Bericht zur Untersuchung des Jobs verwenden können, finden Sie unterVerfolgen von Auftragsstatus- und Abschluss.

Überlegungen zur S3-Batch-Replikation

Bevor Sie S3 Batch Replication verwenden, sollten Sie sich die folgende Liste von Überlegungen ansehen:

- Ihr Quell-Bucket muss über eine vorhandene Replikationskonfiguration verfügen. Informationen zum Aktivieren der Replikation finden Sie unter Überblick über die Einrichtung der Live-Replikation und Beispiele für die Konfiguration der Live-Replikation.
- Wenn Sie S3 Lifecycle für Ihren Bucket konfiguriert haben, empfehlen wir, Ihre Lebenszyklusregeln zu deaktivieren, solange der Batch Replication-Job aktiv ist. Auf diese Weise wird die Parität zwischen den Quell- und Ziel-Buckets gewährleistet. Andernfalls könnten diese Buckets voneinander abweichen, und der Ziel-Bucket ist kein exaktes Replikat des Quell-Buckets. Betrachten wir folgendes Beispielszenario:
  - Ihr Quell-Bucket enthält mehrere Versionen eines Objekts und eine Löschmarkierung für dieses Objekt.
  - Ihre Quell- und Ziel-Buckets verfügen über eine Lebenszyklus-Konfiguration zum Entfernen abgelaufener Löschmarkierungen.

  In diesem Szenario repliziert die Batch-Replikation möglicherweise die Löschmarkierung in den Ziel-Bucket, bevor die Objektversionen repliziert werden. Dieses Verhalten könnte dazu führen, dass Ihre Lebenszykluskonfiguration die Löschmarkierung als abgelaufen markiert und die Löschmarkierung aus dem Ziel-Bucket entfernt, bevor die Objektversionen repliziert werden.
- Die Rolle AWS Identity and Access Management (IAM), die Sie für die Ausführung des Batch Operations-Jobs angeben, muss über die erforderlichen Berechtigungen verfügen, um den zugrunde liegenden Batch-Replikationsvorgang auszuführen. Weitere Informationen zum Erstellen von IAM Rollen finden Sie unterKonfiguration einer IAM Rolle für die S3-Batch-Replikation.
- Für die Batch-Replikation ist ein Manifest erforderlich, das von Amazon S3 generiert werden kann. Das generierte Manifest muss im selben Ordner AWS-Region wie der Quell-Bucket gespeichert

werden. Wenn Sie das Manifest nicht generieren möchten, können Sie einen Amazon S3 S3-Inventarbericht oder eine CSV Datei bereitstellen, die die Objekte enthält, die Sie replizieren möchten. Weitere Informationen finden Sie unter the section called “Angeben eines Manifests für einen Batch-Replikationsauftrag”.

- Die Batch-Replikation unterstützt nicht das erneute Replizieren von Objekten, die durch Angabe der Versions-ID des Objekts aus dem Ziel-Bucket gelöscht wurden. Wenn Sie diese Objekte erneut replizieren möchten, können Sie die Quellobjekte mit einem Batch–Kopierauftrag kopieren. Wenn Sie diese Objekte an Ort und Stelle kopieren, werden neue Versionen der Objekte im Quell-Bucket erstellt und die Replikation in den Ziel-Bucket automatisch initiiert. Durch Löschen und Neuerstellen des Ziel-Buckets wird die Replikation nicht initiiert.

  Weitere Informationen zu Batch Copy finden Sie unterBeispiele, die Batch-Vorgänge zum Kopieren von Objekten verwenden.
- Wenn Sie eine Replikationsregel für den Quell-Bucket verwenden, stellen Sie sicher, dass Sie Ihre Replikationskonfiguration aktualisieren, indem Sie der IAM Rolle, die der Replikationsregel zugeordnet ist, die entsprechenden Berechtigungen zum Replizieren von Objekten gewähren. Diese IAM Rolle muss über die erforderlichen Berechtigungen verfügen, um die Replikation sowohl für den Quell- als auch für den Ziel-Bucket durchzuführen.
- Wenn Sie innerhalb eines kurzen Zeitraums mehrere Batch-Replikationsaufträge für denselben Bucket einreichen, führt Amazon S3 diese Jobs gleichzeitig aus.
- Wenn Sie mehrere Batch-Replikationsaufträge für zwei verschiedene Buckets einreichen, beachten Sie, dass Amazon S3 möglicherweise nicht alle Jobs gleichzeitig ausführt. Wenn Sie die Anzahl der Batch-Replikationsaufträge überschreiten, die gleichzeitig auf Ihrem Konto ausgeführt werden können, pausiert Amazon S3 die Jobs mit niedrigerer Priorität, sodass sie an den Jobs mit höherer Priorität arbeiten. Nachdem die Jobs mit der höheren Priorität abgeschlossen sind, werden alle angehaltenen Jobs wieder aktiv.
- Die Batch-Replikation wird nicht für Objekte unterstützt, die in den Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive gespeichert sind.
- Um S3 Intelligent-Tiering-Objekte, die in den Speicherstufen Archive Access oder Deep Archive Access gespeichert sind, stapelweise zu replizieren, müssen Sie zunächst eine Wiederherstellungsanforderung initiieren und warten, bis die Objekte auf die Stufe für häufigen Zugriff verschoben wurden.

Angeben eines Manifests für einen Batch-Replikationsauftrag

Ein Manifest ist ein Amazon-S3-Objekt, das Objektschlüssel enthält, die Amazon S3 bearbeiten soll. Wenn Sie einen Batch-Replikationsauftrag erstellen möchten, müssen Sie entweder ein benutzergeneriertes Manifest angeben oder Amazon S3 ein Manifest auf der Grundlage Ihrer Replikationskonfiguration generieren lassen.

Wenn Sie ein benutzergeneriertes Manifest bereitstellen, muss es in Form eines Amazon S3 S3-Lagerberichts oder einer CSV Datei vorliegen. Wenn sich die Objekte in Ihrem Manifest in einem versionierten Bucket befinden, müssen Sie die Version IDs für die Objekte angeben. Nur das Objekt mit der im Manifest angegebenen Versions-ID wird repliziert. Weitere Informationen zum Angeben eines Manifests finden Sie unter Angeben eines Manifests.

Wenn Sie Amazon S3 in Ihrem Namen eine Manifestdatei generieren lassen möchten, verwenden die aufgelisteten Objekte denselben Quell-Bucket, dasselbe Präfix und dieselben Tags wie Ihre Replikationskonfigurationen im Quell-Bucket. Mit einem generierten Manifest repliziert Amazon S3 alle geeigneten Versionen Ihrer Objekte.

> **Note**
>
> Wenn Sie Amazon S3 das Manifest generieren lassen möchten, muss das Manifest im selben Bucket gespeichert werden AWS-Region wie der Quell-Bucket.

Filter für einen Batch-Replikationsauftrag

Bei der Erstellung Ihres Batch-Replikationsjobs können Sie optional zusätzliche Filter angeben, z. B. das Datum der Objekterstellung und den Replikationsstatus, um den Umfang des Jobs zu reduzieren.

Sie können Objekte filtern, die repliziert werden sollen, basierend auf dem `ObjectReplicationStatuses`-Wert, indem Sie einen oder mehrere der folgenden Werte angeben:

- `"NONE"` – Zeigt an, dass Amazon S3 noch nie versucht hat, das Objekt zu replizieren.
- `"FAILED"`— Zeigt an, dass Amazon S3 versucht hat, das Objekt zuvor zu replizieren, dies jedoch fehlgeschlagen ist.
- `"COMPLETED"` – Zeigt an, dass Amazon S3 das Objekt zuvor erfolgreich repliziert hat.
- `"REPLICA"`— Zeigt an, dass es sich bei diesem Objekt um ein Replikat handelt, das Amazon S3 aus einem anderen Quell-Bucket repliziert hat.

Weitere Informationen zum Replikationsstatus finden Sie unter Abrufen von Replikationsstatusinformationen.

Wenn Sie Ihren Batch-Replikationsauftrag nicht filtern, versucht Batch Operations, alle Objekte (unabhängig von ihren`ObjectReplicationStatus`) in Ihrem Manifest zu replizieren, die den Regeln in Ihrer Replikationskonfiguration entsprechen, mit Ausnahme bestimmter Objekte, die standardmäßig nicht repliziert werden. Weitere Informationen finden Sie unter the section called “Was wird mit Replikationskonfigurationen nicht repliziert?”

Abhängig von Ihrem Ziel können Sie einen oder mehrere der folgenden Werte festlegen`ObjectReplicationStatuses`:

- Um nur vorhandene Objekte zu replizieren, die noch nie repliziert wurden, schließen Sie nur ein. `"NONE"`
- Um erneut zu versuchen, nur Objekte zu replizieren, die zuvor nicht repliziert werden konnten, schließen Sie nur ein. `"FAILED"`
- Um sowohl vorhandene Objekte zu replizieren als auch erneut zu versuchen, Objekte zu replizieren, die zuvor nicht repliziert werden konnten, schließen Sie sowohl als auch ein. `"NONE"` `"FAILED"`
- Um einen Ziel-Bucket mit Objekten aufzufüllen, die an ein anderes Ziel repliziert wurden, schließen Sie Folgendes ein: `"COMPLETED"`
- Um Objekte zu replizieren, die zuvor repliziert wurden, schließen Sie Folgendes ein: `"REPLICA"`

Abschlussbericht zur Batchreplikation

Wenn Sie einen Batch-Replikationsauftrag erstellen, können Sie einen CSV Abschlussbericht anfordern. Dieser Bericht enthält die Objekte, Erfolgs- oder Fehlercodes für die Replikation, Ausgaben und Beschreibungen. Weitere Informationen zur Auftragsverfolgung und zu Abschlussberichten finden Sie unterAbschlussberichte.

Eine Liste der Fehlercodes und Beschreibungen der Replikationsfehler finden Sie unterGründe für das Fehlschlagen der Replikation in Amazon S3.

Informationen zur Problembehandlung bei der Batch-Replikation finden Sie unterFehler bei der Batchreplikation.

Erste Schritte mit der Batch-Replikation

Weitere Informationen zur Verwendung der Batch-Replikation finden Sie unter Tutorial: Replizieren vorhandener Objekte in Ihren-Amazon S3-Buckets mit S3 Batch Replication.

Konfiguration einer IAM Rolle für die S3-Batch-Replikation

Da es sich bei Amazon S3 Batch Replication um eine Art Batch Operations-Job handelt, müssen Sie eine AWS Identity and Access Management (IAM) -Rolle erstellen, um Batch Operations-Berechtigungen zur Ausführung von Aktionen in Ihrem Namen zu erteilen. Sie müssen der IAM Rolle Batch Operations auch eine IAM Batch-Replikationsrichtlinie hinzufügen.

Verwenden Sie die folgenden Verfahren, um eine Richtlinie und eine IAM Rolle zu erstellen, die Batch Operations die Berechtigung erteilen, einen Batch-Replikationsauftrag zu initiieren.

So erstellen Sie eine Richtlinie für die Batch-Replikation

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie unter Access management (Zugriffsverwaltung) Policies (Richtlinien) aus.
3. Wählen Sie Create Policy (Richtlinie erstellen) aus.
4. Wählen Sie auf der Seite „Berechtigungen angeben“ die Option JSON.
5. Fügen Sie eine der folgenden Richtlinien ein, je nachdem, ob Ihr Manifest von Amazon S3 generiert wurde oder ob Sie Ihr eigenes Manifest angeben. Weitere Informationen zu Manifesten finden Sie unter Angeben eines Manifests für einen Batch-Replikationsauftrag.

   Bevor Sie diese Richtlinien verwenden, ersetzen Sie die *`user input placeholders`* in den folgenden Richtlinien enthaltenen Richtlinien durch die Namen Ihres Quell-Buckets, Ihres Manifest-Buckets und Ihres Abschlussbericht-Buckets.

   > **Note**
   >
   > Ihre IAM Rolle für Batch Replication benötigt unterschiedliche Berechtigungen, je nachdem, ob Sie ein Manifest generieren oder eines bereitstellen. Stellen Sie daher sicher, dass Sie die entsprechende Richtlinie aus den folgenden Beispielen auswählen.

   Richtlinie für die Verwendung und Speicherung eines von Amazon S3 generierten Manifests

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action":[
            "s3:InitiateReplication"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
         ]
      },
      {
         "Action":[
            "s3:GetReplicationConfiguration",
            "s3:PutInventoryConfiguration"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
         ]
      },
      {
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*",
            "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
         ]
      }
   ]
```

```
}
```

Richtlinie bei Verwendung eines vom Benutzer bereitgestellten Manifests

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Action":[
            "s3:InitiateReplication"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-source-bucket/*"
         ]
      },
            {
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion"
         ],
         "Effect":"Allow",
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-manifest-bucket/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:PutObject"
         ],
         "Resource":[
            "arn:aws:s3:::amzn-s3-demo-completion-report-bucket/*"
         ]
      }
   ]
}
```

6. Wählen Sie Weiter.
7. Geben Sie einen Namen für die Richtlinie an und wählen Sie dann Richtlinie erstellen aus.

So erstellen Sie eine IAM Rolle für Batch Replication

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie unter Access management (Zugriffsverwaltung) Roles (Rollen) aus.
3. Wählen Sie Rolle erstellen.
4. Wählen Sie AWS-Serviceals Typ der vertrauenswürdigen Entität aus. Wählen Sie im Abschnitt Anwendungsfall S3 als Dienst und S3 Batch Operations als Anwendungsfall aus.
5. Wählen Sie Weiter. Die Seite „Berechtigungen hinzufügen“ wird angezeigt. Suchen Sie im Suchfeld nach der Richtlinie, die Sie im vorherigen Verfahren erstellt haben. Aktivieren Sie das Kontrollkästchen neben dem Richtliniennamen und wählen Sie dann Weiter aus.
6. Geben Sie auf der Seite Name, Überprüfung und Erstellung einen Namen für Ihre IAM Rolle ein.
7. Vergewissern Sie sich im Abschnitt Schritt 1: Identitäten vertrauen, dass Ihre IAM Rolle die folgende Vertrauensrichtlinie verwendet:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"batchoperations.s3.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

8. Stellen Sie im Abschnitt Schritt 2: Berechtigungen hinzufügen sicher, dass Ihre IAM Rolle die Richtlinie verwendet, die Sie zuvor erstellt haben.
9. Wählen Sie Rolle erstellen.

Erstellen Sie einen Batch-Replikationsauftrag für neue Replikationsregeln oder Ziele

In Amazon S3 repliziert die Live-Replikation keine Objekte, die bereits in Ihrem Quell-Bucket vorhanden waren, bevor Sie eine Replikationskonfiguration erstellt haben. Bei der Live-Replikation werden automatisch nur neue und aktualisierte Objekte repliziert, die nach der Erstellung der

Replikationskonfiguration in den Bucket geschrieben wurden. Um bereits vorhandene Objekte zu replizieren, können Sie S3 Batch Replication verwenden, um diese Objekte bei Bedarf zu replizieren.

Wenn Sie die erste Regel in einer neuen Live-Replikationskonfiguration erstellen oder über die Amazon S3 S3-Konsole einen neuen Ziel-Bucket zu einer bestehenden Replikationskonfiguration hinzufügen, können Sie optional einen Batch-Replikationsauftrag erstellen. Sie können diesen Batch-Replikationsjob verwenden, um vorhandene Objekte im Quell-Bucket in den Ziel-Bucket zu replizieren.

Informationen zur Verwendung der Batch-Replikation für eine bestehende Konfiguration, ohne einen neuen Ziel-Bucket hinzuzufügen, finden Sie unterErstellen eines Batch-Replikationsauftrags für vorhandene Replikationsregeln.

Voraussetzungen

Bevor Sie Ihren Batch Replication-Job erstellen, müssen Sie eine Batch Operations AWS Identity and Access Management (IAM) -Rolle erstellen, um Amazon S3 Berechtigungen zur Ausführung von Aktionen in Ihrem Namen zu erteilen. Weitere Informationen finden Sie unter Konfiguration einer IAM Rolle für die S3-Batch-Replikation.

Verwenden der Batch-Replikation für eine neue Replikationsregel oder ein neues Replikationsziel über die Amazon S3 S3-Konsole

Wenn Sie die erste Regel in einer neuen Replikationskonfiguration erstellen oder über die Amazon S3 S3-Konsole einen neuen Ziel-Bucket zu einer bestehenden Konfiguration hinzufügen, können Sie wählen, ob Sie einen Batch-Replikationsauftrag erstellen möchten, um bestehende Objekte im Quell-Bucket zu replizieren.

So erstellen Sie einen Batch-Replikationsauftrag beim Erstellen oder Aktualisieren einer Replikationskonfiguration

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Allgemeine Buckets den Namen des Buckets aus, der die Objekte enthält, die Sie replizieren möchten.
4. Um eine neue Replikationsregel zu erstellen oder eine bestehende Regel zu bearbeiten, wählen Sie die Registerkarte Verwaltung und scrollen Sie nach unten zu Replikationsregeln:

- Wählen Sie zum Erstellen einer neuen Replikationsregel Create replication rule (Erstellen einer Replikationsregel). Beispiele für die Einrichtung einer grundlegenden Replikationsregel finden Sie unterBeispiele für die Konfiguration der Live-Replikation.
- Um eine bestehende Replikationsregel zu bearbeiten, klicken Sie auf das Optionsfeld neben dem Regelnamen und wählen Sie dann Regel bearbeiten aus.

5. Erstellen Sie Ihre neue Replikationsregel oder bearbeiten Sie das Ziel für Ihre vorhandene Replikationsregel und wählen Sie Save (Speichern).

   Nachdem Sie die erste Regel in einer neuen Replikationskonfiguration erstellt oder eine vorhandene Konfiguration bearbeitet haben, um ein neues Ziel hinzuzufügen, wird der Dialog Replicate existing objects? (Vorhandene Objekte replizieren?) angezeigt, über den Sie einen Batch-Replikationsauftrag erstellen können.

6. Wenn Sie diesen Job jetzt erstellen und ausführen möchten, wählen Sie Ja, vorhandene Objekte replizieren.

   Wenn Sie zu einem späteren Zeitpunkt einen Batch-Replikationsauftrag erstellen möchten, wählen Sie Nein, vorhandene Objekte nicht replizieren.

7. Wenn Sie Ja, vorhandene Objekte replizieren wählen, wird die Jobseite Batch Operations erstellen angezeigt. Der S3-Batch-Replikationsauftrag hat die folgenden Einstellungen:

   Optionen für die Ausführung von Job

   Wenn Sie möchten, dass der S3-Batch-Replikationsauftrag sofort ausgeführt wird, wählen Sie Den Job automatisch ausführen, wenn er bereit ist. Wenn Sie den Job zu einem späteren Zeitpunkt ausführen möchten, wählen Sie Warten, bis der Job fertig ist.

   > **Note**
   >
   > Wenn Sie den Job automatisch ausführen, wenn er fertig ist wählen, können Sie kein Batch Operations-Manifest erstellen und speichern. Um das Batch Operations-Manifest zu speichern, wählen Sie Warten, um den Job auszuführen, wenn er fertig ist.

Batchvorgangs-Manifest

Wenn Sie „Warten, bis der Job ausgeführt wird, wenn er bereit ist“ ausgewählt haben, wird der Abschnitt „Batch Operations Manifest“ angezeigt. Das Manifest ist eine Liste aller Objekte, für die die festgelegte Aktion ausgeführt werden soll. Sie können wählen, ob Sie das Manifest speichern möchten. Ähnlich wie bei S3-Inventardateien wird das Manifest als CSV Datei gespeichert und in einem Bucket gespeichert. Weitere Informationen über die Batchvorgangs-Manifeste finden Sie unter Angeben eines Manifests.

Abschlussbericht

S3 Batch Operations führt für jedes im Manifest angegebene Objekt eine Aufgabe aus. Abschlussberichte bieten eine einfache Möglichkeit, die Ergebnisse von Aufgaben in einem konsolidierten Format ohne zusätzliche Einrichtung anzuzeigen. Sie können einen Abschlussbericht für alle oder nur für die fehlgeschlagenen Aufgaben anfordern. Weitere Informationen zu Abschlussberichten finden Sie unter Abschlussberichte.

Berechtigungen

Eine der häufigsten Ursachen für Replikationsfehler sind unzureichende Berechtigungen in der Rolle provided AWS Identity and Access Management (IAM). Weitere Informationen zum Erstellen dieser Rolle finden Sie unter Konfiguration einer IAM Rolle für die S3-Batch-Replikation. Stellen Sie sicher, dass Sie eine IAM Rolle erstellen oder auswählen, die über die erforderlichen Berechtigungen für die Batch-Replikation verfügt.

8. Wählen Sie Save (Speichern) aus.

Erstellen eines Batch-Replikationsauftrags für vorhandene Replikationsregeln

In Amazon S3 repliziert die Live-Replikation keine Objekte, die bereits in Ihrem Quell-Bucket vorhanden waren, bevor Sie eine Replikationskonfiguration erstellt haben. Bei der Live-Replikation werden automatisch nur neue und aktualisierte Objekte repliziert, die nach der Erstellung der Replikationskonfiguration in den Bucket geschrieben wurden. Um bereits vorhandene Objekte zu replizieren, können Sie S3 Batch Replication verwenden, um diese Objekte bei Bedarf zu replizieren.

Sie können S3 Batch Replication für eine bestehende Replikationskonfiguration konfigurieren, indem Sie die AWS SDKs, AWS Command Line Interface (AWS CLI) oder die Amazon S3 S3-Konsole verwenden. Einen Überblick über die Batch-Replikation finden Sie unterReplizieren vorhandener Objekte mit Batch Replication.

Wenn der Batch-Replikationsauftrag abgeschlossen ist, erhalten Sie einen Abschlussbericht. Weitere Informationen dazu, wie Sie den Bericht verwenden können, um den Auftrag zu untersuchen, finden Sie unter Verfolgen von Auftragsstatus- und Abschluss.

Voraussetzungen

Bevor Sie Ihren Batch Replication-Job erstellen, müssen Sie eine Batch Operations AWS Identity and Access Management (IAM) -Rolle erstellen, um Amazon S3 Berechtigungen zur Ausführung von Aktionen in Ihrem Namen zu erteilen. Weitere Informationen finden Sie unter Konfiguration einer IAM Rolle für die S3-Batch-Replikation.

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Klicken Sie im linken Navigationsbereich auf Batchvorgänge.
3. Wählen Sie Job erstellen aus.
4. Vergewissern Sie sich, dass im AWS-RegionAbschnitt die Region angezeigt wird, in der Sie Ihren Job erstellen möchten.
5. Geben Sie im Abschnitt Manifest das Manifestformat an, das Sie verwenden möchten. Das Manifest ist eine Liste aller Objekte, für die die festgelegte Aktion ausgeführt werden soll. Weitere Informationen über die Batchvorgangs-Manifeste finden Sie unter Angeben eines Manifests.

   - Wenn Sie ein Manifest vorbereitet haben, wählen Sie S3-Inventarbericht (manifest.json) oder. CSV Wenn sich Ihr Manifest in einem versionierten Bucket befindet, können Sie die Versions-ID für das Manifest angeben. Wenn Sie keine Versions-ID angeben, verwendet Batch Operations die aktuelle Version Ihres Manifests. Weitere Informationen zum Erstellen eines Manifests finden Sie unterAngeben eines Manifests.

     > **Note**
     >
     > Wenn sich die Objekte in Ihrem Manifest in einem versionierten Bucket befinden, müssen Sie die Version IDs für die Objekte angeben. Weitere Informationen finden Sie unter Angeben eines Manifests.

   - Um ein Manifest basierend auf Ihrer Replikationskonfiguration zu erstellen, wählen Sie Create manifest using S3 Replication configuration (Manifest mit der S3-Replikationskonfiguration erstellen). Wählen Sie dann den Quell-Bucket Ihrer Replikationskonfiguration aus.

6. (Optional) Wenn Sie „Manifest mithilfe der S3-Replikationskonfiguration erstellen“ ausgewählt haben, können Sie zusätzliche Filter hinzufügen, z. B. das Erstellungsdatum des Objekts und den Replikationsstatus. Beispiele dafür, wie Sie nach dem Replikationsstatus filtern können, finden Sie unterAngeben eines Manifests für einen Batch-Replikationsauftrag.
7. (Optional) Wenn Sie „Manifest mithilfe der S3-Replikationskonfiguration erstellen“ ausgewählt haben, können Sie das generierte Manifest speichern. Um dieses Manifest zu speichern, wählen Sie Batch Operations-Manifest speichern. Geben Sie dann den Ziel-Bucket für das Manifest an und wählen Sie aus, ob das Manifest verschlüsselt werden soll.

   > **Note**
   >
   > Das generierte Manifest muss im selben Ordner AWS-Region wie der Quell-Bucket gespeichert werden.

8. Wählen Sie Weiter.
9. Wählen Sie auf der Seite „Vorgänge“ die Option „Replizieren“ und anschließend „Weiter“.
10. (Optional) Geben Sie eine Beschreibung ein.
11. Passen Sie die Priority (Priorität) des Auftrags bei Bedarf an. Höhere Nummern bedeuten eine höhere Priorität. Amazon S3 versucht, Aufträge mit höherer Priorität vor Aufträgen mit niedrigerer Priorität auszuführen. Weitere Informationen zur Auftragspriorität finden Sie unter Zuweisen der Auftragspriorität.
12. (Optional) Generieren Sie einen Abschlussbericht. Um diesen Bericht zu generieren, wählen Sie Abschlussbericht erstellen aus.

    Wenn Sie einen Abschlussbericht erstellen möchten, müssen Sie entweder die Meldung von Failed tasks only (Nur fehlgeschlagenen Aufgaben) oder All tasks (Allen Aufgaben) auswählen und einen Ziel-Bucket für den Bericht angeben.
13. Stellen Sie sicher, dass Sie im Abschnitt Berechtigungen eine IAM Rolle auswählen, die über die erforderlichen Berechtigungen für die Batch-Replikation verfügt. Eine der häufigsten Ursachen für Replikationsfehler sind unzureichende Berechtigungen in der angegebenen IAM Rolle. Weitere Informationen zum Erstellen dieser Rolle finden Sie unter Konfiguration einer IAM Rolle für die S3-Batch-Replikation.
14. (Optional) Fügen Sie Auftrags-Tags zum Batch-Replikationsauftrag hinzu.
15. Wählen Sie Weiter.
16. Überprüfen Sie Ihre Job-Konfiguration und wählen Sie dann Create Job aus.

Verwenden von AWS CLI mit einem S3-Manifest

Der folgende `create-job` Beispielbefehl erstellt einen S3-Batch-Replikationsauftrag unter Verwendung eines von S3 generierten Manifests für AWS-Konto *111122223333*. In diesem Beispiel werden vorhandene Objekte und Objekte repliziert, die zuvor nicht repliziert werden konnten. Hinweise zum Filtern nach Replikationsstatus finden Sie unter. Angeben eines Manifests für einen Batch-Replikationsauftrag

Um diesen Befehl zu verwenden, ersetzen Sie den *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen. Ersetzen Sie die IAM Rolle `role/`*batch-Replication-IAM-policy* durch die IAM Rolle, die Sie zuvor erstellt haben. Weitere Informationen finden Sie unter Konfiguration einer IAM Rolle für die S3-Batch-Replikation.

```
aws s3control create-job --account-id 111122223333 \
--operation '{"S3ReplicateObject":{}}' \
--report '{"Bucket":"arn:aws:s3:::amzn-s3-demo-completion-report-bucket",\
"Prefix":"batch-replication-report", \
"Format":"Report_CSV_20180820","Enabled":true,"ReportScope":"AllTasks"}' \
--manifest-generator '{"S3JobManifestGenerator": {"ExpectedBucketOwner":
 "111122223333", \
"SourceBucket": "arn:aws:s3:::amzn-s3-demo-source-bucket", \
"EnableManifestOutput": false, "Filter": {"EligibleForReplication": true, \
"ObjectReplicationStatuses": ["NONE","FAILED"]}}}' \
--priority 1 \
--role-arn arn:aws:iam::111122223333:role/batch-Replication-IAM-policy \
--no-confirmation-required \
--region source-bucket-region
```

> **Note**
>
> Sie müssen den Job aus demselben Bucket initiieren AWS-Region wie der Replizierungsquellen-Bucket.

Nachdem Sie einen Batch-Replikationsauftrag erfolgreich initiiert haben, erhalten Sie die Auftrags-ID als Antwort. Sie können diesen Job mit dem folgenden `describe-job` Befehl überwachen. Um diesen Befehl zu verwenden, ersetzen Sie den *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
aws s3control describe-job --account-id 111122223333 --job-id job-id --region source-bucket-region
```

Verwenden von AWS CLI mit einem vom Benutzer bereitgestellten Manifest

Im folgenden Beispiel wird ein S3-Batch-Replikationsauftrag mithilfe eines benutzerdefinierten Manifests für AWS-Konto *111122223333* erstellt. Wenn sich die Objekte in Ihrem Manifest in einem versionierten Bucket befinden, müssen Sie die Version IDs für die Objekte angeben. Es wird nur das Objekt mit der im Manifest angegebenen Versions-ID repliziert. Weitere Informationen zum Erstellen eines Manifests finden Sie unterAngeben eines Manifests.

Um diesen Befehl zu verwenden, ersetzen Sie den *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen. Ersetzen Sie die IAM Rolle `role/`*batch-Replication-IAM-policy* durch die IAM Rolle, die Sie zuvor erstellt haben. Weitere Informationen finden Sie unter Konfiguration einer IAM Rolle für die S3-Batch-Replikation.

```
aws s3control create-job --account-id 111122223333 \
--operation '{"S3ReplicateObject":{}}' \
--report '{"Bucket":"arn:aws:s3:::amzn-s3-demo-completion-report-bucket",\
"Prefix":"batch-replication-report", \
"Format":"Report_CSV_20180820","Enabled":true,"ReportScope":"AllTasks"}' \
--manifest '{"Spec":{"Format":"S3BatchOperations_CSV_20180820",\
"Fields":["Bucket","Key","VersionId"]},\
"Location":{"ObjectArn":"arn:aws:s3:::amzn-s3-demo-manifest-bucket/manifest.csv",\
"ETag":"Manifest Etag"}}' \
--priority 1 \
--role-arn arn:aws:iam::111122223333:role/batch-Replication-IAM-policy \
--no-confirmation-required \
--region source-bucket-region
```

> **Note**
>
> Sie müssen den Job aus demselben Bucket initiieren AWS-Region wie der Replizierungsquellen-Bucket.

Nachdem Sie einen Batch-Replikationsauftrag erfolgreich initiiert haben, erhalten Sie die Auftrags-ID als Antwort. Sie können diesen Job mit dem folgenden `describe-job` Befehl überwachen.

```
aws s3control describe-job --account-id 111122223333 --job-id job-id --region source-bucket-region
```

Fehlerbehebung bei einer Replikation

In diesem Abschnitt finden Sie Tipps zur Problembehandlung in Amazon S3 Replication und Informationen zu Fehlern in S3 Batch Replication.

Themen

- Tipps zur Fehlerbehebung in S3 Replication
- Fehler bei der Batchreplikation

Tipps zur Fehlerbehebung in S3 Replication

Wenn Objektreplikate nicht im Ziel-Bucket erscheinen, nachdem Sie die Replikation konfiguriert haben, können Sie mit diesen Tipps zur Fehlerbehebung Probleme identifizieren und beheben.

- Die meisten Objekte werden innerhalb von 15 Minuten repliziert. Die von Amazon S3 für die Replikation eines Objekts benötigte Zeit hängt von verschiedenen Faktoren ab, einschließlich des Quell- und Ziel-Regionspaars sowie der Größe des Objekts. Bei großen Objekten kann die Replikation mehrere Stunden dauern. Um Einblick in die Replikationszeiten zu erhalten, können Sie S3 Replication Time Control (S3RTC) verwenden.

  Wenn das replizierte Objekt groß ist, warten Sie eine Weile, bevor Sie überprüfen, ob es im Ziel erscheint. Sie können auch den Replikationsstatus des Quellobjekts überprüfen. Wenn der Status der Objektreplikation lautet`PENDING`, hat Amazon S3 die Replikation nicht abgeschlossen. Wenn der Replikationsstatus des Objekts `FAILED` lautet, überprüfen Sie die Replikationskonfiguration des Quell-Buckets.

  Wenn Sie Informationen zu Fehlern während der Replikation erhalten möchten, können Sie einrichten, dass Sie über Fehlerereignisse bei der Replikation mit Amazon-S3-Ereignisbenachrichtigungen benachrichtigt werden. Weitere Informationen finden Sie unter Erhalten von Amazon-S3-Ereignisbenachrichtigungen über Replikations-Fehlerereignisse.
- Um den Replikationsstatus eines Objekts zu überprüfen, können Sie den `HeadObject` API Vorgang aufrufen. Die `HeadObject` API Operation gibt den `PENDING`, `COMPLETED`, oder `FAILED` Replikationsstatus eines Objekts zurück. In einer Antwort auf einen `HeadObject` API Aufruf wird der Replikationsstatus im `x-amz-replication-status` Header zurückgegeben.

> **Note**
>
> Zum Ausführen von `HeadObject` müssen Sie über Lesezugriff auf das angeforderte Objekt verfügen. Eine `HEAD`-Anforderung bietet die gleichen Optionen wie eine `GET`-Anforderung, ohne eine `GET`-Operation auszuführen. Um beispielsweise eine `HeadObject` Anforderung mithilfe von AWS Command Line Interface (AWS CLI) auszuführen, können Sie den folgenden Befehl ausführen. Ersetzen Sie *`user input placeholders`* durch Ihre Informationen.
>
> ```
> aws s3api head-object --bucket amzn-s3-demo-source-bucket --key index.html
> ```

- Wenn Objekte mit einem `FAILED` Replikationsstatus `HeadObject` zurückgegeben werden, können Sie S3 Batch Replication verwenden, um diese fehlgeschlagenen Objekte zu replizieren. Weitere Informationen finden Sie unter the section called “Replizieren vorhandener Objekte”. Alternativ können Sie die fehlgeschlagenen Objekte erneut in den Quell-Bucket hochladen, wodurch die Replikation für die neuen Objekte initiiert wird.
- Überprüfen Sie in der Replikations-Konfiguration des Quell-Buckets Folgendes:
  - Der Amazon-Ressourcenname (ARN) des Ziel-Buckets ist korrekt.
  - ob das Schlüsselnamenpräfix korrekt ist. Wenn Sie beispielsweise die Konfiguration so einrichten, dass nur Objekte mit dem Präfix `Tax` repliziert werden, werden nur Objekte mit Schlüsselnamen wie beispielsweise `Tax/document1` oder `Tax/document2` repliziert. Ein Objekt mit dem Schlüsselnamen `document3` wird nicht repliziert.
  - Der Status der Replikationsregel lautet `Enabled`.
- Stellen Sie sicher, dass die Versionierung für keinen Bucket in der Replikationskonfiguration ausgesetzt wurde. Sowohl für die Quell- als auch für die Ziel-Buckets muss die Versionsverwaltung aktiviert sein.
- Wenn für eine Replikationsregel die Option Objekteigentümer zum Besitzer des Ziel-Buckets ändern festgelegt ist, muss die Rolle AWS Identity and Access Management (IAM), die für die Replikation verwendet wird, über die `s3:ObjectOwnerOverrideToBucketOwner` entsprechende Berechtigung verfügen. Diese Berechtigung wird für die Ressource (in diesem Fall den Ziel-Bucket) erteilt. Die folgende `Resource`-Anweisung zeigt beispielsweise, wie diese Berechtigung für den Ziel-Bucket erteilt wird:

```
{
  "Effect":"Allow",
```

```
  "Action":[
    "s3:ObjectOwnerOverrideToBucketOwner"
  ],
  "Resource":"arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
}
```

- Wenn sich der Ziel-Bucket im Besitz eines anderen Kontos befindet, muss der Eigentümer des Ziel-Buckets die Berechtigung `s3:ObjectOwnerOverrideToBucketOwner` über die Ziel-Bucket-Richtlinie auch dem Eigentümer des Quell-Buckets erteilen. Wenn Sie die folgende Bucket-Beispielrichtlinie verwenden möchten, ersetzen Sie die *`user input placeholders`* durch Ihre eigenen Informationen:

```
{
  "Version": "2012-10-17",
  "Id": "Policy1644945280205",
  "Statement": [
    {
      "Sid": "Stmt1644945277847",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789101:role/s3-replication-role"
      },
      "Action": [
        "s3:ReplicateObject",
        "s3:ReplicateTags",
        "s3:ObjectOwnerOverrideToBucketOwner"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/*"
    }
  ]
}
```

> **Note**
>
> Wenn die Einstellungen des Ziel-Buckets zur Objekteigentümerschaft Bucket-Eigentümer erzwungen beinhalten, müssen Sie die Einstellung in der Replikationsregel nicht in Objekteigentümerschaft in Eigentümer des Ziel-Buckets ändern ändern. Die Änderung der Objekteigentümerschaft erfolgt standardmäßig. Weitere Informationen zum Ändern der Replikateigentümerschaft finden Sie unter Ändern des Replikat-Eigentümers.

- Wenn Sie die Replikationskonfiguration in einem kontoübergreifenden Szenario festlegen, in dem die Quell- und Ziel-Buckets unterschiedlichen Besitzern gehören AWS-Konten, können die Ziel-Buckets nicht als Bucket konfiguriert werden, der vom Antragsteller bezahlt wird. Weitere Informationen finden Sie unter Nutzen von Buckets mit Zahlung durch den Anforderer für Speicherübertragungen und Nutzung.
- Wenn die Quellobjekte eines Buckets mithilfe einer serverseitigen Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-KMS) verschlüsselt werden, muss die Replikationsregel so konfiguriert werden, dass sie -verschlüsselte Objekte einschließt. AWS KMS Wählen Sie Mit AWS KMS verschlüsselte Objekte replizieren unter den Einstellungen zu Verschlüsselung in der Amazon-S3-Konsole aus. Wählen Sie dann einen AWS KMS Schlüssel für die Verschlüsselung der Zielobjekte aus.

> **Note**
>
> Wenn sich der Ziel-Bucket in einem anderen Konto befindet, geben Sie einen vom AWS KMS Kunden verwalteten Schlüssel an, der dem Zielkonto gehört. Verwenden Sie nicht den von Amazon S3 verwalteten Standardschlüssel (`aws/s3`). Die Verwendung des Standardschlüssels verschlüsselt die Objekte mit dem verwalteten Amazon S3 S3-Schlüssel, der dem Quellkonto gehört, wodurch verhindert wird, dass das Objekt mit einem anderen Konto geteilt wird. Infolgedessen kann das Zielkonto nicht auf die Objekte im Ziel-Bucket zugreifen.

Um einen AWS KMS Schlüssel zu verwenden, der zum Zielkonto gehört, um die Zielobjekte zu verschlüsseln, muss das Zielkonto der Replikationsrolle in der KMS Schlüsselrichtlinie die `kms:Encrypt` Berechtigungen `kms:GenerateDataKey` und gewähren. Um die folgende Beispielanweisung in Ihrer KMS Schlüsselrichtlinie zu verwenden, ersetzen Sie sie durch Ihre eigenen Informationen: *`user input placeholders`*

```
{
    "Sid": "AllowS3ReplicationSourceRoleToUseTheKey",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::123456789101:role/s3-replication-role"
    },
    "Action": ["kms:GenerateDataKey", "kms:Encrypt"],
    "Resource": "*"
}
```

Wenn Sie für die `Resource` Anweisung in der AWS KMS Schlüsselrichtlinie ein Sternchen (`*`) verwenden, erteilt die Richtlinie nur der Replikationsrolle die Erlaubnis, den KMS Schlüssel zu verwenden. Die Richtlinie erlaubt der Replikationsrolle nicht, ihre Berechtigungen zu erweitern.

Standardmäßig gewährt die KMS Schlüsselrichtlinie dem Root-Benutzer volle Rechte für den Schlüssel. Diese Berechtigungen können an andere Benutzer in demselben Konto delegiert werden. Sofern die KMS Quellschlüsselrichtlinie keine `Deny` Anweisungen enthält, ist es ausreichend, der Replikationsrolle mithilfe einer IAM Richtlinie Berechtigungen für den KMS Quellschlüssel zu erteilen.

> **Note**
>
> KMSwichtige Richtlinien, die den Zugriff auf bestimmte CIDR Bereiche, Virtual Private Cloud (VPC) -Endpunkte oder S3-Zugriffspunkte einschränken, können dazu führen, dass die Replikation fehlschlägt.

Wenn entweder der Quell- oder der KMS Zielschlüssel Berechtigungen basierend auf dem Verschlüsselungskontext gewähren, stellen Sie sicher, dass Amazon S3 S3-Bucket Keys für die Buckets aktiviert sind. Wenn die S3-Bucket-Schlüssel für die Buckets aktiviert sind, muss der Verschlüsselungskontext die Ressource auf Bucket-Ebene sein, wie im Folgenden veranschaulicht:

```
"kms:EncryptionContext:arn:aws:arn": [
     "arn:aws:s3:::amzn-s3-demo-source-bucket"
     ]
"kms:EncryptionContext:arn:aws:arn": [
     "arn:aws:s3:::amzn-s3-demo-destination-bucket"
     ]
```

Zusätzlich zu den durch die KMS Schlüsselrichtlinie gewährten Berechtigungen muss das Quellkonto der IAM Richtlinie der Replikationsrolle die folgenden Mindestberechtigungen hinzufügen:

```
{
    "Effect": "Allow",
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
```

```
    ],
    "Resource": [
        "Source-KMS-Key-ARN"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "kms:GenerateDataKey",
        "kms:Encrypt"
    ],
    "Resource": [
        "Destination-KMS-Key-ARN"
    ]
}
```

Weitere Informationen zum Replizieren von Objekten, die mit verschlüsselt sind AWS KMS, finden Sie unter Verschlüsselte Objekte replizieren.

- Wenn der Ziel-Bucket einem anderen gehört AWS-Konto, stellen Sie sicher, dass der Bucket-Besitzer eine Bucket-Richtlinie für den Ziel-Bucket hat, die es dem Besitzer des Quell-Buckets ermöglicht, Objekte zu replizieren. Ein Beispiel finden Sie unter Konfiguration der Replikation für Buckets in verschiedenen Konten.
- Wenn Ihre Objekte auch nach der Bestätigung der Berechtigungen nicht repliziert werden, suchen Sie an den folgenden Stellen nach expliziten `Deny`-Anweisungen:
  - `Deny`-Anweisungen in den Richtlinien des Quell- oder Ziel-Buckets. Die Replikation schlägt fehl, wenn die Bucket-Richtlinie den Zugriff auf die Replikationsrolle für eine der folgenden Aktionen verweigert:

    Quell-Bucket

    ```
            "s3:GetReplicationConfiguration",
            "s3:ListBucket",
            "s3:GetObjectVersionForReplication",
            "s3:GetObjectVersionAcl",
            "s3:GetObjectVersionTagging"
    ```

    Ziel-Buckets:

    ```
            "s3:ReplicateObject",
            "s3:ReplicateDelete",
    ```

```
            "s3:ReplicateTags"
```

  - `Deny`Mit der IAM Rolle verknüpfte Anweisungen oder Berechtigungsgrenzen können dazu führen, dass die Replikation fehlschlägt.
  - `Deny`Anweisungen in AWS Organizations Dienststeuerungsrichtlinien (SCPs), die entweder dem Quell- oder dem Zielkonto zugeordnet sind, können dazu führen, dass die Replikation fehlschlägt.
  - `Deny`Anweisungen in AWS Organizations Ressourcensteuerungsrichtlinien (RCPs), die entweder dem Quell- oder dem Ziel-Bucket zugeordnet sind, können dazu führen, dass die Replikation fehlschlägt.
- Wenn ein Objektreplikat nicht im Ziel-Bucket angezeigt wird, könnten folgende Probleme die Replikation verhindert haben:
  - Amazon S3 repliziert keine Objekte in einem Quell-Bucket, der ein Replikat ist, das mit einer anderen Replikations-Konfiguration erstellt wurde. Wenn Sie beispielsweise die Replikationskonfiguration von Bucket A zu Bucket B zu Bucket C festlegen, repliziert Amazon S3 keine Objektreplikate von Bucket B zu Bucket C.
  - Ein Besitzer des Quell-Buckets kann anderen AWS-Konten Berechtigungen zum Hochladen von Objekten gewähren. Standardmäßig besitzt der Quell-Bucket-Eigentümer keine Berechtigungen für die Objekte, die von anderen Konten erstellt wurden. Die Replikations-Konfiguration repliziert nur die Objekte, für die der Quell-Bucket-Eigentümer über Zugriffsberechtigungen verfügt. Um dieses Problem zu vermeiden, kann der Besitzer des Quell-Buckets andere AWS-Konten Berechtigungen zum Erstellen von Objekten unter bestimmten Bedingungen gewähren. Dazu sind explizite Zugriffsberechtigungen für diese Objekte erforderlich. Eine Beispielrichtlinie finden Sie unter Erteilung von kontoübergreifenden Berechtigungen für das Hochladen von Objekten, wobei sichergestellt wird, dass der Bucket-Eigentümer volle Kontrolle besitzt.
- Angenommen, Sie fügen einer Replikationskonfiguration eine Regel hinzu, um eine Teilmenge von Objekten mit einem spezifischen Tag zu replizieren. In diesem Fall müssen Sie den spezifischen Tag-Schlüssel und -Wert zum Zeitpunkt der Objekterstellung zuweisen, damit Amazon S3 das Objekt replizieren kann. Wenn Sie zuerst ein Objekt erstellen und dann dem vorhandenen Objekt das Tag hinzufügen, repliziert Amazon S3 das Objekt nicht.
- Verwenden Sie Amazon S3 S3-Ereignisbenachrichtigungen, um Sie über Fälle zu informieren, in denen Objekte nicht an ihr Ziel AWS-Region repliziert werden. Amazon S3 S3-Ereignisbenachrichtigungen sind über Amazon Simple Queue Service (AmazonSQS), Amazon Simple Notification Service (AmazonSNS) oder erhältlich AWS Lambda. Weitere Informationen finden Sie unter Erhalten von Amazon-S3-Ereignisbenachrichtigungen über Replikations-Fehlerereignisse.

Sie können Amazon-S3-Ereignisbenachrichtigungen auch verwenden, um die Ursachen für fehlgeschlagene Replikationen anzuzeigen. Eine Liste der Ursachen für fehlgeschlagene Replikationen finden Sie unter Gründe für das Fehlschlagen der Replikation in Amazon S3.

Fehler bei der Batchreplikation

Um Fehler bei Objekten zu beheben, die nicht in den Ziel-Bucket repliziert werden, überprüfen Sie die verschiedenen Berechtigungstypen für Ihre Buckets, die Replikationsrolle und die IAM Rolle, die zur Erstellung des Batch-Replikationsjobs verwendet wurde. Stellen Sie außerdem sicher, dass Sie die Einstellungen „Öffentlichen Zugriff blockieren“ und „S3-Objektbesitz“ für Ihre Buckets überprüfen.

Weitere Tipps zur Problembehandlung bei der Arbeit mit Batch Operations finden Sie unterthe section called “Problembehandlung bei Batch-Vorgängen”.

Während der Batchreplikation können die folgenden Fehler auftreten:

- Bei der Manifestgenerierung wurden keine Schlüssel gefunden, die den Filterkriterien entsprechen.

  Dieser Fehler tritt aus einem der folgenden Gründe auf:

  - Wenn Objekte im Quell-Bucket in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive gespeichert werden.

    Um die Batch-Replikation für diese Objekte zu verwenden, stellen Sie sie zunächst mithilfe eines Restore (`S3InitiateRestoreObjectOperation`) -Vorgangs in einem Batch Operations-Job in der Speicherklasse S3 Standard wieder her. Weitere Informationen finden Sie unter Wiederherstellen eines archivierten Objekts und Wiederherstellen von Objekten (Batch Operations). Nachdem Sie die Objekte wiederhergestellt haben, können Sie sie mithilfe eines Batchreplikationsauftrags replizieren.
  - Wenn die angegebenen Filterkriterien mit keinen gültigen Objekten im Quell-Bucket übereinstimmen.

    Überprüfen und korrigieren Sie die Filterkriterien. In der Regel für die Batch-Replikation suchen die Filterkriterien beispielsweise nach allen Objekten im Quell-Bucket mit dem Präfix`Tax/`. Wenn der Präfixname falsch eingegeben wurde, also mit einem Schrägstrich am Anfang und am Ende `/Tax/` statt nur am Ende, wurden keine S3-Objekte gefunden. Um den Fehler zu beheben, korrigieren Sie das Präfix, in diesem Fall von `/Tax/` bis, `Tax/` in der Replikationsregel.

- Die Batchoperation ist aus folgendem Grund fehlgeschlagen: Der Auftragsbericht konnte nicht in Ihren Berichts-Bucket geschrieben werden.

  Dieser Fehler tritt auf, wenn die IAM Rolle, die für den Batch Operations-Job verwendet wird, den Abschlussbericht nicht an dem Speicherort ablegen kann, der bei der Erstellung des Jobs angegeben wurde. Um diesen Fehler zu beheben, überprüfen Sie, ob die IAM Rolle über die `s3:PutObject` Berechtigung für den Bucket verfügt, in dem Sie den Abschlussbericht für Batch-Operationen speichern möchten. Wir empfehlen, den Bericht in einem anderen Bucket als dem Quell-Bucket bereitzustellen.

  Weitere Tipps zur Behebung dieses Fehlers finden Sie unterthe section called “Der Jobbericht wird nicht zugestellt, wenn ein Berechtigungsproblem vorliegt oder ein Aufbewahrungsmodus aktiviert ist”.
- Die Batchoperation wurde mit Fehlern abgeschlossen und die Gesamtzahl der Fehlschläge ist nicht 0.

  Dieser Fehler tritt auf, wenn bei dem ausgeführten Batchreplikationsauftrag unzureichende Objektberechtigungen vorhanden sind. Wenn Sie eine Replikationsregel für Ihren Batch-Replikationsauftrag verwenden, stellen Sie sicher, dass die IAM Rolle, die für die Replikation verwendet wird, über die richtigen Berechtigungen für den Zugriff auf Objekte aus dem Quell- oder Ziel-Bucket verfügt. Sie können auch den Abschlussbericht zur Batchreplikation anzeigen, um den spezifischen Grund für das Fehlschlagen der Replikation in Amazon S3 zu überprüfen.
- Der Batchauftrag wurde erfolgreich ausgeführt, die Anzahl der im Ziel-Bucket erwarteten Objekte stimmt nicht überein.

  Dieser Fehler tritt auf, wenn die im Manifest des Batchreplikationsauftrags aufgeführten Objekte nicht mit den Filtern übereinstimmen, die Sie während der Auftragserstellung ausgewählt haben. Möglicherweise erhalten Sie diese Meldung auch, wenn die Objekte in Ihrem Quell-Bucket keinen Replikationsregeln entsprechen und nicht im generierten Manifest enthalten sind.

Fehler bei Batch Operations nach dem Hinzufügen einer neuen Replikationsregel zu einer vorhandenen Replikationskonfiguration

Batch Operations versucht, eine vorhandene Objektreplikation für jede Regel in der Replikationskonfiguration des Quell-Buckets auszuführen. Wenn es Probleme mit einer der vorhandenen Replikationsregeln gibt, können Fehler auftreten.

Im Abschlussbericht des Batch-Operations-Auftrags werden die Ursachen für das Fehlschlagen des Auftrags erläutert. Eine Liste mit häufig auftretenden Fehlern finden Sie unter Gründe für das Fehlschlagen der Replikation in Amazon S3.

Überwachung der Replikation mit Metriken, Ereignisbenachrichtigungen und Status

Sie können Ihre Live-Replikationskonfigurationen und Ihre S3-Batch-Replikationsaufträge mithilfe der folgenden Mechanismen überwachen:

- S3-Replikationsmetriken — Wenn Sie S3-Replikationsmetriken aktivieren, CloudWatch gibt Amazon Metriken aus, mit denen Sie ausstehende Bytes, ausstehende Operationen und Replikationslatenz auf der Ebene der Replikationsregeln verfolgen können. Sie können S3-Replikationsmetriken über die Amazon S3 S3-Konsole und die CloudWatch Amazon-Konsole anzeigen. In der Amazon S3 S3-Konsole können Sie diese Metriken auf der Registerkarte Metriken des Quell-Buckets einsehen. Weitere Informationen zu S3-Replikationsmetriken finden Sie unterthe section called “Verwendung von S3-Replikationsmetriken”.
- S3 Storage Lens-Metriken — Zusätzlich zu den S3-Replikationsmetriken können Sie die replikationsbezogenen Datenschutzmetriken verwenden, die in den S3 Storage Lens-Dashboards bereitgestellt werden. Wenn Sie beispielsweise die kostenlosen Metriken in S3 Storage Lens verwenden, können Sie sich Metriken wie die Gesamtzahl der Byte, die aus dem Quell-Bucket repliziert wurden, oder die Anzahl der replizierten Objekte aus dem Quell-Bucket anzeigen lassen.

  Um Ihre allgemeine Einstellung zur Replikation zu überprüfen, können Sie erweiterte Metriken in S3 Storage Lens aktivieren. Mit den erweiterten Metriken in S3 Storage Lens können Sie sehen, über wie viele Replikationsregeln Sie verschiedener Typen verfügen, einschließlich der Anzahl der Replikationsregeln mit einem ungültigen Replikationsziel.

  Weitere Informationen zur Arbeit mit Replikationsmetriken in S3 Storage Lens finden Sie unterthe section called “Replikationsmetriken in den Dashboards von S3 Storage Lens anzeigen”.
- S3-Ereignisbenachrichtigungen — S3-Ereignisbenachrichtigungen können Sie auf Objektebene benachrichtigen, wenn Objekte nicht an ihr Ziel repliziert werden AWS-Region oder wenn Objekte nicht innerhalb bestimmter Schwellenwerte repliziert werden. S3-Ereignisbenachrichtigungen bieten die folgenden Replikationsereignistypen:`s3:Replication:OperationFailedReplication`,`s3:Replication:Ope` `s3:Replication:OperationReplicatedAfterThreshold` und. `s3:Replication:OperationNotTracked`

Amazon S3 S3-Ereignisse sind über Amazon Simple Queue Service (AmazonSQS), Amazon Simple Notification Service (AmazonSNS) oder verfügbar AWS Lambda. Weitere Informationen finden Sie unter the section called “Erhalten von Benachrichtigungen über Replikations-Fehlerereignisse”.

- Werte für den Replikationsstatus — Sie können auch den Replikationsstatus Ihrer Objekte abrufen. Anhand des Replikationsstatus können Sie den aktuellen Status eines Objekts ermitteln, das repliziert wird. Der Replikationsstatus eines Quellobjekts gibt entweder `PENDING`, `COMPLETED` oder `FAILED` zurück. Der Replikationsstatus eines Replikats gibt `REPLICA` zurück.

  Sie können auch Werte für den Replikationsstatus verwenden, wenn Sie S3-Batch-Replikationsaufträge erstellen. Sie können diese Statuswerte beispielsweise verwenden, um Objekte zu replizieren, die entweder nie repliziert wurden oder deren Replikation fehlgeschlagen ist.

  Weitere Informationen zum Abrufen des Replikationsstatus Ihrer Objekte finden Sie unter. the section called “Abrufen des Replikationsstatus” Weitere Hinweise zur Verwendung dieser Werte mit der Batch-Replikation finden Sie unterthe section called “Filter für einen Batch-Replikationsauftrag”.

Themen

- Verwendung von S3-Replikationsmetriken
- Replikationsmetriken in den Dashboards von S3 Storage Lens anzeigen
- Erhalten von Amazon-S3-Ereignisbenachrichtigungen über Replikations-Fehlerereignisse
- Abrufen von Replikationsstatusinformationen

Verwendung von S3-Replikationsmetriken

S3-Replikationsmetriken enthalten detaillierte Metriken für die Replikationsregeln in Ihrer Replikationskonfiguration. Mit Replikationsmetriken können Sie den minute-by-minute Fortschritt überwachen, indem Sie ausstehende Bytes, ausstehende Operationen, Operationen, bei denen die Replikation fehlgeschlagen ist, und die Replikationslatenz verfolgen.

> **Note**
>
> - S3-Replikationsmetriken werden zum gleichen Tarif wie CloudWatch benutzerdefinierte Amazon-Metriken abgerechnet. Weitere Informationen finden Sie unter CloudWatchAmazon-Preise.
> - Wenn Sie S3 Replication Time Control verwenden, CloudWatch beginnt Amazon 15 Minuten, nachdem Sie S3 RTC für die jeweilige Replikationsregel aktiviert haben, mit der Meldung von Replikationsmetriken.

S3-Replikationsmetriken werden automatisch aktiviert, wenn Sie S3 Replication Time Control (S3RTC) aktivieren. Sie können S3-Replikationsmetriken auch unabhängig von S3 aktivieren, RTC während Sie eine Regel erstellen oder bearbeiten. S3 RTC umfasst weitere Funktionen, wie z. B. ein Service Level Agreement (SLA) und Benachrichtigungen bei Überschreitung von Schwellenwerten. Weitere Informationen finden Sie unter Erfüllung von Compliance-Anforderungen mit S3 Replication Time Control.

Wenn S3-Replikationsmetriken aktiviert sind, veröffentlicht Amazon S3 die folgenden Metriken auf Amazon CloudWatch. CloudWatchDie Metriken werden nach bestem Wissen und Gewissen bereitgestellt.

Metrikname	Beschreibung der Metrik	Für welche Objekte gilt diese Metrik?	In welcher Region wird diese Metrik veröffent licht?	Wird diese Metrik immer noch veröffentlicht, wenn der Ziel-Buck et gelöscht wird?	Wird diese Metrik weiterhin veröffentlicht, wenn keine Replikation stattfindet?
Noch ausstehen de Bytes zur Replikation	Die Gesamtanz ahl der Bytes von Objekten, deren Replikati on für eine	Diese Metrik gilt nur für neue Objekte, die mit S3 Cross-Reg ion Replicati	Diese Metrik wird in der Region des Ziel-Buckets veröffentlicht.	Nein	Ja

Metrikname	Beschreibung der Metrik	Für welche Objekte gilt diese Metrik?	In welcher Region wird diese Metrik veröffentlicht?	Wird diese Metrik immer noch veröffentlicht, wenn der Ziel-Bucket gelöscht wird?	Wird diese Metrik weiterhin veröffentlicht, wenn keine Replikation stattfindet?
	bestimmte Replikationsregel aussteht.	on (S3CRR) oder S3 Same-Region Replication (S3) repliziert werden. SRR			
Latenz der Replikation	Die maximale Anzahl von Sekunden, um die sich der Replizierungsziel-Bucket für eine bestimmte Replikationsregel hinter dem Quell-Bucket befindet.	Diese Metrik gilt nur für neue Objekte, die mit S3 CRR oder S3 SRR repliziert werden.	Diese Metrik wird in der Region des Ziel-Buckets veröffentlicht.	Nein	Ja

Metrikname	Beschreibung der Metrik	Für welche Objekte gilt diese Metrik?	In welcher Region wird diese Metrik veröffentlicht?	Wird diese Metrik immer noch veröffentlicht, wenn der Ziel-Bucket gelöscht wird?	Wird diese Metrik weiterhin veröffentlicht, wenn keine Replikation stattfindet?
Operationen, die noch repliziert	Die Anzahl der Operationen, deren Replikation für eine bestimmte Replikationsregel aussteht. Diese Metrik verfolgt Vorgänge im Zusammenhang mit Objekten, Löschmarkierungen, Tags, Zugriffskontrolllisten (ACLs) und S3 Object Lock.	Diese Metrik gilt nur für neue Objekte, die mit S3 CRR oder S3 SRR repliziert werden.	Diese Metrik wird in der Region des Ziel-Buckets veröffentlicht.	Nein	Ja

Metrikname	Beschreibung der Metrik	Für welche Objekte gilt diese Metrik?	In welcher Region wird diese Metrik veröffentlicht?	Wird diese Metrik immer noch veröffentlicht, wenn der Ziel-Bucket gelöscht wird?	Wird diese Metrik weiterhin veröffentlicht, wenn keine Replikation stattfindet?
Operationen sind fehlgeschlagen, Replikation	Die Anzahl von Vorgängen, bei denen die Replikation für eine bestimmte Replikationsregel fehlgeschlagen ist. Diese Metrik verfolgt Vorgänge im Zusammenhang mit Objekten, Löschmarkierungen, Tags, Zugriffskontrolllisten (ACLs) und Objektsperren. Operationen fehlgesch	Diese Metrik gilt sowohl für neue Objekte, die mit S3 CRR oder S3 SRR repliziert werden, als auch für vorhandene Objekte, die mit S3 Batch Replication repliziert werden. **Note** Wenn ein S3-Batch-Replikationsauftrag überhaupt nicht ausgeführ	Diese Metrik wird in der Region des Quell-Buckets veröffentlicht.	Ja	Nein

Metrikname	Beschreibung der Metrik	Für welche Objekte gilt diese Metrik?	In welcher Region wird diese Metrik veröffentlicht?	Wird diese Metrik immer noch veröffentlicht, wenn der Ziel-Bucket gelöscht wird?	Wird diese Metrik weiterhin veröffentlicht, wenn keine Replikation stattfindet?
	lagen Replikation verfolgt S3-Replikationsfehler zusammengefasst in einem minütlichen Intervall. Um die spezifischen Objekte zu ermitteln, bei denen die Replikation fehlgeschlagen ist, und die Fehlerursachen herauszufinden, abonnieren Sie das `OperationFailedReplication` -Ereignis in den Amazon-	t werden kann, werden keine Metriken an Amazon gesendet CloudWatch. Ihr Auftrag wird beispielsweise nicht ausgeführt, wenn Sie nicht über die erforderlichen Berechtigungen verfügen,			

Metrikname	Beschreibung der Metrik	Für welche Objekte gilt diese Metrik?	In welcher Region wird diese Metrik veröffentlicht?	Wird diese Metrik immer noch veröffentlicht, wenn der Ziel-Bucket gelöscht wird?	Wird diese Metrik weiterhin veröffentlicht, wenn keine Replikation stattfindet?
	S3-Ereignisbenachrichtigungen. Weitere Informationen finden Sie unter Erhalten von Amazon-S3-Ereignisbenachrichtigungen über Replikations-Fehlereignisse.	um einen S3-Batch-Replikationsauftrag auszuführen, oder wenn die Tags oder das Präfix in Ihrer Replikationskonfiguration nicht übereinstimmen.			

Informationen zur Arbeit mit diesen Metriken CloudWatch finden Sie unterthe section called “Metriken zur S3-Replikation in CloudWatch”.

Aktivieren von S3-Replikationsmetriken

Sie können S3-Replikationsmetriken mit einer neuen oder einer vorhandenen Replikationsregel verwenden. Alle Informationen zum Erstellen von Replikationsregeln finden Sie unter Konfiguration der Replikation für Buckets im selben Konto. Sie können die Replikationsregel auf einen gesamten S3-Bucket oder auf Amazon-S3-Objekte mit einem bestimmten Präfix oder Tag anwenden.

Dieses Thema enthält Anleitungen zum Aktivieren von S3-Replikationsmetriken in Ihrer Replikationskonfiguration, wenn sich die Quell- und Ziel-Buckets im Eigentum desselben Kontos oder unterschiedlicher AWS-Konten befinden.

Um Replikationsmetriken mithilfe von AWS Command Line Interface (AWS CLI) zu aktivieren, müssen Sie dem Quell-Bucket eine Replikationskonfiguration mit `Metrics` enabled hinzufügen. In dieser Beispielkonfiguration werden Objekte unter dem Präfix *`Tax`* auf den Ziel-Bucket *`amzn-s3-demo-bucket`* repliziert und es werden Metriken für diese Objekte generiert.

```
{
    "Rules": [
        {
            "Status": "Enabled",
            "Filter": {
                "Prefix": "Tax"
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::amzn-s3-demo-bucket",
                "Metrics": {
                    "Status": "Enabled"
                }
            },
            "Priority": 1
        }
    ],
    "Role": "IAM-Role-ARN"
}
```

Anzeigen von Replikationsmetriken

Sie können S3-Replikationsmetriken auf der Registerkarte Metriken des Quell-Buckets in der Amazon S3 S3-Konsole anzeigen. Diese CloudWatch Amazon-Metriken sind auch in der CloudWatch

Amazon-Konsole verfügbar. Wenn Sie S3-Replikationsmetriken aktivieren, CloudWatch gibt Amazon Metriken aus, mit denen Sie ausstehende Bytes, ausstehende Operationen und Replikationslatenz auf der Ebene der Replikationsregeln verfolgen können.

S3-Replikationsmetriken werden automatisch aktiviert, wenn Sie die Replikation mit S3 Replication Time Control (S3RTC) über die Amazon S3-Konsole oder Amazon S3 aktivieren RESTAPI. Sie können S3-Replikationsmetriken auch unabhängig von S3 aktivieren, RTC während Sie eine Regel erstellen oder bearbeiten.

Wenn Sie S3 Replication Time Control verwenden, CloudWatch beginnt Amazon 15 Minuten, nachdem Sie S3 RTC für die jeweilige Replikationsregel aktiviert haben, mit der Meldung von Replikationsmetriken. Weitere Informationen finden Sie unter Verwendung von S3-Replikationsmetriken.

Replikationsmetriken verfolgen die Regel IDs der Replikationskonfiguration. Eine Replikationsregel-ID kann spezifisch für ein Präfix, ein Tag oder eine Kombination aus beiden sein.

Weitere Informationen zu CloudWatch Metriken für Amazon S3 finden Sie unterMetriken mit Amazon überwachen CloudWatch.

Voraussetzungen

Erstellen Sie eine Replikationsregel, für die S3-Replikationsmetriken aktiviert sind. Weitere Informationen finden Sie unter the section called “Aktivieren von Replikationsmetriken”.

Um S3-Replikationsmetriken über den Tab Metriken des Quell-Buckets anzuzeigen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Allgemeine Buckets den Namen des Quell-Buckets aus, der die Objekte enthält, für die Sie Replikationsmetriken benötigen.
4. Wählen Sie die Registerkarte Metriken.
5. Wählen Sie unter Replikationsmetriken die Replikationsregeln aus, für die Sie Metriken anzeigen möchten.
6. Wählen Sie Display charts (Diagramme anzeigen).

Amazon S3 zeigt für die von Ihnen ausgewählten Regeln Diagramme für die Replikationslatenz, zur Replikation ausstehende Byte, zur Replikation ausstehende Vorgänge und zur fehlgeschlagenen Replikation von Vorgängen an.

Replikationsmetriken in den Dashboards von S3 Storage Lens anzeigen

Zusätzlich zu den S3-Replikationsmetriken können Sie die replikationsbezogenen Datenschutzmetriken verwenden, die von S3 Storage Lens bereitgestellt werden. S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Weitere Informationen finden Sie unter Verwenden von S3 Storage Lens zum Schutz Ihrer Daten.

S3 Storage Lens bietet zwei Stufen von Metriken: kostenlose Metriken und erweiterte Metriken und Empfehlungen, auf die Sie gegen Aufpreis ein Upgrade durchführen können. Mit erweiterten Metriken und Empfehlungen können Sie auf zusätzliche Metriken und Funktionen zugreifen, um Erkenntnisse über Ihren Speicher zu erhalten. Weitere Informationen zu den Preisen für S3 Storage Lens finden Sie unter Amazon S3-Preise.

Wenn Sie die kostenlosen Metriken in S3 Storage Lens verwenden, können Sie sich Metriken wie die Gesamtzahl der Byte, die aus dem Quell-Bucket repliziert wurden, oder die Anzahl der replizierten Objekte aus dem Quell-Bucket anzeigen lassen.

Um Ihre allgemeine Einstellung zur Replikation zu überprüfen, können Sie erweiterte Metriken in S3 Storage Lens aktivieren. Mit den erweiterten Metriken in S3 Storage Lens können Sie sehen, über wie viele Replikationsregeln Sie verschiedener Typen verfügen, einschließlich der Anzahl der Replikationsregeln mit einem ungültigen Replikationsziel.

Eine vollständige Liste der S3 Storage Lens-Metriken, einschließlich der Replikationsmetriken in den einzelnen Stufen, finden Sie im Glossar zu S3 Storage Lens-Metriken.

Voraussetzungen

Erstellen Sie eine Live-Replikationskonfiguration oder einen S3-Batch-Replikationsauftrag.

So zeigen Sie Replikationsmetriken in Amazon S3 Storage Lens an

1. Erstellen Sie ein S3 Storage Lens-Dashboard. step-by-stepAnweisungen finden Sie unterthe section called “Verwenden der S3-Konsole”.

2. (Optional) Wenn Sie bei der Einrichtung Ihres Dashboards alle Replikationsmetriken für S3 Storage Lens sehen möchten, wählen Sie Erweiterte Metriken und Empfehlungen und dann Erweiterte Datenschutzmetriken aus. Eine vollständige Liste der Messwerte finden Sie im Glossar zu S3 Storage Lens-Metriken.

   Wenn Sie erweiterte Metriken und Empfehlungen aktivieren, können Sie weitere Einblicke in Ihre Replikationskonfigurationen gewinnen. Sie können beispielsweise Messwerte zur Anzahl der Replikationsregeln von S3 Storage Lens verwenden, um detaillierte Informationen zu Ihren Buckets zu erhalten, die für die Replikation konfiguriert sind. Zu diesen Informationen gehören Replikationsregeln innerhalb von Buckets und Regionen sowie Bucket- und regionsübergreifende Replikationsregeln. Weitere Informationen finden Sie unter the section called “Zählen der Gesamtzahl der Replikationsregeln für jeden Bucket”.

3. Nachdem Sie Ihr Dashboard erstellt haben, öffnen Sie das Dashboard und wählen Sie den Tab Buckets.

4. Scrollen Sie nach unten zum Abschnitt Buckets. Wählen Sie unter Metrics categories (Metrikkategorien) die Option Data protection (Datenschutz) aus. Löschen Sie dann Summary (Zusammenfassung).

5. Um die Buckets-Liste so zu filtern, dass nur Replikationsmetriken angezeigt werden, wählen Sie das Einstellungen-Symbol
   () .

6. Deaktivieren Sie die Schalter für alle Datenschutzmetriken, bis nur noch die Replikationsmetriken ausgewählt sind.

7. (Optional) Wählen Sie unter Page size (Seitengröße) die Anzahl der Buckets aus, die in der Liste angezeigt werden sollen.

8. Klicken Sie auf Weiter.

Erhalten von Amazon-S3-Ereignisbenachrichtigungen über Replikations-Fehlerereignisse

Wenn Sie S3-Replikationsmetriken in Ihrer Replikationskonfiguration aktiviert haben, können Sie Amazon S3 S3-Ereignisbenachrichtigungen einrichten, um Sie zu benachrichtigen, wenn Objekte nicht an ihr Ziel AWS-Region repliziert werden. Wenn Sie S3 Replication Time Control (S3RTC) in Ihrer Replikationskonfiguration aktiviert haben, können Sie auch benachrichtigt werden, wenn Objekte nicht innerhalb des RTC S3-Schwellenwerts von 15 Minuten repliziert werden.

Mithilfe der folgenden `Replication` Ereignistypen können Sie den minute-by-minute Fortschritt von Replikationsereignissen überwachen, indem Sie ausstehende Bytes, ausstehende Vorgänge und die Replikationslatenz verfolgen. Weitere Informationen zu S3-Replikationsmetriken finden Sie unterVerwendung von S3-Replikationsmetriken.

- Der `s3:Replication:OperationFailedReplication`-Ereignistyp benachrichtigt Sie, wenn ein Objekt, das für die Replikation berechtigt war, nicht repliziert werden konnte.
- Der `s3:Replication:OperationMissedThreshold` Ereignistyp benachrichtigt Sie, wenn ein Objekt, das für die Replikation in Frage kam und S3 verwendet, den 15-minütigen Schwellenwert für die Replikation RTC überschreitet.
- Der `s3:Replication:OperationReplicatedAfterThreshold` Ereignistyp benachrichtigt Sie, wenn ein Objekt, das für die Replikation in Frage kam und S3 verwendet, nach dem RTC 15-Minuten-Schwellenwert repliziert wird.
- Der `s3:Replication:OperationNotTracked` Ereignistyp benachrichtigt Sie, wenn ein Objekt, das für die Live-Replikation in Frage kam (entweder Replikation in derselben Region [SRR] oder regionsübergreifende Replikation [CRR]), nicht mehr anhand von Replikationsmetriken verfolgt wird.

Vollständige Beschreibungen aller unterstützten Replikationsereignistypen finden Sie unter. the section called “Unterstützte Ereignistypen für SQSSNS, und Lambda”

Eine Liste der von S3-Ereignisbenachrichtigungen erfassten Fehlercodes finden Sie unterGründe für das Fehlschlagen der Replikation in Amazon S3.

Sie können S3-Ereignisbenachrichtigungen über Amazon Simple Queue Service (AmazonSQS), Amazon Simple Notification Service (AmazonSNS) oder erhalten AWS Lambda. Weitere Informationen finden Sie unter the section called “Amazon-S3-Ereignis-Benachrichtigungen”.

Anweisungen zum Konfigurieren von Amazon-S3-Ereignisbenachrichtigungen finden Sie unter Aktivieren von Ereignisbenachrichtigungen.

> **Note**
>
> Stellen Sie sicher, dass Sie nicht nur Ereignisbenachrichtigungen aktivieren, sondern auch S3-Replikationsmetriken aktivieren. Weitere Informationen finden Sie unter the section called “Aktivieren von Replikationsmetriken”.

Das Folgende ist ein Beispiel für eine Nachricht, die Amazon S3 sendet, um ein `s3:Replication:OperationFailedReplication`-Ereignis zu veröffentlichen. Weitere Informationen finden Sie unter the section called “Struktur von Ereignismeldungen”.


```
{
  "Records": [
    {
      "eventVersion": "2.2",
      "eventSource": "aws:s3",
      "awsRegion": "us-east-1",
      "eventTime": "2024-09-05T21:04:32.527Z",
      "eventName": "Replication:OperationFailedReplication",
      "userIdentity": {
        "principalId": "s3.amazonaws.com"
      },
      "requestParameters": {
        "sourceIPAddress": "s3.amazonaws.com"
      },
      "responseElements": {
        "x-amz-request-id": "123bf045-2b4b-4ca8-a211-c34a63c59426",
        "x-amz-id-2":
 "12VAWNDIHnwJsRhTccqQTeAPoXQmRt22KkewMV8G3XZihAuf9CLDdmkApgZzudaIe2KlLfDqGS0="
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "ReplicationEventName",
        "bucket": {
          "name": "amzn-s3-demo-bucket1",
          "ownerIdentity": {
            "principalId": "111122223333"
          },
          "arn": "arn:aws:s3:::amzn-s3-demo-bucket1"
        },
        "object": {
          "key": "replication-object-put-test.png",
          "size": 520080,
          "eTag": "e12345ca7e88a38428305d3ff7fcb99f",
          "versionId": "abcdeH0Xp66ep__QDjR76LK7Gc9X4wKO",
          "sequencer": "0066DA1CBF104C0D51"
        }
      },
      "replicationEventData": {
        "replicationRuleId": "notification-test-replication-rule",
```

```
        "destinationBucket": "arn:aws:s3:::amzn-s3-demo-bucket2",
        "s3Operation": "OBJECT_PUT",
        "requestTime": "2024-09-05T21:03:59.168Z",
        "failureReason": "AssumeRoleNotPermitted"
      }
    }
  ]
}
```

Gründe für das Fehlschlagen der Replikation in Amazon S3

Die folgende Tabelle enthält Gründe für das Fehlschlagen der Replikation in Amazon S3. Sie können sich diese Gründe ansehen, indem Sie das `s3:Replication:OperationFailedReplication` Ereignis mit Amazon S3 S3-Ereignisbenachrichtigungen erhalten und sich dann den `failureReason` Wert ansehen.

Sie können diese Fehlergründe auch in einem Abschlussbericht der S3-Batch-Replikation anzeigen. Weitere Informationen finden Sie unter Abschlussbericht zur Batchreplikation.

Gründe für das Fehlschlagen der Replikation	Beschreibung
`AssumeRoleNotPermitted`	Amazon S3 kann die Rolle AWS Identity and Access Management (IAM) nicht annehmen, die in der Replikationskonfiguration oder im Batch Operations-Job angegeben ist.
`DstBucketInvalidRegion`	Der Ziel-Bucket befindet sich nicht in demselben AWS-Region Bucket, der im Batch Operations-Job angegeben wurde. Dieser Fehler ist spezifisch für die Batchrepl ikation.
`DstBucketNotFound`	Amazon S3 kann den in der Replikati onskonfiguration angegebenen Ziel-Bucket nicht finden.
`DstBucketObjectLockConfigMissing`	Um Objekte aus einem Quell-Bucket mit aktivierter Objektsperre zu replizieren, muss für den Ziel-Bucket ebenfalls die Objektspe rre aktiviert sein. Dieser Fehler zeigt an, dass

Gründe für das Fehlschlagen der Replikation	Beschreibung
	die Objektsperre im Ziel-Bucket möglicher weise nicht aktiviert ist. Weitere Informationen finden Sie unter Überlegungen zu Object Lock.
`DstBucketUnversioned`	Die Versionsverwaltung ist für den S3-Ziel-Bucket nicht aktiviert. Aktivieren Sie die Versionsverwaltung für den Ziel-Bucket, um Objekte mit S3-Replikation zu replizieren.
`DstDelObjNotPermitted`	Amazon S3 kann Löschmarkierungen nicht in den angegebenen Ziel-Bucket replizier en. Möglicherweise fehlt die `s3:Replic ateDelete` -Berechtigung für den Ziel-Bucket.
`DstKmsKeyInvalidState`	Der Schlüssel AWS Key Management Service (AWS KMS) für den Ziel-Bucket ist nicht gültig. Überprüfen und aktivieren Sie den erforderlichen AWS KMS Schlüssel . Weitere Informationen zur Verwaltung von AWS KMS Schlüsseln finden Sie unter Schlüsselstatus von AWS KMS Schlüsseln im AWS Key Management Service Entwickle rhandbuch.
`DstKmsKeyNotFound`	Der AWS KMS Schlüssel, der für den Ziel-Bucket in der Replikationskonfiguration konfiguriert ist, ist nicht vorhanden.
`DstMultipartCompleteNotPermitted`	Amazon S3 kann mehrteilige Uploads von Objekten im Ziel-Bucket nicht abschließ en. Möglicherweise fehlt die `s3:Replic ateObject` -Berechtigung für den Ziel-Bucket.

Gründe für das Fehlschlagen der Replikation	Beschreibung
`DstMultipartInitNotPermitted`	Amazon S3 kann mehrteilige Uploads von Objekten in den Ziel-Bucket nicht initiieren. Möglicherweise fehlt die `s3:ReplicateObject` -Berechtigung für den Ziel-Bucket.
`DstMultipartPartUploadNotPermitted`	Amazon S3 kann keine mehrteiligen Upload-Objekte in den Ziel-Bucket hochladen. Möglicherweise fehlt die `s3:ReplicateObject` -Berechtigung für den Ziel-Bucket.
`DstObjectHardDeleted`	Die S3-Batchreplikation unterstützt nicht das erneute Replizieren von Objekten, die mit der Versions-ID des Objekts aus dem Ziel-Bucket gelöscht wurden. Dieser Fehler ist spezifisch für die Batchreplikation.
`DstPutAclNotPermitted`	Amazon S3 ist nicht in der Lage, Objektzugriffskontrolllisten (ACLs) in den Ziel-Bucket zu replizieren. Möglicherweise fehlt die `s3:ReplicateObject` -Berechtigung für den Ziel-Bucket.
`DstPutLegalHoldNotPermitted`	Amazon S3 kann keine Objektsperre aufgrund gesetzlicher Aufbewahrungsfristen für die Zielobjekte festlegen, während unveränderliche Objekte repliziert werden. Möglicherweise fehlt die `s3:PutObjectLegalHold` -Berechtigung für den Ziel-Bucket. Weitere Informationen finden Sie unter Rechtliche Aufbewahrungsfristen.

Gründe für das Fehlschlagen der Replikation	Beschreibung
`DstPutObjectNotPermitted`	Amazon S3 kann Objekte nicht in den angegebenen Ziel-Bucket replizieren. Möglicherweise fehlen die `s3:ReplicateObject` - oder `s3:ObjectOwnerOverrideToBucketOwner` -Berechtigungen für den Ziel-Bucket.
`DstPutTaggingNotPermitted`	Amazon S3 kann Objekt-Tags nicht in den angegebenen Ziel-Bucket replizieren. Möglicherweise fehlt die `s3:ReplicateObject` -Berechtigung für den Ziel-Bucket.
`DstVersionNotFound`	Amazon S3 kann die erforderliche Objektversion im Ziel-Bucket nicht finden, für den Metadaten repliziert werden müssen.
`InitiateReplicationNotPermitted`	Amazon S3 kann keine Replikation für Objekte initiieren. Möglicherweise fehlt die `s3:InitiateReplication` -Berechtigung für den Batch-Operationsauftrag. Dieser Fehler ist spezifisch für die Batchreplikation.
`SrcBucketInvalidRegion`	Der Quell-Bucket befindet sich nicht in demselben AWS-Region Bucket, der im Batch Operations-Job angegeben wurde. Dieser Fehler ist spezifisch für die Batchreplikation.
`SrcBucketNotFound`	Amazon S3 kann den Quell-Bucket nicht finden.
`SrcBucketReplicationConfigMissing`	Amazon S3 konnte keine Replikationskonfiguration für den Quell-Bucket finden.

Gründe für das Fehlschlagen der Replikation	Beschreibung
`SrcGetAclNotPermitted`	Amazon S3 kann nicht auf das Objekt im Quell-Bucket für die Replikation zugreifen . Möglicherweise fehlt die `s3:GetObjectVersionAcl` -Berechtigung für das Quell-Bucket-Objekt. Die Objekte im Quell-Bucket müssen sich im Besitz des Bucket-Eigentümers befinden. Wenn ACLs diese Option aktiviert ist, überprüfen Sie, ob „Objekteigentum“ auf „Bucket Owner Preferred“ oder „Object Writer“ gesetzt ist. Wenn Object Ownership auf Bucket Owner Preferred gesetzt ist, müssen die Quell-Bucket-Objekte die Eigenschaft `bucket-owner-full-control` ACL haben, damit der Bucket-Besitzer zum Objekteigentümer wird. Das Quellkonto kann das Eigentum an allen Objekten in seinem Bucket übernehmen, indem es die Option Objekteigentum auf Bucket Ownerforced setzt und deaktivie rtACLs.
`SrcGetLegalHoldNotPermitted`	Amazon S3 kann nicht auf die Informati onen zur gesetzlichen Aufbewahrung von S3 Object Lock zugreifen.
`SrcGetObjectNotPermitted`	Amazon S3 kann nicht auf das Objekt im Quell-Bucket für die Replikation zugreifen . Möglicherweise fehlt die `s3:GetObjectVersionForReplication` -Berechti gung für den Quell-Bucket.

Gründe für das Fehlschlagen der Replikation	Beschreibung
`SrcGetRetentionNotPermitted`	Amazon S3 kann nicht auf die Informationen zum Aufbewahrungszeitraum von S3 Object Lock zugreifen.
`SrcGetTaggingNotPermitted`	Amazon S3 kann nicht auf Objekt-Tag-Informationen aus dem Quell-Bucket zugreifen. Möglicherweise fehlt die `s3:GetObjectVersionTagging` -Berechtigung für den Quell-Bucket.
`SrcHeadObjectNotPermitted`	Amazon S3 kann keine Objektmetadaten aus dem Quell-Bucket abrufen. Möglicherweise fehlt die `s3:GetObjectVersionForReplication` -Berechtigung für den Quell-Bucket.
`SrcKeyNotFound`	Amazon S3 kann den Quellobjektschlüssel für die Replikation nicht finden. Das Quellobjekt wurde möglicherweise gelöscht, bevor die Replikation abgeschlossen war.
`SrcKmsKeyInvalidState`	Der AWS KMS Schlüssel für den Quell-Bucket hat keinen gültigen Status. Überprüfen und aktivieren Sie den erforderlichen AWS KMS Schlüssel. Weitere Informationen zur Verwaltung von AWS KMS Schlüsseln finden Sie unter Schlüsselstatus von AWS KMS Schlüsseln im AWS Key Management Service Entwicklerhandbuch.
`SrcObjectNotEligible`	Einige Objekte kommen nicht für die Replikation infrage. Dies kann an der Speicherklasse des Objekts liegen oder daran, dass die Objekt-Tags nicht mit der Replikationskonfiguration übereinstimmen.

Gründe für das Fehlschlagen der Replikation	Beschreibung
`SrcObjectNotFound`	Das Quellobjekt ist nicht vorhanden.
`SrcReplicationNotPending`	Amazon S3 hat dieses Objekt bereits replizier t. Für dieses Objekt steht keine Replikation mehr aus.
`SrcVersionNotFound`	Amazon S3 kann den Quellobjektversion für die Replikation nicht finden. Die Quellobje ktversion wurde möglicherweise gelöscht, bevor die Replikation abgeschlossen war.

Verwandte Themen

Berechtigungen für die Live-Replikation einrichten

Fehlerbehebung bei einer Replikation

Abrufen von Replikationsstatusinformationen

Der Replikationsstatus hilft Ihnen, den aktuellen Status eines derzeit replizierten Objekts zu bestimmen. Der Replikationsstatus eines Quellobjekts gibt entweder `PENDING`, `COMPLETED` oder `FAILED` zurück. Der Replikationsstatus eines Replikats gibt `REPLICA` zurück.

Sie können auch Werte für den Replikationsstatus verwenden, wenn Sie S3-Batch-Replikationsaufträge erstellen. Sie können diese Statuswerte beispielsweise verwenden, um Objekte zu replizieren, die entweder nie repliziert wurden oder deren Replikation fehlgeschlagen ist. Weitere Hinweise zur Verwendung dieser Werte mit der Batch-Replikation finden Sie unterthe section called “Verwenden von Replikationsstatusinformationen mit Batch-Replikationsaufträgen”.

Themen

- Übersicht über den Replikationsstatus
- Replikationsstatus bei einer Replikation auf mehrere Ziel-Buckets
- Replikationsstatus, wenn die Synchronisierung von Amazon-S3-Replikatänderungen aktiviert ist
- Verwenden von Replikationsstatusinformationen mit Batch-Replikationsaufträgen
- Finden des Replikationsstatus

Übersicht über den Replikationsstatus

Bei der Replikation haben Sie einen Quell-Bucket, in dem Sie die Replikation konfigurieren, und einen oder mehrere Ziel-Buckets, in denen Amazon S3 Objekte repliziert. Wenn Sie ein Objekt (mithilfe`GetObject`) oder Objektmetadaten (mithilfe von`HeadObject`) aus diesen Buckets anfordern, gibt Amazon S3 den `x-amz-replication-status` Header in der Antwort zurück:

- Wenn Sie ein Objekt aus dem Quell-Bucket anfordern, gibt Amazon S3 den Header `x-amz-replication-status` zurück, wenn das Objekt in der Anforderung für die Replikation geeignet ist.

  Nehmen wir beispielsweise an, dass Sie in Ihrer Replikationskonfiguration das Objektpräfix `TaxDocs` angeben, um Amazon S3 anzuweisen, nur Objekte mit dem Schlüsselnamenpräfix `TaxDocs` zu replizieren. Alle Objekte mit diesem Schlüsselnamenpräfix, die Sie hochladen, z. B. `TaxDocs/document1.pdf`, werden repliziert. Für Objektanforderungen mit diesem Schlüsselnamenpräfix gibt Amazon S3 den Header `x-amz-replication-status` mit einem der folgenden Werte für den Replikationsstatus des Objekts zurück: `PENDING`, `COMPLETED` oder `FAILED`.

  > **Note**
  >
  > Wenn nach dem Hochladen eines Objekts die Objektreplikation fehlschlägt, können Sie die fehlgeschlagene Replikation nicht erneut durchzuführen versuchen. Sie müssen das Objekt erneut hochladen, oder Sie müssen S3 Batch Replication verwenden, um alle fehlgeschlagenen Objekte zu replizieren. Weitere Informationen zur Verwendung der Batch-Replikation finden Sie unterthe section called “Replizieren vorhandener Objekte”. Objekte wechseln aufgrund von Problemen wie fehlenden Rollenberechtigungen, AWS Key Management Service (AWS KMS) -Berechtigungen oder Bucket-Berechtigungen für die Replikation in einen bestimmten `FAILED` Status. Bei vorübergehenden Ausfällen, z. B. wenn ein Bucket oder eine Region nicht verfügbar ist, wird der Replikationsstatus nicht auf den Status `FAILED` umgestellt, sondern bleibt bestehen`PENDING`. Nachdem die Ressource wieder online ist, setzt Amazon S3 die Replikation dieser Objekte fort.

- Wenn Sie ein Objekt aus einem Ziel-Bucket anfordern und es sich bei dem Objekt Ihrer Anforderung um ein Replikat handelt, das Amazon S3 erstellt hat, gibt Amazon S3 den Header `x-amz-replication-status` mit dem Wert `REPLICA` zurück.

> **Note**
>
> Bevor Sie ein Objekt aus einem Quell-Bucket löschen, für das die Replikation aktiviert ist, überprüfen Sie den Replikationsstatus des Objekts, um sicherzustellen, dass das Objekt repliziert wurde.
> Wenn eine S3-Lifecycle-Konfiguration im Quell-Bucket aktiviert ist, setzt Amazon S3 Lebenszyklusaktionen aus, bis der Status der Objekte entweder als oder `COMPLETED` markiert wird. `FAILED`

Replikationsstatus bei einer Replikation auf mehrere Ziel-Buckets

Wenn Sie Objekte in mehrere Ziel-Buckets replizieren, verhält sich der Header `x-amz-replication-status` anders. Der Header des Quellobjekts gibt `COMPLETED` nur dann den Wert zurück, wenn die Replikation an alle Ziele erfolgreich ist. Der Header bleibt auf dem Wert `PENDING`, bis die Replikation für alle Ziele abgeschlossen ist. Wenn die Replikation bei einem oder mehreren Zielen fehlschlägt, liefert der Header `FAILED`.

Replikationsstatus, wenn die Synchronisierung von Amazon-S3-Replikatänderungen aktiviert ist

Wenn in Ihren Replikationsregeln die Synchronisierung von Amazon-S3-Replikatänderungen aktiviert ist, können Replikate einen anderen Status als `REPLICA` melden. Wenn Änderungen an Metadaten gerade repliziert werden, gibt der `x-amz-replication-status`-Header den Wert `PENDING` zurück. Wenn die Synchronisierung der Replikatänderung Metadaten nicht repliziert, gibt der Header `FAILED` zurück. Wenn Metadaten korrekt repliziert werden, geben die Replikate den Header `REPLICA` zurück.

Verwenden von Replikationsstatusinformationen mit Batch-Replikationsaufträgen

Wenn Sie einen Batch-Replikationsauftrag erstellen, können Sie optional zusätzliche Filter angeben, z. B. das Erstellungsdatum des Objekts und den Replikationsstatus, um den Umfang des Jobs zu reduzieren.

Sie können Objekte filtern, die repliziert werden sollen, basierend auf dem `ObjectReplicationStatuses`-Wert, indem Sie einen oder mehrere der folgenden Werte angeben:

- `"NONE"` – Zeigt an, dass Amazon S3 noch nie versucht hat, das Objekt zu replizieren.

- `"FAILED"`— Zeigt an, dass Amazon S3 versucht hat, das Objekt zuvor zu replizieren, dies jedoch fehlgeschlagen ist.
- `"COMPLETED"` – Zeigt an, dass Amazon S3 das Objekt zuvor erfolgreich repliziert hat.
- `"REPLICA"`— Zeigt an, dass es sich um ein Replikat-Objekt handelt, das Amazon S3 aus einer anderen Quelle repliziert hat.

Weitere Hinweise zur Verwendung dieser Replikationsstatuswerte mit der Batch-Replikation finden Sie unterthe section called “Filter für einen Batch-Replikationsauftrag”.

Finden des Replikationsstatus

Verwenden Sie das Amazon-S3-Inventory-Tool, um den Replikationsstatus der Objekte in einem Bucket abzurufen. Amazon S3 sendet eine CSV Datei an den Ziel-Bucket, den Sie in der Inventarkonfiguration angeben. Sie können auch Amazon Athena verwenden, um den Replikationsstatus im Bestandsbericht abzufragen. Weitere Informationen zu Amazon S3 Inventory finden Sie unter Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory.

Sie können den Status der Objektreplikation auch mithilfe der Amazon S3 S3-Konsole, der AWS Command Line Interface (AWS CLI) oder der ermitteln AWS SDK.

Verwenden der S3-Konsole

In der Amazon S3 S3-Konsole können Sie den Replikationsstatus für ein Objekt auf der Detailseite des Objekts einsehen.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste der Buckets für allgemeine Zwecke den Namen des Replizierungsquellen-Buckets aus.
4. Wählen Sie in der Liste Objekte den Objektnamen aus. Die Seite mit den Objektdetails wird angezeigt.
5. Scrollen Sie auf der Registerkarte Eigenschaften nach unten zum Abschnitt Übersicht über die Objektverwaltung. Unter Verwaltungskonfigurationen finden Sie den Wert unter Replikationsstatus.

Unter Verwendung des AWS CLI

Verwenden Sie den `head-object` Befehl AWS Command Line Interface (AWS CLI), um Objektmetadaten abzurufen, wie im folgenden Beispiel gezeigt. Ersetzen Sie den *amzn-s3-demo-source-bucket1* durch den Namen Ihres Replizierungsquellen-Buckets und den anderen *user input placeholders* durch Ihre eigenen Informationen.

```
aws s3api head-object --bucket amzn-s3-demo-source-bucket1 --key object-key --version-id object-version-id
```

Der Befehl gibt Informationen zu Objektmetadaten, einschließlich des `ReplicationStatus`, zurück, wie in der folgenden Beispielantwort dargestellt:

```
{
   "AcceptRanges":"bytes",
   "ContentType":"image/jpeg",
   "LastModified":"Mon, 23 Mar 2015 21:02:29 GMT",
   "ContentLength":3191,
   "ReplicationStatus":"COMPLETED",
   "VersionId":"jfnW.HIMOfYiD_9rGbSkmroXsFj3fqZ.",
   "ETag":"\"6805f2cfc46c0f04559748bb039d69ae\"",
   "Metadata":{

   }
}
```

Verwenden Sie den AWS SDKs

Die folgenden Codefragmente ermitteln Ihren Replikationsstatus mithilfe von AWS SDK for Java und AWS SDK for .NET.

Java

```
GetObjectMetadataRequest metadataRequest = new GetObjectMetadataRequest(bucketName,
 key);
ObjectMetadata metadata = s3Client.getObjectMetadata(metadataRequest);

System.out.println("Replication Status : " +
 metadata.getRawMetadataValue(Headers.OBJECT_REPLICATION_STATUS));
```

.NET

```
GetObjectMetadataRequest getmetadataRequest = new GetObjectMetadataRequest
    {
         BucketName = sourceBucket,
         Key        = objectKey
    };

GetObjectMetadataResponse getmetadataResponse =
 client.GetObjectMetadata(getmetadataRequest);
Console.WriteLine("Object replication status: {0}",
 getmetadataResponse.ReplicationStatus);
```

Verwaltung von regionsübergreifendem Verkehr mit multiregionalen Access Points

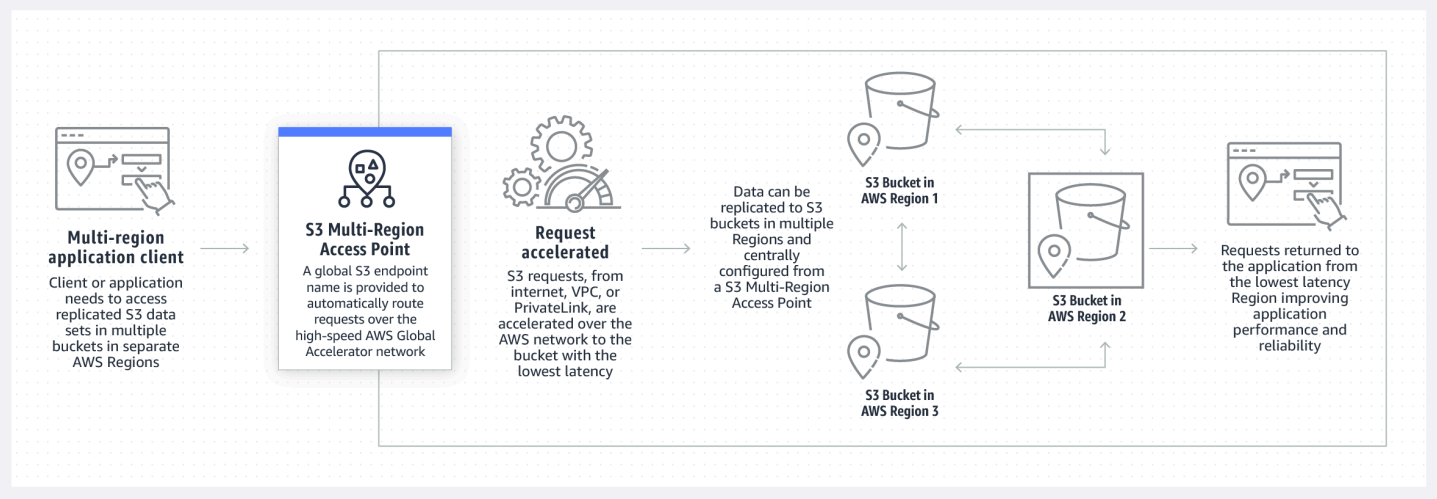
Amazon S3 Multiregion Access Points bieten einen globalen Endpunkt, über den Anwendungen Anfragen von S3-Buckets bearbeiten können, die sich in mehreren befinden. AWS-Regionen Sie können Multi-Regions Access Points verwenden, um Multi-Regions-Anwendungen mit derselben Architektur zu erstellen, die in einer einzelnen Region verwendet wird, und diese Anwendungen dann überall auf der Welt ausführen. Anstatt Anfragen über das überlastete öffentliche Internet zu senden, bieten Multi-Region Access Points eine integrierte Netzwerkstabilität und beschleunigen internetbasierte Anfragen an Amazon S3. Anwendungsanfragen, die an einen globalen Endpunkt mit multiregionalem Access Point gestellt werden AWS Global Accelerator, werden automatisch über das AWS globale Netzwerk zum nächstgelegenen S3-Bucket mit aktivem Routing-Status weitergeleitet.

Wenn es zu einer regionalen Verkehrsunterbrechung kommt, können Sie mithilfe der Failover-Steuerelemente für multiregionale Access Points den S3-Datenanforderungsverkehr zwischen den einzelnen Regionen verlagern AWS-Regionen und den S3-Verkehr innerhalb von Minuten von den Störungen wegleiten. Sie können auch die Widerstandsfähigkeit der Anwendung gegen eine Unterbrechung testen, um einen Anwendungs-Failover durchzuführen und Simulationen für die Notfallwiederherstellung durchzuführen. Wenn Sie Verbindungen zu S3 von außerhalb herstellen und Anfragen an S3 beschleunigen müssenVPC, können Sie Anwendungen und Netzwerkarchitekturen mit Amazon S3 Multiregion Access Points vereinfachen. Ihre Anfragen für multiregionale Access Points werden über das AWS globale Netzwerk und dann innerhalb von zurück an S3 weitergeleitet AWS-Region, ohne das öffentliche Internet nutzen zu müssen. Dadurch können Sie Anwendungen mit höherer Verfügbarkeit erstellen.

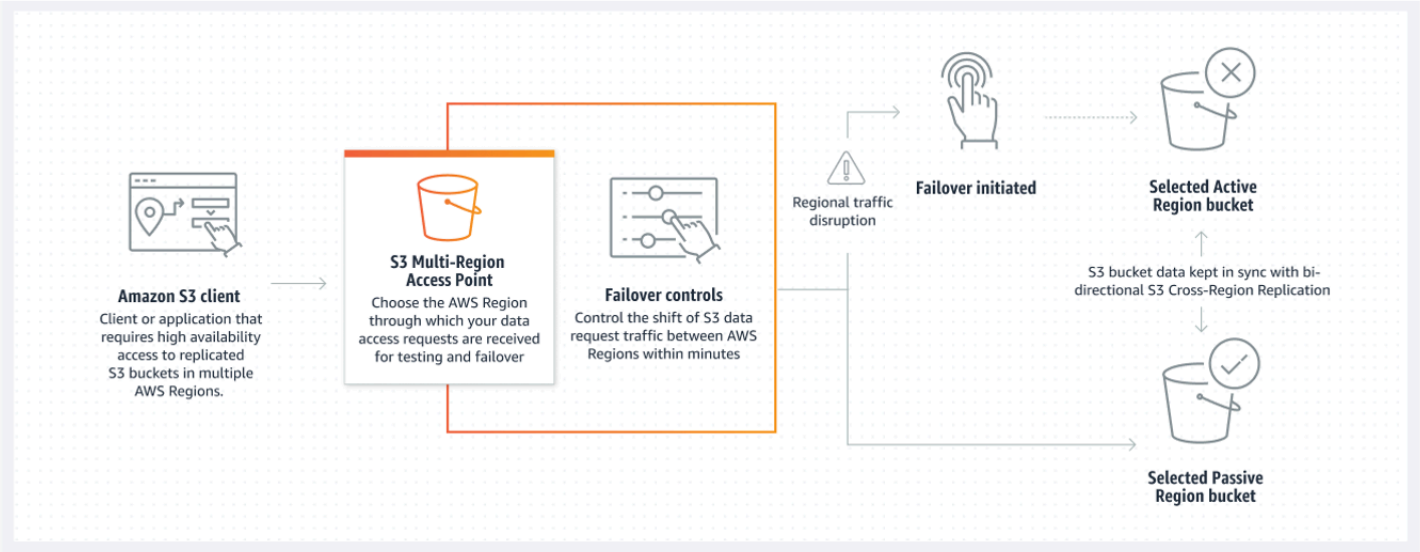
Bei der Erstellung und Einrichtung Ihrer Multi-Region-Access Points geben Sie an, AWS-Regionen wo Sie Daten speichern möchten, die über diesen multiregionalen Access Point bereitgestellt werden sollen. Sie können den angegebenen Endpunktnamen für multiregionale Access Points verwenden, um Ihre Clients zu verbinden. Nachdem Sie Ihre Client-Verbindungen eingerichtet haben, können Sie die vorhandenen oder neuen Buckets auswählen, zwischen denen Sie die Anfragen der Multi-Region Access Points weiterleiten möchten. Verwenden Sie dann die Regeln für die regionsübergreifende Replikation (CRR) von S3, um Daten zwischen Buckets in diesen Regionen zu synchronisieren.

Nachdem Sie Ihren Multi-Region Access Point eingerichtet haben, können Sie Daten über den globalen Endpunkt Multi-Region Access Points anfordern oder schreiben. Amazon S3 bearbeitet automatisch Anfragen an den replizierten Datensatz aus der nächstgelegenen verfügbaren Region. In der AWS Management Console können Sie auch die zugrunde liegende Replikationstopologie und die Replikationsmetriken für Ihre Access Point-Anfragen in mehreren Regionen einsehen. Auf diese Weise können Sie Speicher für Anwendungen mit mehreren Regionen noch einfacher erstellen, verwalten und überwachen. Alternativ können Sie Amazon verwenden, CloudFront um die Erstellung und Konfiguration von S3 Multiregion Access Points zu automatisieren.

Die folgende Abbildung ist eine grafische Darstellung eines Amazon S3 Multi-Region Access Points in einer Aktiv-Aktiv-Konfiguration. Die Grafik zeigt, wie Amazon-S3-Anforderungen automatisch an Buckets in der nächstgelegenen aktiven AWS-Region weitergeleitet werden.



Die folgende Abbildung ist eine grafische Darstellung eines Amazon S3 Multi-Region Access Points in einer Aktiv-Passiv-Konfiguration. Die Grafik zeigt, wie Sie den Amazon-S3-Datenzugriffsverkehr so steuern können, dass ein Failover zwischen aktiven und passiven AWS-Regionen erfolgt.



Weitere Informationen zur Verwendung von Multi-Region Access Points finden Sie unter Tutorial: Getting started with Amazon S3 Multi-Region Access Points.

Themen

- Erstellen Multi-Regions-Zugriffspunkten
- Konfigurieren eines Multi-Region Access Point zur Verwendung mit AWS PrivateLink
- Stellen von Anforderungen über einen Multi-Region Access Point

Erstellen Multi-Regions-Zugriffspunkten

Gehen Sie wie folgt vor, um einen Multi-Region Access Point in Amazon S3 zu erstellen:

- Geben Sie den Namen für den Multi-Region Access Point an.
- Wählen Sie in jedem Bucket AWS-Region , in dem Sie Anfragen für den Multi-Region Access Point bearbeiten möchten, aus.
- Konfigurieren Sie die Einstellungen für Amazon S3 Block Public Access für den Multi-Region Access Point.

Sie geben alle diese Informationen in einer Erstellungsanforderung an, die Amazon S3 asynchron verarbeitet. Amazon S3 bietet ein Token, mit dem Sie den Status der asynchronen Erstellungsanforderung überwachen können.

Beheben Sie Sicherheitswarnungen, Fehler, allgemeine Warnungen und Vorschläge von AWS Identity and Access Management Access Analyzer bevor Sie Ihre Richtlinie speichern. IAMAccess Analyzer führt Richtlinienprüfungen durch, um Ihre Richtlinie anhand der IAM Richtliniengrammatik und der bewährten Methoden zu überprüfen. Diese Prüfungen generieren Ergebnisse und bieten umsetzbare Empfehlungen, die Sie beim Erstellen von Richtlinien unterstützen, die funktionsfähig sind und den bewährten Methoden für Sicherheit entsprechen. Weitere Informationen zur Überprüfung von Richtlinien mit IAM Access Analyzer finden Sie unter IAMAccess Analyzer-Richtlinienvalidierung im IAMBenutzerhandbuch. Eine Liste der Warnungen, Fehler und Vorschläge, die von Access Analyzer zurückgegeben werden, finden Sie unter IAM IAMAccess Analyzer-Referenz zur Richtlinienüberprüfung.

Wenn Sie den verwendenAPI, erfolgt die Anforderung zum Erstellen eines Access Points mit mehreren Regionen asynchron. Wenn Sie eine Anforderung zum Erstellen eines Multi-Regions-Zugriffpunkts übermitteln, autorisiert Amazon S3 die Anforderung synchron. Es gibt dann sofort ein Token zurück, mit dem Sie den Fortschritt der Erstellungsanforderung verfolgen können. Weitere Informationen zum Nachverfolgen asynchroner Anforderungen zum Erstellen und Verwalten von Multi-Regions-Zugriffpunkten finden Sie unter Verwenden von Access Points mit mehreren Regionen mit unterstützten Vorgängen API.

Nachdem Sie den Multi-Regions-Zugriffpunkt erstellt haben, können Sie eine Zugriffssteuerungsrichtlinie für diesen erstellen. Jedem Multi-Regions-Zugriffpunkt kann eine Richtlinie zugeordnet sein. Eine Richtlinie für Multi-Region Access Points ist eine ressourcenbasierte Richtlinie, mit der Sie die Verwendung des Multi-Region Access Point nach Ressource, Benutzer oder anderen Bedingungen einschränken können.

> **Note**
>
> Damit eine Anwendung oder ein Benutzer über einen Multi-Regions Access Point auf ein Objekt zugreifen kann, müssen die folgenden Richtlinien beide die Anforderung zulassen:
>
> - Die Zugriffsrichtlinie für den Multi-Region Access Point
> - Die Zugriffsrichtlinie für den zugrunde liegenden Bucket, der das Objekt enthält
>
> Wenn die beiden Richtlinien unterschiedlich sind, hat die restriktivere Richtlinie Vorrang. Um die Verwaltung von Berechtigungen für Multi-Region Access Points zu vereinfachen, können Sie die Zugriffssteuerung vom Bucket an den Multi-Region Access Point delegieren.

Weitere Informationen finden Sie unter the section called “Beispielrichtlinien für Multi-Region Access Points”.

Die Verwendung eines Buckets mit einem Multi-Region-Access Point ändert nichts am Verhalten des Buckets, wenn auf den Bucket über den vorhandenen Bucket-Namen oder einen Amazon-Ressourcennamen (ARN) zugegriffen wird. Alle vorhandenen Vorgänge für den Bucket funktionieren weiterhin wie zuvor. Einschränkungen, die Sie in eine Multi-Regions-Zugriffspunktrichtlinie einschließen, gelten nur für Anforderungen, die über den Multi-Regions-Zugriffspunkt eingehen.

Sie können die Richtlinie für einen Multi-Regions-Zugriffpunkt nach dem Erstellen aktualisieren, die Richtlinie jedoch nicht löschen. Sie können jedoch die Richtlinie für Multi-Region Access Points aktualisieren, um alle Berechtigungen zu verweigern.

Themen

- Regeln zur Benennung von Amazon S3-Multi-Regions-Zugriffspunkten
- Regeln für die Auswahl von Buckets für Amazon S3-Multi-Regions-Zugriffspunkte
- Erstellen eines Amazon S3-Multi-Region Access Point
- Blockieren des öffentlichen Zugriffs mit Amazon S3-Multi-Regions-Zugriffspunkten
- Anzeigen der Konfigurationsdetails für Amazon S3 Multi-Region Access Points
- Löschen eines Multi-Region Access Point

Regeln zur Benennung von Amazon S3-Multi-Regions-Zugriffspunkten

Wenn Sie einen Multi-Regions-Zugriffpunkt erstellen, geben Sie ihm einen Namen, bei dem es sich um eine von Ihnen ausgewählte Zeichenfolge handelt. Sie können den Namen des Multi-Regions-Zugriffspunkts nach der Erstellung nicht mehr ändern. Der Name muss in Ihrem AWS-Konto einzigartig sein, und es muss den Namensanforderungen entsprechen, die in Einschränkungen und Beschränkungen des Multi-Regions-Zugriffspunkts aufgelistet sind. Wenn Sie den Multi-Regions-Zugriffpunkt identifizieren möchten, verwenden Sie einen Namen, der für Sie oder Ihre Organisation aussagekräftig ist oder der das Szenario widerspiegelt.

Sie verwenden diesen Namen beim Aufrufen von Sie verwenden diesen Namen, wenn Sie Verwaltungsvorgänge für Multi-Regions-Zugriffpunkten, z. B. `GetMultiRegionAccessPoint` und `PutMultiRegionAccessPointPolicy`. Der Name wird nicht verwendet, um Anforderungen an den Multi-Region Access Point zu senden, und er muss Clients, die Anforderungen unter Verwendung des Multi-Region Access Point stellen, nicht offengelegt werden.

Wenn Amazon S3 einen Multi-Regions-Zugriffpunkt erstellt, weist er ihm automatisch einen Alias zu. Dieser Alias ist eine eindeutige alphanumerische Zeichenfolge, die mit `.mrap` endet. Der Alias wird verwendet, um den Hostnamen und den Amazon-Ressourcennamen (ARN) für einen Access Point mit mehreren Regionen zu erstellen. Der vollqualifizierte Name basiert auch auf dem Alias für den Multi-Regions-Zugriffpunkt.

Sie können den Namen eines Multi-Regions-Zugriffspunkts nicht anhand seines Alias ermitteln, sodass Sie einen Alias offenlegen können, ohne den Namen, den Zweck oder den Besitzer des Multi-Regions-Zugriffspunkts zu enthüllen. Amazon S3 wählt den Alias für jeden neuen Multi-Regions-Zugriffspunkt aus, und der Alias kann nicht geändert werden. Weitere Informationen zur Adressierung eines Multi-Regions-Zugriffspunkts finden Sie unter Stellen von Anforderungen über einen Multi-Region Access Point.

Multi-Regions-Zugriffspunkt-Aliase sind im Laufe der Zeit eindeutig und basieren nicht auf dem Namen oder der Konfiguration eines Multi-Regions-Zugriffspunkts. Wenn Sie einen Multi-Regions-Zugriffspunkt erstellen, ihn dann löschen und einen anderen mit demselben Namen und derselben Konfiguration erstellen, hat der zweite Multi-Regions-Zugriffspunkt einen anderen Alias als der erste. Neue Multi-Regions-Zugriffspunkte können nie denselben Alias wie ein vorheriger Multi-Regions-Zugriffspunkt haben.

Regeln für die Auswahl von Buckets für Amazon S3-Multi-Regions-Zugriffspunkte

Jeder Multi-Regions-Zugriffspunkt ist den Regionen zugeordnet, in denen Sie Anforderungen erfüllen möchten. Der Multi-Regions-Zugriffspunkt muss genau einem Bucket in jeder dieser Regionen zugeordnet sein. Sie geben den Namen jedes Buckets in der Anforderung an, um den Multi-Regions-Zugriffspunkts zu erstellen. Buckets, die den Multi-Region-Access Point unterstützen, können sich entweder in demselben Bereich befinden AWS-Konto , dem der Multi-Region-Access Point gehört, oder sie können sich in einem anderen befinden. AWS-Konten

Ein einzelner Bucket kann von mehreren Multi-Regions-Zugriffspunkten verwendet werden.

> **Important**
>
> - Sie können die Buckets, die einem Multi-Regions-Zugriffspunkt zugeordnet sind, nur zum Zeitpunkt seiner Erstellung angeben. Nach der Erstellung können Sie keine Buckets hinzufügen, ändern oder aus der Konfiguration des Multi-Regions-Zugriffspunkts entfernen. Um die Buckets zu ändern, müssen Sie den gesamten Multi-Regions-Zugriffspunkt löschen und einen neuen erstellen.

- Sie können einen Bucket, der Teil eines Multi-Regions-Zugriffspunkts ist, nicht löschen. Wenn Sie einen Bucket löschen möchten, der einem Multi-Region Access Point zugeordnet ist, löschen Sie zuerst den Multi-Region Access Point.
- Wenn Sie Ihrem Multi-Region Access Point einen Bucket hinzufügen, dessen Eigentümer ein anderes Konto ist, muss der Bucket-Eigentümer ebenfalls seine Bucket-Richtlinie aktualisieren, um Zugriffsberechtigungen für den Multi-Region Access Point zu gewähren. Andernfalls kann der Multi-Region Access Point keine Daten aus diesem Bucket abrufen. Beispielrichtlinien, die zeigen, wie ein solcher Zugriff gewährt wird, finden Sie unter Beispielrichtlinien für Multi-Region Access Points.
- Nicht alle Regionen unterstützen Multi-Regions-Zugriffspunkte. Eine Liste der unterstützten Regionen finden Sie unter Einschränkungen und Beschränkungen des Multi-Regions-Zugriffspunkts.

Sie können Replikationsregeln erstellen, um Daten zwischen Buckets zu synchronisieren. Mit diesen Regeln können Sie Daten automatisch aus Quell-Buckets in Ziel-Buckets kopieren. Die Verbindung von Buckets mit einem Multi-Regions-Zugriffspunkt hat keinen Einfluss auf die Funktionsweise der Replikation. Das Konfigurieren der Replikation mit Multi-Regions-Zugriffspunkten wird in einem späteren Abschnitt beschrieben.

Important

Wenn Sie eine Anforderung an einen Multi-Region Access Point stellen, kennt der Multi-Region Access Point die Dateninhalte der Buckets in dem Multi-Region Access Point nicht. Daher kann es sein, dass der Bucket, der die Anforderung erhält, die angeforderten Daten nicht enthält. Um konsistente Datensätze in den Amazon S3 S3-Buckets zu erstellen, die einem Access Point mit mehreren Regionen zugeordnet sind, empfehlen wir, die regionsübergreifende S3-Replikation () zu konfigurieren. CRR Weitere Informationen finden Sie unter Konfigurieren einer Replikation zur Verwendung mit Multi-Region Access Points.

Erstellen eines Amazon S3-Multi-Region Access Point

Das folgende Beispiel zeigt, wie ein Multi-Region Access Point über die Amazon-S3-Konsole erstellt werden kann.

Verwenden der S3-Konsole

So erstellen Sie einen Multi-Regions-Zugriffspunkt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
3. Wählen Sie Multi-Region Access Points erstellen aus, um mit der Erstellung Ihres Multi-Region Access Point zu beginnen.
4. Geben Sie auf der Seite Multi-Region Access Point einen Namen für den Multi-Region Access Point in das Feld Name des Multi-Region Access Point ein.
5. Wählen Sie die Buckets aus, die diesem Multi-Region Access Point zugeordnet werden sollen. Sie können Buckets auswählen, die sich in Ihrem Konto befinden, oder Sie können Buckets aus anderen Konten auswählen.

   > **Note**
   >
   > Sie müssen mindestens einen Bucket aus Ihrem Konto oder anderen Konten hinzufügen. Beachten Sie außerdem, dass Multi-Region-Access Points nur einen Bucket pro AWS-Region unterstützen. Daher können Sie nicht zwei Buckets aus derselben Region hinzufügen. AWS-Regionen , die standardmäßig deaktiviert sind, werden nicht unterstützt.

   - Um einen Bucket hinzuzufügen, der sich in Ihrem Konto befindet, wählen Sie Hinzufügen von Buckets aus. Es wird eine Liste aller Buckets in Ihrem Konto angezeigt. Sie können nach Ihrem Bucket anhand des Namens suchen oder die Bucket-Namen in alphabetischer Reihenfolge sortieren.
   - Um einen Bucket aus einem anderen Konto hinzuzufügen, wählen Sie Bucket aus einem anderen Konto hinzufügen aus. Stellen Sie sicher, dass Sie den genauen Namen und die AWS-Konto ID des Buckets kennen, da Sie in anderen Konten nicht nach Buckets suchen oder danach suchen können.

     > **Note**
     >
     > Sie müssen eine gültige AWS-Konto ID und einen gültigen Bucket-Namen eingeben. Der Bucket muss sich auch in einer unterstützten Region befinden, andernfalls tritt ein

Fehler auf, wenn Sie versuchen, Ihren Multi-Region Access Point zu erstellen. Eine Liste der Regionen, die Multi-Region Access Points unterstützen, finden Sie unter Einschränkungen und Beschränkungen von Multi-Region Access Points.

6. (Optional) Wenn Sie einen hinzugefügten Bucket entfernen müssen, wählen Sie Entfernen aus.

> **Note**
>
> Nach dem Erstellen des Multi-Region Access Point können Sie diesem keine Buckets hinzufügen und keine Buckets daraus entfernen.

7. Wählen Sie unter Block Public Access settings for this Multi-Region Access Point (Einstellungen für die Sperrung des öffentlichen Zugriffs für diesen Multi-Regions-Zugriffspunkt) die Einstellungen für die Sperrung des öffentlichen Zugriffs, die Sie für den Multi-Regions-Zugriffspunkt anwenden möchten. Alle Block-Einstellungen des öffentlichen Zugriffs sind standardmäßig für neue Multi-Regions-Zugriffspunkte aktiviert. Es wird empfohlen, alle Einstellungen aktiviert zu lassen, es sei denn, Sie wissen, dass Sie eine bestimmte Notwendigkeit haben, eine von ihnen zu deaktivieren.

> **Note**
>
> Nach dem Erstellen eines Multi-Region Access Point können Sie die Block-Public-Access-Einstellungen für den Multi-Region Access Point nicht mehr ändern. Wenn Sie den öffentlichen Zugriff blockieren möchten, stellen Sie daher sicher, dass Ihre Anwendungen ohne öffentlichen Zugriff ordnungsgemäß funktionieren, bevor Sie einen Multi-Region Access Point erstellen.

8. Wählen Sie Multi-Regions-Zugriffspunkt erstellen.

> **Important**
>
> Wenn Sie Ihrem Multi-Region Access Point einen Bucket hinzufügen, dessen Eigentümer ein anderes Konto ist, muss der Bucket-Eigentümer auch seine Bucket-Richtlinie aktualisieren, um Zugriffsberechtigungen für den Multi-Region Access Point zu gewähren. Andernfalls kann der Multi-Region Access Point keine Daten aus diesem Bucket abrufen. Beispielrichtlinien,

die zeigen, wie ein solcher Zugriff gewährt wird, finden Sie unter Beispielrichtlinien für Multi-Region Access Points.

Verwenden Sie den AWS CLI

Sie können den verwenden AWS CLI , um einen Access Point für mehrere Regionen zu erstellen. Beim Erstellen des Multi-Region Access Point müssen Sie alle Buckets bereitstellen, die dieser unterstützen soll. Sie können dem Multi-Region Access Point nach der Erstellung keine Buckets hinzuzufügen.

Im folgenden Beispiel wird ein Multi-Region Access Point mit zwei Buckets über die AWS CLI erstellt. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws s3control create-multi-region-access-point --account-id 111122223333 --details '{
        "Name": "simple-multiregionaccesspoint-with-two-regions",
        "PublicAccessBlock": {
            "BlockPublicAcls": true,
            "IgnorePublicAcls": true,
            "BlockPublicPolicy": true,
            "RestrictPublicBuckets": true
        },
        "Regions": [
            { "Bucket": "amzn-s3-demo-bucket1" },
            { "Bucket": "amzn-s3-demo-bucket2" }
        ]
    }' --region us-west-2
```

Blockieren des öffentlichen Zugriffs mit Amazon S3-Multi-Regions-Zugriffspunkten

Jeder Multi-Regions-Zugriffspunkt hat verschiedene Block-Einstellungen des öffentlichen Zugriffs in Amazon S3. Diese Einstellungen funktionieren zusammen mit den Einstellungen für den Block Public Access für denjenigen AWS-Konto , der den Multi-Region Access Point und die zugrunde liegenden Buckets besitzt.

Wenn Amazon S3 eine Anforderung autorisiert, wird die restriktivste Kombination dieser Einstellungen angewendet. Wenn die Block-Public-Access-Einstellungen für eine dieser Ressourcen (das Konto, das Eigentümer des Multi-Region Access Point ist, der zugrunde liegende Bucket oder das Bucket-Eigentümerkonto) den Zugriff für die angeforderte Aktion oder Ressource blockieren, lehnt Amazon S3 die Anforderung ab.

Alle Block-Einstellungen des öffentlichen Zugriffs sind standardmäßig für neue Zugriffspunkte aktiviert, und wir empfehlen Ihnen, alle Einstellungen aktiviert zu lassen, es sei denn, Sie wissen, dass Sie einen bestimmten Wert deaktivieren müssen. Alle Block-Einstellungen des öffentlichen Zugriffs sind standardmäßig für Multi-Regions-Zugriffspunkte aktiviert. Wenn Block Public Access aktiviert ist, kann der Multi-Region Access Point keine internetbasierten Anforderungen annehmen.

> **Important**
>
> Nach dem Erstellen eines Multi-Region Access Point können Sie die Block-Public-Access-Einstellungen für den Access Point nicht mehr ändern.

Weitere Informationen über Block Public Access in Amazon S3 finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

Anzeigen der Konfigurationsdetails für Amazon S3 Multi-Region Access Points

Das folgende Beispiel zeigt, wie Konfigurationsdetails für einen Multi-Region Access Point über die Amazon-S3-Konsole angezeigt werden können.

Verwenden der S3-Konsole

So erstellen Sie einen Multi-Regions-Zugriffspunkt

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
3. Wählen Sie den Namen des Multi-Region Access Point aus, dessen Konfigurationsdetails Sie anzeigen möchten.

   - Auf der Registerkarte Eigenschaften werden alle Buckets aufgeführt, die mit Ihrem Access Point in mehreren Regionen verknüpft sind, das Erstellungsdatum, der Amazon-Ressourcenname (ARN) und der Alias. In der Spalte für die AWS-Konto -ID sind auch alle Buckets aufgeführt, deren Eigentümer externe Konten sind und die Ihrem Multi-Region-Access Point zugeordnet sind.
   - Auf der Registerkarte Berechtigungen sind die Block-Public-Access-Einstellungen aufgeführt, die für die diesem Multi-Regions-Zugriffspunkt zugeordneten Buckets gelten. Sie können sich die Richtlinie für Multi-Region Access Points für Ihren Multi-Region Access Point auch

ansehen, falls Sie eine erstellt haben. In der Info-Warnung auf der Seite Berechtigungen sind außerdem alle Buckets (in Ihrem Konto und anderen Konten) für diesen Multi-Region Access Point aufgeführt, für die die Einstellung Öffentlicher Zugriff ist blockiert aktiviert ist.

- Die Registerkarte Replikation und Failover bietet eine Kartenansicht der Buckets, die Ihrem Multi-Region Access Point zugeordnet sind, sowie der Regionen, in denen sich die Buckets befinden. Wenn es Buckets von einem anderen Konto gibt, für das Sie keine Berechtigung zum Abrufen von Daten haben, ist die Region in der Replikationsübersicht rot markiert, um darauf hinzuweisen, dass es sich um eine AWS-Region handelt, bei der Fehler beim Abrufen des Replikationsstatus aufgetreten sind.

  > **Note**
  >
  > Um Informationen zum Replikationsstatus aus einem Bucket in einem externen Konto abzurufen, muss Ihnen der Bucket-Eigentümer in seiner Bucket-Richtlinie die Berechtigung `s3:GetBucketReplication` erteilen.

  Auf dieser Registerkarte sind auch die Replikationsmetriken, Replikationsregeln und Failover-Status für die Regionen aufgeführt, die mit Ihrem Multi-Region Access Point verwendet werden.

Mit dem AWS CLI

Sie können den verwenden AWS CLI , um die Konfigurationsdetails für einen Access Point mit mehreren Regionen anzuzeigen.

Im folgenden AWS CLI Beispiel wird Ihre aktuelle Multi-Region-Access Point-Konfiguration abgerufen. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws s3control get-multi-region-access-point --account-id 111122223333 --name amzn-s3-demo-bucket
```

Löschen eines Multi-Region Access Point

Das folgende Verfahren zeigt, wie ein Multi-Region Access Point über die Amazon-S3-Konsole gelöscht werden kann.

Beim Löschen eines Multi-Region Access Point werden die dem Multi-Region Access Point zugeordneten Buckets nicht gelöscht, nur der Multi-Region Access Point selbst.

Verwenden der S3-Konsole

So löschen Sie einen Multi-Region Access Point

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
3. Wählen Sie das Optionsfeld neben dem Namen Ihres Multi-Region Access Point aus.
4. Wählen Sie Löschen.
5. Geben Sie im Dialogfeld „Multi-Region Access Point löschen“ den Namen des AWS Buckets ein, den Sie löschen möchten.

   > **Note**
   >
   > Stellen Sie sicher, dass Sie einen gültigen Bucket-Namen eingeben. Andernfalls wird die Schaltfläche Löschen deaktiviert.

6. Wählen Sie Löschen aus, um zu bestätigen, dass Sie Ihren Multi-Region Access Point tatsächlich löschen möchten.

Verwenden Sie den AWS CLI

Sie können den verwenden AWS CLI , um einen Access Point mit mehreren Regionen zu löschen. Durch diese Aktion werden die dem Multi-Region Access Point zugeordneten Buckets nicht gelöscht, nur der Multi-Region Access Point selbst. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control delete-multi-region-access-point --account-id 123456789012 --details
 Name=example-multi-region-access-point-name
```

Konfigurieren eines Multi-Region Access Point zur Verwendung mit AWS PrivateLink

Sie können Multi-Region Access Points verwenden, um Amazon-S3-Anforderungsdatenverkehr zwischen AWS-Regionen weiterzuleiten. Jeder globale Endpunkt von Multi-Region Access Points

leitet den Amazon-S3-Datenanforderungsverkehr aus mehreren Quellen weiter, ohne dass Sie komplexe Netzwerkkonfigurationen mit separaten Endpunkten erstellen müssen. Zu diesen Quellen des Datenanforderungsverkehrs gehören:

- Datenverkehr aus einer Virtual Private Cloud (VPC)
- Datenverkehr aus On-Premises-Rechenzentren, der über AWS PrivateLink läuft
- Datenverkehr aus dem öffentlichen Internet

Wenn Sie eine AWS PrivateLink-Verbindung zu einem S3 Multi-Region Access Point herstellen, können Sie S3-Anforderungen in AWS oder zwischen mehreren AWS-Regionen über eine private Verbindung weiterleiten, indem Sie eine einfache Netzwerkarchitektur und -konfiguration verwenden. Wenn Sie AWS PrivateLink verwenden, müssen Sie keine VPC-Peering-Verbindung konfigurieren.

Themen

- Konfigurieren eines Multi-Region Access Point zur Verwendung mit AWS PrivateLink
- Entfernen des Zugriffs auf einen Multi-Region Access Point von einem VPC-Endpunkt

Konfigurieren eines Multi-Region Access Point zur Verwendung mit AWS PrivateLink

AWS PrivateLink bietet Ihnen private Konnektivität zu Amazon S3 mithilfe von privaten IP-Adressen in Ihrer virtuellen privaten Cloud (VPC). Sie können einen oder mehrere Schnittstellenendpunkte in Ihrem Gerät bereitstellen, um eine Verbindung VPC zu Amazon S3 Multi-Region Access Points herzustellen.

Sie können com.amazonaws.s3-global.accesspoint-Endpunkte für Access Points mit mehreren Regionen über, oder erstellen. AWS Management Console AWS CLI AWS SDKs Weitere Informationen zur Konfiguration eines Schnittstellenendpunkts für Multi-Region-Access Points finden Sie unter Schnittstellenendpunkte im Benutzerhandbuch. VPC VPC

Gehen Sie wie folgt vor, um über Schnittstellenendpunkte Anfragen an einen Multi-Region-Access Point zu richten, um den VPC und den Multi-Region-Access Point zu konfigurieren.

So konfigurieren Sie einen Access Point mit mehreren Regionen zur Verwendung mit AWS PrivateLink

1. Erstellen Sie einen geeigneten VPC Endpunkt, der eine Verbindung zu Access Points mit mehreren Regionen herstellen kann, oder verfügen Sie über einen entsprechenden

Endpunkt. Weitere Informationen zum Erstellen von VPC Endpunkten finden Sie unter VPCSchnittstellenendpunkte im VPC Benutzerhandbuch.

> **Important**
>
> Stellen Sie sicher, dass Sie einen com.amazonaws.s3-global.accesspoint-Endpunkt erstellen. Andere Endpunkttypen können nicht auf Multi-RegionsZugriffspunkte zugreifen.

Nachdem dieser VPC Endpunkt erstellt wurde, werden alle Multi-Region-Access Point-Anfragen auf dem Endpunkt über diesen Endpunkt VPC weitergeleitet, sofern Sie für den Endpunkt private Zugriffe DNS aktiviert haben. Dies ist standardmäßig aktiviert.

2. Wenn die Richtlinie für multiregionale Access Points keine Verbindungen von VPC Endpunkten unterstützt, müssen Sie sie aktualisieren.
3. Stellen Sie sicher, dass die einzelnen Bucket-Richtlinien den Zugriff auf die Benutzer des Multi-Regions-Zugriffspunkts ermöglichen.

Denken Sie daran, dass Multi-Regions-Zugriffspunkte funktionieren, indem sie Anforderungen an Buckets weiterleiten, nicht indem sie selbst Anforderungen erfüllen. Dies ist wichtig, da der Absender der Anforderung über Berechtigungen für den Multi-Regions-Zugriffspunkt verfügen muss und auf die einzelnen Buckets im Multi-Regions-Zugriffspunkt zugreifen kann. Andernfalls wird die Anforderung möglicherweise an einen Bucket weitergeleitet, in dem der Originator keine Berechtigungen hat, um die Anforderung zu erfüllen. Ein Access Point mit mehreren Regionen und die zugehörigen Buckets können demselben oder einem anderen Konto gehören. AWS Ein Multi-Region-Access Point kann jedoch VPCs von unterschiedlichen Konten aus verwendet werden, sofern die Berechtigungen korrekt konfiguriert sind.

Aus diesem Grund muss die VPC Endpunktrichtlinie den Zugriff sowohl auf den Multi-Region-Zugriffspunkt als auch auf alle zugrunde liegenden Buckets ermöglichen, die Sie in der Lage sein sollen, Anfragen zu bearbeiten. Angenommen, Sie haben einen Multi-Region Access Point mit dem Alias `mfzwi23gnjvgw.mrap`. Es wird von Buckets unterstützt `amzn-s3-demo-bucket1` und `amzn-s3-demo-bucket2` alle gehören einem AWS Account. `123456789012` In diesem Fall würde die folgende VPC Endpunktrichtlinie es ermöglichen, dass `GetObject` Anfragen von den VPC gestellten Benutzern von einem der beiden Backing-Buckets erfüllt werden. `mfzwi23gnjvgw.mrap`

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
    {
        "Sid": "Read-buckets-and-MRAP-VPCE-policy",
        "Principal": "*",
        "Action": [
            "s3:GetObject"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket1/*",
            "arn:aws:s3:::amzn-s3-demo-bucket2/*",
            "arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap/object/*"
        ]
    }]
}
```

Wie bereits erwähnt, müssen Sie außerdem sicherstellen, dass die Richtlinie für Access Points für mehrere Regionen so konfiguriert ist, dass sie den Zugriff über einen VPC Endpunkt unterstützt. Sie müssen den VPC Endpunkt, der den Zugriff anfordert, nicht angeben. Die folgende Beispielrichtlinie würde jedem Anforderer Zugriff gewähren, der versucht, den Multi-Region Access Point für die `GetObject`-Anforderungen zu nutzen.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Sid": "Open-read-MRAP-policy",
        "Effect": "Allow",
        "Principal": "*",
        "Action": [
            "s3:GetObject"
          ],
        "Resource": "arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap/object/*"
    }]
}
```

Und natürlich benötigen die einzelnen Buckets jeweils eine Richtlinie zur Unterstützung des Zugriffs auf Anfragen, die über den VPC Endpunkt eingereicht wurden. Die folgende Beispielrichtlinie gewährt allen anonymen Benutzern Lesezugriff, was auch Anfragen einschließt, die über den VPC Endpunkt gestellt wurden.

```
{
```

```
    "Version":"2012-10-17",
   "Statement": [
   {
       "Sid": "Public-read",
       "Effect":"Allow",
       "Principal": "*",
       "Action": "s3:GetObject",
       "Resource": [
           "arn:aws:s3:::amzn-s3-demo-bucket1",
           "arn:aws:s3:::amzn-s3-demo-bucket2/*"]
    }]
}
```

Weitere Informationen zum Bearbeiten einer VPC Endpunktrichtlinie finden Sie unter Steuern des Zugriffs auf Dienste mit VPC Endpunkten im VPCBenutzerhandbuch.

Entfernen des Zugriffs auf einen Multi-Region Access Point von einem VPC-Endpunkt

Wenn Sie einen Multi-Region Access Point besitzen und den Zugriff darauf von einem Schnittstellenendpunkt entfernen möchten, müssen Sie eine neue Zugriffsrichtlinie für den Multi-Region Access Point bereitstellen, die den Zugriff für Anforderungen über VPC-Endpunkte verhindert. Wenn die Buckets in Ihrem Multi-Region Access Point Anforderungen über VPC-Endpunkte unterstützen, werden sie diese Anforderungen allerdings weiterhin unterstützen. Wenn Sie diesen Support verhindern möchten, müssen Sie auch die Richtlinien für die Buckets aktualisieren. Durch die Bereitstellung einer neuen Zugriffsrichtlinie für den Multi-Region Access Point wird nur der Zugriff auf den Multi-Region Access Point verhindert, nicht auf die zugrunde liegenden Buckets.

> **Note**
>
> Sie können eine Zugriffsrichtlinie für einen Multi-Regions-Zugriffspunkt nicht löschen. Um den Zugriff auf einen Multi-Regions-Zugriffspunkt zu entfernen, müssen Sie eine neue Zugriffsrichtlinie mit dem gewünschten geänderten Zugriff bereitstellen.

Anstatt die Zugriffsrichtlinie für den Multi-Region Access Point zu aktualisieren, können Sie die Bucket-Richtlinien aktualisieren, um Anforderungen über VPC-Endpunkte zu verhindern. In diesem Fall können Benutzer weiterhin über den VPC-Endpunkt auf den Multi-Region Access Point zugreifen. Wenn der Multi-Region Access Point jedoch an einen Bucket weitergeleitet wird, in dem die Bucket-Richtlinie den Zugriff verhindert, wird eine Fehlermeldung generiert.

Stellen von Anforderungen über einen Multi-Region Access Point

Wie andere Ressourcen verfügen auch Amazon S3 Multi-Region Access Points über Amazon-Ressourcennamen (ARNs). Sie können diese ARNs verwenden, um Anforderungen über die AWS Command Line Interface (AWS CLI), AWS-SDKs oder die Amazon-S3-API an Multi-Region Access Points zu leiten. Sie können diese ARNs auch verwenden, um Multi-Region Access Points in Zugriffskontrollrichtlinien zu identifizieren. Der ARN eines Multi-Region Access Point beinhaltet den Namen des Multi-Region Access Points nicht und gibt diesen nicht an. Weitere Informationen zur Verwendung von ARNs finden Sie unter Amazon-Ressourcennamen (ARN) im Allgemeine AWS-Referenz.

> **Note**
>
> Der Alias des Multi-Region Access Point und der ARN können nicht gegeneinander ausgetauscht werden.

ARNs von Multi-Region Access Points verwenden das folgende Format:

`arn:aws:s3::`*`account-id`*`:accesspoint/`*`MultiRegionAccessPoint_alias`*

Im Folgenden finden Sie einige Beispiele für ARNs von Multi-Region Access Points:

- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap` repräsentiert den Multi-Region Access Point mit dem Alias `mfzwi23gnjvgw.mrap`, der im Besitz von AWS-Konto `123456789012` ist.
- `arn:aws:s3::123456789012:accesspoint/*` repräsentiert alle Multi-Region Access Points im Konto `123456789012`. Dieser ARN stimmt mit allen Multi-Region Access Points für das Konto `123456789012` überein, entspricht aber keinen regionalen Amazon S3 Access Points, da der ARN keine AWS-Region enthält. Im Gegensatz dazu stimmt der ARN `arn:aws:s3:us-west-2:123456789012:accesspoint/*` mit allen regionalen Amazon S3 Access Points in der Region `us-west-2` für das Konto `123456789012` überein, entspricht aber keinem Multi-Region Access Point.

ARNs für Objekte, auf die über einen Multi-Region Access Point zugegriffen wird, verwenden folgendes Format:

`arn:aws:s3::`*`account_id`*`:accesspoint/`*`MultiRegionAccessPoint_alias`*`//`*`key`*

Wie bei ARNs von Multi-Region Access Points enthalten die ARNs für Objekte, auf die über Multi-Region Access Points zugegriffen wird, keine AWS-Region. Hier sind einige Beispiele.

- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap//-01` repräsentiert `-01`, auf das über den Multi-Region Access Point mit dem Alias `mfzwi23gnjvgw.mrap`, im Besitz von Konto `123456789012`, zugegriffen wird.
- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap//*` repräsentiert alle Objekte, auf die über den Multi-Region Access Point mit dem Alias `mfzwi23gnjvgw.mrap`, in Konto `123456789012`, zugegriffen werden kann.
- `arn:aws:s3::123456789012:accesspoint/mfzwi23gnjvgw.mrap//-01/finance/*` repräsentiert alle Objekte, auf die unter `-01/finance/` für den Multi-Region Access Point mit dem Alias `mfzwi23gnjvgw.mrap`, in Konto `123456789012`, zugegriffen werden kann.

Hostnamen für Multi-Regions-Zugriffspunkte

Sie können über einen Multi-Region Access Point auf Daten in Amazon S3 zugreifen, indem Sie den Hostnamen des Multi-Region Access Points verwenden. Anforderungen können aus dem öffentlichen Internet an diesen Hostnamen gerichtet werden. Wenn Sie ein oder mehrere Internet-Gateways für den Multi-Region Access Point konfiguriert haben, können Anforderungen auch aus einer Virtual Private Cloud (VPC) an diesen Hostnamen weitergeleitet werden. Weitere Informationen zum Erstellen von VPC-Schnittstellenendpunkten zur Verwendung mit mehreren Multi-Regions-Zugriffspunkten finden Sie unter Konfigurieren eines Multi-Region Access Point zur Verwendung mit AWS PrivateLink.

Wenn Sie Anforderungen über einen Multi-Region Access Point von einer VPC unter Verwendung eines VPC-Endpunkts stellen möchten, können Sie AWS PrivateLink verwenden. Wenn Sie Anforderungen an einen Multi-Region Access Point mithilfe von AWS PrivateLink stellen, können Sie den Namen eines endpunktspezifischen regionalen DOMAIN NAME SYSTEM (DNS), der mit *`region`*`.vpce.amazonaws.com` endet, nicht direkt verwenden. Diesem Hostnamen wird kein Zertifikat zugeordnet, so dass er nicht direkt verwendet werden kann. Sie können den DOMAIN NAME SYSTEM (DNS)-Namen des VPC-Endpunkts weiterhin als `CNAME`- oder `ALIAS`-Ziel verwenden. Alternativ können Sie privates DOMAIN NAME SYSTEM (DNS) auf dem Endpunkt aktivieren und den standardmäßigen DNS-Namen *`MultiRegionAccessPoint_alias`*`.accesspoint.s3-global.amazonaws.com` des Multi-Region Access Point wie in diesem Abschnitt beschrieben verwenden.

Wenn Sie Anforderungen an die API für Amazon-S3-Datenoperationen (z. B. `GetObject`) über einen Multi-Region Access Point stellen, lautet der Hostname für die Anforderung wie folgt:

`MultiRegionAccessPoint_alias``.accesspoint.s3-global.amazonaws.com`

Wenn Sie zum Beispiel eine `GetObject`-Anforderung über den Multi-Region Access Point mit dem Alias `mfzwi23gnjvgw.mrap` stellen möchten, stellen Sie eine Anforderung an den Hostnamen `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`. Der `s3-global`-Teil des Hostnamens gibt an, dass dieser Hostname nicht für eine bestimmte Region bestimmt ist.

Das Senden von Anforderungen über einen Multi-Regions-Zugriffspunkt ähnelt der Ausführung von Anforderungen über einen Zugriffspunkt für einzelne Regionen. Es ist jedoch wichtig, folgende Unterschiede zu beachten:

- ARNs von Multi-Region Access Points enthalten keine AWS-Region. Sie folgen dem Format `arn:aws:s3::`*`account-id`*`:accesspoint/`*`MultiRegionAccessPoint_alias`*.
- Für Anforderungen, die über API-Operationen gestellt werden (diese Anforderungen erfordern keinen ARN), verwenden Multi-Region Access Points ein anderes Endpunktschema. Das Schema ist *`MultiRegionAccessPoint_alias`*`.accesspoint.s3-global.amazonaws.com` – z. B. `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`. Beachten Sie die Unterschiede im Vergleich zu einem Zugriffspunkt für einzelne Regionen:
  - Hostnamen des Multi-Regions-Zugriffspunkts verwenden ihren Alias, nicht den Namen des Multi-Regions-Zugriffspunkts.
  - Hostnamen von Multi-Region Access Points enthalten nicht die AWS-Konto-ID des Besitzers.
  - Hostnamen von Multi-Region Access Points enthalten keine AWS-Region.
  - Zu den Multi-Regions-Hostnamen gehören `s3-global.amazonaws.com` anstelle von `s3.amazonaws.com`.
- Anforderungen für Multi-Region Access Points müssen mit Signature Version 4A (SigV4A) signiert werden. Wenn Sie die AWS-SDKs verwenden, konvertiert das SDK ein SigV4 automatisch in SigV4A. Stellen Sie daher sicher, dass Ihr AWS-SDK SigV4A als Signaturimplementierung unterstützt, die zum Signieren der globalen Anforderungen der AWS-Region verwendet wird. Weitere Informationen zu SigV4A finden Sie unter Signieren von AWS-API-Anforderungen in der Allgemeine AWS-Referenz.

Multi-Regions-Zugriffspunkte und Amazon S3 Transfer Acceleration

Amazon S3 Transfer Acceleration ist eine Funktion, die schnelle Datenübertragungen zu Buckets ermöglicht. Transfer Acceleration wird auf der Ebene des einzelnen Buckets konfiguriert. Weitere Informationen zu Transfer Acceleration finden Sie unter Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration.

Multi-Region Access Points verwenden einen ähnlichen beschleunigten Übertragungsmechanismus wie Transfer Acceleration, um große Objekte über das AWS-Netzwerk zu senden. Aus diesem Grund müssen Sie Transfer Acceleration nicht verwenden, wenn Sie Anforderungen über einen Multi-Region Access Point senden. Diese erhöhte Übertragungsleistung wird automatisch in den Multi-Region Access Point integriert.

Themen

- Berechtigungen
- Einschränkungen und Beschränkungen des Multi-Regions-Zugriffspunkts
- Weiterleitung der Multi-Regions-Zugriffspunktanforderung
- Failover-Kontrollen für Amazon S3 Multi-Region Access Points
- Konfigurieren einer Replikation zur Verwendung mit Multi-Region Access Points
- Verwenden von Access Points mit mehreren Regionen mit unterstützten Vorgängen API
- Überwachung und Protokollierung von Anforderungen, die über einen Multi-Regions-Zugriffspunkt an zugrunde liegende Ressourcen erfolgen

Berechtigungen

Amazon S3 Multiregion Access Points können den Datenzugriff für Amazon S3 S3-Buckets in mehreren Bereichen vereinfachen. AWS-Regionen Multi-Region Access Points sind benannte globale Endpunkte, mit denen Sie Datenzugriffsoperationen für Objekte in Amazon S3 ausführen können, z. B. `GetObject` und `PutObject`. Jeder Multi-Region Access Point kann für jede Anforderung, die über den globalen Endpunkt eingehen, über unterschiedliche Berechtigungen und Netzwerkkontrollen verfügen.

Jeder Multi-Region Access Point kann außerdem eine benutzerdefinierte Zugriffsrichtlinie erzwingen, die in Verbindung mit der Bucket-Richtlinie funktioniert, welche dem zugrunde liegenden Bucket angefügt ist. Damit eine Anforderung erfolgreich ist, müssen alle folgenden Komponenten die Operation zulassen:

- Die Richtlinie der Multi-Region Access Points
- Die zugrunde liegende Richtlinie AWS Identity and Access Management () IAM
- Die zugrunde liegende Bucket-Richtlinie (an die die Anforderung weitergeleitet wird)

Sie können jede Access Point-Richtlinie für mehrere Regionen so konfigurieren, dass sie nur Anfragen von bestimmten IAM Benutzern oder Gruppen akzeptiert. Ein Beispiel für diese Vorgehensweise finden Sie in Beispiel 2 unter the section called “Beispielrichtlinien für Multi-Region Access Points”. Um den Amazon S3 S3-Datenzugriff auf ein privates Netzwerk zu beschränken, können Sie die Multi-Region-Zugriffspunkt-Richtlinie so konfigurieren, dass nur Anfragen von einer virtuellen privaten Cloud akzeptiert werden (VPC).

Nehmen wir zum Beispiel an, dass Sie eine `GetObject` Anfrage über einen Multi-Region-Access Point stellen, indem Sie einen Benutzer verwenden, der `AppDataReader` in Ihrem AWS Konto angerufen wird. Um sicherzustellen, dass die Anforderung nicht abgelehnt wird, muss der Benutzer `AppDataReader` die `s3:GetObject`-Berechtigung vom Multi-Region Access Point und von jedem Bucket, der dem Multi-Region Access Point zugrunde liegt, erhalten. `AppDataReader` kann keine Daten aus einem Bucket abrufen, der diese Berechtigung nicht erteilt.

> **Important**
>
> Das Delegieren der Zugriffskontrolle für einen Bucket an eine Multi-Region-Access Point-Richtlinie ändert nichts am Verhalten des Buckets, wenn direkt über seinen Bucket-Namen oder Amazon-Ressourcennamen () ARN auf den Bucket zugegriffen wird. Alle Operationen, die direkt für den Bucket ausgeführt werden, funktionieren weiterhin wie zuvor. Einschränkungen, die Sie in eine Richtlinie für Multi-Region Access Points einschließen, gelten nur für Anforderungen, die über diesen Multi-Region Access Point eingehen.

Verwalten des öffentlichen Zugriffs auf einen Multi-Regions-Zugriffspunkt

Multi-Regions-Zugriffspunkte unterstützen unabhängige Block-Einstellungen des öffentlichen Zugriffs für jeden Multi-Regions-Zugriffspunkt. Wenn Sie einen Multi-Regions-Zugriffspunkt erstellen, können Sie die Block-Einstellungen des öffentlichen Zugriffs für diesen Multi-Regions–Zugriffspunkt festlegen.

> **Note**
>
> Alle Einstellungen für „Öffentlichen Zugriff blockieren“, die unter „Einstellungen „Öffentlichen Zugriff beschränken“ für dieses Konto (in Ihrem eigenen Konto) oder Einstellungen

„Öffentlichen Zugriff beschränken“ für externe Buckets aktiviert sind, gelten auch dann, wenn die unabhängigen Einstellungen für „Öffentlichen Zugriff blockieren“ für Ihren Multi-Region Access Point deaktiviert sind.

Für jede Anforderung, die über einen Multi-Region Access Point erfolgt, wertet Amazon S3 die Einstellungen zum Blockieren des öffentlichen Zugriffs für Folgendes aus:

- Den Multi-Region Access Point
- Die zugrunde liegenden Buckets (einschließlich externer Buckets)
- Das Konto, das den Multi-Region Access Point besitzt
- Das Konto, das die zugrunde liegenden Buckets besitzt (einschließlich externer Konten)

Wenn eine dieser Einstellungen darauf hinweist, dass die Anforderung gesperrt werden soll, lehnt Amazon S3 die Anforderung ab. Weitere Hinweise zur Funktion Amazon S3 Block Public Access finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

Important

Alle Einstellungen für die Blockierung des öffentlichen Zugriffs sind für neue Multi-Region Access Points standardmäßig aktiviert. Sie müssen alle Einstellungen, die Sie nicht auf einen Multi-Regions-Zugriffspunkt anwenden möchten, explizit deaktivieren.
Nach dem Erstellen eines Multi-Region Access Point können Sie die Block-Public-Access-Einstellungen für den Access Point nicht mehr ändern.

Anzeigen der Einstellungen für die Blockierung des öffentlichen Zugriffs für einen Multi-Region Access Point

So zeigen Sie die Einstellungen für die Blockierung des öffentlichen Zugriffs für einen Multi-Region Access Point an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
3. Wählen Sie den Namen des Multi-Region Access Points aus, den Sie überprüfen möchten.

4. Wählen Sie die Registerkarte Berechtigungen.
5. Überprüfen Sie unter Block Public Access settings for this Multi-Region Access Point (Einstellungen für die Blockierung des öffentlichen Zugriffs für diesen Multi-Region Access Point) die Einstellungen für die Blockierung des öffentlichen Zugriffs für Ihren Multi-Region Access Point.

> **Note**
>
> Nachdem der Multi-Region Access Point erstellt wurde, können Sie die Einstellungen für die Blockierung des öffentlichen Zugriffs nicht mehr bearbeiten. Wenn Sie den öffentlichen Zugriff blockieren möchten, stellen Sie daher sicher, dass Ihre Anwendungen ohne öffentlichen Zugriff ordnungsgemäß funktionieren, bevor Sie einen Multi-Region Access Point erstellen.

Verwenden einer Richtlinie für Multi-Region Access Points

Die folgende Beispielrichtlinie für Access Points für mehrere Regionen gewährt einem IAM Benutzer Zugriff auf das Auflisten und Herunterladen von Dateien von Ihrem Multi-Region-Access Point. Wenn Sie diese Beispielrichtlinie verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre eigenen Informationen.

```
{
  "Version":"2012-10-17",
  "Statement":[
     {
        "Effect":"Allow",
        "Principal":{
           "AWS":"arn:aws:iam::123456789012:user/JohnDoe"
        },
        "Action":[
           "s3:ListBucket",
           "s3:GetObject"
        ],
        "Resource":[
           "arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias",
           "arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias/object/
*"
        ]
     }
```

```
    ]
}
```

Verwenden Sie den folgenden Befehl `put-multi-region-access-point-policy`, um die Richtlinie für Multi-Region Access Points über die AWS Command Line Interface (AWS CLI) mit dem angegebenen Multi-Region Access Point zu verknüpfen. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen. Jeder Multi-Region Access Point kann nur über eine Richtlinie verfügen. Daher ersetzt eine an Aktion `put-multi-region-access-point-policy` gerichtete Anforderung jede vorhandene Richtlinie, die dem angegebenen Multi-Region Access Point zugeordnet ist.

AWS CLI

```
aws s3control put-multi-region-access-point-policy
--account-id 111122223333
--details { "Name": "amzn-s3-demo-bucket-MultiRegionAccessPoint",
 "Policy": "{ \"Version\": \"2012-10-17\", \"Statement\": { \"Effect\":
 \"Allow\", \"Principal\": { \"AWS\": \"arn:aws:iam::111122223333:root
\" }, \"Action\": [\"s3:ListBucket\", \"s3:GetObject\"], \"Resource\":
 [ \"arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias",
 \"arn:aws:s3::111122223333:accesspoint/MultiRegionAccessPoint_alias/object/*
\" ] } }" }
```

Verwenden Sie den folgenden Befehl, um die Ergebnisse der vorherigen Operation abzufragen:

AWS CLI

```
aws s3control describe-multi-region-access-point-operation
--account-id 111122223333
--request-token-arn requestArn
```

Verwenden Sie den folgenden Befehl, um Ihre Richtlinie für Multi-Region Access Points abzurufen:

AWS CLI

```
aws s3control get-multi-region-access-point-policy
--account-id 111122223333
--name=amzn-s3-demo-bucket-MultiRegionAccessPoint
```

Bearbeiten der Richtlinie für Multi-Region Access Points

Die Multi-Region-Zugriffspunkt-Richtlinie (beschriebenJSON) ermöglicht den Speicherzugriff auf die Amazon S3 S3-Buckets, die mit diesem Multi-Region-Access Point verwendet werden. Sie können bestimmten Prinzipalen die Ausführung verschiedener Aktionen auf Ihrem Multi-Region Access Point erlauben oder verweigern. Wenn eine Anforderung über den Multi-Region Access Point an einen Bucket weitergeleitet wird, gelten sowohl die Zugriffsrichtlinien für den Multi-Region Access Point als auch den Bucket. Die restriktivere Zugriffsrichtlinie hat immer Vorrang.

> **Note**
>
> Wenn ein Bucket Objekte enthält, die anderen Konten gehören, gilt die Richtlinie für Multi-Region Access Points nicht für Objekte, die anderen AWS-Konten gehören.

Nachdem Sie eine Richtlinie für Multi-Region Access Points angewendet haben, kann die Richtlinie nicht mehr gelöscht werden. Sie können entweder die Richtlinie bearbeiten oder eine neue Richtlinie erstellen, die die bestehende überschreibt.

So bearbeiten Sie die Richtlinie für Multi-Region Access Points

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
3. Wählen Sie den Namen des Multi-Region Access Point aus, für den Sie die Richtlinie bearbeiten möchten.
4. Wählen Sie die Registerkarte Berechtigungen.
5. Scrollen Sie nach unten zum Abschnitt Richtlinie für Multi-Region Access Points. Wählen Sie Bearbeiten, um die Richtlinie zu aktualisieren (inJSON).
6. Die Seite Edit Multi-Region Access Point policy (Richtlinie für Multi-Region Access Points bearbeiten) wird angezeigt. Sie können die Richtlinie entweder direkt in das Textfeld eingeben oder Add statement (Anweisung hinzufügen) auswählen, um Richtlinienelemente aus einer Dropdownliste auszuwählen.

> **Note**
>
> In der Konsole wird automatisch der Amazon Resource Name (ARN) für den multiregionalen Access Point angezeigt, den Sie in der Richtlinie verwenden können. Für Beispielrichtlinien für Multi-Region Access Points vgl. the section called “Beispielrichtlinien für Multi-Region Access Points”.

Beispielrichtlinien für Multi-Region Access Points

Amazon S3 Multiregion Access Points unterstützen Ressourcenrichtlinien AWS Identity and Access Management (IAM). Mithilfe dieser Richtlinien können Sie die Verwendung des Multi-Region Access Point nach Ressource, Benutzer oder anderen Bedingungen steuern. Damit eine Anwendung oder ein Benutzer über einen Multi-Region Access Point auf Objekte zugreifen kann, müssen sowohl der Multi-Region Access Point Zugriffspunkt als auch der zugrunde liegende Bucket den gleichen Zugriff erlauben.

Gehen Sie folgendermaßen vor, um den gleichen Zugriff sowohl auf den Multi-Region Access Point als auch auf den zugrunde liegenden Bucket zu erlauben:

- (Empfohlen) Wenn Sie die Zugriffskontrollen bei der Verwendung eines Amazon S3 Multi-Region Access Points vereinfachen möchten, delegieren Sie die Zugriffskontrolle für den Amazon-S3-Bucket an den Multi-Region Access Point. Ein Beispiel für diese Vorgehensweise finden Sie in Beispiel 1 in diesem Abschnitt.
- Fügen Sie der Richtlinie des zugrunde liegenden Buckets dieselben Berechtigungen hinzu, die in der Richtlinie des Multi Region Access Points enthalten sind.

> **Important**
>
> Das Delegieren der Zugriffskontrolle für einen Bucket an eine Multi-Region-Access Point-Richtlinie ändert nichts am Verhalten des Buckets, wenn direkt über seinen Bucket-Namen oder Amazon-Ressourcennamen () ARN auf den Bucket zugegriffen wird. Alle Operationen, die direkt für den Bucket ausgeführt werden, funktionieren weiterhin wie zuvor. Einschränkungen, die Sie in eine Richtlinie für Multi-Region Access Points einschließen, gelten nur für Anforderungen, die über diesen Multi-Region Access Point eingehen.

Example 1 – Delegieren des Zugriffs auf bestimmte Multi-Region Access Points in Ihrer Bucket-Richtlinie (für dasselbe Konto oder kontoübergreifend)

Das folgende Beispiel für eine Bucket-Richtlinie gewährt vollständigen Bucket-Zugriff auf einen bestimmten Multi-Region Access Point. Dies bedeutet, dass der gesamte Zugriff auf diesen Bucket durch die Richtlinien gesteuert wird, die dem Multi-Region Access Point angefügt sind. Wir empfehlen, Ihre Buckets auf diese Weise für alle Anwendungsfälle zu konfigurieren, die keinen direkten Zugriff auf den Bucket erfordern. Sie können diese Bucket-Richtlinienstruktur für Multi-Region Access Points in demselben Konto oder einem anderen Konto verwenden.

```
{
    "Version": "2012-10-17",
    "Statement" : [
    {
        "Effect": "Allow",
        "Principal" : { "AWS": "*" },
        "Action" : "*",
        "Resource" : [ "Bucket ARN", "Bucket ARN/*"],
        "Condition": {
            "StringEquals" : { "s3:DataAccessPointArn" : "MultiRegionAccessPoint_ARN" }
        }
    }]
}
```

> **Note**
>
> Wenn Sie für mehrere Multi-Region Access Points Zugriff erteilen, stellen Sie sicher, dass Sie jeden einzelnen Multi-Region Access Point auflisten.

Example 2 – Gewähren des Zugriffs auf einen Multi-Region Access Point für ein Konto in Ihrer Richtlinie für Multi-Region Access Points

Die folgende Richtlinie für multiregionale Access Points ermöglicht es dem Konto*123456789012*, die Objekte aufzulisten und zu lesen, die sich in dem multiregionalen Access Point befinden, der durch den *MultiRegionAccessPoint_ARN*.

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

```
      "Effect": "Allow",
      "Principal": {
         "AWS":"arn:aws:iam::123456789012:user/JohnDoe"
      },
      "Action":[
         "s3:ListBucket",
         "s3:GetObject"
      ],
      "Resource":[
         "MultiRegionAccessPoint_ARN",
         "MultiRegionAccessPoint_ARN/object/*"
      ]
    }
  ]
}
```

Example 3 – Richtlinie für Multi-Region Access Points, die eine Bucket-Auflistung ermöglicht

Die folgende Richtlinie für multiregionale Access Points ermöglicht es Benutzerkonten*123456789012*, die Objekte aufzulisten, die sich in dem multiregionalen Access Point befinden, definiert durch *MultiRegionAccessPoint_ARN*.

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
       "Effect": "Allow",
       "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/JohnDoe"
        },
        "Action": "s3:ListBucket",
        "Resource": "MultiRegionAccessPoint_ARN"
       }
    ]
}
```

Einschränkungen und Beschränkungen des Multi-Regions-Zugriffspunkts

Multi-Regions-Zugriffspunkte in Amazon S3 haben die folgenden Einschränkungen und Beschränkungen.

- Namen für Multi-Regions-Zugriffspunkte:

  - Muss innerhalb eines einzelnen AWS Kontos eindeutig sein
  - Muss mit einer Zahl oder einem Kleinbuchstaben beginnen
  - Muss zwischen 3 und 50 Zeichen lang sein.
  - Dürfen nicht mit einem Bindestrich (`-`) beginnen oder enden
  - Dürfen keine Unterstriche (`_`), Großbuchstaben oder Punkte (`.`) enthalten
  - Können nicht mehr bearbeitet werden, nachdem sie erstellt wurden
- Aliase für Multi-Region Access Points werden von Amazon S3 generiert und können nicht bearbeitet oder wiederverwendet werden.
- Sie können über einen Multi-Region Access Point nicht mit Gateway-Endpunkten auf Daten zugreifen. Sie können jedoch über einen Multi-Region Access Point mit Schnittstellenendpunkten auf Daten zugreifen. Zur Verwendung AWS PrivateLink müssen Sie VPC-Endpoints erstellen. Weitere Informationen finden Sie unter Konfigurieren eines Multi-Region Access Point zur Verwendung mit AWS PrivateLink.
- Um Multi-Region Access Points mit Amazon zu verwenden CloudFront, müssen Sie den Multi-Region Access Point als `Custom Origin` Vertriebstyp konfigurieren. Weitere Informationen zu verschiedenen Herkunftstypen finden Sie unter Verwenden verschiedener Herkunft bei CloudFront Verteilungen. Weitere Informationen zur Verwendung von Multi-Region-Access Points mit Amazon CloudFront finden Sie im Storage-Blog unter Aufbau einer Active-Active-Proximity-basierten Anwendung in mehreren Regionen.AWS
- Mindestanforderungen für Multi-Regions-Zugriffspunkte:
  - Transport Layer Security (TLS) v1.2
  - Signatur-Version 4 (SigV4A)

    Multi-Regions-Zugriffspunkte unterstützen Signatur-Version 4A. Diese Version von SigV4 ermöglicht das Signieren von Anforderungen für mehrere AWS-Regionen. Dies ist nützlich bei API-Operationen, die zu Datenzugriff von einer oder mehreren Regionen führen können. Wenn Sie ein AWS SDK verwenden, geben Sie Ihre Anmeldeinformationen an, und für Anfragen an Access Points mit mehreren Regionen wird Signature Version 4A ohne zusätzliche Konfiguration verwendet. Stellen Sie sicher, dass Sie die Kompatibilität Ihres AWS SDK mit dem SigV4A-Algorithmus überprüfen. Weitere Informationen zu SigV4a finden Sie unter Signieren von AWS API-Anfragen im. Allgemeine AWS-Referenz

> **Note**
>
> Um SigV4a mit temporären Sicherheitsanmeldedaten zu verwenden — beispielsweise bei der Verwendung von AWS Identity and Access Management (IAM-) Rollen — können Sie die temporären Anmeldeinformationen von einem regionalen () Endpunkt anfordern. AWS Security Token Service AWS STS Wenn Sie temporäre Anmeldeinformationen vom globalen AWS STS Endpunkt (`sts.amazonaws.com`) anfordern, müssen Sie zunächst die Regionskompatibilität der Sitzungstoken für den globalen Endpunkt so einstellen, dass sie für alle gültig ist. AWS-Regionen Weitere Informationen finden Sie unter Managing AWS STSAWS-Region in an im IAM-Benutzerhandbuch.

- Multi-Region Access Points unterstützen keine anonymen Anforderungen.
- Beschränkungen des Multi-Regions-Zugriffspunkts:
  - IPv6 wird nicht unterstützt.
  - Buckets von Amazon S3 on Outposts werden nicht unterstützt.
  - Multi-Region Access Points unterstützen Kopiervorgänge, bei denen Multi-Region Access Points nur als Ziel verwendet werden, wenn der Multi-Region Access Point ARN verwendet wird.
  - Die Funktion S3 Batch Operations wird nicht unterstützt.
- Bestimmte AWS SDKs werden nicht unterstützt. Informationen darüber, welche AWS SDKs für Access Points mit mehreren Regionen unterstützt werden, finden Sie unter Kompatibilität mit SDKs. AWS
- Service Quotas für Multi-Region Access Points lauten wie folgt:
  - Pro Konto gibt es maximal 100 Multi-Regions-Zugriffspunkte.
  - Es gibt ein Limit von 17 Regionen für einen einzelnen Multi-Region Access Point.
- Nach der Erstellung eines Multi-Region Access Point können Sie keine Buckets hinzufügen, ändern oder aus der Konfiguration des Multi-Region Access Point entfernen. Um die Buckets zu ändern, müssen Sie den gesamten Multi-Regions-Zugriffspunkt löschen und einen neuen erstellen. Wenn ein kontoübergreifender Bucket in Ihrem Multi-Region Access Point gelöscht wird, besteht die einzige Möglichkeit, diesen Bucket erneut zu verbinden, darin, den Bucket neu zu erstellen und dabei denselben Namen und dieselbe Region in diesem Konto zu verwenden.
- Zugrunde liegende Buckets (in demselben Konto), die in einem Multi-Region Access Point verwendet werden, können erst nach dem Löschen eines Multi-Region Access Point gelöscht werden.

- Alle Anforderungen auf Steuerebene zum Erstellen oder Verwalten von Multi-Region Access Points müssen an die Region `US West (Oregon)` weitergeleitet werden. Für Anforderungen von Multi-Region Access Point auf Steuerebene müssen keine Regionen angegeben werden.
- Für die Failover-Steuerebene von Multi-Region Access Points muss die Anforderung an eine dieser fünf unterstützten Regionen weitergeleitet werden:
  - `US East (N. Virginia)`
  - `US West (Oregon)`
  - `Asia Pacific (Sydney)`
  - `Asia Pacific (Tokyo)`
  - `Europe (Ireland)`
- Ihr Access Point für mehrere Regionen unterstützt nur Buckets in den folgenden Bereichen: AWS-Regionen
  - `US East (N. Virginia)`
  - `US East (Ohio)`
  - `US West (N. California)`
  - `US West (Oregon)`
  - `Asia Pacific (Mumbai)`
  - `Asia Pacific (Osaka)`
  - `Asia Pacific (Seoul)`
  - `Asia Pacific (Singapore)`
  - `Asia Pacific (Sydney)`
  - `Asia Pacific (Tokyo)`
  - `Canada (Central)`
  - `Europe (Frankfurt)`
  - `Europe (Ireland)`
  - `Europe (London)`
  - `Europe (Paris)`
  - `Europe (Stockholm)`
  - `South America (São Paulo)`

Weiterleitung der Multi-Regions-Zugriffspunktanforderung

Wenn Sie eine Anforderung über einen Multi-Region Access Point stellen, bestimmt Amazon S3, welche der Buckets, die dem Multi-Region Access Point zugeordnet sind, sich in Ihrer nächster Nähe befinden. Amazon S3 leitet die Anforderung dann an diesen Bucket weiter, unabhängig von der AWS-Region, in der sie sich befindet.

Nachdem der Multi-Region Access Point die Anforderung an den nächstgelegenen Bucket weitergeleitet hat, verarbeitet Amazon S3 die Anforderung so, als ob Sie sie direkt an diesen Bucket gestellt hätten. Multi-Region Access Points kennen den Dateninhalt eines Amazon-S3-Buckets nicht. Daher kann es sein, dass der Bucket, der die Anforderung erhält, die angeforderten Daten nicht enthält. Um konsistente Datensätze in den Amazon-S3-Buckets zu schaffen, die einem Multi-Region Access Point zugeordnet sind, können Sie S3 Cross-Region Replication (CRR, regionsübergreifende Replikation) konfigurieren. Dann kann jeder Bucket die -Anforderung erfolgreich erfüllen.

Amazon S3 leitet Multi-Regions-Zugriffspunktanforderungen gemäß den folgenden Regeln weiter:

- Amazon S3 optimiert Anforderungen, um der Nähe entsprechend erfüllt zu werden. Es untersucht die Buckets, die vom Multi-Region Access Point unterstützt werden, und leitet die Anforderung an den Bucket weiter, der sich in nächster Nähe befindet.
- Wenn die Anforderung eine vorhandene Ressource angibt (z. B. `GetObject`), berücksichtigt Amazon S3 nicht den Namen des Objekts bei der Erfüllung der Anforderung. Selbst wenn ein Objekt in einem Bucket in dem Multi-Region Access Point vorhanden ist, kann Ihre Anforderung somit an einen Bucket weitergeleitet werden, der das Objekt nicht enthält. Dies führt dazu, dass dem Kunden eine 404-Fehlermeldung zurückgegeben wird.

  Um 404-Fehler zu vermeiden, empfehlen wir Ihnen, S3 Cross-Region Replication (CRR) für Ihre Buckets zu konfigurieren. Die Replikation hilft bei der Lösung des potenziellen Problems, das entsteht, wenn sich das gewünschte Objekt in einem Bucket in dem Multi-Region Access Point befindet, nicht jedoch in dem bestimmten Bucket, an den Ihre Anforderung weitergeleitet wurde. Weitere Informationen zum Erstellen einer grundlegenden Replikationskonfiguration finden Sie unter Konfigurieren einer Replikation zur Verwendung mit Multi-Region Access Points.

  Um sicherzustellen, dass Ihre Anforderungen mithilfe der gewünschten Objekte erfüllt werden, wird außerdem empfohlen, die Bucket-Versionsverwaltung zu aktivieren und Versions-IDs in Ihre Anforderungen aufzunehmen. Auf diese Weise stellen Sie sicher, dass Sie über die richtige Version des gesuchten Objekts verfügen. Buckets mit aktivierter Versionsverwaltung können auch bei der

Wiederherstellung von Objekten nach einem versehentlichen Überschreiben hilfreich sein. Weitere Informationen finden Sie unter Verwenden des S3-Versioning in S3-Buckets.

- Wenn sich die Anforderung auf das Erstellen einer Ressource bezieht (z. B. `PutObject` oder `CreateMultipartUpload`), erfüllt Amazon S3 die Anforderung unter Verwendung des nächstgelegenen Buckets. Betrachten wir beispielsweise ein Videounternehmen, das Video-Uploads von überall auf der Welt unterstützen möchte. Wenn ein Benutzer eine PUT-Anforderung an den Multi-Region Access Point stellt, wird das Objekt in den nächstgelegenen Bucket gestellt. Um das hochgeladene Video dann anderen auf der ganzen Welt mit der geringsten Latenz zum Download zur Verfügung zu stellen, können Sie CRR mit bidirektionaler Replikation verwenden. Wenn Sie CRR mit bidirektionaler Replikation verwenden, bleiben die Inhalte aller Buckets, die dem Multi-Region Access Point zugeordnet sind, synchronisiert. Weitere Informationen zur Verwendung der Replikation mit Multi-Region Access Points finden Sie unter Konfigurieren einer Replikation zur Verwendung mit Multi-Region Access Points.

Failover-Kontrollen für Amazon S3 Multi-Region Access Points

Mit den Failover-Kontrollen für Amazon S3 Multi-Region Access Points können Sie die Geschäftskontinuität bei regionalen Verkehrsunterbrechungen aufrechterhalten und Ihren Anwendungen gleichzeitig eine regionsübergreifende Architektur zur Erfüllung von Compliance- und Redundanzanforderungen bieten. Wenn Ihr regionaler Datenverkehr unterbrochen wird, können Sie mithilfe der Failover-Steuerelemente für mehrere Regionen auswählen, welcher AWS-Regionen hinter einem Amazon S3-Access Point mit mehreren Regionen Datenzugriffs- und Speicheranfragen verarbeitet werden soll.

Zur Failover-Unterstützung können Sie Ihren Multi-Region Access Point in einer Aktiv-Passiv-Konfiguration einrichten, wobei der Datenverkehr unter normalen Bedingungen in die aktive Region fließt und sich eine passive Region für den Failover im Standby-Modus befindet.

Um beispielsweise ein Failover auf eine AWS-Region Region Ihrer Wahl durchzuführen, verlagern Sie den Datenverkehr von Ihrer primären (aktiven) Region in Ihre sekundäre (passive) Region. In einer solchen Aktiv-Passiv-Konfiguration ist ein Bucket aktiv und akzeptiert Datenverkehr, während der andere Bucket passiv ist und keinen Traffic akzeptiert. Der passive Bucket wird für die Notfallwiederherstellung verwendet. Wenn Sie den Failover einleiten, wird der gesamte Datenverkehr (wie `GET`- und `PUT`-Anforderungen) an den Bucket im aktiven Zustand (in einer Region) und weg vom Bucket im passiven Zustand (in einer anderen Region) geleitet.

Wenn Sie die regionsübergreifende S3-Replikation (CRR) mit bidirektionalen Replikationsregeln aktiviert haben, können Sie Ihre Buckets während eines Failovers synchronisieren. Wenn Sie die

CRR Aktivierung in einer Active-Active-Konfiguration aktiviert haben, können Amazon S3 Multi-Region Access Points außerdem Daten vom nächstgelegenen Bucket-Standort abrufen, wodurch die Anwendungsleistung verbessert wird.

AWS-Region Unterstützung

Mit den Failover-Kontrollen für Amazon S3 Multi-Region Access Points können sich Ihre S3-Buckets in jeder der 17 Regionen befinden, in denen Multi-Region Access Points unterstützt werden. Sie können ein Failover für zwei beliebige Regionen gleichzeitig initiieren.

> **Note**
>
> Obwohl ein Failover nur zwischen zwei Regionen gleichzeitig initiiert wird, können Sie den Routing-Status für mehrere Regionen gleichzeitig in Ihrem Multi-Region Access Point separat aktualisieren.

In den folgenden Themen wird die Verwendung und Verwaltung von Failover-Kontrollen für Amazon S3 Multi-Region Access Points veranschaulicht.

Themen

- Weiterleitungsstatus von Amazon S3 Multi-Region Access Points
- Verwenden von Failover-Kontrollen für Amazon S3 Multi-Region Access Points
- Failover-Kontrollfehler für Amazon S3 Multi-Region Access Points

Weiterleitungsstatus von Amazon S3 Multi-Region Access Points

Ihre Failover-Konfiguration für Amazon S3 Multi-Region Access Points bestimmt den Weiterleitungsstatus der AWS-Regionen , die mit dem Multi-Region Access Point verwendet werden. Sie können Ihren Amazon S3 Multi-Region Access Point so konfigurieren, dass er sich in einem Aktiv-Aktiv- oder Aktiv-Passiv-Zustand befindet.

- Aktiv-Aktiv – In einer Aktiv-Aktiv-Konfiguration werden alle Anfragen automatisch an die nächstgelegenen AWS-Region in Ihrem Multi-Region Access Point gesendet. Nachdem der Multi-Region Access Point so konfiguriert wurde, dass er sich in einem Aktiv-Aktiv-Zustand befindet, können alle Regionen Datenverkehr empfangen. Wenn in einer Aktiv-Aktiv-Konfiguration eine Verkehrsunterbrechung auftritt, wird der Netzwerkverkehr automatisch in eine der aktiven Regionen umgeleitet.

- Aktiv-Passiv – In einer Aktiv-Passiv-Konfiguration empfangen die aktiven Regionen in Ihrem Multi-Region Access Point Traffic und die passiven Regionen nicht. Wenn Sie beabsichtigen, S3-Failover-Kontrollen zu verwenden, um in einer Notfallsituation ein Failover einzuleiten, richten Sie Ihre Multi-Region Access Points in einer Aktiv-Passiv-Konfiguration ein, während Sie die Notfallwiederherstellungsplanung testen und durchführen.

Verwenden von Failover-Kontrollen für Amazon S3 Multi-Region Access Points

In diesem Abschnitt wird erläutert, wie Sie Ihre Amazon S3 Multi-Region Access Points mithilfe der AWS Management Console verwalten und verwenden.

Es gibt zwei Failover-Steuerelemente im Abschnitt Failover-Konfiguration auf der Detailseite Ihres Access Points für mehrere Regionen unter AWS Management Console: Routing-Status bearbeiten und Failover. Verwenden Sie diese Steuerelemente wie folgt:

- Routing-Status bearbeiten — Sie können den Routing-Status von bis zu 17 AWS-Regionen in einer einzigen Anfrage für Ihren Multi-Region Access Point manuell bearbeiten, indem Sie Routing-Status bearbeiten wählen. Sie können die Option Edit routing status (Weiterleitungsstatus bearbeiten) für folgende Zwecke verwenden:
  - Zum Festlegen oder Bearbeiten der Weiterleitungsstatus einer oder mehrerer Regionen in Ihrem Multi-Region Access Point
  - Zum Erstellen einer Failover-Konfiguration für Ihren Multi-Region Access Point, indem Sie zwei Regionen so konfigurieren, dass sie sich in einem Aktiv-Passiv-Zustand befinden
  - So führen Sie ein manuelles Failover für Ihre Regionen durch
  - So leiten Sie Traffic manuell zwischen Regionen um
- Failover – Wenn Sie ein Failover initiieren, indem Sie Failover auswählen, aktualisieren Sie nur den Weiterleitungsstatus von zwei Regionen, die bereits so konfiguriert sind, dass sie sich in einem Aktiv-Passiv-Zustand befinden. Während eines Failovers, das Sie initiiert haben, indem Sie Failover auswählen, wird der Weiterleitungsstatus zwischen den beiden Regionen automatisch gewechselt.

Bearbeiten des Weiterleitungsstatus der Regionen in Ihrem Multi-Region Access Point

Sie können den Routing-Status von bis zu 17 AWS-Regionen in einer einzigen Anfrage für Ihren Multi-Region Access Point manuell aktualisieren, indem Sie auf Ihrer Detailseite für Multi-Region Access Point im Abschnitt Failover-Konfiguration die Option Routing-Status bearbeiten auswählen. Wenn Sie jedoch ein Failover initiieren, indem Sie Failover auswählen, aktualisieren Sie nur den

Weiterleitungsstatus von zwei Regionen, die bereits so konfiguriert sind, dass sie sich in einem Aktiv-Passiv-Zustand befinden. Während eines Failovers, das Sie initiiert haben, indem Sie Failover auswählen, wird der Weiterleitungsstatus zwischen den beiden Regionen automatisch gewechselt.

Sie können die Option Edit routing status (Weiterleitungsstatus bearbeiten) (wie im folgenden Verfahren beschrieben) für die folgenden Zwecke verwenden:

- Zum Festlegen oder Bearbeiten der Weiterleitungsstatus einer oder mehrerer Regionen in Ihrem Multi-Region Access Point
- Zum Erstellen einer Failover-Konfiguration für Ihren Multi-Region Access Point, indem Sie zwei Regionen so konfigurieren, dass sie sich in einem Aktiv-Passiv-Zustand befinden
- So führen Sie ein manuelles Failover für Ihre Regionen durch
- So leiten Sie Traffic manuell zwischen Regionen um

Verwenden der S3-Konsole

So aktualisieren Sie den Weiterleitungsstatus der Regionen in Ihrem Multi-Region Access Point

1. Melden Sie sich bei der Management Console an. AWS
2. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
3. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
4. Wählen Sie den Multi-Region Access Point aus, den Sie aktualisieren möchten.
5. Wählen Sie die Registerkarte Replication and failover (Replikation und Failover) aus.
6. Wählen Sie eine oder mehrere Regionen aus, deren Weiterleitungsstatus Sie bearbeiten möchten.

   > **Note**
   >
   > Um ein Failover einzuleiten, AWS-Region muss mindestens eine Region als aktiv und eine Region in Ihrem multiregionalen Access Point als passiv gekennzeichnet sein.

7. Wählen Sie Edit routing status (Weiterleitungsstatus bearbeiten) aus.
8. Wählen Sie in dem daraufhin angezeigten Dialogfeld Active (Aktiv) oder Passive (Passiv) als Routing status (Weiterleitungsstatus) für jede Region aus.

Ein aktiver Status ermöglicht die Weiterleitung von Verkehr an die Region. Ein passiver Status verhindert, dass jeglicher Verkehr an die Region weitergeleitet wird.

Wenn Sie eine Failover-Konfiguration für Ihren Multi-Region-Access Point erstellen oder einen Failover einleiten, AWS-Region muss mindestens eine Region in Ihrem Multi-Region-Access Point als Aktiv und eine Region als Passiv gekennzeichnet sein.

9. Wählen Sie Save routing status (Weiterleitungsstatus speichern) aus. Es dauert etwa 2 Minuten, bis der Verkehr umgeleitet wird.

Nachdem Sie den Routing-Status AWS-Regionen für Ihren Multi-Region-Access Point übermittelt haben, können Sie Ihre Änderungen am Routing-Status überprüfen. Um diese Änderungen zu überprüfen, gehen Sie zu Amazon CloudWatch unter, https://console.aws.amazon.com/cloudwatch/um die Verschiebung Ihres Amazon S3 S3-Datenanforderungsverkehrs (z. B. `GET` und `PUT` Anfragen) zwischen aktiven und passiven Regionen zu überwachen. Bestehende Verbindungen werden während des Failovers nicht beendet. Bestehende Verbindungen werden fortgesetzt, bis der Erfolgs- bzw. Misserfolgsstatus angezeigt wird.

Verwenden Sie den AWS CLI

> **Note**
>
> Sie können AWS CLI Routing-Befehle für Access Points mit mehreren Regionen für jede dieser fünf Regionen ausführen:
>
> - `ap-southeast-2`
> - `ap-northeast-1`
> - `us-east-1`
> - `us-west-2`
> - `eu-west-1`

Mit dem folgenden Beispielbefehl wird die aktuelle Weiterleitungskonfiguration Ihres Multi-Region Access Points aktualisiert. Wenn Sie den aktiven oder passiven Status eines Buckets aktualisieren möchten, legen Sie den Wert `TrafficDialPercentage` auf `100` für aktiv und auf `0` für passiv fest. In diesem Beispiel *`amzn-s3-demo-bucket1`* ist auf aktiv gesetzt und *`amzn-s3-demo-bucket2`* ist

auf passiv gesetzt. Wenn Sie dieses Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control submit-multi-region-access-point-routes
--region ap-southeast-2
--account-id 123456789012
--mrap MultiRegionAccessPoint_ARN
--route-updates Bucket=amzn-s3-demo-bucket1,TrafficDialPercentage=100
                Bucket=amzn-s3-demo-bucket2
,TrafficDialPercentage=0
```

Mit dem folgenden Beispielbefehl wird die aktualisierte Weiterleitungskonfiguration Ihres Multi-Region Access Points abgerufen. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control get-multi-region-access-point-routes
--region eu-west-1
--account-id 123456789012
--mrap MultiRegionAccessPoint_ARN
```

Initiieren eines Failovers

Wenn Sie einen Failover einleiten, indem Sie auf der Detailseite Ihres Multi-Region Access Points im Abschnitt Failover configuration (Failover-Konfiguration) die Option Failover auswählen, wird der Amazon-S3-Anforderungsverkehr automatisch auf eine alternative AWS-Region umgeleitet. Der Failover-Prozess ist innerhalb von 2 Minuten abgeschlossen.

Sie können einen Failover in zwei beliebigen Regionen AWS-Regionen gleichzeitig einleiten (von den 17 Regionen, in denen multiregionale Access Points unterstützt werden). Failover-Ereignisse werden dann in AWS CloudTrail protokolliert. Nach Abschluss des Failovers können Sie den Amazon S3 S3-Verkehr und alle Aktualisierungen der Datenverkehrsweiterleitung für die neue aktive Region in Amazon CloudWatch überwachen.

> **Important**
>
> Damit alle Metadaten und Objekte während der Datenreplikation bucketübergreifend synchronisiert werden, empfehlen wir, bidirektionale Replikationsregeln zu erstellen und die Synchronisierung von Replikatänderungen zu aktivieren, bevor Sie Ihre Failover-Kontrollen konfigurieren.

Bidirektionale Replikationsregeln tragen dazu bei, sicherzustellen, dass wenn Daten in den Amazon-S3-Bucket geschrieben werden, auf den der Datenverkehr bei einem Failover zurückgreift, diese Daten dann zurück in den Quell-Bucket repliziert werden. Durch die Synchronisierung von Replikatänderungen wird sichergestellt, dass die Objektmetadaten während der bidirektionalen Replikation auch zwischen Buckets synchronisiert werden. Weitere Informationen zum Konfigurieren der Replikation zur Unterstützung von Failover finden Sie unter the section called “Bucket-Replikation”.

So leiten Sie ein Failover zwischen replizierten Buckets ein

1. Melden Sie sich bei der AWS Management Console an.
2. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
3. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
4. Wählen Sie den Multi-Region Access Point aus, den Sie zum Initiieren des Failovers verwenden möchten.
5. Wählen Sie die Registerkarte Replication and failover (Replikation und Failover) aus.
6. Scrollen Sie nach unten zum Abschnitt Failover configuration (Failover-Konfiguration) und wählen Sie zwei AWS-Regionen aus.

   Note

   Um ein Failover einzuleiten, AWS-Region muss mindestens eine Region als aktiv und eine Region in Ihrem multiregionalen Access Point als passiv gekennzeichnet sein. Ein aktiver Status ermöglicht die Weiterleitung von Verkehr an eine Region. Ein passiver Status verhindert, dass jeglicher Verkehr an die Region weitergeleitet wird.

7. Klicken Sie auf Failover.
8. Wählen Sie im Dialogfeld erneut Failover aus, um den Failover-Vorgang einzuleiten. Während dieses Vorgangs werden die Weiterleitungsstatus der beiden Regionen automatisch gewechselt. Der gesamte neue Traffic wird an die Region geleitet, die aktiv wird, und der Verkehr wird nicht mehr an die Region weitergeleitet, die passiv wird. Es dauert etwa 2 Minuten, bis der Verkehr umgeleitet wird.

   Nachdem Sie den Failover-Prozess eingeleitet haben, können Sie Ihre Verkehrsänderungen überprüfen. Um diese Änderungen zu überprüfen, gehen Sie zu Amazon CloudWatch unter,

https://console.aws.amazon.com/cloudwatch/um die Verschiebung Ihres Amazon S3 S3-Datenanforderungsverkehrs (z. B. GET und PUT Anfragen) zwischen aktiven und passiven Regionen zu überwachen. Bestehende Verbindungen werden während des Failovers nicht beendet. Bestehende Verbindungen werden fortgesetzt, bis der Erfolgs- bzw. Misserfolgsstatus angezeigt wird.

Anzeigen Ihrer Weiterleitungskontrollen für Amazon S3 Multi-Region Access Points

Verwenden der S3-Konsole

So zeigen Sie Ihre Weiterleitungskontrollen für Ihren Amazon S3 Multi-Region Access Point an

1. Melden Sie sich bei der AWS Management Console an.
2. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
3. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
4. Wählen Sie den Multi-Region Access Point aus, den Sie überprüfen möchten.
5. Wählen Sie die Registerkarte Replication and failover (Replikation und Failover) aus. Auf dieser Seite werden die Konfigurationsdetails für die Weiterleitung und eine Zusammenfassung für Ihren Multi-Region Access Point, die zugehörigen Replikationsregeln und Replikationsmetriken angezeigt. Sie können den Weiterleitungsstatus Ihrer Regionen im Abschnitt Failover configuration (Failover-Konfiguration) sehen.

Mit dem AWS CLI

Mit dem folgenden AWS CLI Beispielbefehl wird Ihre aktuelle Routenkonfiguration für Multiregion Access Point für die angegebene Region abgerufen. Wenn Sie dieses Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control get-multi-region-access-point-routes
--region eu-west-1
--account-id 123456789012
--mrap MultiRegionAccessPoint_ARN
```

> **Note**
>
> Dieser Befehl kann nur für diese fünf Regionen ausgeführt werden:

- ap-southeast-2
- ap-northeast-1
- us-east-1
- us-west-2
- eu-west-1

Failover-Kontrollfehler für Amazon S3 Multi-Region Access Points

Wenn Sie die Failover-Konfiguration für Ihren Multiregion Access Point aktualisieren, tritt möglicherweise einer der folgenden Fehler auf:

- HTTP400 Bad Request: Dieser Fehler kann auftreten, wenn Sie ARN bei der Aktualisierung Ihrer Failover-Konfiguration einen ungültigen Access Point für mehrere Regionen eingeben. Sie können Ihren Multi-Region-Access Point ARN anhand Ihrer Richtlinien für Multi-Region-Access Points überprüfen. Informationen zur Überprüfung oder Aktualisierung Ihrer der Richtlinie Ihres Multi-Region Access Points finden Sie unter Bearbeiten der Richtlinie für den Multi-Region Access Point. Dieser Fehler kann auch auftreten, wenn Sie beim Aktualisieren der Failover-Kontrollen für Ihren Amazon S3 Multi-Region Access Point eine leere oder zufällige Zeichenfolge verwenden. Stellen Sie sicher, dass Sie das Format für multiregionale Access Points verwenden: ARN

  `arn:aws:s3::account-id:accesspoint/MultiRegionAccessPoint_alias`
- HTTP503 Verlangsamen: Dieser Fehler tritt auf, wenn Sie in einem kurzen Zeitraum zu viele Anfragen senden. Abgelehnte Anforderungen führen zu einem Fehler.
- HTTP409-Konflikt: Dieser Fehler tritt auf, wenn zwei oder mehr gleichzeitige Anfragen zur Aktualisierung der Routenkonfiguration auf einen einzelnen Access Point mit mehreren Regionen abzielen. Die erste Anforderung ist erfolgreich, aber alle anderen Anforderungen schlagen mit einem Fehler fehl.
- HTTP405 Methode nicht zulässig: Dieser Fehler tritt auf, wenn Sie bei der Initiierung des Failovers einen Access Point mit mehreren Regionen und nur einem AWS-Region ausgewählt haben. Sie müssen zwei Regionen auswählen, bevor Sie den Failover einleiten können. Andernfalls wird ein Fehler zurückgegeben.

Konfigurieren einer Replikation zur Verwendung mit Multi-Region Access Points

Wenn Sie eine Anforderung an einen Endpunkt eines Multi-Region Access Point stellen, leitet Amazon S3 die Anforderung automatisch an den Ihnen nächstgelegenen Bucket weiter. Amazon S3 berücksichtigt die Inhalte der Anforderung bei dieser Entscheidung nicht. Wenn Sie eine Anforderung stellen, um ein Objekt mit `GET` abzurufen, wird Ihre Anforderung möglicherweise an einen Bucket weitergeleitet, der keine Kopie dieses Objekts besitzt. In diesem Fall erhalten Sie den HTTP-Statuscode 404 (Nicht gefunden). Weitere Informationen zur Weiterleitung von Multi-Region-Access-Point-Anforderungen finden Sie unter the section called “Weiterleitung von Anforderungen”.

Wenn Sie möchten, dass der Multi-Region Access Point das Objekt unabhängig davon wiederherstellen kann, welcher Bucket die Anforderung empfängt, müssen Sie die regionsübergreifende Replikation von Amazon S3 (CRR) konfigurieren.

Betrachten Sie einen Multi-Region Access Point mit drei Buckets:

- Ein Bucket mit dem Namen `my-bucket-usw2` in der Region `us-west-2`, der das Objekt `my-image.jpg` enthält
- Ein Bucket mit dem Namen `my-bucket-aps1` in der Region `ap-south-1`, der das Objekt `my-image.jpg` enthält
- Ein Bucket mit dem Namen `my-bucket-euc1` in der Region `eu-central-1`, der das Objekt `my-image.jpg` nicht enthält

Wenn Sie in dieser Situation eine `GetObject`-Anforderung für das Objekt `my-image.jpg` stellen, hängt der Erfolg dieser Anforderung davon ab, welcher Bucket Ihre Anforderung empfängt. Da Amazon S3 den Inhalt der Anforderung nicht berücksichtigt, kann es Ihre `GetObject`-Anforderung an den `my-bucket-euc1`-Bucket weiterleiten, wenn dieser Bucket aus nächster Nähe antwortet. Obwohl sich Ihr Objekt in einem Bucket im Multi-Region Access Point befindet, erhalten Sie einen HTTP-Fehler 404 Not Found, da der einzelne Bucket, der Ihre Anforderung empfangen hat, das Objekt nicht enthielt.

Durch das Aktivieren der regionsübergreifenden Replikationsfunktion (CRR) kann dieses Ergebnis vermieden werden. Mit den entsprechenden Replikationsregeln wird das `my-image.jpg`-Objekt in den `my-bucket-euc1`-Bucket kopiert. Wenn Amazon S3 Ihre Anforderung an diesen Bucket weiterleitet, können Sie das Objekt daher jetzt abrufen.

Die Replikation funktioniert normal mit Buckets, die einem Multi-Regions-Zugriffspunkt zugewiesen sind. Amazon S3 führt keine spezielle Replikationsbehandlung mit Buckets aus, die sich in Multi-

Region Access Points befinden. Weitere Informationen zum Konfigurieren einer Replikation in Ihren Buckets finden Sie unter Überblick über die Einrichtung der Live-Replikation.

Empfehlungen zur Verwendung der Replikationsfunktion mit Multi-Region Access Points

Für beste Replikationsleistung bei der Arbeit mit Multi-Region Access Points empfehlen wir Folgendes:

- Konfigurieren Sie die Begrenzung der S3-Replikationszeit (S3 RTC). Um Ihre Daten innerhalb eines vorhersehbaren Zeitraums über verschiedene Regionen hinweg zu replizieren, können Sie S3 RTC verwenden. S3 RTC repliziert 99,99 Prozent der neuen in Amazon S3 gespeicherten Objekte innerhalb von 15 Minuten (gestützt auf ein Service Level Agreement). Weitere Informationen finden Sie unter the section called “Verwenden der S3-Replikationszeitkontrolle”. Für S3 RTC fallen zusätzliche Gebühren an. Informationen finden Sie unter Amazon S3 – Preise.
- Verwenden Sie die bidirektionale Replikation, um die Synchronisierung von Buckets zu unterstützen, wenn Buckets über den Multi-Region Access Point aktualisiert werden. Weitere Informationen finden Sie unter the section called “Erstellen von bidirektionalen Replikationsregeln für Ihren Multi-Region Access Point”.
- Erstellen Sie kontoübergreifende Multi-Region Access Points für die Replikation von Daten in Buckets in separaten AWS-Konten. Dieses Vorgehen ermöglicht eine Trennung auf Kontoebene, sodass auf Daten von verschiedenen Konten in anderen Regionen als dem Quell-Bucket zugegriffen werden kann und diese repliziert werden können. Die Einrichtung von kontoübergreifenden Multi-Region Access Points ist ohne zusätzliche Kosten möglich. Wenn Sie Bucket-Eigentümer, jedoch nicht Eigentümer des Multi-Region Access Point sind, zahlen Sie nur für die Datenübertragung und die Anforderungskosten. Eigentümer von Multi-Region Access Points tragen die Kosten für die Datenweiterleitung und die Internetbeschleunigung. Weitere Informationen finden Sie unter Amazon S3 – Preise.
- Aktivieren Sie die Synchronisierung von Replikatänderungen für jede Replikationsregel, um auch Metadatenänderungen an Ihren Objekten zu synchronisieren. Weitere Informationen finden Sie unter Aktivieren der Synchronisierung von Replikatänderungen.
- Aktivieren Sie Amazon-CloudWatch-Metriken, um Replikationsereignisse zu überwachen. Es fallen Gebühren für CloudWatch-Metriken an. Weitere Informationen hierzu finden Sie unter Amazon CloudWatch – Preise.

Themen

- Erstellen von unidirektionalen Replikationsregeln für Ihren Multi-Region Access Point

- Erstellen von bidirektionalen Replikationsregeln für Ihren Multi-Region Access Point
- Anzeigen der Replikationsregeln für Ihren Multi-Region Access Point

Erstellen von unidirektionalen Replikationsregeln für Ihren Multi-Region Access Point

Replikationsregeln ermöglichen ein automatisches und asynchrones Kopieren von Objekten über Buckets hinweg. Eine unidirektionale Replikationsregel hilft sicherzustellen, dass Daten vollständig aus einem Quell-Bucket in einer AWS-Region in einen Ziel-Bucket in einer anderen Region repliziert werden. Wenn unidirektionale Replikation eingerichtet ist, wird eine Replikationsregel vom Quell-Bucket (DOC-EXAMPLE-BUCKET-1) zum Ziel-Bucket (DOC-EXAMPLE-BUCKET-2) erstellt. Wie alle Replikationsregeln können Sie die unidirektionale Replikationsregel entweder auf den gesamten Amazon-S3-Bucket oder auf eine anhand eines Präfixes oder von Objekt-Tags gefilterte Teilmenge von Objekten anwenden.

> **Important**
>
> Wir empfehlen eine unidirektionale Replikation, wenn Ihre Benutzer die Objekte in Ihren Ziel-Buckets nur verwenden werden. Wenn Ihre Benutzer die Objekte in Ihren Ziel-Buckets hochladen oder ändern, verwenden Sie die bidirektionale Replikation, damit alle Ihre Buckets synchronisiert bleiben. Wir empfehlen eine bidirektionale Replikation auch, wenn Sie Ihren Multi-Region Access Point für Failover verwenden möchten. Informationen zum Einrichten der bidirektionalen Replikation finden Sie unter the section called “Erstellen von bidirektionalen Replikationsregeln für Ihren Multi-Region Access Point”.

So erstellen Sie eine unidirektionale Replikationsregel für Ihren Multi-Region Access Point

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
3. Wählen Sie den Namen Ihres Multi-Region Access Point aus.
4. Wählen Sie die Registerkarte Replication and failover (Replikation und Failover) aus.
5. Scrollen Sie nach unten zum Abschnitt Replication rules (Replikationsregeln) und wählen Sie dann Create replication rules (Replikationsregeln erstellen) aus. Vergewissern Sie sich, dass Sie über ausreichende Berechtigungen verfügen, um die Replikationsregel zu erstellen. Andernfalls wird die Versionsverwaltung deaktiviert.

> **Note**
>
> Sie können nur für Buckets in Ihrem eigenen Konto Replikationsregeln erstellen. Replikationsregeln für externe Buckets müssen die Bucket-Eigentümer selbst erstellen.

6. Wählen Sie auf der Seite Replikationsregeln erstellen die Vorlage zum Replizieren von Objekten aus einem oder mehreren Quell-Buckets in einen oder mehrere Ziel-Buckets aus.

> **Important**
>
> Wenn Sie Replikationsregeln mithilfe dieser Vorlage erstellen, ersetzen diese alle vorhandenen Replikationsregeln, die dem Bucket bereits zugewiesen sind.
> Wenn Sie Replikationsregeln hinzufügen oder bestehende Regeln ändern möchten, anstatt sie zu ersetzen, wechseln Sie in der Konsole zur Registerkarte Management (Verwaltung) der einzelnen Buckets und bearbeiten Sie dann die Regeln im Abschnitt Replication rules (Replikationsregeln). Sie können bestehende Replikationsregeln auch mithilfe der AWS CLI, SDKs oder der REST-API hinzufügen oder ändern. Weitere Informationen finden Sie unter Elemente der Replikationskonfigurationsdatei.

7. Wählen Sie im Abschnitt Quelle und Ziel unter Quell-Buckets einen oder mehrere Buckets aus, aus dem/denen Sie Objekte replizieren möchten. Für alle für die Replikation ausgewählten Buckets (Quell- und Ziel-Buckets) muss die S3-Versionsverwaltung aktiviert sein und jeder Bucket muss sich in einer anderen AWS-Region befinden. Weitere Informationen zu S3 Versioning finden Sie unter Verwenden der Versionsverwaltung in Amazon-S3-Buckets.

   Wählen Sie unter Ziel-Buckets einen oder mehrere Buckets aus, in die Sie Objekte replizieren möchten.

> **Note**
>
> Stellen Sie sicher, dass Sie über die erforderlichen Lese- und Replikationsberechtigungen verfügen, um die Replikation einzurichten. Andernfalls treten Fehler auf. Weitere Informationen finden Sie unter Erstellen einer IAM-Rolle.

8. Wählen Sie im Abschnitt Replication rule configuration (Konfiguration der Replikationsregel) aus, ob die Replikationsregel bei ihrer Erstellung Enabled (Aktiviert) oder Disabled (Deaktiviert) sein soll.

> **Note**
>
> Sie können keinen Namen in das Feld Replication rule name (Name der Replikationsregel) eingeben. Die Namen der Replikationsregeln werden basierend auf Ihrer Konfiguration generiert, wenn Sie die Replikationsregel erstellen.

9. Wählen Sie im Abschnitt Scope (Bereich) den entsprechenden Bereich für Ihre Replikation aus.

   - Um den gesamten Bucket zu replizieren, wählen Sie Apply to all objects in the bucket (Auf alle Objekte im Bucket anwenden).
   - Um eine Teilmenge der Objekte im Bucket zu replizieren, wählen Sie Limit the scope of this rule using one or more filters (Geltungsbereich dieser Regel mit einem oder mehreren Filtern einschränken) aus.

     Sie können Ihre Objekte nach Präfix, Objekt-Tags oder einer Kombination aus beiden filtern.

     - Wenn Sie die Replikation auf alle Objekte beschränken möchten, die mit derselben Zeichenfolge beginnen (z. B. `pictures`), geben Sie ein Präfix in das Feld Prefix (Präfix) ein.

       Wenn Sie ein Präfix eingeben, bei dem es sich um den Namen eines Ordners handelt, müssen Sie ein Trennzeichen wie z. B. `/` (Schrägstrich) verwenden, um die Hierarchieebene des Ordners anzugeben (z. B. `pictures/`). Weitere Informationen zu Präfixen finden Sie unter Organisieren von Objekten mit Präfixen.
     - Um alle Objekte mit einem oder mehreren Objekt-Tags zu replizieren, wählen Sie Add tag (Tag hinzufügen) aus und geben Sie das Schlüssel-Wert-Paar in die Felder ein. Wiederholen Sie den Vorgang, um ein weiteres Tag hinzuzufügen. Weitere Informationen über Objekt-Markierungen finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen.

10. Scrollen Sie nach unten zum Abschnitt Additional replication options (Zusätzliche Replikationsoptionen) und wählen Sie die Replikationsoptionen aus, die Sie anwenden möchten.

> **Note**
>
> Wir empfehlen Ihnen, folgende Optionen anzuwenden:
>
> - Replication time control (RTC) Begrenzung der Replikationszeit (RTC) – Um Ihre Daten innerhalb eines vorhersehbaren Zeitraums über verschiedene Regionen

hinweg zu replizieren, können Sie die Begrenzung der S3-Replikationszeit (S3 RTC) verwenden. S3 RTC repliziert 99,99 Prozent der neuen in Amazon S3 gespeicherten Objekte innerhalb von 15 Minuten (gestützt auf ein Service Level Agreement). Weitere Informationen finden Sie unter the section called “Verwenden der S3-Replikationszeitkontrolle”.

- Replication metrics and notifications (Replikationsmetriken und -benachrichtigungen) – Aktivieren Sie Amazon-CloudWatch-Metriken, um Replikationsereignisse zu überwachen.
- Replikation der Löschmarkierung – Durch S3-Löschvorgänge erstellte Löschmarkierungen werden repliziert. Durch Lebenszyklusregeln erstellte Löschmarkierungen werden nicht repliziert. Weitere Informationen finden Sie unter Replizieren von Löschmarkierungen auf Buckets.

Es fallen zusätzliche Gebühren für S3 RTC und CloudWatch-Replikationsmetriken und -benachrichtigungen an. Weitere Informationen finden Sie unter Amazon S3 – Preise und Amazon CloudWatch – Preise.

11. Wenn Sie eine neue Replikationsregel schreiben, die eine bestehende ersetzt, wählen Sie I acknowledge that by choosing Create replication rules, these existing replication rules will be overwritten (Ich bestätige, dass diese vorhandenen Replikationsregeln durch Auswählen von „Replikationsregeln erstellen“ überschrieben werden).
12. Wählen Sie Replikationsregeln erstellen aus, um Ihre neue unidirektionale Replikationsregel zu erstellen und zu speichern.

Erstellen von bidirektionalen Replikationsregeln für Ihren Multi-Region Access Point

Replikationsregeln ermöglichen ein automatisches und asynchrones Kopieren von Objekten über Buckets hinweg. Eine bidirektionale Replikationsregel stellt sicher, dass Daten zwischen zwei oder mehr Buckets in verschiedenen AWS-Regionen vollständig synchronisiert werden. Wenn die bidirektionale Replikation eingerichtet ist, wird eine Replikationsregel vom Quell-Bucket (DOC-EXAMPLE-BUCKET-1) zu dem Bucket, der die Replikate enthält (DOC-EXAMPLE-BUCKET-2) erstellt. Anschließend wird eine zweite Replikationsregel von dem Bucket, der die Replikate enthält (DOC-EXAMPLE-BUCKET-2), zum Quell-Bucket (DOC-EXAMPLE-BUCKET-1) erstellt.

Wie alle Replikationsregeln können Sie die bidirektionale Replikationsregel entweder auf den gesamten Amazon-S3-Bucket oder auf eine anhand eines Präfixes oder von Objekt-Tags gefilterte

Teilmenge von Objekten anwenden. Sie können auch Metadatenänderungen an Ihren Objekten u synchronisieren, indem Sie die Synchronisierung von Replikatänderungen für jede Replikationsregel aktivieren. Sie können die Synchronisierung von Replikatänderungen über die Amazon-S3-Konsole, die AWS CLI, die AWS-SDKs, die Amazon-S3-REST-API oder AWS CloudFormation aktivieren.

Um den Replikationsfortschritt von Objekten und Objektmetadaten in Amazon CloudWatch zu überwachen, aktivieren Sie S3-Replikationsmetriken und -benachrichtigungen. Weitere Informationen finden Sie unter Überwachen des Fortschritts mit Replikationsmetriken und Amazon-S3-Ereignisbenachrichtigungen.

So erstellen Sie eine bidirektionale Replikationsregel für Ihren Multi-Region Access Point

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
3. Wählen Sie den Namen des Multi-Region Access Point aus, den Sie aktualisieren möchten.
4. Wählen Sie die Registerkarte Replication and failover (Replikation und Failover) aus.
5. Scrollen Sie nach unten zum Abschnitt Replication rules (Replikationsregeln) und wählen Sie dann Create replication rules (Replikationsregeln erstellen) aus.
6. Wählen Sie auf der Seite Create replication rules (Replikationsregeln erstellen) die Vorlage Replicate objects among all specified buckets (Replizieren von Objekten zwischen allen angegebenen Buckets) aus. Die Vorlage Replicate objects among all specified buckets (Replizieren von Objekten zwischen allen angegebenen Buckets) richtet die bidirektionale Replikation (mit Failover-Funktionen) für Ihre Buckets ein.

> **Important**
>
> Wenn Sie Replikationsregeln mithilfe dieser Vorlage erstellen, ersetzen diese alle vorhandenen Replikationsregeln, die dem Bucket bereits zugewiesen sind.
> Wenn Sie Replikationsregeln hinzufügen oder bestehende Regeln ändern möchten, anstatt sie zu ersetzen, wechseln Sie in der Konsole zur Registerkarte Management (Verwaltung) der einzelnen Buckets und bearbeiten Sie dann die Regeln im Abschnitt Replication rules (Replikationsregeln). Sie können bestehende Replikationsregeln auch über die AWS CLI, AWS-SDKs oder die Amazon-S3-REST-API hinzufügen oder ändern. Weitere Informationen finden Sie unter Elemente der Replikationskonfigurationsdatei.

7. Wählen Sie im Abschnitt Buckets mindestens zwei Buckets aus, aus denen Sie Objekte replizieren möchten. Für alle für die Replikation ausgewählten Buckets muss S3 Versioning aktiviert sein und jeder Bucket muss sich in einer anderen AWS-Region befinden. Weitere Informationen zu S3 Versioning finden Sie unter Verwenden der Versionsverwaltung in Amazon-S3-Buckets.

   > **Note**
   >
   > Stellen Sie sicher, dass Sie über die erforderlichen Lese- und Replikationsberechtigungen verfügen, um die Replikation einzurichten. Andernfalls treten Fehler auf. Weitere Informationen finden Sie unter Erstellen einer IAM-Rolle.

8. Wählen Sie im Abschnitt Replication rule configuration (Konfiguration der Replikationsregel) aus, ob die Replikationsregel bei ihrer Erstellung Enabled (Aktiviert) oder Disabled (Deaktiviert) sein soll.

   > **Note**
   >
   > Sie können keinen Namen in das Feld Replication rule name (Name der Replikationsregel) eingeben. Die Namen der Replikationsregeln werden basierend auf Ihrer Konfiguration generiert, wenn Sie die Replikationsregel erstellen.

9. Wählen Sie im Abschnitt Scope (Bereich) den entsprechenden Bereich für Ihre Replikation aus.

   - Um den gesamten Bucket zu replizieren, wählen Sie Apply to all objects in the bucket (Auf alle Objekte im Bucket anwenden).
   - Um eine Teilmenge der Objekte im Bucket zu replizieren, wählen Sie Limit the scope of this rule using one or more filters (Geltungsbereich dieser Regel mit einem oder mehreren Filtern einschränken) aus.

     Sie können Ihre Objekte nach Präfix, Objekt-Tags oder einer Kombination aus beiden filtern.

     - Wenn Sie die Replikation auf alle Objekte beschränken möchten, die mit derselben Zeichenfolge beginnen (z. B. `pictures`), geben Sie ein Präfix in das Feld Prefix (Präfix) ein.

       Wenn Sie ein Präfix eingeben, das den Namen eines Ordners darstellt, müssen Sie `/` (Schrägstrich) als letztes Zeichen eingeben (z. B. `pictures/`).

- Um alle Objekte mit einem oder mehreren Objekt-Tags zu replizieren, wählen Sie Add tag (Tag hinzufügen) aus und geben Sie das Schlüssel-Wert-Paar in die Felder ein. Wiederholen Sie den Vorgang, um ein weiteres Tag hinzuzufügen. Weitere Informationen über Objekt-Markierungen finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen.

10. Scrollen Sie nach unten zum Abschnitt Additional replication options (Zusätzliche Replikationsoptionen) und wählen Sie die Replikationsoptionen aus, die Sie anwenden möchten.

> **Note**
>
> Wir empfehlen Ihnen, die folgenden Optionen anzuwenden, insbesondere wenn Sie Ihren Multi-Region Access Point so konfigurieren möchten, dass er Failover unterstützt:
>
> - Replication time control (RTC) Begrenzung der Replikationszeit (RTC) – Um Ihre Daten innerhalb eines vorhersehbaren Zeitraums über verschiedene Regionen hinweg zu replizieren, können Sie die Begrenzung der S3-Replikationszeit (S3 RTC) verwenden. S3 RTC repliziert 99,99 Prozent der neuen in Amazon S3 gespeicherten Objekte innerhalb von 15 Minuten (gestützt auf ein Service Level Agreement). Weitere Informationen finden Sie unter the section called “Verwenden der S3-Replikationszeitkontrolle”.
> - Replication metrics and notifications (Replikationsmetriken und -benachrichtigungen) – Aktivieren Sie Amazon-CloudWatch-Metriken, um Replikationsereignisse zu überwachen.
> - Replikation der Löschmarkierung – Durch S3-Löschvorgänge erstellte Löschmarkierungen werden repliziert. Durch Lebenszyklusregeln erstellte Löschmarkierungen werden nicht repliziert. Weitere Informationen finden Sie unter Replizieren von Löschmarkierungen auf Buckets.
> - Replica modification sync (Synchronisierung von Replikatänderungen) – Aktivieren Sie die Synchronisierung von Replikatänderungen für jede Replikationsregel, um auch Metadatenänderungen an Ihren Objekten zu synchronisieren. Weitere Informationen finden Sie unter Aktivieren der Synchronisierung von Replikatänderungen.
>
> Es fallen zusätzliche Gebühren für S3 RTC und CloudWatch-Replikationsmetriken und -benachrichtigungen an. Weitere Informationen finden Sie unter Amazon S3 – Preise und Amazon CloudWatch – Preise.

11. Wenn Sie eine neue Replikationsregel schreiben, die eine bestehende ersetzt, wählen Sie I acknowledge that by choosing Create replication rules, these existing replication rules will be overwritten (Ich bestätige, dass diese vorhandenen Replikationsregeln durch Auswählen von „Replikationsregeln erstellen“ überschrieben werden).
12. Wählen Sie Create replication rules (Replikationsregeln erstellen), um Ihre neuen bidirektionalen Replikationsregeln zu erstellen und zu speichern.

Anzeigen der Replikationsregeln für Ihren Multi-Region Access Point

Mit Multi-Region Access Points können Sie entweder unidirektionale oder bidirektionale Replikationsregeln einrichten. Informationen zur Verwaltung Ihrer Replikationsregeln finden Sie unter Verwalten von Replikationsregeln mit der Amazon-S3-Konsole.

So zeigen Sie die Replikationsregeln für Ihren Multi-Region Access Point an

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die Amazon-S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Multi-Region Access Points aus.
3. Wählen Sie den Namen Ihres Multi-Region Access Point aus.
4. Wählen Sie die Registerkarte Replication and failover (Replikation und Failover) aus.
5. Scrollen Sie nach unten bis zum Abschnitt Replikationsregeln. In diesem Abschnitt sind alle Replikationsregeln aufgeführt, die für Ihren Multi-Region Access Point erstellt wurden.

> **Note**
>
> Wenn Sie diesem Multi-Region Access Point einen Bucket von einem anderen Konto hinzugefügt haben, benötigen Sie die Berechtigung `s3:GetBucketReplication` vom Bucket-Eigentümer, um die Replikationsregeln für diesen Bucket anzeigen zu können.

Verwenden von Access Points mit mehreren Regionen mit unterstützten Vorgängen API

Amazon S3 unterstützt verschiedene Vorgänge, mit denen Sie Multi-Regions-Zugriffspunkte verwalten können. Amazon S3 verarbeitet einige dieser Vorgänge synchron und einige asynchron. Wenn Sie einen asynchronen Vorgang aufrufen, autorisiert Amazon S3 den angeforderten Vorgang

zunächst synchron. Wenn die Autorisierung erfolgreich ist, gibt Amazon S3 ein Token zurück, mit dem Sie den Fortschritt und die Ergebnisse des angeforderten Vorgangs verfolgen können.

> **Note**
>
> Anforderungen, die über die Amazon-S3-Konsole erfolgen, sind immer synchron. Die Konsole wartet, bis die Anforderung abgeschlossen ist, bevor Sie eine weitere Anforderung senden können.

Sie können den aktuellen Status und die Ergebnisse asynchroner Operationen über die Konsole oder über AWS CLI AWS SDKs, oder einsehen. `DescribeMultiRegionAccessPointOperation` REST API Amazon S3 stellt ein Tracking-Token in der Antwort auf einen asynchronen Vorgang bereit. Sie schließen dieses Tracking-Token als Argument für `DescribeMultiRegionAccessPointOperation` ein. Wenn Sie das Tracking-Token einschließen, gibt Amazon S3 den aktuellen Status und die Ergebnisse des angegebenen Vorgangs zurück, einschließlich etwaiger Fehler oder relevanter Ressourceninformationen. Amazon S3 führt `DescribeMultiRegionAccessPointOperation`-Vorgänge synchron durch.

Alle Anforderungen auf Steuerebene zum Erstellen oder Verwalten von Multi-Region Access Points müssen an die Region `US West (Oregon)` weitergeleitet werden. Für Anforderungen von Multi-Region Access Point auf Steuerebene müssen keine Regionen angegeben werden. Für die Failover-Steuerebene von Multi-Region Access Points muss die Anforderung an eine der fünf unterstützten Regionen weitergeleitet werden. Weitere Informationen zu Regionen, die von Access Points in mehreren Regionen unterstützt werden, finden Sie unter. Einschränkungen und Beschränkungen des Multi-Regions-Zugriffspunkts

Darüber hinaus müssen Sie dem Benutzer, der Rolle oder einer anderen AWS Identity and Access Management (IAM) Entität, die eine Anfrage zur Verwaltung eines Access Points mit mehreren Regionen stellt, die `s3:ListAllMyBuckets` Berechtigung erteilen.

Die folgenden Beispiele veranschaulichen, wie Multi-Region Access Points mit kompatiblen Operationen in Amazon S3 verwendet werden.

Themen

- Kompatibilität von Access Points für mehrere Regionen mit und AWS-ServicesAWS SDKs
- Kompatibilität von Multi-Region Access Points mit S3-Operationen
- Anzeigen Ihrer Weiterleitungskonfiguration für Multi-Region Access Points

- Aktualisieren der zugrunde liegende Bucket-Richtlinie von Amazon S3
- Aktualisieren einer Weiterleitungskonfiguration von Multi-Region Access Points
- Hinzufügen eines Objekts zu einem Bucket in Ihrem Multi-Region Access Point
- Abrufen von Objekten von Ihrem Multi-Region Access Point
- Auflisten von Objekten, die in einem Bucket gespeichert sind, der Ihrem Multi-Region Access Point zugrunde liegt
- Verwenden Sie einen URL mit mehreren Regionen vorsignierten Access Points
- Verwenden eines Buckets, der mit „Zahlung durch den Anforderer“ konfiguriert ist, mit Multi-Region Access Points

Kompatibilität von Access Points für mehrere Regionen mit und AWS-ServicesAWS SDKs

Um einen Multi-Region-Access Point mit Anwendungen zu verwenden, die einen Amazon S3 S3-Bucket-Namen benötigen, verwenden Sie den Amazon-Ressourcennamen (ARN) des Multi-Region-Access Points, wenn Sie Anfragen mit einem stellen. AWS SDK Informationen darüber, welche mit Access Points in mehreren Regionen kompatibel AWS SDKs sind, finden Sie unter Kompatibilität mit. AWS SDKs

Kompatibilität von Multi-Region Access Points mit S3-Operationen

Sie können die folgenden Amazon S3 API S3-Datenebenenoperationen verwenden, um Aktionen an Objekten in Buckets durchzuführen, die Ihrem Multi-Region Access Point zugeordnet sind. Die folgenden S3-Operationen können Multi-Region-Access Points akzeptieren: ARNs

- AbortMultipartUpload
- CompleteMultipartUpload
- CreateMultipartUpload
- DeleteObject
- DeleteObjectTagging
- GetObject
- GetObjectAcl
- GetObjectLegalHold
- GetObjectRetention

- GetObjectTagging
- HeadObject
- ListMultipartUploads
- ListObjectsV2
- ListParts
- PutObject
- PutObjectAcl
- PutObjectLegalHold
- PutObjectRetention
- PutObjectTagging
- RestoreObject
- UploadPart

> **Note**
>
> Multi-Region-Access Points unterstützen Kopiervorgänge, bei denen Multi-Region-Access Points nur als Ziel verwendet werden, wenn der Multi-Region-Access Point verwendet wird. ARN

Sie können die folgenden Amazon-S3-Operationen auf Steuerebene verwenden, um Ihre Multi-Region Access Points zu erstellen und zu verwalten:

- CreateMultiRegionAccessPoint
- DescribeMultiRegionAccessPointOperation
- GetMultiRegionAccessPoint
- GetMultiRegionAccessPointPolicy
- GetMultiRegionAccessPointPolicyStatus
- GetMultiRegionAccessPointRoutes
- ListMultiRegionAccessPoints
- PutMultiRegionAccessPointPolicy

- SubmitMultiRegionAccessPointRoutes

Anzeigen Ihrer Weiterleitungskonfiguration für Multi-Region Access Points

AWS CLI

Mit dem folgenden Beispielbefehl wird Ihre Routenkonfiguration für Multi-Region Access Points abgerufen, sodass Sie die aktuellen Weiterleitungstatus für Ihre Buckets sehen können. Wenn Sie dieses Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control get-multi-region-access-point-routes
--region eu-west-1
--account-id 111122223333
--mrap arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap
```

SDK for Java

Im Folgenden wird SDK für Java-Code die Routenkonfiguration Ihres Access Points mit mehreren Regionen abgerufen, sodass Sie den aktuellen Routing-Status für Ihre Buckets einsehen können. Wenn Sie diese Beispielsyntax verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
S3ControlClient s3ControlClient = S3ControlClient.builder()
    .region(Region.US_EAST_1)
    .credentialsProvider(credentialsProvider)
    .build();

GetMultiRegionAccessPointRoutesRequest request =
 GetMultiRegionAccessPointRoutesRequest.builder()
    .accountId("111122223333")
    .mrap("arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap")
    .build();

GetMultiRegionAccessPointRoutesResponse response =
 s3ControlClient.getMultiRegionAccessPointRoutes(request);
```

SDK for JavaScript

Der folgende JavaScript Code ruft Ihre Routenkonfiguration SDK für Multi-Region Access Point ab, sodass Sie den aktuellen Routing-Status für Ihre Buckets sehen können. Wenn Sie diese

Beispielsyntax verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
const REGION = 'us-east-1'

const s3ControlClient = new S3ControlClient({
  region: REGION
})

export const run = async () => {
  try {
    const data = await s3ControlClient.send(
      new GetMultiRegionAccessPointRoutesCommand({
        AccountId: '111122223333',
        Mrap: 'arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap',
      })
    )
    console.log('Success', data)
    return data
  } catch (err) {
    console.log('Error', err)
  }
}

run()
```

SDK for Python

Der folgende SDK Python-Code ruft Ihre Multi-Region Access Point-Routenkonfiguration ab, sodass Sie den aktuellen Routing-Status für Ihre Buckets sehen können. Wenn Sie diese Beispielsyntax verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
s3.get_multi_region_access_point_routes(
        AccountId=111122223333,
        Mrap=arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap)['Routes']
```

Aktualisieren der zugrunde liegende Bucket-Richtlinie von Amazon S3

Damit ein ordnungsgemäßer Zugriff gewährt wird, müssen Sie auch die zugrunde liegende Bucket-Richtlinie von Amazon S3 aktualisieren. In den folgenden Beispielen wird die Zugriffskontrolle an

die Richtlinie für Multi-Region Access Points delegiert. Nachdem Sie die Zugriffskontrolle an die Richtlinie für Multi-Region Access Points delegiert haben, wird die Bucket-Richtlinie nicht mehr für die Zugriffskontrolle verwendet, wenn Anforderungen über den Multi-Region Access Point gestellt werden.

Hier ist ein Beispiel für eine Bucket-Richtlinie, die die Zugriffskontrolle an die Richtlinie der Multi-Region Access Points delegiert. Wenn Sie diese Bucket-Beispielrichtlinie verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre eigenen Informationen. Um diese Richtlinie mithilfe des AWS CLI `put-bucket-policy` Befehls anzuwenden, wie im nächsten Beispiel gezeigt, speichern Sie die Richtlinie beispielsweise in einer Datei. `policy.json`

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Principal": { "AWS": "*" },
    "Effect": "Allow",
    "Action": ["s3:*"],
    "Resource": ["arn:aws:s3:::111122223333/*", "arn:aws:s3:::amzn-s3-demo-bucket"],
    "Condition": {
      "StringEquals": {
        "s3:DataAccessPointAccount": "444455556666"
      }
    }
  }
}
```

Der folgende Beispielbefehl `put-bucket-policy` verknüpft die aktualisierte S3-Bucket-Richtlinie mit Ihrem S3-Bucket:

```
aws s3api put-bucket-policy
  --bucket amzn-s3-demo-bucket
  --policy file:///tmp/policy.json
```

Aktualisieren einer Weiterleitungskonfiguration von Multi-Region Access Points

Mit dem folgenden Beispielbefehl wird die Weiterleitungskonfiguration des Multi-Region Access Points aktualisiert. Weiterleitungsbefehle für den Multi-Region Access Point können in einer dieser fünf Regionen ausgeführt werden:

- `ap-southeast-2`
- `ap-northeast-1`

- us-east-1
- us-west-2
- eu-west-1

In einer Weiterleitungskonfiguration für den Multi-Region Access Point können Sie Buckets auf einen aktiven oder passiven Weiterleitungsstatus festlegen. Aktive Buckets empfangen Traffic, passive Buckets nicht. Sie können den Weiterleitungsstatus eines Buckets festlegen, indem Sie den Wert `TrafficDialPercentage` für den Bucket auf `100` für aktiv oder `0` für passiv einstellen.

AWS CLI

Mit dem folgenden Beispielbefehl wird die Weiterleitungskonfiguration Ihres Multi-Region Access Points aktualisiert. In diesem Beispiel ist *amzn-s3-demo-bucket1* auf den aktiven Status und *amzn-s3-demo-bucket2* auf den passiven Status eingestellt. Wenn Sie dieses Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control submit-multi-region-access-point-routes
--region ap-southeast-2
--account-id 111122223333
--mrap arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap
--route-updates Bucket=amzn-s3-demo-bucket1,TrafficDialPercentage=100
                Bucket=amzn-s3-demo-bucket2,TrafficDialPercentage=0
```

SDK for Java

Im Folgenden SDK für Java-Code wird Ihre Routing-Konfiguration für multiregionale Access Points aktualisiert. Wenn Sie diese Beispielsyntax verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
S3ControlClient s3ControlClient = S3ControlClient.builder()
    .region(Region.ap-southeast-2)
    .credentialsProvider(credentialsProvider)
    .build();

SubmitMultiRegionAccessPointRoutesRequest request =
 SubmitMultiRegionAccessPointRoutesRequest.builder()
    .accountId("111122223333")
    .mrap("arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap")
    .routeUpdates(
        MultiRegionAccessPointRoute.builder()
```

```
            .region("eu-west-1")
            .trafficDialPercentage(100)
            .build(),
        MultiRegionAccessPointRoute.builder()
            .region("ca-central-1")
            .bucket("111122223333")
            .trafficDialPercentage(0)
            .build()
    )
    .build();

SubmitMultiRegionAccessPointRoutesResponse response =
 s3ControlClient.submitMultiRegionAccessPointRoutes(request);
```

SDK for JavaScript

Der folgende JavaScript Code aktualisiert Ihre Routenkonfiguration SDK für multiregionale Access Points. Wenn Sie diese Beispielsyntax verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
const REGION = 'ap-southeast-2'

const s3ControlClient = new S3ControlClient({
  region: REGION
})

export const run = async () => {
  try {
    const data = await s3ControlClient.send(
      new SubmitMultiRegionAccessPointRoutesCommand({
        AccountId: '111122223333',
        Mrap: 'arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap',
        RouteUpdates: [
          {
            Region: 'eu-west-1',
            TrafficDialPercentage: 100,
          },
          {
            Region: 'ca-central-1',
            Bucket: 'amzn-s3-demo-bucket1',
            TrafficDialPercentage: 0,
          },
        ],
```

```
      })
    )
    console.log('Success', data)
    return data
  } catch (err) {
    console.log('Error', err)
  }
}

run()
```

SDK for Python

Der folgende Python-Code aktualisiert Ihre Routenkonfiguration SDK für Multi-Region Access Point. Wenn Sie diese Beispielsyntax verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
s3.submit_multi_region_access_point_routes(
        AccountId=111122223333,
        Mrap=arn:aws:s3::111122223333:accesspoint/abcdef0123456.mrap,
        RouteUpdates= [{
            'Bucket': amzn-s3-demo-bucket,
            'Region': ap-southeast-2,
            'TrafficDialPercentage': 10
        }])
```

Hinzufügen eines Objekts zu einem Bucket in Ihrem Multi-Region Access Point

Um dem Bucket, der dem Multi-Region Access Point zugeordnet ist, ein Objekt hinzuzufügen, können Sie den PutObjectOperation. Aktivieren Sie die regionsübergreifende Replikation, um alle Buckets im Multi-Region Access Point synchron zu halten.

> **Note**
>
> Für diese Operation müssen Sie über die `s3:PutObject`-Berechtigung für den Multi-Region Access Point verfügen. Weitere Informationen zu den Berechtigungsanforderungen für Multi-Region Access Points finden Sie unter Berechtigungen.

AWS CLI

Das folgende Beispiel für Uploads von Datenebenenanfragen *example.txt* zum angegebenen Multi-Region-Access Point. Um dieses Beispiel zu verwenden, ersetzen Sie *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
aws s3api put-object --bucket
 arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap --key example.txt --
body example.txt
```

SDK for Java

```
S3Client s3Client = S3Client.builder()
        .build();

PutObjectRequest objectRequest = PutObjectRequest.builder()
        .bucket("arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap")
        .key("example.txt")
        .build();

s3Client.putObject(objectRequest, RequestBody.fromString("Hello S3!"));
```

SDK for JavaScript

```
const client = new S3Client({});

async function putObjectExample() {
    const command = new PutObjectCommand({
        Bucket: "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap",
        Key: "example.txt",
        Body: "Hello S3!",
    });

    try {
        const response = await client.send(command);
        console.log(response);
    } catch (err) {
        console.error(err);
    }
}
```

SDK for Python

```
import boto3

client = boto3.client('s3')
client.put_object(
    Bucket='arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap',
    Key='example.txt',
    Body='Hello S3!'
)
```

Abrufen von Objekten von Ihrem Multi-Region Access Point

Um Objekte vom Multi-Region Access Point abzurufen, können Sie GetObjectOperation.

> **Note**
>
> Um diesen API Vorgang verwenden zu können, benötigen Sie die `s3:GetObject` Genehmigung für den Multi-Region Access Point. Weitere Informationen zu den Berechtigungsanforderungen für Multi-Region Access Points finden Sie unter Berechtigungen.

AWS CLI

Im folgenden Beispiel wird eine Datenebenenanforderung abgerufen *`example.txt`* vom angegebenen Multi-Region-Access Point und lädt ihn herunter als *`downloaded_example.txt`*. Um dieses Beispiel zu verwenden, ersetzen Sie *`Platzhalter für Benutzereingaben`* mit Ihren eigenen Informationen.

```
aws s3api get-object --bucket
 arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap --
key example.txt downloaded_example.txt
```

SDK for Java

```
S3Client s3 = S3Client
   .builder()
   .build();
```

```
GetObjectRequest getObjectRequest = GetObjectRequest.builder()
    .bucket("arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap")
    .key("example.txt")
    .build();

s3Client.getObject(getObjectRequest);
```

SDK for JavaScript

```
const client = new S3Client({})

async function getObjectExample() {
    const command = new GetObjectCommand({
        Bucket: "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap",
        Key: "example.txt"
    });

    try {
        const response = await client.send(command);
        console.log(response);
    } catch (err) {
        console.error(err);
    }
}
```

SDK for Python

```
import boto3

client = boto3.client('s3')
client.get_object(
    Bucket='arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap',
    Key='example.txt'
)
```

Auflisten von Objekten, die in einem Bucket gespeichert sind, der Ihrem Multi-Region Access Point zugrunde liegt

Um eine Liste von Objekten zurückzugeben, die in einem Bucket gespeichert sind, der Ihrem Multi-Region Access Point zugrunde liegt, verwenden Sie ListObjectsV2Vorgang. Im folgenden

Beispielbefehl werden alle Objekte für den angegebenen Multi-Region-Access Point mithilfe von ARN für den Multi-Region-Access Point aufgelistet. In diesem Fall ist der Multi-Region-Zugriffspunkt: ARN

`arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap`

> **Note**
>
> Um diesen API Vorgang verwenden zu können, benötigen Sie die `s3:ListBucket` Berechtigung für den Multi-Region Access Point und den zugrunde liegenden Bucket. Weitere Informationen zu den Berechtigungsanforderungen für Multi-Region Access Points finden Sie unter Berechtigungen.

AWS CLI

Das folgende Beispiel für eine Datenebenenanforderung listet die Objekte in dem Bucket auf, der dem Multi-Region Access Point zugrunde liegt, der durch den spezifiziert wurde. ARN Um dieses Beispiel zu verwenden, ersetzen Sie *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

```
aws s3api list-objects-v2 --bucket
 arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap
```

SDK for Java

```
S3Client s3Client = S3Client.builder()
        .build();

String bucketName = "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap";

ListObjectsV2Request listObjectsRequest = ListObjectsV2Request
    .builder()
    .bucket(bucketName)
    .build();

 s3Client.listObjectsV2(listObjectsRequest);
```

SDK for JavaScript

```
const client = new S3Client({});
```

```
async function listObjectsExample() {
    const command = new ListObjectsV2Command({
        Bucket: "arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap",
    });

    try {
        const response = await client.send(command);
        console.log(response);
    } catch (err) {
        console.error(err);
    }
}
```

SDK for Python

```
import boto3

client = boto3.client('s3')
client.list_objects_v2(
    Bucket='arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap'
)
```

Verwenden Sie einen URL mit mehreren Regionen vorsignierten Access Points

Sie können einen vorsignierten verwenden, um einen URL zu generieren, der es anderen ermöglichtURL, über einen Amazon S3 S3-Zugriffspunkt für mehrere Regionen auf Ihre Amazon S3 S3-Buckets zuzugreifen. Wenn Sie ein vorsigniertes Objekt erstellenURL, verknüpfen Sie es mit einer bestimmten Objektaktion, z. B. einem S3-Upload (`PutObject`) oder einem S3-Download (). `GetObject` Sie können das vorsignierte Objekt teilenURL, und jeder, der Zugriff darauf hat, kann die darin eingebettete Aktion so ausführen, URL als wäre es der ursprüngliche signierende Benutzer.

Vorsignierte URLs haben ein Ablaufdatum. Wenn die Ablaufzeit erreicht ist, URL funktioniert das nicht mehr.

Bevor Sie S3 Multi-Region Access Points mit vorsignierter Version verwendenURLs, überprüfen Sie die AWS SDKKompatibilität mit dem SigV4A-Algorithmus. Stellen Sie sicher, dass Ihre SDK Version SigV4a als Signaturimplementierung unterstützt, die zum Signieren der globalen Anfragen verwendet wird. AWS-Region Weitere Informationen zur Verwendung von presigned URLs with Amazon S3 finden Sie unter Sharing objects by using URLs presigned.

Die folgenden Beispiele zeigen, wie Sie Multi-Region-Access Points mit vorsignierten Geräten verwenden können. URLs Um diese Beispiele zu verwenden, ersetzen Sie das *Platzhalter für Benutzereingaben* mit Ihren eigenen Informationen.

AWS CLI

```
aws s3 presign
 arn:aws:s3::123456789012:accesspoint/MultiRegionAccessPoint_alias/example-file.txt
```

SDK for Python

```
import logging
import boto3
from botocore.exceptions import ClientError

s3_client = boto3.client('s3',aws_access_key_id='xxx',aws_secret_access_key='xxx')
s3_client.generate_presigned_url(HttpMethod='PUT',ClientMethod="put_object",
 Params={'Bucket':'arn:aws:s3::123456789012:accesspoint/
abcdef0123456.mrap','Key':'example-file'})
```

SDK for Java

```
S3Presigner s3Presigner = S3Presigner.builder()
    .credentialsProvider(StsAssumeRoleCredentialsProvider.builder()
        .refreshRequest(assumeRole)
        .stsClient(stsClient)
        .build())
    .build();

GetObjectRequest getObjectRequest = GetObjectRequest.builder()
    .bucket("arn:aws:s3::123456789012:accesspoint/abcdef0123456.mrap")
    .key("example-file")
    .build();

GetObjectPresignRequest preSignedReq = GetObjectPresignRequest.builder()
    .getObjectRequest(getObjectRequest)
    .signatureDuration(Duration.ofMinutes(10))
    .build();

PresignedGetObjectRequest presignedGetObjectRequest =
 s3Presigner.presignGetObject(preSignedReq);
```

> **Note**
>
> Um SigV4a mit temporären Sicherheitsanmeldedaten zu verwenden — zum Beispiel bei der Verwendung von IAM Rollen — stellen Sie sicher, dass Sie die temporären Anmeldeinformationen von einem regionalen Endpunkt in AWS Security Token Service ()AWS STS und nicht von einem globalen Endpunkt anfordern. Wenn Sie den globalen Endpunkt für AWS STS (`sts.amazonaws.com`) verwenden, AWS STS werden temporäre Anmeldeinformationen von einem globalen Endpunkt generiert, was von Sig4A nicht unterstützt wird. Aus diesem Grund erhalten Sie eine Fehlermeldung. Um dieses Problem zu beheben, verwenden Sie einen der aufgelisteten regionalen Endpunkte für. AWS STS

Verwenden eines Buckets, der mit „Zahlung durch den Anforderer“ konfiguriert ist, mit Multi-Region Access Points

Wenn ein S3-Bucket, der Ihren Multi-Region Access Points zugeordnet ist, für die Verwendung von „Zahlung durch den Anforderer“ konfiguriert ist, trägt der Anforderer die Kosten für die Bucket-Anforderung, den Download und alle mit Multi-Region Access Points verbundenen Kosten. Weitere Informationen finden Sie unter Amazon S3 – Preise.

Hier sehen Sie ein Beispiel für eine Anforderung auf Datenebene an einen Multi-Region Access Point, der mit einem Bucket mit Zahlung durch den Anforderer verbunden ist.

AWS CLI

Um Objekte von einem Access Point mit mehreren Regionen herunterzuladen, der mit einem Bucket verbunden ist, müssen Sie dies im `--request-payer requester` Rahmen Ihres get-object request. Sie müssen auch den Namen der Datei im Bucket sowie den Ort angeben, an dem sie gespeichert werden soll.

```
aws s3api get-object --bucket MultiRegionAccessPoint_ARN --request-payer requester
 --key example-file-in-bucket.txt example-location-of-downloaded-file.txt
```

SDK for Java

Wenn Sie Objekte von einem Multi-Region Access Point herunterladen möchten, der mit einem Bucket mit Zahlung durch den Anforderer verbunden ist, müssen Sie `RequestPayer.REQUESTER` als Teil Ihrer Anforderung `GetObject` angeben. Sie müssen auch den Namen der Datei in dem Bucket sowie den Ort angeben, an dem sie gespeichert werden soll.

```
GetObjectResponse getObjectResponse = s3Client.getObject(GetObjectRequest.builder()
    .key("example-file.txt")
    .bucket("arn:aws:s3::
123456789012:accesspoint/abcdef0123456.mrap")
    .requestPayer(RequestPayer.REQUESTER)
    .build()
).response();
```

Überwachung und Protokollierung von Anforderungen, die über einen Multi-Regions-Zugriffspunkt an zugrunde liegende Ressourcen erfolgen

Amazon S3 protokolliert Anforderungen, die über Multi-Region Access Points gestellt werden, und Anforderungen an die API-Operationen, die sie verwalten, wie z. B. `CreateMultiRegionAccessPoint` und `GetMultiRegionAccessPointPolicy`. Anforderungen, die über einen Multi-Regions-Zugriffspunkt an Amazon S3 gesendet werden, werden in den Zugriffsprotokollen und AWS CloudTrail-Protokollen des Amazon S3-Servers mit dem Hostnamen des Multi-Regions-Zugriffspunkts angezeigt. Der Hostname eines Zugriffspunkts hat das Format *`MRAP_alias`*`.accesspoint.s3-global.amazonaws.com`. Angenommen, Sie haben die folgende Bucket- und Multi-Regions-Zugriffspunktkonfiguration:

- Ein Bucket mit dem Namen `my-bucket-usw2` in der Region `us-west-2`, der das Objekt `my-image.jpg` enthält.
- Ein Bucket mit dem Namen `my-bucket-aps1` in der Region `ap-south-1`, der das Objekt `my-image.jpg` enthält.
- Ein Bucket mit dem Namen `my-bucket-euc1` in der Region `eu-central-1`, der kein Objekt namens `my-image.jpg` enthält.
- Ein Multi-Regions-Zugriffspunkt namens `my-mrap` mit dem Alias `mfzwi23gnjvgw.mrap`, der so konfiguriert ist, dass Anforderungen aus allen drei Buckets erfüllt werden.
- Ihre AWS-Konto-ID ist `123456789012`.

Eine Anforderung, die gesendet wird, um `my-image.jpg` direkt über einen der Buckets anzurufen, wird in Ihren Protokollen mit dem Hostnamen *`bucket_name`*`.s3.`*`Region`*`.amazonaws.com` angezeigt.

Wenn Sie die Anforderung stattdessen über den Multi-Region Access Point stellen, bestimmt Amazon S3 zunächst, welcher der Buckets in den verschiedenen Regionen am nächsten gelegen ist.

Nach der Ermittlung, welcher Bucket zur Erfüllung der Anforderung verwendet werden soll, sendet Amazon S3 die Anforderung an diesen Bucket und protokolliert den Vorgang unter Verwendung des Hostnamens des Multi-Region Access Point. Wenn Amazon S3 in diesem Beispiel die Anforderung an `my-bucket-aps1` weiterleitet, spiegeln Ihre Protokolle eine erfolgreiche `GET`-Anforderung für `my-image.jpg` von `my-bucket-aps1` unter Verwendung des Hostnamens `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com` wider.

> **Important**
>
> Multi-Region Access Points kennen den Dateninhalt der zugrunde liegenden Buckets nicht. Daher kann es sein, dass der Bucket, der die Anforderung erhält, die angeforderten Daten nicht enthält. Wenn Amazon S3 beispielsweise feststellt, dass der Bucket `my-bucket-euc1` am nächsten gelegen ist, spiegeln Ihre Protokolle eine fehlgeschlagene `GET`-Anforderung für `my-image.jpg` von `my-bucket-euc1` unter Verwendung des Hostnamens `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com` wider. Wenn die Anforderung stattdessen an `my-bucket-usw2` weitergeleitet worden wäre, würden Ihre Protokolle eine erfolgreiche `GET`-Anforderung anzeigen.

Weitere Informationen zu Amazon S3-Server-Zugriffsprotokollen finden Sie unter Protokollieren von Anfragen mit Server-Zugriffsprotokollierung. Weitere Informationen zu AWS CloudTrail, finden Sie unter Was ist AWS CloudTrail? im AWS CloudTrail-Benutzerhandbuch.

Überwachen und Protokollieren von Anforderungen an Verwaltungs-API-Operationen von Multi-Region Access Points

Amazon S3 bietet verschiedene API-Operationen für die Verwaltung von Multi-Region Access Points, wie z. B. `CreateMultiRegionAccessPoint` und `GetMultiRegionAccessPointPolicy`. Wenn Sie Anforderungen an diese API-Operationen über die AWS Command Line Interface (AWS CLI), AWS-SDKs oder die Amazon-S3-REST-API stellen, verarbeitet Amazon S3 diese Anforderungen asynchron. Vorausgesetzt, Sie verfügen über die entsprechenden Berechtigungen für die Anforderung, gibt Amazon S3 ein Token für diese Anforderungen zurück. Sie können dieses Token mit `DescribeAsyncOperation` verwenden, um Ihnen zu helfen, den Status von laufenden asynchronen Vorgängen anzuzeigen. Amazon S3 verarbeitet `DescribeAsyncOperation`-Anforderungen synchron. Sie können die Amazon-S3-Konsole, die AWS CLI, SDKs oder die REST-API verwenden, um den Status asynchroner Anforderungen anzuzeigen.

> **Note**
>
> Die Konsole zeigt nur den Status asynchroner Anforderungen an, die innerhalb der letzten 14 Tage gestellt wurden. Verwende Sie AWS CLI, SDKs oder REST-API, um den Status älterer Anforderungen anzuzeigen.

Asynchrone Verwaltungsvorgänge können einen von mehreren Zuständen aufweisen:

NEW

Amazon S3 hat die Anforderung erhalten und bereitet die Ausführung des Vorgangs vor.

IN_PROGRESS

Amazon S3 führt derzeit den Vorgang aus.

SUCCESS

Der Vorgang war erfolgreich. Die Antwort enthält relevante Informationen, z. B. den Multi-Regions-Zugriffspunkt-Alias für eine `CreateMultiRegionAccessPoint`-Anforderung.

FAILED

Der Vorgang schlägt fehl. Die Antwort enthält eine Fehlermeldung, die den Grund für den Anforderungsfehler angibt.

Verwenden von AWS CloudTrail mit Multi-Region Access Points

Mit AWS CloudTrail können Sie Kontoaktivitäten in Ihrer AWS-Infrastruktur anzeigen, suchen, herunterladen, archivieren, analysieren und auf diese reagieren. Mit Multi-Region Access Points und CloudTrail-Protokollierung können Sie Folgendes ermitteln:

- Wer oder was welche Maßnahme ergriffen hat
- Welche Ressourcen in Anspruch genommen wurden
- Wann das Ereignis aufgetreten ist
- Weitere Details zu dem Ereignis

Sie können diese Protokollinformationen für die Analyse und Reaktion auf Aktivitäten verwenden, die über Ihre Multi-Region Access Points erfolgt sind.

Einrichten von AWS CloudTrail-Multi-Regions-Zugriffspunkten

Um die CloudTrail-Protokollierung für Vorgänge im Zusammenhang mit dem Erstellen oder Verwalten von Multi-Region Access Points zu aktivieren, müssen Sie die CloudTrail-Protokollierung so konfigurieren, dass die Ereignisse in der Region USA West (Oregon) aufgezeichnet werden. Sie müssen Ihre Protokollierungskonfiguration auf diese Weise einrichten, unabhängig davon, in welcher Region Sie sich befinden, wenn Sie die Anforderung stellen, oder welche Regionen der Multi-Region Access Point unterstützt. Alle Anforderungen, einen Multi-Region Access Point zu erstellen oder zu verwalten, werden über die Region USA West (Oregon) geleitet. Wir empfehlen Ihnen, diese Region entweder einem vorhandenen Trail hinzuzufügen oder einen neuen Trail zu erstellen, der diese Region und alle Regionen enthält, die mit dem Multi-Region Access Point verknüpft sind.

Amazon S3 protokolliert alle Anforderungen, die über einen Multi-Region Access Point erfolgen, und Anforderungen an die API-Operationen, die Zugriffspunkte verwalten, z. B. `CreateMultiRegionAccessPoint` und `GetMultiRegionAccessPointPolicy`. Wenn Sie diese Anforderungen über einen Multi-Regions-Zugriffspunkt protokollieren, werden sie in Ihren AWS CloudTrail-Protokollen mit dem Hostnamen des Multi-Regions-Zugriffspunkts erscheinen. Wenn Sie beispielsweise Anforderungen an einen Bucket über einen Multi-Region Access Point mit dem Alias `mfzwi23gnjvgw.mrap` stellen, haben die Einträge im CloudTrail-Protokoll den Hostnamen `mfzwi23gnjvgw.mrap.accesspoint.s3-global.amazonaws.com`.

Multi-Region Access Points leiten Anforderungen an den nächstgelegenen Bucket weiter. Aufgrund dieses Verhaltens sind beim Betrachten der CloudTrail-Protokolle für einen Multi-Region Access Point Anforderungen an die zugrunde liegenden Buckets zu sehen. Einige dieser Anforderungen sind möglicherweise direkte Anforderungen an den Bucket, die nicht über den Multi-Region Access Point geleitet werden. Beachten Sie dies bei der Überprüfung des Datenverkehrs. Wenn sich ein Bucket in einem Multi-Regions-Zugriffspunkt befindet, können Anforderungen direkt an diesen Bucket gestellt werden, ohne den Multi-Regions-Zugriffspunkt zu durchlaufen.

Beim Erstellen und Verwalten von Multi-Regions-Zugriffspunkten sind asynchrone Ereignisse erforderlich. Asynchrone Anforderungen haben keine Abschlussereignisse im CloudTrail-Protokoll. Weitere Informationen zu asynchronen Anforderungen finden Sie unter Überwachen und Protokollieren von Anforderungen an Verwaltungs-API-Operationen von Multi-Region Access Points.

Weitere Informationen zu AWS CloudTrail, finden Sie unter Was ist AWS CloudTrail? im AWS CloudTrail-Benutzerhandbuch.

Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning

Das Versioning in Amazon S3 ermöglicht Ihnen, mehrere Versionen eines Objekts im selben Bucket aufzubewahren. Sie können die S3-Versioning-Funktion verwenden, um sämtliche Versionen aller Objekte in Ihren Buckets zu speichern, abzurufen oder wiederherzustellen. Daten lassen sich dank Versioning nach unbeabsichtigten Benutzeraktionen und Anwendungsfehlern leichter wiederherstellen. Wenn Sie das Versioning für einen Bucket aktivieren und Amazon S3 mehrere Schreibanforderungen für dasselbe Objekt gleichzeitig empfängt, werden alle Objekte gespeichert.

Versioning-fähige Buckets erlauben Ihnen, Objekte nach einem versehentlichen Löschen oder Überschreiben wiederherzustellen. Wenn Sie beispielsweise ein Objekt löschen, fügt Amazon S3 eine Löschmarkierung ein, anstatt das Objekt dauerhaft zu entfernen. Die Löschmarkierung wird zur aktuellen Objektversion. Wenn Sie ein Objekt überschreiben, entsteht eine neue Objektversion im Bucket. Sie können die vorherige Version immer wieder herstellen. Weitere Informationen finden Sie unter Löschen von Objekten aus einem versioning-fähigen Bucket.

Standardmäßig ist das S3-Versioning in Buckets deaktiviert und Sie müssen es explizit aktivieren. Weitere Informationen finden Sie unter Aktivieren des Versioning für Buckets.

> **Note**
>
> - Das unterstützt SOAP API keine S3-Versionierung. SOAPSupport Over HTTP ist veraltet, aber es ist immer noch verfügbar. HTTPS Neue Amazon S3 S3-Funktionen werden nicht unterstütztSOAP.
> - Für jede Version eines gespeicherten und übertragenen Objekts gelten die Standardpreise von Amazon S3. Jede Version eines Objekts besteht aus dem vollständigen Objekt, nicht nur aus einem Delta gegenüber der vorherigen Version. Wenn Sie also drei Versionen eines Objekts gespeichert haben, fallen Gebühren für die drei Objekte an.

Nicht versionierte, versionings-fähige und Buckets mit ausgesetztem Versioning

Buckets können einen von drei Zuständen aufweisen:

- Nicht versioniert (Standard)
- Versioningsfähig

- Ausgesetztes Versioning

Sie aktivieren und unterbrechen das Versioning auf Bucket-Ebene. Nachdem Sie einen Bucket versionsfähig gemacht haben, kann er nicht in einen nicht versionsfähigen Status zurückgesetzt werden. Sie können das Versioning für diesen Bucket jedoch aussetzen.

Der Versioning-Status gilt für alle (niemals für eine Untermenge) der Objekte in diesem Bucket. Wenn Sie die Versioning in einem Bucket aktivieren, werden alle neuen Objekte versioniert und mit einer eindeutigen Versions-ID versehen. Objekte, die zum Zeitpunkt der Aktivierung der Versioning im Bucket bereits im Bucket vorhanden waren, werden immer versioniert und erhalten eine eindeutige Versions-ID, wenn sie durch zukünftige Anforderungen geändert werden. Beachten Sie Folgendes:

- Objekte, die in Ihrem Bucket gespeichert waren, bevor Sie den Versioning-Status einrichten, haben die Versions-ID `null`. Wenn Sie das Versioning aktivieren, ändern sich die in Ihrem Bucket vorhandenen Objekte nicht mehr. Was sich ändert, ist, wie Amazon S3 die Objekte in zukünftigen Anfragen verarbeitet. Weitere Informationen finden Sie unter Arbeiten mit Objekten in einem versioning-fähigen Bucket.
- Der Bucket-Eigentümer (oder ein anderer Benutzer mit geeigneten Berechtigungen) kann das Versioning aussetzen, um zu verhindern, dass sich weitere Objektversionen ansammeln. Wenn Sie das Versioning aussetzen, ändern sich die in Ihrem Bucket vorhandenen Objekte nicht. Was sich ändert, ist, wie Amazon S3 die Objekte in zukünftigen Anfragen verarbeitet. Weitere Informationen finden Sie unter Arbeiten mit Objekten in einem Bucket mit ausgesetztem Versioning.

Verwenden des S3-Versioning mit dem S3-Lebenszyklus

Zum Anpassen Ihrer Datenaufbewahrungsmethode und zur Speicherkostenkontrolle verwenden Sie das Objekt-Versioning mit dem S3-Lebenszyklus. Weitere Informationen finden Sie unter Verwaltung des Lebenszyklus von Objekten. Informationen zum Erstellen von S3-Lifecycle-Konfigurationen mit dem AWS Management Console AWS CLI, AWS SDKs, oder dem REST API finden Sie unterEinstellung einer S3-Lifecycle-Konfiguration für einen Bucket.

> **Important**
>
> Wenn in dem nicht-versionierten Bucket eine Lebenszyklus-Konfiguration für den Ablauf von Objekten vorhanden ist und Sie dasselbe Verhalten hinsichtlich einer dauerhaften Löschung beim Aktivieren der Versionsverwaltung beibehalten möchten, müssen Sie eine Konfiguration für den Ablauf von nicht aktuellen Objekten hinzufügen. Die Konfiguration

für den Lebenszyklus mit nicht aktuellem Ablauf verwaltet die Löschungen der veralteten Objektversionen im Bucket mit aktivierter Versionierung. (Ein Bucket mit aktivierter Versionierung verwaltet eine aktuelle und keine oder mehrere nicht aktuelle Objektversionen.) Weitere Informationen finden Sie unter Einstellung einer S3-Lifecycle-Konfiguration für einen Bucket.

Informationen zum Arbeiten mit S3-Versioning finden Sie in den folgenden Themen.

Themen

- Wie funktioniert S3-Versioning
- Aktivieren des Versioning für Buckets
- MFALöschen konfigurieren
- Arbeiten mit Objekten in einem versioning-fähigen Bucket
- Arbeiten mit Objekten in einem Bucket mit ausgesetztem Versioning
- Problembehandlung bei der Versionierung

Wie funktioniert S3-Versioning

Sie können die S3-Versionsverwaltung verwenden, um mehrere Versionen eines Objekts in einem Bucket zu behalten und so Objekte wiederherzustellen, die versehentlich gelöscht oder überschrieben wurden. Wenn Sie beispielsweise die S3-Versionsverwaltung auf einen Bucket anwenden, werden die folgenden Änderungen vorgenommen:

- Wenn Sie ein Objekt löschen, fügt Amazon S3 eine Löschmarkierung hinzu, statt das Objekt dauerhaft zu entfernen. Die Löschmarkierung wird zur aktuellen Objektversion. Sie können dann die vorherige Version wiederherstellen. Weitere Informationen finden Sie unter Löschen von Objekten aus einem versioning-fähigen Bucket.
- Wenn Sie ein Objekt überschreiben, fügt Amazon S3 eine neue Objektversion im Bucket hinzu. Die vorherige Version verbleibt im Bucket und wird zu einer nicht aktuellen Version. Sie können die vorherige Version wiederherstellen.

> **Note**
>
> Für jede Version eines gespeicherten und übertragenen Objekts gelten die Standardpreise von Amazon S3. Jede Version eines Objekts besteht aus dem vollständigen Objekt, nicht aus einem Delta gegenüber der vorherigen Version. Wenn Sie also drei Versionen eines Objekts gespeichert haben, fallen Gebühren für die drei Objekte an.

Jedem von Ihnen erstellten S3-Bucket ist eine Versioning-Subressource zugeordnet. (Weitere Informationen finden Sie unter Optionen für die Bucket-Konfiguration.) Standardmäßig ist Ihr Bucket nicht versioning-fähig, und dementsprechend speichert die Versioning-Subressource eine leere Versioning-Konfiguration wie folgt.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</VersioningConfiguration>
```

Um die Versionsverwaltung zu aktivieren, können Sie eine Anfrage an Amazon S3 mit einer Versionskonfiguration senden, die einen `Enabled`-Status enthält.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
</VersioningConfiguration>
```

Um das Versioning auszusetzen, setzen Sie den Statuswert auf `Suspended`.

> **Note**
>
> Wenn Sie die Versionsverwaltung für einen Bucket zum ersten Mal aktivieren, kann es einen Moment dauern, bis die Änderung vollständig verbreitet ist. Während dieser Änderung können gelegentlich `HTTP 404 NoSuchKey` Fehler bei Anfragen an Objekte auftreten, die nach der Aktivierung der Versionierung erstellt oder aktualisiert wurden. Wir empfehlen, dass Sie nach dem Aktivieren der Versionsverwaltung 15 Minuten warten, bevor Sie Schreibvorgänge (`PUT` oder `DELETE`) für Objekte im Bucket ausführen.

Der Bucket-Besitzer und alle autorisierten AWS Identity and Access Management (IAM) Benutzer können die Versionierung aktivieren. Der Bucket-Besitzer ist derjenige AWS-Konto , der den

Bucket erstellt hat. Weitere Informationen zu Berechtigungen finden Sie unter Identity and Access Management für Amazon S3.

Weitere Informationen zum Aktivieren und Deaktivieren der S3-Versionierung mithilfe von AWS Management Console, AWS Command Line Interface (AWS CLI) oder REST API finden Sie unter. the section called “Aktivieren des Versioning für Buckets”

Themen

- Version IDs
- Versioning-Workflows

Version IDs

Wenn Sie das Versioning für einen Bucket aktivieren, generiert Amazon S3 automatisch eine eindeutige Versions-ID für das Objekt, das gespeichert wird. In einem Bucket können Sie beispielsweise zwei Objekte mit demselben Schlüssel (Objektname), aber unterschiedlicher Version habenIDs, z. B. `photo.gif` (Version 111111) und `photo.gif` (Version 121212).
Diagramm, das einen Bucket mit aktivierter Versionierung darstellt, der zwei Objekte mit demselben Schlüssel, aber unterschiedlicher Version enthält. IDs

Jedes Objekt hat eine Versions-ID, unabhängig davon, ob die S3-Versionierung aktiviert ist oder nicht. Wenn die S3-Versionsverwaltung nicht aktiviert ist, legt Amazon S3 den Wert der Versions-ID auf `null` fest. Wenn das S3-Versioning aktiviert ist, weist Amazon S3 dem Objekt einen Versions-ID-Wert zu. Dieser Wert unterscheidet dieses Objekt von anderen Versionen desselben Schlüssels.

Wenn Sie das S3-Versioning für einen vorhandenen Bucket aktivieren, bleiben Objekte, die bereits im Bucket gespeichert sind, unverändert. Ihre Version IDs (`null`), ihr Inhalt und ihre Berechtigungen bleiben unverändert. Nachdem Sie die S3-Versionsverwaltung aktiviert haben, erhält jedes Objekt, das dem Bucket hinzugefügt wird, eine Versions-ID, die sie von anderen Versionen desselben Schlüssels unterscheidet.

Nur Amazon S3 generiert VersionenIDs, und sie können nicht bearbeitet werden. Bei der Version IDs handelt es sich um Unicode-kodierte, UTF URL -8-kodierte, undurchsichtige Zeichenketten, die nicht länger als 1.024 Byte sind. Im Folgenden wird ein Beispiel gezeigt:

```
3sL4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo
```

> **Note**
>
> Der Einfachheit halber wird in den anderen Beispielen in diesem Thema viel kürzer verwendet. IDs

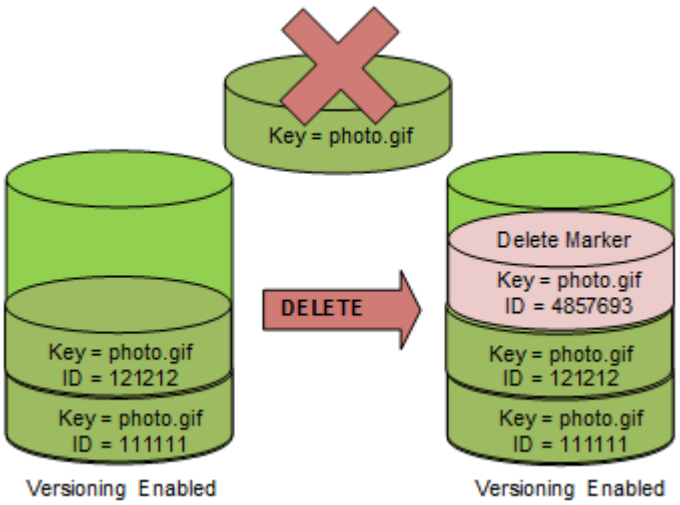
Versioning-Workflows

Wenn Sie mit PUT ein Objekt in einen Bucket mit aktiviertem Versioning schreiben, wird die nicht aktuelle Version nicht überschrieben. Wie in der folgenden Abbildung dargestellt, tritt das folgende Verhalten auf, wenn eine neue Version von `photo.gif` mit PUT in einen Bucket geschrieben wird, der bereits ein Objekt desselben Namens enthält:

- Das ursprüngliche Objekt (ID = 111111) verbleibt im Bucket.
- Amazon S3 generiert eine neue Versions-ID (121212) und fügt diese neuere Version des Objekts dem Bucket hinzu.

Mit dieser Funktion können Sie eine vorherige Version eines Objekts abrufen, wenn ein Objekt versehentlich überschrieben oder gelöscht wurde.

Wenn Sie DELETE für ein Objekt ausführen, bleiben alle Versionen in dem Bucket, und Amazon S3 fügt eine Löschmarkierung ein, wie in der folgenden Abbildung gezeigt.



Die Löschmarkierung wird zur aktuellen Version des Objekts. Standardmäßig wird durch GET-Anforderungen die zuletzt gespeicherte Version abgerufen. Eine `GET Object`-Anforderung gibt

einen `404 Not Found`-Fehler zurück, wenn die aktuelle Version eine Löschmarkierung ist, wie in der folgenden Abbildung gezeigt.

Sie können jedoch mit `GET` eine nicht aktuelle Version eines Objekts abrufen, indem Sie ihre Versions-ID angeben. In der folgenden Abbildung wird mit `GET` eine spezifische Objektversion abgerufen, 111111. Amazon S3 gibt diese Objektversion zurück, auch wenn es nicht die aktuelle Version ist.

Weitere Informationen finden Sie unter Abrufen von Objektversionen aus einem versioning-fähigen Bucket.

Sie können ein Objekt permanent löschen, indem Sie die Version angeben, die Sie löschen wollen. Nur der Besitzer eines Amazon S3 S3-Buckets oder ein autorisierter IAM Benutzer kann eine Version dauerhaft löschen. Wenn Ihre `DELETE`-Operation die `versionId` angibt, wird diese Objektversion dauerhaft gelöscht, und Amazon S3 fügt keine Löschmarkierung ein.

Sie können die Sicherheit erhöhen, indem Sie einen Bucket so konfigurieren, dass das Löschen mit Multi-Faktor-Authentifizierung (MFA) aktiviert wird. Wenn Sie das MFA Löschen für einen Bucket aktivieren, muss der Bucket-Besitzer in jeder Anfrage zwei Arten der Authentifizierung angeben, um eine Version zu löschen oder den Versionsstatus des Buckets zu ändern. Weitere Informationen finden Sie unter MFALöschen konfigurieren.

Wann werden neue Versionen für ein Objekt erstellt?

Neue Versionen werden nur erstellt, wenn Sie für ein neues Objekt `PUT` ausführen. Beachten Sie, dass bestimmte Aktionen wie `CopyObject` durch die Implementierung einer `PUT`-Operation funktionieren.

Einige Aktionen, die das aktuelle Objekt ändern, erstellen keine neue Version, da sie kein neues Objekt mit `PUT` bearbeiten. Dies umfasst Aktionen wie das Ändern der Markierungen für ein Objekt.

> **Important**
>
> Wenn Sie einen deutlichen Anstieg der Anzahl von HTTP 503 Antworten (Service Unavailable) für Amazon S3 `PUT` oder `DELETE` Objektanfragen an einen Bucket mit aktivierter S3-Versionierung feststellen, haben Sie möglicherweise ein oder mehrere Objekte im Bucket, für die es Millionen von Versionen gibt. Weitere Informationen finden Sie im Abschnitt zur S3-Versionsverwaltung unter Problembehandlung bei der Versionierung.

Aktivieren des Versioning für Buckets

Sie können Amazon-S3-Versioning nutzen, um mehrere Versionen eines Objekts in einem Bucket aufzubewahren. Dieser Abschnitt enthält Beispiele dafür, wie die Versionierung für einen Bucket mithilfe der Konsole, REST API AWS SDKs, und AWS Command Line Interface (AWS CLI) aktiviert wird.

> **Note**
>
> Wenn Sie die Versionsverwaltung für einen Bucket zum ersten Mal aktivieren, kann es einen Moment dauern, bis die Änderung vollständig verbreitet ist. Während diese Änderung propagiert wird, können gelegentlich `HTTP 404 NoSuchKey` Fehler bei Anfragen an Objekte auftreten, die nach der Aktivierung der Versionierung erstellt oder aktualisiert wurden. Wir empfehlen, dass Sie nach dem Aktivieren der Versionsverwaltung 15 Minuten warten, bevor Sie Schreibvorgänge (`PUT` oder `DELETE`) für Objekte im Bucket ausführen.

Weitere Informationen über das S3-Versioning finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning. Informationen zum Arbeiten mit Objekten, die sich in einem versioning-fähigen Bucket befinden, finden Sie unter Arbeiten mit Objekten in einem versioning-fähigen Bucket.

Weitere Informationen zur Verwendung von S3 Versioning zum Schutz von Daten finden Sie im Tutorial: Schutz von Daten in Amazon S3 vor versehentlichem Löschen oder Anwendungsfehlern mithilfe von S3 Versioning, S3 Object Lock und S3 Replication.

Jedem von Ihnen erstellten S3-Bucket ist eine Versioning-Subressource zugeordnet. (Weitere Informationen finden Sie unter Optionen für die Bucket-Konfiguration.) Standardmäßig ist Ihr Bucket nicht versioning-fähig, und dementsprechend speichert die Versioning-Subressource eine leere Versioning-Konfiguration wie folgt.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</VersioningConfiguration>
```

Um das Versioning zu aktivieren, können Sie eine Anfrage an Amazon S3 mit einer Versionskonfiguration senden, die einen Status enthält.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
```

```
</VersioningConfiguration>
```

Um das Versioning auszusetzen, setzen Sie den Statuswert auf `Suspended`.

Der Bucket-Eigentümer und alle autorisierten Benutzer können das Versioning aktivieren. Der Bucket-Besitzer ist derjenige AWS-Konto , der den Bucket erstellt hat (das Root-Konto). Weitere Informationen zu Berechtigungen finden Sie unter Identity and Access Management für Amazon S3.

In den folgenden Abschnitten finden Sie weitere Informationen zur Aktivierung der S3-Versionierung mithilfe der Konsole AWS CLI, und der AWS SDKs.

Verwenden der S3-Konsole

Gehen Sie wie folgt vor, AWS Management Console um die Versionierung in einem S3-Bucket zu aktivieren.

Aktivieren und Deaktivieren des Versionings für einen S3-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie Versioning aktivieren möchten.
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie unter Bucket Versioning (Bucket-Versioning) die Option Edit (Bearbeiten).
5. Wählen Sie Suspend (Anhalten) oder Enable (Aktivieren) und dann Save changes (Änderungen speichern).

> **Note**
>
> Sie können die AWS Multi-Faktor-Authentifizierung (MFA) mit Versionierung verwenden. Wenn Sie die Option MFA mit Versionierung verwenden, müssen Sie Ihre Zugangsschlüssel und einen gültigen Code vom MFA Gerät AWS-Konto des Kontos angeben, um eine Objektversion dauerhaft zu löschen oder die Versionierung auszusetzen oder zu reaktivieren. Für die Verwendung MFA mit Versionierung aktivieren Sie. `MFA Delete` Sie können jedoch `MFA Delete` nicht mit der AWS Management Console aktivieren. Sie müssen das AWS Command Line Interface (AWS CLI) oder das API verwenden. Weitere Informationen finden Sie unter MFALöschen konfigurieren.

Mit dem AWS CLI

Im folgenden Beispiel wird das Versioning auf einem S3-Bucket aktiviert.

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket1 --versioning-
configuration Status=Enabled
```

Das folgende Beispiel aktiviert die S3-Versionierung und das Löschen mit Multi-Faktor-Authentifizierung (MFA) für einen Bucket.

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket1 --versioning-
configuration Status=Enabled,MFADelete=Enabled --mfa "SERIAL 123456"
```

> **Note**
>
> Für die Verwendung von MFA delete ist ein zugelassenes physisches oder virtuelles Authentifizierungsgerät erforderlich. Weitere Informationen zur Verwendung von MFA Delete in Amazon S3 finden Sie unterMFALöschen konfigurieren.

Weitere Informationen zur Aktivierung der AWS CLI Versionierung mithilfe von finden Sie put-bucket-versioningin der AWS CLI Befehlsreferenz.

Verwenden von AWS SDKs

In den folgenden Beispielen wird die Versionsverwaltung für einen Bucket aktiviert und anschließend der Versionsstatus mithilfe von AWS SDK for Java und abgerufen. AWS SDK for .NET Informationen zur Verwendung von anderen AWS SDKs finden Sie im AWS Developer Center.

.NET

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem AWS SDK for. NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
using System;
using Amazon.S3;
using Amazon.S3.Model;
```

```
namespace s3.amazon.com.docsamples
{
    class BucketVersioningConfiguration
    {
        static string bucketName = "*** bucket name ***";

        public static void Main(string[] args)
        {
            using (var client = new AmazonS3Client(Amazon.RegionEndpoint.USEast1))
            {
                try
                {
                    EnableVersioningOnBucket(client);
                    string bucketVersioningStatus =
 RetrieveBucketVersioningConfiguration(client);
                }
                catch (AmazonS3Exception amazonS3Exception)
                {
                    if (amazonS3Exception.ErrorCode != null &&
                        (amazonS3Exception.ErrorCode.Equals("InvalidAccessKeyId")
                        ||
                        amazonS3Exception.ErrorCode.Equals("InvalidSecurity")))
                    {
                        Console.WriteLine("Check the provided AWS Credentials.");
                        Console.WriteLine(
                        "To sign up for service, go to http://aws.amazon.com/s3");
                    }
                    else
                    {
                        Console.WriteLine(
                         "Error occurred. Message:'{0}' when listing objects",
                         amazonS3Exception.Message);
                    }
                }
            }

            Console.WriteLine("Press any key to continue...");
            Console.ReadKey();
        }

        static void EnableVersioningOnBucket(IAmazonS3 client)
        {
```

```
                PutBucketVersioningRequest request = new PutBucketVersioningRequest
                {
                    BucketName = bucketName,
                    VersioningConfig = new S3BucketVersioningConfig
                    {
                        Status = VersionStatus.Enabled
                    }
                };

                PutBucketVersioningResponse response =
 client.PutBucketVersioning(request);
        }


        static string RetrieveBucketVersioningConfiguration(IAmazonS3 client)
        {
                GetBucketVersioningRequest request = new GetBucketVersioningRequest
                {
                    BucketName = bucketName
                };

                GetBucketVersioningResponse response =
 client.GetBucketVersioning(request);
                return response.VersioningConfig.Status;
            }
    }
}
```

Java

Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
import java.io.IOException;

import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Region;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3Client;
import com.amazonaws.services.s3.model.AmazonS3Exception;
import com.amazonaws.services.s3.model.BucketVersioningConfiguration;
import com.amazonaws.services.s3.model.SetBucketVersioningConfigurationRequest;
```

```
public class BucketVersioningConfigurationExample {
    public static String bucketName = "*** bucket name ***";
    public static AmazonS3Client s3Client;

    public static void main(String[] args) throws IOException {
        s3Client = new AmazonS3Client(new ProfileCredentialsProvider());
        s3Client.setRegion(Region.getRegion(Regions.US_EAST_1));
        try {

            // 1. Enable versioning on the bucket.
         BucketVersioningConfiguration configuration =
           new BucketVersioningConfiguration().withStatus("Enabled");

   SetBucketVersioningConfigurationRequest setBucketVersioningConfigurationRequest
 =
     new SetBucketVersioningConfigurationRequest(bucketName,configuration);


 s3Client.setBucketVersioningConfiguration(setBucketVersioningConfigurationRequest);

   // 2. Get bucket versioning configuration information.
   BucketVersioningConfiguration conf =
 s3Client.getBucketVersioningConfiguration(bucketName);
    System.out.println("bucket versioning configuration status:    " +
 conf.getStatus());

        } catch (AmazonS3Exception amazonS3Exception) {
            System.out.format("An Amazon S3 error occurred. Exception: %s",
 amazonS3Exception.toString());
        } catch (Exception ex) {
            System.out.format("Exception: %s", ex.toString());
        }
    }
}
```

Python

Das folgende Python-Beispiel erstellt einen Amazon-S3-Bucket, aktiviert ihn für das Versioning und konfiguriert einen Lebenszyklus, der nicht-aktuelle Objektversionen nach 7 Tagen ablaufen lässt.

```
def create_versioned_bucket(bucket_name, prefix):
    """
```

```
    Creates an Amazon S3 bucket, enables it for versioning, and configures a
lifecycle
    that expires noncurrent object versions after 7 days.

    Adding a lifecycle configuration to a versioned bucket is a best practice.
    It helps prevent objects in the bucket from accumulating a large number of
    noncurrent versions, which can slow down request performance.

    Usage is shown in the usage_demo_single_object function at the end of this
module.

    :param bucket_name: The name of the bucket to create.
    :param prefix: Identifies which objects are automatically expired under the
                   configured lifecycle rules.
    :return: The newly created bucket.
    """
    try:
        bucket = s3.create_bucket(
            Bucket=bucket_name,
            CreateBucketConfiguration={
                "LocationConstraint": s3.meta.client.meta.region_name
            },
        )
        logger.info("Created bucket %s.", bucket.name)
    except ClientError as error:
        if error.response["Error"]["Code"] == "BucketAlreadyOwnedByYou":
            logger.warning("Bucket %s already exists! Using it.", bucket_name)
            bucket = s3.Bucket(bucket_name)
        else:
            logger.exception("Couldn't create bucket %s.", bucket_name)
            raise

    try:
        bucket.Versioning().enable()
        logger.info("Enabled versioning on bucket %s.", bucket.name)
    except ClientError:
        logger.exception("Couldn't enable versioning on bucket %s.", bucket.name)
        raise

    try:
        expiration = 7
        bucket.LifecycleConfiguration().put(
            LifecycleConfiguration={
                "Rules": [
```

```
                    {
                        "Status": "Enabled",
                        "Prefix": prefix,
                        "NoncurrentVersionExpiration": {"NoncurrentDays":
expiration},
                    }
                ]
            }
        )
        logger.info(
            "Configured lifecycle to expire noncurrent versions after %s days "
            "on bucket %s.",
            expiration,
            bucket.name,
        )
    except ClientError as error:
        logger.warning(
            "Couldn't configure lifecycle on bucket %s because %s. "
            "Continuing anyway.",
            bucket.name,
            error,
        )

    return bucket
```

MFALöschen konfigurieren

Wenn Sie mit der S3-Versionierung in Amazon S3 S3-Buckets arbeiten, können Sie optional eine weitere Sicherheitsebene hinzufügen, indem Sie einen Bucket so konfigurieren, dass er das Löschen MFA(Multi-Faktor-Authentifizierung) aktiviert. In diesem Fall muss der Bucket-Eigentümer zwei Authentifizierungsformen in jede Anforderung aufnehmen, um eine Version zu löschen oder den Versioning-Status des Buckets zu ändern.

MFAdelete erfordert eine zusätzliche Authentifizierung für einen der folgenden Vorgänge:

- Ändern des Versioning-Status Ihres Buckets
- Dauerhaftes Löschen einer Objektversion

MFAdelete erfordert zwei Arten der Authentifizierung zusammen:

- Ihre Sicherheitsanmeldeinformationen
- Die Verkettung einer gültigen Seriennummer, eines Leerzeichens und des sechsstelligen Codes, der auf einem zugelassenen Authentifizierungsgerät angezeigt wird

MFAdelete bietet somit zusätzliche Sicherheit, falls beispielsweise Ihre Sicherheitsanmeldedaten gefährdet sind. MFAdelete kann dazu beitragen, das versehentliche Löschen von Buckets zu verhindern, indem der Benutzer, der die Löschaktion initiiert, den physischen Besitz eines MFA Geräts anhand eines MFA Codes nachweisen muss. Dadurch wird die Löschaktion noch reibungsloser und sicherer.

Um Buckets zu identifizieren, für die das MFA Löschen aktiviert ist, können Sie Amazon S3 Storage Lens-Metriken verwenden. S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Weitere Informationen finden Sie unter Bewertung Ihrer Speicheraktivität und -nutzung mit S3 Storage Lens. Eine vollständige Liste der Metriken finden Sie im Glossar der S3-Storage-Lens-Metriken.

Der Bucket-Besitzer, derjenige AWS-Konto , der den Bucket erstellt hat (Root-Konto), und alle autorisierten Benutzer können die Versionierung aktivieren. Allerdings kann nur der Bucket-Besitzer (Root-Konto) das MFA Löschen aktivieren. Weitere Informationen finden Sie im AWS Sicherheitsblog unter Sichern des Zugriffs MFA auf die AWS Nutzung.

> **Note**
>
> Um MFA Löschen zusammen mit der Versionierung zu verwenden, aktivieren `MFA Delete` Sie. Sie können jedoch nicht `MFA Delete` mit der AWS Management Console aktivieren. Sie müssen das AWS Command Line Interface (AWS CLI) oder das API verwenden.
> Beispiele für die Verwendung von MFA Delete zusammen mit der Versionierung finden Sie im Abschnitt mit den Beispielen des ThemasAktivieren des Versioning für Buckets.
> Sie können MFA Delete nicht mit Lebenszykluskonfigurationen verwenden. Weitere Informationen zu Lebenszykluskonfigurationen und deren Interaktion mit anderen Konfigurationen finden Sie unter Wie S3 Lifecycle mit anderen Bucket-Konfigurationen interagiert.

Um das MFA Löschen zu aktivieren oder zu deaktivieren, verwenden Sie dasselbe VerfahrenAPI, mit dem Sie die Versionierung für einen Bucket konfigurieren. Amazon S3 speichert die MFA Löschkonfiguration in derselben Versionierungs-Subressource, die den Versionsstatus des Buckets speichert.

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>VersioningState</Status>
  <MfaDelete>MfaDeleteState</MfaDelete>
</VersioningConfiguration>
```

Um MFA Delete zu verwenden, können Sie entweder ein Hardware- oder ein virtuelles MFA Gerät verwenden, um einen Authentifizierungscode zu generieren. Das folgende Beispiel zeigt einen generierten Authentifizierungscode, angezeigt auf einem Hardwaregerät.

MFALöschen und MFA geschützter API Zugriff sind Funktionen, die Schutz für verschiedene Szenarien bieten sollen. Sie konfigurieren das MFA Löschen für einen Bucket, um sicherzustellen, dass die Daten in Ihrem Bucket nicht versehentlich gelöscht werden können. MFA-geschützter API Zugriff wird verwendet, um einen anderen Authentifizierungsfaktor (MFACode) beim Zugriff auf sensible Amazon S3 S3-Ressourcen durchzusetzen. Sie können verlangen, dass alle Operationen mit diesen Amazon S3 S3-Ressourcen mit temporären Anmeldeinformationen ausgeführt werden, die mit erstellt wurdenMFA. Ein Beispiel finden Sie unter Erforderlich MFA.

Weitere Informationen zum Kauf und zur Aktivierung eines Authentifizierungsgeräts finden Sie unter Multi-Faktor-Authentifizierung.

Um die S3-Versionierung zu aktivieren und zu konfigurierenMFA, löschen

Mit dem AWS CLI

Die Seriennummer ist die Nummer, die das MFA Gerät eindeutig identifiziert. Bei physischen MFA Geräten ist dies die eindeutige Seriennummer, die im Lieferumfang des Geräts enthalten ist. Bei virtuellen MFA Geräten ist die Seriennummer das GerätARN.

Das folgende Beispiel aktiviert die S3-Versionierung und das Löschen mit Multi-Faktor-Authentifizierung (MFA) für einen Bucket.

```
aws s3api put-bucket-versioning --bucket amzn-s3-demo-bucket1 --versioning-
configuration Status=Enabled,MFADelete=Enabled --mfa "SERIAL 123456"
```

Mit dem REST API

Weitere Informationen zur Angabe von MFA Delete mithilfe von Amazon S3 REST API finden Sie unter PutBucketVersioningAmazon Simple Storage Service API Reference.

Arbeiten mit Objekten in einem versioning-fähigen Bucket

Objekte, die in einem Amazon-S3-Bucket gespeichert waren, bevor Sie den Versioning-Status einrichten, haben die Versions-ID `null`. Wenn Sie das Versioning aktivieren, ändern sich die in Ihrem Bucket vorhandenen Objekte nicht mehr. Was sich ändert, ist, wie Amazon S3 die Objekte in zukünftigen Anfragen verarbeitet.

Übergang von Objektversionen

Sie können Lebenszyklus-Konfigurationsregeln für Objekte definieren, die einen definierten Lebenszyklus haben, um Objektversionen zu einem bestimmten Zeitpunkt in der Lebensdauer des Objekts in die Speicherklasse S3 Glacier Flexible Retrieval zu überführen. Weitere Informationen finden Sie unter Verwaltung des Lebenszyklus von Objekten.

Die Themen in diesem Abschnitt erklären verschiedene Objekt-Vorgänge in einem Bucket mit aktiviertem Versioning. Weitere Informationen über das Versioning finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

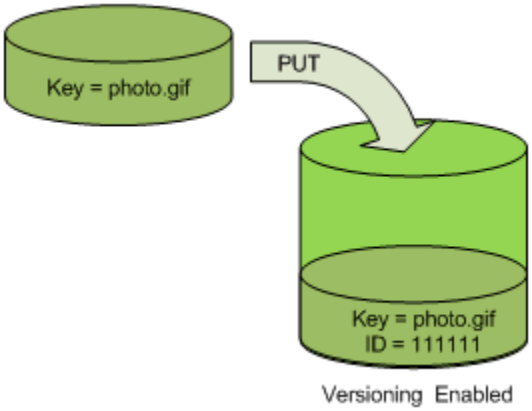
Themen

- Hinzufügen von Objekten zu versioning-fähigen Buckets
- Auflisten von Objekten in einem versioning-fähigen Bucket
- Abrufen von Objektversionen aus einem versioning-fähigen Bucket
- Löschen von Objekten aus einem versioning-fähigen Bucket
- Konfigurieren von versionierten Objektberechtigungen

Hinzufügen von Objekten zu versioning-fähigen Buckets

Nachdem Sie das Versioning für einen Bucket aktiviert haben, fügt Amazon S3 jedem im Bucket gespeicherten Objekt automatisch eine eindeutige Versions-ID hinzu (mit `PUT`, `POST` oder `CopyObject`).

Die folgende Abbildung zeigt, dass Amazon S3 jedem Objekt automatisch eine eindeutige Versions-ID hinzufügt, wenn es einem versioningfähigen Bucket hinzugefügt wird.



> **Note**
>
> Die von Amazon S3 zugewiesenen Versions-ID-Werte sind URL sicher (können als Teil einer enthalten seinURI).

Weitere Informationen über das Versioning finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning. Sie können Objektversionen zu einem Bucket mit aktivierter Versionierung hinzufügen, indem Sie die Konsole, und AWS SDKs verwenden. REST API

Verwenden der Konsole

Detaillierte Anweisungen finden Sie unter Objekte hochladen.

Verwenden von AWS SDKs

Für Beispiele für das Hochladen von Objekten mit dem AWS SDKs für Java,. NET, undPHP, sieheObjekte hochladen. Die Beispiele für das Hochladen von Objekten in versioning-fähige und nicht versioning-fähige Buckets sind gleich, aber bei versioning-fähigen Buckets weist Amazon S3 eine Versionsnummer zu. Andernfalls ist die Versionsnummer null.

Informationen zur Verwendung von anderen AWS SDKs finden Sie im AWS Developer Center.

Mit dem REST API

Hinzufügen von Objekten zu versioning-fähigen Buckets

1. Aktivieren des Versionings für einen Bucket mit einer `PutBucketVersioning`-Anforderung.

Weitere Informationen finden Sie unter PutBucketVersioningin der Amazon Simple Storage Service API Reference.

2. Senden Sie eine `PUT`-, `POST`- oder `CopyObject`-Anforderung, um ein Objekt im Bucket zu speichern.

Wenn Sie einem versioning-fähigen Bucket ein Objekt hinzufügen, gibt Amazon S3 die Versions-ID des Objekts im Antwort-Header `x-amz-version-id` zurück, wie im folgenden Beispiel gezeigt.

```
x-amz-version-id: 3/L4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY
```

Auflisten von Objekten in einem versioning-fähigen Bucket

Dieser Abschnitt zeigt Beispiele, wie Objektversionen aus einem versioning-fähigen Bucket aufgelistet werden. Amazon S3 speichert Versionsinformationen zu Objekten in der versions-Subressource, die dem Bucket zugeordnet ist. Weitere Informationen finden Sie unter Optionen für die Bucket-Konfiguration. Um die Objekte in einem versionsfähigen Bucket aufzulisten, benötigen Sie die `ListBucketVersions`-Berechtigung.

Verwenden der S3-Konsole

Befolgen Sie diese Schritte, um die Amazon-S3-Konsole zu verwenden, um die verschiedenen Versionen eines Objekts anzuzeigen.

Mehrere Versionen eines Objekts anzeigen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.
3. Um eine Liste der Versionen der Objekte in dem Bucket anzuzeigen, wählen Sie den Schalter Show versions (Versionen anzeigen).

   Die Konsole zeigt für jede Objektversion eine eindeutige Versions-ID, das Datum und die Uhrzeit, wann das Objekt erstellt wurde, sowie weitere Eigenschaften an. (Objekte, die in Ihrem Bucket gespeichert waren, bevor Sie den Versioning-Status einrichten, haben die Versions-ID null.)

   Um die Objekte ohne die Versionen aufzulisten, wählen Sie den Schalter List versions (Versionen auflisten) .

Objektversionen können auch in der Objektübersicht auf der Konsole angezeigt, heruntergeladen und gelöscht werden. Weitere Informationen finden Sie unter Anzeigen von Objekteigenschaften in der Amazon-S3-Konsole.

> **Note**
>
> Um auf Objektversionen zuzugreifen, die älter als 300 Versionen sind, müssen Sie die AWS CLI oder die des Objekts verwendenURL.

> **Important**
>
> Sie können den Löschvorgang für ein Objekt nur rückgängig machen, wenn seine aktuelle Version gelöscht wurde. Es ist nicht möglich, das Löschen einer vorherigen Version eines Objekts rückgängig zu machen, das gelöscht wurde. Weitere Informationen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

Verwenden Sie den AWS SDKs

Die Beispiele in diesem Abschnitt veranschaulichen, wie Sie eine Objektliste aus einem versioning-fähigen Bucket abrufen. Jede Anforderung gibt bis zu 1000 Versionen zurück, sofern Sie keine kleinere Anzahl festlegen. Wenn die Versionen im Bucket dieses Limit überschreiten, senden Sie eine Reihe von Anforderungen, um die Liste aller Versionen abzurufen. Dieser Vorgang zur „seitenweisen“ Rückgabe von Ergebnissen wird als Paginierung bezeichnet.

Um zu veranschaulichen, wie Paginierung funktioniert, limitieren die Beispiele jede Antwort auf zwei Objektversionen. Nachdem die erste Seite mit Ergebnissen abgerufen wurde, wird in jedem Beispiel überprüft, ob die Versionsliste abgeschnitten wurde. Wurde sie abgeschnitten, dann wird im Beispiel mit dem seitenweisen Abruf fortgefahren, bis alle Versionen abgerufen wurden.

> **Note**
>
> Die folgenden Beispiele funktionieren auch bei einem Bucket, der versioning-fähig ist, oder bei Objekten ohne individuelle Versionen. In solchen Fällen gibt Amazon S3 die Objektliste mit der Versions-ID von `null` zurück.

Informationen zur Verwendung von anderen AWS SDKs finden Sie im AWS Developer Center.

Java

Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListVersionsRequest;
import com.amazonaws.services.s3.model.S3VersionSummary;
import com.amazonaws.services.s3.model.VersionListing;

public class ListKeysVersioningEnabledBucket {

    public static void main(String[] args) {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Retrieve the list of versions. If the bucket contains more versions
            // than the specified maximum number of results, Amazon S3 returns
            // one page of results per request.
            ListVersionsRequest request = new ListVersionsRequest()
                    .withBucketName(bucketName)
                    .withMaxResults(2);
            VersionListing versionListing = s3Client.listVersions(request);
            int numVersions = 0, numPages = 0;
            while (true) {
                numPages++;
                for (S3VersionSummary objectSummary :
 versionListing.getVersionSummaries()) {
                    System.out.printf("Retrieved object %s, version %s\n",
                            objectSummary.getKey(),
                            objectSummary.getVersionId());
                    numVersions++;
```

```
                }
                // Check whether there are more pages of versions to retrieve. If
                // there are, retrieve them. Otherwise, exit the loop.
                if (versionListing.isTruncated()) {
                    versionListing =
 s3Client.listNextBatchOfVersions(versionListing);
                } else {
                    break;
                }
            }
            System.out.println(numVersions + " object versions retrieved in " +
 numPages + " pages");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

.NET

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem AWS SDK for. NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class ListObjectsVersioningEnabledBucketTest
    {
        static string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
```

```
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main(string[] args)
        {
            s3Client = new AmazonS3Client(bucketRegion);
            GetObjectListWithAllVersionsAsync().Wait();
        }

        static async Task GetObjectListWithAllVersionsAsync()
        {
            try
            {
                ListVersionsRequest request = new ListVersionsRequest()
                {
                    BucketName = bucketName,
                    // You can optionally specify key name prefix in the request
                    // if you want list of object versions of a specific object.

                    // For this example we limit response to return list of 2
versions.
                    MaxKeys = 2
                };
                do
                {
                    ListVersionsResponse response = await
s3Client.ListVersionsAsync(request);
                    // Process response.
                    foreach (S3ObjectVersion entry in response.Versions)
                    {
                        Console.WriteLine("key = {0} size = {1}",
                            entry.Key, entry.Size);
                    }

                    // If response is truncated, set the marker to get the next
                    // set of keys.
                    if (response.IsTruncated)
                    {
                        request.KeyMarker = response.NextKeyMarker;
                        request.VersionIdMarker = response.NextVersionIdMarker;
                    }
                    else
                    {
```

```
                        request = null;
                    }
                } while (request != null);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
    }
}
```

Mit dem REST API

Example – Alle Objektversionen in einem Bucket auflisten

Um alle Versionen aller Objekte in einem Bucket aufzulisten, verwenden Sie die `versions`-Subressource in einer `GET Bucket`-Anfrage. Amazon S3 kann maximal 1000 Objekte abrufen. Jede Objektversion zählt als vollständiges Objekt. Wenn ein Bucket also zwei Schlüssel enthält (z. B. `photo.gif` und `picture.jpg`) und der erste Schlüssel 990 Versionen und der zweite Schlüssel 400 Versionen hat, ruft eine einzelne Abfrage alle 990 Versionen von `photo.gif` und nur die 10 neuesten Versionen von `picture.jpg` ab.

Amazon S3 gibt Objektversionen in der Reihenfolge zurück, in der sie gespeichert wurden, wobei die zuletzt gespeicherte zuerst zurückgegeben wird.

Geben Sie in einer `GET Bucket`-Anforderung die `versions`-Subressource an.

```
GET /?versions HTTP/1.1
Host: bucketName.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 +0000
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Example – Abruf aller Versionen eines Schlüssels

Wenn Sie eine Untermenge von Objektversionen abrufen möchten, verwenden Sie die Anforderungsparameter für `GET Bucket`. Weitere Informationen finden Sie unter GET Bucket.

1. Setzen Sie den Parameter `prefix` auf den Schlüssel des Objekts, das Sie abrufen wollen.
2. Senden Sie eine `GET Bucket`-Anforderung unter Verwendung der `versions`-Subressource und `prefix`.

   ```
   GET /?versions&prefix=objectName HTTP/1.1
   ```

Example – Abrufen von Objekten unter Verwendung eines Präfix

Das folgende Beispiel ruft Objekte ab, deren Schlüssel ist oder damit beginn `myObject`.

```
GET /?versions&prefix=myObject HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Sie können die anderen Anforderungsparameter verwenden, um eine Untermenge aller Versionen des Objekts abzurufen. Weitere Informationen finden Sie unter GET Bucketin der Amazon Simple Storage Service API Reference.

Example – Abrufen einer Liste zusätzlicher Objekte, falls die Antwort gekürzt wurde

Wenn die Anzahl der Objekte, die in einer `GET`-Anforderung zurückgegeben werden können, den Wert `max-keys` überschreitet, enthält die Antwort `<isTruncated>true</isTruncated>`, ebenso wie den ersten Schlüssel (in `NextKeyMarker`) und die erste Versions-ID (in `NextVersionIdMarker`), die die Anforderungskriterien erfüllen, aber nicht zurückgegeben wurden. Diese zurückgegebenen Werte verwenden Sie als Ausgangspunkt in einer nachfolgenden Anforderung, um die zusätzlichen Objekte abzurufen, die die `GET`-Anforderung erfüllen.

Gehen Sie wie folgt vor, um zusätzlichen Objekte abzurufen, die die ursprüngliche `GET Bucket versions`-Anforderung von einem Bucket erfüllen. Weitere Informationen zu`key-marker`, `version-id-marker` `NextKeyMarkerNextVersionIdMarker`, und finden Sie unter GET Bucketin der Amazon Simple Storage Service API Reference.

Im Folgenden finden Sie zusätzliche Antworten, die die ursprüngliche `GET`-Anforderung erfüllen:

- Setzen Sie den Wert von `key-marker` auf den Schlüssel, der in `NextKeyMarker` in der vorherigen Antwort zurückgegeben wurde.
- Setzen Sie den Wert von `version-id-marker` auf die Versions-ID, die in `NextVersionIdMarker` in der vorherigen Antwort zurückgegeben wurde.
- Senden Sie eine `GET Bucket versions`-Anforderung mit `key-marker` und `version-id-marker`.

Example – Abrufen von Objekten ab einem bestimmten Schlüssel und einer bestimmten Versions-ID

```
GET /?versions&key-marker=myObject&version-id-marker=298459348571 HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Mit dem AWS CLI

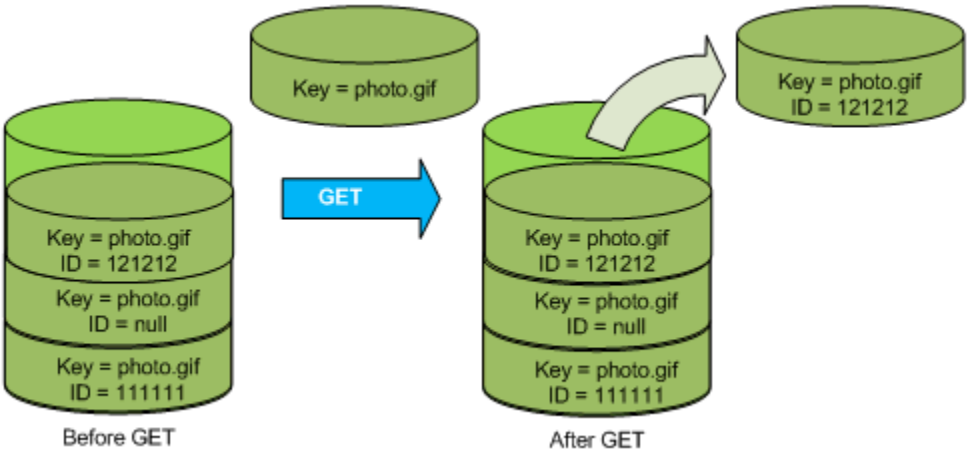
Der folgende Befehl gibt Metadaten zu allen Versionen der Objekte in einem Bucket zurück.

```
aws s3api list-object-versions --bucket amzn-s3-demo-bucket1
```

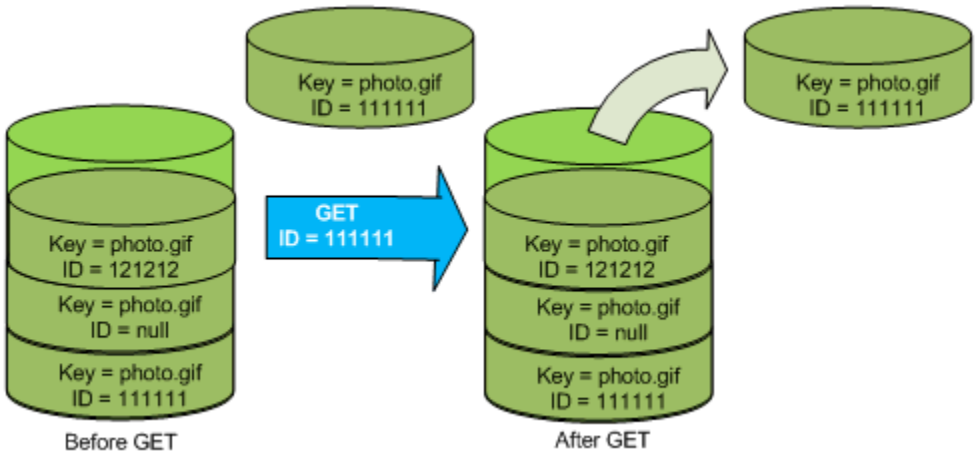
Weitere Informationen zu finden `list-object-versions` Sie unter list-object-versionsin der AWS CLI Befehlsreferenz.

Abrufen von Objektversionen aus einem versioning-fähigen Bucket

Das Versioning in Amazon S3 ist eine Möglichkeit, mehrere Varianten eines Objekts im selben Bucket zu behalten. Eine einfache `GET`-Anforderung ruft die aktuelle Version eines Objekts ab. Die folgende Abbildung zeigt, wie `GET` die aktuelle Version des Objekts zurückgibt, `photo.gif`.



Um eine spezifische Version abzurufen, müssen Sie ihre Versions-ID angeben. Die folgende Abbildung zeigt, dass eine `GET versionId`-Anforderung die angegebene Version des Objekts zurückgibt (nicht unbedingt die aktuelle).



Sie können Objektversionen in Amazon S3 mit der Konsole, AWS SDKs, oder abrufen RESTAPI.

> **Note**
>
> Um auf Objektversionen zuzugreifen, die älter als 300 Versionen sind, müssen Sie die AWS CLI oder die des Objekts verwendenURL.

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.
3. Wählen Sie in der Liste Objects (Objekte) den Namen des Objekts aus.
4. Wählen Sie Versions (Versionen).

   Amazon S3 zeigt alle Versionen für das Objekt an.
5. Aktivieren Sie das Kontrollkästchen neben der Version ID (Versions-ID) für die Versionen, die Sie abrufen möchten.
6. Wählen Sie Actions (Aktionen), wählen Sie Download (Herunterladen) und speichern Sie das Objekt.

Objektversionen können auch in der Objektübersicht angezeigt, heruntergeladen und gelöscht werden. Weitere Informationen finden Sie unter Anzeigen von Objekteigenschaften in der Amazon-S3-Konsole.

> **Important**
>
> Sie können den Löschvorgang für ein Objekt nur rückgängig machen, wenn seine aktuelle Version gelöscht wurde. Es ist nicht möglich, das Löschen einer vorherigen Version eines Objekts rückgängig zu machen, das gelöscht wurde. Weitere Informationen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

Verwenden Sie den AWS SDKs

Die Beispiele für das Hochladen von Objekten in nicht versioning-fähigen und versioning-fähigen Buckets sind dieselben. Für versioning-fähige Buckets weist Amazon S3 jedoch eine Versionsnummer zu. Andernfalls ist die Versionsnummer null.

Beispiele für das Herunterladen von Objekten mit AWS SDKs für Java,. NET, undPHP, siehe Objekte herunterladen.

Beispiele für das Auflisten der Version von Objekten, die AWS SDKs für verwendet werden. NETund Rust, siehe Auflisten der Version von Objekten in einem Amazon S3 S3-Bucket.

Mit dem REST API

Abrufen einer spezifischen Objektversion

1. Setzen Sie den Parameter `versionId` auf die ID der Version des Objekts, die Sie abrufen wollen.
2. Senden Sie eine `GET Object versionId`-Anforderung.

Example – Abrufen eines versionierten Objekts

Die folgende Anforderung ruft die Version `L4kqtJlcpXroDTDmpUMLUo` von ab `my-image.jpg`.

```
GET /my-image.jpg?versionId=L4kqtJlcpXroDTDmpUMLUo HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Sie können nur die Metadaten eines Objekts (nicht den Inhalt) abrufen. Weitere Informationen finden Sie unter the section called “Abrufen von Versionsmetadaten”.

Informationen zum Wiederherstellen einer früheren Objektversion finden Sie unter the section called “Wiederherstellen früherer Versionen”.

Abrufen der Metadaten einer Objektversion

Wenn Sie nur die Metadaten eines Objekts abrufen wollen (nicht seinen Inhalt), verwenden Sie die HEAD-Operation. Standardmäßig erhalten Sie die Metadaten der aktuellsten Version. Um die Metadaten einer spezifischen Objektversion abzurufen, müssen Sie ihre Versions-ID angeben.

Abruf der Metadaten einer Objektversion

1. Setzen Sie den Parameter `versionId` auf die ID der Version des Objekts, dessen Metadaten Sie abrufen wollen.
2. Senden Sie eine `HEAD Object versionId`-Anforderung.

Example – Abrufen der Metadaten eines versionierten Objekts

Die folgende Anforderung ruft die Metadaten der Version `3HL4kqCxf3vjVBH40Nrjfkd` von `my-image.jpg` ab.

```
HEAD /my-image.jpg?versionId=3HL4kqCxf3vjVBH40Nrjfkd HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Nachfolgend ist eine Beispielantwort angezeigt.

```
HTTP/1.1 200 OK
x-amz-id-2: ef8yU9AS1ed4OpIszj7UDNEHGran
x-amz-request-id: 318BC8BC143432E5
x-amz-version-id: 3HL4kqtJlcpXroDTDmjVBH40Nrjfkd
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2006 12:00:00 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
Content-Type: text/plain
Connection: close
Server: AmazonS3
```

Wiederherstellen früherer Versionen

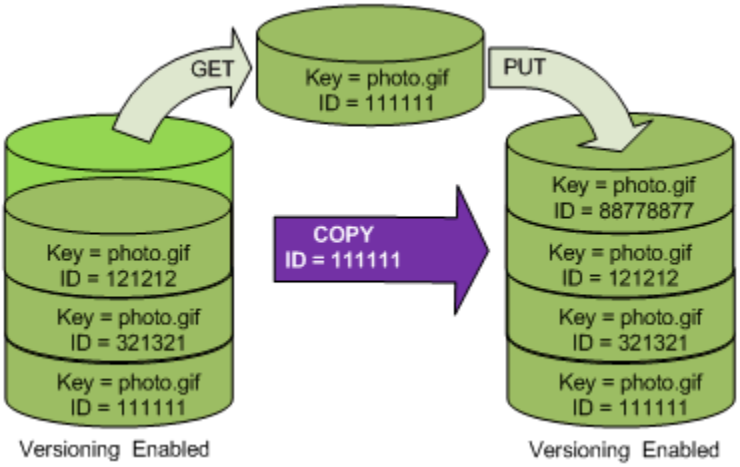
Sie können Versioning verwenden, um frühere Versionen eines Objekts abzurufen. Hierfür gibt es zwei Ansätze:

- Kopieren einer vorhergehenden Version des Objekts in denselben Bucket.

  Das kopierte Objekt wird zur aktuellen Version dieses Objekts, und alle Objektversionen werden beibehalten.
- Dauerhaftes Löschen der aktuellen Version des Objekts

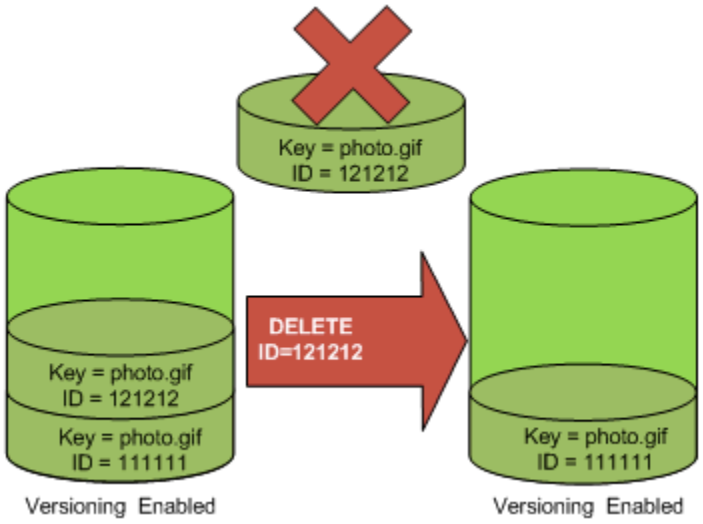
  Wenn Sie die aktuelle Objektversion löschen, wandeln Sie letztlich die nicht vorherige Version in die aktuelle Version dieses Objekts um.

Alle Objektversionen werden aufbewahrt, deshalb können Sie eine frühere Version zur aktuellen Version machen, indem Sie eine spezifische Version des Objekts in denselben Bucket kopieren. In der folgenden Abbildung wird das Quellobjekt (ID = 111111) in denselben Bucket kopiert. Amazon S3 stellt eine neue ID (88778877) bereit und diese wird zur aktuellen Version des Objekts. Der Bucket enthält jetzt also die ursprüngliche Objektversion (111111) und kopiert seine Kopie (88778877). Weitere Informationen zum Abrufen einer früheren Version und zum anschließenden Hochladen, um sie zur aktuellen Version zu machen, finden Sie unter Objektversionen aus einem versionsfähigen Bucket abrufen und Objekte hochladen.



Eine nachfolgende GET ruft Version 88778877 ab.

Die folgende Abbildung zeigt, wie die aktuelle Version (121212) eines Objekts gelöscht wird, sodass die vorhergehende Version (111111) zum aktuellen Objekt wird. Weitere Informationen zum Löschen eines Objekts finden Sie unter Löschen eines einzelnen Objekts.



Ein weiterer GET ruft die Version 111111 ab.

> **Note**
>
> Wenn Sie Objektversionen in Batches wiederherstellen möchten, können Sie die `CopyObject`-Operation verwenden. Die `CopyObject`-Operation kopiert jedes im Manifest angegebene Objekt. Objekte werden jedoch nicht zwingend in derselben Reihenfolge kopiert, in der sie im Manifest erscheinen. Wenn für versionierte Buckets die Beibehaltung der aktuellen/nicht aktuellen Versionsreihenfolge wichtig ist, sollten Sie zuerst alle nicht aktuellen Versionen kopieren. Kopieren Sie dann, nachdem der erste Auftrag abgeschlossen ist, die aktuellen Versionen in einen nachfolgenden Auftrag.

So stellen Sie frühere Objektversionen wieder her

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.
3. Wählen Sie in der Liste Objects (Objekte) den Namen des Objekts aus.
4. Wählen Sie Versions (Versionen).

   Amazon S3 zeigt alle Versionen für das Objekt an.

5. Aktivieren Sie das Kontrollkästchen neben der Version ID (Versions-ID) für die Versionen, die Sie abrufen möchten.
6. Wählen Sie Actions (Aktionen), wählen Sie Download (Herunterladen) und speichern Sie das Objekt.

Objektversionen können auch in der Objektübersicht angezeigt, heruntergeladen und gelöscht werden. Weitere Informationen finden Sie unter Anzeigen von Objekteigenschaften in der Amazon-S3-Konsole.

> **Important**
>
> Sie können den Löschvorgang für ein Objekt nur rückgängig machen, wenn seine aktuelle Version gelöscht wurde. Es ist nicht möglich, das Löschen einer vorherigen Version eines Objekts rückgängig zu machen, das gelöscht wurde. Weitere Informationen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

Verwenden Sie den AWS SDKs

Informationen zur Verwendung von anderen AWS SDKs finden Sie im AWS Developer Center.

Python

Im folgenden Python-Codebeispiel wird die vorherige Version eines versionierten Objekts wiederhergestellt, indem alle Versionen gelöscht werden, die nach der angegebenen Rollback-Version aufgetreten sind.

```
def rollback_object(bucket, object_key, version_id):
    """
    Rolls back an object to an earlier version by deleting all versions that
    occurred after the specified rollback version.

    Usage is shown in the usage_demo_single_object function at the end of this
 module.

    :param bucket: The bucket that holds the object to roll back.
    :param object_key: The object to roll back.
    :param version_id: The version ID to roll back to.
    """
```

```
    # Versions must be sorted by last_modified date because delete markers are
    # at the end of the list even when they are interspersed in time.
    versions = sorted(
        bucket.object_versions.filter(Prefix=object_key),
        key=attrgetter("last_modified"),
        reverse=True,
    )

    logger.debug(
        "Got versions:\n%s",
        "\n".join(
            [
                f"\t{version.version_id}, last modified {version.last_modified}"
                for version in versions
            ]
        ),
    )

    if version_id in [ver.version_id for ver in versions]:
        print(f"Rolling back to version {version_id}")
        for version in versions:
            if version.version_id != version_id:
                version.delete()
                print(f"Deleted version {version.version_id}")
            else:
                break

        print(f"Active version is now {bucket.Object(object_key).version_id}")
    else:
        raise KeyError(
            f"{version_id} was not found in the list of versions for "
f"{object_key}."
        )
```

Löschen von Objekten aus einem versioning-fähigen Bucket

Sie können Objektversionen aus Amazon-S3-Buckets löschen, wann immer Sie möchten. Sie können auch Lebenszyklus-Konfigurationsregeln für Objekte definieren, die einen definierten Lebenszyklus haben, um Amazon S3 aufzufordern, die aktuellen Objektversionen ablaufen zu lassen, oder die nicht aktuellen Objektversionen ständig zu entfernen. Wenn in Ihrem Bucket die

Versionsverwaltung aktiviert ist oder die Versionierung ausgesetzt ist, funktionieren die Aktionen zur Lebenszykluskonfiguration wie folgt:

- Die Aktion `Expiration` gilt für die aktuelle Objektversion. Statt die aktuelle Objektversion zu löschen, behält Amazon S3 die aktuelle Version als nicht aktuelle Version bei, indem es eine Löschmarkierung hinzufügt, die anschließend zur aktuellen Version wird.
- Die `NoncurrentVersionExpiration`-Aktion wird nur auf nicht aktuelle Objektversionen angewendet, und Amazon S3 entfernt diese Objektversionen dauerhaft. Dauerhaft entfernte Objekte können nicht wiederhergestellt werden.

Weitere Informationen zu S3 Lifecycle finden Sie unter Verwaltung des Lebenszyklus von Objekten und Beispiele für S3-Lifecycle-Konfigurationen.

Um zu sehen, wie viele aktuelle und nicht aktuelle Objektversionen Ihre Buckets enthalten, können Sie die Metriken von Amazon S3 Storage Lens verwenden. S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Weitere Informationen finden Sie unter Verwenden von S3 Storage Lens zur Optimierung Ihrer Speicherkosten. Eine vollständige Liste der Metriken finden Sie im Glossar der S3-Storage-Lens-Metriken.

> **Note**
>
> Die normalen Amazon S3 S3-Tarife gelten für jede Version eines Objekts, die gespeichert und übertragen wird, einschließlich nicht aktueller Objektversionen. Weitere Informationen finden Sie unter Amazon S3 – Preise.

Löschen von Anwendungsfällen

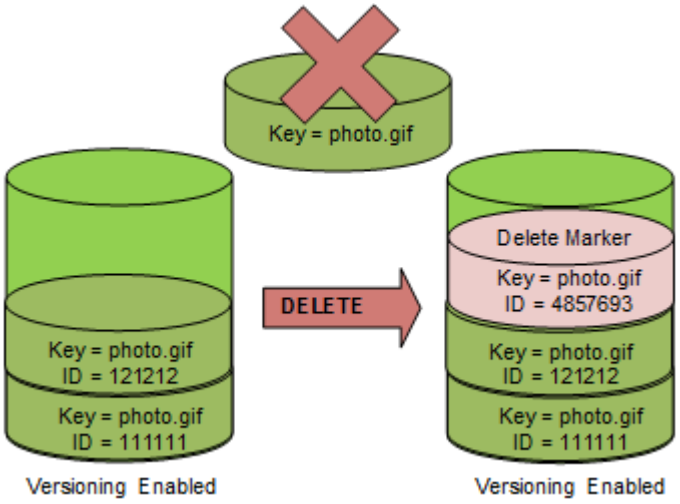
Eine `DELETE`-Anforderung hat die folgenden Anwendungsfälle:

- Wenn das Versioning aktiviert ist, kann ein einfaches `DELETE` ein Objekt nicht dauerhaft löschen. (Eine einfache `DELETE`-Anforderung ist eine Anforderung, die keine Versions-ID angibt.) Stattdessen fügt Amazon S3 eine Löschmarkierung in den Bucket ein, und die Löschmarkierung wird zur aktuellen Objektversion mit einer neuen ID.

  Wenn Sie versuchen, `GET` für ein Objekt auszuführen, dessen aktuelle Version eine Löschmarkierung ist, verhält sich Amazon S3, als wäre das Objekt gelöscht worden (obwohl es

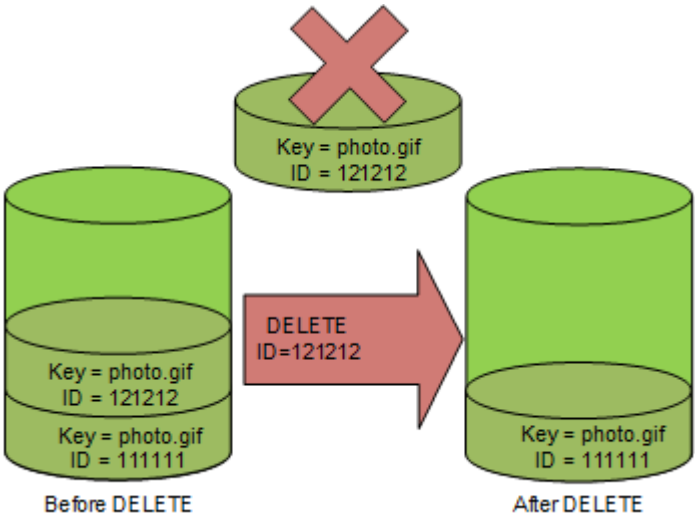
nicht vollständig entfernt wurde), und gibt einen 404-Fehler zurück. Weitere Informationen finden Sie unter Arbeiten mit Löschmarkierungen.

Die folgende Abbildung zeigt, dass ein einfaches `DELETE` das spezifizierte Objekt nicht wirklich löscht. Stattdessen fügt Amazon S3 eine Löschmarkierung ein.



- Um Objektversionen dauerhaft zu löschen, müssen Sie verwende `DELETE Object versionId`.

  Die folgende Abbildung zeigt, dass das beim Löschen einer angegebenen Objektversion dieses Objekt dauerhaft gelöscht wird.



Löschen von Objektversionen

Sie können Objektversionen in Amazon S3 mithilfe der Konsole, AWS SDKs RESTAPI, der oder der löschen AWS Command Line Interface.

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.
3. Wählen Sie in der Liste Objects (Objekte) den Namen des Objekts aus.
4. Wählen Sie Versions (Versionen).

   Amazon S3 zeigt alle Versionen für das Objekt an.
5. Aktivieren Sie das Kontrollkästchen neben der Version ID (Versions-ID) für die Versionen, die Sie permanent löschen möchten.
6. Wählen Sie Delete (Löschen).
7. Geben Sie in Objekte endgültig löschen? **`permanently delete`** ein.

   > ⚠ Warning
   >
   > Wenn Sie eine Objektversion endgültig löschen, kann die Aktion nicht rückgängig gemacht werden.
8. Wählen Sie Delete objects (Objekte löschen).

   Amazon S3 löscht die Objektversion.

Verwenden Sie den AWS SDKs

Beispiele für das Löschen von Objekten mit dem AWS SDKs für Java,. NET, undPHP, sieheLöschen von Amazon-S3-Objekten. Die Beispiele für das Löschen von Objekten in nicht versioning-fähigen und versioning-fähigen Buckets sind dieselben. Für versioning-fähige Buckets weist Amazon S3 jedoch eine Versionsnummer zu. Andernfalls ist die Versionsnummer null.

Informationen zur Verwendung von anderen AWS SDKs finden Sie im AWS Developer Center.

Python

Das folgende Beispiel für Python-Code zeigt das dauerhafte Löschen eines versionierten Objekts, indem alle seine Versionen gelöscht werden.

```
def permanently_delete_object(bucket, object_key):
```

```
    """
    Permanently deletes a versioned object by deleting all of its versions.

    Usage is shown in the usage_demo_single_object function at the end of this
module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to delete.
    """
    try:
        bucket.object_versions.filter(Prefix=object_key).delete()
        logger.info("Permanently deleted all versions of object %s.", object_key)
    except ClientError:
        logger.exception("Couldn't delete all versions of %s.", object_key)
        raise
```

Mit dem REST API

Löschen einer spezifischen Objektversion

- Geben Sie in einem `DELETE` eine Versions-ID ein.

Example – Löschen einer spezifischen Version

Im folgenden Beispiel wird die Version `UIORUnfnd89493jJFJ` von `photo.gif` gelöscht.

```
DELETE /photo.gif?versionId=UIORUnfnd89493jJFJ HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:xQE0diMbLRepdf3YB+FIEXAMPLE=
Content-Type: text/plain
Content-Length: 0
```

Mit dem AWS CLI

Der folgende Befehl löscht ein Objekt namens test.txt aus einem Bucket mit dem Namen *amzn-s3-demo-bucket1*. Zum Entfernen einer bestimmte Version eines Objekts müssen Sie der Bucket-Eigentümer sein und die Versions-ID-Subressource verwenden.

```
aws s3api delete-object --bucket amzn-s3-demo-bucket1 --key test.txt --version-
id versionID
```

Weitere Informationen zu finden `delete-object` Sie unter delete-objectin der AWS CLI Befehlsreferenz.

Weitere Informationen zum Löschen von Objektversionen finden Sie in den folgenden Themen:

- Arbeiten mit Löschmarkierungen
- Entfernen von Löschmarkierungen, um eine ältere Version aktuell zu machen
- Löschen eines Objekts aus einem Bucket mit MFA aktivierter Löschfunktion

Arbeiten mit Löschmarkierungen

Eine Löschmarkierung in Amazon S3 ist ein Platzhalter (bzw. eine Markierung) für ein versioniertes Objekt, das in einer einfachen `DELETE`-Anforderung angegeben wurde. Eine einfache `DELETE`-Anforderung ist eine Anforderung, die keine Versions-ID angibt. Weil sich das Objekt in einem versioning-fähigen Bucket befindet, wurde das Objekt nicht gelöscht. Die Löschmarkierung bewirkt jedoch, dass sich Amazon S3 verhält, als wäre das Objekt gelöscht worden. Sie können einen Amazon S3 API `DELETE` S3-Aufruf für eine Löschmarke verwenden. Dazu müssen Sie die `DELETE` Anfrage mithilfe eines AWS Identity and Access Management (IAM) -Benutzers oder einer Rolle mit den entsprechenden Berechtigungen stellen.

Eine Löschmarkierung hat einen Schlüsselnamen (oder Schlüssel) und eine Versions-ID, so wie jedes andere Objekt. Eine Löschmarkierung unterscheidet sich jedoch wie folgt von anderen Objekten:

- Einer Löschmarkierung sind keine Daten zugeordnet.
- Eine Löschmarkierung ist keinem Wert in der Zugriffskontrollliste (ACL) zugeordnet.
- Wenn Sie eine `GET`-Anforderung für eine Löschmarkierung ausgeben, ruft die `GET`-Anforderung nichts ab, da eine Löschmarkierung keine Daten enthält. Insbesondere, wenn in Ihrer `GET`-Anforderung keine `versionId` angegeben ist, erhalten Sie den Fehler 404 (Nicht gefunden).

Für Löschmarkierungen fällt eine Mindestgebühr für die Speicherung in Amazon S3 an.
Der Speicherbedarf einer Löschmarkierung ist gleich der Größe des Schlüsselnamens der Löschmarkierung. Ein Schlüsselname ist eine Folge von Unicode-Zeichen. Die Kodierung UTF -8 für

den Schlüsselnamen fügt Ihrem Bucket für jedes Zeichen im Namen 1 bis 4 Byte Speicherplatz hinzu. Löschmarkierungen werden in der S3-Standardspeicherklasse gespeichert.

Wenn Sie ermitteln möchten, wie viele Löschmarkierungen Sie haben und in welcher Speicherklasse diese gespeichert sind, können Sie Amazon S3 Storage Lens verwenden. Weitere Informationen erhalten Sie unter Bewerten Ihrer Speicheraktivität und -nutzung mit Amazon S3 Storage Lens und Amazon S3-Storage-Lens-Metrikglossar.

Weitere Informationen zu Schlüsselnamen finden Sie unter Benennen Amazon S3 S3-Objekten. Informationen zum Löschen von Löschmarkierungen finden Sie unter Verwalten von Löschmarkierungen.

Nur Amazon S3 kann eine Löschmarkierung erstellen. Dies erfolgt immer, wenn Sie eine `DeleteObject`-Anfrage für ein Objekt in einem Bucket mit Versioning oder ausgesetztem Versioning stellen. Das in der `DELETE`-Anforderung angegebene Objekt wird nicht wirklich gelöscht. Stattdessen wird die Löschmarkierung zur aktuellen Version des Objekts. Der Schlüsselname (oder Schlüssel) des Objekts wird zum Schlüssel der Löschmarkierung.

Wenn Sie ein Objekt abrufen, ohne eine `versionId` in der Anforderung anzugeben, und die aktuelle Version eine Löschmarkierung ist, antwortet Amazon S3 mit Folgendem:

- Fehler 404 (Nicht gefunden)
- Ein Antwort-Header `x-amz-delete-marker: true`

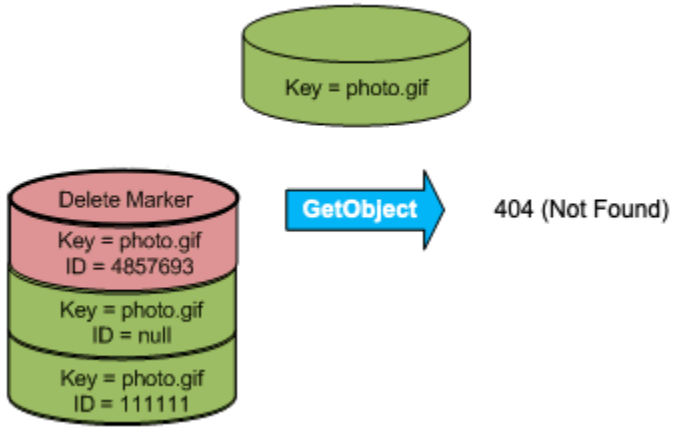
Wenn Sie ein Objekt abrufen, indem Sie eine `versionId` in der Anforderung angeben, und die angegebene Version eine Löschmarkierung ist, antwortet Amazon S3 mit Folgendem:

- Ein Fehler 405 (Method Not Allowed)
- Ein Antwort-Header `x-amz-delete-marker: true`
- Ein Antwort-Header `Last-Modified: timestamp` (nur bei Verwendung der HeadObjectOder-Operationen) GetObjectAPI

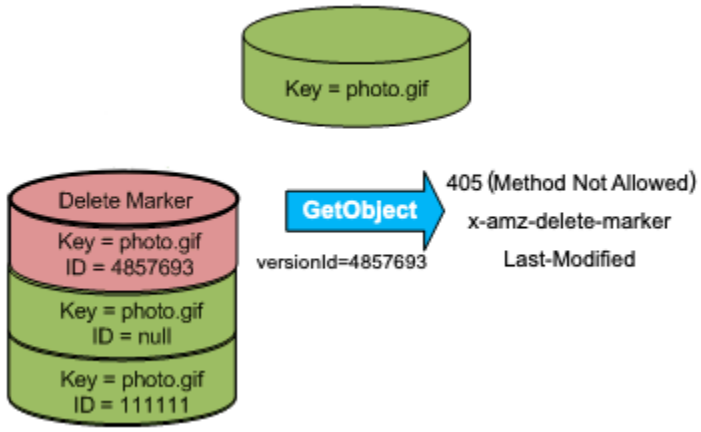
Der Antwort-Header `x-amz-delete-marker: true` teilt Ihnen mit, dass das Objekt, auf das Sie zugegriffen haben, eine Löschmarkierung war. Dieser Antwort-Header gibt niemals `false` zurück, denn wenn der Wert `false` lautet, handelt es sich bei der aktuellen oder angegebenen Version des Objekts nicht um eine Löschmarkierung.

Der Antwort-Header `Last-Modified` gibt die Erstellungszeit der Löschmarkierungen an.

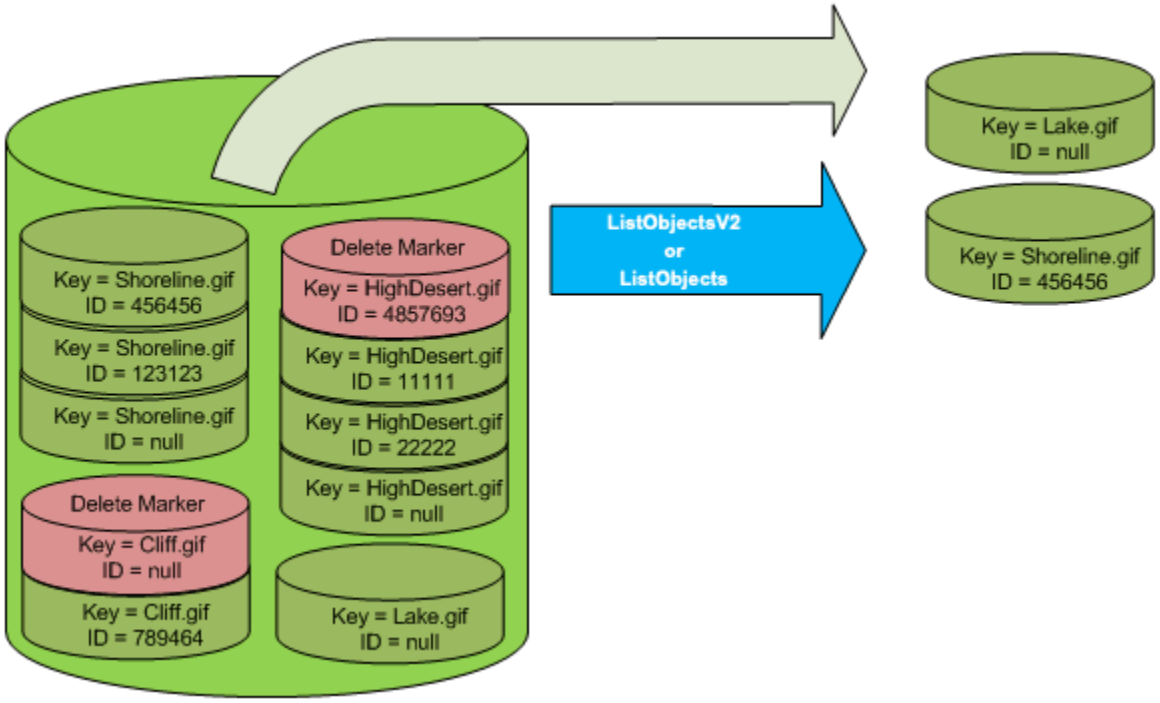
Die folgende Abbildung zeigt, wie ein `GetObject` API Aufruf eines Objekts, dessen aktuelle Version eine Löschmarkierung ist, mit einem 404-Fehler (Not Found) reagiert, und der Antwortheader enthält Folgendes`x-amz-delete-marker: true`.



Wenn Sie einen `GetObject`-Aufruf für ein Objekt durchführen, indem Sie eine `versionId` in Ihrer Anfrage angeben, und wenn die angegebene Version eine Löschmarkierung ist, antwortet Amazon S3 mit dem Fehler 405 (Methode nicht zulässig) und die Antwort-Header enthalten `x-amz-delete-marker: true` und `Last-Modified: timestamp`.



Die einzige Möglichkeit, Löschmarken (und andere Versionen eines Objekts) aufzulisten, besteht darin, die `versions` Subressource in einem ListObjectVersions request. Die folgende Abbildung zeigt, dass ein ListObjectsV2 oder ListObjectsDie Anfrage gibt keine Objekte zurück, deren aktuelle Version eine Löschmarkierung ist.



Verwalten von Löschmarkierungen

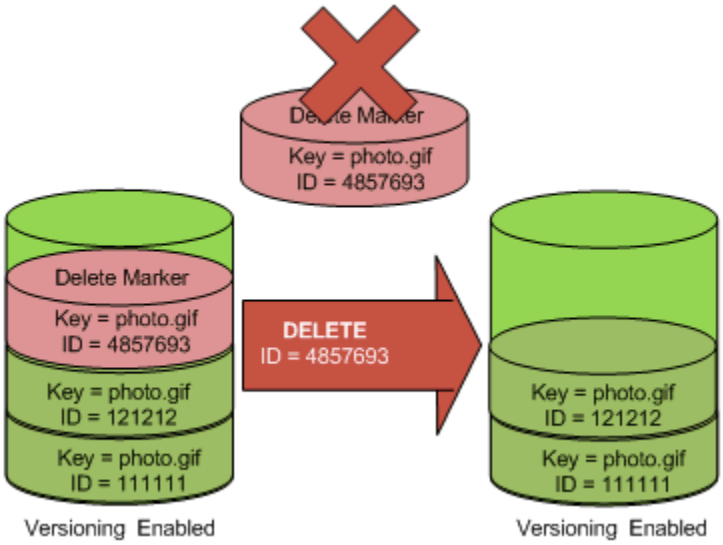
Konfigurieren des Lebenszyklus zum automatischen Bereinigen abgelaufener Löschmarkierungen

Eine abgelaufene Objektlöschmarkierung ist eine, bei der alle Objektversionen gelöscht werden und nur eine einzelne Löschmarkierung erhalten bleibt. Wenn die Lebenszyklus-Konfiguration so eingerichtet ist, dass aktuelle Versionen gelöscht werden, oder die `ExpiredObjectDeleteMarker`-Aktion explizit festgelegt ist, entfernt Amazon S3 die Löschmarkierung des abgelaufenen Objekts. Ein Beispiel finden Sie unter Löschen von Markierungen für das Löschen abgelaufener Objekte in einem Bucket mit aktivierter Versionierung.

Entfernen von Löschmarkierungen, um eine ältere Version aktuell zu machen

Wenn Sie ein Objekt in einem für das Versioning geeignetes Bucket löschen, bleiben alle Versionen in dem Bucket und Amazon S3 erstellt eine Löschmarkierung für das Objekt. Um das Löschen des Objekts rückgängig zu machen, müssen Sie diese Löschmarkierung löschen. Weitere Informationen zum Versioning und zu Löschmarkierungen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

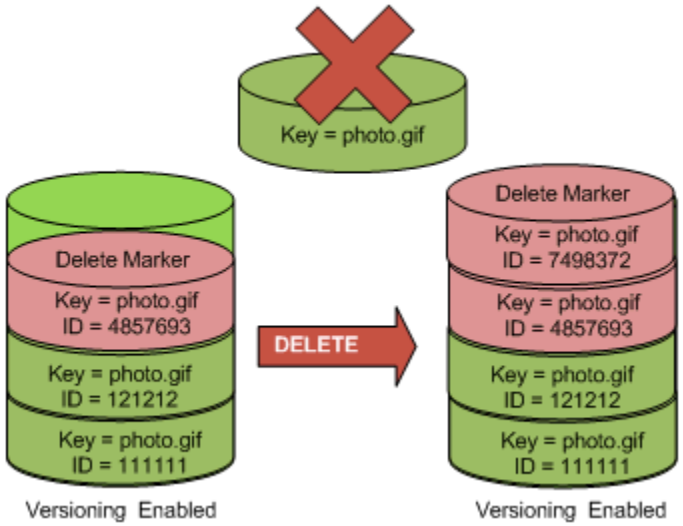
Um eine Löschmarkierung dauerhaft zu löschen, müssen Sie Ihre Versions-ID in einer `DeleteObject versionId`-Anforderung angeben. Die folgende Abbildung zeigt, dass ein einfaches `DeleteObject versionId` eine Löschmarkierung nicht dauerhaft entfernt.



Das Entfernen der Löschmarkierung bewirkt, dass eine einfache `GET`-Anforderung jetzt die aktuelle Versions-ID (121212) des Objekts abruft.

> **Note**
>
> Wenn Sie eine `DeleteObject`-Anforderung verwenden, bei der die aktuelle Version eine Löschmarkierung ist (ohne die Versions-ID der Löschmarkierung anzugeben), löscht Amazon S3 die Löschmarkierung nicht, sondern eine andere `PUT`s Löschmarkierung.

Um eine Löschmarkierung mit einer `NULL`-Versions-ID zu löschen, müssen Sie das `NULL` als Versions-ID in der `DeleteObject`-Anforderung übergeben. Die folgende Abbildung zeigt, wie eine einfache `DeleteObject`-Anfrage ohne Versions-ID, bei der die aktuelle Version eine Löschmarkierung ist, nichts entfernt, sondern stattdessen eine zusätzliche Löschmarkierung mit einer eindeutigen Versions-ID (7498372) hinzufügt.



Verwenden der S3-Konsole

Mit den folgenden Schritten können Sie gelöschte Objekte wiederherstellen, die keine Ordner aus Ihrem S3-Bucket sind, einschließlich der Objekte, die sich in diesen Ordnern befinden.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des von Ihnen erstellten Buckets aus.
3. Um eine Liste der Versionen der Objekte in dem Bucket anzuzeigen, wählen Sie den Schalter List versions (Versionen auflisten). Sie sehen die Löschmarkierungen für gelöschte Objekte.
4. Um das Löschen eines Objekts rückgängig zu machen, müssen Sie die Löschmarkierung löschen. Markieren Sie das Kontrollkästchen neben der Löschmarkierung des Objekts, das wiederhergestellt werden soll, und wählen Sie dann Delete (Löschen).
5. Bestätigen Sie den Löschvorgang auf der Seite Delete objects (Objekte löschen) .
   a. Geben Sie für Permanently delete objects? (Objekte dauerhaft löschen?) **`permanently delete`** ein.
   b. Wählen Sie Delete objects (Objekte löschen).

> **Note**
>
> Sie können die Amazon-S3-Konsole nicht verwenden, um das Löschen von Ordnern rückgängig zu machen. Sie müssen das AWS CLI oder verwendenSDK. Beispiele finden Sie unter How can I retrieve an Amazon S3 object that was deleted in a versioning-enabled

bucket? (Wie kann ich ein Amazon-S3-Objekt wiederherstellen, das in einem Versioning-fähigen Bucket gelöscht wurde?) im AWS -Wissenszentrum

Mit dem REST API

Eine Löschmarkierung dauerhaft entfernen

1. Setzen Sie den Parameter `versionId` auf die ID der Version der Löschmarkierung, die Sie entfernen wollen.
2. Senden Sie eine `DELETE Object versionId`-Anforderung.

Example – Entfernen einer Löschmarkierung

Das folgende Beispiel entfernt die Löschmarkierung für `photo.gif` Version 4857693.

```
DELETE /photo.gif?versionId=4857693 HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Wenn Sie eine Löschmarkierung löschen, nimmt Amazon S3 folgendes in die Antwort auf.

```
204 NoContent
x-amz-version-id: versionID
x-amz-delete-marker: true
```

Mit dem AWS SDKs

Informationen zur Verwendung von anderen AWS SDKs finden Sie im AWS Developer Center.

Python

Das folgende Python-Codebeispiel zeigt, wie Sie einen Löschmarker von einem Objekt entfernen und damit die neueste nicht-aktuelle Version der aktuellsten Version zur aktuellen Version des Objekts machen.

```
def revive_object(bucket, object_key):
    """
    Revives a versioned object that was deleted by removing the object's active
    delete marker.
```

```
    A versioned object presents as deleted when its latest version is a delete
marker.
    By removing the delete marker, we make the previous version the latest version
    and the object then presents as *not* deleted.

    Usage is shown in the usage_demo_single_object function at the end of this
module.

    :param bucket: The bucket that contains the object.
    :param object_key: The object to revive.
    """
    # Get the latest version for the object.
    response = s3.meta.client.list_object_versions(
        Bucket=bucket.name, Prefix=object_key, MaxKeys=1
    )

    if "DeleteMarkers" in response:
        latest_version = response["DeleteMarkers"][0]
        if latest_version["IsLatest"]:
            logger.info(
                "Object %s was indeed deleted on %s. Let's revive it.",
                object_key,
                latest_version["LastModified"],
            )
            obj = bucket.Object(object_key)
            obj.Version(latest_version["VersionId"]).delete()
            logger.info(
                "Revived %s, active version is now %s  with body '%s'",
                object_key,
                obj.version_id,
                obj.get()["Body"].read(),
            )
        else:
            logger.warning(
                "Delete marker is not the latest version for %s!", object_key
            )
    elif "Versions" in response:
        logger.warning("Got an active version for %s, nothing to do.", object_key)
    else:
        logger.error("Couldn't get any version info for %s.", object_key)
```

Löschen eines Objekts aus einem Bucket mit MFA aktivierter Löschfunktion

Wenn für die Versionierungskonfiguration eines Buckets die Option MFA Löschen aktiviert ist, muss der Bucket-Besitzer den `x-amz-mfa` Anforderungsheader in Anfragen aufnehmen, um eine Objektversion dauerhaft zu löschen oder den Versionsstatus des Buckets zu ändern. Anfragen, die Folgendes beinhalten, `x-amz-mfa` müssen verwenden. HTTPS

Der Wert des Headers ist die Verkettung der Seriennummer Ihres Authentifizierungsgeräts, eines Leerzeichens und des darauf angezeigten Authentifizierungscodes. Wenn sie dies im Anforderungs-Header nicht angeben, schlägt die Anforderung fehl.

Weitere Informationen zu Authentifizierungsgeräten finden Sie unter Multi-Factor Authentication.

Example — Löschen eines Objekts aus einem Bucket mit MFA aktivierter Löschfunktion

Im folgenden Beispiel wird ein Objekt `my-image.jpg` (mit der angegebenen Version) gelöscht, das sich in einem Bucket befindet, für das Löschen aktiviert ist. MFA

Beachten Sie den Abstand zwischen *`[SerialNumber]`* and *`[AuthenticationCode]`*Weitere Informationen finden Sie unter . DeleteObjectin der Amazon Simple Storage Service API Reference.

```
DELETE /my-image.jpg?versionId=3HL4kqCxf3vjVBH40Nrjfkd HTTPS/1.1
Host: bucketName.s3.amazonaws.com
x-amz-mfa: 20899872 301749
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
```

Weitere Informationen zur Aktivierung des MFA Löschvorgangs finden Sie unterMFALöschen konfigurieren.

Konfigurieren von versionierten Objektberechtigungen

Die Berechtigungen für Objekte in Amazon S3 werden auf Versionsebene festgelegt. Jede Version hat ihren eigenen Objekteigentümer. AWS-Konto Derjenige, der die Objektversion erstellt, ist der Besitzer. Sie können also unterschiedliche Berechtigungen für unterschiedliche Versionen desselben Objekts einrichten. Dazu geben Sie die Versions-ID des Objekts an, dessen Berechtigungen Sie in einer `PUT Object versionId acl`-Anforderung setzen wollen. Eine ausführliche Beschreibung und Anweisungen zur Verwendung ACLs finden Sie unterIdentity and Access Management für Amazon S3.

Example – Einrichtung von Berechtigungen für eine Objektversion

Die folgende Anforderung legt die Berechtigung für den Empfänger, `BucketOwner@amazon.com`, auf `FULL_CONTROL` für den Schlüssel, `my-image.jpg`, Versions-ID, `3HL4kqtJvjVBH40Nrjfkd` fest.

```
PUT /my-image.jpg?acl&versionId=3HL4kqtJvjVBH40Nrjfkd HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU=
Content-Length: 124

<AccessControlPolicy>
  <Owner>
    <ID>75cc57f09aa0c8caeab4f8c24e99d10f8e7faeebf76c078efc7c6caea54ba06a</ID>
    <DisplayName>mtd@amazon.com</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:type="CanonicalUser">
        <ID>a9a7b886d6fd24a52fe8ca5bef65f89a64e0193f23000e241bf9b1c61be666e9</ID>
        <DisplayName>BucketOwner@amazon.com</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
  </AccessControlPolicy>
```

Analog dazu müssen Sie, um die Berechtigungen für eine spezifische Objektversion zu erhalten, ihre Versions-ID in einer `GET Object versionId acl`-Anforderung angeben. Sie müssen diese Versions-ID angeben, weil `GET Object acl` standardmäßig die Berechtigungen für die aktuelle Version des Objekts zurückgibt.

Example – Abrufen der Berechtigungen für eine bestimmte Objektversion

Im folgenden Beispiel gibt Amazon S3 die Berechtigungen für den Schlüssel, `my-image.jpg`, Versions-ID, `DVBH40Nr8X8gUMLUo` zurück.

```
GET /my-image.jpg?versionId=DVBH40Nr8X8gUMLUo&acl HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
```

```
Authorization: AWS AKIAIOSFODNN7EXAMPLE:0RQf4/cRonhpaBX5sCYVf1bNRuU
```

Weitere Informationen finden Sie unter GetObjectAclin der Amazon Simple Storage Service API Reference.

Arbeiten mit Objekten in einem Bucket mit ausgesetztem Versioning

In Amazon S3 können Sie das Versioning aussetzen, um zu verhindern, dass sich neue Versionen desselben Objekts in einem Bucket ansammeln. Sie könnten dies tun, weil Sie nur eine einzige Version eines Objekts in einem Bucket haben möchten. Oder Sie möchten möglicherweise keine Gebühren für mehrere Versionen anfallen.

Wenn Sie das Versioning aussetzen, ändern sich die in Ihrem Bucket vorhandenen Objekte nicht. Was sich ändert, ist, wie Amazon S3 die Objekte in zukünftigen Anfragen verarbeitet. Die Themen in diesem Abschnitt erläutern verschiedene Objekt-Vorgänge in einem Bucket mit ausgesetztem Versioning, einschließlich Hinzufügen, Abrufen und Löschen von Objekten.

Weitere Informationen über das S3-Versioning finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning. Weitere Informationen über das Abrufen von Objektversionen, finden Sie unter Abrufen von Objektversionen aus einem versioning-fähigen Bucket.

Themen

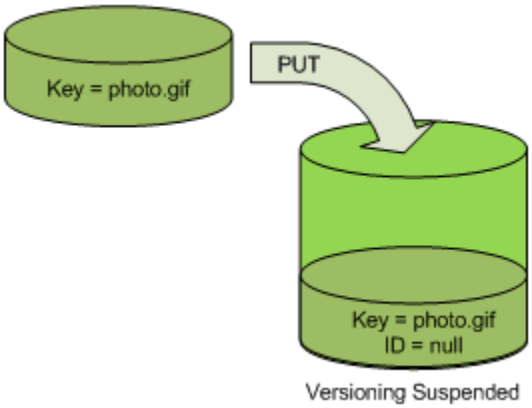
- Hinzufügen von Objekten zu Buckets mit ausgesetztem Versioning
- Abrufen von Objekten aus Buckets mit ausgesetztem Versioning
- Löschen von Objekten aus Buckets mit ausgesetztem Versioning

Hinzufügen von Objekten zu Buckets mit ausgesetztem Versioning

Sie können in Amazon S3 Objekte in Buckets mit ausgesetztem Versioning hinzufügen, um das Objekt mit der Versions-ID null zu erzeugen oder eine Objektversion mit einer übereinstimmenden Versions-ID zu überschreiben.

Nachdem Sie das Versioning für einen Bucket ausgesetzt haben, fügt Amazon S3 automatisch jedem nachfolgend (mit `PUT`, `POST` oder `CopyObject`) in diesem Bucket gespeicherten Objekt die Versions-ID `null` hinzu

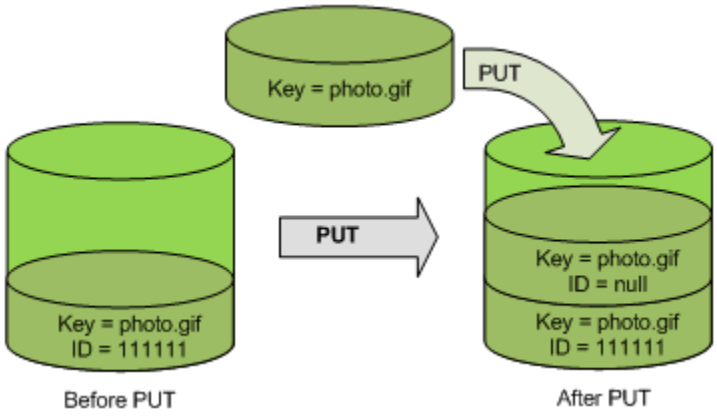
Die folgende Abbildung zeigt, wie Amazon S3 jedem Objekt automatisch die Versions-ID `null` hinzufügt, wenn es einem Bucket mit ausgesetztem Versioning hinzugefügt wird.



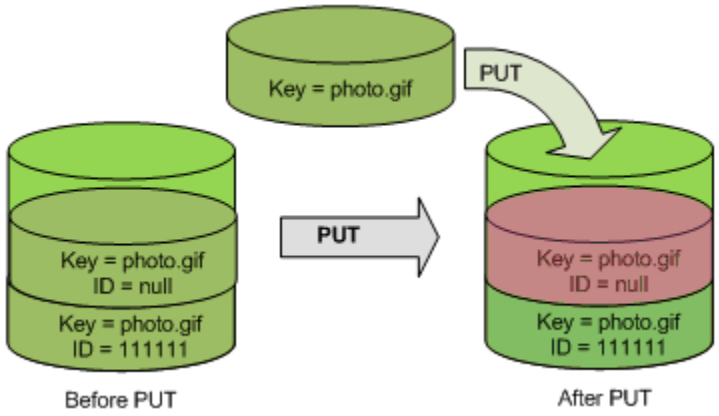
Wenn sich bereits eine Null-Version im Bucket befindet und Sie ein weiteres Objekt mit demselben Schlüssel hinzufügen, überschreibt das hinzugefügte Objekt die ursprüngliche Null-Version.

Wenn es versionsfähige Objekte im Bucket gibt, wird die Version, die Sie mit PUT hinzufügen, zur aktuellen Version des Objekts. Die folgende Abbildung zeigt, wie das Hinzufügen eines Objekts in einen Bucket, der versionsfähige Objekte enthält, das bereits im Bucket enthaltene Objekt nicht überschreibt.

In diesem Fall befand sich Version 111111 bereits im Bucket. Amazon S3 weist dem Objekt, das hinzugefügt und im Bucket gespeichert wird, die Versions-ID null zu. Version 111111 wird nicht überschrieben.



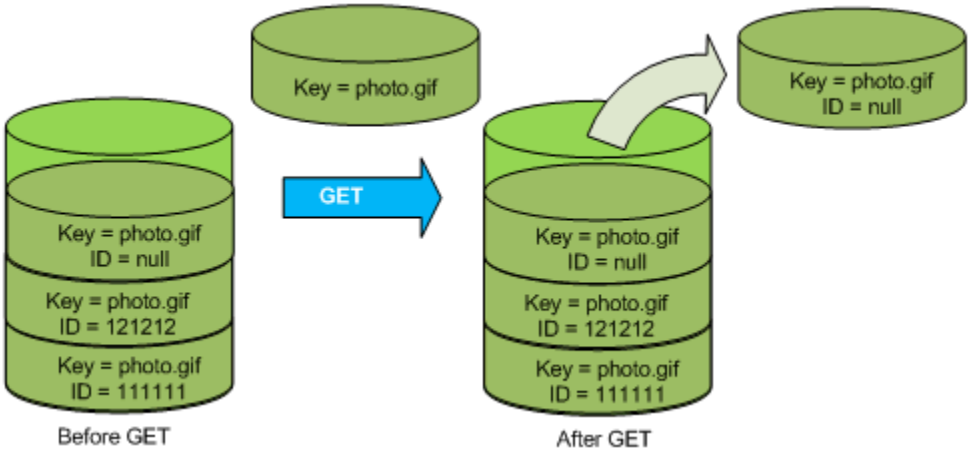
Wenn bereits eine Null-Version in einem Bucket vorhanden ist, wird die Null-Version überschrieben, wie in der folgenden Abbildung gezeigt.



Obwohl der Schlüssel und die Versions-ID (`null`) der Null-Version vor und nach PUT gleich sind, wird der Inhalt der ursprünglich im Bucket gespeicherten Version jedoch durch die Inhalte des Objekts ersetzt, die mit PUT in den Bucket geschrieben wurde.

Abrufen von Objekten aus Buckets mit ausgesetztem Versioning

Eine `GET Object`-Anforderung gibt die aktuelle Version eines Objekts zurück, unabhängig davon, ob Sie das Versioning für einen Bucket aktiviert haben oder nicht. Die folgende Abbildung zeigt, wie ein einfaches `GET` die aktuelle Version des Objekts zurückgibt.



Löschen von Objekten aus Buckets mit ausgesetztem Versioning

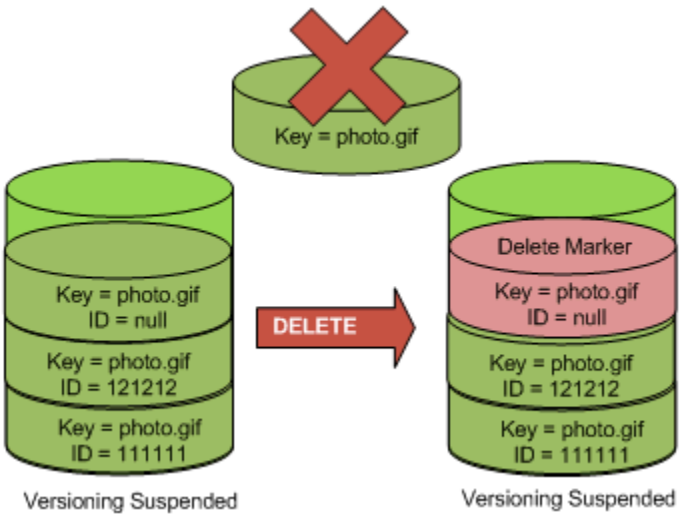
Sie können Objekte aus Buckets mit ausgesetztem Versioning löschen, um ein Objekt mit einer Versions-ID von null zu entfernen.

Wenn das Versioning für ein Bucket ausgesetzt ist, gilt für eine `DELETE`-Anforderung:

- Kann nur ein Objekt entfernen, dessen Versions-ID `null` ist.
- Entfernt nichts, wenn es keine Nullversion des Objekts im Bucket gibt.

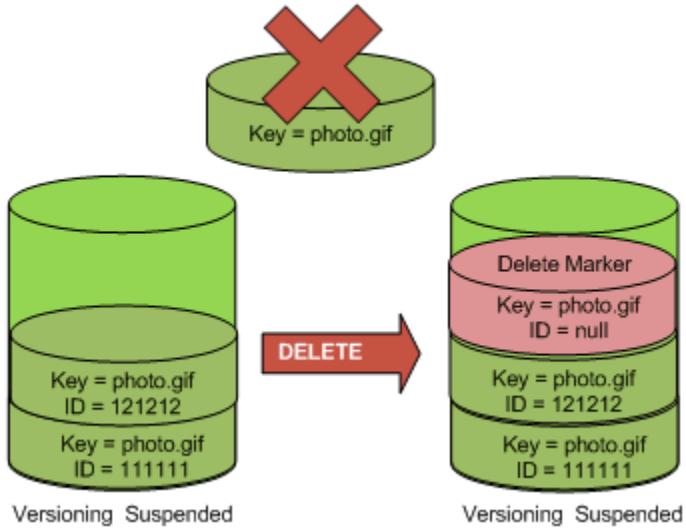
- Fügt eine Löschmarkierung in den Bucket ein.

Wenn die Bucket-Versionierung unterbrochen ist, entfernt der Vorgang das Objekt, das eine Null `versionId` hat. Wenn eine Versions-ID vorhanden ist, fügt Amazon S3 eine Löschmarkierung ein, die zur aktuellen Version des Objekts wird. Die folgende Abbildung zeigt, wie ein Simple eine Nullversion `DELETE` entfernt und Amazon S3 stattdessen eine Löschmarkierung mit einer `null` Versions-ID einfügt.

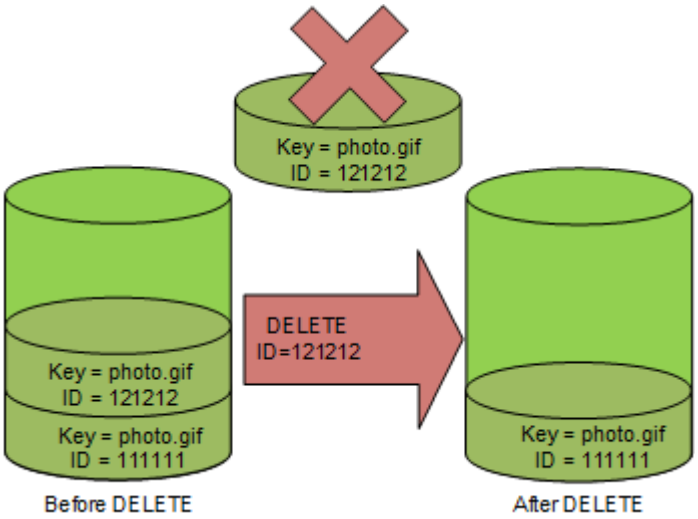


Um ein Objekt mit einer dauerhaft zu löschen`versionId`, müssen Sie die Objekte `versionId` in die Anforderung aufnehmen. Da eine Löschmarke keinen Inhalt enthält, verlieren Sie den Inhalt der `null` Version, wenn sie durch eine Löschmarke ersetzt wird.

Die folgende Abbildung zeigt einen Bucket, der keine Nullversion enthält. In diesem Fall `DELETE` entfernt das nichts. Stattdessen fügt Amazon S3 einfach eine Löschmarkierung ein.



Selbst in einem Bucket mit ausgesetztem Versioning kann der Bucket-Eigentümer eine spezifische Version dauerhaft löschen, indem er die Versions-ID in der DELETE-Anfrage mit einbezieht. Die folgende Abbildung zeigt, dass das beim Löschen einer angegebenen Objektversion diese Version des Objekts dauerhaft gelöscht wird. Nur der Bucket-Eigentümer kann eine spezifische Objektversion löschen.



Problembehandlung bei der Versionierung

Die folgenden Themen können bei der Behebung von allgemeinen Problemen mit der Amazon-S3-Versionsverwaltung hilfreich sein.

Themen

- Ich möchte Objekte wiederherstellen, die in einem Bucket mit aktivierter Versionsverwaltung versehentlich gelöscht wurden.
- Ich möchte versionierte Objekte dauerhaft löschen
- Nach dem Aktivieren der Bucket-Versionsverwaltung stelle ich Leistungseinbußen fest

Ich möchte Objekte wiederherstellen, die in einem Bucket mit aktivierter Versionsverwaltung versehentlich gelöscht wurden.

Wenn Objektversionen aus S3-Buckets gelöscht werden, hat Amazon S3 im Allgemeinen keine Möglichkeit, diese wiederherzustellen. Wenn Sie jedoch die S3-Versionsverwaltung in Ihrem S3-Bucket aktiviert haben, kann ein Objekt mit einer DELETE-Anforderung, in der keine Versions-ID angegeben ist, nicht dauerhaft gelöscht werden. Stattdessen wird eine Löschmarkierung als Platzhalter hinzugefügt. Diese Löschmarkierung wird zur aktuellen Version des Objekts.

Gehen Sie wie folgt vor, um zu überprüfen, ob Ihre gelöschten Objekte dauerhaft oder vorübergehend (mit einer Löschmarkierung an ihrer Stelle) gelöscht wurden:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält.
4. Aktivieren Sie in der Liste Objekte den Schalter Versionen anzeigen rechts neben der Suchleiste und suchen Sie dann in der Suchleiste nach dem gelöschten Objekt. Dieser Schalter ist nur verfügbar, wenn die Versionsverwaltung zuvor für den Bucket aktiviert wurde.

   Sie können auch das S3 Inventory verwenden, um nach gelöschten Objekten zu suchen.
5. Wenn Sie das Objekt nicht finden können, nachdem Sie Versionen anzeigen eingeschaltet oder einen Bestandsbericht erstellt haben, und auch keine Löschmarkierung für das Objekt zu finden ist, ist der Löschvorgang dauerhaft und das Objekt kann nicht wiederhergestellt werden.

Sie können den Status eines gelöschten Objekts auch überprüfen, indem Sie den `HeadObject` API Vorgang von AWS Command Line Interface (AWS CLI) aus verwenden. Verwenden Sie hierfür den folgenden Befehl `head-object` und ersetzen Sie die *`user input placeholders`* durch Ihre eigenen Informationen:

```
aws s3api head-object --bucket amzn-s3-demo-bucket --key index.html
```

Wenn Sie den Befehl `head-object` für ein versioniertes Objekt ausführen, dessen aktuelle Version eine Löschmarkierung ist, erhalten Sie den Fehler 404 Nicht gefunden. Beispielsweise:

Beim Aufrufen der HeadObject Operation ist ein Fehler aufgetreten (404): Not Found

Wenn Sie den Befehl `head-object` für ein versioniertes Objekt ausführen und die Versions-ID des Objekts angeben, ruft Amazon S3 die Metadaten des Objekts ab und bestätigt damit, dass das Objekt noch vorhanden ist und nicht dauerhaft gelöscht wurde.

```
aws s3api head-object --bucket amzn-s3-demo-bucket --key index.html --
version-id versionID
```

```
{
"AcceptRanges": "bytes",
"ContentType": "text/html",
```

```
"LastModified": "Thu, 16 Apr 2015 18:19:14 GMT",
"ContentLength": 77,
"VersionId": "Zg5HyL7m.eZU9iM7AVlJkrqAiE.0UG4q",
"ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",
"Metadata": {}
}
```

Wenn das Objekt gefunden wird und die neueste Version eine Löschmarkierung ist, ist die vorherige Version des Objekts immer noch vorhanden. Da die Löschmarkierung die aktuelle Version des Objekts darstellt, können Sie das Objekt wiederherstellen, indem Sie die Löschmarkierung löschen.

Nachdem Sie die Löschmarkierung dauerhaft entfernt haben, wird die zweitneueste Version des Objekts zur aktuellen Version des Objekts, sodass Ihr Objekt wieder verfügbar ist. Eine visuelle Darstellung der Wiederherstellung von Objekten finden Sie unter Entfernen von Löschmarkierungen.

Um eine bestimmte Version eines Objekts entfernen zu können, müssen Sie der Bucket-Eigentümer sein. Um eine Löschmarkierung dauerhaft zu löschen, müssen Sie Ihre Versions-ID in einer `DeleteObject`-Anforderung angeben. Verwenden Sie zum Löschen der Löschmarkierung den folgenden Befehl und ersetzen Sie die *`user input placeholders`* durch Ihre eigenen Informationen:

`aws s3api delete-object --bucket` *`amzn-s3-demo-bucket`* `--key` *`index.html`* `--version-id` *`versionID`*

Weitere Informationen zu dem `delete-object` Befehl finden Sie unter delete-objectin der AWS CLI Befehlsreferenz. Weitere Informationen zum dauerhaften Löschen von Löschmarkierungen finden Sie unter Verwalten von Löschmarkierungen.

Ich möchte versionierte Objekte dauerhaft löschen

In einem Bucket mit aktivierter Versionsverwaltung kann ein Objekt mit einer `DELETE`-Anforderung ohne Versions-ID nicht dauerhaft gelöscht werden. Bei einer solchen Anforderung wird vielmehr eine Löschmarkierung eingefügt.

Um versionierte Objekte dauerhaft zu löschen, können Sie eine der folgenden Methoden auswählen:

- Erstellen Sie eine S3-Lebenszyklusregel, um nicht aktuelle Versionen dauerhaft zu löschen. Um nicht aktuelle Versionen dauerhaft zu löschen, wählen Sie Nicht aktuelle Versionen von Objekten dauerhaft löschen aus und geben Sie anschließend eine Zahl unter Tage, nachdem Objekte veraltet sind ein. Sie können die Anzahl der beizubehaltenden neueren Versionen optional angeben, indem Sie einen Wert unter Number of newer versions to retain (Anzahl der

beizubehaltenden neueren Versionen) eingeben. Weitere Informationen zum Erstellen dieser Regel finden Sie unter Festlegen einer S3-Lebenszyklus-Konfiguration.

- Löschen Sie eine angegebene Version, indem Sie die Versions-ID in die DELETE-Anforderung aufnehmen. Weitere Informationen hierzu finden Sie unter Dauerhaftes Löschen von versionierten Objekten.
- Erstellen Sie eine Lebenszyklusregel, um aktuelle Versionen ablaufen zu lassen. Um aktuelle Versionen von Objekten ablaufen zu lassen, wählen Sie Aktuelle Objektversionen ablaufen lassen aus und fügen Sie dann eine Zahl unter Tage nach Objekterstellung ein. Weitere Informationen zum Erstellen dieser Lebenszyklusregel finden Sie unter Festlegen einer S3-Lebenszyklus-Konfiguration.
- Um alle versionierten Objekte und Löschmarkierungen dauerhaft zu löschen, erstellen Sie zwei Lebenszyklusregeln: eine, um aktuelle Versionen ablaufen zu lassen und nicht aktuelle Versionen von Objekten dauerhaft zu löschen, und die andere, um Löschmarkierungen für abgelaufene Objekte zu löschen.

In einem Bucket mit aktivierter Versionsverwaltung können mit einer DELETE-Anforderung, in der keine Versions-ID angegeben ist, nur Objekte mit einer Versions-ID NULL entfernt werden. Wenn das Objekt hochgeladen wurde, als die Versionsverwaltung aktiviert war, wird mit einer DELETE-Anforderung, in der keine Versions-ID angegeben ist, eine Löschmarkierung für dieses Objekt erstellt.

> **Note**
>
> Bei Buckets mit aktivierter S3-Objektsperre verursacht eine DELETE-Objektanforderung mit einer ID einer geschützten Objektversion den Fehler 403 Zugriff verweigert. Eine DELETE-Objektanforderung ohne Versions-ID fügt eine Löschmarkierung als neueste Version des Objekts mit der Antwort 200 OK hinzu. Objekte, die durch die Objektsperre geschützt sind, können erst dauerhaft gelöscht werden, wenn ihre Aufbewahrungszeiträume und gesetzlichen Aufbewahrungsfristen aufgehoben sind. Weitere Informationen finden Sie unter the section called “So funktioniert die S3-Objektsperre”.

Nach dem Aktivieren der Bucket-Versionsverwaltung stelle ich Leistungseinbußen fest

Bei Buckets mit aktivierter Versionsverwaltung kann es zu Leistungseinbußen kommen, wenn zu viele Löschmarkierungen oder versionierte Objekte vorhanden sind und wenn bewährte Methoden nicht befolgt werden.

Zu viele Löschmarkierungen

Nachdem Sie die Versionsverwaltung für einen Bucket aktiviert haben, wird mit einer DELETE-Anforderung ohne Versions-ID an ein Objekt eine Löschmarkierung mit einer eindeutigen Versions-ID erstellt. Lebenszyklus-Konfigurationen mit der Regel Aktuelle Objektversionen ablaufen lassen fügen jedem Objekt eine Löschmarkierung mit einer eindeutigen Versions-ID hinzu. Wenn zu viele Löschmarkierungen vorhanden sind, kann dies die Leistung im Bucket beeinträchtigen.

Wenn die Versionsverwaltung für einen Bucket ausgesetzt wird, markiert Amazon S3 die Versions-ID bei neu erstellten Objekten als NULL. Eine Ablaufaktion in einem Bucket mit ausgesetzter Versionsverwaltung bewirkt, dass Amazon S3 eine Löschmarkierung mit der Versions-ID NULL erstellt. In einem Bucket mit ausgesetzter Versionsverwaltung wird für jede Löschanforderung eine NULL-Löschmarkierung erstellt. Diese NULL-Löschmarkierungen werden auch als Löschmarkierungen für abgelaufenes Objekt bezeichnet, wenn alle Objektversionen gelöscht werden und nur eine einzelne Löschmarkierung übrig bleibt. Wenn sich zu viele NULL-Löschmarkierungen ansammeln, kommt es zu Leistungseinbußen im Bucket.

Zu viele versionierte Objekte

Wenn ein Bucket mit aktivierter Versionsverwaltung Objekte mit Millionen von Versionen enthält, kann es vermehrt zu Fehlern 503 Service nicht verfügbar kommen. Wenn Sie feststellen, dass die Anzahl der HTTP 503 Antworten „Service Unavailable“ für einen Bucket mit aktivierter Versionierung PUT oder DELETE Objektanforderungen an einen Bucket mit aktivierter Versionierung deutlich gestiegen ist, befinden sich in dem Bucket möglicherweise ein oder mehrere Objekte mit Millionen von Versionen. Wenn Sie Objekte mit Millionen Versionen haben, drosselt Amazon S3 Anforderungen an den Bucket automatisch. Durch die Drosselung von Anforderungen wird Ihr Bucket vor übermäßigem Anforderungsdatenverkehr geschützt, der andere Anforderungen an denselben Bucket verhindern könnte.

Um festzustellen, zu welchen Objekten es Millionen Versionen gibt, verwenden Sie S3 Inventory. S3 Inventory generiert einen Bericht, der eine flache Dateiliste der Objekte in einem Bucket enthält. Weitere Informationen finden Sie unter Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory.

Um zu überprüfen, ob der Bucket viele versionierte Objekte enthält, verwenden Sie die S3-Storage-Lens-Metriken, um die Anzahl der Objekte der aktuellen Version, die Anzahl nicht aktueller Versionsobjekte und die Anzahl der Löschmarkierungsobjekte anzuzeigen. Weitere Informationen zu Storage-Lens-Metriken finden Sie unter Amazon S3-Storage-Lens-Metrikglossar.

Das Amazon-S3-Team fordert die Kunden auf, Anwendungen zu überprüfen, die wiederholt dasselbe Objekt überschreiben und damit potenziell Millionen Versionen für dieses Objekt erstellen, um

festzustellen, ob die Anwendung wie beabsichtigt funktioniert. Beispielsweise kann eine Anwendung, die eine Woche lang jede Minute dasselbe Objekt überschreibt, über zehntausend Versionen erstellen. Wir empfehlen, für jedes Objekt weniger als einhunderttausend Versionen zu speichern. Wenn Sie einen Anwendungsfall haben, der Millionen von Versionen für ein oder mehrere Objekte erfordert, wenden Sie sich an das AWS Support Team, um Unterstützung bei der Suche nach einer besseren Lösung zu erhalten.

Bewährte Methoden

Um Leistungseinbußen im Zusammenhang mit der Versionsverwaltung zu vermeiden, empfehlen wir Ihnen, den folgenden bewährten Methoden zu folgen:

- Aktivieren Sie eine Lebenszyklusregel, um die vorherigen Versionen von Objekten ablaufen zu lassen. Sie können beispielsweise eine Lebenszyklusregel erstellen, nach der nicht aktuelle Versionen ablaufen, wenn das Objekt 30 Tage nicht mehr aktuell ist. Sie können auch mehrere nicht aktuelle Versionen beibehalten, wenn Sie nicht alle löschen möchten. Weitere Informationen finden Sie unter Festlegen einer S3-Lebenszyklus-Konfiguration.
- Aktivieren Sie eine Lebenszyklusregel, um Löschmarkierungen für abgelaufene Objekte zu löschen, denen keine Datenobjekte im Bucket zugeordnet sind. Weitere Informationen finden Sie unter Löschen abgelaufener Löschmarkierungen für Objekte.

Weitere bewährte Methoden zur Amazon-S3-Leistungsoptimierung finden Sie unter Bewährte Methoden für Designmuster.

Sperren von Objekten mit Object Lock

Mit S3 Object Lock können Sie für einen bestimmten Zeitraum oder auf unbestimmte Zeit verhindern, dass Amazon-S3-Objekte gelöscht oder überschrieben werden. Object Lock verwendet ein write-once-read-many(WORM) -Modell zum Speichern von Objekten. Sie können Object Lock verwenden, um gesetzliche Anforderungen zu erfüllen, die WORM Speicherplatz erfordern, oder um eine weitere Schutzebene gegen Objektänderungen oder das Löschen von Objekten hinzuzufügen.

> **Note**
>
> S3 Object Lock wurde bewertet von Cohasset Associates für den Einsatz in Umgebungen, die den Vorschriften SEC 17a-4,CFTC, und FINRA unterliegen. Weitere Informationen

darüber, wie Object Lock mit diesen Vorschriften zusammenhängt, finden Sie im Cohasset Associates Bewertung der Einhaltung der Vorschriften.

Die Objektsperre stellt zwei Optionen für die Verwaltung der Aufbewahrung von Objekten bereit: Aufbewahrungszeiträume und rechtliche Aufbewahrungsfristen. Für eine Objektversion kann es sowohl einen Aufbewahrungszeitraum als auch eine rechtliche Aufbewahrungsfrist geben oder beides geben.

- Aufbewahrungszeitraum – Ein Aufbewahrungszeitraum gibt einen festen Zeitraum an, für den ein Objekt gesperrt bleibt. Sie können einen eindeutigen Aufbewahrungszeitraum für einzelne Objekte festlegen. Darüber hinaus können Sie eine Standardaufbewahrungsdauer für einen S3-Bucket festlegen. Sie können auch die minimalen und maximalen Aufbewahrungsfristen mit dem `s3:object-lock-remaining-retention-days` Bedingungsschlüssel in der Bucket-Richtlinie einschränken. Auf diese Weise können Sie eine Reihe von Aufbewahrungsfristen festlegen und Aufbewahrungsfristen einschränken, die kürzer oder länger als dieser Zeitraum sein können.
- Gesetzliche Aufbewahrungsfrist – Eine gesetzliche Aufbewahrungsfrist bietet denselben Schutz wie ein Aufbewahrungszeitraum, hat jedoch kein Ablaufdatum. Stattdessen bleibt eine rechtliche Aufbewahrungsfrist gültig, bis Sie diese ausdrücklich entfernen. Rechtliche Aufbewahrungsfristen sind unabhängig von Aufbewahrungsfristen und gelten für einzelne Objektversionen.

Object Lock funktioniert nur in Buckets, für die S3 Versioning aktiviert ist. Wenn Sie eine Objektversion sperren, speichert Amazon S3 die Sperrinformationen in den Metadaten für diese Objektversion. Die Platzierung eines Aufbewahrungszeitraums oder einer rechtlichen Aufbewahrungsfrist für ein Objekt schützt ausschließlich die in der Anforderung angegebene Version. Aufbewahrungsfristen und gesetzliche Aufbewahrungsfristen verhindern nicht, dass neue Versionen des Objekts erstellt werden, und löschen auch keine Markierungen, die über dem Objekt hinzugefügt werden sollen. Weitere Informationen über S3 Versioning finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

Wenn Sie ein Objekt in einem Bucket platzieren, der bereits ein vorhandenes, geschütztes Objekt mit demselben Objektschlüsselnamen enthält, erstellt Amazon S3 eine neue Version dieses Objekts. Die vorhandene, geschützte Version des Objekts bleibt entsprechend seiner Aufbewahrungskonfiguration gesperrt.

So funktioniert die S3-Objektsperre

Themen

- Aufbewahrungszeiträume
- Aufbewahrungsmodi
- Rechtliche Aufbewahrungsfristen
- Bewährte Methoden für die Verwendung von S3 Object Lock
- Erforderliche Berechtigungen

Aufbewahrungszeiträume

Ein Aufbewahrungszeitraum schützt eine Objektversion für einen festen Zeitraum. Wenn Sie für eine Objektversion einen Aufbewahrungszeitraum festlegen, speichert Amazon S3 einen Zeitstempel in den Metadaten der Objektversion, um anzugeben, wann der Aufbewahrungszeitraum abläuft. Nach Ablauf des Aufbewahrungszeitraums kann die Objektversion überschrieben oder gelöscht werden.

Sie können eine Aufbewahrungsfrist explizit für eine einzelne Objektversion oder für die Eigenschaften eines Buckets festlegen, so dass sie automatisch für alle Objekte in dem Bucket gilt. Wenn Sie einen Aufbewahrungszeitraum explizit auf eine Objektversion anwenden, geben Sie ein Retain Until Date (Bis-Aufbewahrungsdatum) für die Objektversion an. Amazon S3 speichert dieses Datum in den Metadaten der Objektversion.

Sie können auch eine Aufbewahrungsfrist in den Eigenschaften eines Buckets festlegen. Wenn Sie eine Aufbewahrungsfrist für einen Bucket festlegen, geben Sie eine Dauer in Tagen oder Jahren an, für die jede im Bucket platzierte Objektversion geschützt werden soll. Wenn Sie ein Objekt in den Bucket legen, berechnet Amazon S3 ein Bis-Aufbewahrungsdatum für die Objektversion, indem es die angegebene Dauer dem Zeitstempel der Erstellung der Objektversion hinzufügt. Die Objektversion wird anschließend so geschützt, als ob Sie für die Objektversion ausdrücklich eine individuelle Sperre mit diesem Aufbewahrungszeitraum angegeben hätten.

> **Note**
>
> Wenn Sie PUT für eine Objektversion mit einem expliziten individuellen Aufbewahrungsmodus und Zeitraum in einem Bucket verwenden, haben die individuellen Object-Lock-Einstellungen der Objektversion Vorrang vor allen Aufbewahrungseinstellungen für Bucket-Eigenschaften.

Wie alle anderen Einstellungen für die Objektsperre gelten Aufbewahrungszeiträume für einzelne Objektversionen. Für verschiedene Versionen desselben Objekts können verschiedene Aufbewahrungsmodi und -zeiträume gelten.

z. B.: Angenommen, Sie haben ein Objekt, von dessen 30-tägigem Aufbewahrungszeitraum 15 Tage abgelaufen sind, und Sie führen die Aktion PUT für ein Objekt in Amazon S3 mit demselben Namen und einem 60-tägigen Aufbewahrungszeitraum durch. In diesem Fall ist die PUT-Anforderung erfolgreich und Amazon S3 erstellt eine neue Version des Objekts mit einem 60-tägigen Aufbewahrungszeitraum. Die ältere Version behält den ursprünglichen Aufbewahrungszeitraum und kann in 15 Tagen gelöscht werden.

Sie können einen Aufbewahrungszeitraum verlängern, nachdem Sie eine Aufbewahrungseinstellung auf eine Objektversion angewendet haben. Senden Sie dazu eine neue Object-Lock-Anforderung für die Objektversion mit einem Aufbewahren bis-Datum, das später liegt als das derzeit für die Objektversion konfigurierte Datum. Amazon S3 ersetzt die bestehende Aufbewahrungsfrist durch den neuen, längeren Zeitraum. Alle Benutzer, die die Berechtigung zur Festlegung von Aufbewahrungszeiträumen für Objekte besitzen, können Aufbewahrungszeiträume für Objektversionen verlängern. Um Aufbewahrungszeitraum festzulegen, müssen Sie die `s3:PutObjectRetention`-Berechtigung besitzen.

Wenn Sie einen Aufbewahrungszeitraum für ein Objekt oder einen S3-Bucket festlegen, müssen Sie einen von zwei Aufbewahrungsmodi auswählen: Compliance oder Governance.

Aufbewahrungsmodi

S3 Object Lock bietet zwei Aufbewahrungsmodi mit unterschiedlichen Schutzebenen für Ihre Objekte:

- Compliance-Modus
- Governance-Modus

Im Compliance-Modus kann eine geschützte Objektversion von keinem Benutzer überschrieben oder gelöscht werden. Dies schließt den Stammbenutzer in Ihrem AWS-Konto ein. Wenn ein Objekt im Compliance-Modus gesperrt wurde, können der Aufbewahrungsmodus nicht geändert und der Aufbewahrungszeitraum nicht verkürzt werden. Der Compliance-Modus stellt sicher, dass eine Objektversion während des Aufbewahrungszeitraums weder überschrieben noch gelöscht werden.

> **Note**
>
> Die einzige Möglichkeit, ein Objekt im Konformitätsmodus vor Ablauf seines Aufbewahrungsdatums zu löschen, besteht darin, das zugehörige Objekt zu löschen AWS-Konto.

Im Governance-Modus können Benutzer eine Objektversion nicht überschreiben oder löschen oder ihre Sperreinstellungen ändern, wenn sie keine speziellen Berechtigungen besitzen. Mit dem Governance-Modus schützen Sie Objekte dagegen, von den meisten Benutzern gelöscht zu werden, Sie können jedoch weiterhin einigen Benutzern die Berechtigung geben, die Aufbewahrungseinstellungen zu ändern oder die Objekte bei Bedarf zu löschen. Sie können den Governance-Modus auch verwenden, um die Einstellungen für Aufbewahrungszeiträume zu testen, bevor Sie einen Aufbewahrungszeitraum im Compliance-Modus erstellen.

Um Aufbewahrungseinstellungen im Governance-Modus zu überschreiben oder zu entfernen, müssen Sie die Berechtigung `s3:BypassGovernanceRetention` besitzen und in jede Anfrage, die ein Überschreiben des Governance-Modus erfordert, ausdrücklich `x-amz-bypass-governance-retention:true` als Anfrage-Header einfügen.

> **Note**
>
> Die Amazon-S3-Konsole enthält standardmäßig den `x-amz-bypass-governance-retention:true`-Header. Wenn Sie versuchen, Objekte zu löschen, die durch den Governance-Modus geschützt sind und über die `s3:BypassGovernanceRetention`-Berechtigung verfügen, verläuft der Vorgang erfolgreich.

Rechtliche Aufbewahrungsfristen

Mit Object Lock können Sie auch eine gesetzliche Aufbewahrungsfrist für eine Objektversion festlegen. Wie Aufbewahrungszeiträume verhindern auch rechtliche Aufbewahrungsfristen das Überschreiben oder Löschen von Objektversionen. Mit gesetzlichen Aufbewahrungsfristen sind jedoch keine Aufbewahrungszeiträume verknüpft. Sie sind gültig, bis sie entfernt werden. Alle Benutzer mit der Berechtigung `s3:PutObjectLegalHold` können rechtliche Aufbewahrungsfristen festlegen und entfernen.

Rechtliche Aufbewahrungsfristen sind von Aufbewahrungszeiträumen unabhängig. Die Festlegung einer rechtlichen Aufbewahrungsfrist für eine Objektversion hat keine Auswirkungen auf den Aufbewahrungsmodus oder -zeitraum für die betreffende Objektversion.

Angenommen, Sie setzen eine gesetzliche Aufbewahrungsfrist für eine Objektversion fest, während diese Objektversion gleichzeitig durch einen Aufbewahrungszeitraum geschützt ist. Wenn die Aufbewahrungsfrist abläuft, verliert das Objekt seinen WORM Schutz nicht. Stattdessen wird das Objekt durch die gesetzliche Aufbewahrungsfrist weiter geschützt, bis ein autorisierter Benutzer diese ausdrücklich entfernt. Wenn Sie eine rechtliche Aufbewahrungsfrist entfernen, während für eine Objektversion ein Aufbewahrungszeitraum gilt, bleibt die Objektversion geschützt, bis der Aufbewahrungszreitraum abläuft.

Bewährte Methoden für die Verwendung von S3 Object Lock

Ziehen Sie den Governance-Modus in Betracht, wenn Sie verhindern möchten, dass Objekte während eines vordefinierten Aufbewahrungszeitraums von den meisten Benutzern gelöscht werden, aber gleichzeitig möchten, dass einige Benutzer mit speziellen Berechtigungen die Flexibilität haben, die Aufbewahrungseinstellungen zu ändern oder die Objekte zu löschen.

Ziehen Sie den Compliance-Modus in Betracht, wenn Sie nicht möchten, dass ein Benutzer, einschließlich des Root-Benutzers AWS-Konto, die Objekte während eines vordefinierten Aufbewahrungszeitraums löschen kann. Sie können diesen Modus verwenden, falls Sie konforme Daten speichern müssen.

Sie können Legal Hold verwenden, wenn Sie sich nicht sicher sind, wie lange Ihre Objekte unveränderlich bleiben sollen. Das könnte daran liegen, dass Sie eine bevorstehende externe Prüfung Ihrer Daten haben und Objekte bis zum Abschluss der Prüfung unveränderlich lassen möchten. Alternativ haben Sie möglicherweise ein laufendes Projekt, das einen Datensatz verwendet, den Sie bis zum Abschluss des Projekts unveränderlich behalten möchten.

Erforderliche Berechtigungen

ObjektsperrenVorgänge erfordern bestimmte Berechtigungen. Abhängig von dem genauen Vorgang, den Sie versuchen, benötigen Sie möglicherweise eine der folgenden Berechtigungen:

- `s3:BypassGovernanceRetention`
- `s3:GetBucketObjectLockConfiguration`
- `s3:GetObjectLegalHold`
- `s3:GetObjectRetention`

- `s3:PutBucketObjectLockConfiguration`
- `s3:PutObjectLegalHold`
- `s3:PutObjectRetention`

Eine vollständige Liste der Amazon S3-Berechtigungen mit Beschreibungen finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference.

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

Informationen zur Verwendung von Bedingungen mit Berechtigungen finden Sie unter Beispiele für Bucket-Richtlinien mit Bedingungsschlüsseln.

Überlegungen zu Object Lock

Amazon S3 Object Lock kann verhindern, dass Objekte für einen bestimmten Zeitraum oder auf unbestimmte Zeit gelöscht oder überschrieben werden.

Sie können die Amazon S3-Konsole, AWS Command Line Interface (AWS CLI) oder Amazon S3 verwenden AWS SDKs, REST API um Object Lock-Informationen anzuzeigen oder festzulegen. Weitere Informationen zu den Funktionen von S3 Object Lock finden Sie unter Sperren von Objekten mit Object Lock.

> **Important**
>
> - Wenn Sie Object Lock für einen Bucket aktiviert haben, können Sie Object Lock nicht deaktivieren oder die Versionsverwaltung für diesen Bucket aussetzen.
> - S3-Buckets mit Object Lock können nicht als Ziel-Buckets für Server-Zugriffsprotokolle verwendet werden. Weitere Informationen finden Sie unter the section called “Protokollierungs-Serverzugriff”.

Themen

- Berechtigungen zum Anzeigen von Sperrinformationen
- Umgehen des Governance-Modus
- Verwenden von Object Lock mit der S3-Replikation

- Verwenden von Object Lock mit Verschlüsselung
- Verwenden von Object Lock mit Amazon S3 Inventory
- Verwaltung von S3-Lebenszyklusrichtlinien mit Object Lock
- Verwaltung von Löschmarkierungen mit Object Lock
- Verwenden von S3 Storage Lens mit Object Lock
- Objekte in einen Bucket mit aktivierter Objektsperre hochladen
- Konfigurieren von Ereignissen und Benachrichtigungen
- Festlegen von Beschränkungen für Aufbewahrungsfristen mit einer Bucket-Richtlinie

Berechtigungen zum Anzeigen von Sperrinformationen

Sie können den Object Lock-Status einer Amazon S3 S3-Objektversion programmgesteuert anzeigen, indem Sie den HeadObject oder GetObjectOperationen. Beide Vorgänge geben den Aufbewahrungsmodus, die Aufbewahrungsfrist und den Status in Bezug auf eine rechtliche Aufbewahrungsfrist für die angegebene Objektversion an. Darüber hinaus können Sie mithilfe von S3 Inventory den Status der Objektsperre für mehrere Objekte in Ihrem S3-Bucket anzeigen.

Um Aufbewahrungsmodus und Aufbewahrungszeitraum für eine Objektversion anzuzeigen, müssen Sie die Berechtigung `s3:GetObjectRetention` besitzen. Um den Status eines Objekts in Bezug auf rechtliche Aufbewahrungsfristen anzuzeigen, müssen Sie die Berechtigung `s3:GetObjectLegalHold` besitzen. Um die Standardaufbewahrungskonfiguration eines Buckets anzuzeigen, benötigen Sie die `s3:GetBucketObjectLockConfiguration`-Berechtigung. Wenn Sie eine Object-Lock-Konfiguration für einen Bucket anfordern, für den S3 Object Lock nicht aktiviert ist, gibt Amazon S3 einen Fehler zurück.

Umgehen des Governance-Modus

Sie können für im Governance-Modus gesperrte Objektversionen, Vorgänge ausführen, als ob sie nicht geschützt wären, wenn Sie die `s3:BypassGovernanceRetention`-Berechtigung besitzen. Zu diesen Vorgängen gehören das Löschen einer Objektversion, das Verkürzen des Aufbewahrungszeitraums oder das Entfernen des Object-Lock-Aufbewahrungszeitraums durch die Platzierung einer neuen `PutObjectRetention`-Anforderung mit leeren Parametern.

Um den Governance-Modus umgehen zu können, müssen Sie in Ihrer Anforderung ausdrücklich angeben, dass Sie den Governance-Modus umgehen möchten. Fügen Sie dazu den `x-amz-bypass-governance-retention:true` Header Ihrer `PutObjectRetention` API Operationsanforderung hinzu oder verwenden Sie den entsprechenden Parameter bei

Anfragen, die über AWS CLI oder gestellt werden AWS SDKs. Die S3-Konsole übernimmt diesen Header automatisch für über die S3-Konsole gestellte Anforderungen, wenn Sie über die `s3:BypassGovernanceRetention`-Berechtigung verfügen, den Governance-Modus zu umgehen.

> **Note**
>
> Die Umgehung des Governance-Modus hat keine Auswirkungen auf den Status einer Objektversion in Bezug auf rechtliche Aufbewahrungsfristen. Wenn für eine Objektversion eine rechtliche Aufbewahrungsfrist aktiviert ist, bleibt diese in Kraft und verhindert, dass die Objektversion durch Anforderungen überschrieben oder gelöscht wird.

Verwenden von Object Lock mit der S3-Replikation

Sie können Object Lock mit S3-Replikation verwenden, um automatisches asynchrones Kopieren von gesperrten Objekten und deren Aufbewahrungs-Metadaten über S3-Buckets hinweg zu aktivieren. Das bedeutet, dass Amazon S3 für replizierte Objekte die Objektsperrkonfiguration des Quell-Buckets verwendet. Mit anderen Worten, wenn im Quell-Bucket Object Lock aktiviert ist, muss in den Ziel-Buckets auch Object Lock aktiviert sein. Wenn ein Objekt direkt in den Ziel-Bucket hochgeladen wird (außerhalb der S3-Replikation), verwendet es die für den Ziel-Bucket festgelegte Objektsperre. Wenn Sie Replikation verwenden, werden Objekte aus einem Quell-Bucket in einen oder mehrere Ziel-Bucket(s) repliziert.

Um die Replikation für einen Bucket mit aktivierter Objektsperre einzurichten, können Sie die S3-Konsole AWS CLI, Amazon S3 REST API oder verwenden AWS SDKs.

> **Note**
>
> Um Object Lock mit der Replikation zu verwenden, müssen Sie zwei zusätzliche Berechtigungen für den S3-Quell-Bucket in der Rolle AWS Identity and Access Management (IAM) gewähren, mit der Sie die Replikation einrichten. Die zwei neuen Berechtigungen sind `s3:GetObjectRetention` und `s3:GetObjectLegalHold`. Wenn die Rolle über eine `s3:Get*`-Berechtigungsanweisung verfügt, ist die Anforderung dadurch erfüllt. Weitere Informationen finden Sie unter Berechtigungen für die Live-Replikation einrichten.
> Allgemeine Informationen zur S3-Replikation finden Sie unter Objekte innerhalb und zwischen Regionen replizieren.
> Beispiele für die Einrichtung der S3-Replikation finden Sie unter Beispiele für die Konfiguration der Live-Replikation.

Verwenden von Object Lock mit Verschlüsselung

Amazon S3 verschlüsselt standardmäßig alle neuen Objekte. Sie können Object Lock mit Ihren verschlüsselten Objekten verwenden. Weitere Informationen finden Sie unter Datenschutz durch Verschlüsselung.

Object Lock kann zwar verhindern, dass Amazon S3 S3-Objekte gelöscht oder überschrieben werden, schützt aber nicht davor, den Zugriff auf die Verschlüsselungsschlüssel zu verlieren oder Verschlüsselungsschlüssel zu löschen. Wenn Sie beispielsweise Ihre Objekte mit AWS KMS serverseitiger Verschlüsselung verschlüsseln und Ihr AWS KMS Schlüssel gelöscht wird, können Ihre Objekte unlesbar werden.

Verwenden von Object Lock mit Amazon S3 Inventory

Sie können Amazon S3 Inventory so konfigurieren, dass Listen der Objekte in einem S3-Bucket nach einem definierten Zeitplan erstellt werden. Sie können Amazon S3 Inventory so konfigurieren, dass es die folgenden Object-Lock-Metadaten für Ihre Objekte enthält:

- Das Aufbewahrungsdatum
- Der Aubewahrungsmodus
- Die gesetzliche Aufbewahrungsfrist

Weitere Informationen finden Sie unter Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory.

Verwaltung von S3-Lebenszyklusrichtlinien mit Object Lock

Konfigurationen für die Verwaltung des Objektlebenszyklus funktionieren für geschützte Objekte weiterhin normal. Dies schließt die Platzierung von Löschmarkierungen ein. Eine gesperrte Version eines Objekts kann jedoch nicht durch eine S3-Lifecycle-Ablaufrichtlinie gelöscht werden. Object Lock wird unabhängig davon beibehalten, in welcher Speicherklasse sich das Objekt befindet, und während des gesamten S3 Lifecycle wird zwischen Speicherklassen gewechselt.

Weitere Informationen zur Verwaltung von Objektlebenszyklen finden Sie unter Verwaltung des Lebenszyklus von Objekten.

Verwaltung von Löschmarkierungen mit Object Lock

Sie können eine geschützte Objektversion zwar nicht löschen, aber eine Löschmarkierung für das betreffende Objekt erstellen. Durch das Setzen einer Löschmarkierung auf einem Objekt wird keine

Objektversion gelöscht. Amazon S3 verhält sich dadurch aber zumeist so, als ob das Objekt gelöscht worden wäre. Weitere Informationen finden Sie unter Arbeiten mit Löschmarkierungen.

> **Note**
>
> Löschmarken sind nicht WORM geschützt, unabhängig von der Aufbewahrungsfrist oder der gesetzlichen Aufbewahrungsfrist für das zugrunde liegende Objekt.

Verwenden von S3 Storage Lens mit Object Lock

Wwnn Sie Metriken für objektsperrefähige Speicherbytes und die Anzahl der Objekte sehen möchten, können Sie Amazon S3 Storage Lens verwenden. S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können.

Weitere Informationen finden Sie unter Verwenden von S3 Storage Lens zum Schutz Ihrer Daten.

Eine vollständige Liste der Metriken finden Sie unter Amazon S3-Storage-Lens-Metrikglossar.

Objekte in einen Bucket mit aktivierter Objektsperre hochladen

Der `x-amz-sdk-checksum-algorithm` Header `Content-MD5` oder ist für jede Anfrage zum Hochladen eines Objekts mit einer mit Object Lock konfigurierten Aufbewahrungsfrist erforderlich. Diese Header sind eine Möglichkeit, die Integrität Ihres Objekts während des Uploads zu überprüfen.

Beim Hochladen eines Objekts mit der Amazon S3 S3-Konsole fügt S3 automatisch den `Content-MD5` Header hinzu. Sie können optional eine zusätzliche Prüfsummenfunktion und einen Prüfsummenwert über die Konsole als Header angeben. `x-amz-sdk-checksum-algorithm` Wenn Sie den verwenden, müssen PutObjectAPISie den `Content-MD5` Header, den `x-amz-sdk-checksum-algorithm` Header oder beides angeben, um die Aufbewahrungsfrist für Object Lock zu konfigurieren.

Weitere Informationen finden Sie unter Überprüfung der Objektintegrität.

Konfigurieren von Ereignissen und Benachrichtigungen

Sie können Amazon S3 Event Notifications verwenden, um den Zugriff und die Änderungen an Ihren Object Lock-Konfigurationen und -Daten nachzuverfolgen, indem Sie AWS CloudTrail Weitere Informationen dazu CloudTrail finden Sie unter Was ist AWS CloudTrail? im AWS CloudTrail Benutzerhandbuch.

Sie können Amazon auch verwenden CloudWatch , um Benachrichtigungen auf der Grundlage dieser Daten zu generieren. Informationen zu CloudWatch finden Sie unter Was ist Amazon CloudWatch? im CloudWatch Amazon-Benutzerhandbuch.

Festlegen von Beschränkungen für Aufbewahrungsfristen mit einer Bucket-Richtlinie

Sie können mittels einer Bucket-Richtlinie mindestens erforderliche und maximal zulässige Aufbewahrungszeiträume für einen Bucket festlegen. Die maximale Aufbewahrungsfrist beträgt 100 Jahre.

Das folgende Beispiel zeigt eine Bucket-Richtlinie, die den Bedingungsschlüssel `s3:object-lock-remaining-retention-days` verwendet, um einen maximalen Aufbewahrungszeitraum von 10 Tagen festzulegen.

```
{
    "Version": "2012-10-17",
    "Id": "SetRetentionLimits",
    "Statement": [
        {
            "Sid": "SetRetentionPeriod",
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "s3:PutObjectRetention"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket1/*",
            "Condition": {
                "NumericGreaterThan": {
                    "s3:object-lock-remaining-retention-days": "10"
                }
            }
        }
    ]
}
```

> **Note**
>
> Wenn es sich bei Ihrem Bucket um den Ziel-Bucket einer Replikationsrichtlinie handelt, könnenSie minimal und maximal zulässige Aufbewahrungszeiträume für Objektreplikate einrichten möchten, die mittels Replikation erstellt werden. Hierzu müssen Sie die `s3:ReplicateObject`-Aktion in Ihrer Bucket-Richtlinie zulassen. Weitere Informationen zur

Verwendung von Replikationsberechtigungen finden Sie unter the section called “Einrichten von Berechtigungen”.

Weitere Informationen zu Bucket-Richtlinien finden Sie in den folgenden Themen:

- Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference

  Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.
- Objektoperationen
- Beispiele für Bucket-Richtlinien mit Bedingungsschlüsseln

Konfigurieren von S3 Object Lock

Mit Amazon S3 Object Lock können Sie Objekte in Amazon S3 speichern, indem Sie ein write-once-read-many(WORM) -Modell verwenden. Mit der S3-Objektsperre können Sie für einen festen Zeitraum oder auf unbegrenzte Zeit verhindern, dass ein Objekt gelöscht oder überschrieben wird. Allgemeine Informationen zu den Funktionen von Object Lock finden Sie unter Sperren von Objekten mit Object Lock.

Bevor Sie Objekte sperren können, müssen Sie die S3-Versionsverwaltung und Object Lock für einen Bucket aktivieren. Anschließend können Sie einen Aufbewahrungszeitraum, eine gesetzliche Aufbewahrungsfrist oder beides festlegen.

Um mit Object Lock zu arbeiten, müssen Sie über bestimmte Berechtigungen verfügen. Eine Liste der Berechtigungen für verschiedene Object-Lock-Vorgänge finden Sie unter the section called “Erforderliche Berechtigungen”.

⚠ Important

- Wenn Sie Object Lock für einen Bucket aktiviert haben, können Sie Object Lock nicht deaktivieren oder die Versionsverwaltung für diesen Bucket aussetzen.
- S3-Buckets mit Object Lock können nicht als Ziel-Buckets für Server-Zugriffsprotokolle verwendet werden. Weitere Informationen finden Sie unter the section called “Protokollierungs-Serverzugriff”.

Themen

- Aktivieren von Object Lock beim Erstellen eines neuen S3-Buckets
- Aktivieren von Object Lock für einen vorhandenen S3-Bucket
- Einrichten oder Ändern einer gesetzlichen Aufbewahrungsfrist für ein S3-Objekt
- Einrichten oder Ändern eines Aufbewahrungszeitraums für ein S3-Objekt
- Einrichten oder Ändern eines Standard-Aufbewahrungszeitraums für einen S3-Bucket

Aktivieren von Object Lock beim Erstellen eines neuen S3-Buckets

Sie können Object Lock aktivieren, wenn Sie einen neuen S3-Bucket erstellen, indem Sie die Amazon S3-Konsole, AWS Command Line Interface (AWS CLI) AWS SDKs, oder Amazon S3 verwenden RESTAPI.

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie Create Bucket (Bucket erstellen) aus.

   Anschließend wird die Seite Bucket erstellen geöffnet.
4. Geben Sie unter Bucket Name (Bucket-Name) einen Namen für den Bucket ein.

   > **Note**
   >
   > Der Name eines einmal erstellten Buckets kann nicht nachträglich geändert werden. Weitere Informationen zur Benennung von Buckets finden Sie unter Allgemeine Benennungsregeln für Buckets.
5. Wählen Sie unter Region den AWS-Region Ort aus, an dem sich der Bucket befinden soll.
6. Wählen Sie unter Objekteigentümer aus, ob Sie Zugriffskontrolllisten (ACLs) deaktivieren oder aktivieren und den Besitz von Objekten kontrollieren möchten, die in Ihren Bucket hochgeladen wurden.
7. Wählen Sie unter Einstellungen "Öffentlichen Zugriff beschränken" für diesen Bucket die Einstellungen zum Beschränken des öffentlichen Zugriffs aus, die Sie auf den Bucket anwenden möchten.

8. Wählen Sie unter Bucket-Versionsverwaltung die Option Aktiviert.

   Object Lock funktioniert nur mit versionsgesteuerten Buckets.

9. (Optional) Unter Tags können Sie auswählen, ob Sie Ihrem Bucket Tags hinzufügen möchten. Tags sind Schlüssel-Wert-Paare, die zur Kategorisierung von Speicher und zur Zuweisung von Kosten verwendet werden.
10. Suchen Sie unter Erweiterte Einstellungen nach Object Lock und wählen Sie Aktivieren aus.

    Sie müssen sich darüber im Klaren sein, dass durch die Aktivierung von Object Lock Objekte in diesem Bucket dauerhaft gesperrt werden können.

11. Wählen Sie Bucket erstellen aus.

Verwenden Sie den AWS CLI

Im folgenden `create-bucket`-Beispiel wird ein neuer S3-Bucket mit dem Namen *`amzn-s3-demo-bucket1`* und aktiviertem Object Lock erstellt:

```
aws s3api create-bucket --bucket amzn-s3-demo-bucket1 --object-lock-enabled-for-bucket
```

Weitere Informationen und Beispiele finden Sie unter create-bucketin der AWS CLI Befehlsreferenz.

> **Note**
>
> Sie können AWS CLI Befehle von der Konsole aus ausführen, indem Sie AWS CloudShell AWS CloudShell ist eine browserbasierte, vorauthentifizierte Shell, die Sie direkt von der aus starten können. AWS Management ConsoleWeitere Informationen finden Sie unter Was ist? CloudShell im AWS CloudShell Benutzerhandbuch.

Mit dem REST API

Sie können den verwenden RESTAPI, um einen neuen S3-Bucket mit aktivierter Objektsperre zu erstellen. Weitere Informationen finden Sie unter CreateBucketin der Amazon Simple Storage Service API Reference.

Mit dem AWS SDKs

Beispiele für die Aktivierung von Object Lock beim Erstellen eines neuen S3-Buckets mit dem finden Sie in den AWS SDKs Codebeispielen in der Amazon S3 API S3-Referenz.

Beispiele dafür, wie Sie die aktuelle Object Lock-Konfiguration mit dem abrufen können AWS SDKs, finden Sie in den Codebeispielen in der Amazon S3 API S3-Referenz.

Ein interaktives Szenario, das verschiedene Object Lock-Funktionen mithilfe von demonstriert AWS SDKs, finden Sie in den Codebeispielen in der Amazon S3 API S3-Referenz.

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS SDKs in der Amazon S3 API S3-Referenz.

Aktivieren von Object Lock für einen vorhandenen S3-Bucket

Sie können Object Lock für einen vorhandenen S3-Bucket aktivieren, indem Sie die Amazon S3-Konsole AWS CLI, den AWS SDKs, oder Amazon S3 verwenden RESTAPI.

Verwenden der S3-Konsole

> **Note**
>
> Object Lock funktioniert nur mit versionsgesteuerten Buckets.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie Object Lock aktivieren möchten.
4. Wählen Sie die Registerkarte Eigenschaften aus.
5. Scrollen Sie unter Eigenschaften nach unten zum Bereich Object Lock und wählen Sie Bearbeiten aus.
6. Wählen Sie unter Object Lock die Option Aktivieren aus.

   Sie müssen sich darüber im Klaren sein, dass durch die Aktivierung von Object Lock Objekte in diesem Bucket dauerhaft gesperrt werden können.
7. Wählen Sie Änderungen speichern.

Verwenden Sie den AWS CLI

Mit dem folgenden `put-object-lock-configuration`-Beispielbefehl wird eine Aufbewahrungsfrist von 50 Tagen für Object Lock für einen Bucket mit dem Namen *amzn-s3-demo-bucket1* festgelegt:

```
aws s3api put-object-lock-configuration --bucket amzn-s3-demo-bucket1 --object-
lock-configuration='{ "ObjectLockEnabled": "Enabled", "Rule": { "DefaultRetention":
 { "Mode": "COMPLIANCE", "Days": 50 }}}'
```

Weitere Informationen und Beispiele finden Sie unter put-object-lock-configurationin der AWS CLI Befehlsreferenz.

> **Note**
>
> Sie können AWS CLI Befehle von der Konsole aus ausführen, indem Sie AWS CloudShell AWS CloudShell ist eine browserbasierte, vorauthentifizierte Shell, die Sie direkt von der aus starten können. AWS Management ConsoleWeitere Informationen finden Sie unter Was ist? CloudShell im AWS CloudShell Benutzerhandbuch.

Mit dem REST API

Sie können Amazon S3 verwenden RESTAPI, um Object Lock für einen vorhandenen S3-Bucket zu aktivieren. Weitere Informationen finden Sie unter PutObjectLockConfigurationin der Amazon Simple Storage Service API Reference.

Mit dem AWS SDKs

Beispiele dafür, wie Sie Object Lock für einen vorhandenen S3-Bucket mit dem aktivieren AWS SDKs, finden Sie in den Codebeispielen in der Amazon S3 API S3-Referenz.

Beispiele dafür, wie Sie die aktuelle Object Lock-Konfiguration mit dem abrufen können AWS SDKs, finden Sie in den Codebeispielen in der Amazon S3 API S3-Referenz.

Ein interaktives Szenario, das verschiedene Object Lock-Funktionen mithilfe von demonstriert AWS SDKs, finden Sie in den Codebeispielen in der Amazon S3 API S3-Referenz.

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS SDKs in der Amazon S3 API S3-Referenz.

Einrichten oder Ändern einer gesetzlichen Aufbewahrungsfrist für ein S3-Objekt

Sie können eine gesetzliche Sperre für ein S3-Objekt mithilfe der Amazon S3-Konsole, AWS CLI AWS SDKs, oder Amazon S3 festlegen oder entfernen RESTAPI.

> **Important**
>
> - Wenn Sie eine gesetzliche Aufbewahrungsfrist für ein Objekt festlegen möchten, muss Object Lock für den Bucket des Objekts bereits aktiviert sein.
> - Wenn Sie PUT für eine Objektversion mit einem expliziten individuellen Aufbewahrungsmodus und Zeitraum in einem Bucket verwenden, haben die individuellen Object-Lock-Einstellungen der Objektversion Vorrang vor allen Aufbewahrungseinstellungen für Bucket-Eigenschaften.

Weitere Informationen finden Sie unter the section called “Rechtliche Aufbewahrungsfristen”.

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält, für das Sie eine gesetzliche Aufbewahrungsfrist einrichten oder ändern möchten.
4. Wähle Sie in der Liste Objekte das Objekt aus, für das Sie eine gesetzliche Aufbewahrungsfrist festlegen oder ändern möchten.
5. Suchen Sie auf der Seite mit den Objekteigenschaften den Bereich Object Lock – Gesetzliche Aufbewahrungsfristen und wählen Sie Bearbeiten.
6. Wählen Sie Aktivieren, um eine gesetzliche Aufbewahrungsfrist festzulegen, oder Deaktivieren, um eine gesetzliche Aufbewahrungsfrist zu entfernen.
7. Wählen Sie Änderungen speichern.

Verwenden Sie den AWS CLI

Im folgenden `put-object-legal-hold` Beispiel wird ein legaler Hold-Wert für das Objekt festgelegt *`my-image.fs`* in dem Bucket mit dem Namen: *`amzn-s3-demo-bucket1`*

```
aws s3api put-object-legal-hold --bucket amzn-s3-demo-bucket1 --key my-image.fs --
legal-hold="Status=ON"
```

Im folgenden `put-object-legal-hold` Beispiel wird eine gesetzliche Sperre für das Objekt aufgehoben *my-image.fs* in dem Bucket mit dem Namen: *amzn-s3-demo-bucket1*

```
aws s3api put-object-legal-hold --bucket amzn-s3-demo-bucket1 --key my-image.fs --
legal-hold="Status=OFF"
```

Weitere Informationen und Beispiele finden Sie unter put-object-legal-holdin der AWS CLI Befehlsreferenz.

> **Note**
>
> Sie können AWS CLI Befehle von der Konsole aus ausführen, indem Sie AWS CloudShell AWS CloudShell ist eine browserbasierte, vorauthentifizierte Shell, die Sie direkt von der aus starten können. AWS Management ConsoleWeitere Informationen finden Sie unter Was ist? CloudShell im AWS CloudShell Benutzerhandbuch.

Mit dem REST API

Sie können den verwenden RESTAPI, um eine gesetzliche Sperre für ein Objekt festzulegen oder zu ändern. Weitere Informationen finden Sie unter PutObjectLegalHoldin der Amazon Simple Storage Service API Reference.

Mit dem AWS SDKs

Beispiele dafür, wie Sie mit dem eine gesetzliche Aufbewahrungsfrist für ein Objekt einrichten können AWS SDKs, finden Sie in den Codebeispielen in der Amazon S3 API S3-Referenz.

Beispiele dafür, wie Sie den aktuellen gesetzlichen Aufbewahrungsstatus mit dem abrufen können AWS SDKs, finden Sie in den Codebeispielen in der Amazon S3 API S3-Referenz.

Ein interaktives Szenario, das verschiedene Object Lock-Funktionen mithilfe von demonstriert AWS SDKs, finden Sie in den Codebeispielen in der Amazon S3 API S3-Referenz.

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS SDKs in der Amazon S3 API S3-Referenz.

Einrichten oder Ändern eines Aufbewahrungszeitraums für ein S3-Objekt

Sie können einen Aufbewahrungszeitraum für ein S3-Objekt mithilfe der Amazon S3-Konsole, AWS CLI AWS SDKs, oder Amazon S3 festlegen oder ändern RESTAPI.

> **Important**
>
> - Wenn Sie einen Aufbewahrungszeitraum für ein Objekt festlegen möchten, muss Object Lock für den Bucket des Objekts bereits aktiviert sein.
> - Wenn Sie PUT für eine Objektversion mit einem expliziten individuellen Aufbewahrungsmodus und Zeitraum in einem Bucket verwenden, haben die individuellen Object-Lock-Einstellungen der Objektversion Vorrang vor allen Aufbewahrungseinstellungen für Bucket-Eigenschaften.
> - Die einzige Möglichkeit, ein Objekt im Compliance-Modus vor Ablauf seines Aufbewahrungsdatums zu löschen, besteht darin, das zugehörige Objekt zu löschen AWS-Konto.

Weitere Informationen finden Sie unter Aufbewahrungszeiträume.

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der das Objekt enthält, für das Sie einen Aufbewahrungszeitraum einrichten oder ändern möchten.
4. Wähle Sie in der Liste Objekte das Objekt aus, für das Sie einen Aufbewahrungszeitraum festlegen oder ändern möchten.
5. Suchen Sie auf der Seite mit den Objekteigenschaften den Bereich Object Lock – Aufbewahrungszeitraum und wählen Sie Bearbeiten.
6. Wählen Sie unter Aufbewahrung die Option Aktivieren aus, um einen Aufbewahrungszeitraum festzulegen, oder Deaktivieren, um einen Aufbewahrungszeitraum zu entfernen.
7. Wenn Sie Aktivieren unter Aufbewahrungsmodus ausgewählt haben, wählen Sie entweder Governance-Modus oder Compliance-Modus. Weitere Informationen finden Sie unter Aufbewahrungsmodi.

8. Wählen Sie unter Aufbewahren bis das Datum aus, an dem der Aufbewahrungszeitraum enden soll. Während dieses Zeitraums ist Ihr Objekt WORM geschützt und kann nicht überschrieben oder gelöscht werden. Weitere Informationen finden Sie unter Aufbewahrungszeiträume.
9. Wählen Sie Save Changes (Änderungen speichern).

Verwenden Sie den AWS CLI

Im folgenden `put-object-retention` Beispiel wird eine Aufbewahrungsfrist für das Objekt festgelegt *`my-image.fs`* in dem Bucket, der *`amzn-s3-demo-bucket1`* bis zum 1. Januar 2025 benannt wurde:

```
aws s3api put-object-retention --bucket amzn-s3-demo-bucket1 --key my-image.fs --
retention='{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

Weitere Informationen und Beispiele finden Sie unter put-object-retentionin der AWS CLI Befehlsreferenz.

> **Note**
>
> Sie können AWS CLI Befehle von der Konsole aus ausführen, indem Sie AWS CloudShell AWS CloudShell ist eine browserbasierte, vorauthentifizierte Shell, die Sie direkt von der aus starten können. AWS Management ConsoleWeitere Informationen finden Sie unter Was ist? CloudShell im AWS CloudShell Benutzerhandbuch.

Mit dem REST API

Sie können den verwenden RESTAPI, um eine Aufbewahrungsfrist für ein Objekt festzulegen. Weitere Informationen finden Sie unter PutObjectRetentionin der Amazon Simple Storage Service API Reference.

Mit dem AWS SDKs

Beispiele dafür, wie Sie mit dem eine Aufbewahrungsfrist für ein Objekt festlegen AWS SDKs, finden Sie in den Codebeispielen in der Amazon S3 API S3-Referenz.

Beispiele dafür, wie Sie die Aufbewahrungsdauer für ein Objekt mit dem ermitteln können AWS SDKs, finden Sie in den Codebeispielen in der Amazon S3 API S3-Referenz.

Ein interaktives Szenario, das verschiedene Object Lock-Funktionen mithilfe von demonstriert AWS SDKs, finden Sie in den Codebeispielen in der Amazon S3 API S3-Referenz.

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS SDKs in der Amazon S3 API S3-Referenz.

Einrichten oder Ändern eines Standard-Aufbewahrungszeitraums für einen S3-Bucket

Sie können eine Standardaufbewahrungsdauer für einen S3-Bucket mithilfe der Amazon S3-Konsole, AWS CLI AWS SDKs, oder Amazon S3 festlegen oder ändern RESTAPI. Sie geben eine Dauer in Tagen oder Jahren an, für die jede in dem Bucket befindliche Objektversion geschützt werden soll.

> **Important**
>
> - Wenn Sie einen Standard-Aufbewahrungszeitraum für einen Bucket festlegen möchten, muss Object Lock für den Bucket bereits aktiviert sein.
> - Wenn Sie für eine Objektversion mit einem expliziten individuellen Aufbewahrungsmodus und Zeitraum in einem Bucket PUT durchführen, haben die individuellen Object-Lock-Einstellungen der Objektversion Vorrang vor allen Aufbewahrungseinstellungen für Bucket-Eigenschaften.
> - Die einzige Möglichkeit, ein Objekt im Compliance-Modus vor Ablauf seines Aufbewahrungsdatums zu löschen, besteht darin, das zugehörige Objekt zu löschen AWS-Konto.

Weitere Informationen finden Sie unter Aufbewahrungszeiträume.

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie einen Aufbewahrungszeitraum einrichten oder ändern möchten.
4. Wählen Sie die Registerkarte Eigenschaften aus.
5. Scrollen Sie unter Eigenschaften nach unten zum Bereich Object Lock und wählen Sie Bearbeiten aus.

6. Wählen Sie unter Standard-Aufbewahrung die Option Aktivieren aus, um einen Standard-Aufbewahrungszeitraum festzulegen, oder Deaktivieren, um einen Standard-Aufbewahrungszeitraum zu entfernen.
7. Wenn Sie Aktivieren unter Aufbewahrungsmodus ausgewählt haben, wählen Sie entweder Governance-Modus oder Compliance-Modus. Weitere Informationen finden Sie unter Aufbewahrungsmodi.
8. Wählen Sie unter Standard-Aufbewahrungszeitraum die Anzahl der Tage oder Jahre aus, für die der Aufbewahrungszeitraum gelten soll. Objekte, die sich in diesem Bucket befinden, werden für diese Anzahl von Tagen oder Jahren gesperrt. Weitere Informationen finden Sie unter Aufbewahrungszeiträume.
9. Wählen Sie Save Changes (Änderungen speichern).

Verwenden Sie den AWS CLI

Mit dem folgenden `put-object-lock-configuration`-Beispielbefehl wird eine Object-Lock-Aufbewahrungsfrist von 50 Tag für einen Bucket mit dem Namen *amzn-s3-demo-bucket1* unter Verwendung des Compliance-Modus festgelegt:

```
aws s3api put-object-lock-configuration --bucket amzn-s3-demo-bucket1 --object-
lock-configuration='{ "ObjectLockEnabled": "Enabled", "Rule": { "DefaultRetention":
 { "Mode": "COMPLIANCE", "Days": 50 }}}'
```

Im folgenden `put-object-lock-configuration`-Beispiel wird die Standard-Aufbewahrungskonfiguration für einen Bucket entfernt:

```
aws s3api put-object-lock-configuration --bucket amzn-s3-demo-bucket1 --object-lock-
configuration='{ "ObjectLockEnabled": "Enabled"}'
```

Weitere Informationen und Beispiele finden Sie unter put-object-lock-configurationin der AWS CLI Befehlsreferenz.

> **Note**
>
> Sie können AWS CLI Befehle von der Konsole aus ausführen, indem Sie AWS CloudShell AWS CloudShell ist eine browserbasierte, vorauthentifizierte Shell, die Sie direkt von der aus starten können. AWS Management ConsoleWeitere Informationen finden Sie unter Was ist? CloudShell im AWS CloudShell Benutzerhandbuch.

Mit dem REST API

Sie können den verwenden RESTAPI, um einen Standard-Aufbewahrungszeitraum für einen vorhandenen S3-Bucket festzulegen. Weitere Informationen finden Sie unter PutObjectLockConfigurationin der Amazon Simple Storage Service API Reference.

Mit dem AWS SDKs

Beispiele dafür, wie Sie mit dem eine Standardaufbewahrungsdauer für einen vorhandenen S3-Bucket festlegen AWS SDKs, finden Sie in den Codebeispielen in der Amazon S3 API S3-Referenz.

Ein interaktives Szenario, das verschiedene Object Lock-Funktionen mithilfe von demonstriert AWS SDKs, finden Sie in den Codebeispielen in der Amazon S3 API S3-Referenz.

Allgemeine Informationen zur Verwendung verschiedener AWS SDKs finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS SDKs in der Amazon S3 API S3-Referenz.

Sicherung Ihrer Amazon S3 S3-Daten

Amazon S3 ist nativ integriert AWS Backup, ein vollständig verwalteter, richtlinienbasierter Service, mit dem Sie Backup-Richtlinien zum Schutz Ihrer Daten in Amazon S3 zentral definieren können. Nachdem Sie Ihre Backup-Richtlinien definiert und Amazon S3 S3-Ressourcen den Richtlinien zugewiesen haben, AWS Backup automatisiert die Erstellung von Amazon S3 S3-Backups und speichert die Backups sicher in einem verschlüsselten Backup-Tresor, den Sie in Ihrem Backup-Plan angeben.

Bei der Verwendung AWS Backup für Amazon S3 können Sie die folgenden Aktionen ausführen:

- Erstellen kontinuierlicher Sicherungen und regelmäßiger Sicherungen. Kontinuierliche Backups sind für die point-in-time Wiederherstellung nützlich, und regelmäßige Backups sind nützlich, um Ihre langfristigen Anforderungen an die Datenspeicherung zu erfüllen.
- Automatisieren der Planung und Aufbewahrung von Backups durch die zentrale Konfigurierung von Backuprichtlinien.
- Wiederherstellen von Backups von Amazon-S3-Daten zu einem bestimmten Zeitpunkt.

Darüber hinaus können Sie S3 Versioning und S3 Replication verwenden AWS Backup, um bei der Wiederherstellung nach versehentlichen Löschungen zu helfen und Ihre eigenen Self-Recovery-Operationen durchzuführen.

Voraussetzungen

Sie müssen S3 Versioning für Ihren Bucket aktivieren, bevor Sie ihn sichern AWS Backup können.

> **Note**
>
> Wir empfehlen, dass Sie eine andere Lebenszyklusablaufregel für versionsfähige Buckets festlegen, von denen ein Backup erstellt wird. Wenn Sie keinen Lebenszyklusablaufzeitraum festlegen, können Ihre Amazon-S3-Speicherkosten steigen, weil AWS Backup alle Versionen Ihrer Amazon-S3-Daten beibehält.

Erste Schritte

Informationen zu den ersten Schritten AWS Backup für Amazon S3 finden Sie unter Erstellen von Amazon S3 S3-Backups im AWS Backup Entwicklerhandbuch.

Beschränkungen und Einschränkungen

Weitere Informationen über die Einschränkungen finden Sie unter Erstellen von Amazon-S3-Backups im Entwicklerhandbuch für AWS Backup .

Kostenoptimierung

Amazon S3 bietet eine Reihe von Funktionen und Speicherklassen, mit denen Sie die Kosten während Ihres gesamten Datenlebenszyklus optimieren können. Speicherklassen bieten die Flexibilität, Ihre Kosten im Griff zu haben, indem sie unterschiedliche Datenzugriffsebenen zu entsprechenden Kosten anbieten, ohne dass Vorabgebühren anfallen oder Sie sich darauf festlegen müssen, wie viele Inhalte Sie speichern. Wie bei anderen AWS Diensten zahlen Sie nutzungsabhängig und zahlen nur für das, was Sie tatsächlich nutzen.

Amazon S3 S3-Speicherklassen wurden speziell entwickelt, um den kostengünstigsten Speicher für unterschiedliche Zugriffsmuster bereitzustellen. Dazu zählen:

- S3-Standard für die allgemeine Speicherung von Daten, auf die häufig zugegriffen wird.
- Amazon S3 Express One Zone für leistungsstarke Daten, auf die häufig zugegriffen wird, in einer Single-Availability Zone.
- S3 Intelligent-Tiering zur automatischen Optimierung der Kosten für Daten mit unbekannten oder sich ändernden Zugriffsmustern.
- S3 Standard-IA (S3 Standard-IA) und S3 One Zone-IA (S3 One Zone-IA) für langlebige Daten, auf die aber seltener zugegriffen wird.
- S3 Glacier Instant Retrieval für Archivdaten, auf die sofort zugegriffen werden muss.
- S3 Glacier für Archivdaten, die keinen sofortigen Zugriff erfordern, aber die Flexibilität benötigen, große Datenmengen kostenlos abzurufen.
- S3 Glacier Deep Archive für Langzeitarchivierung und digitale Aufbewahrung zu den niedrigsten Speicherkosten in der Cloud.

Sie können Objekte jederzeit in die kostengünstigste Speicherklasse verschieben. Darüber hinaus bietet Amazon S3 Funktionen zur Verwaltung Ihres Datenlebenszyklus. Sie können beispielsweise die S3 Lifecycle-Konfiguration verwenden, um die Umstellung von Objekten auf kostengünstigere Speicherklassen zu automatisieren oder um abgelaufene Objekte auf der Grundlage der von Ihnen definierten Regeln automatisch zu löschen.

Funktionen wie S3-Speicherklassenanalyse, Kennzeichnung der Kostenzuweisung sowie Abrechnungs- und Nutzungsberichte helfen Ihnen bei der Analyse Ihrer Kosten- und Nutzungsmuster.

Themen

- Abrechnungs- und Nutzungsberichte für Amazon S3
- Amazon S3 S3-Speicherklassen verstehen und verwalten
- Verwaltung des Lebenszyklus von Objekten

Abrechnungs- und Nutzungsberichte für Amazon S3

Wenn Sie Amazon S3 verwenden, müssen Sie keine Vorabgebühren zahlen oder sich darauf festlegen, wie viele Inhalte Sie speichern werden. Wie bei anderen AWS-Services Unternehmen zahlen Sie nutzungsabhängig und zahlen nur für das, was Sie tatsächlich nutzen.

AWS bietet die folgenden Berichte für Amazon S3:

- Abrechnungsberichte — Mehrere Berichte, die einen Überblick über alle Aktivitäten bieten AWS-Services , die Sie verwenden, einschließlich Amazon S3. AWS stellt dem Besitzer des S3-Buckets immer die Amazon S3 S3-Gebühren in Rechnung, es sei denn, der Bucket wurde als Bucket mit Requester Pays erstellt. Weitere Informationen über Zahlung durch den Anforderer finden Sie unter Nutzen von Buckets mit Zahlung durch den Anforderer für Speicherübertragungen und Nutzung. Weitere Informationen über Fakturierungsberichte finden Sie unter AWS Billing Berichte für Amazon S3.
- Nutzungsbericht – Eine Zusammenfassung von Aktivitäten für einen bestimmten Service, zusammengefasst nach Stunde, Tag und Monat. Sie können wählen, welche Nutzungsart und welche Operation aufgenommen werden soll. Sie können auch festlegen, wie die Daten aggregiert werden. Weitere Informationen finden Sie unter AWS Nutzungsberichte für Amazon S3.

Die folgenden Themen enthalten Informationen zur Fakturierung und zu Nutzungsberichten für Amazon S3.

Themen

- Verwenden von Kostenzuordnungs-Markierungen für S3-Buckets
- AWS Billing Berichte für Amazon S3
- AWS Nutzungsberichte für Amazon S3
- Grundlegendes zu Ihren AWS Abrechnungs- und Nutzungsberichten für Amazon S3
- Abrechnung für Amazon S3 S3-Fehlerantworten

Verwenden von Kostenzuordnungs-Markierungen für S3-Buckets

Um die Speicherkosten oder andere Kriterien für einzelne Projekte oder Gruppen von Projekten nachzuverfolgen, kennzeichnen Sie Ihre Amazon S3-Buckets unter Verwendung der Kostenzuordnungs-Markierungen. Ein Kostenzuordnungs-Tag ist ein Schlüssel-Wert-Paar, das Sie definieren und mit einem S3-Bucket verknüpfen. Nachdem Sie Kostenzuordnungs-Markierungen aktiviert haben, verwendet AWS die Markierungen, um Ihre Ressourcenkosten in Ihrem Kostenzuordnungsbericht zu gruppieren. Kostenzuordnungs-Markierungen können nur zur Kennzeichnung von Buckets verwendet werden. Weitere Informationen zu Markierungen, die zur Kennzeichnung von Objekten verwendet werden, finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen.

Der Kostenzuordnungsbericht listet die AWS -Nutzung für Ihr Konto nach Produktkategorie und verbundenem Kontobenutzer auf. Der Bericht enthält die gleichen Einzelposten wie der detaillierte Fakturierungsbericht (siehe Grundlegendes zu Ihren AWS Abrechnungs- und Nutzungsberichten für Amazon S3), und zusätzliche Spalten für Ihre Tag-Schlüssel.

AWS bietet zwei Arten von Kostenzuordnungs-Tags, einen AWS-generierten Tag und benutzerdefinierte Tags. AWS definiert, erstellt und wendet den AWS-generierten `createdBy`-Tag nach einem Amazon S3-CreateBucket-Ereignis für Sie an. Sie definieren, erstellen und verwenden benutzerdefinierte Markierungen für Ihren S3-Bucket.

Sie müssen beide Tag-Typen in der Fakturierungs- und Kostenverwaltungskonsole separat aktivieren, damit sie in Ihren monatlichen Berichten angezeigt werden können. Weitere Informationen über von AWS generierte Markierungen finden Sie unter Von AWS generierte Kostenzuordnungs-Markierungen.

- Informationen zum Erstellen von Markierungen in der Konsole finden Sie unter Anzeigen der Eigenschaften eines S3-Buckets.
- Informationen zum Erstellen von Markierungen mit der Amazon S3-API finden Sie unter PUT Bucket Tagging in der Amazon Simple Storage Service API-Referenz.
- Informationen zum Erstellen von Markierungen mit der AWS CLI finden Sie unter put-bucket-tagging in der AWS CLI-Befehlsreferenz.
- Weitere Informationen zur Aktivierung von Markierungen finden Sie unter Using cost allocation tags (Verwendung von Kostenzuordnungs-Markierungen) im AWS Billing-Benutzerhandbuch.

Benutzerdefinierte Kostenzuordnungs-Markierungen

Ein benutzerdefiniertes Kostenzuordnungs-Tag umfasst folgende Komponenten:

- Der Tag-Schlüssel. Der Tag-Schlüssel ist der Name des Markierungen. Im Tag project/Trinity beispielsweise ist project der Schlüssel. Die Tag-Schlüssel ist eine Zeichenfolge, in der zwischen Groß- und Kleinschreibung unterschieden wird und die 1 bis 128 Unicode-Zeichen enthalten kann.

- Der Tag-Wert. Der Tag-Wert ist eine erforderliche Zeichenfolge. Im Tag project/Trinity beispielsweise ist Trinity der Wert. Die Tag-Wert ist eine Zeichenfolge, in der zwischen Groß- und Kleinschreibung unterschieden wird und die 0 bis 256 Unicode-Zeichen enthalten kann.

Weitere Informationen zu den zulässigen Zeichen für benutzerdefinierte Markierungen und anderen Einschränkungen finden Sie unter User-Defined Tag Restrictions (Einschränkungen benutzerdefinierter Markierungen) im AWS Billing-Benutzerhandbuch. Weitere Informationen über benutzerdefinierte Markierungen finden Sie unter User-Defined Cost Allocation Markierungen (Benutzerdefinierte Kostenzuordnungs-Markierungen) im AWS Billing-Benutzerhandbuch.

S3-Bucket-Markierungen

Jeder S3-Bucket verfügt über einen Tag-Satz. Ein Tag-Satz enthält alle Markierungen, die diesem Bucket zugewiesen sind. Ein Tag-Satz kann bis zu 50 Markierungen enthalten oder leer sein. Schlüssel müssen eindeutig innerhalb eines Tag-Satzes sein, aber die Werte in einem Tag-Satz müssen nicht eindeutig sein. Sie können beispielsweise denselben Wert in Tag-Sätzen mit den Namen project/Trinity und cost-center/Trinity haben.

Wenn Sie in einem Bucket ein Tag mit demselben Schlüssel wie für ein vorhandenes Tag hinzufügen, wird der alte Wert mit dem neuen überschrieben.

AWS ordnet Ihren Markierungen keine semantische Bedeutung zu. Wir interpretieren Markierungen streng als Zeichenfolgen.

Um Markierungen hinzuzufügen, zu bearbeiten oder zu löschen, verwenden Sie die Amazon S3-Konsole, die AWS Command Line Interface (AWS CLI) oder die Amazon S3-API.

Weitere Infos

- Verwenden von Kostenzuordnungs-Markierungen im AWS Billing-Benutzerhandbuch.
- Grundlegendes zu Ihren AWS Abrechnungs- und Nutzungsberichten für Amazon S3
- AWS Billing Berichte für Amazon S3

AWS Billing Berichte für Amazon S3

Ihre monatliche Rechnung von AWS trennt Ihre Nutzungsinformationen und Kosten nach AWS-Service Funktion. Es sind mehrere AWS Billing Berichte verfügbar: der Monatsbericht, der Kostenzuordnungsbericht und detaillierte Abrechnungsberichte. Informationen zum Anzeigen Ihrer Fakturierungsberichte finden Sie unter Viewing Your Bill (Anzeigen Ihrer Rechnung) im AWS Billing - Benutzerhandbuch.

Um Ihre AWS Nutzung zu verfolgen und die geschätzten Gebühren für Ihr Konto anzugeben, können Sie Folgendes einrichten AWS Cost and Usage Reports. Weitere Informationen finden Sie unter Was sind AWSCost and Usage Reports? im Leitfaden AWS für Datenexporte.

Sie können auch einen Nutzungsbericht herunterladen, der Ihnen weitere Informationen über Ihre Amazon-S3-Nutzung mitteilt, als die Fakturierungs-Berichte. Weitere Informationen finden Sie unter AWS Nutzungsberichte für Amazon S3.

Die folgende Tabelle listet die Gebühren für die Amazon-S3-Nutzung auf.

Gebühr	Kommentare
Speicher	Sie zahlen für das Speichern von Objekten in S3-Buckets. Der Tarif, der Ihnen in Rechnung gestellt wird, hängt von der Größe Ihrer Objekte, der Aufbewahrungsdauer der Objekte im Laufe des Monats und der Speicherklasse ab. Amazon S3 bietet die folgenden Speicherk lassen: S3 Standard, S3 Express One Zone, S3 Intelligent-Tiering, S3 Standard-IA (IA für seltenen Zugriff), S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval , S3 Glacier Deep Archive oder Reduced Redundancy Storage (). RRS Weitere Informati onen über Speicherklassen finden Sie unter Amazon S3 S3-Speicherklassen verstehen und verwalten. Beachten Sie, dass Ihnen bei aktivierter S3-Versionierung jede Version eines Objekts, die

Gebühr	Kommentare
	aufbewahrt wird, in Rechnung gestellt wird. Weitere Informationen über das Versioning finden Sie unter Wie funktioniert S3-Versioning.
Buckets für allgemeine Zwecke	Die ersten 2000 Buckets für allgemeine Zwecke, die Sie in Ihrem Konto erstellen , werden Ihnen nicht in Rechnung gestellt. Für jeden Bucket, den Sie nach den ersten 2000 erstellen, fällt jedoch eine Gebühr pro Bucket an. Dieser Tarif wird pro Bucket/Monat abgerechnet. Informationen zu den Preisen für Allzweck-Buckets finden Sie unter Amazon S3 S3-Preise.
Überwachen und Automatisieren	Für jedes in der Speicherklasse S3 Intelligent-Tiering gespeicherte Objekt zahlen Sie eine monatliche Überwachungs- und Automatis ierungsgebühr zur Überwachung der Zugriffsm uster und Verschiebung der Objekte zwischen den Zugriffsstufen des S3 Intelligent-Tiering.
Anforderungen	Sie zahlen für Anfragen, z. B. `GET` Anfragen, die für Ihre S3-Buckets und -Objekte gestellt werden. Dazu zählen auch Lebenszyklus-Anfra gen. Die Gebühren für Anfragen hängen davon ab, welche Art von Anfrage Sie stellen. Weitere Informationen zu den Anfragepreisen erhalten Sie unter Amazon-S3-Preise.
Abrufe	Sie zahlen für das Abrufen von Objekten, die sich im Speicher S3 Standard-IA, S3 One Zone-IA-, S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive befinden.

Gebühr	Kommentare
Vorzeitige Löschungen	Wenn Sie ein Objekt im Speicher S3 Standard-IA, S3 One Zone-IA, S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive löschen, bevor die minimale Speicherzusage abgelaufen ist, zahlen Sie eine Gebühr für das vorzeitige Löschen für dieses Objekt.
Speicherverwaltung	Sie zahlen für die Speicherverwaltungsfunktionen (Amazon S3 S3-Inventar, Analytik und Objekt-Tagging), die in den Buckets Ihres Kontos aktiviert sind.
Bandbreite	Sie zahlen für die gesamte Bandbreite in und aus Amazon S3 mit Ausnahme des Folgenden: • Datenübertragung aus dem Internet • Daten, die an eine Amazon Elastic Compute Cloud (AmazonEC2) -Instance übertragen werden, wenn sich die Instance im selben AWS-Region wie der S3-Bucket befindet • An Amazon übertragene Daten CloudFront (CloudFront) Sie zahlen auch eine Gebühr für alle Daten, die mit Amazon S3 Transfer Acceleration übertragen werden.

Detaillierte Informationen zu den Amazon S3-Nutzungsgebühren für Speicher, Datenübertragung und Services finden Sie unter Amazon S3 S3-Preise und Amazon S3 FAQs.

Informationen zum Verständnis der in den Abrechnungs- und Nutzungsberichten für Amazon S3 verwendeten Codes und Abkürzungen finden Sie unterGrundlegendes zu Ihren AWS Abrechnungs- und Nutzungsberichten für Amazon S3.

Weitere Informationen

- AWS Nutzungsberichte für Amazon S3
- Verwenden von Kostenzuordnungs-Markierungen für S3-Buckets
- AWS Billing und Kostenmanagement
- Amazon S3 – Preise

AWS Nutzungsberichte für Amazon S3

Wenn Sie einen Nutzungsbericht herunterladen, können Sie die Nutzungsdaten nach Stunde, Tag und Monat zusammenfassen. Der Amazon S3 S3-Nutzungsbericht listet die Vorgänge nach Nutzungstyp und auf AWS-Region. Für detailliertere Nutzungsberichte über Ihre Amazon-S3-Speichernutzung laden Sie dynamisch generierte AWS -Nutzungsberichte herunter. Sie können wählen, welche Nutzungsart, welche Operation und welcher Zeitraum aufgenommen werden soll. Sie können auch festlegen, wie die Daten aggregiert werden. Weitere Informationen zu Nutzungsberichten finden Sie unter AWS Nutzungsbericht im AWS Data Exports User Guide.

Der Amazon-S3-Nutzungsbericht enthält die folgenden Informationen:

- Service – Amazon S3
- Operation – Die Operation, die für Ihren Bucket oder Ihr Objekt durchgeführt wird. Eine detaillierte Erklärung der Amazon-S3-Vorgänge finden Sie unter Nachverfolgen von Vorgängen in Ihren Nutzungsberichten.
- UsageType— Einer der folgenden Werte:
  - Ein Code zur Identifizierung des Speichertyps
  - Ein Code zur Identifizierung der Anfrage
  - Ein Code zur Identifizierung des Abruftyps
  - Ein Code zur Identifizierung des Typs der Datenübertragung
  - Ein Code, der vorzeitiges Löschen aus dem S3 Intelligent-Tiering, S3 Standard-IA-, S3-One-Zone-Infrequent-Access (S3 One Zone-IA)-, S3 Glacier Flexible Retrieval- oder S3 Glacier Deep Archive Speicher identifiziert

- `StorageObjectCount` – Die Anzahl der Objekte in einem bestimmten Bucket

  Eine detaillierte Erklärung der Amazon-S3-Nutzungstypen finden Sie unter Grundlegendes zu Ihren AWS Abrechnungs- und Nutzungsberichten für Amazon S3.

- Resource (Ressource) – Der Name des Buckets, der der aufgelisteten Nutzung zugeordnet ist.
- StartTime— Startzeit des Tages, für den die Verwendung gilt, in koordinierter Weltzeit (UTC).
- EndTime— Endzeit des Tages, für den die Nutzung gilt, in koordinierter Weltzeit (UTC).
- UsageValue— Einer der folgenden Volumenwerte. Die typische Maßeinheit für Daten ist Gigabyte (GB). Abhängig vom Dienst und dem Bericht können stattdessen Terabytes (TB) angezeigt werden.
  - Die Anzahl der Anforderungen in dem angegebenen Zeitraum
  - Die übertragene Datenmenge
  - Die Menge der in einer bestimmten Stunde gespeicherten Daten
  - Die Menge der Daten im Zusammenhang mit Wiederherstellungen aus dem Speicher S3 Standard-IA, S3 One Zone-IA, S3 Glacier Flexible Retrieval-, oder S3 Glacier Deep Archive

> **Tip**
>
> Detaillierte Informationen über alle Anfragen, die Amazon S3 für Ihre Objekte empfängt, aktivieren Sie die Server-Zugriffsprotokollierung für Ihre Buckets Weitere Informationen finden Sie unter Protokollieren von Anfragen mit Server-Zugriffsprotokollierung.

Sie können einen Nutzungsbericht als Datei XML oder als Datei mit kommagetrennten Werten (CSV) herunterladen. Im Folgenden finden Sie ein Beispiel für einen CSV Nutzungsbericht, der in einer Tabellenkalkulationsanwendung geöffnet wurde.

Service	Operation	UsageType	Resource	StartTime	EndTime	UsageValue
AmazonS3	HeadBucket	USW2-C3DataTransfer-Out-Bytes	admin-created3	6/1/2017 0:00	7/1/2017 0:00	15309
AmazonS3	PutObject	USW2-C3DataTransfer-In-Bytes	admin-created3	6/1/2017 0:00	7/1/2017 0:00	19062
AmazonS3	HeadBucket	USW2-Requests-Tier2	admin-created3	6/1/2017 0:00	7/1/2017 0:00	68
AmazonS3	PutObjectForRepl	USW1-Requests-SIA-Tier1	ca-example-bucket	6/1/2017 0:00	7/1/2017 0:00	178294
AmazonS3	PutObjectForRepl	USW1-USW2-AWS-In-Bytes	ca-example-bucket	6/1/2017 0:00	7/1/2017 0:00	387929083
AmazonS3	GetObjectForRepl	USW2-Requests-NoCharge	admin-created3	6/1/2017 0:00	7/1/2017 0:00	108
AmazonS3	GetObjectForRepl	USW2-USW1-AWS-Out-Bytes	my-test-bucket-bash	6/1/2017 0:00	7/1/2017 0:00	387910021

Weitere Informationen finden Sie unter Grundlegendes zu Ihren AWS Abrechnungs- und Nutzungsberichten für Amazon S3.

Der AWS Nutzungsbericht wird heruntergeladen

Sie können einen Nutzungsbericht als CSV Datei XML oder herunterladen.

Den Nutzungsbericht herunterladen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Titelleiste Ihren Benutzernamen oder Ihre Konto-ID und dann Billing and Cost Management aus.
3. Wählen Sie im Navigationsbereich Legacy Pages und dann Kosten- und Nutzungsberichte aus.
4. Wählen Sie unter AWS Nutzungsbericht die Option Nutzungsbericht erstellen aus.
5. Wählen Sie auf der Seite Nutzungsbericht herunterladen die folgenden Einstellungen aus:
   - Services — Wählen Sie Amazon Simple Storage Service.
   - Usage Types (Nutzungstypen) – Eine detaillierte Erläuterung der Amazon-S3-Nutzungstypen finden Sie unter Grundlegendes zu Ihren AWS Abrechnungs- und Nutzungsberichten für Amazon S3.
   - Operation – Eine detaillierte Erläuterung der Amazon-S3-Vorgänge finden Sie unter Nachverfolgen von Vorgängen in Ihren Nutzungsberichten.
   - Time Period (Zeitraum) – Der Zeitraum, der in dem Bericht erfasst werden soll.
   - Report Granularity (Berichtsgranularität) – Gibt an, ob in dem Bericht Zwischensummen nach Stunde, Tag oder Monat angezeigt werden sollen.
6. Wählen Sie Herunterladen, wählen Sie das Download-Format (XMLBericht oder CSVBericht) und folgen Sie dann den Anweisungen, um den Bericht zu öffnen oder zu speichern.

Weitere Informationen

- Grundlegendes zu Ihren AWS Abrechnungs- und Nutzungsberichten für Amazon S3
- AWS Billing Berichte für Amazon S3

Grundlegendes zu Ihren AWS Abrechnungs- und Nutzungsberichten für Amazon S3

Amazon-S3-Fakturierungs- und Nutzungsberichte verwenden Codes und Abkürzungen. Für Nutzungstypen in der folgenden Tabelle ersetzen Sie *regionregion1*, und *region2* durch Abkürzungen aus dieser Liste:

- APE1: Asien-Pazifik (Hongkong)
- APN1: Asien-Pazifik (Tokio)
- APN2: Asien-Pazifik (Seoul)
- APN3: Asien-Pazifik (Osaka)
- APS1: Asien-Pazifik (Singapur)
- APS2: Asien-Pazifik (Sydney)
- APS3: Asien-Pazifik (Mumbai)
- APS4: Asien-Pazifik (Jakarta)
- APS5: Asien-Pazifik (Hyderabad)
- APS6: Asien-Pazifik (Melbourne)
- CAN1: Kanada (Zentral)
- CAN2: Kanada West (Calgary)
- CNN1: China (Peking)
- CNW1: China (Ningxia)
- AFS1: Afrika (Kapstadt)
- EUC2: Europa (Zürich)
- EUN1: Europa (Stockholm)
- EUS2: Europa (Spanien)
- EUC1: Europa (Frankfurt)
- EU: Europa (Irland)
- EUS1: Europa (Mailand)
- EUW2: Europa (London)
- EUW3: Europa (Paris)
- ILC1: Israel (Tel Aviv)

- MEC1: Naher Osten (UAE)
- MES1: Naher Osten (Bahrain)
- SAE1: Südamerika (São Paulo)
- UGW1: AWS GovCloud (US-West)
- UGE1: AWS GovCloud (US-Ost)
- USE1(oder kein Präfix): US East (Nord-Virginia)
- USE2: USA Ost (Ohio)
- USW1: USA West (Nordkalifornien)
- USW2: USA West (Oregon)

Für die Verwendungsarten von S3 Multiregion Access Points in der folgenden Tabelle ersetzen Sie *regiongroup1* und *regiongroup2* durch Abkürzungen aus dieser Liste:

- AP: Asien-Pazifik
- AU: Australien
- EU: Europa
- IN: Indien
- NA: Nordamerika
- SA: Südamerika

Regionalgruppen sind geografische Gruppierungen von mehreren. AWS-Regionen Weitere Informationen finden Sie unter Regionen und Availability Zones. Weitere Informationen zu den Preisen nach AWS-Region finden Sie unter Amazon S3 – Preise.

Die erste Spalte in der folgenden Tabelle listet die Nutzungstypen auf, die in Ihren Fakturierungs- und Nutzungsberichten vorkommen. Die typische Maßeinheit für Daten ist Gigabyte (GB). Abhängig vom Dienst und dem Bericht können stattdessen Terabytes (TB) angezeigt werden.

Verwendungstyp	Einheiten	Granularity	Beschreibung
region1-*region2*-AWS-In-A Bytes	GB	Stundensatz	Die Menge der beschleunigten Daten, die von übertragen *region1* werden *region2*

Verwendungstyp	Einheiten	Granularity	Beschreibung
region1-*region2*-AWS-In-ABytes-T1	GB	Stundensatz	Die Menge der beschleunigten T1-Daten, an die *region1* von übertragen wurde*region2*, wobei T1 sich auf CloudFront Anfragen an Präsenzstellen bezieht (POPs) in den Vereinigten Staaten, Europa und Japan
region1-*region2*-AWS-In-ABytes-T2	GB	Stundensatz	Die Menge der beschleunigten T2-Daten, von der *region1* aus übertragen wurde*region2*, wobei sich T2 auf CloudFront Anfragen bezieht POPs an allen anderen AWS Edge-Standorten
region1-*region2*-AWS-In-Bytes	GB	Stundensatz	Die Menge der übertragenen Daten *region1* von *region2*
region1-*region2*-AWS-Out-ABytes	GB	Stundensatz	Die Menge der beschleunigten Daten, die von *region1* bis übertragen wurden *region2*

Verwendungstyp	Einheiten	Granularity	Beschreibung
region1-*region2*-AWS-Out-ABytes-T1	GB	Stundensatz	Die Menge der beschleunigten T1-Daten*region2*, *region1* die von bis übertragen wurden, wobei sich T1 auf CloudFront Anfragen POPs in den Vereinigten Staaten, Europa und Japan bezieht
region1-*region2*-AWS-Out-ABytes-T2	GB	Stundensatz	Die Menge der beschleunigten T2-Daten, die von *region1* bis übertragen wurden*region2*, wobei T2 sich auf CloudFront Anfragen POPs an alle anderen AWS Edge-Standorte bezieht
region1-*region2*-AWS-Out-Bytes	GB	Stundensatz	Die Datenmenge, die von *region1* bis übertragen wurde *region2*
region-BatchOperations-Jobs	Anzahl	Stundensatz	Die Anzahl der ausgeführten S3-Batchvorgangsaufträge
region-BatchOperations-Objects	Anzahl	Stundensatz	Die Anzahl der von S3-Batchvorgänge ausgeführten Objektvorgänge

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-Bulk-Retrieval-Bytes	GB	Stundensatz	Die Menge der Daten, die mit Massenanforderungen von S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive abgerufen wurden
region-BytesDeleted-GDA	GB	Monatlich	Die Datenmenge, die durch einen DeleteObject Vorgang aus dem S3 Glacier Deep Archive Archive-Speicher gelöscht wurde
region-BytesDeleted-GIR	GB	Monatlich	Die Datenmenge, die durch einen DeleteObject Vorgang aus dem S3 Glacier Instant Retrieval-Speicher gelöscht wurde.
region-BytesDeleted-GLACIER	GB	Monatlich	Die Datenmenge, die durch einen DeleteObject Vorgang aus dem S3 Glacier Flexible Retrieval-Speicher gelöscht wurde
region-BytesDeleted-INT	GB	Monatlich	Die Datenmenge, die durch einen DeleteObject Vorgang aus dem S3 Intelligent-Tiering-Speicher gelöscht wurde

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-BytesDeleted-RRS	GB	Monatlich	Die Datenmenge, die durch einen DeleteObject Vorgang aus dem Speicher mit reduziertem Redundanzspeicher () gelöscht wurde RRS
region-BytesDeleted-SIA	GB	Monatlich	Die Datenmenge, die durch einen DeleteObject Vorgang aus dem S3-Standard-IA-Speicher gelöscht wurde
region-BytesDeleted-STANDARD	GB	Monatlich	Die Datenmenge, die durch einen DeleteObject Vorgang aus dem S3-Standardspeicher gelöscht wurde
region-BytesDeleted-ZIA	GB	Monatlich	Die Datenmenge, die durch einen DeleteObject Vorgang aus dem S3 One Zone-IA-Speicher gelöscht wurde
region-C3DataTransfer-In-Bytes	GB	Stundensatz	Die Datenmenge, die EC2 innerhalb desselben Zeitraums von Amazon nach Amazon S3 übertragen wurde AWS-Region

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-C3DataTransfer-Out-Bytes	GB	Stundensatz	Die Datenmenge, die EC2 innerhalb desselben Zeitraums von Amazon S3 zu Amazon übertragen wurde AWS-Region
region-CloudFront-In-Bytes	GB	Stundensatz	Die Datenmenge, die in eine oder AWS-Region aus einer CloudFront Distribution übertragen wurde
region-CloudFront-Out-Bytes	GB	Stundensatz	Die Datenmenge, die von einer AWS-Region zu einer CloudFront Distribution übertragen wurde
region-DataTransfer-In-Bytes	GB	Stundensatz	Die Menge der aus dem Internet zu Amazon S3 übertragenen Daten
region-DataTransfer-Out-Bytes	GB	Stundensatz	Die Menge der von Amazon S3 ins Internet übertragenen Daten[1]
region-DataTransfer-Regional-Bytes	GB	Stundensatz	Die Datenmenge, die von Amazon S3 auf AWS Ressourcen innerhalb desselben übertragen wurde AWS-Region

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-EarlyDelete-ByteHrs	GB-Stunden	Stündlich	Anteilige Speichernutzung für Objekte, die aus dem Speicher S3 Glacier Flexible Retrieval gelöscht wurden, bevor die 90-tägige Mindestnutzung endete [2]
region-EarlyDelete-GDA	GB-Stunden	Stundensatz	Anteilige Speichernutzung für Objekte, die aus dem S3-Glacier-Deep-Archive-Speicher gelöscht wurden, bevor die 180-tägige Mindestnutzung endete [2]
region-EarlyDelete-GIR	GB-Stunden	Stündlich	Anteilige Speichernutzung für Objekte, die aus dem Speicher S3 Glacier Instant Retrieval gelöscht wurden, bevor die 90-tägige Mindestnutzung endete.
region-EarlyDelete-GIR-SmObjects	GB-Stunden	Stündlich	Anteilige Speichernutzung für kleine Objekte (kleiner als 128 KB), die aus S3 Glacier Instant Retrieval gelöscht wurden, bevor die 90-tägige Mindestnutzung endete.

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-EarlyDelete-SIA	GB-Stunden	Stundensatz	Anteilige Speichernutzung für Objekte, die aus dem S3-Standard-IA-Speicher gelöscht wurden, bevor die 30-tägige Mindestnutzung endet[3]
region-EarlyDelete-SIA-SmObjects	GB-Stunden	Stundensatz	Anteilige Speichernutzung für kleine Objekte (kleiner als 128 KB), die aus dem S3-Standard-IA-Speicher gelöscht wurden, bevor die 30-tägige Mindestnutzung endete[3]
region-EarlyDelete-ZIA	GB-Stunden	Stundensatz	Anteilige Speichernutzung für Objekte, die aus dem S3-One-Zone-IA-Speicher gelöscht wurden, bevor die 30-tägige Mindestnutzung endet[3]
region-EarlyDelete-ZIA-SmObjects	GB-Stunden	Stundensatz	Anteilige Speichernutzung für kleine Objekte (kleiner als 128 KB), die aus dem S3-One-Zone-IA-Speicher gelöscht wurden, bevor die 30-tägige Mindestnutzung endete[3]

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-Expedited-Retrieval-Bytes	GB	Stundensatz	Die Menge der Daten, die mit beschleunigten Anforderungen von S3 Glacier Flexible Retrieval abgerufen wurden
Global-Bucket-Hrs-FreeTier	Bucket	Monatlich	Die Anzahl der Allzweck-Buckets in Ihrem Konto innerhalb des kostenlosen Kontingents für 2000 Buckets auf Kontoebene
Global-Bucket-Hrs	Bucket	Monatlich	Die Anzahl der Allzweck-Buckets in Ihrem Konto, die über das kostenlose Kontingent für 2000 Buckets auf Kontoebene hinausgehen
region-Inventory-Objects Listed	Objekte	Stundensatz	Die Anzahl der Objekte, die für eine Objektgruppe (Objekte werden nach Bucket oder Präfix gruppiert) in einer Lagerbestandsliste aufgelistet werden
region-Monitoring-Automation-INT	Objekte	Stundensatz	Die Anzahl an eindeutigen Objekten, die in der S3-Intelligent-Tiering-Speicherklasse überwacht und automatisch abgestuft werden

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-MRAP-Out-Bytes	GB	Stundensatz	Die Datenmenge, die über einen S3-Endpunkt mit multiregionalen Access Points aus Buckets in einer Region übertragen wird (MRAPPreise für Datenrouting).
region-MRAP-In-Bytes	GB	Stundensatz	Die Datenmenge, die über einen S3-Endpunkt mit multiregionalen Access Points aus Buckets in einer Region übertragen wird (Preise für MRAP Datenrouting).
regiongroup1-regiongroup2- -MRAP-Out-Bytes	GB	Stundensatz	Die Datenmenge, die über einen S3-Endpunkt mit multiregionalen Access Points von einem Bucket *regiongroup1* zu einem Client *regiongroup2* außerhalb des Netzwerks übertragen wird. AWS

Verwendungstyp	Einheiten	Granularity	Beschreibung
regiongroup1-regiongroup2--MRAP-In-Bytes	GB	Stundensatz	Die Datenmenge, die über einen S3-Endpunkt mit multiregionalen Access Points *regiongroup1* von einem Client *regiongroup2* außerhalb des Netzwerks zu einem Bucket übertragen wird AWS .
region-OverwriteBytes-Copy-GDA	GB	Monatlich	Die Datenmenge, die durch einen CopyObject Vorgang aus dem S3 Glacier Deep Archive Archive-Speicher überschrieben wurde
region-OverwriteBytes-Copy-GIR	GB	Monatlich	Die Datenmenge, die durch einen CopyObject Vorgang aus dem S3 Glacier Instant Retrieval -Speicher überschrieben wurde.
region-OverwriteBytes-Copy-GLACIER	GB	Monatlich	Die Datenmenge, die durch einen CopyObject Vorgang aus dem S3 Glacier Flexible Retrieval -Speicher überschrieben wurde

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-OverwriteBytes-Copy-INT	GB	Monatlich	Die Datenmenge, die durch einen CopyObject Vorgang aus dem S3 Intelligent-Tiering-Speicher überschrieben wurde
region-OverwriteBytes-Copy-RRS	GB	Monatlich	Die Datenmenge, die durch einen CopyObject Vorgang aus dem Speicher mit reduziertem Redundanzspeicher () überschrieben wurde RRS
region-OverwriteBytes-Copy-SIA	GB	Monatlich	Die Datenmenge, die durch einen CopyObject Vorgang aus dem S3-Standard-IA-Speicher überschrieben wurde
region-OverwriteBytes-Copy-STANDARD	GB	Monatlich	Die Datenmenge, die durch einen CopyObject Vorgang aus dem S3 Standard-Speicher überschrieben wurde
region-OverwriteBytes-Copy-ZIA	GB	Monatlich	Die Datenmenge, die durch einen CopyObject Vorgang aus dem S3 One Zone-IA-Speicher überschrieben wurde

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-OverwriteBytes-Put-GDA	GB	Monatlich	Die Datenmenge, die durch einen PutObject Vorgang aus dem S3 Glacier Deep Archive Archive-Speicher überschrieben wurde
region-OverwriteBytes-Put-GIR	GB	Monatlich	Die Datenmenge, die durch einen PutObject Vorgang aus dem S3 Glacier Instant Retrieval -Speicher überschrieben wurde.
region-OverwriteBytes-Put-GLACIER	GB	Monatlich	Die Datenmenge, die durch einen PutObject Vorgang aus dem S3 Glacier Flexible Retrieval -Speicher überschrieben wurde
region-OverwriteBytes-Put-INT	GB	Monatlich	Die Datenmenge, die durch einen PutObject Vorgang aus dem S3 Intelligent-Tiering-Speicher überschrieben wurde
region-OverwriteBytes-Put-RRS	GB	Monatlich	Die Datenmenge, die durch einen PutObject Vorgang aus dem Speicher mit reduziert em Redundanzspeicher () überschrieben wurde RRS

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-OverwriteBytes-Put-SIA	GB	Monatlich	Die Datenmenge, die durch einen PutObject Vorgang aus dem S3-Standard-IA-Speicher überschrieben wurde
region-OverwriteBytes-Put-STANDARD	GB	Monatlich	Die Datenmenge, die durch einen PutObject Vorgang aus dem S3 Standard-Speicher überschrieben wurde
region-OverwriteBytes-Put-ZIA	GB	Monatlich	Die Datenmenge, die durch einen PutObject Vorgang aus dem S3 One Zone-IA-Speicher überschrieben wurde
region1-*region2*-S3RTC-In-Bytes	GB	Monatlich	Die Datenmenge, die für S3 Replication Time Control (S3RTC) von bis *region2* nach den *region1* VorgängenPutObject ReplTime ,,, GetObjectReplTime InitiateMultipartUploadRepl Time UploadPartReplTime , CompleteMultipartUploadReplTime und übertragen wurde WriteACLReplTime

Verwendungstyp	Einheiten	Granularity	Beschreibung
region1-*region2*-S3RTC-Out-Bytes	GB	Monatlich	Die für S3 Replication Time Control (S3RTC) von bis *region1* nach *region2* den WriteACLReplTime VorgängenPutObject ReplTime ,,, GetObjectReplTime InitiateMultipartU ploadReplTime UploadPartReplTime CompleteM ultipartU ploadReplTime , und übertragene Datenmenge
region-Requests-GDA-Tier1	Anzahl	Stündlich	Die Anzahl vonPUT,,COPY, POST CreateMultipartUpl oad UploadPart , oder CompleteMultipartU pload Anfragen für S3 Glacier Deep Archive Archive-Objekte [6]
region-Requests-GDA-Tier2	Anzahl	Stündlich	Die Anzahl von GET und HEAD Anfragen für S3 Glacier Deep Archive Archive-Objekte
region-Requests-GDA-Tier3	Anzahl	Stundensa tz	Die Anzahl der S3 Glacier Deep Archive-Standard-Wiederherstellungsa nforderungen

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-Requests-GDA-Tier5	Anzahl	Stundensatz	Die Anzahl der S3 Glacier Deep Archive-Wiederherstellungsmassenanforderungen
region-Requests-GIR-Tier1	Anzahl	Stündlich	Die Anzahl von PUTCOPY, oder POST -Anfragen für S3 Glacier Instant Retrieval-Objekte.
region-Requests-GIR-Tier2	Anzahl	Stündlich	Die Anzahl GET und alle anderen Anfragen, die nicht zu S3 Glacier Instant Retrieval-Tier1 gehören, für S3 Glacier Instant Retrieval-Objekte.
region-Requests-GLACIER-Tier1	Anzahl	Stündlich	Die Anzahl vonPUT,, COPYPOST, oder CompleteMultipartUpload Anfragen für S3 CreateMultipartUpload Glacier UploadPart Flexible Retrieval-Objekte 6
region-Requests-GLACIER-Tier2	Anzahl	Stündlich	Die Anzahl GET und alle anderen Anfragen, die nicht in S3 Glacier Flexible Retrieval-Objekten aufgeführt sind
region-Requests-INT-Tier1	Anzahl	Stündlich	Die Anzahl von PUTCOPY, oder POST Anfragen für S3 Intelligent-Tiering-Objekte

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-Requests-INT-Tier2	Anzahl	Stündlich	Die Anzahl GET und alle anderen Anfragen, die nicht zu Stufe 1 gehören, für S3 Intelligent-Tiering-Objekte
region-Requests-SIA-Tier1	Anzahl	Stündlich	Die Anzahl vonPUT, oder POST -Anfragen für S3 COPY Standard-IA-Objekte
region-Requests-SIA-Tier2	Anzahl	Stündlich	Die Anzahl GET und alle anderen Nicht-S3 Glacier Instant Retrieval-Tier1-Anfragen für S3 Standard-IA-Objekte
region-Requests-Tier1	Anzahl	Stündlich	Die Anzahl von PUTCOPY, oder POST Anfragen für S3 Standard und Tags sowie Anfragen für RRS alle Buckets und Objekte LIST
region-Requests-Tier2	Anzahl	Stündlich	Die Anzahl GET und alle anderen Anfragen, die nicht zu Stufe 1 gehören

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-Requests-Tier3	Anzahl	Stündlich	Die Anzahl der Lebenszyklusanforderungen an Wiederherstellungsanforderungen von S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive und Standard S3 Glacier Flexieble Retrieval
region-Requests-Tier4	Anzahl	Stündlich	Die Anzahl der Lebenszyklus-Übergänge in Speicher von S3 Glacier Instant Flexible Retrieval, S3 Intelligent-Tiering, S3 Standard-IA, oder S3 One Zone-IA
region-Requests-Tier5	Anzahl	Stündlich	Die Anzahl der Wiederherstellungsmassenanforderungen von S3 Glacier Flexible Retrieval
region-Requests-Tier6	Anzahl	Stündlich	Die Anzahl der beschleunigten Wiederherstellungsanforderungen von S3 Glacier Flexible Retrieval
region-Requests-Tier8	Anzahl	Stündlich	Die Anzahl der S3 Access Grants-Anfragen
region-Requests-XZ-Tier1	Anzahl	Stündlich	Die Anzahl der PUT COPY Or-Anfragen für S3 Express One Zone-Objekte

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-Requests-XZ-Tier2	Anzahl	Stündlich	Die Anzahl GET und alle anderen Anfragen, die nicht zu S3 Express One Zone-Tier1 gehören, für S3 Express One Zone-Objekte
region-Requests-ZIA-Tier1	Anzahl	Stündlich	Die Anzahl von PUTCOPY, oder POST Anfragen für S3 One Zone-IA-Objekte
region-Requests-ZIA-Tier2	Anzahl	Stündlich	Die Anzahl der GET und aller anderen Nicht-S3 One Zone-IA-Tier 1-Anfragen für S3 One Zone-IA-Objekte
region-Retrieval-GIR	GB	Stundensatz	Die Menge der Daten, die aus dem Speicher S3 Glacier Instant Retrieval abgerufen wurden.
region-Retrieval-SIA	GB	Stundensatz	Die Menge der Daten, die aus dem S3-Standard-IA-Speicher abgerufen wurden
region-Retrieval-XZ	GB	Stundensatz	Der Teil der Daten, der in einer bestimmten Abrufanforderung (PUToderCOPY) mit S3 Express One Zone-Speicher 512 KB überschreitet

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-Retrieval-ZIA	GB	Stundensatz	Die Menge der Daten, die aus dem S3-One-Zone-IA Speicher abgerufen wurde
region-S3DSSE-In-Bytes	GB	Monatlich	Die Menge der von Amazon S3 doppelt verschlüsselten Daten
region-S3DSSE-Out-Bytes	GB	Monatlich	Die Menge der doppelt verschlüsselten Daten, die von Amazon S3 entschlüsselt wurden
region-S3G-DataTransfer-In-Bytes	GB	Stundensatz	Die Datenmenge, die in Amazon S3 übertragen wird, um Objekte aus dem S3 Glacier Flexible Retrieval- oder S3 Glacier Deep Archive-Speicher wiederherzustellen
region-S3G-DataTransfer-Out-Bytes	GB	Stundensatz	Die Menge der Daten, die von Amazon S3 zum Übergang von Objekten in den Speicher S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive übertragen werden
region-Select-Returned-Bytes	GB	Stundensatz	Die Menge der Daten, die mit Select-Anforderungen aus dem S3-Standard-Speicher zurückgegeben wurden

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-Select-Returned-GIR-Bytes	GB	Stundensatz	Die Datenmenge, die mit Select-Anfragen vom Speicher S3 Glacier Instant Retrieval zurückgegeben wird.
region-Select-Returned-INT-Bytes	GB	Stundensatz	Die Menge der Daten, die mit Select-Anforderungen aus dem S3-Intelligent-Tiering-Speicher zurückgegeben wurden
region-Select-Returned-SIA-Bytes	GB	Stundensatz	Die Menge der Daten, die mit Select-Anforderungen aus dem S3-Standard-IA-Speicher zurückgegeben wurden
region-Select-Returned-ZIA-Bytes	GB	Stundensatz	Die Menge der Daten, die mit Select-Anforderungen aus dem S3-One-Zone-IA-Speicher zurückgegeben wurden
region-Select-Scanned-Bytes	GB	Stundensatz	Die Menge der Daten, die mit Select-Anforderungen aus dem S3-Standard-Speicher gescannt wurden
region-Select-Scanned-GIR-Bytes	GB	Stundensatz	Die Datenmenge, die mit Select-Anfragen aus dem Speicher S3 Glacier Instant Retrieval gescannt wurde.

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-Select-Scanned-INT-Bytes	GB	Stundensatz	Die Menge der Daten, die mit Select-Anforderungen aus dem S3-Intelligent-Tiering-Speicher gescannt wurden
region-Select-Scanned-SIA-Bytes	GB	Stundensatz	Die Menge der Daten, die mit Select-Anforderungen aus dem S3-Standard-IA-Speicher gescannt wurden
region-Select-Scanned-ZIA-Bytes	GB	Stundensatz	Die Menge der Daten, die mit Select-Anforderungen aus dem S3-One-Zone-IA-Speicher gescannt wurden
region-Standard-Retrieval-Bytes	GB	Stundensatz	Die Anzahl der Bytes der Daten, die mit Standardanforderungen von S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive abgerufen wurden
region-StorageAnalytics-ObjCount	Objekte	Stündlich	Die Anzahl der eindeutigen Objekte, die in jeder Storage-Class-Analysis-Konfiguration überwacht werden.

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-StorageLens-ObjCount	Objekte	Täglich	Die Anzahl der eindeutigen Objekte in jedem S3 Storage Lens Dashboard, die durch erweiterte Metriken und Empfehlungen von S3 Storage Lens verfolgt werden.
region-StorageLensFreeTier-ObjCount	Objekte	Täglich	Die Anzahl der eindeutigen Objekte in jedem S3 Storage Lens Dashboard, die durch Nutzungsmetriken von S3 Storage Lens verfolgt werden.
StorageObjectCount	Anzahl	Täglich	Die Anzahl der Objekte in einem bestimmten Bucket
region-TagStorage-TagHrs	Tag-Hours	Täglich	Die Gesamtzahl der Markierungen für alle Objekte im Bucket, gemeldet nach Stunde
region-TimedStorage-ByteHrs	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten im S3-Standardspeicher gespeichert wurden
region-TimedStorage-GDA-ByteHrs	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten im S3 Glacier Deep Archive Archive-Speicher gespeichert wurden

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-TimedStorage-GDA-Staging	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten im S3 Glacier Deep Archive Archive-Staging-Speicher gespeichert wurden
region-TimedStorage-GIR-ByteHrs	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten im S3 Glacier Instant Retrieval-Speicher gespeichert wurden.
region-TimedStorage-GIR-SmObjects	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen kleine Objekte (kleiner als 128 KB) im S3 Glacier Instant Retrieval-Speicher gespeichert wurden.
region-TimedStorage-GlacierByteHrs	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten im S3 Glacier Flexible Retrieval-Speicher gespeichert wurden
region-TimedStorage-GlacierStaging	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten im S3 Glacier Flexible Retrieval Staging-Speicher gespeichert wurden

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-TimedStorage-INT-FA-ByteHrs	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten auf der Stufe für häufigen Zugriff des S3 Intelligent-Tiering-Speichers gespeichert wurden 5
region-TimedStorage-INT-IA-ByteHrs	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten auf der Stufe „Infrequent Access“ des S3 Intelligent-Tiering-Speichers gespeichert wurden
region-TimedStorage-INT-AA-ByteHrs	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten in der Archive Access-Stufe des S3 Intelligent-Tiering-Speichers gespeichert wurden
region-TimedStorage-INT-AIA-ByteHrs	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten in der Archive Instant Access-Stufe des S3 Intelligent-Tiering-Speichers gespeichert wurden
region-TimedStorage-INT-DAA-ByteHrs	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten in der Deep Archive Access-Stufe des S3 Intelligent-Tiering-Speichers gespeichert wurden

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-TimedStorage-RRS-ByteHrs	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten im Speicher mit reduzierter Redundanz () gespeichert wurden RRS
region-TimedStorage-SIA-ByteHrs	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten im S3 Standard-IA-Speicher gespeichert wurden
region-TimedStorage-SIA-SmObjects	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen kleine Objekte (kleiner als 128 KB) im S3 Standard-IA-Speicher 4 gespeichert wurden
region-TimedStorage-XZ-ByteHrs	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten im S3 Express One Zone-Speicher gespeichert wurden
region-TimedStorage-ZIA-ByteHrs	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen Daten im S3 One Zone-IA-Speicher gespeichert wurden
region-TimedStorage-ZIA-SmObjects	GB-Monat	Täglich	Die Anzahl der GB-Monate, in denen kleine Objekte (kleiner als 128 KB) im S3 One Zone-IA-Speicher gespeichert wurden

Verwendungstyp	Einheiten	Granularity	Beschreibung
region-Upload-XZ	GB	Stundensatz	Die Datenmenge, die in einer bestimmten Upload-Anfrage (PUToderCOPY) mit S3 Express One Zone 512 KB überschreitet

Hinweise

1. Die Global-Bucket-Hrs-FreeTier Nutzungsarten Global-Bucket-Hrs und -arten gelten für Allzweck-Buckets im gewerblichen Bereich AWS-Regionen und AWS GovCloud (US).
2. Wenn Sie eine Übertragung vor dem Abschluss beenden, kann die Menge der übertragenen Daten die Menge der von Ihrer Anwendung erhaltenen Daten überschreiten. Diese Diskrepanz kann auftreten, weil eine Anfrage zur Beendigung einer Übertragung nicht sofort ausgeführt werden kann und ein Teil der Datenmenge möglicherweise gerade übertragen wird, bis die Abbruchanfrage ausgeführt wird. Diese übertragenen Daten werden als „ausgehende“ Daten in Rechnung gestellt.
3. Wenn Objekte, die in der Speicherklasse S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive archiviert wurden, gelöscht, überschrieben oder in eine andere Speicherklasse verschoben werden, bevor die Mindestspeicherverpflichtung abgelaufen ist, die 90 Tage für S3 Glacier Instant Retrieval und S3 Glacier Flexible Retrieval oder 180 Tage für S3 Glacier Deep Archive beträgt, fällt eine anteilige Gebühr pro Gigabyte für die verbleibenden Tage an.
4. Für Objekte, die sich im S3 Standard-IA- oder S3 One Zone-IA-Speicher befinden, fällt für die verbleibenden Tage eine anteilige Gebühr pro Gigabyte an, wenn sie vor 30 Tagen gelöscht, überschrieben oder in eine andere Speicherklasse übertragen werden.
5. Für kleine Objekte (kleiner als 128 KB), die sich im S3 Standard-IA- oder S3 One Zone-IA-Speicher befinden, fällt für die verbleibenden Tage eine anteilige Gebühr pro Gigabyte an, wenn sie vor 30 Tagen gelöscht, überschrieben oder in eine andere Speicherklasse übertragen werden.
6. In der S3-Intelligent-Tiering-Speicherklasse gibt es keine mindest-abrechnungsfähige Objektgröße. Objekte, die kleiner als 128 KB sind, werden nicht überwacht oder können nicht automatisch gestuft werden. Kleinere Objekte werden immer in der S3-Intelligent-Tiering-Frequent-Access-Stufe gespeichert.

7. Wenn Sie eine `CreateMultipartUploadUploadPart`, oder `UploadPartCopy` - Anforderung für die Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive initiieren, werden Anfragen zu den S3 Standard-Anforderungstarifen abgerechnet, bis Sie den mehrteiligen Upload abgeschlossen haben. Nach Abschluss des Uploads wird für die einzelne `CompleteMultipartUpload` Anfrage der PUT Tarif für den S3 Glacier-Zielspeicher berechnet. In Bearbeitung befindliche mehrteilige Uploads für die Speicherklasse S3 Glacier Flexible Retrieval werden PUT bis zum Abschluss des Uploads als S3 Glacier Flexible Retrieval Staging Storage zu den S3-Standardspeichertarifen abgerechnet. In ähnlicher Weise werden Teile, die in Bearbeitung sind, mehrteilige Uploads für eine PUT Speicherklasse in die Speicherklasse S3 Glacier Deep Archive als S3 Glacier Deep Archive Staging Storage zu den S3-Standardspeichertarifen abgerechnet, bis der Upload abgeschlossen ist.
8. S3 Express One Zone berechnet für Anforderungsgrößen bis zu 512 KB eine Pauschalgebühr pro Anfrage. Für PUT Anfragen und GET Anfragen für den Teil der Anfrage, der größer als 512 KB ist, wird eine zusätzliche Gebühr pro GB erhoben.
9. Informationen zu den unterstützten Features der Speicherklasse S3 Express One Zone finden Sie unter Amazon S3 S3-Funktionen, die von Directory-Buckets nicht unterstützt werden .
10. Nutzungstypen mit Einheiten, die in GB abgerechnet werden, werden in den Nutzungsberichten in Byte berechnet.
11. Ein GB-Monat wird abgeleitet, indem die Gesamtzahl der GB-Stunden über einen Monat aggregiert und dann durch die Anzahl der Stunden in diesem Monat dividiert wird. Weitere Informationen finden Sie unter Häufig gestellte Fragen: Wie wird mir meine Nutzung von Amazon S3 in Rechnung gestellt und abgerechnet?

> **Note**
>
> Im Allgemeinen werden den Besitzern von S3-Buckets Anfragen mit HTTP `200 OK` erfolgreichen Antworten und HTTP 4XX Kundenfehlerantworten in Rechnung gestellt. Bucket-Besitzern werden keine HTTP 5XX Serverfehlerantworten, wie HTTP `503 Slow Down` z. B. Fehler, in Rechnung gestellt. Weitere Informationen zu S3-Fehlercodes unter HTTP 3XX und zu 4XX Statuscodes, die nicht in Rechnung gestellt werden, finden Sie unter. Abrechnung für Amazon S3 S3-Fehlerantworten Weitere Informationen zu Abrechnungsgebühren, wenn Ihr Bucket als Bucket konfiguriert ist, für den der Antragsteller zahlt, finden Sie unter. Die Gebühren bei Zahlung durch den Anforderer

Nachverfolgen von Vorgängen in Ihren Nutzungsberichten

Operationen beschreiben die Aktion, die der angegebene Nutzungstyp an Ihrem AWS Objekt oder Bucket ausgeführt hat. Vorgänge werden durch selbsterklärende Codes beschrieben, z. B. `PutObject` oder `ListBucket`. Um zu prüfen, welche Aktionen für Ihren Bucket einen bestimmten Nutzungstyp generiert haben, verwenden Sie diese Codes. Wenn Sie einen Nutzungsbericht erstellen, können Sie wählen, All Operations aufzunehmen oder eine bestimmte Operation, z. B. `GetObject`, für die ein Bericht erstellt werden soll.

Weitere Informationen

- AWS Nutzungsberichte für Amazon S3
- AWS Billing Berichte für Amazon S3
- Amazon S3 – Preise
- Amazon S3 FAQs

Abrechnung für Amazon S3 S3-Fehlerantworten

Im Allgemeinen werden S3-Bucket-Besitzern Anfragen mit HTTP `200 OK` erfolgreichen Antworten und HTTP `4XX` Kundenfehlerantworten in Rechnung gestellt. Bucket-Besitzern werden keine HTTP `5XX` Serverfehlerantworten, wie HTTP `503 Slow Down` z. B. Fehler, in Rechnung gestellt. Weitere Informationen zu Abrechnungsgebühren, wenn Ihr Bucket als Bucket konfiguriert ist, für den der Antragsteller zahlt, finden Sie unter. Die Gebühren bei Zahlung durch den Anforderer

In der folgenden Tabelle sind spezifische Fehlercodes HTTP `3XX` und `4XX` Statuscodes aufgeführt, die nicht in Rechnung gestellt werden. Für Buckets, die mit Website-Hosting konfiguriert sind, fallen weiterhin Gebühren für Anfragen und andere Gebühren an, wenn S3 ein benutzerdefiniertes Fehlerdokument zurückgibt, oder für benutzerdefinierte Weiterleitungen.

> **Note**
>
> Für `AccessDenied` (HTTP`403 Forbidden`) berechnet S3 dem Bucket-Besitzer keine Gebühren, wenn die Anfrage außerhalb des individuellen AWS Kontos des Bucket-Besitzers oder der AWS Organisation des Bucket-Besitzers initiiert wird.

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
301 Moved Permanently (301 Dauerhaft verschoben)	PermanentRedirect	Der Bucket, auf den Sie zugreifen möchten, muss über den angegebenen Endpunkt adressiert werden. Senden Sie alle future Anfragen an diesen Endpunkt.	
	PermanentRedirectControlError	Der API Vorgang, auf den Sie zugreifen möchten, muss über den angegebenen Endpunkt ausgeführt werden. Senden Sie alle future Anfragen an diesen Endpunkt.	
307 Temporäre Weiterleitung	TemporaryRedirect	Sie werden zum Bucket umgeleitet, während der Domain Name System (DNS) -Server aktualisiert wird.	
400 Ungültige Anfrage	AuthorizationHeaderMalformed	Der von Ihnen angegebene Autorisierungsheader ist ungültig.	
	AuthorizationQueryParametersError	Die von Ihnen angegebenen Autorisierungsabfrageparameter sind nicht gültig.	
	ConnectionClosedByRequester	Wird an den ursprünglichen Aufrufer zurückgegeben, wenn beim Lesen	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
		des WriteGetObjectResponse Hauptteils ein Fehler aufgetreten ist.	
	DeviceNotActiveError	Das Gerät ist derzeit nicht aktiv.	
	EndpointNotFound	Direkte Anfragen an den richtigen Endpunkt.	
	ExpiredToken	Das bereitgestellte Token ist abgelaufen.	
	IllegalLocationConstraintException	Sie versuchen, auf einen Bucket aus einer anderen Region als der Region zuzugreifen, in der der Bucket existiert. Verwenden Sie die `--region` Option, um diesen Fehler zu vermeiden. Beispiel: `aws s3 cp awsexample.txt s3://amzn-s3-demo-bucket/ --region ap-east-1` .	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
	InvalidArgument	Dieser Fehler kann aus folgenden Gründen auftreten: • Das angegebene Argument war nicht gültig. • In der Anfrage fehlte ein erforderlicher Header. • Das angegebene Argument war unvollständig oder hatte das falsche Format. • Das angegebene Argument muss eine Länge größer oder gleich 3 haben.	
	InvalidBucketOwnerAWSAccountID	Der Wert des Parameters Expected Bucket Owner muss eine AWS-Konto ID sein.	
	InvalidDigest	Der von Ihnen angegebene Inhalts- MD5 oder Prüfsummenwert ist nicht gültig.	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
	InvalidEncryptionAlgorithmError	Die von Ihnen angegebene Verschlüsselungsanfrage ist nicht gültig. Der gültige Wert lautet AES256.	
	InvalidHostHeader	Die in der Anfrage angegebenen Host-Header verwendeten die falsche Adressierung.	
	InvalidHttpMethod	Die Anfrage wird mit einer unerwarteten HTTP Methode gestellt.	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
	InvalidRequest	Dieser Fehler kann aus folgenden Gründen auftreten: • Die Anfrage verwendet die falsche Signaturversion. Verwenden Sie `AWS4-HMAC-SHA256` (Signaturversion 4). • Ein Access Point kann nur für einen vorhandenen Bucket erstellt werden. • Der Access Point befindet sich nicht in einem Zustand, in dem er gelöscht werden kann. • Ein Access Point kann nur für einen vorhandenen Bucket aufgeführt werden. • Das nächste Token ist nicht gültig. • In einer Lebenszyklusregel muss mindestens eine Aktion angegeben werden. •	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
		Es muss mindestens eine Lebenszyklusregel angegeben werden. • Die Anzahl der Lebenszyklusregeln darf die zulässige Grenze von 1000 Regeln nicht überschreiten. • Der Bereich für den `MaxResults` Parameter ist nicht gültig. • SOAPAnfragen müssen über eine HTTPS Verbindung gestellt werden. • Amazon S3 Transfer Acceleration wird für Buckets mit nicht DNS konformen Namen nicht unterstützt. • Amazon S3 Transfer Acceleration wird für Buckets mit Punkten (.) im Namen nicht unterstützt. •	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
		Der Amazon S3 Transfer Acceleration Acceleration-Endpunkt unterstützt nur Anfragen im virtuellen Stil. • Amazon S3 Transfer Acceleration ist für diesen Bucket nicht konfiguriert. • Amazon S3 Transfer Acceleration ist in diesem Bucket deaktiviert. • Amazon S3 Transfer Acceleration wird in diesem Bucket nicht unterstützt. Wenn Sie Hilfe benötigen, wenden Sie sich an AWS Support. • Amazon S3 Transfer Acceleration kann für diesen Bucket nicht aktiviert werden. Wenn Sie Hilfe benötigen, wenden Sie sich an AWS Support. •	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
		Widersprüchliche Werte in HTTP Headern und Abfrageparametern. • Widersprüchliche Werte in HTTP Kopfzeilen und Formularfeldern. POST • CopyObject Anfrage für Objekte mit einer Größe von mehr als 5 GB.	
	InvalidSessionException	Wird zurückgegeben, wenn die Sitzung nicht mehr existiert, weil das Zeitlimit überschritten wurde oder sie abgelaufen ist.	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
	InvalidSignature	Die vom Server berechnete Anforderungssignatur stimmt nicht mit der von Ihnen angegebenen Signatur überein. Überprüfen Sie Ihren AWS geheimen Zugriffsschlüssel und Ihre Signaturmethode. Weitere Informationen finden Sie unter RESTAnfragen signieren und authentifizieren.	
	Ich nvalidSOAPRequest	Der SOAP Text der Anfrage ist nicht gültig.	
	InvalidStorageClass	Die von Ihnen angegebene Speicherklasse ist ungültig.	
	InvalidTag	Ihre Anfrage enthält eine ungültige Tag-Einabe. Ihre Anfrage könnte beispielsweise doppelte Schlüssel, Schlüssel oder Werte, die zu lang sind, oder Systemtags enthalten.	
	InvalidToken	Das angegebene Token ist falsch formatiert oder anderweitig nicht gültig.	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
	Ungültig URI	Die angegebene URI Datei konnte nicht analysiert werden.	
	KeyTooLongError	Ihr Schlüssel ist zu lang.	
	KMS.DisabledException	Die Anfrage wurde abgelehnt, weil der angegebene KMS Schlüssel nicht aktiviert ist.	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
	KMS.InvalidKeyUsageException	Die Anfrage wurde aus einem der folgenden Gründe abgelehnt: • Der KeyUsage Wert des KMS Schlüssels ist mit dem API Vorgang nicht kompatibel. • Der für den Vorgang angegebene Verschlüsselungs- oder Signaturalgorithmus ist nicht mit dem Typ des Schlüsselmaterials im KMS Schlüssel (KeySpec) kompatibel. Zum Verschlüsseln, Entschlüsseln, erneuten Verschlüsseln und Generieren von Datenschlüsseln muss der KeyUsage Wert _ sein. ENCRYPT DECRYPT Zum Signieren und Überprüfen von Nachrichten muss der Wert _ sein. KeyUsage SIGN VERIFY Zum Generieren und Überprüfen von Nachrichtenauthentifizierungscodes	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
		(MACs) KeyUsage muss GENERATE _ _ VERIFY sein. MAC Um wichtige Vertragsgeheimnisse abzuleiten, KeyUsage muss der Wert _ seinKEY. AGREEMENT Verwenden Sie die DescribeKey Operation, um den Wert KeyUsage eines KMS Schlüssels zu ermitteln. Verwenden Sie die DescribeKey Operation, um die Verschlüsselungs- oder Signierungsalgorithmen zu finden, die für einen bestimmten KMS Schlüssel unterstützt werden.	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
	KMS.KMSInvalidStateException	Die Anfrage wurde abgelehnt, da der Status der angegebenen Ressource für diese Anfrage nicht gültig ist. Diese Ausnahme bedeutet eine der folgenden Ursachen: • Der Schlüsselstatus des KMS Schlüssels ist nicht mit dem Vorgang kompatibel. Verwenden Sie die DescribeKey Operation, um den Schlüsselstatus zu ermitteln. Weitere Informationen darüber, welche Schlüsselstatus mit den einzelnen KMS Vorgängen kompatibel sind, finden Sie AWS KMS keys im AWS Key Management Service Entwicklerhandbuch unter Schlüsselstatus von. • Bei kryptografischen Vorgängen mit KMS Schlüsseln in benutzerdefinierten Schlüsselspeichern	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
		stellt diese Ausnahme einen allgemeinen Fehler mit vielen möglichen Ursachen dar. Informationen zur Identifizierung der Ursache finden Sie in der Fehlermeldung, die der Ausnahme beiliegt.	
	KMS.NotFoundException	Die Anfrage wurde abgelehnt, weil die angegebene Entität oder Ressource nicht gefunden werden konnte.	
	LambdaInvalidResponse	Kehrt zum ursprünglichen Anrufer zurück, wenn er mit WriteGetObjectResponse ValidationError auf AWS Lambda antwortet. Weitere Informationen finden Sie in der ValidationError Nachricht. Nicht alle Fälle ValidationError führen zu einem LambdaInvalidResponse Fehler.	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
	LambdaInvocationFailed	Der Aufruf der Lambda-Funktion ist fehlgeschlagen. Aufrufer erhalten möglicherweise den folgenden Fehler, wenn S3 Object Lambda die konfigurierte Lambda-Funktion nicht erfolgreich aufrufen kann. Die Fehlermeldung kann Details zu einem eventuellen Fehler enthalten, den der AWS Lambda Dienst beim Aufrufen der Funktion zurückgibt (z. B. Statuscode, Fehlercode, Fehlermeldung und Anforderungs-ID).	
	MalformedACLError	Die von ACL Ihnen angegebene Datei war falsch formatiert oder wurde nicht anhand unseres veröffentlichten Schemas validiert.	
	MalformedPOSTRequest	Der Hauptteil Ihrer POST Anfrage besteht nicht aus wohlgeformten Multipart/Form-Daten.	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
	Falsch formatiert XML	Die von XML Ihnen angegebene Datei war falsch formatiert oder wurde nicht anhand unseres veröffentlichten Schemas validiert.	
	MaxPostPreDataLengthExceededError	Ihre POST Anforderungsfelder vor der Upload-Datei waren zu groß.	
	MetadataTooLarge	Ihre Metadaten-Header überschreiten die maximal zulässige Metadatengröße.	
	MissingAttachment	Es wurde ein SOAP Anhang erwartet, aber keiner gefunden.	
	MissingRequestBodyError	Sie haben ein leeres XML Dokument als Anfrage gesendet.	
	MissingSecurityHeader	In Ihrer Anfrage fehlt ein erforderlicher Header.	
	NoLoggingStatusForKey	So etwas wie eine Unterressource für den Protokollierungsstatus eines Schlüssels gibt es nicht.	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
	NotDeviceOwnerError	Das Gerät, das das Token generiert hat, gehört nicht dem authentifizierten Benutzer.	
	ResponseInterrupted	Wird an den ursprünglichen Anrufer zurückgegeben, wenn beim Lesen des WriteGetObjectResponse Textes ein Fehler auftritt.	
	RequestHeaderSectionTooLarge	Der Anforderungsheader und die Abfrageparameter, mit denen die Anfrage gestellt wurde, überschreiten die maximal zulässigen Größen	
	TokenCodeInvalidError	Die von Ihnen angegebene Seriennummer und/oder der Token-Code sind nicht gültig.	
	UnexpectedContent	Diese Anfrage enthält Inhalte, die nicht unterstützt werden.	
	UnsupportedArgument	Die Anfrage enthielt ein nicht unterstütztes Argument.	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
	UnsupportedSignature	Die bereitgestellte Anfrage ist mit einer nicht unterstützten STS Token-Version signiert, oder die Signaturversion wird nicht unterstützt.	
	UserKeyMustBeSpecified	Die POST Bucket-Anfrage muss den angegebenen Feldnamen enthalten. Wenn es angegeben ist, überprüfen Sie die Reihenfolge der Felder.	
	IncorrectEndpoint	Der angegebene Bucket ist in einer anderen Region vorhanden. Direkte Anfragen an den richtigen Endpunkt.	
	ValidationError	Bei der WriteGetObjectResponse API Operation können Validierungsfehler auftreten, die aus zahlreichen Gründen auftreten können. Weitere Informationen finden Sie in der Fehlermeldung.	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
403 Verboten	RequestTimeTooSkewed	Der Unterschied zwischen der Anforderungszeit und der Serverzeit ist zu groß.	
	SignatureDoesNotMatch	Die vom Server berechnete Anforderungssignatur stimmt nicht mit der von Ihnen angegebenen Signatur überein. Überprüfen Sie Ihren AWS geheimen Zugriffsschlüssel und Ihre Signaturmethode. Weitere Informationen finden Sie unter RESTAuthentifizierung und SOAPAuthentifizierung.	
	NotSignedUp	Ihr Konto ist nicht für den Amazon S3 S3-Service registriert. Sie müssen sich registrieren, bevor Sie Amazon S3 verwenden können. Sie können sich wie folgt registrierenURL: https://aws.amazon.com/s3	
	InvalidSecurity	Die angegebenen Sicherheitsanmeldedaten sind nicht gültig.	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
	InvalidPayer	Der gesamte Zugriff auf dieses Objekt wurde deaktiviert. Weitere Unterstützung finden Sie unter Kontaktieren Sie uns.	
	InvalidAccessKeyId	Die von Ihnen angegebene AWS Zugriffsschlüssel-ID ist in unseren Aufzeichnungen nicht vorhanden.	
	AccountProblem	Es liegt ein Problem mit Ihrem vor AWS-Konto , das verhindert, dass der Vorgang erfolgreich abgeschlossen werden kann. Weitere Unterstützung finden Sie unter Kontaktieren Sie uns.	
	UnauthorizedAccessError	Gilt nur in chinesischen Regionen. Wird zurückgegeben, wenn eine Anfrage an einen Bucket gestellt wird, der keine ICP Lizenz hat. Weitere Informationen finden Sie unter ICPRecordal.	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
	U nexpectedIPError	Gilt nur in chinesischen Regionen. Diese Anfrage wurde abgelehnt, da die IP unerwartet war.	
	MissingAuthenticationToken	Die Anfrage wurde nicht signiert.	
	LambdaPermissionError	Der Aufrufer ist nicht berechtigt, die Lambda-Funktion aufzurufen. Der Aufrufer muss die Erlaubnis haben, die Lambda-Funktion aufzurufen. Überprüfen Sie die dem Aufrufer beigefügten Richtlinien und stellen Sie sicher, dass sie `lambda:InvokeFunction` für die konfigurierte Funktion verwendet werden dürfen. Die Fehlermeldung kann Details zu einem eventuellen Fehler enthalten, den der Lambda-Dienst beim Aufrufen der Funktion zurückgibt (z. B. Statuscode, Fehlercode, Fehlermeldung und Anforderungs-ID).	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes
404 Not Found (404 Nicht gefunden)	LambdaNotFound	Die AWS Lambda Funktion wurde nicht gefunden. Die konfigurierte Lambda-Funktion, Version oder Alias wurde beim Versuch, sie aufzurufen, nicht gefunden. Stellen Sie sicher, dass die Konfiguration des S3 Object Lambda Access Points auf die richtige Lambda-Funktion verweist. ARN Die Fehlermeldung kann Details zu einem eventuellen Fehler enthalten, den der AWS Lambda Dienst beim Aufrufen der Funktion zurückgibt (z. B. Statuscode, Fehlercode, Fehlermeldung und Anforderungs-ID).
	NoSuchAsyncRequest	Die angegebene Anfrage wurde nicht gefunden.
	NoSuchObjectLockConfiguration	Das angegebene Objekt hat keine ObjectLock Konfiguration.

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes	
	NoSuchUpload	Der angegebene mehrteilige Upload ist nicht vorhanden. Die Upload-ID ist möglicherweise nicht gültig, oder der mehrteilige Upload wurde möglicherweise abgebrochen oder abgeschlossen.	
	NoSuchWebsiteConfiguration	Der angegebene Bucket hat keine Website-Konfiguration.	
	NoTransformationDefined	Für diesen Object Lambda Access Point wurde keine Transformation gefunden.	
	ObjectLockConfigurationNotFoundError	Die Object Lock-Konfiguration ist für diesen Bucket nicht vorhanden.	
405 Methode nicht zulässig	MethodNotAllowed	Die angegebene Methode ist für diese Ressource nicht zulässig.	

HTTPStatuscode	Fehlercode	Beschreibung des Fehlercodes
409 Konflikt	BucketAlreadyExists	Der angeforderte Bucket-Name ist nicht verfügbar. Der Bucket-Namespace wird von allen Benutzern des Systems gemeinsam genutzt. Geben Sie einen anderen Namen an und versuchen Sie es erneut.
	InvalidBucketState	Die Anfrage ist für den aktuellen Status des Buckets nicht gültig.
	OperationAborted	Für diese Ressource wird derzeit ein bedingter Vorgang ausgeführt, bei dem ein Konflikt aufgetreten ist. Versuchen Sie es erneut.
411 Länge erforderlich	MissingContentLength	Sie müssen den HTTP Content-Length-Header angeben.
4.1.2 Die Vorbedingung ist fehlgeschlagen	RequestIsNotMultiPartContent	Bei einer POST Bucket-Anfrage muss es sich um eine Anlage vom Typ Multipart/Form-Data handeln.
4.1.6 Der angeforderte Bereich ist nicht erfüllbar	InvalidRange	Der angeforderte Bereich ist für die Anfrage nicht gültig. Versuchen Sie es mit einem anderen Bereich.

Amazon S3 S3-Speicherklassen verstehen und verwalten

Jedem Objekt in Amazon S3 ist eine Speicherklasse zugeordnet. Standardmäßig werden Objekte in S3 in der Speicherklasse S3 Standard gespeichert. Amazon S3 bietet jedoch eine Reihe anderer Speicherklassen für die Objekte, die Sie speichern. Sie wählen eine Speicherklasse abhängig von Ihrem Anwendungsszenario und den Leistungsanforderungen für den Zugriff aus. Durch die Auswahl einer Speicherklasse, die für Ihren Anwendungsfall konzipiert ist, können Sie die Speicherkosten, Leistung und Verfügbarkeit Ihrer Objekte optimieren. Alle diese Speicherklassen bieten höchste Haltbarkeit.

Die folgenden Abschnitte enthalten Details zu den verschiedenen Speicherklassen und zum Festlegen einer Speicherklasse für Ihre Objekte.

Themen

- Speicherklassen für Objekte mit häufigem Zugriff
- Speicherklasse zur automatischen Optimierung von Daten mit sich ändernden oder unbekannten Zugriffsmustern
- Speicherklassen für Objekte mit seltenem Zugriff
- Speicherklassen für Objekte, auf die selten zugegriffen wird
- Speicherklasse für Amazon S3 in Outposts
- Vergleich der Amazon-S3-Speicherklassen
- Einrichten der Speicherklasse eines Objekts
- Amazon S3 analytics – Speicherklassen-Analyse
- Verwaltung der Speicherkosten mit Amazon S3 Intelligent-Tiering
- Grundlegendes zu S3 Glacier-Speicherklassen für die langfristige Datenspeicherung
- Arbeiten mit archivierten Objekten

Speicherklassen für Objekte mit häufigem Zugriff

Für leistungssensible Anwendungsfälle (die eine Zugriffszeit von Millisekunden erfordern) und Daten mit häufigem Zugriff stellt Amazon S3 die folgenden Speicherklassen bereit:

- S3 Standard – Die Standardspeicherklasse. Wenn Sie beim Upload eines Objekts keine Speicherklasse angeben, weist Amazon S3 die Speicherklasse S3 Standard zu. Um Ihnen

zu helfen, die Kosten zwischen S3 Standard und S3 Standard-IA zu optimieren, können Sie Folgendes verwenden. Amazon S3 analytics – Speicherklassen-Analyse

- S3 Express One Zone – Amazon S3 Express One Zone ist eine hochleistungsfähige Amazon-S3-Speicherklasse mit einer einzelnen Zone, die entwickelt wurde, um einen konsistenten Datenzugriff im einstelligen Millisekundenbereich für Ihre Daten und für latenzempfindliche Anwendungen zu gewährleisten. S3 Express One Zone ist die derzeit verfügbare Cloud-Objektspeicherklasse mit der niedrigsten Latenz. Sie bietet eine bis zu zehnmal schnellere Datenzugriffsgeschwindigkeit und 50 Prozent niedrigere Anforderungskosten als S3 Standard. Mit S3 Express One Zone werden Ihre Daten redundant auf mehreren Geräten innerhalb einer einzigen Availability Zone gespeichert. Weitere Informationen finden Sie unter S3 Express One Zone.
- Reduzierte Redundanz — Die Speicherklasse Reduced Redundancy Storage (RRS) wurde für unkritische, reproduzierbare Daten konzipiert, die mit weniger Redundanz gespeichert werden können als die Speicherklasse S3 Standard.

  > **Important**
  >
  > Wir empfehlen, diese Speicherklasse nicht zu verwenden. Die Speicherklasse S3 Standard ist kostengünstiger.

  Aus Gründen der Haltbarkeit wird bei RRS Objekten ein durchschnittlicher jährlicher Verlust von 0,01 Prozent der Objekte erwartet. Wenn ein RRS Objekt verloren geht und Anfragen an dieses Objekt gestellt werden, gibt Amazon S3 einen 405-Fehler zurück.

Speicherklasse zur automatischen Optimierung von Daten mit sich ändernden oder unbekannten Zugriffsmustern

S3 Intelligent-Tiering ist eine Amazon-S3-Speicherklasse zur Optimierung der Speicherkosten durch automatisches Verschieben von Daten auf die kostengünstigste Zugriffsebene ohne Leistungsauswirkung oder Betriebsaufwand. S3 Intelligent-Tiering ist die einzige Cloud-Speicherklasse, die automatische Kosteneinsparungen durch Verschieben von Daten auf granularer Objektebene zwischen den Zugriffsebenen ermöglicht, wenn sich die Zugriffsmuster ändern. S3 Intelligent-Tiering ist die ideale Speicherklasse, wenn Sie die Speicherkosten für Daten mit unbekannten oder sich ändernden Zugriffsmustern optimieren möchten. Bei S3 Intelligent-Tiering werden keine Abrufgebühren erhoben.

Gegen eine geringe monatliche Gebühr für Objektüberwachung und Automatisierung überwacht S3 Intelligent-Tiering die Zugriffsmuster und verschiebt die Objekte, auf die nicht zugegriffen wurde, automatisch in kostengünstigere Zugriffsebenen. S3 Intelligent-Tiering ermöglicht automatische Speicherkosteneinsparungen in drei Zugriffsebenen mit niedriger Latenz und hohem Durchsatz. Für Daten, auf die asynchron zugegriffen werden kann, können Sie innerhalb der S3 Intelligent-Tiering-Speicherklasse die automatische Archivfunktionen aktivieren. S3 Intelligent-Tiering ist auf 99,9 % Verfügbarkeit und 99,9999999 % Haltbarkeit ausgelegt.

S3 Intelligent-Tiering speichert Objekte automatisch in drei Zugriffsebenen:

- Frequent Access (Häufige Zugriffe) – In S3 Intelligent-Tiering hochgeladene oder übertragene Objekte werden automatisch in der Ebene Frequent Access gespeichert.
- Infrequent Access (Seltene Zugriffe) – S3 Intelligent-Tiering verschiebt Objekte, auf die 30 aufeinanderfolgende Tage lang nicht zugegriffen wurde, in die Ebene Infrequent Access.
- Archive Instant Access (Archiv für sofortigen Zugriff) – Mit S3 Intelligent-Tiering werden alle vorhandenen Objekte, auf die 90 aufeinanderfolgende Tage lang nicht zugegriffen wurde, automatisch in die Ebene Archive Instant Access verschoben.

Zusätzlich zu diesen drei Stufen bietet S3 Intelligent-Tiering zwei optionale Archivzugriffsebenen:

- Archive Access (Archivzugriff) – S3 Intelligent-Tiering bietet Ihnen die Möglichkeit, die Archivzugriffsebene für Daten zu aktivieren, auf die asynchron zugegriffen werden kann. Nach der Aktivierung archiviert die Ebene Archive Access automatisch Objekte, auf die mindestens 90 aufeinanderfolgende Tage lang nicht zugegriffen wurde.
- Deep Archive Access (tiefer Archivzugriff) – S3 Intelligent-Tiering bietet Ihnen die Möglichkeit, die Stufe Deep Archive Access für Daten zu aktivieren, auf die asynchron zugegriffen werden kann. Nach der Aktivierung archiviert die Ebene Deep Archive Access automatisch Objekte, auf die mindestens 180 aufeinanderfolgende Tage lang nicht zugegriffen wurde.

> **Note**
>
> - Aktivieren Sie die Stufe Archive Access nur für 90 Tage, wenn Sie die Stufe Instant Access Archiv umgehen möchten. Die Stufe Archive Access bietet etwas kostengünstigeren Speicherplatz mit minute-to-hour Abrufzeiten. Die Stufe Archive Instant Access bietet Millisekunden-Zugriff und hohe Durchsatzleistung.

- Aktivieren Sie die Ebenen Archivzugriff und Deep Archive Access nur, wenn Ihre Anwendung asynchron auf Ihre Objekte zugreifen kann. Wenn das Objekt, das Sie abrufen, in den Stufen Archive Access oder Deep Archive Access gespeichert ist, stellen Sie das Objekt zunächst mithilfe von `RestoreObject` wieder her.

Sie können neu erstellte Daten in S3 Intelligent-Tiering verschieben und dies als Standardspeicherklasse festlegen. Sie können auch wählen, ob Sie eine oder beide Archivzugriffsebenen aktivieren möchten, indem Sie die `PutBucketIntelligentTieringConfiguration`APIOperation AWS CLI, die oder die Amazon S3 S3-Konsole verwenden. Weitere Informationen zur Verwendung von S3 Intelligent-Tiering und zur Aktivierung der Archivzugriffsebenen finden Sie unter Verwenden von S3 Intelligent-Tiering.

Um auf Objekte in den Stufen „Archive Access“ oder „Deep Archive Access“ zugreifen zu können, müssen Sie sie zunächst wiederherstellen. Weitere Informationen finden Sie unter Wiederherstellen von Objekten aus den Stufen S3 Intelligent-Tiering Archive Access und Deep Archive Access.

> **Note**
>
> Wenn ein Objekt kleiner als 128 KB ist, wird es nicht überwacht und ist nicht für das Auto-Tiering geeignet. Kleinere Objekte werden immer in der Stufe für häufige Zugriffe gespeichert. Weitere Informationen zu S3 Intelligent-Tiering finden Sie unter S3-Intelligent-Tiering-Zugriffsebenen.

Speicherklassen für Objekte mit seltenem Zugriff

Die Speicherklassen S3 Standard-IA und S3 One Zone-IA sind für langlebige Daten, auf die selten zugegriffen wird, konzipiert. (IA steht für Infrequent Access (seltener Zugriff).) S3 Standard-IA- und S3 One Zone-IA-Objekte sind für den Zugriff in Millisekunden verfügbar (ähnlich wie die Speicherklasse S3 Standard). Da Amazon S3 eine Abrufgebühr für diese Objekte erhebt, sind sie am besten für Daten mit seltenem Zugriff geeignet. Informationen zu den Preisen finden Sie unter Amazon S3 – Preise.

Beispielsweise können Sie die Speicherklassen S3 Standard-IA und S3 One Zone-IA für Folgendes auswählen:

- Zum Speichern von Sicherungen.

- Für ältere Daten, auf die selten zugegriffen wird, für die aber dennoch ein Zugriff in Millisekunden erforderlich ist. Beim Upload von Daten können Sie beispielsweise die S3 Standard-Speicherklasse auswählen und dann Amazon S3 mithilfe der Lebenszyklus-Konfiguration anweisen, die Objekte in die Klasse S3 Standard-IA oder S3 One Zone-IA zu überführen.

  Weitere Informationen zur Lebenszyklusverwaltung finden Sie unter Verwaltung des Lebenszyklus von Objekten.

> **Note**
>
> Die Speicherklassen S3 Standard-IA und S3 One Zone-IA sind für Objekte mit mehr als 128 KB geeignet, die Sie für mindestens 30 Tage aufbewahren wollen. Ist ein Objekt kleiner als 128 KB, fallen Amazon-S3-Gebühren für 128 KB an. Wenn Sie ein Objekt vor dem Ablauf der 30-tägigen Mindestspeicherdauer löschen, werden Ihnen 30 Tage berechnet. Für Objekte, die vor Ablauf der 30-Tage-Frist gelöscht, überschrieben oder in eine andere Speicherklasse verschoben werden, wird die übliche nutzungsabhängige Gebühr sowie eine anteilige Gebühr für die restlichen Tage des 30-Tage-Minimums in Rechnung gestellt. Informationen zu den Preisen finden Sie unter Amazon S3 – Preise.

Diese Speicherklassen weisen folgende Unterschiede auf:

- S3 Standard-IA – Amazon S3 speichert die Objektdaten redundant in mehreren geografisch getrennten Availability Zones (ähnlich wie die Speicherklasse S3 Standard). S3 Standard-IA-Objekte sind von einem Availability Zone-Ausfall nicht betroffen. Diese Speicherklasse bietet mehr Verfügbarkeit und Stabilität als die Klasse S3 One Zone-IA. Um Ihnen zu helfen, die Kosten zwischen S3 Standard und S3 Standard-IA zu optimieren, können Sie Folgendes verwenden Amazon S3 analytics – Speicherklassen-Analyse
- S3 One Zone-IA – Amazon S3 speichert die Objektdaten nur in einer Availability Zone, daher ist diese Variante kostengünstiger als S3 Standard-IA. Allerdings sind die Daten bei einem physischen Ausfall der Availability Zone (aufgrund von Katastrophen wie z. B. Erdbeben und Überflutungen) nicht sicher. Die Speicherklasse S3 One Zone-IA bietet dieselbe Zuverlässigkeit wie S3 Standard-IA, ist jedoch weniger verfügbar und weniger ausfallsicher. Einen Vergleich der Zuverlässigkeit und Verfügbarkeit von Speicherklassen finden Sie unter Vergleich der Amazon-S3-Speicherklassen am Ende dieses Abschnitts. Informationen zu den Preisen finden Sie unter Amazon S3 – Preise.

Wir empfehlen Folgendes:

- S3 Standard-IA – Verwenden Sie diese Klasse für die primäre (oder einzige) Kopie der Daten, die nicht neu erstellt werden können.
- S3 One Zone-IA — Verwenden Sie diese Option, wenn Sie die Daten bei einem Ausfall der Availability Zone neu erstellen können, für Objektreplikate bei der Konfiguration der regionsübergreifenden S3-Replikation (). CRR

Speicherklassen für Objekte, auf die selten zugegriffen wird

Die Speicherklassen S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive sind für die kostengünstige, langfristige Datenspeicherung und Datenarchivierung konzipiert. Diese Speicherklassen erfordern eine Mindestspeicherdauer und Mindestabrufgebühren, sodass sie für Daten, auf die selten zugegriffen wird, am effektivsten sind. Weitere Informationen zu S3 Glacier-Speicherklassen finden Sie unter. Grundlegendes zu S3 Glacier-Speicherklassen für die langfristige Datenspeicherung

Amazon S3 bietet die folgenden S3 Glacier-Speicherklassen:

- S3 Glacier Instant Retrieval — Wird für Langzeitdaten verwendet, auf die selten zugegriffen wird und deren Abruf Millisekunden erfordert. Daten in dieser Speicherklasse sind für den Echtzeitzugriff verfügbar.
- S3 Glacier Flexible Retrieval – Diese Klasse eignet sich für Archive, bei denen Teile der Daten möglicherweise innerhalb von wenigen Minuten abgerufen werden müssen. Daten in dieser Speicherklasse sind archiviert und für den Echtzeitzugriff nicht verfügbar.
- S3 Glacier Deep Archive – Verwenden Sie diese Speicherklasse zur Archivierung von Daten, auf die nur selten zugegriffen werden muss. Daten in dieser Speicherklasse sind archiviert und für den Echtzeitzugriff nicht verfügbar.

Abrufen archivierter Objekte

Sie können die Speicherklasse eines Objekts auf S3 Glacier Flexieble Retrieval oder S3 Glacier Deep Archive setzen, genau so wie für andere Speicherklassen, wie im Abschnitt Einrichten der Speicherklasse eines Objekts beschrieben. S3 Glacier Flexible Retrieval- und S3 Glacier Deep Archive Archive-Objekte werden jedoch archiviert und sind nicht für den Echtzeitzugriff verfügbar. Weitere Informationen finden Sie unter Grundlegendes zum Archivspeicher in S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive.

> **Note**
>
> Wenn Sie S3 Glacier-Speicherklassen verwenden, verbleiben Ihre Objekte in Amazon S3. Sie können nicht direkt über den separaten Amazon-S3-Glacier-Service darauf zugreifen. Informationen zum Amazon S3 Glacier-Service finden Sie im Amazon S3 Glacier Developer Guide.

Speicherklasse für Amazon S3 in Outposts

Mit Amazon S3 on Outposts können Sie S3-Buckets auf Ihren AWS Outposts Ressourcen erstellen und Objekte vor Ort für Anwendungen speichern und abrufen, die lokalen Datenzugriff, lokale Datenverarbeitung und Datenresidenz erfordern. Sie können dieselben API Operationen und Funktionen AWS Outposts wie auf Amazon S3 verwenden, einschließlich Zugriffsrichtlinien, Verschlüsselung und Tagging. Sie können S3 auf Outposts über das AWS Management Console, AWS CLI AWS SDKs, oder REST API verwenden.

S3 in Outposts bietet eine neue Speicherklasse: S3 Outposts (`OUTPOSTS`). Die Speicherklasse S3 Outposts ist nur für Objekte verfügbar, die in Buckets in Outposts gespeichert sind. Wenn Sie versuchen, diese Speicherklasse mit einem S3-Bucket in einem zu verwenden AWS-Region, tritt ein `InvalidStorageClass` Fehler auf. Sie erhalten dieselbe Fehlermeldung, wenn Sie versuchen, andere S3-Speicherklassen mit Objekten zu verwenden, die in S3-on-Outposts-Buckets gespeichert sind.

Objekte, die in der Speicherklasse S3 Outposts (`OUTPOSTS`) gespeichert sind, werden immer mit serverseitiger Verschlüsselung mit von Amazon S3 verwalteten Verschlüsselungsschlüsseln (SSE-S3) verschlüsselt. Weitere Informationen finden Sie unter Serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden (SSE-S3).

Sie können sich auch explizit dafür entscheiden, Objekte, die in der Speicherklasse S3 Outposts gespeichert sind, zu verschlüsseln, indem Sie serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (-C) verwenden. SSE Weitere Informationen finden Sie unter Verwenden Sie serverseitige Verschlüsselung mit vom Kunden bereitgestellten Schlüsseln (-C) SSE.

> **Note**
>
> S3 on Outposts unterstützt keine serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-KMS).

Weitere Informationen zu S3 on Outposts finden Sie unter Was ist S3 on Outposts im Amazon S3 on Outposts User Guide.

Vergleich der Amazon-S3-Speicherklassen

In der folgenden Tabelle werden die Speicherklassen verglichen, einschließlich ihrer Verfügbarkeit, Haltbarkeit, Mindestspeicherdauer und anderer Faktoren.

Storage Class	Designed for	Durability (designed for)	Availability (designed for)	Availability Zones	Min storage duration	Min billable object size	Other Considerations
STANDARD	Frequently accessed data	99.999999999%	99.99%	>= 3	None	None	None
STANDARD_IA	Long-lived, infrequently accessed data	99.999999999%	99.9%	>= 3	30 days	128 KB	Per GB retrieval fees apply.
INTELLIGENT_TIERING	Long-lived data with changing or unknown access patterns	99.999999999%	99.9%	>= 3	30 days	None	Monitoring and automation fees per object apply. No retrieval fees.
ONEZONE_IA	Long-lived, infrequently accessed, non-critical data	99.999999999%	99.5%	1	30 days	128 KB	Per GB retrieval fees apply. Not resilient to the loss of the Availability Zone.
GLACIER	Long-term data archiving with retrieval times ranging from minutes to hours	99.999999999%	99.99% (after you restore objects)	>= 3	90 days	None	Per GB retrieval fees apply. You must first restore archived objects before you can access them. For more information, see Restoring Archived Objects.
DEEP_ARCHIVE	Archiving rarely accessed data with a default retrieval time of 12 hours	99.999999999%	99.99% (after you restore objects)	>= 3	180 days	None	Per GB retrieval fees apply. You must first restore archived objects before you can access them. For more information, see Restoring Archived Objects.
RRS (Not recommended)	Frequently accessed, non-critical data	99.99%	99.99%	>= 3	None	None	None

* S3 Glacier Flexible Retrieval erfordert 40 KB zusätzliche Metadaten für jedes archivierte Objekt. Dazu gehören 32 KB Metadaten, die mit dem Tarif von S3 Glacier Flexible Retrieval abgerechnet werden (erforderlich, um Ihre Daten zu identifizieren und abzurufen), und weitere 8 KB Daten, die zum S3-Standardtarif berechnet werden. Der S3-Standardtarif ist erforderlich, um den benutzerdefinierten Namen und die Metadaten für Objekte zu verwalten, die in S3 Glacier Flexible Retrieval archiviert wurden. Weitere Informationen zu Speicherklassen finden Sie unter Amazon-S3-Speicherklassen.

** S3 Glacier Deep Archive benötigt 40 KB an zusätzlichen Metadaten für jedes archivierte Objekt. Dazu gehören 32 KB Metadaten, die mit dem Tarif von S3 Glacier Deep Archive abgerechnet werden

(erforderlich, um Ihre Daten zu identifizieren und abzurufen), und weitere 8 KB Daten, die zum S3-Standardtarif berechnet werden. Der S3-Standardtarif ist erforderlich, um den benutzerdefinierten Namen und die Metadaten für Objekte zu verwalten, die in Amazon S3 Glacier Deep Archive archiviert wurden. Weitere Informationen zu Speicherklassen finden Sie unter Amazon-S3-Speicherklassen.

Beachten Sie, dass alle Speicherklassen außer S3 One Zone-IA und S3 Express One Zone gegenüber dem physischen Verlust einer Availability Zone aufgrund von Katastrophen ausfallsicher sind. Berücksichtigen Sie neben den Leistungsanforderungen Ihres Anwendungsszenarios auch die Kosten. Preisinformationen für Speicherklassen finden Sie unter Amazon S3 – Preise.

Einrichten der Speicherklasse eines Objekts

Sie können beim Hochladen eine Speicherklasse für ein Objekt angeben. Wenn Sie dies nicht tun, verwendet Amazon S3 die standardmäßige Amazon S3 S3-Standardspeicherklasse. Sie können auch die Speicherklasse eines Objekts, das bereits in Amazon S3 gespeichert ist, mithilfe der Amazon S3 S3-Konsole oder der AWS Command Line Interface (AWS CLI) in eine andere Speicherklasse ändern. AWS SDKs Alle diese Ansätze verwenden Amazon API S3-Operationen, um Anfragen an Amazon S3 zu senden.

Sie können Amazon S3 anweisen, die Speicherklasse von Objekten automatisch zu ändern, indem Sie einem Bucket eine S3-Lifecycle-Konfiguration hinzufügen. Weitere Informationen finden Sie unter Verwaltung des Lebenszyklus von Objekten.

Beim Einrichten einer S3-Replikationskonfiguration können Sie die Speicherklasse für replizierte Objekte auf eine beliebige andere Speicherklasse festlegen. Sie können jedoch nicht in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive gespeicherte Objekte replizieren. Weitere Informationen finden Sie unter Elemente der Replikationskonfigurationsdatei.

Wenn Sie die Speicherklasse programmgesteuert festlegen, geben Sie den Wert der Speicherklasse an. Im Folgenden finden Sie eine Liste der Konsolennamen für Speicherklassen mit ihren entsprechenden API Werten:

- S3-Standard — `STANDARD`
- S3-Standard—IA — `STANDARD_IA`
- S3 Eine Zone-IA — `ONEZONE_IA`
- S3 Express Eine Zone — `EXPRESS_ONEZONE`
- Intelligentes S3-Tiering — `INTELLIGENT_TIERING`

- Sofortiger Abruf von S3 Glacier — `GLACIER_IR`
- Flexibler Abruf von S3 Glacier — `GLACIER`
- S3 Glacier Deep Archive — `DEEP_ARCHIVE`
- REDUCED_REDUNDANCY – `REDUCED_REDUNDANCY`

Einstellung der Speicherklasse für ein neues Objekt

Um die Speicherklasse beim Hochladen eines Objekts festzulegen, können Sie die folgenden Methoden verwenden.

Verwenden der S3-Konsole

So legen Sie die Speicherklasse beim Hochladen eines neuen Objekts in der Konsole fest:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter: https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, in den Ihre Ordner oder Dateien hochgeladen werden sollen.
4. Klicken Sie auf Upload.
5. Wählen Sie im Upload-Fenster die Option Eigenschaften aus.
6. Wählen Sie unter Speicherklasse eine Speicherklasse für die Dateien aus, die Sie hochladen.
7. (Optional) Konfigurieren Sie zusätzliche Eigenschaften für die Dateien, die Sie hochladen. Weitere Informationen finden Sie unter Objekte hochladen
8. Führen Sie im Fenster Upload einen der folgenden Schritte aus:
   - Ziehen Sie Dateien und Ordner in das Upload-Fenster.
   - Wählen Sie Datei hinzufügen oder Ordner hinzufügen aus, wählen Sie die hochzuladenden Dateien oder Ordner und anschließend Öffnen aus.
9. Wählen Sie unten auf der Seite die Option Hochladen aus.

Verwenden Sie den REST API

Sie können die Speicherklasse für ein Objekt angeben, wenn Sie es mit den `CreateMultipartUpload` API Operationen`PutObject`, `POST Object` Object und add the

`x-amz-storage-class` request header erstellen. Wenn Sie diesen Header nicht hinzufügen, verwendet Amazon S3 die Standardspeicherklasse S3 Standard (`STANDARD`).

Diese Beispielanforderung verwendet den `PutObject` Befehl, um die Speicherklasse für ein neues Objekt auf S3 Intelligent-Tiering festzulegen:

```
PUT /my-image.jpg HTTP/1.1
Host: amzn-s3-demo-bucket1.s3.Region.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: image/jpeg
Content-Length: 11434
Expect: 100-continue
x-amz-storage-class: DEEP_ARCHIVE
```

Mit dem AWS CLI

In diesem Beispiel wird der `put-object` Befehl verwendet, um das hochzuladen *`my_images.tar.bz2`* to *`amzn-s3-demo-bucket1`* in der *`GLACIER`* Speicherklasse:

```
aws s3api put-object --bucket amzn-s3-demo-bucket1 --key dir-1/my_images.tar.bz2 --
storage-class GLACIER --body my_images.tar.bz2
```

Wenn die Objektgröße mehr als 5 GB beträgt, verwenden Sie den folgenden Befehl, um die Speicherklasse festzulegen:

```
aws s3 cp large_test_file s3://amzn-s3-demo-bucket1 --storage-class GLACIER
```

Ändern der Speicherklasse für ein vorhandenes Objekt

Um die Speicherklasse festzulegen, wenn Sie ein Objekt hochladen, können Sie die folgenden Methoden verwenden.

Verwenden der S3-Konsole

Sie können die Speicherklasse eines Objekts mithilfe der Amazon S3 S3-Konsole ändern, wenn die Objektgröße weniger als 5 GB beträgt. Bei größeren Objekten empfehlen wir, eine S3-Lebenszykluskonfiguration hinzuzufügen, um die Speicherklasse des Objekts zu ändern.

Um die Speicherklasse eines Objekts in der Konsole zu ändern:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets und dann den Tab Buckets für allgemeine Zwecke aus. Navigieren Sie zu dem Amazon S3 S3-Bucket oder -Ordner, der die Objekte enthält, die Sie ändern möchten.
3. Aktivieren Sie das Kontrollkästchen links neben den Namen der Objekte, die Sie ändern möchten.
4. Wählen Sie im Menü Aktionen aus der daraufhin angezeigten Optionsliste die Option Speicherklasse bearbeiten aus.
5. Wählen Sie eine der für Ihr Objekt verfügbaren Speicherklassen aus.
6. Wählen Sie unter Zusätzliche Kopiereinstellungen aus, ob Sie Quelleinstellungen kopieren, Keine Einstellungen angeben oder Einstellungen angeben möchten. Quelleinstellungen kopieren ist die Standardoption. Wenn Sie das Objekt nur ohne die Quelleinstellungsattribute kopieren möchten, wählen Sie „Keine Einstellungen angeben“. Wählen Sie „Einstellungen angeben“, um Einstellungen für SpeicherklasseACLs, Objekt-Tags, Metadaten, serverseitige Verschlüsselung und zusätzliche Prüfsummen anzugeben.
7. Wählen Sie in der unteren rechten Ecke die Option Änderungen speichern aus. Amazon S3 speichert Ihre Änderungen.

Mit dem REST API

Verwenden Sie die folgenden Methoden, um die Speicherklasse eines vorhandenen Objekts zu ändern.

In dieser Beispielanforderung wird der `PutObject` Befehl verwendet, um die Speicherklasse für ein neues Objekt auf S3 Intelligent-Tiering festzulegen:

```
PUT /my-image.jpg HTTP/1.1
```

```
Host: amzn-s3-demo-bucket1.s3.Region.amazonaws.com
Date: Wed, 12 Oct 2009 17:50:00 GMT
Authorization: authorization string
Content-Type: image/jpeg
Content-Length: 11434
Expect: 100-continue
x-amz-storage-class: DEEP_ARCHIVE
```

Mit dem AWS CLI

In diesem Beispiel wird der `put-object` Befehl verwendet, um das zu aktualisieren *my_images.tar.bz2* to *amzn-s3-demo-bucket1* in der *DEEP_ARCHIVE* Speicherklasse:

```
aws s3api put-object --bucket amzn-s3-demo-bucket1 --key dir-1/my_images.tar.bz2 --
storage-class DEEP_ARCHIVE --body my_images.tar.bz2
```

Wenn die Objektgröße mehr als 5 GB beträgt, verwenden Sie den folgenden Befehl, um die Speicherklasse zu ändern:

```
aws s3 cp object_S3_URI object_S3_URI --storage-class DEEP_ARCHIVE
```

Einschränken von Zugriffsrichtlinienberechtigungen auf eine bestimmte Speicherklasse

Wenn Sie Zugriffsrichtlinienberechtigungen für Amazon-S3-Vorgänge erteilen, können Sie mit dem Bedingungsschlüssel `s3:x-amz-storage-class` einschränken, welche Speicherklasse beim Speichern hochgeladener Objekte verwendet werden soll. Wenn Sie beispielsweise die Berechtigung `s3:PutObject` erteilen, können Sie das Hochladen von Objekten auf eine bestimmte Speicherklasse einschränken. Eine Beispielrichtlinie finden Sie unter Beispiel: Beschränkung von Objekt-Uploads auf Objekte mit einer bestimmten Speicherklasse.

Weitere Informationen zur Verwendung von Bedingungen in Richtlinien und die vollständige Liste der Bedingungsschlüssel in Amazon S3 finden Sie in den folgenden Themen:

- Aktionen, Ressourcen und Bedingungsschlüssel für Amazon S3 in der Service Authorization Reference

Weitere Informationen zu den Berechtigungen für API S3-Operationen nach S3-Ressourcentypen finden Sie unterErforderliche Berechtigungen für Amazon S3 API S3-Operationen.

- Beispiele für Bucket-Richtlinien mit Bedingungsschlüsseln

Amazon S3 analytics – Speicherklassen-Analyse

Mit der Speicherklassen-Analyse von Amazon S3 Analytics können Sie Speicherzugriffsmuster analysieren, anhand derer Sie entscheiden können, ob Sie die richtigen Daten in die richtige Speicherklasse einordnen. Diese neue Amazon S3 S3-Analysefunktion beobachtet Datenzugriffsmuster, sodass Sie bestimmen können, wann Speicher mit seltenem Zugriff auf die STANDARD Speicherklasse STANDARD _IA (IA, für seltenen Zugriff) umgestellt werden sollte. Weitere Informationen über Speicherklassen finden Sie unter Amazon S3 S3-Speicherklassen verstehen und verwalten.

Wenn die Speicherklassen-Analyse die seltenen Zugriffsmuster für eine gefilterte Datenmenge im Laufe der Zeit beobachtet, können Sie Ihre Lebenszykluskonfigurationen unter Verwendung der Analyseergebnisse verbessern. Sie können die Speicherklassen-Analyse so konfigurieren, dass alle Objekte in einem Bucket analysiert werden. Sie können aber auch Filter konfigurieren, um Objekte nach einem gemeinsamen Präfix zu gruppieren (d. h. für Objekte, deren Namen mit einer gemeinsamen Zeichenkette beginnen), nach Objekt-Markierungen oder sowohl nach Präfix und Markierungen. Wahrscheinlich werden Sie feststellen, dass die Filterung nach Objektgruppen die beste Methode darstellt, von der Speicherklassen-Analyse zu profitieren.

> **Important**
>
> Die Speicherklassenanalyse stellt nur Empfehlungen für die Klassen Standard bis Standard-IA bereit.

Sie können mehrere Speicherklassen-Analysefilter pro Bucket einrichten, bis zu 1000, und erhalten für jeden Filter eine separate Analyse. Mithilfe mehrerer Filterkonfigurationen können Sie bestimmte Objektgruppen analysieren, um Ihre Lebenszykluskonfigurationen zu verbessern, bei denen Objekte auf _IA umgestellt werden. STANDARD

Die Speicherklassen-Analyse bietet Schaubilder zur Speichernutzung in der Amazon-S3-Konsole, die täglich aktualisiert werden. Sie können diese täglichen Nutzungsdaten auch in einen S3-Bucket

exportieren und sie in einer Tabellenkalkulationsanwendung oder mit Business Intelligence-Tools wie Amazon QuickSight anzeigen.

Mit der Speicherklassen-Analyse sind Kosten verbunden. Preisinformationen finden Sie unter Verwaltung und Replikation Amazon-S3-Preise.

Themen

- Wie richte ich die Speicherklassen-Analyse ein?
- Wie verwende ich die Speicherklassen-Analyse?
- Wie kann ich die Daten der Speicherklassen-Analyse exportieren?
- Konfigurieren der Speicherklassen-Analyse

Wie richte ich die Speicherklassen-Analyse ein?

Sie können die Speicherklassen-Analyse Einrichten, indem Sie konfigurieren, welche Objektdaten analysiert werden sollen. Sie können die Speicherklassenanalsyse für die folgenden Aufgaben konfigurieren:

- Analyse des gesamten Inhalts eines Buckets.

  Sie erhalten eine Analyse für alle Objekte im Bucket.
- Analyse von nach Präfix und Tags gruppierten Objekten.

  Sie können Filter konfigurieren, um Objekte nach einem gemeinsamen Präfix oder nach Objekt-Markierungen zu gruppieren, oder nach einer Kombination aus Präfix und Markierungen. Sie erhalten für jeden konfigurierten Filter eine separate Analyse. Sie können maximal 1000 Filterkonfigurationen pro Bucket verwenden.
- Export von Analysedaten.

  Wenn Sie die Speicherklassen-Analyse für einen Bucket oder Filter konfigurieren, können Sie die Analysedaten täglich in eine Datei exportieren lassen. Die Analyse für den Tag wird der Datei hinzugefügt, um ein Verlaufsanalyseprotokoll für den konfigurierten Filter zu erstellen. Die Datei wird täglich an dem Ziel Ihrer Wahl aktualisiert. Wenn Sie Dateien zum Export auswählen, geben Sie einen Ziel-Bucket und ein optionales Ziel-Präfix an, wohin die Datei geschrieben werden soll.

Sie können die Amazon S3 S3-Konsole, die oder die oder verwenden RESTAPI, AWS SDKs um die AWS CLI Speicherklassenanalyse zu konfigurieren.

- Informationen zum Konfigurieren der Analyse von Speicherklassen in der Amazon-S3-Konsole finden Sie unter Konfigurieren der Speicherklassen-Analyse.
- Um Amazon S3 zu verwendenAPI, verwenden Sie das PutBucketAnalyticsConfigurationRESTAPIoder das Äquivalent aus dem AWS CLI oder AWS SDKs.

Wie verwende ich die Speicherklassen-Analyse?

Mithilfe der Speicherklassenanalyse beobachten Sie Ihre Datenzugriffsmuster im Laufe der Zeit, um Informationen zu sammeln, die Ihnen helfen, das Lebenszyklusmanagement Ihres STANDARD _IA-Speichers zu verbessern. Nachdem Sie einen Filter konfiguriert haben, sehen Sie die Datenanalyse basierend auf dem Filter in der Amazon-S3-Konsole über 24 bis 48 Stunden. Die Speicherklassen-Analyse beobachtet jedoch die Zugriffsmuster einer gefilterten Datenmenge für 30 Tage oder Länger, um Informationen für eine Analyse zu sammeln, bevor ein Ergebnis zurückgegeben wird. Die Analyse wird nach dem ersten Ergebnis fortgesetzt und aktualisiert das Ergebnis, wenn sich die Zugriffsmuster ändern.

Wenn Sie einen Filter zum ersten Mal konfigurieren, dauert es möglicherweise einen Moment, um die Daten der Amazon-S3-Konsole zu analysieren.

Die Speicherklassen-Analyse beobachtet die Zugriffsmuster einer gefilterten Objekt-Datenmenge für 30 Tage oder Länger, um ausreichend viele Informationen für eine Analyse zu sammeln. Nachdem die Speicherklassen-Analyse genügend Informationen gesammelt hat, wird in der Amazon-S3-Konsole eine Meldung angezeigt, dass die Analyse abgeschlossen ist.

Bei der Durchführung der Analyse für Objekte mit seltenem Zugriff betrachtet die Speicherklassen-Analyse die gefilterte gruppierte Objektmenge basierend auf der Dauer für die sie in Amazon S3 hochgeladen waren. Die Speicherklassen-Analyse stellt fest, ob für die Altersgruppe selten ein Zugriff stattfindet, indem sie die folgenden Faktoren für die gefilterte Datenmenge betrachtet.

- Objekte in der STANDARD Speicherklasse, die größer als 128 KB sind.
- Wie viel durchschnittlichen Gesamtspeicher Sie pro Altersgruppe haben.
- Die durchschnittliche Anzahl an Bytes, die pro Altersgruppe nach außen übertragen werden (nicht die Frequenz).
- Analyseexportdaten enthalten nur Anfragen mit Daten, die für die Speicherklassen-Analyse relevant sind. Dies kann Differenzen in der Anzahl der Anfragen sowie im Hinblick auf die

gesamten Bytes für Upload und Anfrage im Vergleich zu den Angaben in der Speichermetrik oder in den Ergebnissen Ihrer eigenen internen Systeme verursachen.

- GETFehlgeschlagen und PUT Anfragen werden für die Analyse nicht gezählt. In den Speichermetriken sehen Sie jedoch fehlgeschlagene Anfragen.

Wie viel von meinem Speicher habe ich abgerufen?

Die Amazon-S3-Konsole zeigt in einer Grafik an, wie viel von dem Speicher in der gefilterten Datenmenge für den Beobachtungszeitraum abgerufen wurde.

Welchen Prozentsatz meines Speichers habe ich abgerufen?

Die Amazon-S3-Konsole zeigt in einer Grafik an, welcher Prozentsatz des Speichers in der gefilterten Datenmenge für den Beobachtungszeitraum abgerufen wurde.

Wie in diesem Thema bereits dargelegt, betrachtet die Speicherklassen-Analyse bei der Analyse für Objekte mit seltenem Zugriff die gefilterte Objektemenge, gruppiert nach dem Zeitpunkt, zu dem die Objekte in Amazon S3 hochgeladen wurden. Die Speicherklassen-Analyse verwendet die folgenden vordefinierten Objektaltersgruppen:

- Amazon-S3-Objekte, die weniger als 15 Tage alt sind
- Amazon S3 Objekte 15-29 Tage alt
- Amazon S3 Objekte 30-44 Tage alt
- Amazon S3 Objekte 45-59 Tage alt
- Amazon S3 Objekte 60-74 Tage alt
- Amazon S3 Objekte 75-89 Tage alt
- Amazon S3 Objekte 90-119 Tage alt
- Amazon S3 Objekte 120-149 Tage alt
- Amazon S3 Objekte 150-179 Tage alt
- Amazon S3 Objekte 180-364 Tage alt
- Amazon S3 Objekte 365-729 Tage alt
- Amazon-S3-Objekte, 730 Tage alt und älter

In der Regel benötigt sie über 30 Tage beobachtete Zugriffsmuster, um ausreichend viele Informationen für ein Analyseergebnis zu sammeln. Abhängig von dem spezifischen Zugriffsmuster Ihrer Daten könnte dies länger als 30 Tage dauern. Nachdem Sie jedoch einen Filter konfiguriert

haben, sehen Sie die Datenanalyse basierend auf dem Filter in der Amazon-S3-Konsole über 24 bis 48 Stunden. Sie sehen die Analyse zum Objektzugriff auf täglicher Basis, unterteilt nach Objektaltersgruppe in der Amazon-S3-Konsole.

Wie viel von meinem Speicher wird selten abgerufen?

Die Amazon-S3-Konsole zeigt die Zugriffsmuster, die nach den vordefinierten Altersgruppen des Objekts gruppiert sind. Der angezeigte Text Frequently accessed (Häufig aufgerufen) oder Infrequently accessed (Selten aufgerufen) ist als visuelle Hilfe bei der Erstellung des Lebenszyklus gedacht.

Wie kann ich die Daten der Speicherklassen-Analyse exportieren?

Sie können festlegen, dass Storage Class Analysis Analyseberichte in eine Flatfile mit kommagetrennten Werten (CSV) exportiert. Berichte werden täglich aktualisiert und basieren auf den von Ihnen konfigurierten Objektaltersgruppenfiltern. Bei Verwendung der Amazon-S3-Konsole können Sie die Exportberichtsoption wählen, wenn Sie einen Filter erstellen. Wenn Sie Dateien zum Export auswählen, geben Sie einen Ziel-Bucket und ein optionales Ziel-Präfix an, wohin die Datei geschrieben werden soll. Sie können die Daten in einen Ziel-Bucket in einem anderen Konto exportieren. Der Ziel-Bucket muss sich in derselben Region befinden wie der Bucket, dessen Analyse Sie konfiguriert haben.

Sie müssen eine Bucket-Richtlinie für den Ziel-Bucket erstellen, um Amazon S3 Berechtigungen zu erteilen, um zu überprüfen, AWS-Konto wem der Bucket gehört, und um Objekte in den Bucket am definierten Speicherort zu schreiben. Eine Beispielrichtlinie finden Sie unter Gewähren von Berechtigungen für S3 Inventory und S3 Analytics.

Nachdem Sie Speicherklassen-Analyseberichte konfiguriert haben, erhalten Sie den exportierten Bericht nach 24 Stunden täglich. Anschließend setzt Amazon S3 die Überwachung fort und stellt tägliche Berichte bereit.

Sie können die CSV Datei in einer Tabellenkalkulationsanwendung öffnen oder die Datei in andere Anwendungen wie Amazon QuickSight importieren. Informationen zur Verwendung von Amazon S3 S3-Dateien mit Amazon QuickSight finden Sie unter Erstellen eines Datensatzes mithilfe von Amazon S3 S3-Dateien im QuickSight Amazon-Benutzerhandbuch.

Daten in den exportierten Dateien werden nach dem Datum innerhalb der Objektaltersgruppe sortiert, wie in den folgenden Beispielen gezeigt. Wenn es sich bei der Speicherklasse um STANDARD die Zeile handelt, enthält sie auch Daten für die Spalten `ObjectAgeForSIATransition` und`RecommendedObjectAgeForSIATransition`.

Date	ConfigId	Filter	StorageClass	ObjectAge	ObjectCount	DataUploaded_MB	Storage_MB	DataRetrieved_MB	GetRequestCount	CumulativeAccessRatio	ObjectAgeForSIATransition	RecommendedObjectAgeForSIATransition
8/17/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
9/2/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/22/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
8/25/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
9/6/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/30/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/28/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		
8/21/2021	SalesMaterial	SalesMaterial	STANDARD	000-014			0.4313			0		
9/5/2021	SalesMaterial	SalesMaterial	STANDARD	000-014						0.04096734		

Am Ende des Berichts wird die Altersgruppe des Objekts als angegebenALL. Die ALL Zeilen enthalten kumulierte Gesamtwerte, einschließlich Objekten, die kleiner als 128 KB sind, für alle Altersgruppen für diesen Tag.

8/24/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014
9/3/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.02426125	015-029
8/28/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.03545875	015-029
8/17/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014
8/25/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014
9/6/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.0209529	015-029
9/4/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.02304819	015-029
8/22/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014
8/21/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014
8/30/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0.03073092	015-029
8/20/2021	SalesMaterial	SalesMaterial	STANDARD	ALL	3		0.4599			0	000-014

Der nächste Abschnitt beschreibt die im Bericht verwendeten Spalten.

Layout der exportierten Datei

In der folgenden Tabelle wird das Layout der Exportdatei für die Amazon S3 S3-Speicherklassenanalyse beschrieben.

Konfigurieren der Speicherklassen-Analyse

Unter Verwendung des Tools für die Speicherklassen-Analyse von Amazon S3 können Sie Speicherzugriffsmuster analysieren, anhand derer Sie entscheiden können, ob Sie die richtigen Daten in die richtige Speicherklasse einordnen. Bei der Speicherklassenanalyse werden Datenzugriffsmuster beobachtet, sodass Sie ermitteln können, wann Speicher mit seltenem Zugriff auf die STANDARD Speicherklasse STANDARD _IA (IA, für seltenen Zugriff) umgestellt werden sollte. Weitere Informationen zu STANDARD _IA finden Sie unter Amazon S3 FAQ undAmazon S3 S3-Speicherklassen verstehen und verwalten.

Sie können die Speicherklassen-Analyse Einrichten, indem Sie konfigurieren, welche Objektdaten analysiert werden sollen. Sie können die Speicherklassenanalsyse für die folgenden Aufgaben konfigurieren:

- Analyse des gesamten Inhalts eines Buckets.

  Sie erhalten eine Analyse für alle Objekte im Bucket.
- Analyse von nach Präfix und Tags gruppierten Objekten.

Sie können Filter konfigurieren, um Objekte nach einem gemeinsamen Präfix oder nach Objekt-Markierungen zu gruppieren, oder nach einer Kombination aus Präfix und Markierungen. Sie erhalten für jeden konfigurierten Filter eine separate Analyse. Sie können maximal 1000 Filterkonfigurationen pro Bucket verwenden.

- Export von Analysedaten.

  Wenn Sie die Speicherklassen-Analyse für einen Bucket oder Filter konfigurieren, können Sie die Analysedaten täglich in eine Datei exportieren lassen. Die Analyse für den Tag wird der Datei hinzugefügt, um ein Verlaufsanalyseprotokoll für den konfigurierten Filter zu erstellen. Die Datei wird täglich an dem Ziel Ihrer Wahl aktualisiert. Wenn Sie Dateien zum Export auswählen, geben Sie einen Ziel-Bucket und ein optionales Ziel-Präfix an, wohin die Datei geschrieben werden soll.

Sie können die Amazon S3 S3-Konsole, die oder die oder verwenden RESTAPI, AWS SDKs um die AWS CLI Speicherklassenanalyse zu konfigurieren.

> **Important**
>
> Die Speicherklassenanalyse gibt keine Empfehlungen für den Übergang zu den Speicherklassen ONEZONE _IA oder S3 Glacier Flexible Retrieval.
> Wenn Sie die Speicherklassenanalyse so konfigurieren möchten, dass Ihre Ergebnisse als .csv-Datei exportiert werden und der Ziel-Bucket die Standard-Bucket-Verschlüsselung mit a verwendet AWS KMS key, müssen Sie die AWS KMS Schlüsselrichtlinie aktualisieren, um Amazon S3 die Berechtigung zum Verschlüsseln der .csv-Datei zu erteilen. Detaillierte Anweisungen finden Sie unter Erteilen der Berechtigung an Amazon S3 zur Verwendung Ihres vom Kunden verwalteten Schlüssels für die Verschlüsselung.

Weitere Informationen zu Analytics finden Sie unter Amazon S3 analytics – Speicherklassen-Analyse.

Verwenden der S3-Konsole

Konfigurieren der Speicherklassen-Analyse

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie die Speicherklassen-Analyse konfigurieren wollen.

3. Wählen Sie den Tab Metrics.
4. Wählen Sie unter Storage Class Analysis (Speicherklassen-Analyse) die Option Create analytics configuration (Analysenkonfiguration erstellen).
5. Geben Sie einen Namen für den Filter ein. Wenn der komplette Bucket analysiert werden soll, tragen Sie nichts in das Feld Prefix (Präfix) ein.
6. Geben Sie im Feld Prefix (Präfix) Text für das Präfix für die Objekte ein, die Sie analysieren möchten.
7. Um einen Tag hinzuzufügen, wählen Sie Add tag (Tag hinzufügen). Geben Sie einen Schlüssel und Wert für den Tag ein. Sie können ein Präfix und mehrere Markierungen eingeben.
8. Optional können Sie unter Exportieren die Option Aktivieren wählen, CSV um Analyseberichte in eine Flatfile mit kommagetrennten Werten (.csv) zu exportieren. Wählen Sie einen Ziel-Bucket aus, in dem die Datei gespeichert werden kann. Sie können ein Präfix für den Ziel-Bucket eingeben. Der Ziel-Bucket muss sich in demselben Bucket befinden AWS-Region wie der Bucket, für den Sie die Analyse einrichten. Der Ziel-Bucket kann sich in einem anderem AWS-Konto befinden.

   Wenn der Ziel-Bucket für die .csv-Datei die Standard-Bucket-Verschlüsselung mit einem KMS Schlüssel verwendet, müssen Sie die AWS KMS Schlüsselrichtlinie aktualisieren, um Amazon S3 die Erlaubnis zu erteilen, die .csv-Datei zu verschlüsseln. Detaillierte Anweisungen finden Sie unter Erteilen der Berechtigung an Amazon S3 zur Verwendung Ihres vom Kunden verwalteten Schlüssels für die Verschlüsselung.
9. Wählen Sie Create configuration (Konfiguration erstellen).

Amazon S3 erstellt eine Bucket-Richtlinie für den Ziel-Bucket, mit der Amazon S3 Schreibberechtigung gewährt wird. Dadurch können die Exportdaten in den Bucket geschrieben werden.

Wenn beim Erstellen der Bucket-Richtlinie ein Fehjler auftritt, erhalten Sie Anweisungen zur Fehlerbehebung. Wenn Sie beispielsweise einen Ziel-Bucket in einem anderen AWS-Konto ausgewählt haben und nicht über die Berechtigungen verfügen, die Bucket-Richtlinie zu lesen und zu verändern, wird die folgende Meldung angezeigt. Sie müssen den Eigentümer des Ziel-Buckets auffordern, dem Ziel-Bucket die angezeigte Bucket-Richtlinie hinzuzufügen. Wird die Richtlinie dem Ziel-Bucket nicht hinzugefügt, erhalten Sie keine Exportdaten, weil Amazon S3 nicht die Berechtigung besitzt, in den Ziel-Bucket zu schreiben. Wenn der Quell-Bucket nicht dem Konto des aktuellen Benutzers gehört, muss die richtige Konto-ID des Quell-Buckets in der Richtlinie ersetzt werden.

Informationen über die exportierten Daten und die Funktionsweise des Filters finden Sie unter Amazon S3 analytics – Speicherklassen-Analyse.

Mit dem REST API

Um Storage Class Analysis mit dem zu konfigurieren RESTAPI, verwenden Sie den PutBucketAnalyticsConfiguration. Sie können die entsprechende Operation auch mit dem AWS CLI oder verwenden AWS SDKs.

Sie können Folgendes verwenden RESTAPIs, um mit Storage Class Analysis zu arbeiten:

- DELETEBucket Analytics-Konfiguration
- GETBucket Analytics-Konfiguration
- List Bucket Analytics Configuration

Verwaltung der Speicherkosten mit Amazon S3 Intelligent-Tiering

Die Speicherklasse S3 Intelligent-Tiering wurde entwickelt, um die Speicherkosten zu optimieren, indem Daten automatisch auf die kostengünstigste Zugriffsebene verschoben werden, wenn sich die Zugriffsmuster ändern, ohne dass Betriebsaufwand oder Leistungseinbußen auftreten. Gegen eine geringe monatliche Gebühr für Objektüberwachung und Automatisierung überwacht S3 Intelligent-Tiering die Zugriffsmuster und verschiebt die Objekte, auf die nicht zugegriffen wurde, automatisch in kostengünstigere Zugriffsebenen.

S3 Intelligent-Tiering ermöglicht automatische Speicherkosteneinsparungen in drei Zugriffsebenen mit niedriger Latenz und hohem Durchsatz. Für Daten, auf die asynchron zugegriffen werden kann, können Sie innerhalb der S3 Intelligent-Tiering-Speicherklasse die automatische Archivfunktionen aktivieren. Bei S3 Intelligent-Tiering werden keine Abrufgebühren erhoben. Wenn später auf ein Objekt in der Stufe Infrequent Access oder Archive Instant Access zugegriffen wird, wird es automatisch zurück in die Stufe Frequent Access verschoben. Es werden keine Zusatzgebühren erhoben, wenn Objekte innerhalb der S3-Intelligent-Tiering-Speicherklasse zwischen Zugriffsstufen verschoben werden.

S3 Intelligent-Tiering ist die empfohlene Speicherklasse für Daten mit unbekannten, sich ändernden oder unvorhersehbaren Zugriffsmustern, unabhängig von der Objektgröße oder dem Aufbewahrungszeitraum, wie Data Lakes, Datenanalytik und neue Anwendungen.

Die Speicherklasse S3 Intelligent-Tiering unterstützt alle Amazon-S3-Funktionen, einschließlich der folgenden:

- S3 Inventory zur Überprüfung der Zugriffsebene von Objekten
- S3-Replikation, für die Replikation von Daten auf beliebige AWS-Region
- S3 Storage Lens zum Anzeigen von Speichernutzungs- und Aktivitätsmetriken
- Serverseitige Verschlüsselung, für den Schutz von Objektdaten
- S3-Objektsperre, um versehentliches Löschen von Daten zu verhindern
- AWS PrivateLink, für den Zugriff auf Amazon S3 über einen privaten Endpunkt in einer virtuellen privaten Cloud (VPC)

Weitere Informationen zur Verwendung von S3 Intelligent-Tiering finden Sie in den folgenden Abschnitten:

Themen

- So funktioniert S3 Intelligent-Tiering
- Verwenden von S3 Intelligent-Tiering
- Verwenden von S3 Intelligent-Tiering

So funktioniert S3 Intelligent-Tiering

Die Speicherklasse Amazon S3 Intelligent-Tiering speichert Objekte automatisch in drei Zugriffsebenen. Eine Stufe ist für häufigen Zugriff optimiert, eine kostengünstigere Stufe ist für seltenen Zugriff optimiert und eine andere sehr kostengünstige Stufe ist für selten zugängliche Daten optimiert. Gegen eine geringe monatliche Gebühr für Objektüberwachung und Automatisierung überwacht S3 Intelligent-Tiering die Zugriffsmuster und verschiebt Objekte automatisch in die Ebene für seltenen Zugriff, wenn auf sie 30 aufeinanderfolgende Tage lang nicht zugegriffen wurde. Nach 90 Tagen ohne Zugriff werden die Objekte ohne Leistungseinbußen oder Betriebsaufwand auf die Ebene des Archivs Instant Access verschoben.

Aktivieren Sie Archivierungsfunktionen, um zwei zusätzliche Zugriffsebenen hinzuzufügen und die niedrigsten Speicherkosten für Daten zu erhalten, auf die in Minuten bis Stunden zugegriffen werden kann. Sie können Objekte schichtweise auf die Archive Access Tier oder Deep Archive Access Tier oder beiden reduzieren. Mit Archive Access verschiebt S3 Intelligent-Tiering Objekte, auf die mindestens 90 aufeinanderfolgende Tage lang nicht zugegriffen wurde, in die Ebene „Archive Access“. Mit Deep Archive Access verschiebt S3 Intelligent-Tiering Objekte, auf die mindestens 180 aufeinanderfolgende Tage lang nicht zugegriffen wurde, in die Ebene „Deep Archive Access“. Für beide Ebenen können Sie die Anzahl der Tage ohne Zugriff je nach Bedarf konfigurieren.

Die folgenden Aktionen stellen einen Zugriff dar, der ein Herabstufen Ihrer Objekte in die Ebene „Archive Access“ oder „Deep Archive Access“ verhindert:

- Herunterladen oder Kopieren eines Objekts über die Amazon S3-Konsole.
- Aufrufen CopyObject, UploadPartCopy, oder das Replizieren von Objekten mit S3 Batch Replication. In diesen Fällen werden die Quellobjekte der Kopier- oder Replikationsvorgänge hochgestuft.
- Aufrufen GetObject, PutObject, RestoreObject, CompleteMultipartUpload, ListParts, oder SelectObjectContent.

Wenn beispielsweise vor Ablauf der von Ihnen festgelegte Anzahl von Tagen ohne Zugriff (z. B. 180 Tage) über `SelectObjectContent` ein Zugriff auf Ihre Objekte erfolgt, wird durch diese Aktion der Timer zurückgesetzt. Ihre Objekte werden erst in die Ebene „Archive Access“ oder „Deep Archive Access“ verschoben, wenn die Zeit nach der letzten `SelectObjectContent`-Anforderung die von Ihnen festgelegte Anzahl von Tagen erreicht hat.

Wenn später auf ein Objekt in der Stufe Infrequent Access oder Archive Instant Access zugegriffen wird, wird es automatisch zurück in die Stufe Frequent Access verschoben.

Die folgenden Aktionen stellen einen Zugriff dar, bei dem Objekte automatisch von der Ebene „Infrequent Access“ oder „Archive Instant Access“ zurück in die Ebene „Frequent Access“ verschoben werden:

- Herunterladen oder Kopieren eines Objekts über die Amazon S3-Konsole.
- Aufrufen CopyObject, UploadPartCopy, oder das Replizieren von Objekten mit Batch Replication. In diesen Fällen werden die Quellobjekte der Kopier- oder Replikationsvorgänge hochgestuft.
- Aufrufen GetObject, PutObject, RestoreObject, CompleteMultipartUpload, oder ListParts.

Andere Aktionen stellen keinen Zugriff dar, bei dem Objekte automatisch von der Ebene „Infrequent Access“ oder „Archive Instant Access“ zurück in die Ebene „Frequent Access“ verschoben werden. Im Folgenden finden Sie ein Beispiel, keine endgültige Liste, solcher Aktionen:

- Aufrufen HeadObject, GetObjectTagging, PutObjectTagging, ListObjects, ListObjectsV2, oder ListObjectVersions.
- Aufrufen SelectObjectContentstellt keinen Zugriff dar, der Objekte einer Stufe für häufigen Zugriff zuordnet. Darüber hinaus wird dadurch nicht verhindert, dass Objekte von der Stufe „Frequent

Access“ auf die Stufe „Infrequent Access“ und von dort aus auf die Stufe „Archive Instant Access“ herabgestuft werden.

Sie können S3 Intelligent-Tiering als Standardspeicherklasse für neu erstellte Daten verwenden, indem Sie dies `INTELLIGENT-TIERING` im `x-amz-storage-class`Anforderungsheader angeben, wenn Sie die Operationen `PutObject`, `CopyObject`, oder aufrufen. `CreateMultipartUpload` S3 Intelligent-Tiering ist auf 99,9 % Verfügbarkeit und 99,9999999 % Haltbarkeit ausgelegt.

> **Note**
>
> Wenn die Größe eines Objekts weniger als 128 KB beträgt, wird es nicht überwacht und ist nicht für das automatische Tiering geeignet. Kleinere Objekte werden immer in der Stufe für häufige Zugriffe gespeichert.

S3-Intelligent-Tiering-Zugriffsebenen

Im folgenden Abschnitt werden die verschiedenen automatischen und optionalen Zugriffsebenen erläutert. Wenn Objekte zwischen Zugriffsebenen verschoben werden, bleibt die Speicherklasse gleich (S3 Intelligent-Tiering).

Stufe für häufige Zugriffe (automatisch)

Hierbei handelt es sich um die Standardzugriffsebene, in der Objekte, die in S3 Intelligent-Tiering erstellt oder überführt sind, seinen Lebenszyklus beginnt. Ein Objekt verbleibt in dieser Ebene, solange darauf zugegriffen wird. Die Stufe für häufige Zugriffe bietet niedrige Latenz und hohe Durchsatzleistung.

Stufe für seltene Zugriffe (automatisch)

Wenn 30 aufeinanderfolgende Tage auf ein Objekt nicht zugegriffen wird, wird das Objekt in die Stufe für seltene Zugriffe verschoben. Die Stufe für seltene Zugriffe bietet niedrige Latenz und hohe Durchsatzleistung.

Stife Archive Instant Access (automatisch)

Wenn 90 aufeinanderfolgende Tage auf ein Objekt nicht zugegriffen wird, wird das Objekt in die Stufe Archive Instant Access verschoben. Die Stufe Archive Instant Access bietet niedrige Latenzzeiten und hohe Durchsatzleistung.

Archivzugriffsebene (optional)

S3 Intelligent-Tiering bietet Ihnen die Möglichkeit, die Archivzugriffsebene für Daten zu aktivieren, auf die asynchron zugegriffen werden kann. Nach der Aktivierung archiviert die Ebene Archive Access automatisch Objekte, auf die mindestens 90 aufeinanderfolgende Tage lang nicht zugegriffen wurde. Sie können die letzte Zugriffszeit für die Archivierung auf maximal 730 Tage verlängern. Die Archivzugriffsebene hat dieselbe Leistung wie die Speicherklasse S3 Glacier Flexible Retrieval.

Die Standardabrufzeiten für diese Zugriffebene können zwischen 3 und 5 Stunden liegen. Wenn Sie Ihre Wiederherstellungsanfrage mithilfe von S3 Batch Operations starten, beginnt Ihre Wiederherstellung innerhalb von Minuten. Weitere Informationen zu den Abrufoptionen finden -Zeiten finden Sie unter the section called “Wiederherstellen von Objekten aus den Stufen S3 Intelligent-Tiering Archive Access und Deep Archive Access”.

> **Note**
>
> Aktivieren Sie die Stufe Archive Access nur für 90 Tage, wenn Sie die Stufe Instant Access Archiv umgehen möchten. Die Stufe Archive Access bietet etwas geringere Speicherkosten bei minute-to-hour gleichzeitiger Abrufzeit. Die Stufe Archive Instant Access bietet Millisekunden-Zugriff und hohe Durchsatzleistung.

Zugriffsebene für Deep Archive (optional)

S3 Intelligent-Tiering bietet Ihnen die Möglichkeit, die Stufe Deep Archive Access für Daten zu aktivieren, auf die asynchron zugegriffen werden kann. Nach der Aktivierung archiviert die Ebene Deep Archive Access automatisch Objekte, auf die mindestens 180 aufeinanderfolgende Tage lang nicht zugegriffen wurde. Sie können die letzte Zugriffszeit für die Archivierung auf maximal 730 Tage verlängern. Die Deep-Archive-Zugriffsebene hat dieselbe Leistung wie die S3 Glacier Deep Archive-Speicherklasse.

Der Standardabruf von Objekten in dieser Zugriffsebene erfolgt innerhalb von 12 Stunden. Wenn Sie Ihre Wiederherstellungsanfrage mithilfe von S3 Batch Operations starten, beginnt Ihre Wiederherstellung innerhalb von 9 Stunden. Weitere Informationen zu den Abrufoptionen finden -Zeiten finden Sie unter the section called “Wiederherstellen von Objekten aus den Stufen S3 Intelligent-Tiering Archive Access und Deep Archive Access”.

> **Note**
>
> Aktivieren Sie die Ebenen Archivzugriff und Deep Archive Access nur, wenn Ihre Anwendung asynchron auf Ihre Objekte zugreifen kann. Wenn das Objekt, das Sie abrufen, in den Ebenen „Archive Access“ oder „Deep Archive Access“ gespeichert ist, müssen Sie das Objekt zunächst mithilfe des `RestoreObject`-Vorgangs wieder herstellen.

Verwenden von S3 Intelligent-Tiering

Sie können die Speicherklasse S3 Intelligent-Tiering verwenden, um die Speicherkosten automatisch zu optimieren. S3 Intelligent-Tiering ermöglicht automatische Kosteneinsparungen durch Verschieben von Daten auf granularer Objektebene zwischen den Zugriffsebenen, wenn sich die Zugriffsmuster ändern. Für Daten, auf die asynchron zugegriffen werden kann, können Sie wählen, ob Sie die automatische Archivierung innerhalb der Speicherklasse S3 Intelligent-Tiering mithilfe von AWS Management Console AWS CLI, oder Amazon S3 aktivieren möchten. API

Verschieben von Daten nach S3 Intelligent-Tiering

Es gibt zwei Möglichkeiten, Daten in S3 Intelligent-Tiering zu übertragen. Sie können Objekte direkt von der Konsole aus oder programmgesteuert mithilfe eines Vorgangs in S3 Intelligent-Tiering hochladen. PUT Weitere Informationen finden Sie unter Einrichten der Speicherklasse eines Objekts. Sie können S3-Lifecycle-Konfigurationen auch so konfigurieren, dass Objekte von S3 Standard oder S3 Standard-Infrequent Access auf S3 Intelligent-Tiering umgestellt werden.

Daten mithilfe von Direct auf S3 Intelligent-Tiering hochladen PUT

Wenn Sie mithilfe der PUTAPIOperation ein Objekt in die S3 Intelligent-Tiering-Speicherklasse hochladen, geben Sie S3 Intelligent-Tiering im Anforderungsheader an. `x-amz-storage-class`

Die folgende Anforderung speichert das Image, `my-image.jpg`, im `myBucket`-Bucket. Die Anforderung verwendet den `x-amz-storage-class`-Header, um anzufordern, dass das Objekt mit der Speicherklasse S3 Intelligent-Tiering gespeichert ist.

Example

```
PUT /my-image.jpg HTTP/1.1
Host: myBucket.s3.<Region>.amazonaws.com (http://amazonaws.com/)
Date: Wed, 1 Sep 2021 17:50:00 GMT
```

```
Authorization: authorization string
Content-Type: image/jpeg
Content-Length: 11434
Expect: 100-continue
x-amz-storage-class: INTELLIGENT_TIERING
```

Übergang von Daten zu S3 Intelligent-Tiering von S3 Standard oder S3 Standard-Infrequent Access mit S3-Lebenszyklus

Sie können Regeln zu Ihrer S3-Lebenszykluskonfiguration hinzufügen und damit Amazon S3 anweisen, Objekte von einer Speicherklasse in eine andere zu wechseln. Informationen zu unterstützten Übergängen und zugehörigen Einschränkungen finden Sie unter Migrieren von Objekten mit S3-Lebenszyklus.

Sie können S3-Lebenszykluskonfigurationen auf der Bucket- oder Präfixebene angeben. In dieser S3-Lebenszyklus-Konfigurationregel spezifiziert der Filter ein Schlüsselpräfix (`documents/`). Aus diesem Grund gilt die Regel für Objekte mit dem Schlüsselnamenpräfix `documents/`, wie beispielsweise `documents/doc1.txt` und `documents/doc2.txt`. Die Regel gibt eine `Transition`-Aktion an, die Amazon S3 anweist, Objekte 0 Tage nach der Erstellung in die Speicherklasse S3 Intelligent-Tiering zu übertragen. In diesem Fall können Objekte um Mitternacht nach ihrer Erstellung auf S3 Intelligent-Tiering umgestellt werden. UTC

Example

```
<LifecycleConfiguration>
  <Rule>
    <ID>ExampleRule</ID>
    <Filter>
       <Prefix>documents/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>INTELLIGENT_TIERING</StorageClass>
    </Transition>
 </Rule>
</LifecycleConfiguration>
```

Ein versionsfähiger Bucket behält eine aktuelle Objektversion sowie und null oder mehr langfristige Objektversionen bei. Sie können separate Lebenszyklusrichtlinie für aktuelle und nicht aktuelle Objektversionen definieren.

Weitere Informationen finden Sie unter Elemente der Lebenszyklus-Konfiguration.

Aktivieren der Stufen S3 Intelligent-Tiering Archive Access und Deep Archive Access

Um die niedrigsten Speicherkosten für Daten zu erzielen, auf die innerhalb von Minuten bis Stunden zugegriffen werden kann, können Sie eine oder beide Archivzugriffsebenen aktivieren, indem Sie mit, oder Amazon S3 eine Konfiguration auf Bucket-, Präfix- oder Objekt-Tag-Ebene erstellenAPI. AWS Management Console AWS CLI

Verwenden der S3-Konsole

So aktivieren Sie die automatische S3 Intelligent-Tiering-Archivierung

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des von Ihnen erstellten Buckets aus.
3. Wählen Sie Properties (Eigenschaften).
4. Navigieren Sie zu den S3 Intelligent-Tiering-Archive-Konfigurationen und wählen Sie Erstellen einer Konfiguration aus.
5. Geben Sie im Abschnitt Archiveinstellungen einen beschreibenden Konfigurationsnamen für die Konfiguration des S3 Intelligent-Tiering-Archivs an.
6. Wählen Sie unter Auswählen eines Konfigurationsbereichs einen zu verwendenden Konfigurationsbereich aus. Optional können Sie den Konfigurationsbereich auf bestimmte Objekte innerhalb eines Buckets beschränken, indem Sie ein gemeinsames Präfix, ein Objekt-Tag oder eine Kombination dieser beiden verwenden.
   a. Um den Umfang der Konfiguration einzuschränken, wählen Sie Begrenzen des Umfangs dieser Konfiguration mithilfe eines oder mehrerer Filter aus.
   b. Um den Bereich der Konfiguration mit einem einzigen Präfix zu beschränken, geben Sie den Präfix unter Präfix ein.
   c. Um den Umfang der Konfiguration mithilfe von Objekt-Tags einzuschränken, wählen Sie Hinzufügen eines Tags, und geben Sie einen Wert für Schlüssel ein.
7. Wählen Sie unter Status Aktivieren aus.
8. Wählen Sie im Abschnitt Archiveinstellungen eine oder beide der Archivzugriffsebenen aus, die aktiviert werden sollen.
9. Wählen Sie Create (Erstellen) aus.

Verwenden Sie den AWS CLI

Sie können die folgenden AWS CLI Befehle verwenden, um S3 Intelligent-Tiering-Konfigurationen zu verwalten:

- delete-bucket-intelligent-tiering-configuration
- get-bucket-intelligent-tiering-configuration
- list-bucket-intelligent-tiering-configurations
- put-bucket-intelligent-tiering-configuration

Anweisungen zur Einrichtung von finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS CLI in der Amazon S3 API S3-Referenz. AWS CLI

Wenn Sie den verwenden AWS CLI, können Sie die Konfiguration nicht als XML Datei angeben. Sie müssen JSON stattdessen die angeben. Im Folgenden finden Sie ein Beispiel für eine XML S3 Intelligent-Tiering-Konfiguration und eine entsprechende KonfigurationJSON, die Sie in einem Befehl angeben können. AWS CLI

Im folgenden Beispiel wird eine S3-Intelligent-Tiering-Konfiguration in den angegebenen Bucket übertragen.

Example **put-bucket-intelligent-tiering-configuration**

JSON

```
{
  "Id": "string",
  "Filter": {
    "Prefix": "string",
    "Tag": {
      "Key": "string",
      "Value": "string"
    },
    "And": {
      "Prefix": "string",
      "Tags": [
        {
          "Key": "string",
          "Value": "string"
        }
```

```
        ...
      ]
    }
  },
  "Status": "Enabled"|"Disabled",
  "Tierings": [
    {
      "Days": integer,
      "AccessTier": "ARCHIVE_ACCESS"|"DEEP_ARCHIVE_ACCESS"
    }
    ...
  ]
}
```

XML

```
PUT /?intelligent-tiering&id=Id HTTP/1.1
Host: Bucket.s3.amazonaws.com
<?xml version="1.0" encoding="UTF-8"?>
<IntelligentTieringConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
   <Id>string</Id>
   <Filter>
      <And>
         <Prefix>string</Prefix>
         <Tag>
            <Key>string</Key>
            <Value>string</Value>
         </Tag>
         ...
      </And>
      <Prefix>string</Prefix>
      <Tag>
         <Key>string</Key>
         <Value>string</Value>
      </Tag>
   </Filter>
   <Status>string</Status>
   <Tiering>
      <AccessTier>string</AccessTier>
      <Days>integer</Days>
   </Tiering>
   ...
</IntelligentTieringConfiguration>
```

Verwenden der Operation PUT API

Sie können den `PutBucketIntelligentTieringConfiguration`-Vorgang für einen angegebenen Bucket und bis zu 1000 S3-Intelligent-Tiering-Konfigurationen pro Bucket nutzen. Mit einem gemeinsamen Präfix oder Objekt-Tag können Sie festlegen, welche Objekte innerhalb eines Buckets für die Archivzugriffsstufen berechtigt sind. Durch die Verwendung eines gemeinsamen Präfixes oder Objekt-Tags können Sie auf bestimmte Geschäftsanwendungen, Workflows oder interne Organisationen ausrichten. Außerdem haben Sie die Flexibilität, die Stufe Archivzugriff, die Stufe Deep Archive Access oder beide zu aktivieren.

Erste Schritte mit S3 Intelligent-Tiering

Weitere Informationen zur Verwendung von S3 Intelligent-Tiering finden Sie unter Tutorial: Erste Schritte mit S3 Intelligent-Tiering.

Verwenden von S3 Intelligent-Tiering

Die Speicherklasse S3 Intelligent-Tiering ermöglicht automatische Speicherkosteneinsparungen in drei Zugriffsebenen mit niedriger Latenz und hohem Durchsatz. Sie bietet außerdem optionale Archivierungsfunktionen, mit denen Sie die niedrigsten Speicherkosten in der Cloud für Daten erzielen, auf die innerhalb von Minuten bis Stunden zugegriffen werden kann.

Festlegen welche der S3-Intelligent-Tiering-Zugriffsebene-Objekte gespeichert sind in

Wenn Sie eine Liste Ihrer Objekte und der zugehörigen Metadaten, einschließlich der Stufe des S3-Intelligent-Tiering-Zugriffs, erhalten möchten, können Sie Amazon S3 Inventory nutzen. S3 Inventory bietetCSV,ORC, oder Parquet Ausgabedateien, die Ihre Objekte und die entsprechenden Metadaten auflisten. Sie können diese Bestandsberichte entweder täglich oder wöchentlich für einen Amazon-S3-Bucket oder ein gemeinsames Präfix erhalten. (Gemeinsames Präfix bezieht sich auf Objekte, die Namen haben, die mit einer bestimmten Zeichenfolge beginnen.)

Anzeigen des Archivstatus eines Objekts in S3 Intelligent-Tiering

Wenn Sie eine Benachrichtigung erhalten möchten, sobald ein Objekt innerhalb der Speicherklasse S3 Intelligent-Tiering entweder auf die Stufe Archive Access oder die Stufe Deep Archive Access verschoben wurde, können Sie S3-Ereignisbenachrichtigungen einrichten. Weitere Informationen finden Sie unter Aktivieren von Ereignisbenachrichtigungen.

Amazon S3 kann Ereignisbenachrichtigungen zu einem Amazon Simple Notification Service (AmazonSNS) -Thema, einer Amazon Simple Queue Service (AmazonSQS) -Warteschlange oder

einer AWS Lambda Funktion veröffentlichen. Weitere Informationen finden Sie unter Amazon-S3-Ereignis-Benachrichtigungen.

Das Folgende ist ein Beispiel für eine Nachricht, die Amazon S3 sendet, um ein `s3:IntelligentTiering`-Ereignis zu veröffentlichen. Weitere Informationen finden Sie unter the section called “Struktur von Ereignismeldungen”.

```
{
   "Records":[
      {
         "eventVersion":"2.3",
         "eventSource":"aws:s3",
         "awsRegion":"us-west-2",
         "eventTime":"1970-01-01T00:00:00.000Z",
         "eventName":"IntelligentTiering",
         "userIdentity":{
            "principalId":"s3.amazonaws.com"
         },
         "requestParameters":{
            "sourceIPAddress":"s3.amazonaws.com"
         },
         "responseElements":{
            "x-amz-request-id":"C3D13FE58DE4C810",
            "x-amz-id-2":"FMyUVURIY8/IgAtTv8xRjskZQpcIZ9KG4V5Wp6S7S/
JRWeUWerMUE5JgHvANOjpD"
         },
         "s3":{
            "s3SchemaVersion":"1.0",
            "configurationId":"testConfigRule",
            "bucket":{
               "name":"amzn-s3-demo-bucket",
               "ownerIdentity":{
                  "principalId":"A3NL1KOZZKExample"
               },
               "arn":"arn:aws:s3:::amzn-s3-demo-bucket"
            },
            "object":{
               "key":"HappyFace.jpg",
               "size":1024,
               "eTag":"d41d8cd98f00b204e9800998ecf8427e",
            }
         },
         "intelligentTieringEventData":{
```

```
            "destinationAccessTier": "ARCHIVE_ACCESS"
            }
      }
   ]
}
```

Sie können auch eine HEAD-Objekt-Anfrage nutzen, um den Archivstatus eines Objekts anzuzeigen. Wenn ein Objekt in der Speicherklasse S3 Intelligent-Tiering gespeichert wird und sich in einer der Archivebenen befindet, zeigt die HEAD-Objektantwort die aktuelle Archivebene an. Um die Archivierungsstufe anzuzeigen, verwendet die Anfrage x-amz-archive-statusKopfzeile.

Die folgende HEAD-Objektanforderung gibt die Metadaten eines Objekts zurück (in diesem Fall *my-image.jpg*).

Example

```
HEAD /my-image.jpg HTTP/1.1
Host: bucket.s3.region.amazonaws.com
Date: Wed, 28 Oct 2009 22:32:00 GMT
Authorization: AWS AKIAIOSFODNN7EXAMPLE:02236Q3V0RonhpaBX5sCYVf1bNRuU=
```

Darüber hinaus können Sie HEAD-Objektanforderungen verwenden, um den Status einer `restore-object`-Anforderung zu überwachen. Wenn die Archivwiederherstellung im Gange ist, enthält die HEAD Objektantwort den x-amz-restoreKopfzeile.

Das folgende Beispiel veranschaulicht die HEAD-Objektantwort, die ein mit S3 Intelligent-Tiering archiviertes Objekt mit einer Wiederherstellungsanforderung anzeigt.

Example

```
HTTP/1.1 200 OK
x-amz-id-2: FSVaTMjrmBp3Izs1NnwBZeu7M19iI8UbxMbi0A8AirHANJBo+hEftBuiESACOMJp
x-amz-request-id: E5CEFCB143EB505A
Date: Fri, 13 Nov 2020 00:28:38 GMT
Last-Modified: Mon, 15 Oct 2012 21:58:07 GMT
ETag: "1accb31fcf202eba0c0f41fa2f09b4d7"
x-amz-storage-class: 'INTELLIGENT_TIERING'
x-amz-archive-status: 'ARCHIVE_ACCESS'
x-amz-restore: 'ongoing-request="true"'
x-amz-restore-request-date: 'Fri, 13 Nov 2020 00:20:00 GMT'
Accept-Ranges: bytes
```

```
Content-Type: binary/octet-stream
Content-Length: 300
Server: AmazonS3
```

Wiederherstellen von Objekten aus den Stufen S3 Intelligent-Tiering Archive Access und Deep Archive Access

Um auf Objekte in den Stufen S3 Intelligent-Tiering Archive Access und Deep Archive Access zuzugreifen, müssen Sie eine Wiederherstellungsanforderung initiieren und dann warten, bis das Objekt in die Stufe „Häufiger Zugriff“ verschoben wird. Weitere Informationen zu archivierten Objekten finden Sie unter. the section called “Arbeiten mit archivierten Objekten”

Wenn Sie ein Objekt aus den Stufen Archive Access oder Deep Archive Access wiederherstellen, wird das Objekt zurück in die Stufe für häufige Zugriffe übergehen. Wenn anschließend nach 30 aufeinanderfolgenden Tagen nicht auf das Objekt zugegriffen wird, wird es automatisch in die Stufe „Infrequent Access“ verschoben. Das Objekt wechselt dann nach mindestens 90 aufeinanderfolgenden Tagen ohne Zugriff automatisch in die Stufe „Archive Access“. Das Objekt wechselt nach mindestens 180 aufeinanderfolgenden Tagen ohne Zugriff automatisch in die Stufe „Deep Archive Access“. Weitere Informationen finden Sie unter the section called “So funktioniert S3 Intelligent-Tiering”.

Sie können ein archiviertes Objekt mithilfe der Amazon S3-Konsole, S3 Batch Operations, Amazon S3 REST API AWS SDKs, The oder AWS Command Line Interface (AWS CLI) wiederherstellen. Weitere Informationen finden Sie unter the section called “Arbeiten mit archivierten Objekten”.

Grundlegendes zu S3 Glacier-Speicherklassen für die langfristige Datenspeicherung

Sie können Amazon S3 S3 Glacier-Speicherklassen verwenden, um kostengünstige Lösungen für die Speicherung langfristiger Daten bereitzustellen, auf die nicht oft zugegriffen wird. Die S3 Glacier-Speicherklassen sind:

- S3 Glacier Instant Retrieval
- S3 Glacier Flexible Retrieval
- S3 Glacier Deep Archive

Sie wählen eine dieser Speicherklassen basierend darauf aus, wie oft Sie auf Ihre Daten zugreifen und wie schnell Sie sie abrufen müssen. Jede dieser Speicherklassen bietet dieselbe Haltbarkeit

und Stabilität wie die Speicherklasse S3 Standard, jedoch zu niedrigeren Speicherkosten. Weitere Informationen zu den S3 Glacier-Speicherklassen finden Sie unter https://aws.amazon.com/s3/storage-classes/glacier/.

Topics

- Vergleich der S3 Glacier-Speicherklassen
- S3 Glacier Instant Retrieval
- S3 Glacier Flexible Retrieval
- S3 Glacier Deep Archive
- Grundlegendes zum Archivspeicher in S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive
- Wie unterscheiden sich diese Speicherklassen vom S3 Glacier-Service

Vergleich der S3 Glacier-Speicherklassen

Jede S3 Glacier-Speicherklasse hat eine Mindestspeicherdauer für alle Objekte. Wenn Sie das Objekt vor Ablauf der Mindestspeicherklasse löschen, überschreiben oder in eine andere Speicherklasse verschieben, wird Ihnen der Rest dieser Speicherdauer in Rechnung gestellt.

Einige S3 Glacier-Speicherklassen sind archivierbar, was bedeutet, dass die in diesen Klassen gespeicherten Objekte archiviert werden und nicht für den Echtzeitzugriff verfügbar sind. Weitere Informationen finden Sie unter Grundlegendes zum Archivspeicher in S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive.

Speicherklassen, die für weniger häufige Zugriffsmuster mit längeren Abrufzeiten konzipiert sind, bieten niedrigere Speicherkosten. Preisinformationen finden Sie unterhttps://aws.amazon.com/s3/pricing/.

In der folgenden Tabelle sind die wichtigsten Punkte zusammengefasst, die bei der Auswahl einer S3 Glacier-Speicherklasse zu berücksichtigen sind:

S3 Glacier Instant Retrieval

Wir empfehlen die Verwendung von S3 Glacier Instant Retrieval für Langzeitdaten, auf die einmal pro Quartal zugegriffen wird und für die Abrufzeiten im Millisekundenbereich erforderlich sind. Diese Speicherklasse ist ideal für leistungskritische Anwendungsfälle wie das Hosten von Bildern, Filesharing-Anwendungen und das Speichern von Krankenakten für den Zugriff bei Terminen.

Die Speicherklasse S3 Glacier Instant Retrieval bietet Echtzeitzugriff auf Ihre Objekte mit derselben Latenz und Durchsatzleistung wie die Speicherklasse S3 Standard-IA. Im Vergleich zu S3 Standard-IA hat S3 Glacier Instant Retrieval niedrigere Speicherkosten, aber höhere Datenzugriffskosten.

Für Daten, die in der Speicherklasse S3 Glacier Instant Retrieval gespeichert sind, gilt eine minimale Objektgröße von 128 KB. Diese Speicherklasse hat außerdem eine Mindestspeicherdauer von 90 Tagen.

S3 Glacier Flexible Retrieval

Wir empfehlen die Verwendung von S3 Glacier Flexible Retrieval für Archivdaten, auf die ein- bis zweimal pro Jahr zugegriffen wird und für die kein sofortiger Zugriff erforderlich ist. S3 Glacier Flexible Retrieval bietet flexible Abrufzeiten, um Ihnen dabei zu helfen, die Kosten im Gleichgewicht zu halten. Die Zugriffszeiten reichen von wenigen Minuten bis hin zu Stunden und kostenlose Sammelabrufe. Diese Speicherklasse ist ideal für Backup und Disaster Recovery.

In S3 Glacier Flexible Retrieval gespeicherte Objekte werden archiviert und sind nicht für den Echtzeitzugriff verfügbar. Weitere Informationen finden Sie unter Grundlegendes zum Archivspeicher in S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive. Um auf diese Objekte zuzugreifen, initiieren Sie zunächst eine Wiederherstellungsanforderung, die eine temporäre Kopie des Objekts erstellt, auf die Sie zugreifen können, wenn die Anforderung abgeschlossen ist. Weitere Informationen finden Sie unter Arbeiten mit archivierten Objekten. Wenn Sie ein Objekt wiederherstellen, können Sie eine Abrufstufe wählen, die Ihrem Anwendungsfall entspricht. Bei längeren Wiederherstellungszeiten sind die Kosten geringer.

Die folgenden Abrufstufen sind für S3 Glacier Flexible Retrieval verfügbar:

- Beschleunigter Abruf — Stellt das Objekt in der Regel innerhalb von 1—5 Minuten wieder her. Beschleunigte Abrufe unterliegen der Nachfrage. Um zuverlässige und vorhersehbare Wiederherstellungszeiten zu gewährleisten, empfehlen wir Ihnen, bereitgestellte Abrufkapazität zu erwerben. Weitere Informationen finden Sie unter Bereitgestellte Kapazität.
- Standardabruf — Stellt das Objekt in der Regel innerhalb von 3—5 Stunden oder innerhalb von 1 Minute bis 5 Stunden wieder her, wenn Sie S3 Batch Operations verwenden. Weitere Informationen finden Sie unter Objekte mit Batch Operations wiederherstellen.
- Massenabruf — Das Objekt wird in der Regel innerhalb von 5—12 Stunden wiederhergestellt. Massenabrufe sind kostenlos.

Die Mindestspeicherdauer für Objekte in der Speicherklasse S3 Glacier Flexible Retrieval beträgt 90 Tage.

S3 Glacier Flexible Retrieval erfordert 40 KB an zusätzlichen Metadaten für jedes Objekt. Dazu gehören 32 KB an Metadaten, die zur Identifizierung und zum Abrufen Ihrer Daten erforderlich sind. Diese werden zum Standardtarif für S3 Glacier Flexible Retrieval berechnet. Weitere 8 KB Daten sind erforderlich, um den benutzerdefinierten Namen und die Metadaten für archivierte Objekte zu verwalten, und werden zum S3-Standardtarif berechnet.

S3 Glacier Deep Archive

Wir empfehlen die Verwendung von S3 Glacier Deep Archive für Archivdaten, auf die weniger als einmal pro Jahr zugegriffen wird. Diese Speicherklasse ist für die Aufbewahrung von Datensätzen über mehrere Jahre konzipiert, um Compliance-Anforderungen zu erfüllen. Sie kann auch für Backup oder Disaster Recovery oder für Daten verwendet werden, auf die Sie selten zugreifen und auf deren Abruf Sie bis zu 72 Stunden warten können. S3 Glacier Deep Archive ist die kostengünstigste Speicheroption in AWS.

In S3 Glacier Deep Archive gespeicherte Objekte sind archiviert und für den Echtzeitzugriff nicht verfügbar. Weitere Informationen finden Sie unter Grundlegendes zum Archivspeicher in S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive. Um auf diese Objekte zuzugreifen, initiieren Sie zunächst eine Wiederherstellungsanforderung, die eine temporäre Kopie des Objekts erstellt, auf die Sie zugreifen können, wenn die Anfrage abgeschlossen ist. Weitere Informationen finden Sie unter Arbeiten mit archivierten Objekten. Wenn Sie ein Objekt wiederherstellen, können Sie eine Abrufstufe wählen, die Ihrem Anwendungsfall entspricht. Bei längeren Wiederherstellungszeiten sind die Kosten geringer.

Die folgenden Abrufstufen sind für S3 Glacier Deep Archive verfügbar:

- Standardabruf — Stellt das Objekt in der Regel innerhalb von 12 Stunden oder innerhalb von 9—12 Stunden wieder her, wenn Sie S3 Batch Operations verwenden. Weitere Informationen finden Sie unter Objekte mit Batch Operations wiederherstellen.
- Massenabruf — Stellt das Objekt in der Regel innerhalb von 48 Stunden wieder her, und das zu einem Bruchteil der Kosten des Standard-Abrufs.

Die Mindestspeicherdauer für Objekte in der Speicherklasse S3 Glacier Deep Archive beträgt 180 Tage.

S3 Glacier Deep Archive benötigt 40 KB an zusätzlichen Metadaten für jedes Objekt. Dazu gehören 32 KB an Metadaten, die zur Identifizierung und zum Abrufen Ihrer Daten erforderlich sind. Diese werden zum Standardtarif für S3 Glacier Deep Archive berechnet. Weitere 8 KB Daten sind erforderlich, um den benutzerdefinierten Namen und die Metadaten für archivierte Objekte beizubehalten, und werden zum S3-Standardtarif berechnet.

Grundlegendes zum Archivspeicher in S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive

S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive sind Archivspeicherklassen. Das heißt, wenn Sie ein Objekt in diesen Speicherklassen speichern, wird dieses Objekt archiviert und es kann nicht direkt darauf zugegriffen werden. Um auf ein archiviertes Objekt zuzugreifen, senden Sie eine Wiederherstellungsanforderung für das Objekt und warten dann, bis der Dienst das Objekt wiederherstellt. Die Wiederherstellungsanforderung stellt eine temporäre Kopie des Objekts wieder her. Diese Kopie wird gelöscht, wenn die von Ihnen in der Anforderung angegebene Dauer abgelaufen ist. Weitere Informationen finden Sie unter Arbeiten mit archivierten Objekten.

Die Überführung von Objekten in die Speicherklasse S3 Glacier Deep Archive ist nicht umkehrbar.

Wenn Sie die Speicherklasse eines archivierten Objekts in eine andere Speicherklasse ändern möchten, müssen Sie die Wiederherstellungsoperation verwenden, um zunächst eine temporäre Kopie des Objekts zu erstellen. Verwenden Sie dann den Kopiervorgang, um das Objekt zu überschreiben, indem Sie S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval oder Reduced Redundancy Storage als Speicherklasse angeben.

> **Note**
>
> Die Copy-Operation für wiederhergestellte Objekte wird in der Amazon-S3-Konsole für Objekte in der Speicherklasse S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive nicht unterstützt. Verwenden Sie für diese Art von Kopiervorgang die AWS Command Line Interface (AWS CLI) AWS SDKs, die oder die RESTAPI.

Sie können archivierte Objekte in diesen Speicherklassen mit bis zu 1.000 Transaktionen pro Sekunde (TPS) an Objektwiederherstellungsanforderungen pro Konto pro Konto wiederherstellen AWS-Region.

Kostenüberlegungen

Wenn Sie vorhaben, Daten mit seltenem Zugriff für einen Zeitraum von Monaten oder Jahren zu archivieren, können die Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive Ihre Speicherkosten reduzieren. Sie sollten jedoch Folgendes berücksichtigen, um sicherzustellen, dass die Speicherklasse S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive geeignet für Sie ist:

- Speichergemeinkosten — Für jedes archivierte Objekt sind 40 KB an zusätzlichen Metadaten erforderlich. Dazu gehören 32 KB an Metadaten, die für die Identifizierung und den Abruf Ihrer Daten erforderlich sind. Diese werden zum Standardtarif für diese Speicherklasse berechnet. Weitere 8 KB Daten sind erforderlich, um den benutzerdefinierten Namen und die Metadaten für archivierte Objekte beizubehalten, und werden zum S3-Standardtarif berechnet.

  Wenn Sie kleine Objekte archivieren, sollten Sie diese Speichergebühren berücksichtigen. Ziehen Sie auch in Betracht, viele kleine Objekte in wenigen großen Objekten zusammenzufassen, um Kosten für den Verwaltungsaufwand zu reduzieren.
- Preise für mehrteilige Uploads — Objekte in S3-storage-class-glacier; und S3 Glacier Deep Archive werden zu den Tarifen der Speicherklasse S3 Standard abgerechnet, wenn Sie sie mithilfe von mehrteiligen Uploads hochladen. Weitere Informationen finden Sie unter Mehrteiliger Upload und Preise.
- Speichergebühren für mindestens 30 Tage — S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive sind langfristige Archivierungslösungen. Die minimale Speicherdauer beträgt 90 Tage für die Speicherklasse S3 Glacier Flexible Retrieval und 180 Tage für S3 Glacier Deep Archive. Für das Löschen von Daten, die in diesen Speicherklassen archiviert wurden, fallen keine Gebühren an, wenn die von Ihnen gelöschten Objekte länger als die Mindestspeicherdauer archiviert werden. Wenn Sie ein archiviertes Objekt innerhalb der Mindestdauer löschen oder überschreiben, berechnet Amazon S3 für den Rest dieser Dauer Gebühren.
- Gebühren für den Datenabruf — Wenn Sie archivierte Objekte auf S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive wiederherstellen, fallen Gebühren für den Datenabruf pro Anfrage an. Diese Gebühren variieren je nach der Abrufstufe, die Sie bei der Initiierung einer Wiederherstellung wählen. Informationen zu den Preisen finden Sie unter Amazon S3 – Preise.
- S3 Lifecycle — Wenn Sie archivierte Objekte auf S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive wiederherstellen, fallen pro Anfrage Gebühren für den Datenabruf an. Diese Gebühren variieren je nach der Abrufstufe, die Sie bei der Initiierung einer Wiederherstellung wählen. Informationen zu den Preisen finden Sie unter Amazon S3 – Preise.

Wiederherstellen archivierter Objekte

Auf archivierte Objekte kann nicht in Echtzeit zugegriffen werden. Sie müssen zuerst eine Anforderung zur Wiederherstellung initiieren und dann warten, bis eine temporäre Kopie des Objekts für die Dauer bereitgestellt wird, die Sie in der Anforderung angegeben haben. Nachdem Sie eine temporäre Kopie des wiederhergestellten Objekts erhalten haben, bleibt die Speicherklasse des Objekts S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive. (A) HeadObject oder GetObjectAPIDie Betriebsanforderung gibt S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive als Speicherklasse zurück.)

> **Note**
>
> Bei einer Wiederherstellung eines Archivs zahlen Sie sowohl für das Archiv (Tarif für S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive) als auch für die temporär wiederhergestellte Kopie (S3-Standard-Speichertarif). Informationen zu Preisen finden Sie unter Amazon S3 – Preise.

Für den Zugriff auf ein kopiertes Objekt müssen Sie eine Anfrage zur Wiederherstellung initiieren. Das kann programmgesteuert oder über die Amazon-S3-Konsole erfolgen. Amazon S3 verarbeitet jeweils nur eine Anfrage pro Objekt. Weitere Informationen finden Sie unter Wiederherstellen eines archivierten Objekts.

Wie unterscheiden sich diese Speicherklassen vom S3 Glacier-Service

Die S3 Glacier-Speicherklassen sind Teil des Amazon S3 S3-Service und speichern Daten als Objekte in S3-Buckets. Sie können Objekte in diesen Speicherklassen mit der S3-Konsole oder programmgesteuert mit S3 oder verwalten. APIs SDKs Wenn Sie Objekte in S3 Glacier-Speicherklassen speichern, können Sie S3-Funktionen wie erweiterte Verschlüsselung, Objekt-Tagging und S3-Lifecycle-Konfigurationen verwenden, um den Datenzugriff und die Kosten zu verwalten.

> **Important**
>
> Wir empfehlen, die S3 Glacier-Speicherklassen innerhalb des Amazon S3 S3-Service für all Ihre langfristigen Daten zu verwenden.

Der Amazon S3 Glacier (S3 Glacier) Service ist ein separater Service, der Daten als Archive in Tresoren speichert. Dieser Service unterstützt keine Amazon S3 S3-Funktionen und bietet keine Konsolenunterstützung für das Hoch- und Herunterladen von Daten. Wir empfehlen nicht, den S3 Glacier-Service für Ihre langfristigen Daten zu verwenden. Auf die in diesem Service gespeicherten Daten kann über den Amazon S3 S3-Service nicht zugegriffen werden. Informationen zum S3 Glacier-Service finden Sie im Amazon S3 Glacier Developer Guide. Informationen zum Übertragen von Daten vom Amazon S3 Glacier-Service zu einer Speicherklasse in Amazon S3 finden Sie unter Datenübertragung von Amazon S3 Glacier Vaults zu Amazon S3 in der AWS Lösungsbibliothek.

Arbeiten mit archivierten Objekten

Sie können Objekte, auf die selten zugegriffen wird, archivieren, um Ihre Speicherkosten dafür zu senken. Wenn Sie ein Objekt archivieren, wird es in einen kostengünstigen Speicher verschoben. Sie können also nicht in Echtzeit darauf zugreifen.

Auf archivierte Objekte kann zwar nicht in Echtzeit zugegriffen werden, Sie können sie jedoch je nach Speicherklasse innerhalb von Minuten oder Stunden wiederherstellen. Sie können ein archiviertes Objekt mithilfe der Amazon S3 S3-Konsole, S3 Batch Operations RESTAPI, the AWS SDKs, the und AWS Command Line Interface (AWS CLI) wiederherstellen. Detaillierte Anweisungen finden Sie unter Wiederherstellen eines archivierten Objekts.

Amazon-S3-Objekte in den folgenden Speicherklassen oder -stufen werden archiviert. Es kann nicht in Echtzeit darauf zugegriffen werden:

- Die Speicherklasse S3 Glacier Flexible Retrieval
- Die Speicherklasse S3 Glacier Deep Archive
- Die Zugriffsebene S3 Intelligent-Tiering Archive
- Die S3-Intelligent-Tiering-Deep-Archive-Zugriffsebene

Um archivierte Objekte wiederherzustellen, müssen Sie die folgenden Schritte ausführen:

- Für Objekte in den Speicherklassen S3 Glacier Flexible Retrieval und S3-Glacier Deep Archive müssen Sie eine Wiederherstellungsanforderung einleiten und warten, bis eine temporäre Kopie des Objekts verfügbar ist. Wenn eine temporäre Kopie des wiederhergestellten Objekts erstellt wird, bleibt die Speicherklasse des Objekts unverändert. (A) HeadObject oder GetObjectAPIDie Operationsanforderung gibt S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive als Speicherklasse zurück.)

- Für Objekte in den Zugriffsebenen S3 Intelligent-Tiering Archive und S3 Intelligent-Tiering Deep Archive müssen Sie eine Wiederherstellungsanforderung einleiten und warten, bis das Objekt in die Stufe für häufigen Zugriff verschoben wird.

Weitere Informationen darüber, wie sich alle Amazon S3-Speicherklassen vergleichen, finden Sie unter Amazon S3 S3-Speicherklassen verstehen und verwalten. Weitere Informationen zu S3 Intelligent-Tiering finden Sie unter the section called “So funktioniert S3 Intelligent-Tiering”.

Die Zeit, die für den Abschluss eines Wiederherstellungsauftrags benötigt wird, hängt davon ab, welche Archivspeicherklasse oder Speicherebene Sie verwenden und welche Abrufoption Sie angeben: Beschleunigt (nur für S3 Glacier Flexible Retrieval und S3 Intelligent Tiering Archive Access verfügbar), Standard, oder Bulk. Weitere Informationen finden Sie unter Die Optionen zum Abrufen von Archiven verstehen.

Sie können über Amazon-S3-Ereignisbenachrichtigungen über den Abschluss der Wiederherstellung benachrichtigt werden. Weitere Informationen finden Sie unter Amazon-S3-Ereignis-Benachrichtigungen.

Wiederherstellen von Objekten aus S3 Glacier

Wenn Sie S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive verwenden, stellt Amazon S3 eine temporäre Kopie des Objekts nur für die angegebene Dauer wieder her. Anschließend wird die Kopie des wiederhergestellten Objekts gelöscht. Sie können den Ablaufzeitraum einer wiederhergestellten Kopie durch die erneute Ausgabe einer Wiederherstellungsanforderung ändern. In diesem Fall aktualisiert Amazon S3 den Ablaufzeitraum relativ zum aktuellen Zeitraum.

> **Note**
>
> Bei einer Wiederherstellung eines archivierten Objekts von S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive zahlen Sie sowohl für das archivierte Objekt als auch für die temporär wiederhergestellte Kopie. Informationen zu Preisen finden Sie unter Amazon S3 – Preise.

Amazon S3 berechnet die Ablaufzeit der wiederhergestellten Objektkopie durch Addition der Anzahl der in der Wiederherstellungsanforderung angegebenen Tage zum Zeitpunkt, wenn die angeforderte Wiederherstellung abgeschlossen wird. Amazon S3 rundet dann die sich ergebende Zeit auf den nächsten Tag um Mitternacht (Koordinierte WeltzeitUTC) () ab. Nehmen wir zum Beispiel an, dass

am 15. Oktober 2012 um 10:30 Uhr UTC eine wiederhergestellte Objektkopie erstellt wurde und der Wiederherstellungszeitraum auf 3 Tage festgelegt wurde. In diesem Fall läuft die wiederhergestellte Kopie am 19. Oktober 2012 um 00:00 Uhr UTC ab. Zu diesem Zeitpunkt löscht Amazon S3 die Objektkopie.

Wiederherstellen von Objekten aus S3 Intelligent-Tiering

Wenn Sie ein Objekt aus der S3 Intelligent-Tiering Archive Access Tier oder S3 Intelligent-Tiering Deep Archive Access Tier wiederherstellen, wird das Objekt zurück in die S3 Intelligent-Tiering Frequent Access Tier verschoben. Wenn dnach 30 aufeinanderfolgenden Tagen nicht auf das Objekt zugegriffen wird, wird es automatisch in Infrequent Access Tier verschoben. Das Objekt wechselt nach mindestens 90 aufeinanderfolgenden Tagen ohne Zugriff automatisch in die S3 Intelligent-Tiering Archive Access Tier. Wenn mindestens 180 aufeinanderfolgende Tage nicht auf ein Objekt zugegriffen wird, wird das Objekt in Deep Archive Access Tier verschoben.

> **Note**
>
> Im Gegensatz zu den Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive akzeptieren Wiederherstellungsanforderungen für S3-Intelligent-Tiering-Objekte den `Days`-Wert nicht.

Verwenden von S3-Batch-Operationen mit Wiederherstellungsanforderungen

Zur Wiederherstellung von mehr als einem Amazon S3-Objekt in einer einzigen Anforderung können Sie S3-Batch-Vorgänge nutzen. Sie stellen S3-Batch-Vorgänge eine Liste von Objekten zur Verfügung, für die Vorgänge ausgeführt werden sollen. S3 Batch Operations ruft die entsprechende API Operation auf, um die angegebene Operation auszuführen. Ein einzelner Batch-Vorgangsauftrag kann die angegebene Operation für Milliarden von Objekten ausführen, die Exabytes von Daten enthalten.

Themen

- Die Optionen zum Abrufen von Archiven verstehen
- Wiederherstellen eines archivierten Objekts

Die Optionen zum Abrufen von Archiven verstehen

Nachfolgend finden Sie die verfügbaren Abrufoptionen bei der Wiederherstellung eines archivierten Objekts in Amazon S3:

- Beschleunigt: Greifen Sie schnell auf Ihre Daten zu, die in der Speicherklasse „S3 Glacier Flexible Retrieval“ oder der S3 Intelligent-Tiering-Zugriffsebene „ Archive“ gespeichert sind. Sie können diese Option verwenden, wenn gelegentliche dringende Anforderungen für eine Teilmenge von Archiven erforderlich sind. Daten, die unter Verwendung von beschleunigten Abrufen abgerufen werden, stehen normalerweise innerhalb von 1 bis 5 Minuten zur Verfügung, außer es handelt sich um die größten archivierten Objekte (250 MB+).

  > **Note**
  >
  > Beschleunigte Abrufe sind eine Premium-Funktion und werden zum Tarif für beschleunigte Anfragen und Abrufe berechnet.
  > Informationen zu Amazon-S3-Preisen finden Sie unter Amazon S3-Preise.

  Die bereitgestellte Kapazität hilft sicherzustellen, dass für Expedited-Abrufe aus S3 Glacier Flexible Retrieval Abrufkapazität verfügbar ist, wenn Sie sie benötigen. Weitere Informationen finden Sie unter Bereitgestellte Kapazität.
- Standard: Greifen Sie auf Ihre archivierten Objekte innerhalb einiger Stunden zu. Standard ist die Standardoption für Abrufanforderungen, in denen keine Abrufoption angegeben ist. Die Standardabrufe enden in der Regel innerhalb von 3 bis 5 Stunden für Objekte, die in der S3 Glacier Flexible Retrieval Speicherklasse oder der S3 Intelligent Tiering Archive Access Stufe gespeichert sind. Diese Abrufe enden in der Regel innerhalb von 12 Stunden für Objekte, die in der Speicherklasse S3 Glacier Deep Archive oder S3 Intelligent-Tiering Deep Archive Access Tier gespeichert sind. Standardabrufe sind für Objekte, die in S3 Intelligent-Tiering gespeichert sind, kostenlos.

  > **Note**
  >
  > - Bei Objekten, die in der Speicherklasse S3 Glacier Flexible Retrieval oder der Stufe S3 Intelligent-Tiering Archive Access gespeichert sind, beginnen Standardabrufe, die mithilfe des Wiederherstellungsvorgangs S3 Batch Operations initiiert wurden, in der Regel innerhalb von Minuten und enden innerhalb von 3 bis 5 Stunden.

- Bei Objekten der S3 Glacier Deep Archive Archive-Speicherklasse oder der S3 Intelligent-Tiering Deep Archive Access-Stufe beginnen Standardabrufe, die mithilfe des Batch Operations-Wiederherstellungsvorgangs initiiert wurden, in der Regel innerhalb von 9 Stunden und enden innerhalb von 12 Stunden.

- Bulk: Greifen Sie über die kostengünstigste Abrufoption von Amazon S3 Glacier auf Ihre Daten zu. Mit Massenabrufen können Sie Daten in großen Mengen (sogar im Petabyte-Bereich) kostengünstig abrufen.

  Bei Objekten, die in der Speicherklasse S3 Glacier Flexible Retrieval oder der Stufe S3 Intelligent-Tiering Archive Access gespeichert sind, werden Massenabrufe in der Regel innerhalb von 5—12 Stunden abgeschlossen. Bei Objekten, die in der Speicherklasse S3 Glacier Deep Archive oder der S3 Intelligent-Tiering Deep Archive Access-Stufe gespeichert sind, werden diese Abrufe in der Regel innerhalb von 48 Stunden abgeschlossen.

  Massenabrufe sind für Objekte, die in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Intelligent-Tiering gespeichert sind, kostenlos.

Die folgende Tabelle fasst die Archivabrufoptionen zusammen. Informationen zu Preisen finden Sie unter Amazon S3 – Preise.

Um einen, oder `Bulk` Abruf durchzuführen `ExpeditedStandard`, setzen Sie das Anforderungselement in der `Tier` RestoreObjectRESTAPIDie Operationsanforderung auf die gewünschte Option oder das Äquivalent in AWS Command Line Interface (AWS CLI) oder AWS SDKs. Wenn Sie bereitgestellte Durchsatzkapazität gekauft haben, werden alle Expedited-Abrufe automatisch über Ihre bereitgestellte Kapazität erledigt.

Bereitgestellte Kapazität

Die bereitgestellte Kapazität hilft sicherzustellen, dass für Ihre Expedited-Abrufe aus S3 Glacier Flexible Retrieval Abrufkapazität verfügbar ist, wenn Sie sie benötigen. Jede Kapazitätseinheit sorgt dafür, dass mindestens drei beschleunigte Abrufe alle 5 Minuten ausgeführt werden können, und sie bietet einen Abrufdurchsatz von bis zu 150 Megabyte pro Sekunde (). MBps

Denken Sie über den Kauf bereitgestellter Abrufkapazität nach, wenn Ihr Workload einen sehr zuverlässigen und vorhersehbaren Zugriff auf eine Untermenge Ihrer Daten innerhalb von Minuten erforderlich macht. Ohne bereitgestellte Kapazität werden beschleunigte Abrufe in Zeiten hoher Nachfrage möglicherweise nicht akzeptiert. Wenn Sie unbedingt Zugriff auf beschleunigte Abrufe benötigen, sollten Sie eine bereitgestellte Abrufkapazität kaufen.

Bereitgestellte Kapazitätseinheiten werden einem zugewiesen. AWS-Konto Daher sollte der Antragsteller des beschleunigten Datenabrufs die bereitgestellte Kapazitätseinheit erwerben, nicht der Bucket-Eigentümer.

Sie können bereitgestellte Kapazität erwerben, indem Sie die Amazon S3-Konsole, die Amazon S3 Glacier-Konsole, den REST API Vorgang Purchase Provisioned Capacity AWS SDKs, den oder den verwenden. AWS CLI Weitere Informationen zu den Preisen für die bereitgestellte Kapazität finden Sie in der Amazon S3-Preisliste.

Geschwindigkeiten für Wiederherstellungsanforderungen für S3 Glacier

Wenn Sie Wiederherstellungsanforderungen für Objekte initiieren, die in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive gespeichert sind, wird ein Kontingent für Wiederherstellungsanforderungen für Ihr AWS-Konto angewendet. S3 Glacier unterstützt Wiederherstellungsanforderungen mit einer Geschwindigkeit von 1 000 Transaktionen pro Sekunde. Wenn diese Rate überschritten wird, werden andernfalls gültige Anfragen gedrosselt oder zurückgewiesen und Amazon S3 gibt einen `ThrottlingException`-Fehler zurück.

Optional können Sie auch S3-Massenoperationen verwenden, um eine große Anzahl von Objekten, die in S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive gespeichert sind, mit einer einzigen Anforderung abzurufenu. Weitere Informationen finden Sie unter Massenausführung von Objektoperationen mit Batch Operations.

Wiederherstellen eines archivierten Objekts

Amazon-S3-Objekte in den folgenden Speicherklassen oder -stufen werden archiviert. Es kann nicht in Echtzeit darauf zugegriffen werden:

- Die Speicherklasse S3 Glacier Flexible Retrieval
- Die Speicherklasse S3 Glacier Deep Archive
- Die Zugriffsebene S3 Intelligent-Tiering Archive
- Die S3-Intelligent-Tiering-Deep-Archive-Zugriffsebene

Ein Zugriff auf in der Speicherklasse S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive gespeicherte Amazon-S3-Objekte ist nicht unmittelbar möglich. Um auf ein Objekt in diesen Speicherklassen zuzugreifen, müssen Sie eine temporäre Kopie des Objekts für eine angegebene Dauer (Anzahl von Tagen) in seinem S3-Bucket wiederherstellen. Wenn Sie eine permanente Kopie des Objekts benötigen, stellen Sie das Objekt wieder her und legen Sie dann eine Kopie davon

in Ihrem Amazon S3-Bucket an. Das Kopieren wiederhergestellter Objekte wird in der Amazon-S3-Konsole nicht unterstützt. Verwenden Sie für diese Art von Kopiervorgang AWS Command Line Interface (AWS CLI) AWS SDKs, das oder das. REST API Wenn Sie keine Kopie erstellen und die Speicherklasse ändern, wird das Objekt weiterhin in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive gespeichert. Informationen zur Verwendung dieser Speicherklassen finden Sie unter Speicherklassen für Objekte, auf die selten zugegriffen wird.

Für Zugriff auf Objekte in den S3 Intelligent-Tiering Archive Access und Deep Archive Access müssen Sie eine Wiederherstellungsanforderung initiieren und warten, bis das Objekt in die Stufe für häufigen Zugriff verschoben wird, um darauf zuzugreifen. Wenn Sie ein Objekt aus den Stufen Archive Access oder Deep Archive Access wiederherstellen, wird das Objekt zurück in die Stufe für häufige Zugriffe übergehen. Informationen zur Verwendung dieser Speicherklassen finden Sie unter Speicherklasse zur automatischen Optimierung von Daten mit sich ändernden oder unbekannten Zugriffsmustern.

Allgemeine Informationen über archivierte Objekte finden Sie unter Arbeiten mit archivierten Objekten.

> **Note**
>
> - Wenn Sie ein archiviertes Objekt aus den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive wiederherstellen, zahlen Sie sowohl für das archivierte Objekt als auch für die Kopie, die Sie vorübergehend wiederhergestellt haben.
> - Wenn Sie ein Objekt aus S3 Intelligent-Tiering wiederherstellen, fallen keine Abrufgebühren für Standard- oder Massenabrufe an.
> - Nachfolgende Wiederherstellungsanforderungen, die für archivierte Objekte aufgerufen werden, die bereits wiederhergestellt wurden, werden als Anfragen abgerechnet. `GET` Informationen zu Preisen finden Sie unter Amazon S3 – Preise.

Wiederherstellen eines archivierten Objekts

Sie können ein archiviertes Objekt mithilfe der Amazon S3-Konsole, Amazon S3 RESTAPI, the AWS SDKs, the AWS Command Line Interface (AWS CLI) oder S3 Batch Operations wiederherstellen.

Verwenden der S3-Konsole

Wiederherstellung von Objekten mithilfe der Amazon-S3-Konsole

Gehen Sie wie folgt vor, um ein Objekt wiederherzustellen, das in den Speicherklassen „S3 Glacier Flexible Retrieval“ oder „S3 Glacier Deep Archive“ bzw. in den Speicherstufen „S3 Intelligent-Tiering Archive Access“ oder „Deep Archive Access“ archiviert wurde.

So stellen Sie ein archiviertes Objekt wieder her

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der die Objekte enthält, die Sie wiederherstellen möchten.
4. Wählen Sie in der Liste Object (Objekt) das/die Objekt(e) aus, das/die Sie wiederherstellen möchten. Wählen Sie anschließend Actions (Aktionen) und dann Initiate restore (Wiederherstellung initiieren) aus.
5. Wenn Sie von S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive wiederherstellen, geben Sie im Dialogfeld Anzahl der Tage, die die wiederhergestellte Kopie verfügbar ist die Anzahl der Tage ein, die die archivierten Daten verfügbar sein sollen.
6. Führen Sie bei Abrufoptionen einen der folgenden Schritte aus:

   - Wählen Sie Massenabruf oder Standardabruf und dann Wiederherstellen starten aus.
   - Wählen Sie Expedited retrieval (Expedited-Abruf) (nur für S3 Glacier Flexible Retrieval oder S3 Intelligent Tiering Archive Access verfügbar). Wenn Sie ein Objekt in S3 Glacier Flexible Retrieval wiederherstellen, können Sie auswählen, ob Sie bereitgestellte Kapazität für Ihren Expressabruf erwerben möchten. Wenn Sie bereitgestellte Kapazität erwerben möchten, fahren Sie mit dem nächsten Schritt fort. Wenn Sie dies nicht wünschen, wählen Sie Wiederherstellen starten aus.

   > **Note**
   >
   > Objekte aus den Stufen S3 Intelligent-Tiering Archive Access und Deep Archive Access werden automatisch in der Stufe für häufige Zugriffe wiederhergestellt.

7. (Optional) Wenn Sie ein Objekt in S3 Glacier Flexible Retrieval wiederherstellen und Beschleunigter Abruf auswählen, können Sie sich entscheiden, ob Sie bereitgestellte Kapazität erwerben möchten. Die bereitgestellte Kapazität steht nur für Objekte in S3 Glacier

Flexible Retrieval zur Verfügung. Wenn Sie bereits über bereitgestellte Kapazität verfügen, wählen Sie Wiederherstellen starten aus, um einen bereitgestellten Abruf zu starten.

Wenn Sie bereitgestellte Durchsatzkapazität gekauft haben, werden alle Ihre Expedited-Abrufe automatisch über Ihre bereitgestellte Kapazität erledigt. Weitere Informationen finden Sie unter Bereitgestellte Kapazität.

- Wenn Sie nicht über bereitgestellte Kapazität verfügen und auch keine kaufen möchten, wählen Sie Wiederherstellen starten aus.
- Wenn Sie keine bereitgestellte Kapazität haben, aber bereitgestellte Kapazitätseinheiten (PCUs) kaufen möchten, wählen Sie Kaufen. PCUs Wählen Sie im PCUs Dialogfeld Kaufen aus, wie viele PCUs Sie kaufen möchten, bestätigen Sie Ihren Kauf und wählen Sie dann Kaufen aus. PCUs Wenn Sie die Meldung Kauf erfolgreich erhalten, wählen Sie Wiederherstellen starten aus, um den bereitgestellten Abruf zu starten.

Verwenden Sie den AWS CLI

Objekte aus S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive wiederherstellen

Im folgenden Beispiel wird der `restore-object` Befehl verwendet, um das Objekt wiederherzustellen *dir1/example.obj* 25 Tage im Eimer. *amzn-s3-demo-bucket*

```
aws s3api restore-object --bucket amzn-s3-demo-bucket --key dir1/example.obj --restore-request '{"Days":25,"GlacierJobParameters":{"Tier":"Standard"}}'
```

Wenn die im Beispiel verwendete JSON Syntax zu einem Fehler auf einem Windows-Client führt, ersetzen Sie die Wiederherstellungsanforderung durch die folgende Syntax:

```
--restore-request Days=25,GlacierJobParameters={"Tier"="Standard"}
```

Objekte aus S3 Intelligent-Tiering Archive Access und Deep Archive Access wiederherstellen

Im folgenden Beispiel wird der `restore-object` Befehl verwendet, um das Objekt wiederherzustellen *dir1/example.obj* im Bucket zur Stufe für häufigen *amzn-s3-demo-bucket* Zugriff.

```
aws s3api restore-object --bucket amzn-s3-demo-bucket --key dir1/example.obj --restore-request '{}'
```

> **Note**
>
> Im Gegensatz zu den Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive akzeptieren Wiederherstellungsanforderungen für S3-Intelligent-Tiering-Objekte den `Days`-Wert nicht.

Den Wiederherstellungsstatus überwachen

Sie können den folgenden `head-object`-Befehl verwenden, um den Status Ihrer `restore-object`-Anforderung zu überwachen:

```
aws s3api head-object --bucket amzn-s3-demo-bucket --key dir1/example.obj
```

Weitere Informationen finden Sie unter restore-objectin der AWS CLI Befehlsreferenz.

Mit dem REST API

Amazon S3 bietet Ihnen einen API Vorgang, mit dem Sie die Wiederherstellung eines archivierten Objekts initiieren können. Weitere Informationen finden Sie unter RestoreObjectin der Amazon Simple Storage Service API Reference.

Mit dem AWS SDKs

Beispiele für die Wiederherstellung archivierter Objekte in S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive mit dem finden Sie in den AWS SDKs Codebeispielen in der Amazon S3 API S3-Referenz.

Verwendung von S3 Batch Operations

Zur Wiederherstellung von mehr als einem archivierten Objekt in einer einzigen Anforderung können Sie S3 Batch Operations nutzen. Sie stellen S3 Batch Operations eine Liste von Objekten zur Verfügung, für die Vorgänge ausgeführt werden sollen. S3 Batch Operations ruft die entsprechende API Operation auf, um die angegebene Operation auszuführen. Ein einzelner Batch-Vorgangsauftrag kann die angegebene Operation für Milliarden von Objekten ausführen, die Exabytes von Daten enthalten.

Um einen Batch-Operations-Auftrag zu erstellen, benötigen Sie ein Manifest, das nur die Objekte enthält, die Sie wiederherstellen möchten. Sie können mithilfe von S3 Inventory ein Manifest erstellen

oder eine CSV Datei mit den erforderlichen Informationen bereitstellen. Weitere Informationen finden Sie unter the section called “Angeben eines Manifests”.

Bevor Sie S3-Batch-Operations-Aufträge erstellen und ausführen, müssen Sie Amazon S3 die Erlaubnis erteilen, S3-Batch-Operations in Ihrem Namen durchzuführen. Die erforderlichen Berechtigungen finden Sie unter the section called “Gewähren von Berechtigungen”.

> **Note**
>
> Batch-Operations-Aufträge können entweder mit Objekte der Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive oder mit Objekten der Speicherstufen S3 Intelligent-Tiering Archive Access und Deep Archive Access arbeiten. Batch Operations kann im selben Auftrag nicht für beide Typen von archivierten Objekten ausgeführt werden. Um Objekte beider Typen wiederherzustellen, müssen Sie separate Batchoperations-Aufgaben erstellen.
>
> Weitere Informationen zur Verwendung von Batch Operations zum Wiederherstellen archivierter Objekte finden Sie unter the section called “Wiederherstellen von Objekten”.

So erstellen Sie einen S3-Auftrag „Initiate Restore Object Batch Operations“

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Klicken Sie im linken Navigationsbereich auf Batchvorgänge.
3. Wählen Sie Job erstellen aus.
4. Wählen Sie für AWS-Region die Region aus, in der Sie Ihren Auftrag erstellen möchten.
5. Wählen Sie unter Manifestformat das zu verwendende Manifest aus.

   - Wenn Sie den S3-Inventarbericht wählen, geben Sie den Pfad zu dem `manifest.json` Objekt ein, das Amazon S3 als Teil des CSV -formatierten Inventarberichts generiert hat. Wenn Sie nicht das aktuelle Manifest verwenden möchten, können Sie optional die Versions-ID des `manifest.json`-Objekts angeben.
   - Wenn Sie möchten CSV, geben Sie den Pfad zu einem Manifest-Objekt im CSV -Format ein. Das Manifestobjekt muss das in der Konsole beschriebene Format befolgen. Wenn Sie nicht die aktuelle Version verwenden möchten, können Sie optional die Versions-ID des Manifestobjekts angeben.
6. Wählen Sie Weiter.

7. Wählen Sie im Abschnitt Operation die Option Wiederherstellen aus.
8. Wählen Sie im Abschnitt Wiederherstellen bei Quelle wiederherstellen entweder Glacier Flexible Retrieval oder Glacier Deep Archive bzw. die Stufe Intelligent-Tiering Archive Access oder Deep Archive Access aus.

   Wenn Sie Glacier Flexible Retrieval oder Glacier Deep Archive auswählen, geben Sie eine Zahl für die Anzahl der Tage, an denen die wiederhergestellte Kopie verfügbar ist ein.

   Wählen Sie bei Abrufstufe die zu verwendende Stufe aus.
9. Wählen Sie Weiter.
10. Füllen Sie auf der Seite Zusätzliche Optionen konfigurieren die folgenden Abschnitte aus:

    - Geben Sie im Abschnitt Zusätzliche Optionen eine Beschreibung des Auftrags und eine Prioritätsnummer für den Auftrag an. Höhere Nummern bedeuten eine höhere Priorität. Weitere Informationen finden Sie unter the section called “Zuweisen der Auftragspriorität”.
    - Wählen Sie im Abschnitt Abschlussbericht aus, ob Batch Operations einen Abschlussbericht erstellen soll. Weitere Informationen zu den Fertigstellungsberichten finden Sie unter the section called “Abschlussberichte”.
    - Im Abschnitt Berechtigungen müssen Sie Amazon S3 die Erlaubnis erteilen, Batch Operations in Ihrem Namen durchzuführen. Die erforderlichen Berechtigungen finden Sie unter the section called “Gewähren von Berechtigungen”.
    - (Optional) fügen Sie im Abschnitt Auftrags-Tags Tags in Schlüssel-Wert-Paaren hinzu. Weitere Informationen finden Sie unter the section called “Verwenden von Markierungen”.

    Wählen Sie Weiter aus, sobald Sie fertig sind.
11. Überprüfen Sie die Einstellungen auf der Seite Review. Wenn Sie Änderungen vornehmen müssen, wählen Sie Previous. Wählen Sie andernfalls Auftrag erstellen.

Weitere Informationen über Batch Operations finden Sie unter Objekte mit Batch Operations wiederherstellen und Erstellen eines S3-Batch-Vorgangsauftrags.

Überprüfen Sie den Wiederherstellungsstatus und das Ablaufdatum

Sie können den Status einer Wiederherstellungsanfrage oder das Ablaufdatum überprüfen, indem Sie die Amazon S3-Konsole, Amazon S3 S3-Ereignisbenachrichtigungen AWS CLI, die oder Amazon S3 verwenden RESTAPI.

> **Note**
>
> Objekte, die aus den Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive wiederhergestellt wurden, werden nur für die von Ihnen angegebene Anzahl von Tagen gespeichert. Die folgenden Verfahren geben das Ablaufdatum für diese Kopien zurück. Objekte, die aus den Speicherstufen S3 Intelligent-Tiering Archive Access und Deep Archive Access wiederhergestellt wurden, haben kein Ablaufdatum und werden stattdessen zurück in die Stufe für häufigen Zugriff verschoben.

Verwenden der S3-Konsole

So überprüfen Sie den Wiederherstellungsstatus und das Ablaufdatum eines Objekts in der Amazon-S3-Konsole

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der die Objekte enthält, die Sie wiederherstellen.
4. Wählen Sie in der Liste Objekte das Objekt aus, das Sie wiederherstellen. Die Seite mit den Objektdetails wird angezeigt.

   - Wenn die Wiederherstellung noch nicht abgeschlossen ist, wird oben auf der Seite der Abschnitt Wiederherstellung in Bearbeitung angezeigt.
   - Wenn die Wiederherstellung abgeschlossen ist, wird oben auf der Seite der Abschnitt Wiederherstellung abgeschlossen angezeigt. Wenn Sie aus S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive wiederherstellen, werden in diesem Abschnitt auch das Ablaufdatum der Wiederherstellung angezeigt. Amazon S3 entfernt die wiederhergestellte Kopie aus dem archivierten Objekt an diesem Datum.

Verwenden von Amazon S3 S3-Ereignisbenachrichtigungen

Sie können über den Abschluss der Objektwiederherstellung informiert werden, indem Sie die `s3:ObjectRestore:Completed` Aktion mit der Amazon S3 S3-Funktion „Event Notifications“ verwenden. Weitere Informationen zum Aktivieren von Ereignisbenachrichtigungen finden Sie unter Aktivieren von Benachrichtigungen mithilfe von Amazon SQSSNS, Amazon und AWS Lambda.

Weitere Informationen zu den verschiedenen `ObjectRestore` Ereignistypen finden Sie unterthe section called “Unterstützte Ereignistypen für SQSSNS, und Lambda”.

Mit dem AWS CLI

Überprüfen Sie den Wiederherstellungsstatus und das Ablaufdatum eines Objekts mit dem AWS CLI

Im folgenden Beispiel wird der `head-object` Befehl verwendet, um Metadaten für das Objekt anzuzeigen *`dir1/example.obj`* im Eimer. *`amzn-s3-demo-bucket`* Wenn Sie diesen Befehl für ein wiederhergestelltes Objekt ausführen, gibt Amazon S3 zurück, ob die Wiederherstellung noch läuft und (falls zutreffend) das Ablaufdatum.

```
aws s3api head-object --bucket amzn-s3-demo-bucket --key dir1/example.obj
```

Erwartete Ausgabe (Wiederherstellung läuft):

```
{
    "Restore": "ongoing-request=\"true\"",
    "LastModified": "2020-06-16T21:55:22+00:00",
    "ContentLength": 405,
    "ETag": "\"b662d79adeb7c8d787ea7eafb9ef6207\"",
    "VersionId": "wbYaE2vtOV0iIBXrOqGAJt3fP1cHB8Wi",
    "ContentType": "binary/octet-stream",
    "ServerSideEncryption": "AES256",
    "Metadata": {},
    "StorageClass": "GLACIER"
}
```

Erwartete Ausgabe (Wiederherstellung abgeschlossen):

```
{
    "Restore": "ongoing-request=\"false\", expiry-date=\"Wed, 12 Aug 2020 00:00:00 GMT
\"",
    "LastModified": "2020-06-16T21:55:22+00:00",
    "ContentLength": 405,
    "ETag": "\"b662d79adeb7c8d787ea7eafb9ef6207\"",
    "VersionId": "wbYaE2vtOV0iIBXrOqGAJt3fP1cHB8Wi",
    "ContentType": "binary/octet-stream",
    "ServerSideEncryption": "AES256",
    "Metadata": {},
    "StorageClass": "GLACIER"
}
```

Weitere Informationen über `head-object` finden Sie unter head-objectin der AWS CLI Befehlsreferenz.

Mit dem REST API

Amazon S3 bietet Ihnen einen API Vorgang zum Abrufen von Objektmetadaten. Informationen zum Überprüfen des Wiederherstellungsstatus und des Ablaufdatums eines archivierten Objekts mithilfe von finden Sie unter REST API HeadObjectin der Amazon Simple Storage Service API Reference.

Upgraden der Geschwindigkeit einer Wiederherstellung in Bearbeitung

Sie können die Geschwindigkeit einer Wiederherstellung in Bearbeitung upgraden.

So upgraden Sie eine Wiederherstellung in Bearbeitung auf eine schnellere Stufe:

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, der die Objekte enthält, die Sie wiederherstellen möchten.
4. Wählen Sie in der Liste Objekte das Objekt aus, das Sie wiederherstellen. Die Seite mit den Objektdetails wird angezeigt. Wählen Sie auf der Detailseite des Objekts Upgrade-Abrufstufe. Informationen zum Überprüfen des Wiederherstellungsstatus eines Objekts finden Sie unter Überprüfen Sie den Wiederherstellungsstatus und das Ablaufdatum.
5. Wählen Sie die Stufe aus, auf die Sie upgraden möchten, und dann Wiederherstellen starten aus.

Verwaltung des Lebenszyklus von Objekten

S3 Lifecycle hilft Ihnen dabei, Objekte während ihres gesamten Lebenszyklus kostengünstig zu speichern, indem Sie sie in kostengünstigere Speicherklassen umwandeln oder abgelaufene Objekte in Ihrem Namen löschen. Um den Lebenszyklus Ihrer Objekte zu verwalten, erstellen Sie eine S3-Lifecycle-Konfiguration für Ihren Bucket. Eine S3-Lebenszyklus-Konfiguration besteht aus einer Reihe von Regeln, mit denen Aktionen definiert werden, die Amazon S3 auf eine Gruppe von Objekten anwendet. Es gibt zwei Aktionstypen:

- Übergangsaktionen – Diese Aktionen definieren, wann Objekte in eine andere Speicherklasse übergehen. Beispielsweise können Sie festlegen, dass Objekte 30 Tage nach ihrer Erstellung in die Speicherklasse S3 Standard-IA (IA steht für „Infrequent Access“, seltener Zugriff) übergehen

und ein Jahr nach ihrer Erstellung in der Speicherklasse S3 Glacier Flexible Retrieval archiviert werden sollen. Weitere Informationen finden Sie unter Amazon S3 S3-Speicherklassen verstehen und verwalten.

Mit Lebenszyklus-Überführungsanforderungen sind Kosten verbunden. Informationen zu den Preisen finden Sie unter Amazon S3 – Preise.

- Ablaufaktionen – Diese Aktionen definieren, wann Objekte ablaufen. In Amazon S3 werden die abgelaufenen Objekte für Sie gelöscht. Sie könnten sich beispielsweise dafür entscheiden, Objekte ablaufen zu lassen, nachdem sie für einen bestimmten Zeitraum aufbewahrt wurden. Weitere Informationen finden Sie unter Auslaufende Objekte.

  Potenzielle Kosten sind nur dann mit dem Ablauf des Lebenszyklus verbunden, wenn Sie Objekte in einer Speicherklasse mit einer Mindestspeicherdauer ablaufen lassen. Weitere Informationen finden Sie unter Gebühren für Mindestspeicherdauer.

> **Important**
>
> Sie können keine Bucket-Richtlinie verwenden, um Löschungen oder Übergänge durch eine S3-Lebenszyklusregel zu verhindern. Selbst wenn Ihre Bucket-Richtlinie beispielsweise alle Aktionen für alle Principals ablehnt, funktioniert Ihre S3-Lifecycle-Konfiguration weiterhin wie gewohnt.

Bestehende und neue Objekte

Wenn Sie einem Bucket eine Lebenszyklus-Konfiguration hinzufügen, gelten die Konfigurationsregeln für vorhandene Objekte und für Objekte, die Sie später hinzufügen. Wenn Sie beispielsweise heute eine Lebenszyklus-Konfigurationsregel mit einer Ablaufaktion hinzufügen, die bewirkt, dass Objekte 30 Tage nach ihrer Erstellung ablaufen, stellt Amazon S3 eine Warteschlange für die Entfernung aller vorhandenen Objekte, die älter als 30 Tage sind, in die Warteschlange.

Änderungen bei der Abrechnung

Wenn es zwischen dem Zeitpunkt, an dem ein Objekt für eine Lebenszyklusaktion berechtigt wird, und dem Zeitpunkt, zu dem Amazon S3 Ihr Objekt überträgt oder abläuft, zu Verzögerungen kommt, werden die Abrechnungsänderungen angewendet, sobald das Objekt für die Lebenszyklusaktion berechtigt ist. Wenn ein Objekt beispielsweise abläuft und Amazon S3 das Objekt nicht sofort ablaufen lässt, wird Ihnen nach Ablauf der Ablaufzeit kein Speicherplatz in Rechnung gestellt.

Die einzige Ausnahme von diesem Verhalten ist, wenn Sie eine Lebenszyklusregel für den Übergang zur S3-Intelligent-Tiering-Speicherklasse haben. In diesem Fall treten Abrechnungsänderungen erst auf, wenn das Objekt in S3 Intelligent-Tiering übergeben worden ist. Weitere Informationen zu S3-Lebenszyklusregeln finden Sie unter Elemente der Lebenszyklus-Konfiguration.

> **Note**
>
> Für Lebenszyklusübergänge fallen keine Gebühren für den Datenabruf an. Es fallen jedoch Gebühren pro Anfrage an, wenn Sie Lebenszyklusregeln verwenden `PUTCOPY`, um Daten in eine beliebige S3-Speicherklasse zu verschieben. Berücksichtigen Sie die Kosten für die Aufnahme oder Übertragung, bevor Sie Objekte in eine beliebige Speicherklasse verschieben. Weitere Information zu Kostenaspekten finden Sie unter Amazon S3 – Preise.

Überwachung der Auswirkungen von Lebenszyklusregeln

Informationen zur Überwachung der Auswirkungen von Aktualisierungen, die durch aktive Lebenszyklusregeln vorgenommen wurden, finden Sie unterthe section called “Wie überwache ich die im Rahmen meiner Lebenszyklusregeln ergriffenen Maßnahmen?”.

Verwaltung des gesamten Lebenszyklus von Objekten

Mit den S3 Lifecycle-Konfigurationsregeln können Sie Amazon S3 anweisen, Objekte in kostengünstigere Speicherklassen zu überführen, zu archivieren oder zu löschen. z. B.:

- Wenn Sie regelmäßig Protokolle in einen Bucket hochladen, werden diese möglicherweise nur für eine Woche oder einen Monat von Ihrer Anwendung benötigt. Anschließend sollen diese gelöscht werden.
- Auf einige Dokumente erfolgt für einen begrenzten Zeitraum ein häufiger Zugriff. Danach wird nur selten auf diese zugegriffen. Irgendwann benötigen Sie keinen Echtzeitzugriff mehr auf die Dokumente, aber seitens Ihrer Organisation oder der Gesetzgebung müssen Sie diese noch für einen gewissen Zeitraum archivieren. Nach Ablauf dieser Zeitspanne können Sie sie löschen.
- Sie können auch einige Datentypen primär für Archivierungszwecke in Amazon S3 hochladen. Sie könnten beispielsweise digitale Medien, Finanz- und Gesundheitsdaten, rohe Genomsequenzdaten, langfristige Datenbank-Backups und Daten archivieren, die aus Gründen der Einhaltung gesetzlicher Vorschriften aufbewahrt werden müssen.

Indem Sie S3-Lifecycle-Aktionen kombinieren, um den gesamten Lebenszyklus eines Objekts zu verwalten. Angenommen, die von Ihnen erstellten Objekte haben einen definierten Lebenszyklus. Anfänglich erfolgt ein häufiger Zugriff auf die Objekte für einen Zeitraum von 30 Tagen. Danach erfolgt nur noch ein seltener Zugriff für bis zu 90 Tage. Anschließend werden die Objekte nicht mehr benötigt, daher können Sie entscheiden, sie zu archivieren oder zu löschen.

In diesem Szenario erstellen Sie eine S3-Lebenszyklusregel, in der Sie eine erste Überführungsaktion zu S3 Intelligent-Tiering, S3 Standard-IA oder S3 One Zone-IA, eine weitere Überführungsaktion zu S3 Glacier Flexible Retrieval zu Archivierungszwecken und schließlich eine Ablaufaktion angeben. Wenn Sie die Objekte von einer Speicherklasse in eine andere verschieben, sparen Sie Speicherkosten. Weitere Information zu Kostenaspekten finden Sie unter Amazon S3 – Preise.

Themen

- Übergang von Objekten mit Amazon-S3-Lebenszyklus
- Auslaufende Objekte
- Einstellung einer S3-Lifecycle-Konfiguration für einen Bucket
- Wie S3 Lifecycle mit anderen Bucket-Konfigurationen interagiert
- Konfiguration von S3 Lifecycle-Ereignisbenachrichtigungen
- Elemente der Lebenszyklus-Konfiguration
- Wie Amazon Amazon S3 mit Konflikten in Lebenszykluskonfigurationen umgeht
- Beispiele für S3-Lifecycle-Konfigurationen
- Behebung von Problemen mit Amazon S3 Lifecycle

Übergang von Objekten mit Amazon-S3-Lebenszyklus

Sie können einer S3-Lifecycle-Konfiguration Übergangsaktionen hinzufügen, um Amazon S3 anzuweisen, Objekte in eine andere Amazon S3 S3-Speicherklasse zu verschieben. Weitere Informationen über Speicherklassen finden Sie unter Amazon S3 S3-Speicherklassen verstehen und verwalten. Einige Beispiele dafür, wann Sie S3-Lebenszykluskonfigurationen auf diese Weise verwenden könnten, sind die folgenden:

- Wenn Sie wissen, dass nur selten auf bestimmte Objekte zugegriffen wird, können Sie diese in die Speicherklasse S3 Standard-IA übergeben.

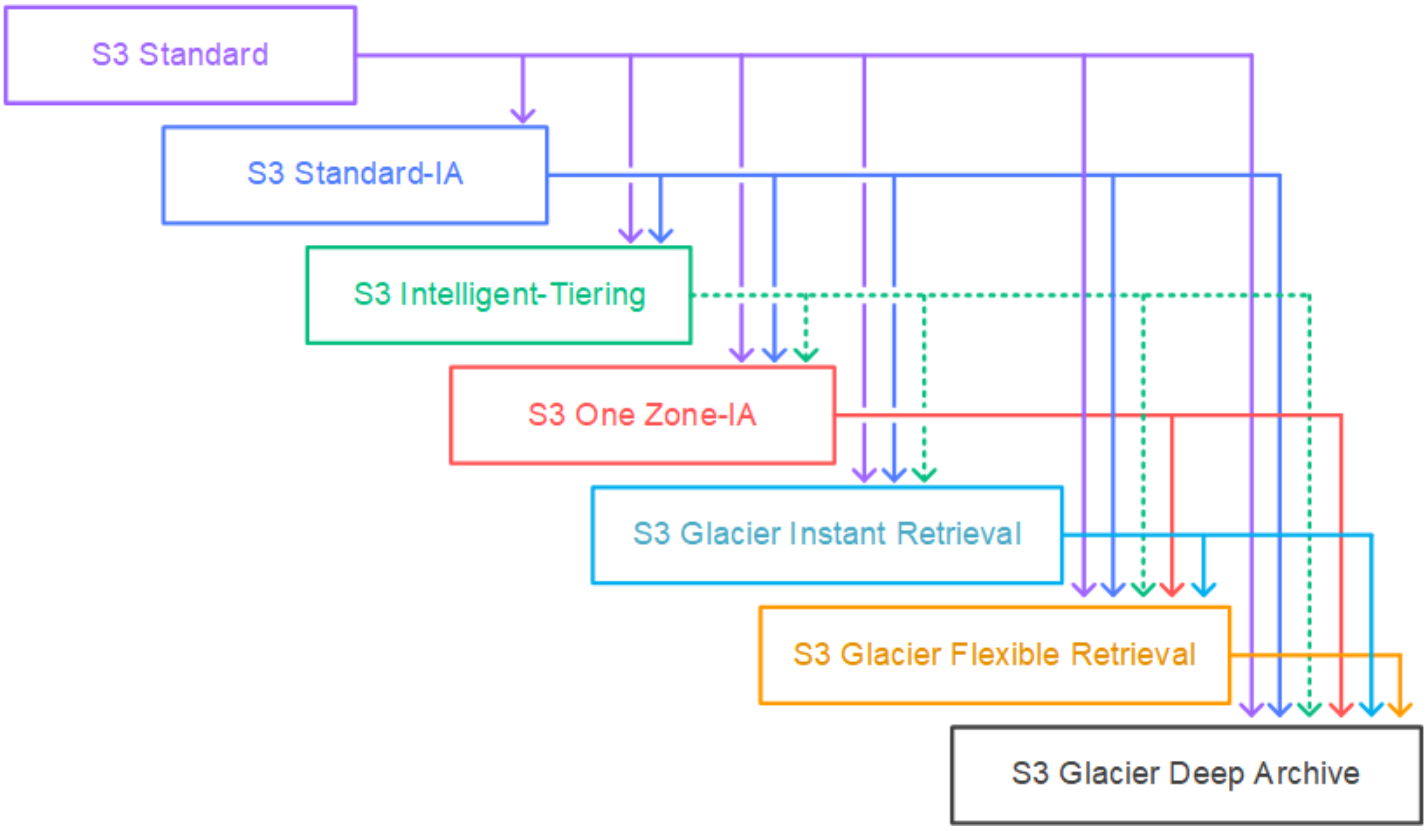
- Möglicherweise möchten Sie Objekte, auf die Sie nicht zugreifen müssen, in Echtzeit in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive archivieren.

> **Note**
>
> Verschlüsselte Objekte bleiben während des gesamten Übergangsprozesses der Speicherklasse verschlüsselt.

Unterstützte Übergänge

Sie können in einer S3-Lebenszyklus-Konfiguration Regeln definieren, um Objekte aus einer Speicherklasse zu einer anderen Speicherklasse zu überführen, um Speicherkosten zu sparen. Wenn Sie die Zugriffsmuster Ihrer Objekte nicht kennen oder sich Ihre Zugriffsmuster über die Zeit verändern, können Sie die Objekte zur Speicherklasse S3 Intelligent-Tiering überführen, um automatische Kosteneinsparungen zu erzielen. Weitere Informationen über Speicherklassen finden Sie unter Amazon S3 S3-Speicherklassen verstehen und verwalten.

Amazon S3 unterstützt das Wasserfallmodell für die Überführung zwischen Speicherklassen wie im folgenden Diagramm gezeigt.



Unterstützte Lebenszyklusübergänge

Amazon S3 unterstützt die folgenden Lebenszyklusübergänge zwischen Speicherklassen mittels einer S3-Lebenszykluskonfiguration.

Die folgenden Übergänge sind möglich:

- Von der Speicherklasse S3 Standard zu einer beliebigen anderen Speicherklasse.
- Von der Speicherklasse S3-Standard-IA zu den Speicherklassen S3 Intelligent-Tiering, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive.
- Von der Speicherklasse S3 Intelligent-Tiering zu den Speicherklassen S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive.

> **Note**
>
> Es gibt einige Ausnahmen für die Übertragung von Objekten von der Speicherklasse S3 Intelligent-Tiering auf S3 One Zone-IA und einige S3 Glacier-Speicherklassen. Weitere Informationen finden Sie unter the section called “Nicht unterstützte Lebenszyklusübergänge”.

- Von der Speicherklasse S3 One Zone-IA zu den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive.
- Von der Speicherklasse S3 Glacier Instant Retrieval zu den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive.
- Die Speicherklasse S3 Glacier Flexible Retrieval zur Speicherklasse S3 Glacier Deep Archive.

> **Note**
>
> S3 Lifecycle überträgt Objekte asynchron auf S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive. Es kann eine Verzögerung zwischen dem Übergangsdatum in der S3-Lebenszyklus-Konfigurationsregel und dem Datum des physischen Übergangs geben. In diesem Fall wird Ihnen der Standardtarif der Speicherklasse, von der Sie umgestellt haben, auf der Grundlage des in der Regel angegebenen Übergangsdatums berechnet.

- Von einer beliebigen Speicherklasse zur Speicherklasse S3 Glacier Deep Archive.

Nicht unterstützte Lebenszyklusübergänge

Amazon S3 unterstützt keine der folgenden Lebenszyklusübergänge.

Die folgenden Übergänge sind nicht möglich:

- Bei Buckets mit aktivierter Versionierung oder angehaltener Versionierung alle Objekte mit einem Replikationsstatus. `Pending`
- Von jeder Speicherklasse zur Speicherklasse S3 Standard.
- Beliebige Speicherklasse der Klasse Reduced Redundancy Storage (). RRS
- Von der S3-One-Zone-IA-Speicherklasse zu den Speicherklassen S3 Intelligent-Tiering, S3 Standard-IA und S3 Glacier Instant Retrieval.
- Von der Speicherklasse S3 Intelligent-Tiering (alle Stufen) zur Speicherklasse S3 Standard-IA.
- Die Archive-Instant-Access-Speicherklasse S3 Intelligent-Tiering entspricht der Stufe S3 One Zone-IA.
- Die Archive-Access-Stufe der Speicherklasse S3 Intelligent-Tiering für S3 One Zone-IA oder S3 Glacier Instant Retrieval.
- Die Deep Archive Access-Stufe der Speicherklasse S3 Intelligent-Tiering umfasst S3 One Zone-IA, S3 Glacier Instant Retrieval oder S3 Glacier Flexible Retrieval.

Einschränkungen und Überlegungen für Übergänge

Für die Übergänge der Lebenszyklus-Speicherklassen gelten folgende Einschränkungen:

Objekte, die kleiner als 128 KB sind, werden standardmäßig nicht in eine Speicherklasse umgewandelt

Amazon S3 wendet auf S3-Lifecycle-Konfigurationen ein Standardverhalten an, das verhindert, dass Objekte, die kleiner als 128 KB sind, in eine beliebige Speicherklasse übertragen werden. Wir empfehlen nicht, Objekte mit weniger als 128 KB zu übertragen, da Ihnen für jedes Objekt eine Übergangsanfrage in Rechnung gestellt wird. Das bedeutet, dass bei kleineren Objekten die Umstellungskosten die Speichereinsparungen überwiegen können. Weitere Informationen zu den Kosten für Umstellungsanfragen finden Sie unter Anfragen und Datenabrufe auf der Amazon S3 S3-Preisseite auf der Registerkarte Speicher und Anfragen.

Um kleineren Objekten den Übergang zu ermöglichen, können Sie Ihren Lebenszyklus-Übergangsregeln einen Objektgrößenfilter hinzufügen, der eine benutzerdefinierte Mindestgröße (`ObjectSizeGreaterThan`) oder Maximalgröße (`ObjectSizeLessThan`) festlegt. Weitere Informationen finden Sie unter Beispiel: Ermöglicht die Übertragung von Objekten, die kleiner als 128 KB sind.

> **Note**
>
> Im September 2024 hat Amazon S3 das standardmäßige Übergangsverhalten für kleine Objekte wie folgt aktualisiert:
>
> - Aktualisiertes Standard-Übergangsverhalten — Ab September 2024 verhindert das Standardverhalten, dass Objekte, die kleiner als 128 KB sind, in eine beliebige Speicherklasse übertragen werden.
> - Bisheriges Standard-Übergangsverhalten — Vor September 2024 erlaubte das Standardverhalten, dass Objekte, die kleiner als 128 KB waren, nur in die Speicherklassen S3 Glacier und S3 Glacier Deep Archive übertragen wurden.
>
> Konfigurationen, die vor September 2024 erstellt wurden, behalten das vorherige Übergangsverhalten bei, sofern Sie sie nicht ändern. Das heißt, wenn Sie Regeln erstellen, bearbeiten oder löschen, ändert sich das standardmäßige Übergangsverhalten für Ihre Konfiguration in das aktualisierte Verhalten. Falls es Ihr Anwendungsfall erfordert, können Sie das standardmäßige Übergangsverhalten so ändern, dass Objekte, die kleiner als 128 KB

sind, auf S3 Glacier und S3 Glacier Deep Archive umgestellt werden. Verwenden Sie dazu den optionalen `x-amz-transition-object-size-minimum-default` Header in einer PutBucketLifecycleConfigurationAnfrage.

Objekte müssen mindestens 30 Tage lang gespeichert werden, bevor sie auf S3 Standard-IA oder S3 One Zone-IA umgestellt werden

Bevor Sie Objekte in S3 Standard-IA oder S3 One Zone-IA überführen, müssen diese mindestens 30 Tage in Amazon S3 gespeichert worden sein. Beispielsweise können Sie keine Lebenszyklusregel erstellen, mit der Objekte einen Tag nach dem Erstellungsdatum in die Speicherklasse S3 Standard-IA übergehen. Amazon S3 unterstützt diesen Übergang erst nach 30 Tagen, da auf neuere Objekte meist häufiger zugegriffen wird oder sie schneller gelöscht werden, als das für S3 Standard-IA- oder S3 One Zone-IA-Speicher sinnvoll wäre.

Beim Übergang von nicht aktuellen Objekten (in versionsfähigen Buckets) können dementsprechend nur Objekte, die seit mindestens 30 Tagen nicht mehr aktuell sind, in S3 Standard-IA- oder S3 One Zone-IA-Speicher übergehen. Eine Liste der Mindestspeicherdauer für alle Speicherklassen finden Sie unter. Vergleich der Amazon-S3-Speicherklassen

Ihnen wird die Übertragung von Objekten vor Ablauf ihrer Mindestspeicherdauer in Rechnung gestellt

Bestimmte Speicherklassen haben eine Mindestspeicherdauer für Objekte. Wenn Sie Objekte vor Ablauf der Mindestdauer aus diesen Speicherklassen herausnehmen, wird Ihnen der Rest dieser Dauer in Rechnung gestellt. Weitere Informationen darüber, welche Speicherklassen eine Mindestspeicherdauer haben, finden Sie unterVergleich der Amazon-S3-Speicherklassen.

Sie können keine einzelne Lebenszyklusregel erstellen, die Objekte vor Ablauf der Mindestspeicherdauer von einer Speicherklasse in eine andere überträgt.

Beispielsweise hat S3 Glacier Instant Retrieval eine Mindestspeicherdauer von 90 Tagen. Sie können keine Lebenszyklusregel angeben, die Objekte nach 4 Tagen in S3 Glacier Instant Retrieval überführt und Objekte dann nach 20 Tagen in S3 Glacier Deep Archive überträgt. In diesem Fall muss die Umstellung auf S3 Glacier Deep Archive nach mindestens 94 Tagen erfolgen.

Sie können dafür zwei Regeln angeben, zahlen jedoch die Speichergebühren für die Mindestdauer. Weitere Information zu Kostenaspekten finden Sie unter Amazon S3 – Preise.

Weitere Informationen zum Erstellen eines S3-Lebenszyklus finden Sie unterEinstellung einer S3-Lifecycle-Konfiguration für einen Bucket.

Übergang in die Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive (Objektarchivierung)

Mithilfe einer S3 Lifecycle-Konfiguration können Sie Objekte zur Archivierung in die Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive übertragen.

Bevor Sie Ihre Objekte archivieren, lesen Sie die folgenden Abschnitte, wo Sie weitere relevante Aspekte finden.

Allgemeine Überlegungen

Bevor Sie Objekte archivieren, sollten Sie die folgenden allgemeinen Überlegungen in Betracht ziehen:

- Verschlüsselte Objekte bleiben während des gesamten Übergangsprozesses der Speicherklasse verschlüsselt.
- Objekte, die in den Speicherklassen S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive gespeichert sind, sind nicht in Echtzeit verfügbar.

  Archivierte Objekte sind Amazon-S3-Objekte, aber um auf ein archiviertes Objekt zugreifen zu können, müssen Sie zuerst eine temporäre Kopie davon wiederherstellen. Die wiederhergestellte Objektkopie ist nur für die Dauer verfügbar, die Sie in der Wiederherstellungsanforderung angeben. Danach löscht Amazon S3 die temporäre Kopie und das Objekt bleibt in S3 Glacier Flexible Retrieval archiviert.

  Sie können ein Objekt mithilfe der Amazon S3-Konsole oder programmgesteuert mithilfe der AWS SDK Wrapper-Bibliotheken oder Amazon S3 REST API in Ihrem Code wiederherstellen. Weitere Informationen finden Sie unter Wiederherstellen eines archivierten Objekts.
- Objekte, die in der Speicherklasse S3 Glacier Flexible Retrieval gespeichert sind, können nur in die Speicherklasse S3 Glacier Deep Archive überführt werden.

  Sie können mit einer S3-Lebenszyklusregel die Speicherklasse eines Objekts von S3 Glacier Flexible Retrieval nur zur Speicherklasse S3 Glacier Deep Archive ändern. Wenn Sie die Speicherklasse eines in S3 Glacier Flexible Retrieval gespeicherten Objekts zu einer anderen Speicherklasse als S3 Glacier Deep Archive ändern möchten, müssen Sie die Wiederherstellungsoperation verwenden, um zunächst eine temporäre Kopie des Objekts zu erstellen. Anschließend verwenden Sie die Kopieroperation, um das Objekt zu überschreiben und dabei S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 One Zone-IA oder Reduced Redundancy als Speicherklasse anzugeben.

- Die Überführung von Objekten in die Speicherklasse S3 Glacier Deep Archive ist nicht umkehrbar.

  Sie können nicht mit einer S3-Lebenszyklusregel die Speicherklasse eines Objekts von S3 Glacier Deep Archive zu einer anderen Speicherklasse ändern. Wenn Sie die Speicherklasse eines archivierten Objekts in eine andere Speicherklasse ändern möchten, müssen Sie die Wiederherstellungsoperation verwenden, um zunächst eine temporäre Kopie des Objekts zu erstellen. Verwenden Sie dann den Kopiervorgang, um das Objekt zu überschreiben, indem Sie S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval oder Reduced Redundancy Storage als Speicherklasse angeben.

  > **Note**
  >
  > Die Copy-Operation für wiederhergestellte Objekte wird in der Amazon-S3-Konsole für Objekte in der Speicherklasse S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive nicht unterstützt. Verwenden Sie für diese Art von Kopiervorgang die AWS Command Line Interface (AWS CLI) AWS SDKs, die oder die. REST API

  Die Objekte, die in den Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive gespeichert sind, sind nur über Amazon S3 sichtbar und verfügbar. Sie sind nicht über den separaten Amazon-S3-Glacier-Service verfügbar.

  Dies sind Amazon S3-Objekte, auf die Sie nur über die Amazon S3 S3-Konsole oder Amazon S3 zugreifen könnenAPI. Sie können nicht über die separate Amazon S3 Glacier-Konsole oder Amazon S3 Glacier auf die archivierten Objekte zugreifenAPI.

Kostenüberlegungen

Wenn Sie vorhaben, Daten mit seltenem Zugriff für einen Zeitraum von Monaten oder Jahren zu archivieren, können die Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive Ihre Speicherkosten reduzieren. Sie sollten jedoch Folgendes berücksichtigen, um sicherzustellen, dass die Speicherklasse S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive geeignet für Sie ist:

- Gebühren für zusätzlichen Speicheraufwand – Wenn Sie Objekte zur Speicherklasse S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive überführen, wird jedem Objekt eine feste Speichermenge hinzugefügt, um die Metadaten für die Verwaltung des Objekts zu berücksichtigen.

- Für jedes in S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive archivierte Objekt verwendet Amazon S3 8 KB Speicher für den Namen des Objekts und andere Metadaten. Amazon S3 speichert diese Metadaten, sodass Sie mithilfe von Amazon S3 eine Echtzeitliste Ihrer archivierten Objekte abrufen könnenAPI. Weitere Informationen finden Sie unter Get Bucket (List Objects). Für diesen zusätzlichen Speicherplatz werden Ihnen -S3-Standardgebühren in Rechnung gestellt.
- Für jedes Objekt, das in S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive archiviert wird, fügt Amazon S3 32 KB Speicher für Index und zugehörige Metadaten hinzu. Diese zusätzlichen Daten sind erforderlich, um Ihr Objekt zu identifizieren und wiederherzustellen. Für diesen zusätzlichen Speicherplatz werden Ihnen die Gebühren für S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive in Rechnung gestellt.

  Wenn Sie kleine Objekte archivieren, sollten Sie diese Speichergebühren berücksichtigen. Ziehen Sie auch in Betracht, viele kleine Objekte in wenigen großen Objekten zusammenzufassen, um Kosten für den Verwaltungsaufwand zu reduzieren.

- Anzahl der Tage, für die die Objekte archiviert werden sollen – S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive sind langfristige Archivierungslösungen. Die minimale Speicherdauer beträgt 90 Tage für die Speicherklasse S3 Glacier Flexible Retrieval und 180 Tage für S3 Glacier Deep Archive. Beim Löschen von Daten, die in Amazon S3 Glacier archiviert wurden, fallen keine Gebühren an, wenn die von Ihnen gelöschten Objekte länger als die minimale Speicherdauer archiviert werden. Wenn Sie ein archiviertes Objekt innerhalb der minimalen Speicherdauer löschen oder überschreiben, stellt Amazon S3 eine anteilige Gebühr für das vorzeitige Löschen in Rechnung. Weitere Informationen zu den Kosten für die vorzeitige Löschungen finden Sie unter der Frage „Wie wird das Löschen von Objekten aus Amazon S3 Glacier berechnet, die weniger als 90 Tage alt sind?“ Frage zum Amazon S3 FAQ.
- Gebühren für Übergangsanfragen von S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive – Jedes Objekt, das Sie auf die Speicherklasse S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive umstellen, stellt eine Übergangsanfrage dar. Für jede dieser Anforderungen entstehen Kosten. Wenn Sie vorhaben, sehr viele Objekte zu überführen, sollten Sie die Anforderungskosten in Betracht ziehen. Wenn Sie eine Mischung von Objekten archivieren, die kleine Objekte enthält, insbesondere solche unter 128 KB, empfehlen wir, den Lifecycle-Objektgrößenfilter zu verwenden, um kleine Objekte aus Ihrer Umstellung herauszufiltern, um die Anforderungskosten zu senken.
- Datenwiederherstellungsgebühren für S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive – S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive wurden für die Langzeitarchivierung von Daten entwickelt, auf die Sie selten zugreifen. Informationen zu Gebühren für die Datenwiederherstellung finden Sie unter der Frage „Wie viel kostet das Abrufen von Daten aus

Amazon S3 Glacier?“ Frage zum Amazon S3 FAQ. Informationen zum Wiederherstellen von Daten aus Amazon S3 Glacier finden Sie unter Wiederherstellen eines archivierten Objekts.

Wenn Sie Objekte mithilfe der S3-Lebenszyklusverwaltung in Amazon S3 Glacier archivieren, führt Amazon S3 diese Objekte asynchron über. Es kann eine Verzögerung zwischen dem Übergangsdatum in der S3-Lebenszyklus-Konfigurationsregel und dem Datum des physischen Übergangs geben. Ihnen werden die Preise von Amazon S3 Glacier basierend auf dem in der Regel angegebenen Übergangsdatum berechnet. Weitere Informationen finden Sie im Abschnitt Amazon S3 Glacier von Amazon S3 FAQ.

Die Amazon-S3-Produktdetailseite enthält Preisinformationen und Beispielberechnungen für die Archivierung von Amazon-S3-Objekten. Weitere Informationen finden Sie unter den folgenden Themen:

- Wie werden die Speicherkosten für in Amazon S3 Glacier archivierte Amazon-S3-Objekte berechnet? auf dem Amazon S3 FAQ.
- „Wie wird mir das Löschen von Objekten aus Amazon S3 Glacier in Rechnung gestellt, die weniger als 90 Tage alt sind?“ auf dem Amazon S3 FAQ.
- „Was kostet das Abrufen von Daten aus Amazon S3 Glacier?“ auf dem Amazon S3 FAQ.
- Amazon S3 – Preise für die Speicherkosten der unterschiedlichen Speicherklassen.

Wiederherstellen archivierter Objekte

Auf archivierte Objekte kann nicht in Echtzeit zugegriffen werden. Sie müssen zuerst eine Anforderung zur Wiederherstellung initiieren und dann warten, bis eine temporäre Kopie des Objekts für die Dauer bereitgestellt wird, die Sie in der Anforderung angegeben haben. Nachdem Sie eine temporäre Kopie des wiederhergestellten Objekts erhalten haben, bleibt die Speicherklasse des Objekts S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive. (A) HeadObject oder GetObjectAPIDie Betriebsanforderung gibt S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive als Speicherklasse zurück.)

> **Note**
>
> Bei einer Wiederherstellung eines Archivs zahlen Sie sowohl für das Archiv (Tarif für S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive) als auch für die temporär

wiederhergestellte Kopie (S3-Standard-Speichertarif). Informationen zu Preisen finden Sie unter Amazon S3 – Preise.

Für den Zugriff auf ein kopiertes Objekt müssen Sie eine Anfrage zur Wiederherstellung initiieren. Das kann programmgesteuert oder über die Amazon-S3-Konsole erfolgen. Amazon S3 verarbeitet jeweils nur eine Anfrage pro Objekt. Weitere Informationen finden Sie unter Wiederherstellen eines archivierten Objekts.

Auslaufende Objekte

Sie können einer S3-Lifecycle-Konfiguration Übergangsaktionen hinzufügen, um Amazon S3 anzuweisen, Objekte am Ende ihrer Lebensdauer zu löschen. Wenn ein Objekt aufgrund seiner Lebenszykluskonfiguration das Ende seiner Lebensdauer erreicht, ergreift Amazon S3 eine `Expiration` Aktion, die darauf basiert, in welchem S3-Versionierungsstatus sich der Bucket befindet:

- Bucket ohne Versionsnummer — Amazon S3 stellt das Objekt zur Entfernung in die Warteschlange und entfernt es asynchron, wodurch das Objekt dauerhaft entfernt wird.
- Bucket mit aktiviertem Versioning – Wenn die aktuelle Objektversion keine Löschmarkierung ist, fügt Amazon S3 die Löschmarkierung mit einer eindeutigen Versions-ID hinzu. Damit ist die aktuelle Version nicht mehr aktuell und die Löschmarkierung wird zur aktuellen Version.
- Bucket mit ausgesetztem Versioning – Amazon S3 erstellt eine Löschmarkierung mit der Versions-ID null. Diese Löschmarkierung ersetzt jede Objektversion mi einer Versions-ID von null in der Versionshierarchie, womit das Objekt effektiv gelöscht wird.

Für einen versionierten Bucket (d. h. Versionierung aktiviert oder Version gesperrt) gibt es mehrere Überlegungen, die bestimmen, wie Amazon S3 mit der Aktion umgeht. `Expiration` Für Buckets mit aktivierter oder ausgesetzter Versionsverwaltung gilt Folgendes:

- Die Objektablaufaktion wird nur auf die aktuelle Version eines Objekts angewendet (sie wirkt sich nicht auf nicht aktuelle Objektversionen aus).
- Amazon S3 führt keine Aktion aus, wenn es eine oder mehrere Objektversionen gibt und die Löschmarkierung die aktuelle Version ist.
- Wenn die aktuelle Objektversion die einzige Objektversion und auch eine Löschmarkierung ist (auch als Löschmarkierung eines abgelaufenen Objekts bezeichnet, wobei alle Objektversionen gelöscht werden und nur noch eine Löschmarkierung beibehalten wird), entfernt Amazon S3 die

Löschmarkierung des abgelaufenen Objekts. Sie können die `Expiration` Aktion auch verwenden, um Amazon S3 anzuweisen, alle Markierungen zum Löschen abgelaufener Objekte zu entfernen. Ein Beispiel finden Sie unter Löschen von Markierungen für das Löschen abgelaufener Objekte in einem Bucket mit aktivierter Versionierung.

- Sie können das `NoncurrentVersionExpiration` Action-Element verwenden, um Amazon S3 anzuweisen, nicht aktuelle Versionen von Objekten dauerhaft zu löschen. Diese gelöschten Objekte können nicht wiederhergestellt werden. Sie können dieses Ablaufdatum auf eine bestimmte Anzahl von Tagen beziehen, seit die Objekte nicht mehr aktuell sind. Zusätzlich zur Anzahl der Tage können Sie auch eine maximale Anzahl an veralteten Versionen angeben, die aufbewahrt werden sollen (zwischen 1 und 100). Dieser Wert gibt die Anzahl der neueren nicht aktuellen Versionen an, die vorhanden sein müssen, damit Amazon S3 die zugehörige Aktion für eine bestimmte Version ausführen kann. Um die maximale Anzahl an nicht aktuellen Versionen anzugeben, müssen Sie auch ein `Filter` Element angeben. Wenn Sie kein `Filter` Element angeben, generiert Amazon S3 einen `InvalidRequest` Fehler, wenn Sie eine maximale Anzahl von nicht aktuellen Versionen angeben. Weitere Informationen zur Verwendung des `NoncurrentVersionExpiration` Action-Elements finden Sie unterthe section called “Elemente, die Lebenszyklus-Aktionen beschreiben”.
- Amazon S3 ergreift keine Maßnahmen für ältere Versionen von Objekten, auf die die S3 Object Lock-Konfiguration angewendet wurde.
- Bei Objekten mit einem `Pending` Replikationsstatus ergreift Amazon S3 keine Maßnahmen für aktuelle oder nicht aktuelle Versionen von Objekten.

Weitere Informationen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

> **Important**
>
> Wenn Sie mehrere Regeln in einer S3-Lifecycle-Konfiguration haben, kann ein Objekt für mehrere S3 Lifecycle-Aktionen am selben Tag in Frage kommen. In solchen Fällen folgt Amazon S3 diesen allgemeinen Regeln:
>
> - Das permanente Löschen hat Vorrang vor einem Übergang.
> - Der Übergang hat Vorrang vor der Erstellung von Löschmarkierungen.
> - Wenn ein Objekt sowohl für eine Umstellung auf S3 Glacier Flexible Retrieval als auch für eine Umstellung auf S3 Standard-IA (oder eine S3 One Zone-IA) in Frage kommt, entscheidet sich Amazon S3 für die Umstellung auf S3 Glacier Flexible Retrieval.

Beispiele finden Sie unter Beispiele für sich überschneidende Filter und widersprüchliche Lebenszyklusaktionen.

Bestehende und neue Objekte

Wenn Sie einem Bucket eine Lebenszyklus-Konfiguration hinzufügen, gelten die Konfigurationsregeln für vorhandene Objekte und für Objekte, die Sie später hinzufügen. Wenn Sie beispielsweise heute eine Lebenszyklus-Konfigurationsregel mit einer Ablaufaktion hinzufügen, die bewirkt, dass Objekte mit einem bestimmten Präfix 30 Tage nach ihrer Erstellung ablaufen, stellt Amazon S3 alle vorhandenen Objekte, die älter als 30 Tage sind und das angegebene Präfix haben, zur Entfernung in eine Warteschlange.

Important

Sie können keine Bucket-Richtlinie verwenden, um Löschungen oder Übergänge durch eine S3-Lebenszyklusregel zu verhindern. Selbst wenn Ihre Bucket-Richtlinie beispielsweise alle Aktionen für alle Principals ablehnt, funktioniert Ihre S3-Lifecycle-Konfiguration weiterhin wie gewohnt.

So finden Sie heraus, wann Objekte ablaufen

Um herauszufinden, wann die aktuelle Version eines Objekts ablaufen soll, verwenden Sie den HeadObject oder GetObjectAPIOperation. Diese API Operationen geben Antwortheader zurück, die das Datum und die Uhrzeit angeben, zu dem die aktuelle Version des Objekts nicht mehr zwischengespeichert werden kann.

Note

- Es kann eine Verzögerung zwischen dem Ablaufdatum und dem Datum geben, an dem Amazon S3 ein Objekt entfernt. Der Ablauf oder die mit einem abgelaufenen Objekt verbundene Speicherdauer wird Ihnen nicht in Rechnung gestellt.
- Verwenden Sie vor dem Aktualisieren, Deaktivieren oder Löschen von Lebenszyklusregeln die folgenden `LIST` API Operationen (z. B. ListObjectsV2, ListObjectVersions, und ListMultipartUploads) oder Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory

um zu überprüfen, ob Amazon S3 aufgrund Ihrer Anwendungsfälle umgestellt wurde und berechtigte Objekte abgelaufen sind.

Gebühren für Mindestspeicherdauer

Wenn Sie eine S3-Lebenszyklusablaufregel erstellen, mit der Objekte ablaufen, die sich seit weniger als 30 Tagen in den Speicherklassen S3 Standard-IA oder S3 One Zone-IA befinden, werden Ihnen Gebühren für 30 Tage in Rechnung gestellt. Wenn Sie eine Lebenszyklusablaufregel erstellen, mit der Objekte ablaufen, die weniger als 90 Tage im Speicher S3 Glacier Flexible Retrieval gewesen sind, werden Ihnen Gebühren für 90 Tage in Rechnung gestellt. Wenn Sie eine Lebenszyklusablaufregel erstellen, mit der Objekte ablaufen, die weniger als 180 Tage im S3 Glacier Deep Archive-Speicher gewesen sind, werden Ihnen Gebühren für 180 Tage in Rechnung gestellt.

Weitere Informationen finden Sie unter Amazon S3 – Preise.

Einstellung einer S3-Lifecycle-Konfiguration für einen Bucket

Sie können eine Amazon S3 Lifecycle-Konfiguration für einen Bucket einrichten, indem Sie die Amazon S3-Konsole, die AWS Command Line Interface (AWS CLI) AWS SDKs, die oder Amazon S3 verwenden RESTAPI. Weitere Informationen zur S3-Lebenszyklus-Konfiguration finden Sie unter Verwaltung des Lebenszyklus von Objekten.

In Ihrer S3-Lifecycle-Konfiguration verwenden Sie Lebenszyklusregeln, um Aktionen zu definieren, die Amazon S3 während der Lebensdauer eines Objekts ausführen soll. Sie können beispielsweise Regeln definieren, um Objekte in eine andere Speicherklasse zu überführen, Objekte zu archivieren oder Objekte nach einem bestimmten Zeitraum ablaufen (zu löschen).

Überlegungen zum S3-Lebenszyklus

Bevor Sie eine Lebenszyklus-Konfiguration festlegen, beachten Sie Folgendes:

Verzögerung bei der Übertragung der Lebenszykluskonfiguration

Wenn Sie eine S3-Lebenszyklus-Konfiguration zu einem Bucket hinzufügen, tritt eine gewisse Verzögerung ein, bis eine neue oder aktualisierte Lebenszyklus-Konfiguration vollständig auf alle Amazon-S3-Systeme verteilt ist. Rechnen Sie mit einer Verzögerung von einigen Minuten, bis die Konfiguration vollständig wirksam ist. Diese Verzögerung kann auch auftreten, wenn Sie eine S3-Lebenszyklus-Konfiguration löschen.

Verzögerung bei Übergang oder Ablauf

Es gibt eine Verzögerung zwischen dem Zeitpunkt, an dem eine Lebenszyklusregel erfüllt ist, und dem Abschluss der Aktion für die Regel. Nehmen wir beispielsweise an, dass eine Gruppe von Objekten aufgrund einer Lebenszyklusregel am 1. Januar abgelaufen ist. Obwohl die Ablaufregel am 1. Januar erfüllt wurde, löscht Amazon S3 diese Objekte möglicherweise erst Tage oder sogar Wochen später. Diese Verzögerung tritt auf, weil S3 Lifecycle Objekte für Übergänge oder Ablaufzeiten asynchron in die Warteschlange stellt. Änderungen an der Abrechnung werden jedoch normalerweise angewendet, wenn die Lebenszyklusregel erfüllt ist, auch wenn die Aktion noch nicht abgeschlossen ist. Weitere Informationen finden Sie unter Änderungen bei der Abrechnung. Informationen zur Überwachung der Auswirkungen von Aktualisierungen, die durch aktive Lebenszyklusregeln vorgenommen wurden, finden Sie unter the section called “Wie überwache ich die im Rahmen meiner Lebenszyklusregeln ergriffenen Maßnahmen?”

Lebenszyklusregeln aktualisieren, deaktivieren oder löschen

Wenn Sie Lebenszyklusregeln deaktivieren oder löschen, beendet Amazon S3 nach einer kleinen Verzögerung die Planung neuer Objekte für das Löschen oder den Übergang. Die Planung aller bereits geplanten Objekte wird aufgehoben und sie werden nicht gelöscht oder überführt.

> **Note**
>
> Bevor Sie Lebenszyklusregeln aktualisieren, deaktivieren oder löschen, verwenden Sie die folgenden `LIST` API Operationen (wie ListObjectsV2, ListObjectVersions, und ListMultipartUploads) oder Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory um zu überprüfen, ob Amazon S3 aufgrund Ihrer Anwendungsfälle umgestellt wurde und berechtigte Objekte abgelaufen sind. Falls Sie Probleme beim Aktualisieren, Deaktivieren oder Löschen von Lebenszyklusregeln haben, finden Sie weitere Informationen unter. Behebung von Problemen mit Amazon S3 Lifecycle

Bestehende und neue Objekte

Wenn Sie einem Bucket eine Lebenszyklus-Konfiguration hinzufügen, gelten die Konfigurationsregeln für vorhandene Objekte und für Objekte, die Sie später hinzufügen. Wenn Sie beispielsweise heute eine Lebenszyklus-Konfigurationsregel mit einer Ablaufaktion hinzufügen, die bewirkt, dass Objekte mit einem bestimmten Präfix 30 Tage nach ihrer Erstellung ablaufen, stellt Amazon S3 eine Warteschlange für die Entfernung aller vorhandenen Objekte, die älter als 30 Tage sind und das angegebene Präfix haben, in die Warteschlange.

Überwachung der Auswirkungen von Lebenszyklusregeln

Informationen zur Überwachung der Auswirkungen von Aktualisierungen, die durch aktive Lebenszyklusregeln vorgenommen wurden, finden Sie unter the section called “Wie überwache ich die im Rahmen meiner Lebenszyklusregeln ergriffenen Maßnahmen?”

Änderungen bei der Abrechnung

Möglicherweise besteht eine Verzögerung zwischen dem Zeitpunkt, an dem die Lebenszyklus-Konfigurationsregeln erfüllt sind, und dem Zeitpunkt, zu dem die Aktion ausgeführt wird, die durch die Erfüllung der Regel ausgelöst wurde. Änderungen an der Abrechnung werden jedoch vorgenommen, sobald die Lifecycle-Konfigurationsregel erfüllt ist, auch wenn die Maßnahme noch nicht ergriffen wurde.

Beispielsweise wird Ihnen nach Ablauf des Objektablaufs kein Speicherplatz in Rechnung gestellt, auch wenn das Objekt nicht sofort gelöscht wird. Ebenso werden Ihnen nach Ablauf der Objektübergangszeit die Speichertarife für S3 Glacier Flexible Retrieval berechnet, auch wenn das Objekt nicht sofort in die Speicherklasse S3 Glacier Flexible Retrieval umgestellt wird.

Lebenszyklusübergänge zur Speicherklasse S3 Intelligent-Tiering sind jedoch die Ausnahme. Änderungen an der Abrechnung werden erst vorgenommen, nachdem das Objekt in die S3 Intelligent-Tiering-Speicherklasse übergegangen ist.

Mehrere oder widersprüchliche Regeln

Wenn Sie mehrere Regeln in einer S3 Lifecycle-Konfiguration haben, kann ein Objekt am selben Tag für mehrere S3 Lifecycle-Aktionen in Frage kommen. In solchen Fällen folgt Amazon S3 diesen allgemeinen Regeln:

- Das permanente Löschen hat Vorrang vor einem Übergang.
- Der Übergang hat Vorrang vor der Erstellung von Löschmarkierungen.
- Wenn ein Objekt sowohl für eine Umstellung auf S3 Glacier Flexible Retrieval als auch für eine Umstellung auf S3 Standard-IA (oder eine S3 One Zone-IA) in Frage kommt, entscheidet sich Amazon S3 für die Umstellung auf S3 Glacier Flexible Retrieval.

Beispiele finden Sie unter Beispiele für sich überschneidende Filter und widersprüchliche Lebenszyklusaktionen.

Wie richte ich eine S3-Lifecycle-Konfiguration ein

Sie können eine Amazon S3 Lifecycle-Konfiguration für einen Bucket einrichten, indem Sie die Amazon S3-Konsole, die AWS Command Line Interface (AWS CLI) AWS SDKs, die oder Amazon S3 verwenden RESTAPI.

Informationen zu AWS CloudFormation Vorlagen und Beispielen finden Sie unter Arbeiten mit AWS CloudFormation Vorlagen und AWS::S3::Bucket im AWS CloudFormation -Benutzerhandbuch.

Verwenden der S3-Konsole

Sie können Lebenszyklusregeln für alle Objekte oder eine Teilmenge von Objekten in einem Bucket definieren, indem Sie ein gemeinsames Präfix (Objektnamen, die mit einer gemeinsamen Zeichenfolge beginnen) oder ein Tag verwenden. In Ihrer Lebenszyklusregel können Sie Aktionen definieren, die für aktuelle und nicht aktuelle Objektversionen spezifisch sind. Weitere Informationen finden Sie hier:

- Verwaltung des Lebenszyklus von Objekten
- Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning

So erstellen Sie eine Lebenszyklusregel:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie eine Lebenszyklusregel erstellen möchten.
3. Wählen Sie den Tab Management (Verwaltung) und dann die Option Create lifecycle rule (Lebenszyklusregel erstellen).
4. Geben Sie unter Lifecycle rule name (Name der Lebenszyklusregel) einen Namen für Ihre Regel ein.

   Der Name muss innerhalb des Buckets eindeutig sein.
5. Wählen Sie den Umfang der Lebenszyklusregel:

   - Um diese Lebenszyklusregel auf alle Objekte mit einem bestimmten Präfix oder Tag anzuwenden, wählen Sie Umfang auf bestimmte Präfixe oder Markierungen beschränken aus.
     - Um den Bereich nach Präfix zu beschränken, geben Sie unter Prefix (Präfix) das Präfix ein.

- Um den Bereich nach Tag einzuschränken, wählen Sie Add tag (Tag hinzufügen), und geben Sie den Tag-Schlüssel und den Wert ein.

Weitere Hinweise zu Präfixen für Objektnamen finden Sie unter Benennen Amazon S3 S3-Objekten. Weitere Informationen über Objekt-Markierungen finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen.

- Um diese Lebenszyklusregel auf alle Objekte im Bucket anzuwenden, wählen Sie Diese Regel gilt für alle Objekte im Bucket und anschließend Ich bestätige, dass diese Regel für alle Objekte im Bucket gilt.

6. Um eine Regel nach Objektgröße zu filtern, können Sie „Minimale Objektgröße angeben“, „Maximale Objektgröße angeben“ oder beide Optionen auswählen.

   - Wenn Sie einen Wert für Minimale Objektgröße oder Maximale Objektgröße angeben, muss der Wert größer als 0 Byte und bis zu 5 TB sein. Sie können diesen Wert in Byte, KB, MB oder GB angeben.
   - Wenn Sie beide Werte angeben, muss die maximale Objektgröße größer als die minimale Objektgröße sein.

   > **Note**
   >
   > Die Filter Minimale Objektgröße und Maximale Objektgröße schließen die angegebenen Werte aus. Wenn Sie beispielsweise einen Filter so einrichten, dass Objekte mit einer Mindestobjektgröße von 128 KB ablaufen, laufen Objekte, die genau 128 KB groß sind, nicht ab. Stattdessen gilt die Regel nur für Objekte mit einer Größe von mehr als 128 KB.

7. Wählen Sie unter Lifecycle rule actions (Lebenszyklusregelaktionen) die Aktionen aus, die Ihre Lebenszyklusregel ausführen soll:

   - Umsetzung aktueller Versionen von Objekten zwischen Speicherklassen
   - Umsetzung früherer Versionen von Objekten zwischen Speicherklassen
   - Ablauf aktueller Versionen von Objekten

   > **Note**
   >
   > Bei Buckets, für die die S3-Versionierung nicht aktiviert ist, führt das Ablaufen aktueller Versionen dazu, dass Amazon S3 die Objekte dauerhaft löscht. Weitere Informationen

finden Sie unter the section called “Lebenszyklus-Aktionen und der Versioning-Status eines Buckets”.

- Dauerhaftes Löschen früherer Versionen von Objekten
- Löschen abgelaufener Löschmarkierungen oder unvollständiger mehrteiliger Uploads

Abhängig von den von Ihnen ausgewählten Aktionen werden verschiedene Optionen angezeigt.

8. Um aktuelle Versionen von Objekten zwischen Speicherklassen zu übertragen, gehen Sie unter Aktuelle Versionen von Objekten zwischen Speicherklassen übertragen wie folgt vor:

   a. Wählen Sie unter Speicherklassenübergänge die Speicherklasse aus, zu der der Übergang erfolgen soll. Eine Liste möglicher Übergänge finden Sie unterthe section called “Unterstützte Lebenszyklusübergänge”. Sie können aus den folgenden Speicherklassen wählen:

      - S3 Standard-IA
      - S3 Intelligent-Tiering
      - S3 One Zone-IA
      - S3 Glacier Instant Retrieval
      - S3 Glacier Flexible Retrieval
      - S3 Glacier Deep Archive

   b. Geben Sie im Feld Days after object creation (Tage nach der Objekterstellung) die Anzahl der Tage nach der Erstellung für die Umsetzung des Objekts ein.

   Weitere Informationen über Speicherklassen finden Sie unter Amazon S3 S3-Speicherklassen verstehen und verwalten. Sie können Übergänge für aktuelle oder vorhergehende Objektversionen definieren oder sowohl für aktuelle als auch vorhergehende Versionen. Versioning ermöglicht Ihnen, mehrere Versionen eines Objekts in einem Bucket aufzubewahren. Weitere Informationen über die Versionssteuerung finden Sie unter Verwenden der S3-Konsole.

   > **Important**
   >
   > Wenn Sie die Speicherklasse S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval oder Glacier Deep Archive wählen, verbleiben Ihre Objekte in Amazon S3. Sie können

nicht direkt über den separaten Amazon-S3-Glacier-Service auf sie zugreifen. Weitere Informationen finden Sie unter Übergang von Objekten mit Amazon-S3-Lebenszyklus.

9. Um ältere Versionen von Objekten zwischen Speicherklassen zu übertragen, gehen Sie unter Übergang von nicht aktuellen Versionen von Objekten zwischen Speicherklassen wie folgt vor:

   a. Wählen Sie unter Speicherklassenübergänge die Speicherklasse aus, zu der der Übergang erfolgen soll. Eine Liste möglicher Übergänge finden Sie unterthe section called “Unterstützte Lebenszyklusübergänge”. Sie können aus den folgenden Speicherklassen wählen:

      - S3 Standard-IA
      - S3 Intelligent-Tiering
      - S3 One Zone-IA
      - S3 Glacier Instant Retrieval
      - S3 Glacier Flexible Retrieval
      - S3 Glacier Deep Archive

   b. Geben Sie im Feld Tage, nachdem das Objekt nicht mehr aktuell ist, die Anzahl der Tage nach der Erstellung ein, um das Objekt umzustellen.

10. Um den Ablauf aktueller Versionen von Objekten auszulösen, geben Sie unter Expire current versions of objects (Ablauf aktueller Objektversionen) in Number of days after object creation (Anzahl der Tage nach der Objekterstellung) die Anzahl der Tage ein.

    Important

    In einem Bucket ohne Versionierung führt die Ablaufaktion dazu, dass Amazon S3 das Objekt dauerhaft entfernt. Weitere Informationen über die Lebenszyklus-Aktionen finden Sie unter Elemente, die Lebenszyklus-Aktionen beschreiben.

11. Um frühere Versionen von Objekten dauerhaft zu löschen, geben Sie unter Permanently delete previous versions of objects (Dauerhaftes Löschen nicht aktueller Versionen von Objekten) in Days after objects become noncurrent (Tage nach dem Erlöschen von Objekten) die Anzahl der Tage ein. Sie können die Anzahl der beizubehaltenden neueren Versionen optional angeben, indem Sie einen Wert unter Number of newer versions to retain (Anzahl der beizubehaltenden neueren Versionen) eingeben.

12. Wählen Sie unter Delete expired delete markers or incomplete multipart uploads (Abgelaufene Löschmarkierungen oder unvollständige mehrteilige Uploads löschen) Delete expired object delete markers (Abgelaufene Objektlöschmarken löschen) und Delete incomplete multipart uploads (Unvollständige mehrteilige Uploads löschen) aus. Geben Sie dann die Anzahl der Tage nach der mehrteiligen Upload-Initiierung ein, die Sie beenden und für die Sie unvollständige mehrteilige Uploads löschen möchten.

    Weitere Informationen über mehrteilige Uploads finden Sie unter Hochladen und Kopieren von Objekten mit mehrteiligen Uploads.

13. Wählen Sie Create rule aus.

    Wenn die Regel keine Fehler enthält, aktiviert Amazon S3 sie, und Sie können sie im Tab Management (Verwaltung) unter Lifecycle rules (Lebenszyklusregeln) sehen.

Mit dem AWS CLI

Sie können die folgenden AWS CLI Befehle verwenden, um S3 Lifecycle-Konfigurationen zu verwalten:

- `put-bucket-lifecycle-configuration`
- `get-bucket-lifecycle-configuration`
- `delete-bucket-lifecycle`

Anweisungen zur Einrichtung von finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS CLI in der Amazon S3 API S3-Referenz. AWS CLI

Die Amazon S3 Lifecycle-Konfiguration ist eine XML Datei. Aber wenn Sie die verwenden AWS CLI, können Sie das XML Format nicht angeben. Sie müssen stattdessen das JSON Format angeben. Im Folgenden finden Sie Beispiele für XML Lebenszykluskonfigurationen und die entsprechenden JSON Konfigurationen, die Sie in einem AWS CLI Befehl angeben können.

Betrachten Sie das folgende Beispiel einer S3-Lebenszyklus-Konfiguration:

Example Beispiel 1

Example

XML

```
<LifecycleConfiguration>
    <Rule>
        <ID>ExampleRule</ID>
        <Filter>
           <Prefix>documents/</Prefix>
        </Filter>
        <Status>Enabled</Status>
        <Transition>
           <Days>365</Days>
           <StorageClass>GLACIER</StorageClass>
        </Transition>
        <Expiration>
             <Days>3650</Days>
        </Expiration>
    </Rule>
</LifecycleConfiguration>
```

JSON

```
{
    "Rules": [
        {
            "Filter": {
                "Prefix": "documents/"
            },
            "Status": "Enabled",
            "Transitions": [
                {
                    "Days": 365,
                    "StorageClass": "GLACIER"
                }
            ],
            "Expiration": {
                "Days": 3650
            },
            "ID": "ExampleRule"
```

```
        }
    ]
        }
```

Example Beispiel 2

Example

XML

```
<LifecycleConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
    <Rule>
        <ID>id-1</ID>
        <Expiration>
            <Days>1</Days>
        </Expiration>
        <Filter>
            <And>
                <Prefix>myprefix</Prefix>
                <Tag>
                    <Key>mytagkey1</Key>
                    <Value>mytagvalue1</Value>
                </Tag>
                <Tag>
                    <Key>mytagkey2</Key>
                    <Value>mytagvalue2</Value>
                </Tag>
            </And>
        </Filter>
        <Status>Enabled</Status>
    </Rule>
</LifecycleConfiguration>
```

JSON

```
{
    "Rules": [
        {
            "ID": "id-1",
            "Filter": {
                "And": {
```

```
                    "Prefix": "myprefix",
                    "Tags": [
                        {
                            "Value": "mytagvalue1",
                            "Key": "mytagkey1"
                        },
                        {
                            "Value": "mytagvalue2",
                            "Key": "mytagkey2"
                        }
                    ]
                }
            },
            "Status": "Enabled",
            "Expiration": {
                "Days": 1
            }
        }
    ]
}
```

Sie können den Befehl `put-bucket-lifecycle-configuration` wie folgt testen.

So testen Sie die Konfiguration

1. Speichern Sie die JSON Lifecycle-Konfiguration in einer Datei (z. B. *`life-cycle.json`*).
2. Führen Sie den folgenden AWS CLI Befehl aus, um die Lifecycle-Konfiguration für Ihren Bucket festzulegen. Ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

   ```
   $ aws s3api put-bucket-lifecycle-configuration  \
   --bucket amzn-s3-demo-bucket  \
   --lifecycle-configuration file://lifecycle.json
   ```

3. Rufen Sie zur Überprüfung die S3-Lifecycle-Konfiguration mit dem folgenden `get-bucket-lifecycle-configuration` AWS CLI Befehl ab:

   ```
   $ aws s3api get-bucket-lifecycle-configuration  \
   --bucket amzn-s3-demo-bucket
   ```

4. Verwenden Sie den `delete-bucket-lifecycle` AWS CLI Befehl wie folgt, um die S3 Lifecycle-Konfiguration zu löschen:

```
aws s3api delete-bucket-lifecycle \
--bucket amzn-s3-demo-bucket
```

Mit dem AWS SDKs

Java

Sie können den verwenden AWS SDK for Java , um die S3-Lifecycle-Konfiguration eines Buckets zu verwalten. Weitere Informationen zur Verwaltung einer S3-Lebenszyklus-Konfiguration finden Sie unter Verwaltung des Lebenszyklus von Objekten.

> **Note**
>
> Wenn Sie einem Bucket eine S3-Lifecycle-Konfiguration hinzufügen, ersetzt Amazon S3 die aktuelle Lifecycle-Konfiguration des Buckets, falls es eine gibt. Um eine Lebenszyklus-Konfiguration zu aktualisieren, rufen Sie sie ab, nehmen die gewünschten Änderungen vor und fügen die geänderte Konfiguration dann dem Bucket hinzu.

Das folgende Beispiel zeigt, wie Sie die AWS SDK for Java Lifecycle-Konfiguration eines Buckets hinzufügen, aktualisieren und löschen können. Das Beispiel erledigt Folgendes:

- Fügt eine Lebenszyklus-Konfiguration zu einem Bucket hinzu.
- Ruft die Lebenszyklus-Konfiguration und Updates durch Hinzufügen einer weiteren Regel ab.
- Fügt die abgeänderte Lebenszyklus-Konfiguration dem Bucket hinzu. Amazon S3 ersetzt die vorhandene Konfiguration.
- Ruft die Konfiguration erneut ab und überprüft, ob sie die richtige Anzahl von Regeln enthält, indem die Anzahl der Regeln gedruckt wird.
- Löscht die Lebenszyklus-Konfiguration und überprüft, ob sie gelöscht wurde, indem versucht wird, sie erneut abzurufen.

Anweisungen zum Erstellen und Testen eines Arbeitsbeispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
```

```
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketLifecycleConfiguration;
import com.amazonaws.services.s3.model.BucketLifecycleConfiguration.Transition;
import com.amazonaws.services.s3.model.StorageClass;
import com.amazonaws.services.s3.model.Tag;
import com.amazonaws.services.s3.model.lifecycle.LifecycleAndOperator;
import com.amazonaws.services.s3.model.lifecycle.LifecycleFilter;
import com.amazonaws.services.s3.model.lifecycle.LifecyclePrefixPredicate;
import com.amazonaws.services.s3.model.lifecycle.LifecycleTagPredicate;

import java.io.IOException;
import java.util.Arrays;

public class LifecycleConfiguration {

        public static void main(String[] args) throws IOException {
                Regions clientRegion = Regions.DEFAULT_REGION;
                String bucketName = "*** Bucket name ***";

                // Create a rule to archive objects with the "glacierobjects/"
 prefix to Glacier
                // immediately.
                BucketLifecycleConfiguration.Rule rule1 = new
 BucketLifecycleConfiguration.Rule()
                                .withId("Archive immediately rule")
                                .withFilter(new LifecycleFilter(new
 LifecyclePrefixPredicate("glacierobjects/")))
                                .addTransition(new
 Transition().withDays(0).withStorageClass(StorageClass.Glacier))
                                .withStatus(BucketLifecycleConfiguration.ENABLED);

                // Create a rule to transition objects to the Standard-Infrequent
 Access storage
                // class
                // after 30 days, then to Glacier after 365 days. Amazon S3 will
 delete the
                // objects after 3650 days.
                // The rule applies to all objects with the tag "archive" set to
 "true".
```

```
                BucketLifecycleConfiguration.Rule rule2 = new
BucketLifecycleConfiguration.Rule()
                                .withId("Archive and then delete rule")
                                .withFilter(new LifecycleFilter(new
LifecycleTagPredicate(new Tag("archive", "true"))))
                                .addTransition(new Transition().withDays(30)

.withStorageClass(StorageClass.StandardInfrequentAccess))
                                .addTransition(new
Transition().withDays(365).withStorageClass(StorageClass.Glacier))
                                .withExpirationInDays(3650)
                                .withStatus(BucketLifecycleConfiguration.ENABLED);

                // Add the rules to a new BucketLifecycleConfiguration.
                BucketLifecycleConfiguration configuration = new
BucketLifecycleConfiguration()
                                .withRules(Arrays.asList(rule1, rule2));

                try {
                        AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                                        .withCredentials(new
ProfileCredentialsProvider())
                                        .withRegion(clientRegion)
                                        .build();

                        // Save the configuration.
                        s3Client.setBucketLifecycleConfiguration(bucketName,
configuration);

                        // Retrieve the configuration.
                        configuration =
s3Client.getBucketLifecycleConfiguration(bucketName);

                        // Add a new rule with both a prefix predicate and a tag
predicate.
                        configuration.getRules().add(new
BucketLifecycleConfiguration.Rule().withId("NewRule")
                                        .withFilter(new LifecycleFilter(new
LifecycleAndOperator(
                                                        Arrays.asList(new
LifecyclePrefixPredicate("YearlyDocuments/"),
                                                                        new
LifecycleTagPredicate(new Tag(
```

```
   "expire_after",

   "ten_years"))))))
                                        .withExpirationInDays(3650)

.withStatus(BucketLifecycleConfiguration.ENABLED));

                        // Save the configuration.
                        s3Client.setBucketLifecycleConfiguration(bucketName,
configuration);

                        // Retrieve the configuration.
                        configuration =
s3Client.getBucketLifecycleConfiguration(bucketName);

                        // Verify that the configuration now has three rules.
                        configuration =
s3Client.getBucketLifecycleConfiguration(bucketName);
                        System.out.println("Expected # of rules = 3; found: " +
configuration.getRules().size());

                        // Delete the configuration.
                        s3Client.deleteBucketLifecycleConfiguration(bucketName);

                        // Verify that the configuration has been deleted by
attempting to retrieve it.
                        configuration =
s3Client.getBucketLifecycleConfiguration(bucketName);
                        String s = (configuration == null) ? "No configuration
found." : "Configuration found.";
                        System.out.println(s);
                } catch (AmazonServiceException e) {
                        // The call was transmitted successfully, but Amazon S3
couldn't process
                        // it, so it returned an error response.
                        e.printStackTrace();
                } catch (SdkClientException e) {
                        // Amazon S3 couldn't be contacted for a response, or the
client
                        // couldn't parse the response from Amazon S3.
                        e.printStackTrace();
                }
        }
```

```
}
```

.NET

Sie können den verwenden AWS SDK for .NET , um die S3-Lifecycle-Konfiguration in einem Bucket zu verwalten. Weitere Informationen zur Verwaltung einer Lebenszyklus-Konfiguration finden Sie unter Verwaltung des Lebenszyklus von Objekten.

> **Note**
>
> Wenn Sie eine Lebenszyklus-Konfiguration hinzufügen, ersetzt Amazon S3 die vorhandene Konfiguration für den angegebenen Bucket. Um eine Konfiguration zu aktualisieren, müssen Sie zuerst die Lebenszyklus-Konfiguration abrufen, die Änderungen vornehmen und dann die geänderte Lebenszyklus-Konfiguration dem Bucket hinzufügen.

Das folgende Beispiel zeigt, wie Sie die AWS SDK for .NET Lifecycle-Konfiguration eines Buckets hinzufügen, aktualisieren und löschen können. Das Codebeispiel führt die folgenden Aufgaben durch:

- Fügt eine Lebenszyklus-Konfiguration zu einem Bucket hinzu.
- Ruft die Lebenszyklus-Konfiguration und Updates durch Hinzufügen einer weiteren Regel ab.
- Fügt die abgeänderte Lebenszyklus-Konfiguration dem Bucket hinzu. Amazon S3 ersetzt die vorhandene Lebenszyklus-Konfiguration.
- Ruft die Konfiguration erneut ab und überprüft sie durch Ausgabe der Anzahl von Regeln in der Konfiguration.
- Löscht die Lebenszyklus-Konfiguration und überprüft den Löschvorgang.

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Getting Started with the AWS SDK for .NET im AWS SDK for .NET Entwicklerhandbuch.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;
```

```
namespace Amazon.DocSamples.S3
{
    class LifecycleTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;
        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            AddUpdateDeleteLifecycleConfigAsync().Wait();
        }

        private static async Task AddUpdateDeleteLifecycleConfigAsync()
        {
            try
            {
                var lifeCycleConfiguration = new LifecycleConfiguration()
                {
                    Rules = new List<LifecycleRule>
                        {
                            new LifecycleRule
                            {
                                 Id = "Archive immediately rule",
                                 Filter = new LifecycleFilter()
                                 {
                                     LifecycleFilterPredicate = new
 LifecyclePrefixPredicate()
                                     {
                                         Prefix = "glacierobjects/"
                                     }
                                 },
                                 Status = LifecycleRuleStatus.Enabled,
                                 Transitions = new List<LifecycleTransition>
                                 {
                                      new LifecycleTransition
                                      {
                                           Days = 0,
                                           StorageClass = S3StorageClass.Glacier
                                      }
                                  },
```

```
                            },
                            new LifecycleRule
                            {
                                 Id = "Archive and then delete rule",
                                  Filter = new LifecycleFilter()
                                 {
                                     LifecycleFilterPredicate = new
LifecyclePrefixPredicate()
                                     {
                                         Prefix = "projectdocs/"
                                     }
                                 },
                                 Status = LifecycleRuleStatus.Enabled,
                                 Transitions = new List<LifecycleTransition>
                                 {
                                      new LifecycleTransition
                                      {
                                           Days = 30,
                                           StorageClass =
S3StorageClass.StandardInfrequentAccess
                                      },
                                      new LifecycleTransition
                                      {
                                        Days = 365,
                                        StorageClass = S3StorageClass.Glacier
                                      }
                                 },
                                 Expiration = new LifecycleRuleExpiration()
                                 {
                                       Days = 3650
                                 }
                            }
                        }
                };

                // Add the configuration to the bucket.
                await AddExampleLifecycleConfigAsync(client,
lifeCycleConfiguration);

                // Retrieve an existing configuration.
                lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);

                // Add a new rule.
                lifeCycleConfiguration.Rules.Add(new LifecycleRule
```

```
                {
                    Id = "NewRule",
                    Filter = new LifecycleFilter()
                    {
                        LifecycleFilterPredicate = new LifecyclePrefixPredicate()
                        {
                            Prefix = "YearlyDocuments/"
                        }
                    },
                    Expiration = new LifecycleRuleExpiration()
                    {
                        Days = 3650
                    }
                });

                // Add the configuration to the bucket.
                await AddExampleLifecycleConfigAsync(client,
lifeCycleConfiguration);

                // Verify that there are now three rules.
                lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);
                Console.WriteLine("Expected # of rulest=3; found:{0}",
lifeCycleConfiguration.Rules.Count);

                // Delete the configuration.
                await RemoveLifecycleConfigAsync(client);

                // Retrieve a nonexistent configuration.
                lifeCycleConfiguration = await RetrieveLifecycleConfigAsync(client);

            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered ***. Message:'{0}' when writing
an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }
```

```
        static async Task AddExampleLifecycleConfigAsync(IAmazonS3 client,
LifecycleConfiguration configuration)
        {

            PutLifecycleConfigurationRequest request = new
PutLifecycleConfigurationRequest
            {
                BucketName = bucketName,
                Configuration = configuration
            };
            var response = await client.PutLifecycleConfigurationAsync(request);
        }

        static async Task<LifecycleConfiguration>
RetrieveLifecycleConfigAsync(IAmazonS3 client)
        {
            GetLifecycleConfigurationRequest request = new
GetLifecycleConfigurationRequest
            {
                BucketName = bucketName
            };
            var response = await client.GetLifecycleConfigurationAsync(request);
            var configuration = response.Configuration;
            return configuration;
        }

        static async Task RemoveLifecycleConfigAsync(IAmazonS3 client)
        {
            DeleteLifecycleConfigurationRequest request = new
DeleteLifecycleConfigurationRequest
            {
                BucketName = bucketName
            };
            await client.DeleteLifecycleConfigurationAsync(request);
        }
    }
}
```

Ruby

Sie können die verwenden AWS SDK for Ruby , um eine S3-Lifecycle-Konfiguration in einem Bucket zu verwalten, indem Sie die Klasse verwenden AWS::S3::BucketLifecycleConfiguration.

Weitere Informationen zur Verwaltung der S3 Lifecycle-Konfiguration finden Sie unterVerwaltung des Lebenszyklus von Objekten.

Mit dem REST API

In den folgenden Themen in der Amazon Simple Storage Service API Reference werden die REST API Vorgänge im Zusammenhang mit der S3 Lifecycle-Konfiguration beschrieben:

- PutBucketLifecycleConfiguration
- GetBucketLifecycleConfiguration
- DeleteBucketLifecycle

Fehlerbehebung bei S3 Lifecycle

Informationen zu häufigen Problemen, die bei der Arbeit mit S3 Lifecycle auftreten können, finden Sie unterthe section called “Behebung von Problemen im Lebenszyklus”.

Wie S3 Lifecycle mit anderen Bucket-Konfigurationen interagiert

Neben den S3-Lebenszyklus-Konfigurationen können Sie Ihrem Bucket auch weitere Konfigurationen zuordnen. In diesem Abschnitt wird erläutert, wie sich die S3-Lebenszyklus-Konfiguration auf andere Bucket-Konfigurationen auswirkt.

S3-Lebenszyklus und S3-Versionierung

Sie können nicht versionsfähigen Buckets und versionsfähigen Buckets S3-Lebenszyklus-Konfigurationen hinzufügen. Weitere Informationen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

Ein versionsfähiger Bucket behält eine aktuelle Objektversion sowie und null oder mehr langfristige Objektversionen bei. Sie können separate Lebenszyklusrichtlinie für aktuelle und nicht aktuelle Objektversionen definieren.

Weitere Informationen finden Sie unter Elemente der Lebenszyklus-Konfiguration.

Important

Wenn Sie mehrere Regeln in einer S3 Lifecycle-Konfiguration haben, kann ein Objekt am selben Tag für mehrere S3 Lifecycle-Aktionen in Frage kommen. In solchen Fällen folgt Amazon S3 diesen allgemeinen Regeln:

- Das permanente Löschen hat Vorrang vor einem Übergang.
- Der Übergang hat Vorrang vor der Erstellung von Löschmarkierungen.
- Wenn ein Objekt sowohl für einen S3 Glacier Flexible Retrieval als auch für einen S3 Standard-IA (oder S3 One Zone-IA) -Übergang in Frage kommt, entscheidet sich Amazon S3 für den Übergang zu S3 Glacier Flexible Retrieval.

Beispiele finden Sie unter Beispiele für sich überschneidende Filter und widersprüchliche Lebenszyklusaktionen.

S3-Lifecycle-Konfiguration für MFA -aktivierte Buckets

Die S3-Lifecycle-Konfiguration für Buckets mit Multi-Faktor-Authentifizierung (MFA) wird nicht unterstützt.

S3-Lebenszyklus und Protokollierung

Amazon S3 Lifecycle-Aktionen werden nicht durch die Protokollierung auf AWS CloudTrail Objektebene erfasst. CloudTrailerfasst API Anfragen an externe Amazon S3 S3-Endpunkte, wohingegen S3 Lifecycle-Aktionen mithilfe interner Amazon S3 S3-Endpunkte ausgeführt werden.

Sie können Amazon S3 S3-Serverzugriffsprotokolle in einem S3-Bucket aktivieren, um S3-Lifecycle-bezogene Aktionen zu erfassen, wie Objektübergänge in eine andere Speicherklasse und Objektablaufzeiten, die zu einer dauerhaften oder logischen Löschung führen. Weitere Informationen finden Sie unter the section called “Protokollierungs-Serverzugriff”.

Wenn die Protokollierung für Ihren Bucket aktiviert ist, melden Amazon-S3-Server-Zugriffsprotokolle die Ergebnisse der folgenden Vorgänge.

Operationsprotokoll	Beschreibung
`S3.EXPIRE.OBJECT`	Amazon S3 löscht das Objekt aufgrund der `Expiration` Lebenszyklusaktion dauerhaft.
`S3.CREATE.DELETEMARKER`	Amazon S3 löscht die aktuelle Version logisch, indem es eine Löschmarkierung in einem Bucket mit aktivierter Versionierung hinzufügt.
`S3.TRANSITION_SIA.OBJECT`	Amazon S3 überträgt das Objekt in die Speicherklasse S3 Standard-IA.
`S3.TRANSITION_ZIA.OBJECT`	Amazon S3 überträgt das Objekt in die Speicherklasse S3 One Zone-IA.
`S3.TRANSITION_INT.OBJECT`	Amazon S3 überträgt das Objekt in die Speicherklasse S3 Intelligent-Tiering.
`S3.TRANSITION_GIR.OBJECT`	Amazon S3 initiiert den Übergang des Objekts in die Speicherklasse S3 Glacier Instant Retrieval.
`S3.TRANSITION.OBJECT`	Amazon S3 initiiert den Übergang des Objekts zur Speicherklasse S3 Glacier Flexible Retrieval.
`S3.TRANSITION_GDA.OBJECT`	Amazon S3 initiiert den Übergang des Objekts zur Speicherklasse S3 Glacier Deep Archive.
`S3.DELETE.UPLOAD`	Amazon S3 bricht einen unvollständigen mehrteiligen Upload ab.

Note

Die Aufzeichnungen der Amazon S3 S3-Serverzugriffsprotokolle werden nach bestem Wissen und Gewissen bereitgestellt und können nicht für eine vollständige Abrechnung aller Amazon S3 S3-Anfragen verwendet werden.

Konfiguration von S3 Lifecycle-Ereignisbenachrichtigungen

Um eine Benachrichtigung zu erhalten, wenn Amazon S3 aufgrund der Einhaltung einer S3-Lebenszyklusregel ein Objekt löscht oder es in eine andere Amazon S3-Speicherklasse überführt, können Sie eine Amazon S3 S3-Ereignisbenachrichtigung einrichten.

Sie können Benachrichtigungen für die folgenden S3 Lifecycle-Ereignisse erhalten:

- Übergangsereignisse — Mithilfe des `s3:LifecycleTransition` Ereignistyps können Sie eine Benachrichtigung erhalten, wenn ein Objekt durch eine S3-Lifecycle-Konfiguration von einer Amazon S3-Speicherklasse in eine andere übertragen wird.
- Ablaufereignisse (Lösch-) Ereignisse — Mithilfe der `LifecycleExpiration` Ereignistypen können Sie Benachrichtigungen erhalten, wenn Amazon S3 ein Objekt auf der Grundlage Ihrer S3-Lifecycle-Konfiguration löscht.

  Es gibt zwei Arten von Ablaufereignissen:

  - Der `s3:LifecycleExpiration:Delete` Ereignistyp benachrichtigt Sie, wenn ein Objekt in einem Bucket ohne Version gelöscht wird. `s3:LifecycleExpiration:Delete`benachrichtigt Sie auch, wenn eine Objektversion durch eine S3-Lifecycle-Konfiguration dauerhaft gelöscht wird.
  - Der `s3:LifecycleExpiration:DeleteMarkerCreated` Ereignistyp benachrichtigt Sie, wenn S3 Lifecycle eine Löschmarkierung erstellt, nachdem eine aktuelle Version eines Objekts in einem versionierten Bucket gelöscht wurde. Weitere Informationen finden Sie unter the section called “Löschen von Objektversionen”.

Amazon S3 kann Ereignisbenachrichtigungen zu einem Amazon Simple Notification Service (AmazonSNS) -Thema, einer Amazon Simple Queue Service (AmazonSQS) -Warteschlange oder einer AWS Lambda Funktion veröffentlichen. Weitere Informationen finden Sie unter Amazon-S3-Ereignis-Benachrichtigungen.

Anweisungen zur Konfiguration von Amazon S3 S3-Ereignisbenachrichtigungen finden Sie unter Aktivieren von Ereignisbenachrichtigungen mithilfe von Amazon SQSSNS, Amazon und AWS Lambda.

Das Folgende ist ein Beispiel für eine Nachricht, die Amazon S3 sendet, um ein `s3:LifecycleExpiration:Delete`-Ereignis zu veröffentlichen. Weitere Informationen finden Sie unter the section called “Struktur von Ereignismeldungen”.

```
{
   "Records":[
      {
         "eventVersion":"2.3",
         "eventSource":"aws:s3",
         "awsRegion":"us-west-2",
         "eventTime":"1970-01-01T00:00:00.000Z",
         "eventName":"LifecycleExpiration:Delete",
         "userIdentity":{
            "principalId":"s3.amazonaws.com"
         },
         "requestParameters":{
            "sourceIPAddress":"s3.amazonaws.com"
         },
         "responseElements":{
            "x-amz-request-id":"C3D13FE58DE4C810",
            "x-amz-id-2":"FMyUVURIY8/IgAtTv8xRjskZQpcIZ9KG4V5Wp6S7S/
JRWeUWerMUE5JgHvANOjpD"
         },
         "s3":{
            "s3SchemaVersion":"1.0",
            "configurationId":"testConfigRule",
            "bucket":{
               "name":"amzn-s3-demo-bucket",
               "ownerIdentity":{
                  "principalId":"A3NL1KOZZKExample"
               },
               "arn":"arn:aws:s3:::amzn-s3-demo-bucket"
            },
            "object":{
               "key":"expiration/delete",
               "sequencer":"0055AED6DCD90281E5",
            }
         }
      }
   ]
}
```

Nachrichten, die Amazon S3 sendet, um ein `s3:LifecycleTransition` Ereignis zu veröffentlichen, enthalten auch die folgenden Informationen:

```
"lifecycleEventData":{
    "transitionEventData": {
```

```
        "destinationStorageClass": the destination storage class for the object
    }
}
```

Elemente der Lebenszyklus-Konfiguration

Eine S3-Lifecycle-Konfiguration besteht aus Lifecycle-Regeln, die verschiedene Elemente enthalten, die die Aktionen beschreiben, die Amazon S3 während der Lebensdauer von Objekten durchführt. Sie geben eine Amazon S3 S3-Lifecycle-Konfiguration anXML, die aus einer oder mehreren Lebenszyklusregeln besteht, wobei jede Regel aus einem oder mehreren Elementen besteht.

```
<LifecycleConfiguration>
    <Rule>
         <Element>
    </Rule>
    <Rule>
         <Element>
         <Element>
    </Rule>
</LifecycleConfiguration>
```

Jeder Regel umfasst Folgendes:

- Regelmetadaten, die eine Regel-ID und einen Status enthalten, der angibt, ob die Regel aktiviert oder deaktiviert ist. Wenn eine Regel deaktiviert ist, führt Amazon S3 keine in der Regel spezifizierten Aktionen aus.
- Ein Filter, der die Objekte identifiziert, für die die Regel gilt. Sie können einen Filter mithilfe der Objektgröße, des Objektschlüsselpräfixes, eines oder mehrerer Objekt-Tags oder einer Kombination von Filtern angeben.
- Eine oder mehrere Übergangs- oder Ablaufaktionen mit einem Datum oder einem Zeitintervall innerhalb der Lebensdauer des Objekts, zu denen Amazon S3 die angegebene Aktion ausführen soll.

Themen

- ID-Element
- Statuselement
- Filterelement
- Elemente, die Lebenszyklus-Aktionen beschreiben

- Filter zu Lebenszyklusregeln hinzufügen

In den folgenden Abschnitten werden die XML Elemente einer S3-Lifecycle-Konfiguration beschrieben. Beispielkonfigurationen finden Sie unter Beispiele für S3-Lifecycle-Konfigurationen.

ID-Element

Eine S3-Lebenszyklus-Konfiguration kann bis zu 1.000 Regeln haben. Diese Grenze ist nicht einstellbar. Das `<ID>` Element identifiziert eine Regel eindeutig. Die Länge der ID ist auf 255 Zeichen begrenzt.

Statuselement

Der `<Status>` Elementwert kann entweder `Enabled` oder sein`Disabled`. Wenn eine Regel deaktiviert ist, führt Amazon S3 keine in der Regel definierten Aktionen aus.

Filterelement

Eine S3-Lebenszyklusregel kann für alle oder eine Teilmenge von Objekten in einem Bucket gelten, basierend auf dem `<Filter>` Element, das Sie in der Regel angeben.

Sie können Objekte nach dem Schlüsselpräfix, Objekt-Markierungen oder Kombinationen aus beidem filtern (bei einer Kombination verwendet Amazon S3 ein logisches `AND` für die Filter). Beispiele und weitere Informationen zu Filtern finden Sie unterFilter zu Lebenszyklusregeln hinzufügen.

- Angeben eines Filters mithilfe von Schlüsselpräfixen — Dieses Beispiel zeigt eine S3-Lebenszyklusregel, die für eine Teilmenge von Objekten gilt, die auf dem Schlüsselnamenpräfix () `logs/` basieren. Die Lebenszyklusregel gilt beispielsweise für die Objekte `logs/mylog.txt`logs/temp1.txt, und. `logs/test.txt` Die Regel gilt nicht für das Objekt `example.jpg`.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
           <Prefix>logs/</Prefix>
        </Filter>
        transition/expiration actions
         ...
    </Rule>
    ...
```

```
</LifecycleConfiguration>
```

Wenn Sie eine Lebenszyklusaktion auf eine Teilmenge von Objekten anwenden möchten, die auf unterschiedlichen Schlüsselnamenpräfixen basieren, geben Sie separate Regeln an. Geben Sie in jeder Regel einen auf einem Präfix basierenden Filter an. Um beispielsweise eine Lebenszyklusaktion für Objekte mit den Schlüsselpräfixen `projectA/` und zu beschreiben`projectB/`, geben Sie zwei Regeln wie folgt an:

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
           <Prefix>projectA/</Prefix>
        </Filter>
        transition/expiration actions
         ...
    </Rule>

    <Rule>
        <Filter>
           <Prefix>projectB/</Prefix>
        </Filter>
        transition/expiration actions
         ...
    </Rule>
</LifecycleConfiguration>
```

Weitere Informationen über Objektschlüssel finden Sie unter Benennen Amazon S3 S3-Objekten.

- Angeben eines Filters auf der Grundlage von Objekt-Tags — Im folgenden Beispiel gibt die Lebenszyklusregel einen Filter an, der auf einem Tag (*key*) und einem Wert (*value*) basiert. Die Regel wird dann nur auf eine Untermenge von Objekten mit dem spezifischen Tag angewendet.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
           <Tag>
              <Key>key</Key>
              <Value>value</Value>
           </Tag>
        </Filter>
        transition/expiration actions
        ...
```

```
    </Rule>
</LifecycleConfiguration>
```

Sie können einen Filter auf mehreren Markierungen basierend angeben. Sie müssen die Tags in das <And> Element einschließen, wie im folgenden Beispiel gezeigt. Die Regel weist Amazon S3 an, Lebenszyklus-Aktionen für Objekte mit zwei Markierungen auszuführen (mit dem spezifischen Tag-Schlüssel und -wert).

```
<LifecycleConfiguration>
    <Rule>
      <Filter>
         <And>
            <Tag>
               <Key>key1</Key>
               <Value>value1</Value>
            </Tag>
            <Tag>
               <Key>key2</Key>
               <Value>value2</Value>
            </Tag>
             ...
          </And>
      </Filter>
      transition/expiration actions
    </Rule>
</Lifecycle>
```

Die Lebenszyklusregel gilt für Objekte, für die beide Markierungen angegeben sind. Amazon S3 führt eine logische AND-Operation aus. Beachten Sie Folgendes:

- Jedes Tag muss sowohl dem Schlüssel als auch dem Wert exakt entsprechen. Wenn Sie nur ein <Key> Element und kein <Value> Element angeben, gilt die Regel nur für Objekte, die dem Tag-Schlüssel entsprechen und für die kein Wert angegeben ist.
- Die Regel gilt für die Untermenge der Objekte, , die alle in der Regel angegebenen Markierungen besitzt. Wenn für ein Objekt zusätzliche Markierungen angegeben sind, gilt die Regel weiterhin.

> **Note**
>
> Wenn Sie mehrere Markierungen in einem Filter spezifizieren, muss jeder Tag-Schlüssel eindeutig sein.

- Angeben eines Filters, der sowohl auf dem Präfix als auch auf einem oder mehreren Tags basiert — In einer Lebenszyklusregel können Sie einen Filter angeben, der sowohl auf dem key prefix als auch auf einem oder mehreren Tags basiert. Auch hier müssen Sie alle diese Filterelemente wie folgt in das <And> Element einschließen:

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
          <And>
             <Prefix>key-prefix</Prefix>
             <Tag>
                <Key>key1</Key>
                <Value>value1</Value>
             </Tag>
             <Tag>
                <Key>key2</Key>
                <Value>value2</Value>
             </Tag>
              ...
          </And>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

  Amazon S3 kombiniert diese Filter mithilfe eines logischen AND. Das heißt, die Regel gilt für die Teilmenge der Objekte mit dem angegebenen key prefix und den angegebenen Tags. Ein Filter kann höchstens ein Präfix und null oder mehr Markierungen aufweisen.

- Sie können einen leeren Filter angeben, dann gilt die Regel für alle Objekte in dem Bucket.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

- Um eine Regel nach Objektgröße zu filtern, können Sie eine Mindestgröße (`ObjectSizeGreaterThan`) oder eine Höchstgröße (`ObjectSizeLessThan`) oder einen Bereich von Objektgrößen angeben.

  Objektgrößenwerte sind in Bytes. Standardmäßig werden Objekte, die kleiner als 128 KB sind, nicht in eine Speicherklasse übertragen, es sei denn, Sie geben eine kleinere Mindestgröße (`ObjectSizeGreaterThan`) oder eine Maximalgröße (`ObjectSizeLessThan`) an. Weitere Informationen finden Sie unter Beispiel: Ermöglicht die Übertragung von Objekten, die kleiner als 128 KB sind.

```
                    <LifecycleConfiguration>
    <Rule>
        <Filter>
            <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

> **Note**
>
> Die `ObjectSizeLessThan` Filter `ObjectSizeGreaterThan` und schließen die angegebenen Werte aus. Wenn Sie beispielsweise Objekte mit einer Größe von 128 KB bis 1024 KB so einstellen, dass sie von der Speicherklasse S3 Standard zur Speicherklasse S3 Standard-IA verschoben werden, werden Objekte, die genau 1024 KB und 128 KB groß sind, nicht in S3 Standard-IA umgewandelt. Stattdessen gilt die Regel nur für Objekte, die größer als 128 KB und kleiner als 1024 KB sind.

Wenn Sie einen Objektgrößenbereich angeben, muss die `ObjectSizeGreaterThan`-Ganzzahl kleiner als der `ObjectSizeLessThan`-Wert sein. Wenn Sie mehr als einen Filter verwenden, müssen Sie die Filter in ein <And>-Element packen. Das folgende Beispiel zeigt, wie Objekte in einem Bereich zwischen 500 Byte und 64.000 Byte angegeben werden.

```
<LifecycleConfiguration>
    <Rule>
```

```
        <Filter>
            <And>
                <Prefix>key-prefix</Prefix>
                <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
                <ObjectSizeLessThan>64000</ObjectSizeLessThan>
            </And>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

Elemente, die Lebenszyklus-Aktionen beschreiben

Sie können Amazon S3 anweisen, spezifische Aktionen innerhalb der Lebensdauer eines Objekts auszuführen, indem Sie eine oder mehrere vordefinierte Aktionen in einer S3-Lebenszyklusregel angeben. Die Wirkung dieser Aktionen ist vom Versioning-Status Ihres Buckets abhängig.

- **`Transition`**Aktionselement — Sie geben die `Transition` Aktion an, um Objekte von einer Speicherklasse in eine andere zu übertragen. Weitere Informationen zum Übergang von Objekten finden Sie unter Unterstützte Übergänge. Wenn ein vorgegebenes Datum oder einem Zeitintervall innerhalb der Lebensdauer des Objekts erreicht ist, führt Amazon S3 den Übergang aus.

  Für einen versionsfähigen Bucket (Bucket mit aktiviertem oder ausgesetztem Versioning) wird die `Transition`-Aktion auf die aktuelle Objektversion angewendet. Um nicht aktuelle Versionen zu verwalten, definiert Amazon S3 die `NoncurrentVersionTransition`-Aktion (wird später in diesem Thema beschrieben).

- **`Expiration`**Aktionselement — Die `Expiration` Aktion lässt die in der Regel identifizierten Objekte ablaufen und gilt für geeignete Objekte in einer der Amazon S3 S3-Speicherklassen. Weitere Informationen über Speicherklassen finden Sie unter Amazon S3 S3-Speicherklassen verstehen und verwalten. Amazon S3 lässt keinen Zugriff auf abgelaufene Objekte zu. Ob die Objekte dauerhaft entfernt werden, ist vom Versioning-Status des Buckets abhängig.
  - Bucket ohne Versionsnummer — Die `Expiration` Aktion führt dazu, dass Amazon S3 das Objekt dauerhaft entfernt.
  - Versionsfähiger Bucket – Für einen versionsfähigen Bucket (d. h. das Versioning ist aktiviert oder ausgesetzt) gibt es mehrere Aspekte, die bestimmen, wie Amazon S3 `Expiration`-Aktion verarbeitet. Für Buckets mit aktivierter oder ausgesetzter Versionsverwaltung gilt Folgendes:

- Die `Expiration`-Aktion wird nur auf die aktuelle Version angewendet (sie wirkt sich nicht auf nicht aktuelle Objektversionen aus).
- Amazon S3 führt keine Aktion aus, wenn es eine oder mehrere Objektversionen gibt und die Löschmarkierung die aktuelle Version ist.
- Wenn die aktuelle Objektversion die einzige Objektversion und auch eine Löschmarkierung ist (auch als Löschmarkierung eines abgelaufenen Objekts bezeichnet, wobei alle Objektversionen gelöscht werden und nur noch eine Löschmarkierung beibehalten wird), entfernt Amazon S3 die Löschmarkierung des abgelaufenen Objekts. Sie können die Ablaufaktion auch verwenden, um Amazon S3 anzuweisen, alle abgelaufenen Löschmarkierungen zu entfernen. Ein Beispiel finden Sie unter Löschen von Markierungen für das Löschen abgelaufener Objekte in einem Bucket mit aktivierter Versionierung.

Weitere Informationen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

Berücksichtigen Sie auch die folgenden Punkte, wenn Sie Amazon S3 zur Ablaufverwaltung einrichten:

- Bucket mit aktiviertem Versioning

  Wenn die aktuelle Objektversion keine Löschmarkierung ist, fügt Amazon S3 die Löschmarkierung mit einer eindeutigen Versions-ID hinzu. Damit ist die aktuelle Version nicht mehr aktuell und die Löschmarkierung wird zur aktuellen Version.

- Bucket mit ausgesetztem Versioning

  In einem Bucket, für das die Versionierung gesperrt wurde, veranlasst die Ablaufaktion, dass Amazon S3 eine Löschmarkierung mit der `null` Versions-ID erstellt. Diese Löschmarkierung ersetzt jede Objektversion mi einer Versions-ID von null in der Versionshierarchie, womit das Objekt effektiv gelöscht wird.

Darüber hinaus unterstützt Amazon S3 die folgenden Aktionen, mit denen Sie nicht aktuelle Objektversionen in einem versionsfähigen Bucket verwalten können (d. h. für Buckets mit aktivierten und ausgesetztem Versioning).

- **`NoncurrentVersionTransition`**Aktionselement — Verwenden Sie diese Aktion, um anzugeben, wann Amazon S3 Objekte in die angegebene Speicherklasse überführt. Sie können diesem Ablauf eine bestimmte Anzahl von Tagen zugrunde legen, seit die Objekte nicht mehr aktuell sind. Zusätzlich zur Anzahl der Tage können Sie auch eine maximale Anzahl an veralteten

Versionen angeben, die aufbewahrt werden sollen (zwischen 1 und 100). Dieser Wert bestimmt, wie viele neuere, nicht aktuelle Versionen vorhanden sein müssen, bevor Amazon S3 die zugehörige Aktion für eine bestimmte Version ausführen kann. Amazon S3 stellt alle zusätzlichen Langzeitversionen um, die über die angegebene Anzahl hinaus beibehalten werden sollen.

Um die maximale Anzahl an nicht aktuellen Versionen anzugeben, müssen Sie auch ein `Filter` Element angeben. Wenn Sie kein `Filter` Element angeben, generiert Amazon S3 einen `InvalidRequest` Fehler, wenn Sie eine maximale Anzahl von nicht aktuellen Versionen angeben.

Weitere Informationen zum Übergang von Objekten finden Sie unter Unterstützte Übergänge. Ausführliche Informationen dazu, wie Amazon S3 das Datum berechnet, wenn Sie die Anzahl der Tage in der `NoncurrentVersionTransition`-Aktion angeben, finden Sie unter Lebenszyklusregeln: Basierend auf dem Alter eines Objekts.

- **`NoncurrentVersionExpiration`**Aktionselement — Verwenden Sie diese Aktion, um Amazon S3 anzuweisen, nicht aktuelle Versionen von Objekten dauerhaft zu löschen. Diese gelöschten Objekte können nicht wiederhergestellt werden. Sie können dieses Ablaufdatum auf eine bestimmte Anzahl von Tagen beziehen, seit die Objekte nicht mehr aktuell sind. Zusätzlich zur Anzahl der Tage können Sie auch eine maximale Anzahl an veralteten Versionen angeben, die aufbewahrt werden sollen (zwischen 1 und 100). Dieser Wert gibt die Anzahl der neueren nicht aktuellen Versionen an, die vorhanden sein müssen, damit Amazon S3 die zugehörige Aktion für eine bestimmte Version ausführen kann. Amazon S3 löscht alle zusätzlichen älteren Versionen, die über die angegebene Anzahl hinaus aufbewahrt werden sollen, dauerhaft.

  Um die maximale Anzahl an nicht aktuellen Versionen anzugeben, müssen Sie auch ein `Filter` Element angeben. Wenn Sie kein `Filter` Element angeben, generiert Amazon S3 einen `InvalidRequest` Fehler, wenn Sie eine maximale Anzahl von nicht aktuellen Versionen angeben.

  Das verzögerte Entfernen nicht aktueller Objekte kann hilfreich sein, wenn Sie versehentliche Lösch- oder Überschreibvorgänge korrigieren müssen. Beispielsweise können Sie eine Ablaufregel konfigurieren, um nicht aktuelle Versionen fünf Tage nach dem Zeitpunkt zu löschen, zu dem sie nicht aktuell geworden sind. Nehmen wir zum Beispiel an, dass Sie am 1.1.2014 um 10:30 Uhr UTC ein Objekt mit dem Namen `photo.gif` (Versions-ID 111111) erstellen. Am 02.01.2014 um 11:30 Uhr löschen Sie versehentlich `photo.gif` (Versions-ID 111111)UTC, wodurch eine Löschmarke mit einer neuen Versions-ID (z. B. Versions-ID 4857693) erstellt wird. Jetzt haben Sie fünf Tage Zeit, die Originalversion von `photo.gif` (Versions-ID 111111) wiederherzustellen, bevor das Löschen permanent wird. Am 08.01.2014 um 00:00 UTC Uhr wird die Lebenszyklusregel

für den Ablauf ausgeführt und dauerhaft gelöscht `photo.gif` (Versions-ID 111111), fünf Tage, nachdem sie zu einer nicht aktuellen Version geworden ist.

Weitere Informationen darüber, wie Amazon S3 das Datum berechnet, an dem Sie die Anzahl der Tage `NoncurrentVersionExpiration` Aktion angeben finden Sie in Lebenszyklusregeln: Basierend auf dem Alter eines Objekts.

> **Note**
>
> Bei den Lebenszykluskonfigurationen für den Ablauf von Objekten werden unvollständige mehrteilige Uploads nicht entfernt. Um unvollständige mehrteilige Uploads zu entfernen, müssen Sie die Aktion „`AbortIncompleteMultipartUpload`Lebenszyklus-Konfiguration“ verwenden, die weiter unten in diesem Abschnitt beschrieben wird.

Zusätzlich zu den Übergangs- und Ablaufaktionen können Sie die folgenden Lifecycle-Konfigurationsaktionen verwenden, um Amazon S3 anzuweisen, unvollständige mehrteilige Uploads zu beenden oder Markierungen für das Löschen abgelaufener Objekte zu entfernen:

- **`AbortIncompleteMultipartUpload`**Aktionselement — Verwenden Sie dieses Element, um eine maximale Dauer (in Tagen) festzulegen, für die mehrteilige Uploads in Bearbeitung bleiben sollen. Wenn die entsprechenden mehrteiligen Uploads (bestimmt durch den in der Lebenszyklusregel `prefix` angegebenen Schlüsselnamen) nicht innerhalb des vordefinierten Zeitraums erfolgreich abgeschlossen werden, stoppt Amazon S3 die unvollständigen mehrteiligen Uploads. Weitere Informationen finden Sie unter Abbrechen eines mehrteiligen Uploads.

  > **Note**
  >
  > Sie können diese Lebenszyklusaktion nicht in einer Regel angeben, die über einen Filter verfügt, der Objekt-Tags verwendet.

- **`ExpiredObjectDeleteMarker`**Aktionselement — In einem Bucket mit aktivierter Versionierung wird eine Löschmarkierung ohne ältere Versionen als Löschmarkierung für abgelaufene Objekte bezeichnet. Sie können diese Lebenszyklusaktion verwenden, um Amazon S3 anzuweisen, Markierungen für das Löschen abgelaufener Objekte zu entfernen. Ein Beispiel finden Sie unter Löschen von Markierungen für das Löschen abgelaufener Objekte in einem Bucket mit aktivierter Versionierung.

> **Note**
>
> Sie können diese Lebenszyklusaktion nicht in einer Regel angeben, die über einen Filter verfügt, der Objekt-Tags verwendet.

Wie Amazon S3 berechnet, wie lange ein Objekt nicht aktuell ist

In einem versionsfähigen Bucket können Sie mehrere Versionen eines Objekts haben. Es gibt immer eine aktuelle Version und null oder mehr nicht aktuelle Versionen. Immer wenn Sie ein Objekt hochladen, wird die aktuelle Version als die nicht aktuelle Version beibehalten, und die neu hinzugefügte Version, der Nachfolger, wird zur aktuellen Version. Um die Anzahl der Tage zu bestimmen, wie lange ein Objekt nicht aktuell ist, wertet Amazon S3 es aus, wenn der Nachfolger erstellt wurde. Amazon S3 verwendet die Anzahl der Tage, seit der Nachfolger erstellt wurde, als die Anzahl der Tage, wie lange ein Objekt nicht aktuell ist.

> **Wiederherstellung vorheriger Versionen eines Objekts bei Verwendung von S3-Lebenszyklus-Konfigurationen**
>
> Wie unter erklärtWiederherstellen früherer Versionen, können Sie eine der folgenden beiden Methoden verwenden, um frühere Versionen eines Objekts abzurufen:
>
> - Methode 1 — Kopieren Sie eine nicht aktuelle Version des Objekts in denselben Bucket. Das kopierte Objekt wird zur aktuellen Version dieses Objekts, und alle Objektversionen werden beibehalten.
> - Methode 2 — Die aktuelle Version des Objekts dauerhaft löschen. Wenn Sie die aktuelle Objektversion löschen, wandeln Sie letztlich die nicht aktuelle Version in die aktuelle Version dieses Objekts um.
>
> Wenn Sie S3 Lifecycle-Konfigurationsregeln mit Buckets mit aktivierter Versionierung verwenden, empfehlen wir als bewährte Methode, Methode 1 zu verwenden.
> Der S3-Lebenszyklus arbeitet nach einem letztlich konsistenten Modell. Eine aktuelle Version, die Sie dauerhaft gelöscht haben, verschwindet möglicherweise erst, wenn die Änderungen auf alle Amazon S3-Systeme übertragen wurden. (Daher weiß Amazon S3 möglicherweise vorübergehend nichts von dieser Löschung.) In der Zwischenzeit kann die Lebenszyklusregel, die Sie für den Ablauf nicht aktueller Objekte konfiguriert haben, die nicht aktuellen Objekte dauerhaft entfernen, auch dasjenige, das Sie wiederherstellen möchten.

Daher ist das Kopieren der alten Version, wie in Methode 1 empfohlen, die sicherere Alternative.

Lebenszyklus-Aktionen und der Versioning-Status eines Buckets

Lebenszyklusregeln: Basierend auf dem Alter eines Objekts

Sie können einen Zeitraum in der Anzahl der Tage ab der Erstellung (oder Änderung) des Objekts angeben, in dem Amazon S3 die angegebene Aktion ausführen kann.

Wenn Sie die Anzahl der Tage in den `Transition`- und `Expiration`-Aktionen in einer S3-Lebenszyklus-Konfiguration angeben, beachten Sie Folgendes:

- Der Wert, den Sie angeben, ist die Anzahl der Tage seit der Objekterstellung, an denen die Aktion ausgeführt wird.
- Amazon S3 berechnet die Zeit, indem es die in der Regel angegebene Anzahl von Tagen zur Objekterstellungszeit addiert und die resultierende Zeit auf den nächsten Tag um Mitternacht rundet. UTC Wenn ein Objekt beispielsweise am 15.01.2014 um 10:30 Uhr erstellt wurde UTC und Sie in einer Übergangsregel 3 Tage angeben, würde das Übergangsdatum des Objekts als 19.01.2014 00:00 Uhr berechnet. UTC

> **Note**
>
> Amazon S3 behält nur das letzte Änderungsdatum für jedes Objekt bei. Beispielsweise zeigt die Amazon S3 S3-Konsole das Datum der letzten Änderung im Eigenschaftenbereich des Objekts an. Wenn Sie zum ersten Mal ein neues Objekt erstellen, spiegelt dieses Datum das Datum wider, an dem das Objekt erstellt wurde. Wenn Sie das Objekt ersetzen, ändert sich das Datum entsprechend. Daher ist das Erstellungsdatum gleichbedeutend mit dem Datum der letzten Änderung.

Wenn Sie die Anzahl der Tage in den `NoncurrentVersionTransition`- und `NoncurrentVersionExpiration`-Aktionen in einer Lebenszyklus-Konfiguration angeben, beachten Sie Folgendes:

- Der Wert, den Sie angeben, ist die Anzahl der Tage, ab dem die Version des Objekts veraltet ist (d. h. wenn das Objekt überschrieben oder gelöscht wird), an denen Amazon S3 die Aktion für das oder die angegebenen Objekte ausführt.
- Amazon S3 berechnet die Zeit, indem es die in der Regel angegebene Anzahl von Tagen zu dem Zeitpunkt addiert, zu dem die neue Nachfolgerversion des Objekts erstellt wird, und die resultierende Zeit auf den nächsten Tag um Mitternacht rundet. UTC Nehmen wir zum Beispiel an, dass Sie in Ihrem Bucket eine aktuelle Version eines Objekts haben, das am 1.1.2014 um 10:30 Uhr erstellt wurde. UTC Wenn die neue Version des Objekts, das die aktuelle Version ersetzt, am 15.01.2014 um 10:30 Uhr erstellt wird und Sie in einer Übergangsregel 3 Tage angebenUTC, wird das Übergangsdatum des Objekts als 19.01.2014 00:00 Uhr berechnet. UTC

Lebenszyklusregeln: Basierend auf einem spezifischen Datum

Wenn Sie in einer S3-Lebenszyklus-Regel eine Aktion angeben, können Sie ein Datum angeben, wann Amazon S3 die Aktion ausführen soll. Wenn das spezifische Datum erreicht ist, wendet Amazon S3 die Aktion auf alle qualifizierten Objekte an (basierend auf den Filterkriterien).

Wenn Sie eine S3-Lebenszyklusaktion mit einem Datum angeben, das in der Vergangenheit liegt, kommen alle qualifizierten Objekte sofort für diese Lebenszyklusaktion in Frage.

> **Important**
>
> Die auf einem Datum basierende Aktion ist keine einmalige Aktion. Amazon S3 wendet die auf dem Datum basierende Aktion auch an, nachdem das Datum erreicht wurde, solange der Regel-Status is `Enabled`.
> Nehmen wir beispielsweise an, dass Sie eine datumsbasierte `Expiration` Aktion angeben, um alle Objekte zu löschen (gehen Sie davon aus, dass in der Regel kein Filter angegeben ist). Amazon S3 lässt zu dem angegebenen Datum alle Objekte in dem Bucket ablaufen. Amazon S3 läuft auch weiterhin für alle neuen Objekte ab, die Sie im Bucket erstellen. Um die Lebenszyklusaktion zu beenden, müssen Sie entweder die Aktion aus der Lebenszyklusregel entfernen, die Regel deaktivieren oder die Regel aus der Lebenszykluskonfiguration löschen.

Der Datumswert muss dem Format ISO 8601 entsprechen. Die Zeit ist immer Mitternacht. UTC

> **Note**
>
> Sie können mit der Amazon S3 S3-Konsole keine datumsbasierten Lebenszyklusregeln erstellen, aber Sie können solche Regeln anzeigen, deaktivieren oder löschen.

Filter zu Lebenszyklusregeln hinzufügen

Filter sind ein optionales Lifecyle-Regelelement, mit dem Sie angeben können, für welche Objekte die Regel gilt.

Die folgenden Elemente können zum Filtern von Objekten verwendet werden:

Schlüsselpräfix

Sie können Objekte anhand eines Präfixes filtern. Wenn Sie Lebenszyklusaktionen für eine Teilmenge von Objekten unter mehr als einem Präfix anwenden möchten, können Sie separate Regeln angeben. Geben Sie in jeder Regel einen auf einem Präfix basierenden Filter an. Weitere Informationen finden Sie unter [Beispiel]

Objekt-Tags

Sie können Objekte anhand eines oder mehrerer Tags filtern. Jedes Tag muss sowohl dem Schlüssel als auch dem Wert exakt entsprechen, und wenn Sie mehrere Tags angeben, muss jeder Tagschlüssel eindeutig sein. Ein Filter mit mehreren Objekt-Tags gilt für eine Teilmenge von Objekten, für die alle Tags angegeben sind. Wenn für ein Objekt zusätzliche Tags angegeben wurden, gilt der Filter trotzdem.

> **Note**
>
> Wenn Sie nur ein `Key` Element und kein `Value` Element angeben, gilt die Regel nur für Objekte, die dem Tag-Schlüssel entsprechen und für die kein Wert angegeben ist.

Minimale oder maximale Objektgröße

Sie können Objekte nach ihrer Größe filtern. Sie können eine Mindestgröße (`ObjectSizeGreaterThan`) oder eine Maximalgröße (`ObjectSizeLessThan`) angeben, oder Sie können einen Bereich von Objektgrößen in demselben Filter angeben. Objektgrößenwerte

sind in Bytes. Die maximale Filtergröße beträgt 5 TB. Amazon S3 wendet eine standardmäßige Mindestobjektgröße für die Lebenszykluskonfiguration an. Weitere Informationen finden Sie unter Beispiel: Ermöglicht die Übertragung von Objekten, die kleiner als 128 KB sind.

Sie können verschiedene Filterelemente kombinieren. In diesem Fall verwendet Amazon S3 ein logischesAND.

Beispiele filtern

Die folgenden Beispiele zeigen, wie Sie verschiedene Filterelemente verwenden können:

- Angeben eines Filters mithilfe von Schlüsselpräfixen — Dieses Beispiel zeigt eine S3-Lebenszyklusregel, die für eine Teilmenge von Objekten gilt, die auf dem Schlüsselnamenpräfix () `logs/` basieren. Die Lebenszyklusregel gilt beispielsweise für die Objekte `logs/mylog.txt`logs/temp1.txt, und. `logs/test.txt` Die Regel gilt nicht für das Objekt `example.jpg`.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
           <Prefix>logs/</Prefix>
        </Filter>
        transition/expiration actions
         ...
    </Rule>
    ...
</LifecycleConfiguration>
```

> **Note**
>
> Wenn Sie über ein oder mehrere Präfixe verfügen, die mit denselben Zeichen beginnen, können Sie all diese Präfixe in Ihre Regel aufnehmen, indem Sie im Filter ein Teilpräfix ohne abschließenden Schrägstrich (/) angeben. Nehmen wir zum Beispiel an, Sie haben diese Präfixe:
>
> ```
> sales1999/
>                sales2000/
>                sales2001/
> ```

Um alle drei Präfixe in Ihre Regel aufzunehmen, geben Sie `sales` als Präfix in Ihrer Lebenszyklusregel an.

Wenn Sie eine Lebenszyklusaktion auf eine Teilmenge von Objekten anwenden möchten, die auf unterschiedlichen Schlüsselnamenpräfixen basieren, geben Sie separate Regeln an. Geben Sie in jeder Regel einen auf einem Präfix basierenden Filter an. Um beispielsweise eine Lebenszyklusaktion für Objekte mit den Schlüsselpräfixen `projectA/` und zu beschreiben`projectB/`, geben Sie zwei Regeln wie folgt an:

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
           <Prefix>projectA/</Prefix>
        </Filter>
        transition/expiration actions
         ...
    </Rule>

    <Rule>
        <Filter>
           <Prefix>projectB/</Prefix>
        </Filter>
        transition/expiration actions
         ...
    </Rule>
</LifecycleConfiguration>
```

Weitere Informationen über Objektschlüssel finden Sie unter Benennen Amazon S3 S3-Objekten.

- Angeben eines Filters auf der Grundlage von Objekt-Tags — Im folgenden Beispiel gibt die Lebenszyklusregel einen Filter an, der auf einem Tag (*key*) und einem Wert (*value*) basiert. Die Regel wird dann nur auf eine Untermenge von Objekten mit dem spezifischen Tag angewendet.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
           <Tag>
              <Key>key</Key>
              <Value>value</Value>
```

```
            </Tag>
         </Filter>
         transition/expiration actions
         ...
    </Rule>
</LifecycleConfiguration>
```

Sie können einen Filter auf mehreren Markierungen basierend angeben. Sie müssen die Tags in das <And> Element einschließen, wie im folgenden Beispiel gezeigt. Die Regel weist Amazon S3 an, Lebenszyklus-Aktionen für Objekte mit zwei Markierungen auszuführen (mit dem spezifischen Tag-Schlüssel und -wert).

```
<LifecycleConfiguration>
    <Rule>
      <Filter>
         <And>
            <Tag>
               <Key>key1</Key>
               <Value>value1</Value>
            </Tag>
            <Tag>
               <Key>key2</Key>
               <Value>value2</Value>
            </Tag>
             ...
          </And>
      </Filter>
      transition/expiration actions
    </Rule>
</Lifecycle>
```

Die Lebenszyklusregel gilt für Objekte, für die beide Markierungen angegeben sind. Amazon S3 führt eine logische AND-Operation aus. Beachten Sie Folgendes:

- Jedes Tag muss sowohl dem Schlüssel als auch dem Wert exakt entsprechen. Wenn Sie nur ein <Key> Element und kein <Value> Element angeben, gilt die Regel nur für Objekte, die dem Tag-Schlüssel entsprechen und für die kein Wert angegeben ist.
- Die Regel gilt für die Untermenge der Objekte, , die alle in der Regel angegebenen Markierungen besitzt. Wenn für ein Objekt zusätzliche Markierungen angegeben sind, gilt die Regel weiterhin.

> **Note**
>
> Wenn Sie mehrere Markierungen in einem Filter spezifizieren, muss jeder Tag-Schlüssel eindeutig sein.

- Angeben eines Filters, der sowohl auf dem Präfix als auch auf einem oder mehreren Tags basiert — In einer Lebenszyklusregel können Sie einen Filter angeben, der sowohl auf dem key prefix als auch auf einem oder mehreren Tags basiert. Auch hier müssen Sie alle diese Filterelemente wie folgt in das <And> Element einschließen:

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
          <And>
             <Prefix>key-prefix</Prefix>
             <Tag>
                <Key>key1</Key>
                <Value>value1</Value>
             </Tag>
             <Tag>
                <Key>key2</Key>
                <Value>value2</Value>
             </Tag>
              ...
          </And>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

  Amazon S3 kombiniert diese Filter mithilfe eines logischen AND. Das heißt, die Regel gilt für die Teilmenge der Objekte mit dem angegebenen key prefix und den angegebenen Tags. Ein Filter kann höchstens ein Präfix und null oder mehr Markierungen aufweisen.

- Einen leeren Filter angeben — Sie können einen leeren Filter angeben. In diesem Fall gilt die Regel für alle Objekte im Bucket.

```
<LifecycleConfiguration>
    <Rule>
```

```
        <Filter>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

- >Einen Objektgrößenfilter angeben — Um eine Regel nach Objektgröße zu filtern, können Sie eine Mindestgröße (`ObjectSizeGreaterThan`) oder eine Maximalgröße (`ObjectSizeLessThan`) oder einen Bereich von Objektgrößen angeben.

  Objektgrößenwerte sind in Bytes. Die maximale Filtergröße beträgt 5 TB. Für einige Speicherklassen gelten Mindestbeschränkungen für die Objektgröße. Weitere Informationen finden Sie unter Vergleich der Amazon-S3-Speicherklassen.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
            <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

> **Note**
>
> Die `ObjectSizeLessThan` Filter `ObjectSizeGreaterThan` und schließen die angegebenen Werte aus. Wenn Sie beispielsweise Objekte mit einer Größe von 128 KB bis 1024 KB so einstellen, dass sie von der Speicherklasse S3 Standard zur Speicherklasse S3 Standard-IA verschoben werden, werden Objekte, die genau 1024 KB und 128 KB groß sind, nicht in S3 Standard-IA umgewandelt. Stattdessen gilt die Regel nur für Objekte, die größer als 128 KB und kleiner als 1024 KB sind.

  Wenn Sie einen Objektgrößenbereich angeben, muss die `ObjectSizeGreaterThan`-Ganzzahl kleiner als der `ObjectSizeLessThan`-Wert sein. Wenn Sie mehr als einen Filter verwenden, müssen Sie die Filter in ein <And>-Element packen. Das folgende Beispiel zeigt, wie Objekte in einem Bereich zwischen 500 Byte und 64.000 Byte angegeben werden.

```
<LifecycleConfiguration>
    <Rule>
        <Filter>
            <And>
                <Prefix>key-prefix</Prefix>
                <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
                <ObjectSizeLessThan>64000</ObjectSizeLessThan>
            </And>
        </Filter>
        <Status>Enabled</Status>
        transition/expiration actions
    </Rule>
</LifecycleConfiguration>
```

Wie Amazon Amazon S3 mit Konflikten in Lebenszykluskonfigurationen umgeht

Im Allgemeinen gibt Amazon S3 Lifecycle Kostenoptimierung Vorrang. Wenn sich z. B. zwei Ablaufrichtlinien überschneiden, wird die Ablaufrichtlinie mit der kürzeren Frist durchgesetzt, sodass die Daten nicht länger als erwartet gespeichert werden. Wenn sich zwei Übergangsrichtlinien überschneiden, überführt S3 Lifecycle Ihre Objekte in die Speicherklasse mit geringeren Kosten.

In beiden Fällen versucht S3 Lifecycle den für Sie kostengünstigsten Pfad auszuwählen. Die Speicherklasse S3 Intelligent-Tiering ist von dieser Regel ausgenommen. S3 Intelligent-Tiering wird von S3 Lifecycle allen Speicherklassen vorgezogen, mit Ausnahme der Speicherklassen S3 Glacier und S3 Glacier Deep Archive.

Wenn Sie mehrere Regeln in einer S3 Lifecycle-Konfiguration haben, kann ein Objekt für mehrere S3 Lifecycle-Aktionen am selben Tag in Frage kommen. In solchen Fällen folgt Amazon S3 diesen allgemeinen Regeln:

- Das permanente Löschen hat Vorrang vor einem Übergang.
- Der Übergang hat Vorrang vor der Erstellung von Löschmarkierungen.
- Wenn ein Objekt sowohl für eine Umstellung auf S3 Glacier Flexible Retrieval als auch für eine Umstellung auf S3 Standard-IA (oder eine S3 One Zone-IA) in Frage kommt, entscheidet sich Amazon S3 für die Umstellung auf S3 Glacier Flexible Retrieval.

Beispiele für sich überschneidende Filter und widersprüchliche Lebenszyklusaktionen

Sie könnten eine S3-Lebenszyklus-Konfiguration angeben, in der Sie überlappende Präfixe oder Aktionen spezifizieren. Die folgenden Beispiele zeigen, wie Amazon S3 potenzielle Konflikte löst.

Example 1: Überlappende Präfixe (kein Konflikt)

Die folgende Beispielkonfiguration weist zwei Regeln auf, die überlappenden Präfixe spezifizieren, wie folgt:

- Die erste Regel spezifiziert einen leeren Filter, d. h. alle Objekte in dem Bucket werden angesprochen.
- Die zweite Regel spezifiziert ein Schlüsselnamenpräfix (`logs/`), d. h. nur eine Untermenge von Objekten.

Regel 1 fordert Amazon S3 auf, alle Objekte ein Jahr nach der Erstellung zu löschen. Regel 2 fordert Amazon S3 auf, 30 Tage nach der Erstellung eine Teilmenge von Objekten in die S3-Standard-IA-Speicherklasse zu überführen.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
  <Rule>
    <ID>Rule 2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>STANDARD_IA<StorageClass>
      <Days>30</Days>
    </Transition>
   </Rule>
</LifecycleConfiguration>
```

Da in diesem Fall kein Konflikt besteht, überführt Amazon S3 die Objekte mit dem Präfix `logs/` 30 Tage nach der Erstellung in die Speicherklasse S3 Standard-IA. Wenn ein Objekt ein Jahr nach der Erstellung erreicht, wird es gelöscht.

Example 2: Widersprüchliche Lebenszyklus-Aktionen

In dieser Beispielkonfiguration gibt es zwei Regeln, die Amazon S3 anweisen, zwei unterschiedliche Aktionen für dieselbe Objektmenge zur selben Zeit in der Lebensdauer des Objekts auszuführen:

- Beide Regeln geben dasselbe Schlüsselnamenpräfix an, deshalb gelten beide Regeln für dieselbe Objektmenge.
- Beide Regeln spezifizieren dieselben 365 Tage nach der Erstellung des Objekts, wann die Regeln angewendet werden sollen.
- Eine Regel weist Amazon S3 an, Objekte zur S3 Standard-IA-Speicherklasse zu überführen. Eine andere Regel weist Amazon S3 an, die Objekte zur gleichen Zeit ablaufen zu lassen.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
  <Rule>
    <ID>Rule 2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>STANDARD_IA<StorageClass>
      <Days>365</Days>
    </Transition>
   </Rule>
</LifecycleConfiguration>
```

In diesem Fall wollen Sie, dass Objekte ablaufen (zu entfernen), deshalb macht es keinen Sinn, die Speicherklasse zu ändern, und Amazon S3 wählt einfach die Ablaufaktion für diese Objekte aus.

Example 3: Überlappende Präfixe, die zu widersprüchlichen Lebenszyklus-Aktionen führen

In diesem Beispiel besitzt die Konfiguration zwei Regeln, die überlappende Präfixe angeben, wie folgt:

- Regel 1 legt ein leeres Präfix fest (was für alle Objekte gilt).
- Regel 2 spezifiziert ein Schlüsselnamenpräfix (`logs/`), das eine Untermenge aller Objekte angibt.

Für die Untermenge der Objekte mit dem Schlüsselnamenpräfix `logs/` werden die S3-Lebenszyklus-Aktionen aus beiden Regeln angewendet. Eine Regel weist Amazon S3 an, Objekte 10 Tage nach der Erstellung zu übertragen, und eine andere Regel weist Amazon S3 an, Objekte 365 Tage nach dem Erstellen zu übertragen.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
      <Prefix></Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>STANDARD_IA<StorageClass>
      <Days>10</Days>
    </Transition>
  </Rule>
  <Rule>
    <ID>Rule 2</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>STANDARD_IA<StorageClass>
      <Days>365</Days>
    </Transition>
   </Rule>
</LifecycleConfiguration>
```

In diesem Fall entscheidet Amazon S3 den Übergang 10 Tage nach der Erstellung auszuführen.

Example 4: Tag-basiertes Filtern, das zu widersprüchlichen Lebenszyklus-Aktionen führt

Angenommen, Sie haben die folgende S3-Lebenszykluskonfiguration, die zwei Regeln enthält, die beide einen Tag-Filter spezifizieren:

- Regel 1 spezifiziert einen Tag-basierten Filter (`tag1/value1`). Diese Regel weist Amazon S3 an, Objekte 365 Tage nach der Erstellung in die Speicherklasse S3 Glacier Flexible Retrieval zu übertragen.
- Regel 2 spezifiziert einen Tag-basierten Filter (`tag2/value2`). Diese Regel weist Amazon S3 an, Objekte 14 Tage nach der Erstellung ablaufen zu lassen.

Die S3-Lebenszyklus-Konfiguration wird im Folgenden angezeigt.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule 1</ID>
    <Filter>
      <Tag>
         <Key>tag1</Key>
         <Value>value1</Value>
      </Tag>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <StorageClass>GLACIER<StorageClass>
      <Days>365</Days>
    </Transition>
  </Rule>
  <Rule>
    <ID>Rule 2</ID>
    <Filter>
      <Tag>
         <Key>tag2</Key>
         <Value>value2</Value>
      </Tag>
    </Filter>
    <Status>Enabled</Status>
    <Expiration>
      <Days>14</Days>
    </Expiration>
```

```
   </Rule>
</LifecycleConfiguration>
```

Wenn ein Objekt beide Tags hat, muss Amazon S3 entscheiden, welche Regel befolgt werden soll. In diesem Fall lässt Amazon S3 das Objekt 14 Tage nach der Erstellung ablaufen. Das Objekt wird entfernt und die Übergangsaktion wird daher nicht angewendet.

Beispiele für S3-Lifecycle-Konfigurationen

Dieser Abschnitt enthält Beispiele für S3-Lebenszyklus-Konfigurationen. Jedes Beispiel zeigt, wie Sie das XML in den einzelnen Beispielszenarien spezifizieren können.

Themen

- Archivierung aller Objekte innerhalb eines Tages nach der Erstellung
- Lebenszyklusregeln vorübergehend deaktivieren
- Verkleinerung der Speicherklasse im Laufe der Lebensdauer eines Objekts
- Angabe mehrerer Regeln
- Angabe einer Lebenszyklusregel für einen Bucket mit aktivierter Versionierung
- Löschen von Markierungen für das Löschen abgelaufener Objekte in einem Bucket mit aktivierter Versionierung
- Lifecycle-Konfiguration zum Abbrechen mehrteiliger Uploads
- Ablaufende, nicht aktuelle Objekte, die keine Daten enthalten
- Beispiel: Ermöglicht die Übertragung von Objekten, die kleiner als 128 KB sind

Archivierung aller Objekte innerhalb eines Tages nach der Erstellung

Jede S3-Lebenszyklusregel enthält einen Filter, mit dem Sie eine Untermenge der Objekte in Ihrem Bucket identifizieren können, auf die sich die S3-Lebenszyklusregel bezieht. Das folgenden S3 Lifecycle-Konfigurationen zeigen Beispiele dafür, wie Sie einen Filter spezifizieren können.

- In dieser S3-Lebenszyklus-Konfigurationregel spezifiziert der Filter ein Schlüsselpräfix (`tax/`). Aus diesem Grund gilt die Regel für Objekte mit dem Schlüsselnamenpräfix `tax/`, wie beispielsweise `tax/doc1.txt` und `tax/doc2.txt`.

  Die Regel spezifiziert zwei Aktionen, die Amazon S3 zu Folgendem anweisen:

- Übergang von Objekten in die Speicherklasse S3 Glacier Flexible Retrieval 365 Tage (ein Jahr) nach der Erstellung.
- Objekte 3.650 Tage (10 Jahre) nach der Erstellung löschen (die `Expiration`-Aktion).

```
<LifecycleConfiguration>
  <Rule>
    <ID>Transition and Expiration Rule</ID>
    <Filter>
       <Prefix>tax/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
    <Expiration>
      <Days>3650</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

Anstatt das Objektalter in Tagen nach der Erstellung anzugeben, können Sie für jede Aktion ein Datum angeben. Sie können `Date` und `Days` nicht in derselben Regel kombinieren.

- Wenn Sie wollen, dass die S3-Lebenszyklusregel für alle Objekte im Bucket gilt, geben Sie ein leeres Präfix an. In der folgenden Konfiguration gibt die Regel eine `Transition`-Aktion an, die Amazon S3 anweist, Objekte 0 Tage nach der Erstellung in die S3 Glacier Flexible Retrieval-Speicherklasse zu überführen. Diese Regel bedeutet, dass die Objekte um Mitternacht UTC nach der Erstellung in S3 Glacier Flexible Retrieval archiviert werden können. Weitere Informationen zu Lebenszykluseinschränkungen finden Sie unter Einschränkungen und Überlegungen für Übergänge.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Archive all object same-day upon creation</ID>
    <Filter>
      <Prefix></Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>0</Days>
```

```
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

- Sie können null oder mehrere Schlüsselnamenpräfixe und null oder mehr Objekt-Markierungen in einem Filter angeben. Der folgende Beispiel-Code wendet die S3-Lebenszyklusregel auf eine Untermenge von Objekten mit dem Schüsselpräfix `tax/` an, ebenso wie auf Objekte mit zwei Markierungen mit spezifischem Schüssel und Wert. Wenn Sie mehr als einen Filter angeben, müssen Sie das <And>-Element wie gezeigt einschließen (Amazon S3 wendet ein logisches `AND` an, um die angegebenen Filterbedingungen zu kombinieren).

```
...
<Filter>
   <And>
      <Prefix>tax/</Prefix>
      <Tag>
         <Key>key1</Key>
         <Value>value1</Value>
      </Tag>
      <Tag>
         <Key>key2</Key>
         <Value>value2</Value>
      </Tag>
    </And>
</Filter>
...
```

- Sie können Objekte nur auf Markierungen basierend filtern. Die folgende S3-Lebenszyklusregel beispielsweise wird auf Objekte angewendet, die die beiden spezifizierten Markierungen aufweisen (es wird kein Präfix angegeben).

```
...
<Filter>
   <And>
      <Tag>
         <Key>key1</Key>
         <Value>value1</Value>
      </Tag>
      <Tag>
         <Key>key2</Key>
```

```
          <Value>value2</Value>
       </Tag>
    </And>
</Filter>
...
```

> **Important**
>
> Wenn Sie mehrere Regeln in einer S3 Lifecycle-Konfiguration haben, kann ein Objekt am selben Tag für mehrere S3 Lifecycle-Aktionen in Frage kommen. In solchen Fällen folgt Amazon S3 diesen allgemeinen Regeln:
>
> - Das permanente Löschen hat Vorrang vor einem Übergang.
> - Der Übergang hat Vorrang vor der Erstellung von Löschmarkierungen.
> - Wenn ein Objekt sowohl für eine Umstellung auf S3 Glacier Flexible Retrieval als auch für eine Umstellung auf S3 Standard-IA (oder S3 One Zone-IA) in Frage kommt, entscheidet sich Amazon S3 für die Umstellung auf S3 Glacier Flexible Retrieval.
>
> Beispiele finden Sie unter Beispiele für sich überschneidende Filter und widersprüchliche Lebenszyklusaktionen.

Lebenszyklusregeln vorübergehend deaktivieren

Sie können eine S3-Lebenszyklusregel mithilfe des `status` Elements vorübergehend deaktivieren. Dies kann nützlich sein, wenn Sie neue Regeln testen oder Probleme mit Ihrer Konfiguration beheben möchten, ohne Ihre vorhandenen Regeln zu überschreiben. Die folgende S3-Lebenszyklus-Konfiguration spezifiziert zwei Regeln:

- Regel 1 weist Amazon S3 an, Objekte mit dem Präfix `logs/` bald nach der Erstellung in die Speicherklasse S3 Glacier Flexible Retrieval zu übertragen.
- Regel 2 weist Amazon S3 an, Objekte mit dem Präfix `documents/` bald nach der Erstellung in die Speicherklasse S3 Glacier Flexible Retrieval zu übertragen.

In der Konfiguration ist Regel 1 aktiviert und Regel 2 ist deaktiviert. Amazon S3 ignoriert die deaktivierte Regel.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Rule1</ID>
    <Filter>
      <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
  <Rule>
    <ID>Rule2</ID>
    <Filter>
      <Prefix>documents/</Prefix>
    </Filter>
    <Status>Disabled</Status>
    <Transition>
      <Days>0</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

Verkleinerung der Speicherklasse im Laufe der Lebensdauer eines Objekts

In diesem Beispiel nutzen Sie die S3-Lebenszyklus-Konfiguration, um die Speicherklasse von Objekten über ihre Lebensdauer stufenweise zu reduzieren. Diese schichtweise Reduzierung kann dazu beitragen, Speicherkosten zu reduzieren. Weitere Informationen zu Preisen finden Sie unter Amazon-S3-Preise.

Die folgende S3-Lebenszyklus-Konfiguration spezifiziert eine Regel, die auf Objekte mit Schlüsselnamenpräfix `logs/` angewendet wird. Die Regel definiert die folgenden Aktionen:

- Zwei Übergangsaktionen:
  - Übergang von Objekten in die Speicherklasse S3 Standard-IA 30 Tage nach der Erstellung.

  - Übergang von Objekten in die Speicherklasse S3 Glacier Flexible Retrieval 90 Tage nach der Erstellung.
- Eine Ablaufaktion, die Amazon S3 anweist, Objekte ein Jahr nach ihrer Erstellung zu löschen.

```
<LifecycleConfiguration>
  <Rule>
    <ID>example-id</ID>
    <Filter>
       <Prefix>logs/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>STANDARD_IA</StorageClass>
    </Transition>
    <Transition>
      <Days>90</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
    <Expiration>
      <Days>365</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

Note

Sie können eine Regel verwenden, um alle S3-Lebenszyklus-Aktionen zu beschreiben, die für dieselbe Objektmenge angewendet werden (identifiziert durch den Filter). Andernfalls können Sie mehrere Regeln hinzufügen, die jeweils einen unterschiedlichen Filter angeben.

Important

Wenn Sie mehrere Regeln in einer S3 Lifecycle-Konfiguration haben, kann ein Objekt für mehrere S3 Lifecycle-Aktionen am selben Tag in Frage kommen. In solchen Fällen folgt Amazon S3 diesen allgemeinen Regeln:

  - Das permanente Löschen hat Vorrang vor einem Übergang.

- Der Übergang hat Vorrang vor der Erstellung von Löschmarkierungen.
- Wenn ein Objekt sowohl für eine Umstellung auf S3 Glacier Flexible Retrieval als auch für eine Umstellung auf S3 Standard-IA (oder S3 One Zone-IA) in Frage kommt, entscheidet sich Amazon S3 für die Umstellung auf S3 Glacier Flexible Retrieval.

Beispiele finden Sie unter Beispiele für sich überschneidende Filter und widersprüchliche Lebenszyklusaktionen.

Angabe mehrerer Regeln

Sie können mehrere Regeln angeben, wenn Sie unterschiedliche S3-Lebenszyklus-Aktionen auf unterschiedliche Objekte anwenden wollen. Die folgende S3-Lebenszyklus-Konfiguration spezifiziert zwei Regeln:

- Regel 1 gilt für Objekte mit dem Schlüsselnamenpräfix `classA/`. Sie weist Amazon S3 an, Objekte ein Jahr nach der Erstellung in die Speicherklasse S3 Glacier Flexible Retrieval zu übertragen, und diese Objekte 10 Jahre nach dem Erstellen ablaufen zu lassen.
- Regel 2 gilt für Objekte mit dem Schlüsselnamenpräfix `classB/`. Sie weist Amazon S3 an, Objekte 90 Tage nach der Erstellung in die Speicherklasse S3 Standard-IA zu übertragen, und sie ein Jahr nach dem zu löschen.

```
<LifecycleConfiguration>
    <Rule>
        <ID>ClassADocRule</ID>
        <Filter>
           <Prefix>classA/</Prefix>
        </Filter>
        <Status>Enabled</Status>
        <Transition>
           <Days>365</Days>
           <StorageClass>GLACIER</StorageClass>
        </Transition>
        <Expiration>
             <Days>3650</Days>
        </Expiration>
    </Rule>
    <Rule>
```

```
        <ID>ClassBDocRule</ID>
        <Filter>
            <Prefix>classB/</Prefix>
        </Filter>
        <Status>Enabled</Status>
        <Transition>
           <Days>90</Days>
           <StorageClass>STANDARD_IA</StorageClass>
        </Transition>
        <Expiration>
             <Days>365</Days>
        </Expiration>
    </Rule>
</LifecycleConfiguration>
```

Important

Wenn Sie mehrere Regeln in einer S3 Lifecycle-Konfiguration haben, kann ein Objekt am selben Tag für mehrere S3 Lifecycle-Aktionen in Frage kommen. In solchen Fällen folgt Amazon S3 diesen allgemeinen Regeln:

- Das permanente Löschen hat Vorrang vor einem Übergang.
- Der Übergang hat Vorrang vor der Erstellung von Löschmarkierungen.
- Wenn ein Objekt sowohl für eine Umstellung auf S3 Glacier Flexible Retrieval als auch für eine Umstellung auf S3 Standard-IA (oder S3 One Zone-IA) in Frage kommt, entscheidet sich Amazon S3 für die Umstellung auf S3 Glacier Flexible Retrieval.

Beispiele finden Sie unter Beispiele für sich überschneidende Filter und widersprüchliche Lebenszyklusaktionen.

Angabe einer Lebenszyklusregel für einen Bucket mit aktivierter Versionierung

Angenommen, Sie haben einen Versioning-fähigen Bucket, d. h. Sie haben für jedes Objekt eine aktuelle Version und keine oder mehr nicht aktuelle Versionen. (Weitere Informationen über das S3-Versioning finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.) In diesem Beispiel möchten Sie den Verlauf eines Jahres verwalten und die nicht aktuellen Versionen löschen. S3-Lebenszykluskonfigurationen unterstützen die Aufbewahrung von 1 bis 100 Versionen eines beliebigen Objekts.

Um Speicherkosten zu sparen, sollten Sie nicht aktuelle Versionen 30 Tage, nachdem sie nicht mehr aktuell werden, auf S3 Glacier Flexible Retrieval verschieben (vorausgesetzt, diese nicht aktuellen Objekte sind kalte Daten, für die Sie keinen Echtzeitzugriff benötigen). Darüber hinaus gehen Sie davon aus, dass die Zugriffshäufigkeit der aktuellen Versionen 90 Tage nach der Erstellung abnehmen wird. Sie könnten sich also dafür entscheiden, diese Objekte in die Speicherklasse S3 Standard-IA zu verschieben.

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
        <Filter>
           <Prefix></Prefix>
        </Filter>
        <Status>Enabled</Status>
        <Transition>
           <Days>90</Days>
           <StorageClass>STANDARD_IA</StorageClass>
        </Transition>
        <NoncurrentVersionTransition>
            <NoncurrentDays>30</NoncurrentDays>
            <StorageClass>GLACIER</StorageClass>
        </NoncurrentVersionTransition>
       <NoncurrentVersionExpiration>
            <NewerNoncurrentVersions>5</NewerNoncurrentVersions>
            <NoncurrentDays>365</NoncurrentDays>
       </NoncurrentVersionExpiration>
    </Rule>
</LifecycleConfiguration>
```

Löschen von Markierungen für das Löschen abgelaufener Objekte in einem Bucket mit aktivierter Versionierung

Ein Bucket mit Versioning enthält eine aktuelle Version und null oder mehr nicht aktuelle Versionen für jedes Objekt. Beachten Sie beim Löschen eines Objekts Folgendes:

- Wenn Sie keine Versions-ID in Ihrer Löschanfrage angeben, fügt Amazon S3 eine Löschmarkierung hinzu, statt das Objekt zu löschen. Das aktuelle Version wird nicht aktuell, und die Löschmarkierung wird zur aktuellen Version.
- Wenn Sie eine Versions-ID in Ihrer Löschanfrage angeben, löscht Amazon S3 die Objektversion permanent (es wird keine Löschmarkierung erstellt).

- Eine Löschmarkierung mit null nicht aktuellen Versionen wird als Löschmarkierung für das abgelaufene Objekt bezeichnet.

Dieses Beispiel zeigt ein Szenario, das Löschmarkierungen für abgelaufene Objekte in Ihrem Bucket erstellen kann, und demonstriert, wie Sie mit einer S3-Lebenszyklus-Konfiguration Amazon S3 anweisen können, die Löschmarkierungen für abgelaufene Objekte zu löschen.

Angenommen, Sie schreiben eine S3-Lifecycle-Konfiguration, die die `NoncurrentVersionExpiration` Aktion verwendet, um die nicht aktuellen Versionen 30 Tage, nachdem sie veraltet sind, zu entfernen und maximal 10 nicht aktuelle Versionen beizubehalten, wie im folgenden Beispiel gezeigt.

```
<LifecycleConfiguration>
    <Rule>
        ...
        <NoncurrentVersionExpiration>
            <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
            <NoncurrentDays>30</NoncurrentDays>
        </NoncurrentVersionExpiration>
    </Rule>
</LifecycleConfiguration>
```

Die Aktion `NoncurrentVersionExpiration` gilt nicht für die aktuellen Objektversionen. Sie entfernt nur nicht aktuelle Versionen.

Für aktuelle Objektversionen haben Sie die folgenden Optionen, ihre Lebensdauer zu verwalten, abhängig davon, ob die aktuellen Objektversionen einen definieren Lebenszyklus haben:

- Aktuelle Objektversionen folgen einem gut definierten Lebenszyklus.

  In diesem Fall können Sie eine S3-Lebenszykluskonfiguration mit der Aktion `Expiration` verwenden, um Amazon S3 anzuweisen, die aktuellen Versionen zu entfernen, wie im folgenden Beispiel gezeigt.

  ```
  <LifecycleConfiguration>
      <Rule>
          ...
          <Expiration>
             <Days>60</Days>
          </Expiration>
  ```

```
        <NoncurrentVersionExpiration>
            <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
            <NoncurrentDays>30</NoncurrentDays>
        </NoncurrentVersionExpiration>
    </Rule>
</LifecycleConfiguration>
```

In diesem Beispiel entfernt Amazon S3 aktuelle Versionen 60 Tage nach ihrer Erstellung, indem für jede der aktuellen Objektversionen eine Löschmarkierung hinzugefügt wird. Durch diesen Vorgang wird die aktuelle Version nicht mehr aktuell und die Löschmarkierung wird zur aktuellen Version. Weitere Informationen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

> **Note**
>
> Sie können nicht sowohl ein `Days` als auch `ExpiredObjectDeleteMarker`-Tag für dieselbe Regel angeben. Wenn Sie das `Days`-Tag angeben, führt Amazon S3 automatisch eine `ExpiredObjectDeleteMarker`-Bereinigung durch, wenn die Löschmarkierungen alt genug sind, um die Alterskriterien zu erfüllen. Um Löschmarkierungen zu bereinigen, sobald sie die einzige Version werden, erstellen Sie eine separate Regel nur mit dem `ExpiredObjectDeleteMarker`-Tag.

Die `NoncurrentVersionExpiration`-Aktion in derselben S3-Lebenszyklus-Konfiguration entfernt nicht aktuelle Objekte 30 Tage, nachdem sie nicht aktuell wurden. Somit werden in diesem Beispiel 90 Tage nach der Objekterstellung alle Objektversionen dauerhaft entfernt. Obwohl während dieses Vorgangs Löschmarkierungen für abgelaufene Objekte erstellt werden, erkennt und entfernt Amazon S3 die Löschmarkierungen für abgelaufene Objekte für Sie.

- Aktuelle Objektversionen folgen keinem gut definierten Lebenszyklus.

  In diesem Fall müssen Sie die Objekte möglicherweise manuell entfernen, wenn Sie sie nicht mehr brauchen, und eine Löschmarkierungen mit einer oder mehreren nicht aktuellen Versionen erstellen. Wenn Ihre S3-Lebenszyklus-Konfiguration mit der `NoncurrentVersionExpiration`-Aktion alle nicht aktuellen Versionen löscht, haben Sie jetzt Löschmarkierungen für abgelaufene Objekte.

  Speziell für dieses Szenario stellt die S3-Lebenszykluskonfiguration die `Expiration`-Aktion bereit, mit der Sie die Löschmarkierungen für abgelaufene Objekte entfernen können.

```
<LifecycleConfiguration>
    <Rule>
       <ID>Rule 1</ID>
        <Filter>
          <Prefix>logs/</Prefix>
        </Filter>
        <Status>Enabled</Status>
        <Expiration>
           <ExpiredObjectDeleteMarker>true</ExpiredObjectDeleteMarker>
        </Expiration>
        <NoncurrentVersionExpiration>
            <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
            <NoncurrentDays>30</NoncurrentDays>
        </NoncurrentVersionExpiration>
    </Rule>
</LifecycleConfiguration>
```

Setzen Sie das `ExpiredObjectDeleteMarker`-Element in der `Expiration`-Aktion auf `true`, um Amazon S3 anzuweisen, Löschmarkierungen für abgelaufene Objekte zu entfernen.

> **Note**
>
> Bei Verwendung der `ExpiredObjectDeleteMarker`-S3-Lebenszyklus-Aktion kann die Regel keinen Tag-basierten Filter angeben.

Lifecycle-Konfiguration zum Abbrechen mehrteiliger Uploads

Sie können die mehrteiligen REST API Upload-Operationen von Amazon S3 verwenden, um große Objekte in Teilen hochzuladen. Weitere Informationen über mehrteilige Uploads finden Sie unter Hochladen und Kopieren von Objekten mit mehrteiligen Uploads.

Mithilfe einer S3-Lifecycle-Konfiguration können Sie Amazon S3 anweisen, unvollständige mehrteilige Uploads (identifiziert durch das in der Regel angegebene Schlüsselnamenpräfix) zu beenden, wenn sie nicht innerhalb einer bestimmten Anzahl von Tagen nach der Initiierung abgeschlossen werden. Wenn Amazon S3 einen mehrteiligen Upload abbricht, werden alle diesem mehrteiligen Upload zugeordneten Teile gelöscht. Dieser Prozess hilft, Ihre Speicherkosten zu kontrollieren, indem Sie

sicherstellen, dass Sie keine unvollständigen mehrteiligen Uploads mit Teilen haben, die in Amazon S3 gespeichert sind.

> **Note**
>
> Bei Verwendung der `AbortIncompleteMultipartUpload`-S3-Lebenszyklus-Aktion kann die Regel keinen Tag-basierten Filter angeben.

Das folgende Beispiel zeigt eine S3-Lebenszyklus-Konfiguration, die eine Regel mit der Aktion `AbortIncompleteMultipartUpload` spezifiziert. Diese Aktion leitet Amazon S3 dazu, unvollständige mehrteilige Uploads sieben Tage nach der Initiierung abzubrechen.

```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
        <Filter>
           <Prefix>SomeKeyPrefix/</Prefix>
        </Filter>
        <Status>rule-status</Status>
        <AbortIncompleteMultipartUpload>
          <DaysAfterInitiation>7</DaysAfterInitiation>
        </AbortIncompleteMultipartUpload>
    </Rule>
</LifecycleConfiguration>
```

Ablaufende, nicht aktuelle Objekte, die keine Daten enthalten

Sie können Regeln erstellen, die Objekte nur basierend auf ihrer Größe übergehen. Sie können eine Mindestgröße (`ObjectSizeGreaterThan`) oder eine Maximalgröße (`ObjectSizeLessThan`) angeben, oder Sie können einen Bereich von Objektgrößen in Bytes angeben. Wenn Sie mehr als einen Filter verwenden, z. B. ein Präfix und eine Größenregel, müssen Sie die Filter in ein <And>-Element umfassen.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Transition with a prefix and based on size</ID>
    <Filter>
       <And>
          <Prefix>tax/</Prefix>
          <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
```

```
        </And>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

Wenn Sie einen Bereich mit den `ObjectSizeGreaterThan`- und `ObjectSizeLessThan`-Elementen angeben, muss die maximale Objektgröße größer als die minimale Objektgröße sein. Wenn Sie mehr als einen Filter verwenden, müssen Sie die Filter in ein <And>-Element packen. Das folgende Beispiel zeigt, wie Objekte in einem Bereich zwischen 500 Byte und 64.000 Byte angegeben werden. Wenn Sie einen Bereich angeben, schließen die `ObjectSizeLessThan` Filter `ObjectSizeGreaterThan` und die angegebenen Werte aus. Weitere Informationen finden Sie unter the section called “Filterelement”.

```
<LifecycleConfiguration>
    <Rule>
        ...
          <And>
             <ObjectSizeGreaterThan>500</ObjectSizeGreaterThan>
             <ObjectSizeLessThan>64000</ObjectSizeLessThan>
          </And>
    </Rule>
</LifecycleConfiguration>
```

Sie können auch Regeln erstellen, um nicht aktuelle Objekte, die keine Daten enthalten, ausdrücklich ablaufen zu lassen, einschließlich nicht aktueller Löschmarkierungsobjekte, die in einem Bucket mit aktivierter Versionsverwaltung erstellt wurden. Das folgende Beispiel verwendet die `NoncurrentVersionExpiration`-Aktion angibt, um nicht aktuelle Versionen 30 Tage, nachdem sie nicht mehr aktuell sind, zu entfernen und höchstens 10 nicht aktuelle Versionen beizubehalten, wie im folgenden Beispiel gezeigt. Außerdem wird das `ObjectSizeLessThan`-Element verwendet, um nur Objekte ohne Daten zu filtern.

```
<LifecycleConfiguration>
  <Rule>
    <ID>Expire noncurrent with size less than 1 byte</ID>
    <Filter>
```

```
        <ObjectSizeLessThan>1</ObjectSizeLessThan>
    </Filter>
    <Status>Enabled</Status>
    <NoncurrentVersionExpiration>
       <NewerNoncurrentVersions>10</NewerNoncurrentVersions>
       <NoncurrentDays>30</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

Beispiel: Ermöglicht die Übertragung von Objekten, die kleiner als 128 KB sind

Amazon S3 wendet auf Ihre Lifecycle-Konfiguration ein Standardverhalten an, das verhindert, dass Objekte, die kleiner als 128 KB sind, in eine beliebige Speicherklasse übertragen werden. Sie können den Übergang kleinerer Objekte zulassen, indem Sie der Konfiguration einen Filter für die Mindestgröße (`ObjectSizeGreaterThan`) oder die maximale Größe (`ObjectSizeLessThan`) hinzufügen, der eine kleinere Größe angibt. Das folgende Beispiel ermöglicht jedem Objekt, das kleiner als 128 KB ist, den Übergang zur S3 Glacier Instant Retrieval-Speicherklasse:

```
<LifecycleConfiguration>
  <Rule>
    <ID>Allow small object transitions</ID>
    <Filter>
          <ObjectSizeGreaterThan>1</ObjectSizeGreaterThan>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>GLACIER_IR</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

> **Note**
>
> Im September 2024 hat Amazon S3 das standardmäßige Übergangsverhalten für kleine Objekte wie folgt aktualisiert:
>
> - Aktualisiertes Standard-Übergangsverhalten — Ab September 2024 verhindert das Standardverhalten, dass Objekte, die kleiner als 128 KB sind, in eine beliebige Speicherklasse übertragen werden.

- Bisheriges Standard-Übergangsverhalten — Vor September 2024 erlaubte das Standardverhalten, dass Objekte, die kleiner als 128 KB waren, nur in die Speicherklassen S3 Glacier und S3 Glacier Deep Archive übertragen wurden.

Konfigurationen, die vor September 2024 erstellt wurden, behalten das vorherige Übergangsverhalten bei, sofern Sie sie nicht ändern. Das heißt, wenn Sie Regeln erstellen, bearbeiten oder löschen, ändert sich das standardmäßige Übergangsverhalten für Ihre Konfiguration in das aktualisierte Verhalten. Falls es Ihr Anwendungsfall erfordert, können Sie das standardmäßige Übergangsverhalten so ändern, dass Objekte, die kleiner als 128 KB sind, auf S3 Glacier und S3 Glacier Deep Archive umgestellt werden. Verwenden Sie dazu den optionalen `x-amz-transition-object-size-minimum-default` Header in einer PutBucketLifecycleConfigurationAnfrage.

Das folgende Beispiel zeigt, wie der `x-amz-transition-object-size-minimum-default` Header in einer PutBucketLifecycleConfigurationAnfrage verwendet wird, um das `varies_by_storage_class` standardmäßige Übergangsverhalten auf eine Lifecycle-Konfiguration anzuwenden. Dieses Verhalten ermöglicht Objekten, die kleiner als 128 KB sind, den Übergang zu den Speicherklassen S3 Glacier oder S3 Glacier Deep Archive. Standardmäßig verhindern alle anderen Speicherklassen Übergänge, die kleiner als 128 KB sind. Sie können weiterhin benutzerdefinierte Filter verwenden, um die minimale Übergangsgröße für jede Speicherklasse zu ändern. Benutzerdefinierte Filter haben immer Vorrang vor dem standardmäßigen Übergangsverhalten:

```
HTTP/1.1 200
x-amz-transition-object-size-minimum-default: varies_by_storage_class
<?xml version="1.0" encoding="UTF-8"?>
...
```

Behebung von Problemen mit Amazon S3 Lifecycle

Die folgenden Informationen können Ihnen helfen, häufiger auftretende Probleme mit den Lebenszyklusregeln von Amazon S3 zu beheben.

Themen

- Ich habe eine Auflistungsoperation für meinen Bucket ausgeführt und es wurden Objekte angezeigt, von denen ich dachte, dass sie abgelaufen oder aufgrund einer Lebenszyklusregel übergeben worden waren.
- Wie überwache ich die im Rahmen meiner Lebenszyklusregeln ergriffenen Maßnahmen?
- Die Anzahl meiner S3-Objekte steigt immer noch, auch wenn ich Lebenszyklusregeln für einen Bucket mit aktivierter Versionierung eingerichtet habe.
- Wie leere ich meinen S3-Bucket mithilfe von Lebenszyklusregeln?
- Meine Abrechnung für Amazon S3 weist nach der Übergabe von Objekten in eine kostengünstigere Speicherklasse höhere Kosten auf.
- Ich habe meine Bucket-Richtlinie aktualisiert, meine S3-Objekte werden jedoch noch immer aufgrund abgelaufener Lebenszyklusregeln gelöscht.
- Kann ich S3-Objekte wiederherstellen, die aufgrund von S3-Lebenszyklusregeln abgelaufen sind?
- Wie kann ich ein Präfix aus meiner Lebenszyklusregel ausschließen?
- Wie kann ich mehrere Präfixe in meine Lebenszyklusregel aufnehmen?

Ich habe eine Auflistungsoperation für meinen Bucket ausgeführt und es wurden Objekte angezeigt, von denen ich dachte, dass sie abgelaufen oder aufgrund einer Lebenszyklusregel übergeben worden waren.

Das Übergeben von Objekten und das Ablaufen von Objekten in S3-Lebenszyklen sind asynchrone Operationen. Daher kann es zu einer Verzögerung zwischen dem Zeitpunkt kommen, zu dem die Objekte für ein Ablaufen oder eine Übergabe infrage kommen, und dem Zeitpunkt, zu dem sie tatsächlich übergeben werden oder ablaufen. Änderungen an der Abrechnung werden sofort nach Erfüllung der Lebenszyklusregel angewendet, auch wenn die Aktion noch nicht abgeschlossen ist. Eine Ausnahme von diesem Verhalten tritt auf, wenn Sie eine Lebenszyklusregel für die Übergabe in die Speicherklasse S3 Intelligent Tiering festgelegt haben. In diesem Fall treten Abrechnungsänderungen erst auf, wenn das Objekt in S3 Intelligent-Tiering übergeben worden ist. Weitere Informationen zu Änderungen der Abrechnung finden Sie unter Festlegen der Lebenszyklus-Konfiguration für einen Bucket.

> **Note**
>
> Amazon S3 übergibt keine Objekte, die kleiner als 128 KB sind, von der Speicherklasse S3 Standard oder S3 Standard-IA in die Klassen S3 Intelligent-Tiering, S3 Standard-IA oder S3 One Zone-IA.

Wie überwache ich die im Rahmen meiner Lebenszyklusregeln ergriffenen Maßnahmen?

Um die im Rahmen von Lebenszyklusregeln durchgeführten Aktionen zu überwachen, können Sie die folgenden Funktionen verwenden:

- S3-Ereignisbenachrichtigungen — Sie können S3-Ereignisbenachrichtigungen so einrichten, dass Sie über alle Ablaufs- oder Übergangsereignisse von S3 Lifecycle informiert werden.
- S3-Serverzugriffsprotokolle — Sie können Serverzugriffsprotokolle für Ihre S3-Buckets aktivieren, um S3-Lifecycle-Aktionen wie Objektübergänge in eine andere Speicherklasse oder Objektablaufzeiten zu erfassen. Weitere Informationen finden Sie unter Lebenszyklus und Protokollieren.

Um die Änderungen an Ihrem Speicher, die durch Lebenszyklusaktionen verursacht werden, täglich zu sehen, empfehlen wir, S3 Storage Lens-Dashboards zu verwenden, anstatt CloudWatch Amazon-Metriken zu verwenden. In Ihrem Storage Lens-Dashboard können Sie sich die folgenden Metriken ansehen, die die Anzahl oder Größe von Objekten überwachen:

- Aktuelle Versions-Bytes
- Anzahl der Objekte der aktuellen Version
- Nicht aktuelle Versions-Bytes
- Anzahl nicht aktueller Versionsobjekte
- Anzahl der Löschmarkierungsobjekte
- Markierungsspeicherbytes löschen
- Unvollständige Bytes für mehrteilige Uploads
- Anzahl unvollständiger mehrteiliger Uploads

Die Anzahl meiner S3-Objekte steigt immer noch, auch wenn ich Lebenszyklusregeln für einen Bucket mit aktivierter Versionierung eingerichtet habe.

Wenn in einem Bucket mit aktivierter Versionierung ein Objekt abgelaufen ist, wird das Objekt nicht vollständig aus dem Bucket gelöscht. Stattdessen wird eine Löschmarkierung als neueste Version des Objekts erstellt. Löschmarkierungen werden weiterhin als Objekte gezählt. Wenn also die Lebenszyklusregel erstellt wird, dass nur die aktuellen Versionen ablaufen, erhöht sich die Anzahl der Objekte im S3-Bucket, anstatt zu sinken.

Nehmen wir beispielsweise an, dass für einen S3-Bucket mit 100 Objekten die Versionsverwaltung aktiviert und die Lebenszyklusregel festgelegt ist, dass aktuelle Versionen des Objekts nach 7 Tagen ablaufen. Nach dem siebten Tag erhöht sich die Anzahl der Objekte auf 200, da zusätzlich zu den 100 ursprünglichen Objekten, bei denen es sich jetzt um die nicht aktuellen Versionen handelt, 100 Löschmarkierungen erstellt werden. Weitere Informationen zu den Aktionen der S3-Lebenszyklus-Konfigurationsregel für Buckets mit aktivierter Versionsverwaltung finden Sie unter Festlegen der Lebenszyklus-Konfiguration für einen Bucket.

Um Objekte dauerhaft zu entfernen, fügen Sie eine zusätzliche Lebenszykluskonfiguration hinzu, um die vorherigen Versionen der Objekte, abgelaufene Löschmarkierungen und unvollständige mehrteilige Uploads zu löschen. Anweisungen zum Erstellen neuer Lebenszyklusregeln finden Sie unter Festlegen der Lebenszyklus-Konfiguration für einen Bucket.

> **Note**
>
> - Amazon S3 rundet das Übergangs- oder Verfallsdatum eines Objekts auf Mitternacht UTC des nächsten Tages ab.
>
> Bei der Bewertung von Objekten auf Lebenszyklusaktionen verwendet Amazon S3 die Objekterstellungszeit inUTC. Stellen Sie sich zum Beispiel einen Bucket ohne Versionsnummer mit einer Lebenszyklusregel vor, der so konfiguriert ist, dass Objekte nach einem Tag ablaufen. Angenommen, ein Objekt wurde am 1. Januar um 17:05 Uhr PDT (PDT) erstellt, was dem 2. Januar um 00:05 Uhr entspricht. UTC Das Objekt wird UTC am 3. Januar um 00:05 Uhr einen Tag alt, sodass es ablaufen kann, wenn S3 Lifecycle Objekte am 4. Januar um 00:00 UTC Uhr auswertet.
>
> Da Amazon S3 S3-Lebenszyklusaktionen asynchron ausgeführt werden, kann es zu Verzögerungen zwischen dem in der Lebenszyklusregel angegebenen Datum und dem

tatsächlichen physischen Übergang des Objekts kommen. Weitere Informationen finden Sie unter Übergangs- oder Ablaufverzögerung.

Weitere Informationen finden Sie unter Lebenszyklusregeln: Basierend auf dem Alter eines Objekts.

- Bei S3-Objekten, die durch die Objektsperre geschützt sind, werden aktuelle Versionen nicht dauerhaft gelöscht. Stattdessen wird den Objekten eine Löschmarkierung hinzugefügt, sodass sie nicht mehr aktuell sind. Nicht aktuelle Versionen werden dann beibehalten und laufen nicht dauerhaft ab.

Wie leere ich meinen S3-Bucket mithilfe von Lebenszyklusregeln?

S3-Lebenszyklusregeln sind ein effektives Tool zum Leeren eines S3-Buckets mit Millionen von Objekten. Um eine große Anzahl von Objekten aus Ihrem S3-Bucket zu löschen, stellen Sie sicher, dass Sie diese beiden Paare von Lebenszyklusregeln verwenden:

- Aktuelle Objektversionen ablaufen lassen und Vorherige Versionen von Objekten dauerhaft löschen
- Löschmarkierungen für abgelaufenes Objekt löschen und Unvollständige mehrteilige Uploads löschen

Anweisungen zum Erstellen einer neuen Lebenszyklus-Konfigurationsregel finden Sie unter Festlegen der Lebenszyklus-Konfiguration für einen Bucket.

> **Note**
>
> Bei S3-Objekten, die durch die Objektsperre geschützt sind, werden aktuelle Versionen nicht dauerhaft gelöscht. Stattdessen wird den Objekten eine Löschmarkierung hinzugefügt, sodass sie nicht mehr aktuell sind. Nicht aktuelle Versionen werden dann beibehalten und laufen nicht dauerhaft ab.

Meine Abrechnung für Amazon S3 weist nach der Übergabe von Objekten in eine kostengünstigere Speicherklasse höhere Kosten auf.

Es gibt mehrere Gründe, warum Ihre Abrechnung nach der Übergabe von Objekten in eine kostengünstigere Speicherklasse höhere Kosten aufweisen kann:

- S3-Glacier-Overheadgebühren für kleine Objekte

  Bei jedem Objekt, das an S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive übergeben wird, ist ein Gesamt-Overhead von 40 KB mit dieser Speicheraktualisierung verbunden. Im Rahmen des Overheads von 40 KB werden 8 KB zum Speichern der Metadaten und des Objektnamens verwendet. Diese 8 KB werden gemäß den S3-Standardgebühren berechnet. Die restlichen 32 KB werden für die Indexierung und die zugehörigen Metadaten verwendet. Diese 32 KB werden gemäß den Gebühren von S3 Glacier Flexible Retrieval oder S3 Glacier Deep Archive berechnet.

  Wenn Sie viele kleinere Objekte speichern, sollten Sie daher keine Lebenszyklusübergaben verwenden. Stattdessen sollten Sie viele kleinere Objekte zu einer kleineren Anzahl von großen Objekten zusammenzufassen, bevor Sie sie in Amazon S3 speichern, um so die Overheadgebühren zu reduzieren. Weitere Informationen zu den Kostenüberlegungen finden Sie unter Übergang in die Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive (Objektarchivierung).

- Mindestspeichergebühren

  Für einige S3-Speicherklassen gelten Mindestanforderungen an die Speicherdauer. Für Objekte, die vor Ablauf der Mindestdauer gelöscht, überschrieben oder aus diesen Klassen übergeben werden, wird eine anteilige Gebühr für die vorzeitige Übergabe oder das Löschen berechnet. Diese Mindestanforderungen an die Speicherdauer lauten wie folgt:

  - S3 Standard-IA und S3 One Zone-IA – 30 Tage
  - S3 Glacier Flexible Retrieval und S3 Glacier Instant Retrieval – 90 Tage
  - S3 Glacier Deep Archive – 180 Tage

  Weitere Informationen zu diesen Anforderungen finden Sie im Abschnitt Beschränkungen unter Übergang von Objekten mit S3-Lebenszyklus. Allgemeine S3-Preisinformationen finden Sie unter Amazon S3 S3-Preise und AWS Preisrechner.

- Kosten für die Lebenszyklusübergabe

  Immer wenn ein Objekt durch eine Lebenszyklusregel in eine andere Speicherklasse übergeben wird, zählt Amazon S3 diese Übergabe als eine Übergabeanforderung. Die Kosten für diese Übergabeanforderungen kommen zu den Kosten dieser Speicherklassen hinzu. Wenn Sie vorhaben, sehr viele Objekte zu übergeben, sollten Sie die Anforderungskosten für Übergaben in niedrigere Stufen berücksichtigen. Weitere Informationen finden Sie unter Amazon S3 – Preise.

Ich habe meine Bucket-Richtlinie aktualisiert, meine S3-Objekte werden jedoch noch immer aufgrund abgelaufener Lebenszyklusregeln gelöscht.

`Deny`-Anweisungen in einer Bucket-Richtlinie verhindern den Ablauf der in einer Lebenszyklusregel definierten Objekte nicht. Lebenszyklusaktionen (wie Übergaben oder der Ablauf) verwenden die S3-Operation `DeleteObject` nicht. Stattdessen werden S3-Lebenszyklusaktionen mithilfe interner S3-Endpunkte ausgeführt. (Weitere Informationen finden Sie unter Lebenszyklus und Protokollieren.)

Um zu verhindern, dass Ihre Lebenszyklusregel Aktionen ausführt, müssen Sie die Regel bearbeiten, löschen oder deaktivieren.

Kann ich S3-Objekte wiederherstellen, die aufgrund von S3-Lebenszyklusregeln abgelaufen sind?

Die einzige Möglichkeit, aufgrund des S3-Lebenszyklus abgelaufene Objekte wiederherzustellen, besteht in der Versionsverwaltung. Diese muss vor dem Ablauf der Objekte aktiviert sein. Sie können die Ablaufoperationen, die durch Lebenszyklusregeln ausgeführt werden, nicht rückgängig machen. Wenn Objekte aufgrund der geltenden S3-Lebenszyklusregeln dauerhaft gelöscht werden, können Sie diese Objekte nicht wiederherstellen. Informationen zum Aktivieren der Versionsverwaltung für einen Bucket finden Sie unter the section called “Beibehaltung mehrerer Versionen von Objekten”.

Wenn Sie die Versionsverwaltung auf den Bucket angewendet haben und die nicht aktuellen Versionen der Objekte noch intakt sind, können Sie frühere Versionen der abgelaufenen Objekte wiederherstellen. Weitere Informationen zum Verhalten der Aktionen von S3-Lebenszyklusregeln und zu Versionsverwaltungsstatus finden Sie in der Tabelle Lebenszyklus-Aktionen und der Versioning-Status eines Buckets in Elemente, die Lebenszyklus-Aktionen beschreiben.

> **Note**
>
> Wenn der S3-Bucket geschützt ist durch AWS Backup oder S3-Replikation, möglicherweise können Sie diese Funktionen auch verwenden, um Ihre abgelaufenen Objekte wiederherzustellen.

Wie kann ich ein Präfix aus meiner Lebenszyklusregel ausschließen?

S3 Lifecycle unterstützt das Ausschließen von Präfixen in Ihren Regeln nicht. Verwenden Sie stattdessen Tags, um alle Objekte zu kennzeichnen, die Sie in die Regel aufnehmen möchten.

Weitere Informationen zur Verwendung von Tags in Ihren Lebenszyklusregeln finden Sie unterthe section called “Archivierung aller Objekte innerhalb eines Tages nach der Erstellung”.

Wie kann ich mehrere Präfixe in meine Lebenszyklusregel aufnehmen?

S3 Lifecycle unterstützt nicht, mehrere Präfixe in Ihre Regeln aufzunehmen. Verwenden Sie stattdessen Tags, um alle Objekte zu kennzeichnen, die Sie in die Regel aufnehmen möchten. Weitere Informationen zur Verwendung von Tags in Ihren Lebenszyklusregeln finden Sie unterthe section called “Archivierung aller Objekte innerhalb eines Tages nach der Erstellung”.

Wenn Sie jedoch über ein oder mehrere Präfixe verfügen, die mit denselben Zeichen beginnen, können Sie all diese Präfixe in Ihre Regel aufnehmen, indem Sie im Filter ein Teilpräfix ohne abschließenden Schrägstrich (`/`) angeben. Nehmen wir zum Beispiel an, Sie haben diese Präfixe:

```
sales1999/
sales2000/
sales2001/
```

Um alle drei Präfixe in Ihre Regel aufzunehmen, geben Sie dies `<Prefix>sales</Prefix>` in Ihrer Lebenszyklusregel an.

Protokollierung und Überwachung in Amazon S3

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von Amazon S3 und Ihren AWS Lösungen. Wir empfehlen, Überwachungsdaten aus allen Teilen Ihrer AWS Lösung zu sammeln, damit Sie einen etwaigen Ausfall an mehreren Punkten leichter debuggen können. Bevor Sie mit der Überwachung von Amazon S3 beginnen, sollten Sie einen Überwachungsplan mit Antworten auf die folgenden Fragen erstellen:

- Was sind Ihre Ziele bei der Überwachung?
- Welche Ressourcen werden überwacht?
- Wie oft werden diese Ressourcen überwacht?
- Welche Überwachungstools werden verwendet?
- Wer soll die Überwachungsaufgaben ausführen?
- Wer soll benachrichtigt werden, wenn Fehler auftreten?

Weitere Informationen zur Protokollierung und Überwachung in Amazon S3 finden Sie in den folgenden Themen.

> **Note**
>
> Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter S3 Express One Zone und Arbeiten mit Verzeichnis-Buckets.

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von Amazon S3 und Ihren AWS Lösungen. Sie sollten Überwachungsdaten aus allen Teilen Ihrer AWS Lösung sammeln, damit Sie einen etwaigen Ausfall an mehreren Stellen leichter debuggen können. AWS bietet verschiedene Tools zur Überwachung Ihrer Amazon S3 S3-Ressourcen und zur Reaktion auf potenzielle Vorfälle.

CloudWatch Amazon-Alarme

Mithilfe von CloudWatch Amazon-Alarmen beobachten Sie eine einzelne Metrik über einen von Ihnen angegebenen Zeitraum. Wenn die Metrik einen bestimmten Schwellenwert überschreitet,

wird eine Benachrichtigung an ein SNS Thema oder eine AWS Auto Scaling Richtlinie von Amazon gesendet. CloudWatch Alarme lösen keine Aktionen aus, da sie sich in einem bestimmten Status befinden. Der Status muss sich stattdessen geändert haben und für eine festgelegte Anzahl an Zeiträumen aufrechterhalten worden sein. Weitere Informationen finden Sie unter Metriken mit Amazon überwachen CloudWatch.

AWS CloudTrail Logs

CloudTrail bietet eine Aufzeichnung der Aktionen, die von einem Benutzer, einer Rolle oder einem AWS Service in Amazon S3 ausgeführt wurden. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Amazon S3 gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln. Weitere Informationen finden Sie unter Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail.

Amazon GuardDuty

Amazon GuardDuty ist ein Service zur Bedrohungserkennung, der Ihre Konten, Container, Workloads und die Daten in Ihrer AWS Umgebung kontinuierlich überwacht, um potenzielle Bedrohungen oder Sicherheitsrisiken für Ihre S3-Buckets zu identifizieren. GuardDuty bietet außerdem umfangreiche Informationen zu den erkannten Bedrohungen. GuardDuty überwacht AWS CloudTrail Verwaltungsprotokolle auf Bedrohungen und veröffentlicht sicherheitsrelevante Informationen. Dazu gehören beispielsweise Faktoren einer API Anfrage, wie der Benutzer, der die Anfrage gestellt hat, der Standort, von dem aus die Anfrage gestellt wurde, und die spezifische API Anfrage, die in Ihrer Umgebung ungewöhnlich sein könnten. GuardDuty GuardDuty S3 Protection überwacht die von allen S3-Buckets in Ihrer Umgebung gesammelten S3-Datenereignisse CloudTrail und identifiziert potenziell anomales und bösartiges Verhalten.

Amazon-S3-Zugriffsprotokolle

Server-Zugriffsprotokolle enthalten detaillierte Aufzeichnungen über die Anfragen, die an einen Bucket gestellt wurden. Server-Zugriffsprotokolle sind für viele Anwendungen nützlich. Beispielsweise können Zugriffsprotokoll-Informationen bei Sicherheits- und Zugriffsprüfungen nützlich sein. Weitere Informationen finden Sie unter Protokollieren von Anfragen mit Server-Zugriffsprotokollierung.

AWS Trusted Advisor

Trusted Advisor stützt sich auf bewährte Verfahren, die wir bei der Betreuung von Hunderttausenden von Kunden gelernt haben. AWS Trusted Advisor untersucht Ihre AWS Umgebung und gibt dann Empfehlungen, wenn Möglichkeiten bestehen, Geld zu sparen, die

Systemverfügbarkeit und -leistung zu verbessern oder Sicherheitslücken zu schließen. Alle AWS Kunden haben Zugriff auf fünf Trusted Advisor Schecks. Kunden mit einem Business- oder Enterprise-Supportplan können alle Trusted Advisor Schecks einsehen.

Trusted Advisor hat die folgenden Amazon S3-bezogenen Prüfungen:

- Protokollierungskonfiguration von Amazon-S3-Buckets.
- Sicherheitsprüfungen für Amazon-S3-Buckets mit offenen Zugriffsberechtigungen.
- Fehlertoleranzprüfungen für Amazon-S3-Buckets, für die keine Versionsverwaltung aktiviert oder deren Versionsverwaltung ausgesetzt ist

Weitere Informationen finden Sie unter AWS Trusted Advisor im AWS Support - Benutzerhandbuch.

Die folgenden bewährten Sicherheitsmethoden umfassen ebenfalls die Protokollierung und Überwachung:

- Identify and audit all your Amazon S3 buckets
- Implement monitoring using Amazon Web Services monitoring tools
- Aktivieren AWS Config
- Enable Amazon S3 server access logging
- Use CloudTrail
- Monitor Amazon Web Services security advisories

Themen

- Überwachungstools
- Protokollierungsoptionen für Amazon S3
- Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail
- Protokollieren von Anfragen mit Server-Zugriffsprotokollierung
- Metriken mit Amazon überwachen CloudWatch
- Amazon-S3-Ereignis-Benachrichtigungen
- Bewerten Ihrer Speicheraktivität und -nutzung mit Amazon S3 Storage Lens
- Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory

Überwachungstools

AWS bietet verschiedene Tools, mit denen Sie Amazon S3 überwachen können. Sie können einige dieser Tools so konfigurieren, dass diese die Überwachung für Sie übernehmen, während bei anderen Tools ein manuelles Eingreifen nötig ist. Wir empfehlen, dass Sie die Überwachungsaufgaben möglichst automatisieren.

Automatisierte Überwachungstools

Sie können die folgenden automatisierten Tools zur Überwachung von Amazon S3 verwenden und auftretende Probleme melden:

- Amazon CloudWatch Alarms — Überwachen Sie eine einzelne Metrik über einen von Ihnen angegebenen Zeitraum und führen Sie eine oder mehrere Aktionen aus, die auf dem Wert der Metrik im Verhältnis zu einem bestimmten Schwellenwert über mehrere Zeiträume basieren. Die Aktion ist eine Benachrichtigung, die an ein Amazon Simple Notification Service (AmazonSNS) - Thema oder eine Amazon EC2 Auto Scaling Scaling-Richtlinie gesendet wird. CloudWatch Alarme lösen keine Aktionen aus, nur weil sie sich in einem bestimmten Status befinden. Der Status muss sich geändert haben und für eine festgelegte Anzahl an Zeiträumen aufrechterhalten worden sein. Weitere Informationen finden Sie unter Metriken mit Amazon überwachen CloudWatch.
- AWS CloudTrail Protokollüberwachung — Teilen Sie Protokolldateien zwischen Konten, überwachen CloudTrail Sie Protokolldateien in Echtzeit, indem Sie sie an CloudWatch Logs senden, schreiben Sie Protokollverarbeitungsanwendungen in Java und überprüfen Sie, ob sich Ihre Protokolldateien nach der Lieferung von CloudTrail nicht geändert haben. Weitere Informationen finden Sie unter Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail.

Manuelle Überwachungstools

Ein weiterer wichtiger Teil der Überwachung von Amazon S3 ist die manuelle Überwachung der Elemente, die von den CloudWatch Alarmen nicht abgedeckt werden. Die Amazon S3, CloudWatch Trusted Advisor, und andere AWS Management Console Dashboards bieten einen at-a-glance Überblick über den Zustand Ihrer AWS Umgebung. Möglicherweise wollen Sie die Server-Zugriffsprotokollierung aktivieren, die Zugriffsanforderungen für den Bucket verfolgt. Jeder Zugriffsprotokolldatensatz enthält Details über eine Zugriffsanforderung, z. B. Auftraggeber, Bucket-Name, Anforderungszeit, Anforderungsaktion, Antwortstatus und Fehlercode, falls vorhanden. Weitere Informationen finden Sie unter Protokollieren von Anfragen mit Server-Zugriffsprotokollierung.

- Auf dem Amazon-S3-Dashboard wird Folgendes angezeigt:
  - Ihre Buckets und die Eigenschaften und Objekte, die sie enthalten.
- Auf der CloudWatch Startseite wird Folgendes angezeigt:
  - Aktuelle Alarme und Status
  - Diagramme mit Alarmen und Ressourcen
  - Servicestatus

  Darüber hinaus können CloudWatch Sie Folgendes verwenden:
  - Erstellen von benutzerdefinierten Dashboards zur Überwachung des gewünschten Services.
  - Aufzeichnen von Metrikdaten, um Probleme zu beheben und Trends zu erkennen.
  - Suchen und durchsuchen Sie alle Ihre AWS Ressourcenmetriken.
  - Erstellen und Bearbeiten von Alarmen, um über Probleme benachrichtigt zu werden.
- AWS Trusted Advisor kann Ihnen helfen, Ihre AWS Ressourcen zu überwachen, um Leistung, Zuverlässigkeit, Sicherheit und Wirtschaftlichkeit zu verbessern. Vier Trusted Advisor -Prüfungen stehen allen Benutzern zur Verfügung; mehr als 50 Überprüfungen stehen Benutzern mit einem Business- oder Enterprise-Supportplan zur Verfügung. Weitere Informationen finden Sie unter AWS Trusted Advisor.

  Trusted Advisor hat diese Prüfungen, die sich auf Amazon S3 beziehen:
  - Überprüfungen der Protokollierungskonfiguration von Amazon-S3-Buckets.
  - Sicherheitsprüfungen für Amazon-S3-Buckets mit offenen Zugriffsberechtigungen.
  - Fehlertoleranzprüfungen für Amazon-S3-Buckets, für die kein Versioning aktiviert oder deren Versioning ausgesetzt ist.

Protokollierungsoptionen für Amazon S3

Sie können die Aktionen aufzeichnen, die von Benutzern, Rollen oder AWS-Services Amazon S3 S3-Ressourcen ausgeführt werden, und Protokollaufzeichnungen für Prüf- und Compliance-Zwecke verwalten. Dazu können Sie die Serverzugriffsprotokollierung, die AWS CloudTrail -Protokollierung oder eine Kombination aus beiden verwenden. Wir empfehlen Ihnen, CloudTrail für die Protokollierung von Aktionen auf Bucket- und Objektebene für Ihre Amazon S3 S3-Ressourcen zu verwenden. Weitere Informationen zu jeder Option finden Sie in folgenden Abschnitten:

- Protokollieren von Anfragen mit Server-Zugriffsprotokollierung

- Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail

In der folgenden Tabelle sind die wichtigsten Eigenschaften von CloudTrail Protokollen und Amazon S3-Serverzugriffsprotokollen aufgeführt. Sehen Sie sich die Tabelle und die Hinweise an, um sicherzustellen, dass dies Ihren Sicherheitsanforderungen CloudTrail entspricht.

Protokolleigenschaften	AWS CloudTrail	Amazon-S3-Serverprotokolle
Kann an andere Systeme weitergeleitet werden (Amazon CloudWatch Logs, Amazon CloudWatch Events)	Ja	Nein
Protokolle an mehreren Zielen bereitstellen (Beispiel: dieselben Protokolle an zwei verschiedene Buckets senden)	Ja	Nein
Protokolle für eine Teilmenge von Objekten aktivieren (Präfix)	Ja	Nein
Kontoübergreifende Protokoll bereitstellung (Ziel- und Quell-Bucket in Besitz von verschied enen Konten)	Ja	Nein
Integritätsprüfung der Protokoll datei mit digitaler Signatur oder Hashing	Ja	Nein
Standardeinstellung oder Auswahl der Verschlüsselung für Protokolldateien	Ja	Nein
Objektoperationen (mithilfe von Amazon S3APIs)	Ja	Ja

Protokolleigenschaften	AWS CloudTrail	Amazon-S3-Serverprotokolle
Bucket-Operationen (mithilfe von Amazon S3APIs)	Ja	Ja
Durchsuchbare UI für Protokolle	Ja	Nein
Felder für Objektsperrparameter, Amazon S3 Select-Eigenschaften für Protokolldatensätze	Ja	Nein
Felder für `Object Size`, `Total Time`, `Turn-Around Time` und `HTTP Referer` für Protokolldatensätze	Nein	Ja
Lebenszyklusübertragungen, Ablaufaktionen, Wiederherstellungen	Nein	Ja
Protokollieren von Schlüsseln in einer Batch-Delete-Operation	Nein	Ja
Authentifizierungsfehler[1]	Nein	Ja
Konten, an die Protokolle geliefert werden	Bucket-Eigentümer[2] und Auftraggeber	Nur Bucket-Eigentümer
Performance and Cost	AWS CloudTrail	Amazon S3 Server Logs
Preis	Verwaltungsereignisse (erste Bereitstellung) sind kostenlos. Für Datenereignisse fällt zusätzlich zur Speicherung der Protokolle eine Gebühr an	Keine Zusatzkosten neben der Speicherung der Protokolle

Protokolleigenschaften	AWS CloudTrail	Amazon-S3-Serverprotokolle
Geschwindigkeit der Protokoll bereitstellung	Datenereignisse alle 5 Minuten; Verwaltungsereignisse alle 15 Minuten	Innerhalb weniger Stunden
Protokollformat	JSON	Protokolldatei mit durch Leerzeichen getrennten, durch neue Zeilen getrennten Datensätzen

Hinweise

1. CloudTrail liefert keine Protokolle für Anfragen, bei denen die Authentifizierung fehlschlägt (bei denen die angegebenen Anmeldeinformationen nicht gültig sind). Es enthält jedoch Protokolle für Anforderungen, deren Authentifizierung fehlschlägt (`AccessDenied`), und Anforderungen, die von anonymen Benutzern gestellt werden.
2. Der S3-Bucket-Besitzer erhält CloudTrail Protokolle, wenn das Konto keinen vollen Zugriff auf das Objekt in der Anfrage hat. Weitere Informationen finden Sie unter Amazon S3 S3-Aktionen auf Objektebene in kontoübergreifenden Szenarien.
3. S3 unterstützt nicht die Übermittlung von CloudTrail Protokollen oder Serverzugriffsprotokollen an den Anforderer oder den Bucket-Besitzer für VPC Endpunktanfragen, wenn die VPC Endpunktrichtlinie sie ablehnt, oder für Anfragen, die fehlschlagen, bevor die VPC Richtlinie ausgewertet wurde.

Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail

Amazon S3 ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem ausgeführten Aktionen bereitstellt AWS-Service. CloudTrail erfasst alle API Aufrufe für Amazon S3 als Ereignisse. Zu den erfassten Anrufen gehören Anrufe von der Amazon S3 S3-Konsole und Code-Aufrufe an die Amazon S3 API S3-Operationen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Amazon S3 gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, den Zeitpunkt der Anfrage und weitere Details ermitteln.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Anmeldeinformationen des Root-Benutzers oder des Benutzers gestellt wurde.
- Ob die Anfrage im Namen eines IAM Identity Center-Benutzers gestellt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anforderung aus einem anderen AWS-Service gesendet wurde.

CloudTrail ist in Ihrem aktiv AWS-Konto , wenn Sie das Konto erstellen, und Sie haben automatisch Zugriff auf den CloudTrail Eventverlauf. Der CloudTrail Ereignisverlauf bietet eine einsehbare, durchsuchbare, herunterladbare und unveränderliche Aufzeichnung der aufgezeichneten Verwaltungsereignisse der letzten 90 Tage in einem. AWS-Region Weitere Informationen finden Sie im AWS CloudTrail Benutzerhandbuch unter Arbeiten mit dem CloudTrail Ereignisverlauf. Für die Anzeige des Eventverlaufs CloudTrail fallen keine Gebühren an.

Für eine fortlaufende Aufzeichnung der Ereignisse in AWS-Konto den letzten 90 Tagen erstellen Sie einen Trail- oder CloudTrailLake-Event-Datenspeicher.

CloudTrail Pfade

Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Alle mit dem erstellten Pfade AWS Management Console sind regionsübergreifend. Sie können einen Pfad mit einer oder mehreren Regionen erstellen, indem Sie den verwenden. AWS CLI Es wird empfohlen, einen Trail mit mehreren Regionen zu erstellen, da Sie alle Aktivitäten in Ihrem Konto AWS-Regionen erfassen. Wenn du einen Trail mit nur einer Region erstellst, kannst du dir nur die Ereignisse ansehen, die in den Trails protokolliert wurden. AWS-Region Weitere Informationen zu Trails finden Sie unter Einen Trail für Sie erstellen AWS-Konto und Einen Trail für eine Organisation erstellen im AWS CloudTrail Benutzerhandbuch.

Sie können eine Kopie Ihrer laufenden Verwaltungsereignisse kostenlos an Ihren Amazon S3 S3-Bucket senden, CloudTrail indem Sie einen Trail erstellen. Es fallen jedoch Amazon S3 S3-Speichergebühren an. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter AWS CloudTrail Preise. Informationen zu Amazon-S3-Preisen finden Sie unter Amazon S3-Preise.

CloudTrail Datenspeicher für Ereignisse in Lake

CloudTrail Mit Lake können Sie SQL basierte Abfragen zu Ihren Ereignissen ausführen. CloudTrail Lake konvertiert bestehende Ereignisse im zeilenbasierten JSON Format in das

ORCApache-Format. ORCist ein spaltenförmiges Speicherformat, das für den schnellen Abruf von Daten optimiert ist. Die Ereignisse werden in Ereignisdatenspeichern zusammengefasst, bei denen es sich um unveränderliche Sammlungen von Ereignissen handelt, die auf Kriterien basieren, die Sie mit Hilfe von erweiterten Ereignisselektoren auswählen. Die Selektoren, die Sie auf einen Ereignisdatenspeicher anwenden, steuern, welche Ereignisse bestehen bleiben und für Sie zur Abfrage verfügbar sind. Weitere Informationen zu CloudTrail Lake finden Sie im AWS CloudTrail Benutzerhandbuch unter Arbeiten mit AWS CloudTrail Lake.

CloudTrail Für das Speichern und Abfragen von Ereignisdaten in Lake fallen Kosten an. Beim Erstellen eines Ereignisdatenspeichers wählen Sie die Preisoption aus, die für den Ereignisdatenspeicher genutzt werden soll. Die Preisoption bestimmt die Kosten für die Erfassung und Speicherung von Ereignissen sowie die standardmäßige und maximale Aufbewahrungsdauer für den Ereignisdatenspeicher. Weitere Informationen zur Preisgestaltung finden Sie unter CloudTrail AWS CloudTrail Preisgestaltung.

Sie können Ihre Protokolldateien beliebig lange im Bucket speichern. Sie können aber auch Amazon S3-Lebenszyklusregeln aufstellen, anhand derer die Protokolldateien automatisch archiviert oder gelöscht werden. Standardmäßig werden Ihre Protokolldateien mithilfe der serverseitigen Amazon S3 S3-Verschlüsselung (SSE) verschlüsselt.

Verwenden von CloudTrail Protokollen mit Amazon S3 S3-Serverzugriffsprotokollen und CloudWatch Protokollen

AWS CloudTrail Protokolle zeichnen Aktionen auf, die von einem Benutzer, einer Rolle oder einem AWS Service in Amazon S3 ausgeführt wurden, während Amazon S3-Serverzugriffsprotokolle detaillierte Aufzeichnungen über die Anfragen enthalten, die an einen S3-Bucket gestellt werden. Weitere Informationen zur Funktionsweise der unterschiedlichen Protokolle und ihren Eigenschaften, ihrer Leistung und ihrer Kosten finden Sie unter the section called “Protokollierungsoptionen”.

Sie können AWS CloudTrail Protokolle zusammen mit Serverzugriffsprotokollen für Amazon S3 verwenden. CloudTrail Protokolle bieten Ihnen eine detaillierte API Nachverfolgung von Amazon S3 S3-Vorgängen auf Bucket- und Objektebene. Serverzugriffsprotokolle für Amazon S3 bieten Ihnen Einblick in Vorgänge auf Objektebene mit Ihren Daten in Amazon S3. Weitere Informationen zu Server-Zugriffsprotokollen finden Sie unter Protokollieren von Anfragen mit Server-Zugriffsprotokollierung.

Sie können CloudTrail Protokolle auch zusammen mit Amazon CloudWatch für Amazon S3 verwenden. CloudTrail Durch die Integration mit CloudWatch Logs werden API Aktivitäten auf S3-

Bucket-Ebene, die von erfasst wurden CloudTrail , in einem CloudWatch Protokollstream in der von Ihnen CloudWatch angegebenen Protokollgruppe bereitgestellt. Sie können CloudWatch Alarme zur Überwachung bestimmter API Aktivitäten erstellen und E-Mail-Benachrichtigungen erhalten, wenn die bestimmte API Aktivität auftritt. Weitere Informationen zu CloudWatch Alarmen zur Überwachung bestimmter API Aktivitäten finden Sie im AWS CloudTrail Benutzerhandbuch. Weitere Informationen zur Verwendung CloudWatch mit Amazon S3 finden Sie unterMetriken mit Amazon überwachen CloudWatch.

> **Note**
>
> S3 unterstützt nicht die Übermittlung von CloudTrail Protokollen an den Anforderer oder den Bucket-Besitzer für VPC Endpunktanfragen, wenn die VPC Endpunktrichtlinie diese ablehnt.

CloudTrail Nachverfolgung mit Amazon S3 SOAP API S3-Anrufen

CloudTrail verfolgt Amazon S3 SOAP API S3-Anrufe. Die Amazon S3 SOAP S3-Unterstützung HTTP ist veraltet, aber sie ist weiterhin verfügbar. HTTPS Weitere Informationen zur Amazon S3 SOAP S3-Unterstützung finden Sie im Anhang: SOAP API in der Amazon S3 API S3-Referenz.

> **Important**
>
> Neuere Amazon S3 S3-Funktionen werden nicht unterstütztSOAP. Wir empfehlen Ihnen, entweder den REST API oder den zu verwenden AWS SDKs.

Die folgende Tabelle zeigt Amazon S3 SOAP S3-Aktionen, die durch CloudTrail Protokollierung verfolgt wurden.

SOAPAPIName	APIName des Ereignisses, der im CloudTrail Protokoll verwendet wird
ListAllMyBuckets	`ListBuckets`
CreateBucket	`CreateBucket`
DeleteBucket	`DeleteBucket`

SOAPAPIName	APIName des Ereignisses, der im CloudTrail Protokoll verwendet wird
GetBucketAccessControlPolicy	`GetBucketAcl`
SetBucketAccessControlPolicy	`PutBucketAcl`
GetBucketLoggingStatus	`GetBucketLogging`
SetBucketLoggingStatus	`PutBucketLogging`

Weitere Informationen zu Amazon S3 CloudTrail und Amazon S3 finden Sie in den folgenden Themen:

Themen

- Amazon S3 CloudTrail S3-Ereignisse
- CloudTrail Protokolldateieinträge für Amazon S3 und S3 auf Outposts
- Aktivieren der CloudTrail Ereignisprotokollierung für S3-Buckets und -Objekte
- Identifizieren von Amazon S3 S3-Anfragen mithilfe von CloudTrail

Amazon S3 CloudTrail S3-Ereignisse

> **Important**
>
> Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist in den AWS CloudTrail Protokollen, im S3-Inventar, in der S3-Speicherlinse, in der Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header im AWS Command Line Interface und AWS SDKs verfügbar. Weitere Informationen finden Sie unter Standardverschlüsselung FAQ.

Dieser Abschnitt enthält Informationen zu den Ereignissen, die S3 protokolliert CloudTrail.

Amazon S3 S3-Datenereignisse in CloudTrail

Datenereignisse liefern Informationen über die Ressourcenoperationen, die auf oder in einer Ressource ausgeführt werden (z. B. Lesen oder Schreiben in ein Amazon-S3-Objekt). Sie werden auch als Vorgänge auf Datenebene bezeichnet. Datenereignisse sind oft Aktivitäten mit hohem Volume. Protokolliert standardmäßig CloudTrail keine Datenereignisse. Der CloudTrail Ereignisverlauf zeichnet keine Datenereignisse auf.

Für Datenereignisse werden zusätzliche Gebühren fällig. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter AWS CloudTrail Preisgestaltung.

Sie können Datenereignisse für die Amazon S3 S3-Ressourcentypen mithilfe der CloudTrail Konsole oder mithilfe von CloudTrail API Vorgängen protokollieren. AWS CLI Weitere Informationen zum Protokollieren von Datenereignissen finden Sie unter Protokollieren von Datenereignissen mit der AWS Management Console und Protokollieren von Datenereignissen mit dem AWS Command Line Interface im AWS CloudTrail Benutzerhandbuch.

In der folgenden Tabelle sind die Amazon S3 S3-Ressourcentypen aufgeführt, für die Sie Datenereignisse protokollieren können. In der Spalte Datenereignistyp (Konsole) wird der Wert angezeigt, den Sie aus der Liste Datenereignistyp auf der CloudTrail Konsole auswählen können. In der Wertspalte resources.type wird der `resources.type` Wert angezeigt, den Sie bei der Konfiguration erweiterter Event-Selektoren mithilfe von oder angeben würden. AWS CLI CloudTrail APIs In der CloudTrail Spalte APIsProtokollierte Daten werden die API Aufrufe angezeigt, die CloudTrail für den Ressourcentyp protokolliert wurden.

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten, die APIs protokolliert wurden CloudTrail
S3	`AWS::S3::Object`	• AbortMultipartUpload • CompleteMultipartUpload • CopyObject • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • DeleteObjects • GetObject

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten, die APIs protokolliert wurden CloudTrail
		• GetObjectAcl • GetObjectAttributes • GetObjectLegalHold • GetObjectRetention • GetObjectTagging • GetObjectTorrent • HeadObject • ListMultipartUploads • ListObjectVersions • ListObjects • ListParts • PutObject • PutObjectAcl • PutObjectLegalHold • PutObjectRetention • PutObjectTagging • RestoreObject • SelectObjectContent • UploadPart • UploadPartCopy

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten, die APIs protokolliert wurden CloudTrail
S3 Express Eine Zone	`AWS::S3Express::Object`	• AbortMultipartUpload • CompleteMultipartUpload • CreateSession • CopyObject • CreateMultipartUpload • DeleteObject • DeleteObjects • GetObject • GetObjectAttributes • HeadBucket • HeadObject • ListObjectsV2 • ListParts • PutObject • UploadPart • UploadPartCopy

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten, die APIs protokolliert wurden CloudTrail
S3-Zugangspunkt	`AWS::S3::Access Point`	• AbortMultipartUpload • CompleteMultipartUpload • CopyObject (nur Kopien in derselben Region) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetBucketAcl • GetBucketCors • GetBucketLocation • GetBucketNotificationConfiguration • GetBucketPolicy • GetObject • GetObjectAcl • GetObjectAttributes • GetObjectLegalHold • GetObjectRetention • GetObjectTagging • HeadBucket • HeadObject • ListMultipartUploads • ListObjects • ListObjectsV2 • ListObjectVersions • ListParts • Presign • PutObject

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten, die APIs protokolliert wurden CloudTrail
		• PutObjectLegalHold • PutObjectRetention • PutObjectAcl • PutObjectTagging • RestoreObject • UploadPart • UploadPartCopy (nur Kopien in derselben Region)

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten, die APIs protokolliert wurden CloudTrail
S3 Objekt Lambda	`AWS::S3ObjectLambda::AccessPoint`	• AbortMultipartUpload • CompleteMultipartUpload • CopyObject (nur Kopien in derselben Region) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetObject • GetObjectAcl • GetObjectLegalHold • GetObjectRetention • GetObjectTagging • HeadObject • ListMultipartUploads • ListObjects • ListObjectVersions • ListParts • PutObject • PutObjectLegalHold • PutObjectRetention • PutObjectAcl • PutObjectTagging • RestoreObject • UploadPart • WriteGetObjectResponse

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten, die APIs protokolliert wurden CloudTrail
S3-Outposts	`AWS::S3Outposts::Object`	• AbortMultipartUpload • CompleteMultipartUpload • CopyObject (nur Kopien in derselben Region) • CreateMultipartUpload • DeleteObject • DeleteObjectTagging • GetObject • GetObjectTagging • HeadObject • ListMultipartUploads • ListObjects • ListObjectsV2 • ListParts • PutObject • PutObjectTagging • UploadPart • UploadPartCopy

Sie können erweiterte Event-Selektoren so konfigurieren, dass sie nach den `resources.ARN` Feldern `eventNamereadOnly`, und filtern, sodass nur die Ereignisse protokolliert werden, die für Sie wichtig sind. Weitere Informationen zu diesen Feldern finden Sie unter AdvancedFieldSelectorin der AWS CloudTrail APIReferenz.

Amazon S3 S3-Managementereignisse in CloudTrail

Amazon S3 protokolliert alle Operationen auf der Kontrollebene als Verwaltungsereignisse. Weitere Informationen zu API S3-Vorgängen finden Sie in der Amazon S3 API S3-Referenz.

So werden Anfragen an Amazon S3 CloudTrail erfasst

Standardmäßig CloudTrail protokolliert API S3-Aufrufe auf Bucket-Ebene, die in den letzten 90 Tagen getätigt wurden, protokolliert jedoch keine Anfragen an Objekte. Aufrufe auf Bucket-Ebene sind Ereignisse wie `CreateBucket`, `DeleteBucket`, `PutBucketLifecycle`, `PutBucketPolicy` usw. Sie können Ereignisse auf Bucket-Ebene auf der Konsole sehen. CloudTrail Sie können dort jedoch keine Datenereignisse (Amazon S3 S3-Aufrufe auf Objektebene) anzeigen — Sie müssen sie analysieren oder die Protokolle danach abfragen. CloudTrail

Amazon S3 S3-Aktionen auf Kontoebene, die durch Protokollierung verfolgt werden CloudTrail

CloudTrail protokolliert Aktionen auf Kontoebene. Amazon S3 S3-Datensätze werden zusammen mit anderen AWS-Service Datensätzen in eine Protokolldatei geschrieben. CloudTrail bestimmt anhand eines Zeitraums und der Dateigröße, wann eine neue Datei erstellt und in diese geschrieben werden soll.

In den Tabellen in diesem Abschnitt sind die Amazon S3 S3-Aktionen auf Kontoebene aufgeführt, die für die Protokollierung von unterstützt werden. CloudTrail

APIAktionen auf Amazon S3 S3-Kontoebene, die durch CloudTrail Protokollierung verfolgt werden, werden als die folgenden Ereignisnamen angezeigt. Die Namen der CloudTrail Ereignisse unterscheiden sich vom API Aktionsnamen. Zum Beispiel DeletePublicAccessBlock ist DeleteAccountPublicAccessBlock.

- DeleteAccountPublicAccessBlock
- GetAccountPublicAccessBlock
- PutAccountPublicAccessBlock

Amazon S3 S3-Aktionen auf Bucket-Ebene, die durch Protokollierung verfolgt werden CloudTrail

Standardmäßig werden Aktionen auf Bucket-Ebene für allgemeine Buckets CloudTrail protokolliert. Amazon S3 S3-Datensätze werden zusammen mit anderen AWS Serviceaufzeichnungen in eine Protokolldatei geschrieben. CloudTrail bestimmt anhand eines Zeitraums und der Dateigröße, wann eine neue Datei erstellt und in diese geschrieben werden soll.

In diesem Abschnitt sind die Amazon S3 S3-Aktionen auf Bucket-Ebene aufgeführt, die für die Protokollierung von unterstützt werden. CloudTrail

APIAktionen auf Amazon S3 S3-Bucket-Ebene, die durch CloudTrail Protokollierung verfolgt werden, werden als die folgenden Ereignisnamen angezeigt. In einigen Fällen unterscheidet sich der Name des CloudTrail Ereignisses vom API Aktionsnamen. Zum Beispie, `PutBucketLifecycleConfiguration` ist `PutBucketLifecycle`.

- CreateBucket
- DeleteBucket
- DeleteBucketAnalyticsConfiguration
- DeleteBucketCors
- DeleteBucketEncryption
- DeleteBucketIntelligentTieringConfiguration
- DeleteBucketInventoryConfiguration
- DeleteBucketLifecycle
- DeleteBucketMetricsConfiguration
- DeleteBucketOwnershipControls
- DeleteBucketPolicy
- DeleteBucketPublicAccessBlock
- DeleteBucketReplication
- DeleteBucketTagging
- GetAccelerateConfiguration
- GetBucketAcl
- GetBucketAnalyticsConfiguration
- GetBucketCors
- GetBucketEncryption
- GetBucketIntelligentTieringConfiguration
- GetBucketInventoryConfiguration
- GetBucketLifecycle
- GetBucketLocation
- GetBucketLogging

- GetBucketMetricsConfiguration
- GetBucketNotification
- GetBucketObjectLockConfiguration
- GetBucketOwnershipControls
- GetBucketPolicy
- GetBucketPolicyStatus
- GetBucketPublicAccessBlock
- GetBucketReplication
- GetBucketRequestPayment
- GetBucketTagging
- GetBucketVersioning
- GetBucketWebsite
- HeadBucket
- ListBuckets
- PutAccelerateConfiguration
- PutBucketAcl
- PutBucketAnalyticsConfiguration
- PutBucketCors
- PutBucketEncryption
- PutBucketIntelligentTieringConfiguration
- PutBucketInventoryConfiguration
- PutBucketLifecycle
- PutBucketLogging
- PutBucketMetricsConfiguration
- PutBucketNotification
- PutBucketObjectLockConfiguration
- PutBucketOwnershipControls
- PutBucketPolicy
- PutBucketPublicAccessBlock

- PutBucketReplication
- PutBucketRequestPayment
- PutBucketTagging
- PutBucketVersioning
- PutBucketWebsite

Zusätzlich zu diesen API Operationen können Sie auch die OPTIONS Aktion auf Objektebene. Diese Aktion wird in der CloudTrail Protokollierung wie eine Aktion auf Bucket-Ebene behandelt, da die Aktion die CORS Konfiguration eines Buckets überprüft.

Amazon S3 Express One Zone-Aktionen auf Bucket-Ebene (regionaler API Endpunkt), die durch Protokollierung verfolgt werden CloudTrail

Standardmäßig werden Aktionen auf Bucket-Ebene für Verzeichnis-Buckets als Verwaltungsereignisse CloudTrail protokolliert. Die `eventsource` für CloudTrail Verwaltungsereignisse für S3 Express One Zone ist. `s3express.amazonaws.com`

Die folgenden regionalen API Endpunktvorgänge werden protokolliert CloudTrail.

- CreateBucket
- DeleteBucket
- DeleteBucketPolicy
- GetBucketPolicy
- PutBucketPolicy
- ListDirectoryBuckets
- PutBucketEncryption
- GetBucketEncryption
- DeleteBucketEncryption

Weitere Informationen finden Sie unter Protokollierung mit AWS CloudTrail für S3 Express One Zone

Amazon S3 S3-Aktionen auf Objektebene in kontoübergreifenden Szenarien

Im Folgenden finden Sie spezielle Anwendungsfälle, die API Aufrufe auf Objektebene in kontoübergreifenden Szenarien und die Art und Weise, wie Protokolle gemeldet werden, betreffen.

CloudTrail CloudTrail übermittelt Protokolle an den Anforderer (das Konto, das den API Anruf getätigt hat), außer in Fällen, in denen der Zugriff verweigert wurde, in denen Protokolleinträge geschwärzt oder weggelassen werden. Bei der Einrichtung von kontoübergreifendem Zugriff sehen Sie sich die Beispiele in diesem Abschnitt an.

> **Note**
>
> In den Beispielen wird davon ausgegangen, dass die CloudTrail Protokolle entsprechend konfiguriert sind.

Beispiel 1: CloudTrail Liefert Logs an den Bucket-Besitzer

CloudTrail übermittelt Protokolle an den Bucket-Besitzer, auch wenn der Bucket-Besitzer keine Berechtigungen für dieselbe API Objektoperation besitzt. Sehen Sie sich das folgende kontenübergreifende Szenario vor:

- Konto A ist Eigentümer des Buckets.
- Konto B (Anforderer) versucht, auf ein Objekt in diesem Bucket zuzugreifen.
- Konto C besitzt das Objekt. Konto C kann dasselbe Konto wie Konto A sein oder auch nicht.

> **Note**
>
> CloudTrail übermittelt dem Anforderer (Konto B) immer API Logs auf Objektebene. Darüber hinaus werden dieselben Protokolle CloudTrail auch dann an den Bucket-Besitzer (Konto A) übermittelt, wenn der Bucket-Besitzer das Objekt (Konto C) nicht besitzt oder keine Berechtigungen für dieselben API Operationen an diesem Objekt besitzt.

Beispiel 2: E-Mail-Adressen, die bei der Objekteinstellung verwendet werden, werden CloudTrail nicht häufig verwendet ACLs

Sehen Sie sich das folgende kontenübergreifende Szenario vor:

- Konto A ist Eigentümer des Buckets.
- Konto B (der Antragsteller) sendet mithilfe einer E-Mail-Adresse eine Anfrage zur Einrichtung ACL einer Objektzuweisung. Weitere Informationen zu finden Sie ACLs unterÜbersicht über die Zugriffskontrollliste (ACL).

Der Anforderer erhält die Protokolle zusammen mit der E-Mail-Information. Der Bucket-Besitzer erhält jedoch — sofern er berechtigt ist, Logs zu empfangen, wie in Beispiel 1 — das CloudTrail Protokoll, das das Ereignis meldet. Der Bucket-Besitzer erhält jedoch nicht die ACL Konfigurationsinformationen, insbesondere die E-Mail-Adresse des Empfängers und den Zuschuss. Die einzige Information, die das Protokoll dem Bucket-Besitzer mitteilt, ist, dass ein ACL API Anruf von Konto B getätigt wurde.

CloudTrail Protokolldateieinträge für Amazon S3 und S3 auf Outposts

> **Important**
>
> Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist in den AWS CloudTrail Protokollen, im S3-Inventar, in der S3-Speicherlinse, in der Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header im AWS Command Line Interface und AWS SDKs verfügbar. Weitere Informationen finden Sie unter Standardverschlüsselung FAQ.

Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über den angeforderten API Vorgang, Datum und Uhrzeit des Vorgangs, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API Aufrufe, sodass Ereignisse nicht in einer bestimmten Reihenfolge angezeigt werden.

> **Note**
>
> Beispiele für CloudTrail Protokolldateien für Amazon S3 Express One Zone finden Sie unter Beispiele für CloudTrail Protokolldateien für S3 Express One Zone.

Weitere Informationen finden Sie in den folgenden Beispielen.

Themen

- Beispiel: CloudTrail Protokolldateieintrag für Amazon S3

Beispiel: CloudTrail Protokolldateieintrag für Amazon S3

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die GET Dienst, PutBucketAcl, und GetBucketVersioningAktionen.


```
{
    "Records": [
    {
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "111122223333",
            "arn": "arn:aws:iam::111122223333:user/myUserName",
            "accountId": "111122223333",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "myUserName"
        },
        "eventTime": "2019-02-01T03:18:19Z",
        "eventSource": "s3.amazonaws.com",
        "eventName": "ListBuckets",
        "awsRegion": "us-west-2",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "[]",
        "requestParameters": {
            "host": [
                "s3.us-west-2.amazonaws.com"
            ]
        },
        "responseElements": null,
        "additionalEventData": {
            "SignatureVersion": "SigV2",
            "AuthenticationMethod": "QueryString",
            "aclRequired": "Yes"
    },
        "requestID": "47B8E8D397DCE7A6",
        "eventID": "cdc4b7ed-e171-4cef-975a-ad829d4123e8",
        "eventType": "AwsApiCall",
        "recipientAccountId": "444455556666",
        "tlsDetails": {
            "tlsVersion": "TLSv1.2",
            "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
            "clientProvidedHostHeader": "s3.amazonaws.com"
    }
```

```
    },
    {
       "eventVersion": "1.03",
       "userIdentity": {
            "type": "IAMUser",
            "principalId": "111122223333",
            "arn": "arn:aws:iam::111122223333:user/myUserName",
            "accountId": "111122223333",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "myUserName"
        },
      "eventTime": "2019-02-01T03:22:33Z",
      "eventSource": "s3.amazonaws.com",
      "eventName": "PutBucketAcl",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "",
      "userAgent": "[]",
      "requestParameters": {
          "bucketName": "",
          "AccessControlPolicy": {
              "AccessControlList": {
                  "Grant": {
                      "Grantee": {
                          "xsi:type": "CanonicalUser",
                          "xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",
                          "ID":
"d25639fbe9c19cd30a4c0f43fbf00e2d3f96400a9aa8dabfbbebe1906Example"
                       },
                      "Permission": "FULL_CONTROL"
                   }
              },
              "xmlns": "http://s3.amazonaws.com/doc/2006-03-01/",
              "Owner": {
                  "ID":
"d25639fbe9c19cd30a4c0f43fbf00e2d3f96400a9aa8dabfbbebe1906Example"
              }
          },
          "host": [
              "s3.us-west-2.amazonaws.com"
          ],
          "acl": [
              ""
          ]
      },
```

```
      "responseElements": null,
      "additionalEventData": {
          "SignatureVersion": "SigV4",
          "CipherSuite": "ECDHE-RSA-AES128-SHA",
          "AuthenticationMethod": "AuthHeader"
      },
      "requestID": "BD8798EACDD16751",
      "eventID": "607b9532-1423-41c7-b048-ec2641693c47",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333",
      "tlsDetails": {
            "tlsVersion": "TLSv1.2",
            "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
            "clientProvidedHostHeader": "s3.amazonaws.com"
    }
    },
    {
      "eventVersion": "1.03",
      "userIdentity": {
          "type": "IAMUser",
          "principalId": "111122223333",
          "arn": "arn:aws:iam::111122223333:user/myUserName",
          "accountId": "111122223333",
          "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
          "userName": "myUserName"
        },
      "eventTime": "2019-02-01T03:26:37Z",
      "eventSource": "s3.amazonaws.com",
      "eventName": "GetBucketVersioning",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "",
      "userAgent": "[]",
      "requestParameters": {
          "host": [
              "s3.us-west-2.amazonaws.com"
          ],
          "bucketName": "amzn-s3-demo-bucket1",
          "versioning": [
              ""
          ]
      },
      "responseElements": null,
      "additionalEventData": {
          "SignatureVersion": "SigV4",
```

```
            "CipherSuite": "ECDHE-RSA-AES128-SHA",
            "AuthenticationMethod": "AuthHeader"
        },
          "requestID": "07D681279BD94AED",
          "eventID": "f2b287f3-0df1-4961-a2f4-c4bdfed47657",
          "eventType": "AwsApiCall",
          "recipientAccountId": "111122223333",
          "tlsDetails": {
                "tlsVersion": "TLSv1.2",
                "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
                "clientProvidedHostHeader": "s3.amazonaws.com"
        }
        }
      ]
}
```


Aktivieren der CloudTrail Ereignisprotokollierung für S3-Buckets und -Objekte

Sie können CloudTrail Datenereignisse verwenden, um Informationen über Anfragen auf Bucket- und Objektebene in Amazon S3 abzurufen. Um CloudTrail Datenereignisse für alle Ihre Buckets oder für eine Liste bestimmter Buckets zu aktivieren, müssen Sie manuell einen Trail in erstellen. CloudTrail

> **Note**
>
> - Die Standardeinstellung für CloudTrail ist, dass nur Verwaltungsereignisse gesucht werden. Prüfen Sie, ob Datenereignisse für das Konto aktiviert wurden.
> - Ein S3-Bucket mit hoher Workload kann in kurzer Zeit Tausende Protokolle generieren. Denken Sie daran, wie lange Sie CloudTrail Datenereignisse für einen stark frequentierten Bereich aktivieren möchten.

CloudTrail speichert Amazon S3 S3-Datenereignisprotokolle in einem S3-Bucket Ihrer Wahl. Erwägen Sie die Verwendung eines Buckets in einem separaten Bucket AWS-Konto , um Ereignisse aus mehreren Buckets, die Sie möglicherweise besitzen, besser an einem zentralen Ort zu organisieren, um Abfragen und Analysen zu vereinfachen. AWS Organizations hilft Ihnen dabei AWS-Konto , einen zu erstellen, der mit dem Konto verknüpft ist, dem der Bucket gehört, den Sie überwachen. Weitere Informationen finden Sie unter Was ist AWS Organizations? im AWS Organizations Benutzerhandbuch.

Wenn Sie Datenereignisse für einen Trail in protokollieren, können Sie wählen CloudTrail, ob Sie erweiterte oder einfache Event-Selektoren verwenden möchten, um Datenereignisse für Objekte zu protokollieren, die in Allzweck-Buckets gespeichert sind. Um Datenereignisse für Objekte zu protokollieren, die in Verzeichnis-Buckets gespeichert sind, müssen Sie erweiterte Ereignisselektoren verwenden. Weitere Informationen finden Sie unter Logging with AWS CloudTrail für S3 Express One Zone.

Wenn Sie in der CloudTrail Konsole mithilfe erweiterter Event-Selektoren einen Trail erstellen, können Sie im Bereich Datenereignisse die Option Alle Ereignisse protokollieren für die Protokollauswahlvorlage auswählen, um alle Ereignisse auf Objektebene zu protokollieren. Wenn Sie in der CloudTrail Konsole mithilfe einfacher Ereignisauswahlen einen Trail erstellen, können Sie im Bereich Datenereignisse das Kontrollkästchen Alle S3-Buckets in Ihrem Konto auswählen aktivieren, um alle Ereignisse auf Objektebene zu protokollieren.

> **Note**
>
> - Eine bewährte Methode besteht darin, eine Lebenszykluskonfiguration für Ihren Datenereignis-Bucket in AWS CloudTrail zu erstellen. Konfigurieren Sie die Lebenszykluskonfiguration zum regelmäßigen Entfernen von Protokolldateien nach dem Zeitraum, der Ihres Erachtens für die Überprüfung erforderlich ist. Dadurch wird die Menge der Daten reduziert, die Athena in einer Abfrage analysiert. Weitere Informationen finden Sie unter Einstellung einer S3-Lifecycle-Konfiguration für einen Bucket.
> - Weitere Informationen zum Format der Protokolle finden Sie unter Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail.
> - Beispiele für das Abfragen von CloudTrail Protokollen finden Sie im AWS Big-Data-Blogbeitrag Analyze Security, Compliance, and Operational Activity Using AWS CloudTrail and Amazon Athena.

Aktivieren der Protokollierung für Objekte in einem Bucket mit der Konsole

Sie können die Amazon S3 S3-Konsole verwenden, um einen AWS CloudTrail Trail zum Protokollieren von Datenereignissen für Objekte in einem S3-Bucket zu konfigurieren. CloudTrail unterstützt die Protokollierung von Amazon S3 API S3-Vorgängen auf Objektebene wie `GetObject`, `DeleteObject`, und. `PutObject` Diese Ereignisse werden als Datenereignisse bezeichnet.

Standardmäßig protokollieren CloudTrail Trails keine Datenereignisse, aber Sie können Trails so konfigurieren, dass Datenereignisse für von Ihnen angegebene S3-Buckets oder Datenereignisse für alle Amazon S3 S3-Buckets in Ihrem protokolliert werden. AWS-Konto Weitere Informationen finden Sie unter Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail.

CloudTrail fügt keine Datenereignisse in den CloudTrail Ereignisverlauf ein. Darüber hinaus werden nicht alle Aktionen auf Bucket-Ebene in den CloudTrail Ereignisverlauf eingetragen. Weitere Informationen zu den API Aktionen auf Amazon S3 S3-Bucket-Ebene, die durch CloudTrail Protokollierung verfolgt werden, finden Sie unter. Amazon S3 S3-Aktionen auf Bucket-Ebene, die durch Protokollierung verfolgt werden CloudTrail Weitere Informationen zum Abfragen von CloudTrail Protokollen finden Sie im AWS Knowledge Center-Artikel über die Verwendung von Amazon CloudWatch Logs-Filtermustern und Amazon Athena zum Abfragen von CloudTrail Protokollen.

Um einen Trail zum Protokollieren von Datenereignissen für einen S3-Bucket zu konfigurieren, können Sie entweder die AWS CloudTrail -Konsole oder die Amazon-S3-Konsole verwenden. Wenn Sie einen Trail konfigurieren, um Datenereignisse für alle Amazon S3 S3-Buckets in Ihrem zu protokollieren AWS-Konto, ist es einfacher, die CloudTrail Konsole zu verwenden. Informationen zur Verwendung der CloudTrail Konsole zur Konfiguration eines Trails zur Protokollierung von S3-Datenereignissen finden Sie unter Datenereignisse im AWS CloudTrail Benutzerhandbuch.

> **Important**
>
> Für Datenereignisse werden zusätzliche Gebühren fällig. Weitere Informationen finden Sie unter AWS CloudTrail – Preise.

Das folgende Verfahren zeigt, wie Sie mit der Amazon S3 S3-Konsole einen CloudTrail Trail zum Protokollieren von Datenereignissen für einen S3-Bucket konfigurieren.

Um die Protokollierung von CloudTrail Datenereignissen für Objekte in einem S3-Allzweck-Bucket oder in einem S3-Verzeichnis-Bucket zu aktivieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus.
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie unter AWS CloudTrail Datenereignisse die Option Konfigurieren in CloudTrail.

Sie können einen neuen CloudTrail Trail erstellen oder einen vorhandenen Trail wiederverwenden und Amazon S3 S3-Datenereignisse so konfigurieren, dass sie in Ihrem Trail protokolliert werden. Informationen zum Erstellen von Trails in der CloudTrail Konsole finden Sie unter Erstellen und Aktualisieren eines Trails mit der Konsole im AWS CloudTrail Benutzerhandbuch. Informationen zur Konfiguration der Amazon S3 S3-Datenereignisprotokollierung in der CloudTrail Konsole finden Sie unter Protokollieren von Datenereignissen für Amazon S3 S3-Objekte im AWS CloudTrail Benutzerhandbuch.

> **Note**
>
> Wenn Sie die CloudTrail Konsole oder die Amazon S3 S3-Konsole verwenden, um einen Trail zum Protokollieren von Datenereignissen für einen S3-Bucket zu konfigurieren, zeigt die Amazon S3 S3-Konsole an, dass die Protokollierung auf Objektebene für den Bucket aktiviert ist.

Um die Protokollierung CloudTrail von Datenereignissen für Objekte in einem S3-Bucket zu deaktivieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die CloudTrail Konsole unter https://console.aws.amazon.com/cloudtrail/.
2. Wählen Sie im linken Navigationsbereich Trails aus.
3. Wählen Sie den Namen des Trails aus, den Sie erstellt haben, um Ereignisse für den Bucket zu protokollieren.
4. Wählen Sie oben rechts auf der Detailseite des Trails Protokollierung beenden aus.
5. Wählen Sie im anschließend angezeigten Dialogfeld Protokollierung beenden aus.

Weitere Informationen zum Aktivieren der Protokollierung auf Objektebene, wenn Sie einen S3-Bucket erstellen, finden Sie unter Erstellen eines Buckets.

Weitere Informationen zur CloudTrail Protokollierung mit S3-Buckets finden Sie in den folgenden Themen:

- Anzeigen der Eigenschaften eines S3-Buckets
- Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail
- Arbeiten mit CloudTrail Protokolldateien im AWS CloudTrail Benutzerhandbuch

Identifizieren von Amazon S3 S3-Anfragen mithilfe von CloudTrail

In Amazon S3 können Sie Anfragen anhand eines AWS CloudTrail Ereignisprotokolls identifizieren. AWS CloudTrail ist die bevorzugte Methode zur Identifizierung von Amazon S3 S3-Anfragen. Wenn Sie jedoch Amazon S3 S3-Serverzugriffsprotokolle verwenden, finden Sie weitere Informationen unterthe section called “Identifizieren von S3-Anfragen”.

Themen

- Identifizieren von Anfragen an Amazon S3 in einem CloudTrail Protokoll
- Identifizieren von Amazon S3 Signature Version 2-Anfragen mithilfe von CloudTrail
- Identifizieren des Zugriffs auf S3-Objekte mithilfe von CloudTrail

Identifizieren von Anfragen an Amazon S3 in einem CloudTrail Protokoll

Nachdem Sie die Bereitstellung von Ereignissen CloudTrail an einen Bucket eingerichtet haben, sollten Sie auf der Amazon S3 S3-Konsole sehen, dass Objekte an Ihren Ziel-Bucket gesendet werden. Diese sind folgendermaßen formatiert:

```
s3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/Region/yyyy/mm/dd
```

Von protokollierte Ereignisse CloudTrail werden komprimiert gespeichert, gzipped JSONObjekte in Ihrem S3-Bucket. Um Anfragen effizient zu finden, sollten Sie einen Dienst wie Amazon Athena verwenden, um die CloudTrail Protokolle zu indizieren und abzufragen.

Weitere Informationen zu CloudTrail und Athena finden Sie unter Erstellen der Tabelle für AWS CloudTrail Protokolle in Athena mithilfe der Partitionsprojektion im Amazon Athena Athena-Benutzerhandbuch.

Identifizieren von Amazon S3 Signature Version 2-Anfragen mithilfe von CloudTrail

Sie können ein CloudTrail Ereignisprotokoll verwenden, um zu ermitteln, welche API Signaturversion zum Signieren einer Anfrage in Amazon S3 verwendet wurde. Diese Funktion ist wichtig, weil die Unterstützung für Signature Version 2 deaktiviert wird (veraltet). Danach akzeptiert Amazon S3 keine Anforderungen mit Signature Version 2 mehr, alle Anforderungen müssen also mit Signature Version 4 signiert werden.

Wir empfehlen dringend, dass Sie es verwenden CloudTrail , um festzustellen, ob in einem Ihrer Workflows Signature Version 2-Signaturen verwendet werden. Korrigieren Sie diese Workflows,

indem Sie die betreffenden Bibliotheken und den Code so aktualisieren, dass mittels Signature Version 4 signiert wird. Dadurch können Beeinträchtigungen der geschäftlichen Prozesse werden.

Weitere Informationen finden Sie unter Ankündigung: AWS CloudTrail Für Amazon S3 werden neue Felder für erweiterte Sicherheitsüberprüfungen hinzugefügt AWS re:Post.

> **Note**
>
> CloudTrail Ereignisse für Amazon S3 enthalten die Signaturversion in den Anforderungsdetails unter dem Schlüsselnamen '`additionalEventData`. Um die Signaturversion von Anfragen für Objekte in Amazon S3 wie`GET`, `PUT` und `DELETE` Anfragen zu finden, müssen Sie CloudTrail Datenereignisse aktivieren. (Diese Option ist standardmäßig deaktiviert.)

AWS CloudTrail ist die bevorzugte Methode zur Identifizierung von Signature Version 2-Anfragen. Wenn Sie Zugriffsprotokolle von Amazon S3 Server verwenden, lesen Sie Identifizieren von Signature-Version-2-Anforderungen mittels Amazon-S3-Zugriffsprotokollen.

Themen

- Athena-Abfragebeispiele zum Identifizieren von Anforderungen zur Amazon S3 Signature Version 2
- Partitionieren von Signature Version 2-Daten

Athena-Abfragebeispiele zum Identifizieren von Anforderungen zur Amazon S3 Signature Version 2

Example – Wählen Sie alle Signature-Version-2-Ereignisse aus und drucken Sie nur **`EventTime`**, **`S3_Action`**, **`Request_Parameters`**, **`Region`**, **`SourceIP`** und **`UserAgent`**

Ersetzen Sie in der folgenden Athena-Abfrage *`s3_cloudtrail_events_db.cloudtrail_table`* mit Ihren Athena-Daten und erhöhen oder entfernen Sie das Limit nach Bedarf.

```
SELECT EventTime, EventName as S3_Action, requestParameters as Request_Parameters,
 awsregion as AWS_Region, sourceipaddress as Source_IP, useragent as User_Agent
FROM s3_cloudtrail_events_db.cloudtrail_table
WHERE eventsource='s3.amazonaws.com'
AND json_extract_scalar(additionalEventData, '$.SignatureVersion')='SigV2'
LIMIT 10;
```

Example – Alle Anforderer auswählen, die Signature Version 2-Datenverkehr senden

```
SELECT useridentity.arn, Count(requestid) as RequestCount
FROM s3_cloudtrail_events_db.cloudtrail_table
WHERE eventsource='s3.amazonaws.com'
    and json_extract_scalar(additionalEventData, '$.SignatureVersion')='SigV2'
Group by useridentity.arn
```

Partitionieren von Signature Version 2-Daten

Wenn Sie über eine große Datenmenge verfügen, die abgefragt werden muss, können Sie Kosten und Laufzeit für Athena reduzieren, indem Sie eine partitionierte Tabelle erstellen.

Erstellen Sie zu diesem Zweck folgendermaßen eine neue Tabelle mit Partitionen.

```
  CREATE EXTERNAL TABLE s3_cloudtrail_events_db.cloudtrail_table_partitioned(
       eventversion STRING,
       userIdentity STRUCT<
           type:STRING,
           principalid:STRING,
           arn:STRING,
           accountid:STRING,
           invokedby:STRING,
           accesskeyid:STRING,
           userName:STRING,
        sessioncontext:STRUCT<
                   attributes:STRUCT<
                   mfaauthenticated:STRING,
                   creationdate:STRING>,
                   sessionIssuer:STRUCT<
                   type:STRING,
                   principalId:STRING,
                   arn:STRING,
                   accountId:STRING,
                   userName:STRING>
               >
            >,
       eventTime STRING,
       eventSource STRING,
       eventName STRING,
```

```
        awsRegion STRING,
        sourceIpAddress STRING,
        userAgent STRING,
        errorCode STRING,
        errorMessage STRING,
        requestParameters STRING,
        responseElements STRING,
        additionalEventData STRING,
        requestId STRING,
        eventId STRING,
        resources ARRAY<STRUCT<ARN:STRING,accountId: STRING,type:STRING>>,
        eventType STRING,
        apiVersion STRING,
        readOnly STRING,
        recipientAccountId STRING,
        serviceEventDetails STRING,
        sharedEventID STRING,
        vpcEndpointId STRING
    )
    PARTITIONED BY (region string, year string, month string, day string)
    ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
    STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
    OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
    LOCATION 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/';
```

Erstellen Sie dann die einzelnen Partitionen. Sie können keine Resultate aus noch nicht erstellten Daten ermitteln.

```
ALTER TABLE s3_cloudtrail_events_db.cloudtrail_table_partitioned ADD
    PARTITION (region= 'us-east-1', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/us-east-1/2019/02/19/'
    PARTITION (region= 'us-west-1', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/us-west-1/2019/02/19/'
    PARTITION (region= 'us-west-2', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/us-west-2/2019/02/19/'
    PARTITION (region= 'ap-southeast-1', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/ap-southeast-1/2019/02/19/'
    PARTITION (region= 'ap-southeast-2', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/ap-southeast-2/2019/02/19/'
    PARTITION (region= 'ap-northeast-1', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/ap-northeast-1/2019/02/19/'
```

```
    PARTITION (region= 'eu-west-1', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/eu-west-1/2019/02/19/'
    PARTITION (region= 'sa-east-1', year= '2019', month= '02', day= '19') LOCATION
 's3://amzn-s3-demo-bucket1/AWSLogs/111122223333/CloudTrail/sa-east-1/2019/02/19/';
```

Anschließend können Sie die Anforderung basierend auf diesen Partitionen erstellen und müssen nicht mehr den gesamten Bucket laden.

```
SELECT useridentity.arn,
Count(requestid) AS RequestCount
FROM s3_cloudtrail_events_db.cloudtrail_table_partitioned
WHERE eventsource='s3.amazonaws.com'
AND json_extract_scalar(additionalEventData, '$.SignatureVersion')='SigV2'
AND region='us-east-1'
AND year='2019'
AND month='02'
AND day='19'
Group by useridentity.arn
```

Identifizieren des Zugriffs auf S3-Objekte mithilfe von CloudTrail

Sie können Ihre AWS CloudTrail Ereignisprotokolle verwenden, um Amazon S3 S3-Objektzugriffsanforderungen für Datenereignisse wie `GetObject` `DeleteObject``PutObject`, und zu identifizieren und zusätzliche Informationen zu diesen Anfragen zu erhalten.

Das folgende Beispiel zeigt, wie alle PUT Objektanforderungen für Amazon S3 aus einem AWS CloudTrail Ereignisprotokoll abgerufen werden.

Themen

- Athena-Abfragebeispiele zur Identifizierung von Amazon-S3-Objektzugriffsanfragen

Athena-Abfragebeispiele zur Identifizierung von Amazon-S3-Objektzugriffsanfragen

Ersetzen Sie in den folgenden Athena-Abfragebeispielen *s3_cloudtrail_events_db.cloudtrail_table* mit Ihren Athena-Daten und ändern Sie den Datumsbereich nach Bedarf.

Example – Wählen Sie alle Ereignisse aus, für die **PUT**-Objektzugriffsanforderungen vorliegen, und drucken Sie nur **EventTime**, **EventSource**, **SourceIP**, **UserAgent**, **BucketName**, **object** und **UserARN**

```
SELECT
  eventTime,
  eventName,
  eventSource,
  sourceIpAddress,
  userAgent,
  json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
  json_extract_scalar(requestParameters, '$.key') as object,
  userIdentity.arn as userArn
FROM
  s3_cloudtrail_events_db.cloudtrail_table
WHERE
  eventName = 'PutObject'
  AND eventTime BETWEEN '2019-07-05T00:00:00Z' and '2019-07-06T00:00:00Z'
```

Example – Wählen Sie alle Ereignisse aus, für die **GET**-Objektzugriffsanforderungen vorliegen, und drucken Sie nur **EventTime**, **EventSource**, **SourceIP**, **UserAgent**, **BucketName**, **object** und **UserARN**

```
SELECT
  eventTime,
  eventName,
  eventSource,
  sourceIpAddress,
  userAgent,
  json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
  json_extract_scalar(requestParameters, '$.key') as object,
  userIdentity.arn as userArn
FROM
  s3_cloudtrail_events_db.cloudtrail_table
WHERE
  eventName = 'GetObject'
  AND eventTime BETWEEN '2019-07-05T00:00:00Z' and '2019-07-06T00:00:00Z'
```

Example – Wählen Sie alle anonymen Anforderungsereignisse für einen Bucket in einem bestimmten Zeitraum aus und drucken Sie nur **`EventTime`**, **`EventName`**, **`EventSource`**, **`SourceIP`**, **`UserAgent`**, **`BucketName`**, **`UserARN`** und **`AccountID`**

```
SELECT
  eventTime,
  eventName,
  eventSource,
  sourceIpAddress,
  userAgent,
  json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
  userIdentity.arn as userArn,
  userIdentity.accountId
FROM
  s3_cloudtrail_events_db.cloudtrail_table
WHERE
  userIdentity.accountId = 'anonymous'
  AND eventTime BETWEEN '2019-07-05T00:00:00Z' and '2019-07-06T00:00:00Z'
```

Example — Identifizieren Sie alle Anfragen, ACL für die eine Autorisierung erforderlich war

Das folgende Amazon Athena Athena-Abfragebeispiel zeigt, wie Sie alle Anfragen an Ihre S3-Buckets identifizieren können, für die eine Zugriffskontrollliste (ACL) für die Autorisierung erforderlich war. Wenn ACL für die Anfrage eine Autorisierung erforderlich war, ist der `aclRequired` Wert in`additionalEventData`. `Yes` Wenn keine erforderlich ACLs waren, `aclRequired` ist er nicht vorhanden. Sie können diese Informationen verwenden, um diese ACL Berechtigungen auf die entsprechenden Bucket-Richtlinien zu migrieren. Nachdem Sie diese Bucket-Richtlinien erstellt haben, können Sie sie ACLs für diese Buckets deaktivieren. Weitere Informationen zur Deaktivierung ACLs finden Sie unter. Voraussetzungen für die Deaktivierung ACLs

```
SELECT
  eventTime,
  eventName,
  eventSource,
  sourceIpAddress,
  userAgent,
  userIdentity.arn as userArn,
  json_extract_scalar(requestParameters, '$.bucketName') as bucketName,
  json_extract_scalar(requestParameters, '$.key') as object,
  json_extract_scalar(additionalEventData, '$.aclRequired') as aclRequired
FROM
```

```
  s3_cloudtrail_events_db.cloudtrail_table
WHERE
  json_extract_scalar(additionalEventData, '$.aclRequired') = 'Yes'
  AND eventTime BETWEEN '2022-05-10T00:00:00Z' and '2022-08-10T00:00:00Z'
```

> **Note**
>
> - Diese Abfragebeispiele können auch für die Sicherheitsüberwachung nützlich sein. Sie können die Ergebnisse auf `PutObject`- oder `GetObject`-Aufrufe von unerwarteten oder nicht autorisierten IP-Adressen oder Anforderern und zum Identifizieren anonymer Anforderungen an Ihre Buckets prüfen.
> - Diese Abfrage ruft nur Informationen von der Zeit ab, zu der die Protokollierung aktiviert wurde.

Wenn Sie Amazon-S3-Server-Zugriffsprotokolle verwenden, lesen Sie Identifizieren von Objektzugriffsanforderungen mittels Amazon-S3-Zugriffsprotokollen.

Protokollieren von Anfragen mit Server-Zugriffsprotokollierung

Die Server-Zugriffsprotokollierung bietet detaillierte Aufzeichnungen über die Anforderungen, die an einen Bucket gestellt wurden. Server-Zugriffsprotokolle sind für viele Anwendungen nützlich. Beispielsweise können Zugriffsprotokoll-Informationen bei Sicherheits- und Zugriffsprüfungen nützlich sein. Diese Informationen können Ihnen auch helfen, mehr über Ihre Kundenbasis zu erfahren und Ihre Amazon-S3-Rechnung zu verstehen.

> **Note**
>
> Serverzugriffsprotokolle protokollieren keine Informationen zu Fehlern wegen Umleitungen in falsche Regionen, die nach dem 20. März 2019 gestartet wurden. Fehler wegen Umleitungen in falsche Regionen treten auf, wenn eine Anforderung für ein Objekt oder einen Bucket außerhalb der Region gesendet wird, in der sich der Bucket befindet.

Wie aktiviere ich die Protokollzustellung?

Um die Protokollbereitstellung zu aktivieren, führen Sie die folgenden grundlegenden Schritte aus. Details hierzu finden Sie unter Aktivieren Sie die Amazon-S3-Server-Zugriffsprotokollierung.

1. Geben Sie den Namen des Ziel-Buckets (auch Ziel-Bucket genannt) an. In diesem Bucket soll Amazon S3 die Zugriffsprotokolle als Objekte speichern. Quell- und Ziel-Bucket müssen sich in derselben AWS-Region befinden und demselben Konto gehören. Der Ziel-Bucket darf keine S3-Object-Lock-Standardkonfiguration für die Aufbewahrungsdauer besitzen. Für den Ziel-Bucket darf außerdem die Option „Zahlung durch den Anforderer“ nicht aktiviert sein.

   Sie können Protokolle in jeden Bucket speichern lassen, der sich in der gleichen Region wie der Quell-Bucket befindet, einschließlich des Quell-Buckets selbst. Zur einfacheren Protokollverwaltung empfehlen wir jedoch, Zugriffsprotokolle in einem anderen Bucket zu speichern.

   Wenn Quell- und Ziel-Bucket identisch sind, werden zusätzliche Protokolle für die Protokolle erstellt, die zum Bucket geschrieben werden. Hierdurch entsteht eine Endlosprotokollschleife. Diese Vorgehensweise wird nicht empfohlen, da sie zu einer geringfügigen Erhöhung der Speicherkosten führen könnte. Weiterhin könnten die zusätzlichen Protokolle über Protokolle das Auffinden des gesuchten Protokolls erschweren.

   Wenn Sie Zugriffsprotokolle im Quell-Bucket speichern möchten, sollten Sie ein Zielpräfix (Zielpräfix) für alle Protokollobjektschlüssel angeben. Wenn Sie ein Präfix angeben, beginnen alle Protokollobjektnamen mit einer gemeinsamen Zeichenfolge. So können Sie die Protokollobjekte leichter identifizieren.

2. (Optional) Weisen Sie allen Amazon-S3-Protokollobjektschlüsseln ein Zielpräfix zu. Das Zielpräfix (Zielpräfix) macht es einfacher für Sie, die Protokollobjekte zu finden. Wenn Sie beispielsweise den Präfixwert `logs/` angeben, beginnt jedes von Amazon S3 erstellte Protokollobjekt mit dem Präfix `logs/` im Schlüssel, z. B.:

   ```
   logs/2013-11-01-21-32-16-E568B2907131C0C0
   ```

   Wenn Sie den Präfixwert `logs` angeben, sieht das Protokollobjekt wie folgt aus:

   ```
   logs2013-11-01-21-32-16-E568B2907131C0C0
   ```

   Präfixe sind auch nützlich, um zwischen Quell-Buckets zu unterscheiden, wenn mehrere Buckets zum selben Ziel-Bucket protokolliert werden.

   Dieses Präfix ist auch beim Löschen der Protokolle nützlich. Beispielsweise können Sie eine Lebenszyklus-Konfigurationsregel für Amazon S3 festlegen, um Objekte mit einem

bestimmten Präfix zu löschen. Weitere Informationen finden Sie unter Löschen von Amazon-S3-Protokolldateien.

3. (Optional) Legen Sie Berechtigungen fest, um anderen Benutzern Zugriff auf die generierten Protokolle zu gewähren. Standardmäßig hat nur der Bucket-Eigentümer stets vollen Zugriff auf die Protokollobjekte. Wenn Ihr Ziel-Bucket die vom Bucket-Besitzer erzwungene Einstellung für S3-Objektbesitz verwendet, um Zugriffskontrolllisten (ACLs) zu deaktivieren, können Sie keine Berechtigungen für Zielzuschüsse (auch als Zielzuschüsse bezeichnet) gewähren, die verwendenACLs. Sie können jedoch Ihre Bucket-Richtlinie für den Ziel-Bucket aktualisieren, um anderen Benutzern Zugriff zu gewähren. Weitere Informationen erhalten Sie unter Identity and Access Management für Amazon S3 und Berechtigungen für Protokollbereitstellung.
4. (Optional) Legen Sie ein Protokollobjekt-Schlüsselformat für die Protokolldateien fest. Es gibt zwei Optionen für das Protokollobjekt-Schlüsselformat (auch als Zielobjekt-Schlüsselformat bezeichnet):
   - Non-date-based Partitionierung — Dies ist das ursprüngliche Schlüsselformat für Log-Objekte. Wenn Sie dieses Format wählen, sieht das Protokolldatei-Schlüsselformat wie folgt aus:

     ```
     [DestinationPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
     ```

     Wenn Sie beispielsweise `logs/` als Präfix angeben, werden Ihre Protokollobjekte wie folgt benannt:

     ```
     logs/2013-11-01-21-32-16-E568B2907131C0C0
     ```

   - Partitionierung mit Datum – Wenn Sie die Partitionierung mit Datum wählen, können Sie die Ereignis- oder Bereitstellungszeit für die Protokolldatei als die Datumsquelle wählen, die im Protokollformat verwendet wird. Dieses Format vereinfacht das Abfragen der Protokolle.

     Wenn Sie die Partitionierung mit Datum wählen, sieht das Protokolldatei-Schlüsselformat wie folgt aus:

     ```
     [DestinationPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/
     [DD]/[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
     ```

     Wenn Sie beispielsweise `logs/` als Zielpräfix angeben, werden Ihre Protokollobjekte wie folgt benannt:

     ```
     logs/123456789012/us-west-2/amzn-s3-demo-source-
     bucket/2023/03/01/2023-03-01-21-32-16-E568B2907131C0C0
     ```

Bei der Bereitstellung nach Bereitstellungszeit entspricht die Zeit in den Namen der Protokolldateien der Bereitstellungszeit für die Protokolldateien.

Bei der Bereitstellung nach Ereigniszeit entsprechen die Angaben für Jahr, Monat und Tag dem Tag, an dem das Ereignis eingetreten ist. Stunde, Minuten und Sekunden sind im Schlüssel auf `00` festgelegt. Die in diesen Protokolldateien bereitgestellten Protokolle beziehen sich nur auf einen bestimmten Tag.

Wenn Sie Ihre Protokolle über AWS Command Line Interface (AWS CLI) oder Amazon S3 konfigurieren AWS SDKs, verwenden Sie RESTAPI, `TargetObjectKeyFormat` um das Schlüsselformat für das Protokollobjekt anzugeben. Um die non-date-based Partitionierung anzugeben, verwenden Sie`SimplePrefix`. Um eine Partitionierung mit Datum anzugeben, verwenden Sie `PartitionedPrefix`. Bei Verwendung von `PartitionedPrefix` verwenden Sie `PartitionDateSource`, um `EventTime` oder `DeliveryTime` anzugeben.

Für `SimplePrefix` sieht das Protokolldatei-Schlüsselformat wie folgt aus:

```
[TargetPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Für `PartitionedPrefix` mit Ereignis- oder Bereitstellungszeit sieht das Protokolldatei-Schlüsselformat wie folgt aus:

```
[TargetPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/
[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Protokollobjekt-Schlüsselformat

Amazon S3 verwendet die folgenden Objektschlüsselformate für die Protokollobjekte, die in den Ziel-Bucket hochgeladen werden:

- Non-date-based Partitionierung — Dies ist das ursprüngliche Schlüsselformat für Protokollobjekte. Wenn Sie dieses Format wählen, sieht das Protokolldatei-Schlüsselformat wie folgt aus:

```
[DestinationPrefix][YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

- Partitionierung mit Datum – Wenn Sie die Partitionierung mit Datum wählen, können Sie die Ereignis- oder Bereitstellungszeit für die Protokolldatei als die Datumsquelle wählen, die im Protokollformat verwendet wird. Dieses Format vereinfacht das Abfragen der Protokolle.

  Wenn Sie die Partitionierung mit Datum wählen, sieht das Protokolldatei-Schlüsselformat wie folgt aus:

```
[DestinationPrefix][SourceAccountId]/[SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/
[YYYY]-[MM]-[DD]-[hh]-[mm]-[ss]-[UniqueString]
```

Im Protokollobjektschlüssel sind `YYYY`, `MM`, `DD`, `hh`, `mm` und `ss` jeweils die Stellen für Jahr, Monat, Tag, Stunde, Minute und Sekunden. Diese Daten und Uhrzeiten sind in koordinierter Weltzeit (UTC) angegeben.

Eine Protokolldatei, die zu einem bestimmten Zeitpunkt bereitgestellt wurde, kann Datensätze enthalten, die an einem beliebigen Zeitpunkt davor geschrieben wurden. Es lässt sich nicht feststellen, ob alle Protokoll-Datensätze für ein bestimmtes Zeitintervall bereitgestellt wurden oder nicht.

Die `UniqueString`-Komponente des Schlüssels verhindert, dass Dateien überschrieben werden. Sie hat keine Bedeutung und wird normalerweise von Protokollverarbeitungssoftware ignoriert.

Wie werden Protokolle ausgeliefert?

Amazon S3 sammelt regelmäßig Zugriffsprotokoll-Datensätze, konsolidiert die Datensätze in Protokolldateien und lädt anschließend die Protokolldateien als Protokollobjekte zu Ihrem Ziel-Bucket hoch. Wenn Sie die Protokollierung für mehrere Quell-Buckets mit demselben Ziel-Bucket aktivieren, besitzt der Ziel-Bucket Zugriffsprotokolle für alle diese Quell-Buckets. Jedes Protokollobjekt gibt jedoch Zugriffsprotokoll-Datensätze für einen bestimmten Quell-Bucket aus.

Amazon S3 verwendet ein spezielles Protokollbereitstellungskonto zum Schreiben der Server-Zugriffsprotokolle. Für diese Protokolle gelten die normalen Zugriffskontrollbeschränkungen. Wir empfehlen Ihnen, die Bucket-Richtlinie für den Ziel-Bucket zu aktualisieren, um Zugriff auf den Protokollservice-Prinzipal (`logging.s3.amazonaws.com`) zur Zugriffsprotokollbereitstellung zu gewähren. Sie können der S3-Protokollzustellungsgruppe auch über Ihre Bucket-Zugriffskontrollliste (ACL) Zugriff auf die Übermittlung von Zugriffsprotokollen gewähren. Es wird jedoch nicht empfohlen, mithilfe Ihres Buckets ACL Zugriff auf die S3-Protokollbereitstellungsgruppe zu gewähren.

Wenn Sie die Serverzugriffsprotokollierung über die Ziel-Bucket-Richtlinie aktivieren und Zugriff für die Zugriffsprotokollbereitstellung gewähren, müssen Sie die Richtlinie aktualisieren, um `s3:PutObject`-Zugriff für den Protokollierungsservice-Prinzipal zuzulassen. Wenn Sie die Serverzugriffsprotokollierung über die Amazon-S3-Konsole aktivieren, aktualisiert die Konsole die Ziel-Bucket-Richtlinie automatisch, um dem Protokollierungsservice-Prinzipal diese Berechtigungen zu gewähren. Weitere Informationen zum Erteilen von Berechtigungen für die Zustellung von Serverzugriffsprotokollen finden Sie unter Berechtigungen für Protokollbereitstellung.

> **Note**
>
> S3 unterstützt nicht die Übermittlung von CloudTrail Protokollen oder Serverzugriffsprotokollen an den Anforderer oder den Bucket-Besitzer für VPC Endpunktanforderungen, wenn die VPC Endpunktrichtlinie sie ablehnt, oder für Anfragen, die fehlschlagen, bevor die VPC Richtlinie ausgewertet wurde.

Vom Bucket-Eigentümer erzwungene Einstellung für S3 Object Ownership

Wenn der Ziel-Bucket die erzwungene Einstellung des Bucket-Besitzers für Objekteigentum verwendet, ACLs sind sie deaktiviert und wirken sich nicht mehr auf die Berechtigungen aus. Sie müssen die Bucket-Richtlinie aktualisieren, damit der Ziel-Bucket Zugriff auf den Protokollierungsservice-Prinzipal gewähren kann. Informationen zu Object Ownership finden Sie unter Gewähren von Zugriff auf die S3-Protokollbereitstellungsgruppe für die Protokollierung des Serverzugriffs.

Best-Effort-Serverprotokollbereitstellung

Serverzugriffsprotokoll-Datensätze werden auf Best-Effort-Grundlage bereitgestellt. Die meisten Anforderungen nach einem Bucket, der für die Protokollierung richtig konfiguriert ist, führen zu einem ausgelieferten Protokollsatz. Die meisten Protokollsätze werden innerhalb weniger Stunden nach der Aufnahme geliefert, können aber häufiger geliefert werden.

Die Vollständigkeit und Aktualität der Serverprotokollierung wird nicht garantiert. Der Protokolldatensatz für eine bestimmte Anforderung wird möglicherweise viel später bereitgestellt, als die Anforderung tatsächlich verarbeitet wurde; es kann auch sein, dass er gar nicht bereitgestellt wird. Es ist möglich, dass Sie sogar ein Duplikat eines Protokolldatensatzes sehen. Der Zweck der Serverprotokolle besteht darin, Ihnen einen Überblick über die Art des Datenverkehrs zu und von Ihrem Bucket zu vermitteln. Auch wenn Protokolldatensätze selten verloren gehen oder dupliziert

werden, sollten Sie wissen, dass die Serverprotokollierung nicht der vollständigen Erfassung aller Anforderungen dient.

Aufgrund der Best-Effort-Grundlage der Serverprotokollierung enthalten Ihre Nutzungsberichte möglicherweise eine oder mehrere Zugriffsanforderungen, die nicht in einem bereitgestellten Serverprotokoll angezeigt werden. Sie finden diese Nutzungsberichte unter Kosten- und Nutzungsberichte in der AWS Billing and Cost Management -Konsole.

Statusänderungen in der Bucket-Protokollierung werden mit der Zeit wirksam

Änderungen am Protokollierungsstatus eines Buckets benötigen einige Zeit, bis sie sich auf die Bereitstellung von Protokolldateien auswirken. Wenn Sie die Protokollierung für einen Bucket aktivieren, werden in der folgenden Stunde möglicherweise einige Anforderungen protokolliert und andere nicht. Angenommen, Sie ändern den Ziel-Bucket für die Protokollierung von Bucket A in Bucket B. In der nächsten Stunde werden möglicherweise einige Protokolle in Bucket A bereitgestellt, während andere Protokolle im neuen Ziel-Bucket B bereitgestellt werden. In jedem Fall werden die neuen Einstellungen letztendlich ohne weiteres Eingreifen Ihrerseits wirksam.

Weitere Informationen zu Protokollierungs- und Protokolldateien finden Sie in den folgenden Abschnitten:

Themen

- Aktivieren Sie die Amazon-S3-Server-Zugriffsprotokollierung
- Amazon-S3-Server-Zugriffsprotokollformat
- Löschen von Amazon-S3-Protokolldateien
- Verwenden von Amazon-S3-Serverzugriffsprotokollen zur Identifizierung von Anforderungen
- Behebung von Fehlern bei der Server-Zugriffsprotokollierung

Aktivieren Sie die Amazon-S3-Server-Zugriffsprotokollierung

Die Server-Zugriffsprotokollierung bietet detaillierte Aufzeichnungen über die Anfragen, die an einen Amazon-S3-Bucket gestellt wurden. Server-Zugriffsprotokolle sind für viele Anwendungen nützlich. Beispielsweise können Zugriffsprotokoll-Informationen bei Sicherheits- und Zugriffsprüfungen nützlich sein. Diese Informationen können Ihnen auch helfen, mehr über Ihre Kundenbasis zu erfahren und Ihre Amazon-S3-Rechnung zu verstehen.

Standardmäßig erfasst Amazon S3 keine Server-Zugriffsprotokolle. Wenn Sie die Protokollierung aktivieren, stellt Amazon S3 Zugriffsprotokolle für einen Quell-Bucket in einem von Ihnen ausgewählten Ziel-Bucket (Ziel-Bucket) bereit. Der Ziel-Bucket muss sich im selben AWS-Region und AWS-Konto wie der Quell-Bucket befinden.

Ein Zugriffsprotokollsatz enthält Details über die Anforderungen, die an einen Bucket gestellt werden. Dabei kann es sich um den Anforderungstyp, die in der Anfrage angegebenen Ressourcen sowie Uhrzeit und Datum der Anfrageverarbeitung handeln. Weitere Informationen zu den Grundlagen der Protokollierung finden Sie unter Protokollieren von Anfragen mit Server-Zugriffsprotokollierung.

Important

- Für die Aktivierung der Server-Zugriffsprotokollierung auf einem Amazon-S3-Bucket fallen keine zusätzlichen Kosten an. Für die Speicherung der Protokolldateien, die das System an Sie überträgt, fallen allerdings die normalen Gebühren an. (Sie können die Protokolldateien jederzeit löschen.) Wir berechnen keine Datenübertragungskosten für die Bereitstellung von Protokolldateien. Für den Zugriff auf die Protokolldateien fallen jedoch die regulären Kosten für Datenübertragungen an.
- Für Ihren Ziel-Bucket sollte keine Serverzugriffsprotokollierung aktiviert sein. Sie können Protokolle in jeden Bucket speichern lassen, der sich in der gleichen Region wie der Quell-Bucket befindet, einschließlich des Quell-Buckets selbst. Die Zustellung von Protokollen an den Quell-Bucket führt jedoch zu einer unendlichen Schleife von Protokollen und wird nicht empfohlen. Zur einfacheren Protokollverwaltung empfehlen wir, Zugriffsprotokolle in einem anderen Bucket zu speichern. Weitere Informationen finden Sie unter Wie aktiviere ich die Protokollzustellung?
- S3-Buckets, für die S3 Object Lock aktiviert ist, können nicht als Ziel-Buckets für Serverzugriffsprotokolle verwendet werden. Ihr Ziel-Bucket darf nicht für eine Standardaufbewahrungsdauer konfiguriert sein.
- Für den Ziel-Bucket darf die Option „Zahlung durch den Anforderer“ nicht aktiviert sein.
- Sie können die Standard-Bucket-Verschlüsselung für den Ziel-Bucket nur verwenden, wenn Sie serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verwenden, die den 256-Bit-Advanced Encryption Standard (-256) verwenden. AES Die serverseitige Standardverschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln (SSE-) wird nicht unterstützt. KMS

Sie können die Serverzugriffsprotokollierung aktivieren oder deaktivieren, indem Sie die Amazon S3-Konsole, Amazon S3API, die AWS Command Line Interface (AWS CLI) oder verwenden AWS SDKs.

Berechtigungen für Protokollbereitstellung

Amazon S3 verwendet ein spezielles Protokollbereitstellungskonto zum Schreiben der Server-Zugriffsprotokolle. Für diese Protokolle gelten die normalen Zugriffskontrollbeschränkungen. Zur Bereitstellung von Zugriffsprotokollen müssen Sie dem Protokollierungsservice-Prinzipal (`logging.s3.amazonaws.com`) Zugriff auf den Ziel-Bucket gewähren.

Um Amazon S3 Berechtigungen für die Protokollzustellung zu gewähren, können Sie entweder eine Bucket-Richtlinie oder Bucket-Zugriffskontrolllisten (ACLs) verwenden, abhängig von den Einstellungen für den Besitz von S3-Objekten in Ihrem Ziel-Bucket. Wir empfehlen jedoch, stattdessen eine Bucket-Richtlinie zu verwendenACLs.

Vom Bucket-Eigentümer erzwungene Einstellung für S3 Object Ownership

Wenn der Ziel-Bucket die erzwungene Einstellung für Object Ownership verwendet, ACLs sind sie deaktiviert und wirken sich nicht mehr auf die Berechtigungen aus. In diesem Fall müssen Sie die Bucket-Richtlinie für den Ziel-Bucket aktualisieren, um Zugriff auf den Protokollierungsservice-Prinzipal zu gewähren. Sie können Ihren Bucket nicht aktualisierenACL, um Zugriff auf die S3-Protokollbereitstellungsgruppe zu gewähren. Sie können auch keine Zielzuschüsse (auch Zielzuschüsse genannt) in Ihre aufnehmen PutBucketLogging-Konfiguration.

Informationen zur Migration eines vorhandenen Buckets ACLs für die Zugriffs-Log-Zustellung zu einer Bucket-Richtlinie finden Sie unterGewähren von Zugriff auf die S3-Protokollbereitstellungsgruppe für die Protokollierung des Serverzugriffs. Informationen zu Object Ownership finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket. Wenn Sie neue Buckets erstellen, ACLs sind diese standardmäßig deaktiviert.

Gewähren von Zugriff über eine Bucket-Richtlinie

Um über die Bucket-Richtlinie Zugriff auf den Ziel-Bucket zu gewähren, aktualisieren Sie die Bucket-Richtlinie so, dass sie die `s3:PutObject`-Berechtigung für den Protokollierungsservice-Prinzipal zulässt. Wenn Sie die Serverzugriffsprotokollierung über die Amazon-S3-Konsole aktivieren, aktualisiert die Konsole die Bucket-Richtlinie für den Ziel-Bucket automatisch, um dem Protokollierungsservice-Prinzipal diese Berechtigung zu gewähren. Wenn Sie die Serverzugriffsprotokollierung programmgesteuert aktivieren, müssen Sie die Bucket-Richtlinie für den Ziel-Bucket manuell aktualisieren, um Zugriff auf den Protokollierungsservice-Prinzipal zu gewähren.

Ein Beispiel für eine Bucket-Richtlinie, die Zugriff auf den Protokollierungsservice-Prinzipal gewährt, finden Sie unter the section called “Erteilen von Berechtigungen für den Prinzipal des Protokollierungsservices mithilfe einer Bucket-Richtlinie”.

Zugriff mithilfe eines Buckets gewähren ACLs

Sie können auch Bucket verwenden, ACLs um Zugriff auf die Zugriffs-Log-Zustellung zu gewähren. Sie fügen dem Bucket einen Grant-Eintrag hinzuACL, der der S3-Protokollbereitstellungsgruppe `READ_ACP` Berechtigungen gewährt`WRITE`. Es wird jedoch nicht empfohlen, mithilfe eines Buckets ACLs Zugriff auf die S3-Protokollbereitstellungsgruppe zu gewähren. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket. Hinweise zur Migration eines vorhandenen Buckets ACLs für die Zugriffs-Log-Zustellung zu einer Bucket-Richtlinie finden Sie unterGewähren von Zugriff auf die S3-Protokollbereitstellungsgruppe für die Protokollierung des Serverzugriffs. Ein BeispielACL, das Zugriff auf den Logging-Dienstprinzipal gewährt, finden Sie unterthe section called “Gewähren Sie der Protokollbereitstellungsgruppe mithilfe eines Buckets Berechtigungen ACL”.

Erteilen von Berechtigungen für den Prinzipal des Protokollierungsservices mithilfe einer Bucket-Richtlinie

Diese Bucket-Beispielrichtlinie gewährt dem Protokollierungsservice-Prinzipal (`logging.s3.amazonaws.com`) die `s3:PutObject`-Berechtigung. Wenn Sie diese Bucket-Richtlinie verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre eigenen Informationen. In der folgenden Richtlinie *`amzn-s3-demo-destination-bucket`* handelt es sich um den Ziel-Bucket, in den Serverzugriffsprotokolle übermittelt werden, und *`amzn-s3-demo-source-bucket`* um den Quell-Bucket. *`EXAMPLE-LOGGING-PREFIX`*ist das optionale Zielpräfix (auch als Zielpräfix bezeichnet), das Sie für Ihre Protokollobjekte verwenden möchten. *`SOURCE-ACCOUNT-ID`*ist der AWS-Konto , dem der Quell-Bucket gehört.

> **Note**
>
> Wenn Ihre Bucket-Richtlinie `Deny`-Anweisungen enthält, stellen Sie sicher, dass diese Anweisungen Amazon S3 nicht daran hindern, Zugriffsprotokolle bereitzustellen.

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

```
        {
            "Sid": "S3ServerAccessLogsPolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "logging.s3.amazonaws.com"
            },
            "Action": [
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket/EXAMPLE-LOGGING-
PREFIX*",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::amzn-s3-demo-source-bucket"
                },
                "StringEquals": {
                    "aws:SourceAccount": "SOURCE-ACCOUNT-ID"
                }
            }
        }
    ]
}
```

Gewähren Sie der Protokollbereitstellungsgruppe mithilfe eines Buckets Berechtigungen ACL

> **Note**
>
> Aus Sicherheitsgründen deaktiviert Amazon S3 die Zugriffskontrolllisten (ACLs) standardmäßig in allen neuen Buckets. Weitere Informationen zu ACL Berechtigungen in der Amazon S3 S3-Konsole finden Sie unterKonfiguration ACLs.

Obwohl wir diesen Ansatz nicht empfehlen, können Sie der Protokollbereitstellungsgruppe mithilfe eines Buckets Berechtigungen erteilenACL. Wenn der Ziel-Bucket jedoch die vom Bucket-Besitzer erzwungene Einstellung für Object Ownership verwendet, können Sie Bucket oder Objekt ACLs nicht festlegen. Sie können auch keine Zielzuschüsse (auch als Zielzuschüsse bezeichnet) in Ihre PutBucketLogging-Konfiguration. Stattdessen müssen Sie eine Bucket-Richtlinie verwenden, um dem Prinzipal des Protokollierungsservices (`logging.s3.amazonaws.com`) Zugriff zu gewähren. Weitere Informationen finden Sie unter Berechtigungen für Protokollbereitstellung.

Im Bucket ACL wird die Protokollbereitstellungsgruppe wie folgt dargestelltURL:

```
http://acs.amazonaws.com/groups/s3/LogDelivery
```

Um Berechtigungen `WRITE` und `READ_ACP` (ACLLese-) Berechtigungen zu erteilen, fügen Sie dem Ziel-Bucket die folgenden Berechtigungen hinzuACL:

```
<Grant>
    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
    </Grantee>
    <Permission>WRITE</Permission>
</Grant>
<Grant>
    <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:type="Group">
        <URI>http://acs.amazonaws.com/groups/s3/LogDelivery</URI>
    </Grantee>
    <Permission>READ_ACP</Permission>
</Grant>
```

Beispiele für das programmgesteuerte Hinzufügen von ACL Zuschüssen finden Sie unter. Konfiguration ACLs

> **Important**
>
> Wenn Sie die Amazon S3 S3-Serverzugriffsprotokollierung mithilfe AWS CloudFormation eines Buckets aktivieren und Zugriff auf die S3-Protokollzustellungsgruppe gewähren möchten, müssen Sie Ihrer CloudFormation Vorlage auch `"AccessControl": "LogDeliveryWrite"`hinzufügen. ACLs Dies ist wichtig, da Sie diese Berechtigungen nur gewähren können, indem Sie eine ACL für den Bucket erstellen, aber Sie können keine benutzerdefinierten Berechtigungen ACLs für Buckets in CloudFormation erstellen. Sie können nur in Dosen ACLs mit CloudFormation verwenden.

Vorgehensweise zum Aktivieren der Server-Zugriffsprotokollierung

Gehen Sie wie folgt vor, um die Serverzugriffsprotokollierung mithilfe der Amazon S3-Konsole AWS CLI, Amazon S3 RESTAPI, und zu aktivieren. AWS SDKs

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie die Server-Zugriffsprotokollierung aktivieren möchten.
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie im Abschnitt Server access logging (Server-Zugriffsprotokollierung) die Option Edit (Bearbeiten) aus.
5. Wählen Sie unter Serverzugriffsprotokollierung) die Option Aktivieren aus.
6. Geben Sie unter Ziel-Bucket einen Bucket und ein optionales Präfix an. Wenn Sie ein Präfix angeben, sollten Sie nach dem Präfix einen Schrägstrich (/) einfügen, damit Sie Ihre Protokolle leichter finden können.

   > **Note**
   >
   > Wenn Sie ein Präfix mit einem Schrägstrich (/) angeben, können Sie die Protokollobjekte leichter finden. Wenn Sie beispielsweise den Präfixwert `logs/` angeben, beginnt jedes von Amazon S3 erstellte Protokollobjekt mit dem Präfix `logs/` im Schlüssel wie folgt:
   >
   > ```
   > logs/2013-11-01-21-32-16-E568B2907131C0C0
   > ```
   >
   > Wenn Sie den Präfixwert `logs` angeben, sieht das Protokollobjekt wie folgt aus:
   >
   > ```
   > logs2013-11-01-21-32-16-E568B2907131C0C0
   > ```

7. Führen Sie unter Protokollobjekt-Schlüsselformat eine der folgenden Aktionen aus:

   - Um die non-date-based Partitionierung auszuwählen, wählen Sie [DestinationPrefixYYYY] [] - [MM] - [DD] - [hh] - [mm] - [ss] - [UniqueString].
   - Um die datumsbasierte Partitionierung zu wählen, wählen Sie [DestinationPrefixSourceAccountId] [SourceRegion]/[SourceBucket]/[YYYY]/[MM]/[DD]/[] - [MMYYYY] - [MM] - [DD] - [hh] - [mm] - [ss] - [UniqueString] und wählen Sie dann S3 event time oder Log file delivery time.

8. Wählen Sie Änderungen speichern.

Wenn Sie die Serverzugriffsprotokollierung für einen Bucket aktivieren, aktiviert die Konsole die Protokollierung für den Quell-Bucket und aktualisiert die Bucket-Richtlinie für den Ziel-Bucket, um dem Protokollierungsservice-Prinzipal (`logging.s3.amazonaws.com`) die `s3:PutObject`-Berechtigung zu gewähren. Weitere Informationen zu dieser Bucket-Richtlinie finden Sie unter Erteilen von Berechtigungen für den Prinzipal des Protokollierungsservices mithilfe einer Bucket-Richtlinie.

Sie können die Protokolle im Ziel-Bucket anzeigen. Nachdem Sie die Server-Zugriffsprotokollierung aktiviert haben, kann es einige Stunden dauern, bis die Protokolle in den Ziel-Bucket geliefert werden. Weitere Informationen darüber, wie und wann Protokolle bereitgestellt werden, finden Sie unter Wie werden Protokolle ausgeliefert?.

Weitere Informationen finden Sie unter Anzeigen der Eigenschaften eines S3-Buckets.

Verwenden Sie den REST API

Um die Protokollierung zu aktivieren, reichen Sie ein PutBucketLoggingAnfrage zum Hinzufügen der Logging-Konfiguration zum Quell-Bucket. Die Anforderung gibt den Ziel-Bucket (Ziel-Bucket) und optional das Präfix an, das für alle Protokollobjektschlüssel verwendet werden soll.

Das folgende Beispiel identifiziert sich *amzn-s3-demo-destination-bucket* als Ziel-Bucket und *logs/* als Präfix.

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <LoggingEnabled>
    <TargetBucket>amzn-s3-demo-destination-bucket</TargetBucket>
    <TargetPrefix>logs/</TargetPrefix>
  </LoggingEnabled>
</BucketLoggingStatus>
```

Das folgende Beispiel identifiziert sich *amzn-s3-demo-destination-bucket* als Ziel-Bucket, *logs/* als Präfix und `EventTime` als Schlüsselformat für das Protokollobjekt.

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
  <LoggingEnabled>
    <TargetBucket>amzn-s3-demo-destination-bucket</TargetBucket>
    <TargetPrefix>logs/</TargetPrefix>
    <TargetObjectKeyFormat>
      <PartitionedPrefix>
```

```
        <PartitionDateSource>EventTime</PartitionDateSource>
      </PartitionedPrefix>
  </TargetObjectKeyFormat>
  </LoggingEnabled>
</BucketLoggingStatus>
```

Die Protokollobjekte werden vom S3-Protokollübermittlungskonto geschrieben und besitzen dessen Eigentümer, und der Bucket-Eigentümer erhält vollständige Berechtigungen für die Protokollobjekte. Sie können optional Zielgewährungen verwenden (Zielgewährungen), um anderen Benutzern Berechtigungen zu gewähren, damit diese auf die Protokolle zugreifen können. Weitere Informationen finden Sie unter PutBucketLogging.

> **Note**
>
> Wenn der Ziel-Bucket die Einstellung „Von Bucket-Besitzer erzwungen“ für Object Ownership verwendet, können Sie keine Zielgewährungen verwenden, um anderen Benutzern Berechtigungen zu gewähren. Um anderen Benutzern Berechtigungen zu gewähren, können Sie die Bucket-Richtlinie für den Ziel-Bucket aktualisieren. Weitere Informationen finden Sie unter Berechtigungen für Protokollbereitstellung.

Um die Logging-Konfiguration für einen Bucket abzurufen, verwenden Sie GetBucketLoggingAPIOperation.

Um die Protokollierungskonfiguration zu löschen, senden Sie eine `PutBucketLogging`-Anforderung mit einem leeren `BucketLoggingStatus`:

```
<BucketLoggingStatus xmlns="http://doc.s3.amazonaws.com/2006-03-01">
</BucketLoggingStatus>
```

Um die Protokollierung in einem Bucket zu aktivieren, können Sie entweder die Amazon S3 API - oder die AWS SDK Wrapper-Bibliotheken verwenden.

Mit dem AWS SDKs

In den folgenden Beispielen wird die Protokollierung für einen Bucket aktiviert. Sie müssen zwei Buckets erstellen, einen Quell-Bucket und einen Ziel-Bucket. In den Beispielen wird zuerst der Bucket ACL im Ziel-Bucket aktualisiert. Anschließend werden der Protokollbereitstellungsgruppe die notwendigen Berechtigungen für das Schreiben von Protokollen zum Ziel-Bucket gewährt, bevor die Protokollierung für den Quell-Bucket aktiviert wird.

Diese Beispiele funktionieren nicht für Ziel-Buckets, die die Einstellung „Von Bucket-Besitzer erzwungen“ für Object Ownership verwenden.

Wenn der Ziel-Bucket (Ziel-Bucket) die vom Bucket-Besitzer erzwungene Einstellung für Objekteigentum verwendet, können Sie weder Bucket noch Objekt ACLs festlegen. Sie können auch keine Zielzuschüsse (Zielzuschüsse) in Ihre PutBucketLoggingKonfiguration aufnehmen. Sie müssen eine Bucket-Richtlinie verwenden, um dem Protokollierungsserviceprinzipal (`logging.s3.amazonaws.com`) Zugriff zu gewähren. Weitere Informationen finden Sie unter Berechtigungen für Protokollbereitstellung.

.NET

AWS SDK for .NET

> **Note**
>
> Es gibt noch mehr dazu GitHub. Sie sehen das vollständige Beispiel und erfahren, wie Sie das AWS -Code-Beispiel-Repository einrichten und ausführen.

```
using System;
using System.IO;
using System.Threading.Tasks;
using Amazon.S3;
using Amazon.S3.Model;
using Microsoft.Extensions.Configuration;

/// <summary>
/// This example shows how to enable logging on an Amazon Simple Storage
/// Service (Amazon S3) bucket. You need to have two Amazon S3 buckets for
/// this example. The first is the bucket for which you wish to enable
/// logging, and the second is the location where you want to store the
/// logs.
/// </summary>
public class ServerAccessLogging
{
    private static IConfiguration _configuration = null!;

    public static async Task Main()
    {
        LoadConfig();
```

```
            string bucketName = _configuration["BucketName"];
            string logBucketName = _configuration["LogBucketName"];
            string logObjectKeyPrefix = _configuration["LogObjectKeyPrefix"];
            string accountId = _configuration["AccountId"];

            // If the AWS Region defined for your default user is different
            // from the Region where your Amazon S3 bucket is located,
            // pass the Region name to the Amazon S3 client object's constructor.
            // For example: RegionEndpoint.USWest2 or RegionEndpoint.USEast2.
            IAmazonS3 client = new AmazonS3Client();

            try
            {
                // Update bucket policy for target bucket to allow delivery of
logs to it.
                await SetBucketPolicyToAllowLogDelivery(
                    client,
                    bucketName,
                    logBucketName,
                    logObjectKeyPrefix,
                    accountId);

                // Enable logging on the source bucket.
                await EnableLoggingAsync(
                    client,
                    bucketName,
                    logBucketName,
                    logObjectKeyPrefix);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine($"Error: {e.Message}");
            }
        }

        /// <summary>
        /// This method grants appropriate permissions for logging to the
        /// Amazon S3 bucket where the logs will be stored.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client which will be
used
        /// to apply the bucket policy.</param>
        /// <param name="sourceBucketName">The name of the source bucket.</param>
```

```
        /// <param name="logBucketName">The name of the bucket where logging
        /// information will be stored.</param>
        /// <param name="logPrefix">The logging prefix where the logs should be
delivered.</param>
        /// <param name="accountId">The account id of the account where the
source bucket exists.</param>
        /// <returns>Async task.</returns>
        public static async Task SetBucketPolicyToAllowLogDelivery(
            IAmazonS3 client,
            string sourceBucketName,
            string logBucketName,
            string logPrefix,
            string accountId)
        {
            var resourceArn = @""arn:aws:s3:::" + logBucketName + "/" +
logPrefix + @"*""";

            var newPolicy = @"{
                                ""Statement"":[{
                                ""Sid"": ""S3ServerAccessLogsPolicy"",
                                ""Effect"": ""Allow"",
                                ""Principal"": { ""Service"":
""logging.s3.amazonaws.com"" },
                                ""Action"": [""s3:PutObject""],
                                ""Resource"": [" + resourceArn + @"],
                                ""Condition"": {
                                ""ArnLike"": { ""aws:SourceArn"":
""arn:aws:s3:::" + sourceBucketName + @""" },
                                ""StringEquals"": { ""aws:SourceAccount"": """ +
accountId + @""" }
                                        }
                                    }]
                                }";
            Console.WriteLine($"The policy to apply to bucket {logBucketName} to
enable logging:");
            Console.WriteLine(newPolicy);

            PutBucketPolicyRequest putRequest = new PutBucketPolicyRequest
            {
                BucketName = logBucketName,
                Policy = newPolicy,
            };
            await client.PutBucketPolicyAsync(putRequest);
            Console.WriteLine("Policy applied.");
```

```
        }

        /// <summary>
        /// This method enables logging for an Amazon S3 bucket. Logs will be
 stored
        /// in the bucket you selected for logging. Selected prefix
        /// will be prepended to each log object.
        /// </summary>
        /// <param name="client">The initialized Amazon S3 client which will be
 used
        /// to configure and apply logging to the selected Amazon S3 bucket.</
param>
        /// <param name="bucketName">The name of the Amazon S3 bucket for which
 you
        /// wish to enable logging.</param>
        /// <param name="logBucketName">The name of the Amazon S3 bucket where
 logging
        /// information will be stored.</param>
        /// <param name="logObjectKeyPrefix">The prefix to prepend to each
        /// object key.</param>
        /// <returns>Async task.</returns>
        public static async Task EnableLoggingAsync(
            IAmazonS3 client,
            string bucketName,
            string logBucketName,
            string logObjectKeyPrefix)
        {
            Console.WriteLine($"Enabling logging for bucket {bucketName}.");
            var loggingConfig = new S3BucketLoggingConfig
            {
                TargetBucketName = logBucketName,
                TargetPrefix = logObjectKeyPrefix,
            };

            var putBucketLoggingRequest = new PutBucketLoggingRequest
            {
                BucketName = bucketName,
                LoggingConfig = loggingConfig,
            };
            await client.PutBucketLoggingAsync(putBucketLoggingRequest);
            Console.WriteLine($"Logging enabled.");
        }

        /// <summary>
```

```
        /// Loads configuration from settings files.
        /// </summary>
        public static void LoadConfig()
        {
            _configuration = new ConfigurationBuilder()
                .SetBasePath(Directory.GetCurrentDirectory())
                .AddJsonFile("settings.json") // Load settings from .json file.
                .AddJsonFile("settings.local.json", true) // Optionally, load
  local settings.
                .Build();
        }
    }
```

- APIEinzelheiten finden Sie PutBucketLoggingin der AWS SDK for .NET APIReferenz.

Java

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.BucketLoggingStatus;
import software.amazon.awssdk.services.s3.model.LoggingEnabled;
import software.amazon.awssdk.services.s3.model.PartitionedPrefix;
import software.amazon.awssdk.services.s3.model.PutBucketLoggingRequest;
import software.amazon.awssdk.services.s3.model.TargetObjectKeyFormat;

// Class to set a bucket policy on a target S3 bucket and enable server access
 logging on a source S3 bucket.
public class ServerAccessLogging {
    private static S3Client s3Client;

    public static void main(String[] args) {
        String sourceBucketName = "SOURCE-BUCKET";
        String targetBucketName = "TARGET-BUCKET";
        String sourceAccountId = "123456789012";
        String targetPrefix = "logs/";

        // Create S3 Client.
        s3Client = S3Client.builder().
                region(Region.US_EAST_2)
                .build();
```

```
        // Set a bucket policy on the target S3 bucket to enable server access
 logging by granting the
        // logging.s3.amazonaws.com principal permission to use the PutObject
 operation.
        ServerAccessLogging serverAccessLogging = new ServerAccessLogging();
        serverAccessLogging.setTargetBucketPolicy(sourceAccountId, sourceBucketName,
 targetBucketName);

        // Enable server access logging on the source S3 bucket.
        serverAccessLogging.enableServerAccessLogging(sourceBucketName,
 targetBucketName,
                targetPrefix);

    }

    // Function to set a bucket policy on the target S3 bucket to enable server
 access logging by granting the
    // logging.s3.amazonaws.com principal permission to use the PutObject operation.
    public void setTargetBucketPolicy(String sourceAccountId, String
 sourceBucketName, String targetBucketName) {
        String policy = "{\n" +
                "    \"Version\": \"2012-10-17\",\n" +
                "    \"Statement\": [\n" +
                "        {\n" +
                "            \"Sid\": \"S3ServerAccessLogsPolicy\",\n" +
                "            \"Effect\": \"Allow\",\n" +
                "            \"Principal\": {\"Service\": \"logging.s3.amazonaws.com
\"},\n" +
                "            \"Action\": [\n" +
                "                \"s3:PutObject\"\n" +
                "            ],\n" +
                "            \"Resource\": \"arn:aws:s3:::" + targetBucketName + "/*
\",\n" +
                "            \"Condition\": {\n" +
                "                \"ArnLike\": {\n" +
                "                    \"aws:SourceArn\": \"arn:aws:s3:::" +
 sourceBucketName + "\"\n" +
                "                },\n" +
                "                \"StringEquals\": {\n" +
                "                    \"aws:SourceAccount\": \"" + sourceAccountId +
 "\"\n" +
                "                }\n" +
                "            }\n" +
                "        }\n" +
```

```
                "    ]\n" +
                "}";
        s3Client.putBucketPolicy(b -> b.bucket(targetBucketName).policy(policy));
    }

    // Function to enable server access logging on the source S3 bucket.
    public void enableServerAccessLogging(String sourceBucketName, String
 targetBucketName,
            String targetPrefix) {
        TargetObjectKeyFormat targetObjectKeyFormat =
 TargetObjectKeyFormat.builder()

 .partitionedPrefix(PartitionedPrefix.builder().partitionDateSource("EventTime").build())
                .build();
        LoggingEnabled loggingEnabled = LoggingEnabled.builder()
                .targetBucket(targetBucketName)
                .targetPrefix(targetPrefix)
                .targetObjectKeyFormat(targetObjectKeyFormat)
                .build();
        BucketLoggingStatus bucketLoggingStatus = BucketLoggingStatus.builder()
                .loggingEnabled(loggingEnabled)
                .build();
        s3Client.putBucketLogging(PutBucketLoggingRequest.builder()
                .bucket(sourceBucketName)
                .bucketLoggingStatus(bucketLoggingStatus)
                .build());
    }

}
```

Unter Verwendung des AWS CLI

Wir empfehlen, dass Sie in jedem, in dem Sie S3-Buckets haben AWS-Region , einen eigenen Logging-Bucket erstellen. Anschließend können Sie Ihre Amazon-S3-Zugriffsprotokolle in diesem S3-Bucket bereitstellen lassen. Weitere Informationen und Beispiele finden Sie unter put-bucket-loggingin der AWS CLI Referenz.

Wenn der Ziel-Bucket (Ziel-Bucket) die vom Bucket-Besitzer erzwungene Einstellung für Objekteigentum verwendet, können Sie weder Bucket noch Objekt ACLs festlegen. Sie können auch keine Zielzuschüsse (Zielzuschüsse) in Ihre PutBucketLoggingKonfiguration aufnehmen. Sie müssen eine Bucket-Richtlinie verwenden, um dem Protokollierungsserviceprinzipal

(`logging.s3.amazonaws.com`) Zugriff zu gewähren. Weitere Informationen finden Sie unter Berechtigungen für Protokollbereitstellung.

Example – Aktivieren von Zugriffsprotokollen für fünf Buckets in zwei Regionen

In diesem Beispiel werden die folgenden fünf Buckets verwendet:

- *amzn-s3-demo-source-bucket*-us-east-1
- *amzn-s3-demo-source-bucket1*-us-east-1
- *amzn-s3-demo-source-bucket2*-us-east-1
- *amzn-s3-demo-bucket1*-us-west-2
- *amzn-s3-demo-bucket2*-us-west-2

> **Note**
>
> Der letzte Schritt im folgenden Verfahren enthält Bash-Beispielskripts, mit denen Sie Ihre Protokollierungs-Buckets erstellen und die Serverzugriffsprotokollierung für diese Buckets aktivieren können. Um diese Skripts verwenden zu können, müssen Sie die Dateien `policy.json` und `logging.json` erstellen wie im folgenden Verfahren beschrieben.

1. Erstellen Sie in den Regionen USA West (Oregon) und USA Ost (Nord-Virginia) zwei Ziel-Buckets für die Protokollierung mit den folgenden Namen:
   - *amzn-s3-demo-destination-bucket*-logs-us-east-1
   - *amzn-s3-demo-destination-bucket1*-logs-us-west-2
2. Später im Verfahren aktivieren Sie die Serverzugriffsprotokollierung wie folgt:
   - *amzn-s3-demo-source-bucket*-us-east-1 protokolliert zum S3-Bucket *amzn-s3-demo-destination-bucket*-logs-us-east-1 mit dem Präfix *amzn-s3-demo-source-bucket*-us-east-1
   - *amzn-s3-demo-source-bucket1*-us-east-1 protokolliert zum S3-Bucket *amzn-s3-demo-destination-bucket*-logs-us-east-1 mit dem Präfix *amzn-s3-demo-source-bucket1*-us-east-1

- *amzn-s3-demo-source-bucket2*-us-east-1 protokolliert zum S3-Bucket *amzn-s3-demo-destination-bucket*-logs-us-east-1 mit dem Präfix *amzn-s3-demo-source-bucket2*-us-east-1
- *amzn-s3-demo-bucket1*-us-west-2 protokolliert zum S3-Bucket *amzn-s3-demo-destination-bucket1*-logs-us-west-2 mit dem Präfix *amzn-s3-demo-bucket1*-us-west-2
- *amzn-s3-demo-bucket2*-us-west-2 protokolliert zum S3-Bucket *amzn-s3-demo-destination-bucket1*-logs-us-west-2 mit dem Präfix *amzn-s3-demo-bucket2*-us-west-2

3. Erteilen Sie für jeden Ziel-Logging-Bucket mithilfe eines Buckets ACL oder einer Bucket-Richtlinie Berechtigungen für die Übermittlung von Serverzugriffsprotokollen:

   - Aktualisieren Sie die Bucket-Richtlinie (Empfohlen) – Verwenden Sie den folgenden `put-bucket-policy`-Befehl, um dem Prinzipal des Protokollierungsservices Berechtigungen zu erteilen. Ersetzen Sie *amzn-s3-demo-destination-bucket*-logs durch den Namen Ihres Ziel-Buckets.

     ```
     aws s3api put-bucket-policy --bucket amzn-s3-demo-destination-bucket-logs --
     policy file://policy.json
     ```

     `Policy.json`ist ein JSON Dokument im aktuellen Ordner, das die folgende Bucket-Richtlinie enthält. Wenn Sie diese Bucket-Richtlinie verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre eigenen Informationen. In der folgenden Richtlinie *amzn-s3-demo-destination-bucket-Protokolle* ist der Ziel-Bucket, in den Serverzugriffsprotokolle übermittelt werden, und *amzn-s3-demo-source-bucket* ist der Quell-Bucket. *SOURCE-ACCOUNT-ID*ist der AWS-Konto , dem der Quell-Bucket gehört.

     ```
     {
         "Version": "2012-10-17",
         "Statement": [
             {
                 "Sid": "S3ServerAccessLogsPolicy",
                 "Effect": "Allow",
                 "Principal": {
                     "Service": "logging.s3.amazonaws.com"
                 },
                 "Action": [
     ```

```
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket-logs/*",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::amzn-s3-demo-source-bucket"
                },
                "StringEquals": {
                    "aws:SourceAccount": "SOURCE-ACCOUNT-ID"
                }
            }
        }
    ]
}
```

- Bucket aktualisieren ACL — Verwenden Sie den folgenden `put-bucket-acl` Befehl, um der S3-Protokollzustellungsgruppe Berechtigungen zu erteilen. Ersetzen *amzn-s3-demo-destination-bucket-loggt* mit dem Namen Ihres Ziel-Buckets (Ziel-Buckets).

```
aws s3api put-bucket-acl --bucket amzn-s3-demo-destination-bucket-logs  --
grant-write URI=http://acs.amazonaws.com/groups/s3/LogDelivery --grant-read-acp
 URI=http://acs.amazonaws.com/groups/s3/LogDelivery
```

4. Erstellen Sie dann eine `logging.json`-Datei, die Ihre Protokollierungskonfiguration enthält (basierend auf einem der drei folgenden Beispiele). Nach der Erstellung der `logging.json`-Datei können Sie mit dem folgenden `put-bucket-logging`-Befehl die Protokollierungskonfiguration anwenden. Ersetzen *amzn-s3-demo-destination-bucket-Logs* mit dem Namen Ihres Ziel-Buckets (Ziel-Buckets).

```
aws s3api put-bucket-logging --bucket amzn-s3-demo-destination-bucket-logs --
bucket-logging-status file://logging.json
```

> **Note**
>
> Anstatt mit diesem diesen `put-bucket-logging`-Befehl die Protokollierungskonfiguration auf jeden einzelnen Ziel-Bucket anzuwenden, können Sie

eines der im nächsten Schritt bereitgestellten Bash-Skripts verwenden. Um diese Skripts verwenden zu können, müssen Sie die Dateien `policy.json` und `logging.json` erstellen wie in diesem Verfahren beschrieben.

Die `logging.json` Datei ist ein JSON Dokument im aktuellen Ordner, das Ihre Logging-Konfiguration enthält. Wenn ein Ziel-Bucket die Einstellung „Von Bucket-Besitzer erzwungen“ für Object Ownership verwendet, kann Ihre Protokollierungskonfiguration keine Zielgewährungen enthalten. Weitere Informationen finden Sie unter Berechtigungen für Protokollbereitstellung.

Example – **`logging.json`** ohne Zielgewährungen

Die folgende `logging.json`-Beispieldatei enthält keine Zielgewährungen. Daher können Sie diese Konfiguration auf einen Ziel-Bucket anwenden, der die Einstellung „Von Bucket-Besitzer erzwungen“ für Objekt Ownership verwendet.

```
{
    "LoggingEnabled": {
        "TargetBucket": "amzn-s3-demo-destination-bucket-logs",
        "TargetPrefix": "amzn-s3-demo-destination-bucket/"
     }
 }
```

Example – **`logging.json`** mit Zielgewährungen

Die folgende `logging.json`-Beispieldatei enthält Zielgewährungen.

Wenn der Ziel-Bucket die vom Bucket-Besitzer erzwungene Einstellung für Objekteigentum verwendet, können Sie Zielzuschüsse (Zielzuschüsse) nicht in Ihre PutBucketLogging-Konfiguration. Weitere Informationen finden Sie unter Berechtigungen für Protokollbereitstellung.

```
{
    "LoggingEnabled": {
        "TargetBucket": "amzn-s3-demo-destination-bucket-logs",
        "TargetPrefix": "amzn-s3-demo-destination-bucket/",
        "TargetGrants": [
             {
```

```
                "Grantee": {
                    "Type": "AmazonCustomerByEmail",
                    "EmailAddress": "user@example.com"
                 },
                "Permission": "FULL_CONTROL"
            }
        ]
    }
}
```

Example – **`logging.json`** mit auf S3-Ereigniszeit festgelegtem Protokollobjekt-Schlüsselformat

In der folgenden `logging.json`-Datei wird das Protokollobjekt-Schlüsselformat in S3-Ereigniszeit geändert. Weitere Informationen zum Einrichten des Protokollobjekt-Schlüsselformats finden Sie unter the section called “Wie aktiviere ich die Protokollzustellung?”.

```
{
  "LoggingEnabled": {
      "TargetBucket": "amzn-s3-demo-destination-bucket-logs",
      "TargetPrefix": "amzn-s3-demo-destination-bucket/",
      "TargetObjectKeyFormat": {
          "PartitionedPrefix": {
              "PartitionDateSource": "EventTime"
          }
       }
  }
}
```

5. Sie können eines der folgenden Bash-Skripts verwenden, um allen Buckets in Ihrem Konto Zugriffsprotokollierung hinzuzufügen. Ersetzen *`amzn-s3-demo-destination-bucket-Logs`* mit dem Namen Ihres Ziel-Buckets (Ziel-Buckets) und *`us-west-2`* ersetzen Sie es durch den Namen der Region, in der sich Ihre Buckets befinden.

> **Note**
>
> Dieses Skript funktioniert nur, wenn sich alle Buckets in derselben Region befinden. Sind die Buckets auf mehrere Regionen verteilt, müssen Sie das Skript anpassen.

Example - Gewähren Sie Zugriff mit Bucket-Richtlinien und fügen Sie die Protokollierung für die Buckets in Ihrem Konto hinzu

```
  loggingBucket='amzn-s3-demo-destination-bucket-logs'
  region='us-west-2'


  # Create the logging bucket.
  aws s3 mb s3://$loggingBucket --region $region

  aws s3api put-bucket-policy --bucket $loggingBucket --policy file://policy.json

  # List the buckets in this account.
  buckets="$(aws s3 ls | awk '{print $3}')"

  # Put a bucket logging configuration on each bucket.
  for bucket in $buckets
      do
        # This if statement excludes the logging bucket.
        if [ "$bucket" != "$loggingBucket" ] ; then
            continue;
        fi
        printf '{
          "LoggingEnabled": {
            "TargetBucket": "%s",
            "TargetPrefix": "%s/"
        }
      }' "$loggingBucket" "$bucket"  > logging.json
      aws s3api put-bucket-logging --bucket $bucket --bucket-logging-status file://
logging.json
      echo "$bucket done"
  done

  rm logging.json

  echo "Complete"
```

Example — Gewähren Sie Zugriff mit einem Bucket ACLs und fügen Sie die Protokollierung für die Buckets in Ihrem Konto hinzu

```
  loggingBucket='amzn-s3-demo-destination-bucket-logs'
  region='us-west-2'


  # Create the logging bucket.
  aws s3 mb s3://$loggingBucket --region $region

  aws s3api put-bucket-acl --bucket $loggingBucket --grant-write URI=http://
acs.amazonaws.com/groups/s3/LogDelivery --grant-read-acp URI=http://
acs.amazonaws.com/groups/s3/LogDelivery

  # List the buckets in this account.
  buckets="$(aws s3 ls | awk '{print $3}')"

  # Put a bucket logging configuration on each bucket.
  for bucket in $buckets
      do
        # This if statement excludes the logging bucket.
        if [ "$bucket" != "$loggingBucket" ] ; then
            continue;
        fi
        printf '{
          "LoggingEnabled": {
            "TargetBucket": "%s",
            "TargetPrefix": "%s/"
        }
      }' "$loggingBucket" "$bucket"  > logging.json
      aws s3api put-bucket-logging --bucket $bucket --bucket-logging-status file://
logging.json
      echo "$bucket done"
  done

  rm logging.json

  echo "Complete"
```

Überprüfen der Einrichtung Ihrer Serverzugriffsprotokolle

Führen Sie nach dem Aktivieren der Serverzugriffsprotokollierung die folgenden Schritte aus:

- Greifen Sie auf den Ziel-Bucket zu und überprüfen Sie, ob die Protokolldateien bereitgestellt werden. Nachdem die Zugriffsprotokolle eingerichtet wurden, fängt Amazon S3 sofort an, Anfragen zu erfassen und zu protokollieren. Es kann jedoch einige Stunden dauern, bis die Protokolle an den Ziel-Bucket übermittelt werden. Weitere Informationen erhalten Sie unter the section called “Statusänderungen in der Bucket-Protokollierung werden mit der Zeit wirksam” und the section called “Best-Effort-Serverprotokollbereitstellung”.

  Sie können die Protokollzustellung auch automatisch überprüfen, indem Sie Amazon S3 S3-Anforderungsmetriken verwenden und CloudWatch Amazon-Alarme für diese Metriken einrichten. Weitere Informationen finden Sie unter Metriken mit Amazon überwachen CloudWatch.
- Vergewissern Sie sich, dass Sie den Inhalt der Protokolldateien öffnen und lesen können.

Informationen zur Fehlerbehebung bei der Serverzugriffsprotokollierung finden Sie unter Behebung von Fehlern bei der Server-Zugriffsprotokollierung.

Amazon-S3-Server-Zugriffsprotokollformat

Die Server-Zugriffsprotokollierung bietet detaillierte Aufzeichnungen über die Anfragen, die an einen Amazon-S3-Bucket gestellt wurden. Sie können Serverzugriffsprotokolle für folgende Zwecke verwenden:

- Durchführung von Sicherheits- und Zugriffsprüfungen
- Kennenlernen Ihres Kundenstamms
- Nachvollziehen Ihrer Amazon-S3-Rechnung

Dieser Abschnitt beschreibt das Format und andere Details zu Amazon-S3-Serverzugriffprotokolldateien.

Die Server-Zugriffsprotokolldateien bestehen aus einer Reihe von durch Zeilenschaltungen voneinander getrennten Protokolldatensätzen. Jeder Protokolldatensatz stellt eine Anforderung dar und besteht aus durch Leerzeichen voneinander getrennter Felder.

Nachfolgend wird ein Beispielprotokoll mit sechs Protokolldatensätzen gezeigt.

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 3E57427F3EXAMPLE
 REST.GET.VERSIONING - "GET /amzn-s3-demo-bucket1?versioning HTTP/1.1" 200 - 113 - 7 -
 "-" "S3Console/0.4" - s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/
XV/VLi31234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader amzn-s3-demo-bucket1.s3.us-
west-1.amazonaws.com TLSV1.2 arn:aws:s3:us-west-1:123456789012:accesspoint/example-AP
 Yes
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 891CE47D2EXAMPLE
 REST.GET.LOGGING_STATUS - "GET /amzn-s3-demo-bucket1?logging HTTP/1.1" 200 -
 242 - 11 - "-" "S3Console/0.4" - 9vKBE6vMhrNiWHZmb2L0mXOcqPGzQOI5XLnCtZNPxev+Hf
+7tpT6sxDwDty4LHBUOZJG96N1234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader amzn-s3-
demo-bucket1.s3.us-west-1.amazonaws.com TLSV1.2 - -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket1 [06/Feb/2019:00:00:38 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be A1206F460EXAMPLE
 REST.GET.BUCKETPOLICY - "GET /amzn-s3-demo-bucket1?policy HTTP/1.1" 404
 NoSuchBucketPolicy 297 - 38 - "-" "S3Console/0.4" - BNaBsXZQQDbssi6xMBdBU2sLt
+Yf5kZDmeBUP35sFoKa3sLLeMC78iwEIWxs99CRUrbS4n11234= SigV4 ECDHE-RSA-AES128-GCM-SHA256
 AuthHeader amzn-s3-demo-bucket1.s3.us-west-1.amazonaws.com TLSV1.2 - Yes
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket1 [06/Feb/2019:00:01:00 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 7B4A0FABBEXAMPLE
 REST.GET.VERSIONING - "GET /amzn-s3-demo-bucket1?versioning HTTP/1.1" 200 -
 113 - 33 - "-" "S3Console/0.4" - Ke1bUcazaN1jWuUlPJaxF64cQVpUEhoZKEG/hmy/gijN/
I1DeWqDfFvnpybfEseEME/u7ME1234= SigV4 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader amzn-s3-
demo-bucket1.s3.us-west-1.amazonaws.com TLSV1.2 - -
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket1 [06/Feb/2019:00:01:57 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 DD6CC733AEXAMPLE REST.PUT.OBJECT s3-dg.pdf "PUT /amzn-s3-demo-bucket1/
s3-dg.pdf HTTP/1.1" 200 - - 4406583 41754 28 "-" "S3Console/0.4" -
 10S62Zv81kBW7BB6SX4XJ48o6kpcl6LPwEoizZQQxJd5qDSCTLX0TgS37kYUBKQW3+bPdrg1234= SigV4
 ECDHE-RSA-AES128-SHA AuthHeader amzn-s3-demo-bucket1.s3.us-west-1.amazonaws.com
 TLSV1.2 - Yes
```

> **Note**
>
> Jedes Feld kann auf - gesetzt werden, um darauf hinzuweisen, dass die Daten unbekannt oder nicht verfügbar waren, oder dass das Feld für diese Anforderung nicht relevant war.

Themen

- Protokolldatensatzfelder
- Zusätzliche Protokollierung für KopierVorgänge
- Benutzerdefinierte Zugriffsprotokollinformationen
- Aspekte zur Programmierung des erweiterbaren Serverzugriff-Protokollformats

Protokolldatensatzfelder

In der folgenden Liste werden die wichtigsten Protokolldatensatzfelder beschrieben.

Bucket-Eigentümer

Die kanonische Benutzer-ID des Eigentümer des Quell-Buckets. Die kanonische Benutzer-ID ist eine andere Form der AWS-Konto ID. Weitere Informationen zur kanonischen Benutzer-ID finden Sie unter AWS-Konto -Kennungen in der Allgemeine AWS-Referenz. Informationen darüber, wie Sie die kanonische Benutzer-ID für Ihr Konto finden, finden Sie unter Finding the canonical user ID for your AWS-Konto(Die kanonische Benutzer-ID für Ihr AWS-Konto finden).

Beispieleintrag

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Bucket

Der Name des Buckets, für den die Anforderung verarbeitet wurde. Wenn das System eine fehlerhaft aufgebaute Anforderung erhält und den Bucket nicht bestimmen kann, erscheint die Anforderung nicht in einem Server-Zugriffsprotokoll.

Beispieleintrag

```

```

Zeit

Die Uhrzeit, zu der die Anfrage eingegangen ist. Diese Daten und Uhrzeiten sind in koordinierter Weltzeit () UTC angegeben. Das Format unter Verwendung der `strftime()`-Terminologie, nämlich: `[%d/%b/%Y:%H:%M:%S %z]`

Beispieleintrag

```
[06/Feb/2019:00:00:38 +0000]
```

Remote-IP

Die offensichtliche IP-Adresse des Anforderers. Auf dem Weg vorhandene Proxy-Server und Firewalls könnten die tatsächliche IP-Adresse des Computers verbergen, der die Anforderung gestellt hat.

Beispieleintrag

```
192.0.2.3
```

Auftraggeber

Die kanonische Benutzer-ID des Auftraggebers, oder - für nicht authentifizierte Anforderungen. Wenn der Anforderer ein IAM Benutzer war, gibt dieses Feld den Benutzernamen des Anfragenden zusammen mit dem Namen zurück AWS-Konto , zu dem der IAM Benutzer gehört. IAM Diese ID ist dieselbe, die für den Zugriff zu Kontrollzwecken verwendet wird.

Beispieleintrag

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Wenn der Anforderer eine angenommene Rolle verwendet, gibt dieses Feld die angenommene Rolle zurück. IAM

Beispieleintrag

```
arn:aws:sts::123456789012:assumed-role/roleName/test-role
```

Anforderungs-ID

Eine von Amazon S3 generierte Zeichenfolge, die jede Anforderung eindeutig identifiziert.

Beispieleintrag

```
3E57427F33A59F07
```

Operation

Die hier aufgeführte Operation ist deklariert al `SOAP.operation`, `REST.HTTP_method.resource_type`, `WEBSITE.HTTP_method.resource_type` oder `BATCH.DELETE.OBJECT`, oder `S3.action.resource_type` für S3-Lebenszyklus und Protokollierung.

Beispieleintrag

```
REST.PUT.OBJECT
```

Schlüssel

Der Schlüsselteil (Objektname) der Anfrage.

Beispieleintrag

```
/photos/2019/08/puppy.jpg
```

Request-URI

Der `Request-URI` Teil der HTTP Anforderungsnachricht.

Beispieleintrag

```
"GET /amzn-s3-demo-bucket1/photos/2019/08/puppy.jpg?x-foo=bar HTTP/1.1"
```

HTTPStatus

Der numerische HTTP Statuscode der Antwort.

Beispieleintrag

```
200
```

Fehlercode

Die Amazon S3 S3-Fehlerantworten oder - wenn kein Fehler aufgetreten ist.

Beispieleintrag

```
NoSuchBucket
```

Gesendete Bytes

Die Anzahl der gesendeten Antwortbytes, ohne HTTP Protokoll-Overhead, oder - falls Null.

Beispieleintrag

```
2662992
```

Objektgröße

Die Gesamtgröße des betreffenden Objekts.

Beispieleintrag

```
3462992
```

Gesamtzeit

Die Anzahl der Millisekunden (ms), die die Anforderung aus Perspektive des Servers unterwegs war. Dieser Wert wird ab der Zeit gemessen, zu der Ihre Anforderung empfangen wurde, bis zu der Zeit, zu der das letzte Byte der Antwort gesendet wurde. Messungen aus der Perspektive des Clients dauern möglicherweise länger aufgrund der Netzwerklatenz.

Beispieleintrag

```
70
```

Umschlagzeit

Die Anzahl der Millisekunden, die Amazon S3 gebraucht hat, Ihre Anfrage zu verarbeiten. Dieser Wert wird ab der Zeit gemessen, zu der das letzte Byte Ihrer Anforderung empfangen wurde, bis zu der Zeit, zu der das erste Byte der Antwort gesendet wurde.

Beispieleintrag

```
10
```

Referer

Der Wert des HTTP `Referer` Headers, falls vorhanden. HTTPBenutzeragenten (z. B. Browser) setzen diesen Header normalerweise auf den Header URL der verlinkenden oder einbettenden Seite, wenn sie eine Anfrage stellen.

Beispieleintrag

```
"http://www.example.com/webservices"
```

User-Agent

Der Wert des Headers. HTTP `User-Agent`

Beispieleintrag

```
"curl/7.15.1"
```

Versions-ID

Die Versions-ID der Anforderung oder `-`, wenn die Operation keinen `versionId`-Parameter entgegennimmt.

Beispieleintrag

```
3HL4kqtJvjVBH40Nrjfkd
```

Host-ID

Die erweiterte Anforderungs-ID für `x-amz-id-2` oder Amazon S3

Beispieleintrag

```
s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi31234=
```

Signatur-Version

Die Signaturversion, `SigV2` oder `SigV4`, die für die Authentifizierung der Anforderung verwendet wurde, bzw. ein `-` für nicht authentifizierte Anforderungen.

Beispieleintrag

```
SigV2
```

Cipher Suite

Die Secure Sockets Layer (SSL) -Verschlüsselung, die für eine HTTPS Anfrage oder ein `-` für ausgehandelt wurde. HTTP

Beispieleintrag

```
ECDHE-RSA-AES128-GCM-SHA256
```

Authentifizierungstyp

Die Art der verwendeten Anforderungsauthentifizierung: `AuthHeader` für Authentifizierungs-Header, `QueryString` für Abfragezeichenfolgen (vorsigniertURL) oder a `-` für nicht authentifizierte Anfragen.

Beispieleintrag

```
AuthHeader
```

Host Header

Der für die Verbindung mit Amazon S3 verwendete Endpunkt.

Beispieleintrag

```
s3.us-west-2.amazonaws.com
```

Einige frühere Regionen unterstützen Legacy-Endpunkte. Möglicherweise finden Sie diese Endpunkte in Ihren Serverzugriffsprotokollen oder -protokollen. AWS CloudTrail Weitere Informationen finden Sie unter Legacy-Endpunkte. Eine vollständige Liste der Amazon-S3-Regionen und -Endpunkte finden Sie unter Endpunkte und Kontingente von Amazon S3 in der Allgemeine Amazon Web Services-Referenz.

TLSVersion

Die vom Client ausgehandelte Version von Transport Layer Security (TLS). Der Wert ist einer der folgenden:`TLSv1.1`,, `TLSv1.2``TLSv1.3`, oder `-` falls TLS er nicht verwendet wurde.

Beispieleintrag

```
TLSv1.2
```

Zugriffspunkt ARN

Der Amazon-Ressourcenname (ARN) des Zugriffspunkts der Anfrage. Wenn der Access Point falsch ARN formatiert ist oder nicht verwendet wird, enthält das Feld eine`-`. Weitere Hinweise

zu Zugangspunkten finden Sie unter Verwenden von Zugriffspunkten. Weitere Informationen zu ARNs finden Sie unter Amazon Resource Name (ARN) im AWS Referenzhandbuch.

Beispieleintrag

```
arn:aws:s3:us-east-1:123456789012:accesspoint/example-AP
```

aclRequired

Eine Zeichenfolge, die angibt, ob für die Anfrage eine Zugriffskontrollliste (ACL) für die Autorisierung erforderlich war. Wenn ACL für die Anfrage eine Autorisierung erforderlich war, lautet die ZeichenfolgeYes. Wenn keine erforderlich ACLs waren, lautet die Zeichenfolge-. Weitere Informationen zu finden ACLs Sie unterÜbersicht über die Zugriffskontrollliste (ACL). Weitere Informationen zur Verwendung des aclRequired Felds zum Deaktivieren ACLs finden Sie unterKontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

Beispieleintrag

```
Yes
```

Zusätzliche Protokollierung für KopierVorgänge

Eine Kopieroperation umfasst ein GET und ein PUT. Aus diesem Grund zeichnen wir für eine Kopieroperation zwei Datensätze auf. Der vorherige Abschnitt beschreibt die Felder für den PUT-Teil der Operation. Die folgende Liste beschreibt die Felder in dem Datensatz, die sich auf den GET-Teil der Kopieroperation beziehen.

Bucket-Eigentümer

Die kanonische Benutzer-ID des Buckets, der das kopierte Objekt speichert. Die kanonische Benutzer-ID ist eine andere Form der AWS-Konto ID. Weitere Informationen zur kanonischen Benutzer-ID finden Sie unter AWS-Konto -Kennungen in der Allgemeine AWS-Referenz. Informationen darüber, wie Sie die kanonische Benutzer-ID für Ihr Konto finden, finden Sie unter Finding the canonical user ID for your AWS-Konto(Die kanonische Benutzer-ID für Ihr AWS-Konto finden).

Beispieleintrag

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Bucket

Der Name des Buckets, der das kopierte Objekt speichert.

Beispieleintrag

```
```

Zeit

Die Uhrzeit, zu der die Anfrage eingegangen ist. Diese Daten und Uhrzeiten sind in koordinierter Weltzeit () UTC angegeben. Das Format unter Verwendung der `strftime()`-Terminologie, nämlich: `[%d/%B/%Y:%H:%M:%S %z]`

Beispieleintrag

```
[06/Feb/2019:00:00:38 +0000]
```

Remote-IP

Die offensichtliche IP-Adresse des Anforderers. Auf dem Weg vorhandene Proxy-Server und Firewalls könnten die tatsächliche IP-Adresse des Computers verbergen, der die Anforderung gestellt hat.

Beispieleintrag

```
192.0.2.3
```

Auftraggeber

Die kanonische Benutzer-ID des Auftraggebers, oder - für nicht authentifizierte Anforderungen. Wenn der Anforderer ein IAM Benutzer war, gibt dieses Feld den Benutzernamen des IAM Anfragenden zusammen mit dem Namen zurück Root-Benutzer des AWS-Kontos , zu dem der IAM Benutzer gehört. Diese ID ist dieselbe, die für den Zugriff zu Kontrollzwecken verwendet wird.

Beispieleintrag

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

Wenn der Anforderer eine angenommene Rolle verwendet, gibt dieses Feld die angenommene Rolle zurück. IAM

Beispieleintrag

```
arn:aws:sts::123456789012:assumed-role/roleName/test-role
```

Anforderungs-ID

Eine von Amazon S3 generierte Zeichenfolge, die jede Anforderung eindeutig identifiziert.

Beispieleintrag

```
3E57427F33A59F07
```

Operation

Die hier aufgeführte Operation ist deklariert al `SOAP.operation`, `REST.HTTP_method.resource_type`, `WEBSITE.HTTP_method.resource_type` oder `BATCH.DELETE.OBJECT`.

Beispieleintrag

```
REST.COPY.OBJECT_GET
```

Schlüssel

Der Schlüssel (Objektname) des kopierten Objekts oder `-`, wenn die Operation keinen Schlüsselparameter entgegennimmt.

Beispieleintrag

```
/photos/2019/08/puppy.jpg
```

Request-URI

Der `Request-URI` Teil der HTTP Anforderungsnachricht.

Beispieleintrag

```
"GET /amzn-s3-demo-bucket1/photos/2019/08/puppy.jpg?x-foo=bar"
```

HTTPStatus

Der numerische HTTP Statuscode des `GET` Teils des Kopiervorgangs.

Beispieleintrag

```
200
```

Fehlercode

Die Amazon S3 S3-Fehlerantworten für den GET Teil des Kopiervorgangs oder - wenn kein Fehler aufgetreten ist.

Beispieleintrag

```
NoSuchBucket
```

Gesendete Bytes

Die Anzahl der gesendeten Antwortbytes, ohne den HTTP Protokoll-Overhead, oder - falls Null.

Beispieleintrag

```
2662992
```

Objektgröße

Die Gesamtgröße des betreffenden Objekts.

Beispieleintrag

```
3462992
```

Gesamtzeit

Die Anzahl der Millisekunden (ms), die die Anforderung aus Perspektive des Servers unterwegs war. Dieser Wert wird ab der Zeit gemessen, zu der Ihre Anforderung empfangen wurde, bis zu der Zeit, zu der das letzte Byte der Antwort gesendet wurde. Messungen aus der Perspektive des Clients dauern möglicherweise länger aufgrund der Netzwerklatenz.

Beispieleintrag

```
70
```

Umschlagzeit

Die Anzahl der Millisekunden, die Amazon S3 gebraucht hat, Ihre Anfrage zu verarbeiten. Dieser Wert wird ab der Zeit gemessen, zu der das letzte Byte Ihrer Anforderung empfangen wurde, bis zu der Zeit, zu der das erste Byte der Antwort gesendet wurde.

Beispieleintrag

```
10
```

Referer

Der Wert des HTTP `Referer` Headers, falls vorhanden. HTTPBenutzeragenten (z. B. Browser) setzen diesen Header normalerweise auf den Header URL der verlinkenden oder einbettenden Seite, wenn sie eine Anfrage stellen.

Beispieleintrag

```
"http://www.example.com/webservices"
```

User-Agent

Der Wert des Headers. HTTP `User-Agent`

Beispieleintrag

```
"curl/7.15.1"
```

Versions-ID

Die Version-ID des kopierten Objekts oder `-`, wenn der `x-amz-copy-source`-Header keinen `versionId`-Parameter als Teil der Kopierquelle angegeben hat.

Beispieleintrag

```
3HL4kqtJvjVBH40Nrjfkd
```

Host-ID

Die erweiterte Anforderungs-ID für `x-amz-id-2` oder Amazon S3

Beispieleintrag

```
s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi31234=
```

Signatur-Version

Die Signaturversion, `SigV2` oder `SigV4`, die für die Authentifizierung der Anforderung verwendet wurde, bzw. ein `-` für nicht authentifizierte Anforderungen.

Beispieleintrag

```
SigV4
```

Cipher Suite

Die Secure Sockets Layer (SSL) -Chiffre, die für eine HTTPS Anfrage ausgehandelt wurde, oder eine `-` für. HTTP

Beispieleintrag

```
ECDHE-RSA-AES128-GCM-SHA256
```

Authentifizierungstyp

Die Art der verwendeten Anforderungsauthentifizierung: `AuthHeader` für Authentifizierungs-Header, `QueryString` für Abfragezeichenfolgen (vorsigniertURLs) oder a `-` für nicht authentifizierte Anfragen.

Beispieleintrag

```
AuthHeader
```

Host-Header

Der für die Verbindung mit Amazon S3 verwendete Endpunkt.

Beispieleintrag

```
s3.us-west-2.amazonaws.com
```

Einige frühere Regionen unterstützen Legacy-Endpunkte. Möglicherweise finden Sie diese Endpunkte in Ihren Serverzugriffsprotokollen oder -protokollen. AWS CloudTrail Weitere Informationen finden Sie unter Legacy-Endpunkte. Eine vollständige Liste der Amazon-S3-

Regionen und -Endpunkte finden Sie unter Endpunkte und Kontingente von Amazon S3 in der Allgemeine Amazon Web Services-Referenz.

TLSVersion

Die vom Client ausgehandelte Version von Transport Layer Security (TLS). Der Wert ist einer der folgenden:`TLSv1.1`,, `TLSv1.2``TLSv1.3`, oder `-` falls TLS er nicht verwendet wurde.

Beispieleintrag

```
TLSv1.2
```

Zugriffspunkt ARN

Der Amazon-Ressourcenname (ARN) des Zugriffspunkts der Anfrage. Wenn der Access Point falsch ARN formatiert ist oder nicht verwendet wird, enthält das Feld eine`-`. Weitere Hinweise zu Zugangspunkten finden Sie unter Verwenden von Zugriffspunkten. Weitere Informationen zu ARNs finden Sie unter Amazon Resource Name (ARN) im AWS Referenzhandbuch.

Beispieleintrag

```
arn:aws:s3:us-east-1:123456789012:accesspoint/example-AP
```

aclRequired

Eine Zeichenfolge, die angibt, ob für die Anfrage eine Zugriffskontrollliste (ACL) für die Autorisierung erforderlich war. Wenn ACL für die Anfrage eine Autorisierung erforderlich war, lautet die ZeichenfolgeYes. Wenn keine erforderlich ACLs waren, lautet die Zeichenfolge`-`. Weitere Informationen zu finden ACLs Sie unterÜbersicht über die Zugriffskontrollliste (ACL). Weitere Informationen zur Verwendung des `aclRequired` Felds zum Deaktivieren ACLs finden Sie unterKontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

Beispieleintrag

```
Yes
```

Benutzerdefinierte Zugriffsprotokollinformationen

Sie können benutzerdefinierte Informationen angeben, die im Zugriffsprotokolldatensatz für eine Anforderung gespeichert werden. Fügen Sie dazu dem URL für die Anforderung einen benutzerdefinierten Abfragezeichenfolge-Parameter hinzu. Amazon S3 ignoriert Abfrage-

Zeichenfolgenparameter, die mit `x-` beginnen, aber nimmt diese in den Zugriffsprotokoll-Datensatz für die Anforderung auf, als Teil des `Request-URI`-Felds des Protokolldatensatzes.

Beispielsweise verhält sich die Anfrage `GET` für `"s3.amazonaws.com/amzn-s3-demo-bucket1/photos/2019/08/puppy.jpg?x-user=johndoe"` genauso wie die Anfrage für `"s3.amazonaws.com/amzn-s3-demo-bucket1/photos/2019/08/puppy.jpg"`, abgesehen davon, dass die Zeichenfolge `"x-user=johndoe"` in das Feld `Request-URI` des entsprechenden Protokolldatensatzes eingefügt wird. Diese Funktionalität ist nur in der REST Benutzeroberfläche verfügbar.

Aspekte zur Programmierung des erweiterbaren Serverzugriff-Protokollformats

Gelegentlich erweitern wir das Zugriffsprotokoll-Datensatzformat, indem wir am Ende jeder Zeile neue Felder hinzufügen. Daher sollten Sie sicherstellen, dass jeder Code, der Server-Zugriffsprotokolle analysiert, so geschrieben ist, dass er angefügte Felder verarbeiten kann, die er möglicherweise nicht versteht.

Löschen von Amazon-S3-Protokolldateien

Ein Amazon-S3-Bucket mit aktivierter Server-Zugriffsprotokollierung kann im Laufe der Zeit viele Server-Protokollobjekte ansammeln. Möglicherweise benötigt Ihre Anwendung diese Zugriffsprotokolle für einen bestimmten Zeitraum nachdem sie erstellt wurden und danach möchten Sie sie ggf. löschen. Sie können die Amazon-S3-Lebenszykluskonfiguration zum Erstellen von Regeln verwenden, sodass Amazon S3 diese Objekte am Ende ihres Lebenszyklus automatisch zum Löschen bereitstellt.

Sie können mithilfe eines gemeinsamen Präfixes für eine Teilmenge der Objekte in Ihrem S3-Bucket eine Lebenszykluskonfiguration definieren. Wenn Sie in Ihrer Konfiguration für die Server-Zugriffsprotokollierung ein Präfix angegeben haben, können Sie eine Lebenszyklus-Konfigurationsregel festlegen, um Protokollobjekte mit diesem Präfix zu löschen.

Angenommen, Ihre Protokollobjekte verwenden das Präfix `logs/`. Sie können dann über eine Lebenszyklus-Konfigurationsregel alle Objekte mit dem Präfix `logs/` im Bucket nach einem bestimmten Zeitraum löschen.

Weitere Informationen zur Lebenszyklus-Konfiguration finden Sie unter Verwaltung des Lebenszyklus von Objekten.

Allgemeine Informationen zu Server-Zugriffsprotokollen finden Sie unter Protokollieren von Anfragen mit Server-Zugriffsprotokollierung.

Verwenden von Amazon-S3-Serverzugriffsprotokollen zur Identifizierung von Anforderungen

Sie können mittels Amazon-S3-Serverzugriffsprotokollen Amazon-S3-Anforderungen identifizieren.

> **Note**
>
> - Um Amazon S3 S3-Anfragen zu identifizieren, empfehlen wir, AWS CloudTrail Datenereignisse anstelle von Amazon S3 S3-Serverzugriffsprotokollen zu verwenden. CloudTrail Datenereignisse sind einfacher einzurichten und enthalten mehr Informationen. Weitere Informationen finden Sie unter Identifizieren von Amazon S3 S3-Anfragen mithilfe von CloudTrail.
> - Je nachdem, wie viele Zugriffsanfragen Sie erhalten, erfordert die Analyse Ihrer Protokolle möglicherweise mehr Ressourcen oder Zeit als die Verwendung von CloudTrail Datenereignissen.

Themen

- Abfragen von Zugriffsprotokollen für Anforderungen über Amazon Athena
- Identifizieren von Signature-Version-2-Anforderungen mittels Amazon-S3-Zugriffsprotokollen
- Identifizieren von Objektzugriffsanforderungen mittels Amazon-S3-Zugriffsprotokollen

Abfragen von Zugriffsprotokollen für Anforderungen über Amazon Athena

Sie können Amazon-S3-Anforderungen mit Amazon-S3-Zugriffsprotokollen mithilfe von Amazon Athena identifizieren.

Amazon S3 speichert Server-Zugriffsprotokolle als Objekte in einem S3-Bucket. Es ist oft einfacher, ein Tool zu verwenden, mit dem die Protokolle in Amazon S3 analysiert werden können. Athena unterstützt die Analyse von S3-Objekten und kann zur Abfrage von Amazon-S3-Zugriffsprotokollen verwendet werden.

Example

Das folgende Beispiel zeigt, wie Sie Amazon-S3-Server-Zugriffsprotokolle in Amazon Athena abfragen können. Ersetzen Sie die *`user input placeholders`* in den folgenden Beispielen durch eigene Daten.

> **Note**
>
> Um einen Amazon S3 S3-Standort in einer Athena-Abfrage anzugeben, müssen Sie einen S3 URI für den Bucket angeben, in den Ihre Logs geliefert werden. Dies URI muss den Bucket-Namen und das Präfix im folgenden Format enthalten: `s3://amzn-s3-demo-bucket1-logs/prefix/`

1. Öffnen Sie die Athena-Konsole unter https://console.aws.amazon.com/athena/.
2. Führen Sie im Abfrage-Editor einen Befehl wie den folgenden aus. Ersetzen Sie *s3_access_logs_db* durch den Namen, den Sie Ihrer Datenbank geben möchten.

```
CREATE DATABASE s3_access_logs_db
```

> **Note**
>
> Es hat sich bewährt, die Datenbank in derselben Datenbank AWS-Region wie Ihren S3-Bucket zu erstellen.

3. Führen Sie im Abfrage-Editor einen Befehl wie den folgenden aus, um in der in Schritt 2 erstellten Datenbank ein Tabellenschema zu erstellen. Ersetzen Sie *s3_access_logs_db.mybucket_logs* durch den Namen, den Sie Ihrer Tabelle geben möchten. Die Datentypwerte `STRING` und `BIGINT` sind die Zugriffsprotokolleigenschaften. Sie können diese Eigenschaften in Athena abfragen. Geben Sie für `LOCATION` wie oben erwähnt den Pfad von S3-Bucket und Präfix ein.

```
CREATE EXTERNAL TABLE `s3_access_logs_db.mybucket_logs`(
  `bucketowner` STRING,
  `bucket_name` STRING,
  `requestdatetime` STRING,
  `remoteip` STRING,
  `requester` STRING,
  `requestid` STRING,
  `operation` STRING,
  `key` STRING,
  `request_uri` STRING,
  `httpstatus` STRING,
  `errorcode` STRING,
  `bytessent` BIGINT,
```

```
  `objectsize` BIGINT,
  `totaltime` STRING,
  `turnaroundtime` STRING,
  `referrer` STRING,
  `useragent` STRING,
  `versionid` STRING,
  `hostid` STRING,
  `sigv` STRING,
  `ciphersuite` STRING,
  `authtype` STRING,
  `endpoint` STRING,
  `tlsversion` STRING,
  `accesspointarn` STRING,
  `aclrequired` STRING)
ROW FORMAT SERDE
  'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  'input.regex'='([^ ]*) ([^ ]*) \\[(.*?)\\] ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 ([^ ]*) (\"[^\"]*\"|-) (-|[0-9]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 (\"[^\"]*\"|-) ([^ ]*)(?: ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 ([^ ]*))?.*$')
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://amzn-s3-demo-bucket1-logs/prefix/'
```

4. Wählen Sie im Navigationsbereich unter Database (Datenbank) die Datenbank aus.
5. Wählen Sie unter Tables (Tabellen)neben dem Namen der Tabelle Preview table (Tabellenvorschau) aus.

   Im Fensterbereich Results (Ergebnisse) sollten Daten aus den Server-Zugriffsprotokollen angezeigt werden, also `bucketowner`, `bucket`, `requestdatetime` usw. Dies bedeutet, dass die Athena-Tabelle erfolgreich erstellt wurde. Sie können jetzt die Amazon-S3-Server-Zugriffsprotokolle abfragen.

Example — Zeigt an, wer ein Objekt wann gelöscht hat (Zeitstempel, IP-Adresse und IAM Benutzer)

```
SELECT requestdatetime, remoteip, requester, key
FROM s3_access_logs_db.mybucket_logs
```

```
WHERE key = 'images/picture.jpg' AND operation like '%DELETE%';
```

Example — Zeigt alle Operationen an, die von einem IAM Benutzer ausgeführt wurden

```
SELECT *
FROM s3_access_logs_db.mybucket_logs
WHERE requester='arn:aws:iam::123456789123:user/user_name';
```

Example – Anzeigen aller Vorgänge, die in einem bestimmten Zeitraum für ein Objekt ausgeführt wurden

```
SELECT *
FROM s3_access_logs_db.mybucket_logs
WHERE Key='prefix/images/picture.jpg'
    AND parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
    BETWEEN parse_datetime('2017-02-18:07:00:00','yyyy-MM-dd:HH:mm:ss')
    AND parse_datetime('2017-02-18:08:00:00','yyyy-MM-dd:HH:mm:ss');
```

Example – Anzeigen der Menge der in einem festgelegten Zeitraum an eine bestimmten IP-Adresse übertragenen Daten

```
SELECT coalesce(SUM(bytessent), 0) AS bytessenttotal
FROM s3_access_logs_db.mybucket_logs
WHERE remoteip='192.0.2.1'
AND parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2022-06-01','yyyy-MM-dd')
AND parse_datetime('2022-07-01','yyyy-MM-dd');
```

> **Note**
>
> Zur Reduzierung der Beibehaltungsdauer Ihrer Protokolle können Sie für Ihren Serverzugriffsprotokoll-Bucket eine S3-Lebenszykluskonfiguration erstellen. Erstellen Sie Lebenszykluskonfigurationsregeln, um Protokolldateien regelmäßig zu entfernen.

Dadurch wird die Menge der Daten reduziert, die Athena in einer Abfrage analysiert. Weitere Informationen finden Sie unter Einstellung einer S3-Lifecycle-Konfiguration für einen Bucket.

Identifizieren von Signature-Version-2-Anforderungen mittels Amazon-S3-Zugriffsprotokollen

Die Amazon-S3-Unterstützung für Signature Version 2 wird deaktiviert (veraltet). Danach akzeptiert Amazon S3 keine Anforderungen mit Signature Version 2 mehr, alle Anforderungen müssen also mit Signature Version 4 signiert werden. Sie können Signature-Version-2-Zugriffsanforderungen mittels Amazon-S3-Zugriffsprotokollen identifizieren.

Note

Um Signature Version 2-Anfragen zu identifizieren, empfehlen wir, AWS CloudTrail Datenereignisse anstelle von Amazon S3 S3-Serverzugriffsprotokollen zu verwenden. CloudTrail Datenereignisse sind einfacher einzurichten und enthalten mehr Informationen als Serverzugriffsprotokolle. Weitere Informationen finden Sie unter Identifizieren von Amazon S3 Signature Version 2-Anfragen mithilfe von CloudTrail.

Example – Anzeigen aller Anforderer, die Signature Version 2-Datenverkehr senden

```
SELECT requester, sigv, Count(sigv) as sigcount
FROM s3_access_logs_db.mybucket_logs
GROUP BY requester, sigv;
```

Identifizieren von Objektzugriffsanforderungen mittels Amazon-S3-Zugriffsprotokollen

Sie können Abfragen an Amazon-S3-Server-Zugriffsprotokolle verwenden, um Amazon-S3-Objektzugriffsanforderungen für Vorgänge zu identifizieren, wie etwa `GET`, `PUT` und `DELETE`, und weitere Informationen über diese Anforderungen zu entdecken.

Das folgende Amazon-Athena-Abfragebeispiel zeigt, wie Sie alle `PUT`-Objektanforderungen für Amazon S3 aus einem Serverzugriffsprotokoll abrufen können.

Example – Anzeigen aller Anforderer, die **PUT**-Objektanforderungen in einem bestimmten Zeitraum senden

```
SELECT bucket_name, requester, remoteip, key, httpstatus, errorcode, requestdatetime
FROM s3_access_logs_db
WHERE operation='REST.PUT.OBJECT' AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

Das folgende Amazon Athena-Abfragebeispiel zeigt, wie alle GET-Objektanfragen für Amazon S3 aus dem Server-Zugriffsprotokoll abgerufen werden.

Example – Anzeigen aller Anforderer, die **GET**-Objektanforderungen in einem bestimmten Zeitraum senden

```
SELECT bucket_name, requester, remoteip, key, httpstatus, errorcode, requestdatetime
FROM s3_access_logs_db
WHERE operation='REST.GET.OBJECT' AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

Das folgende Amazon Athena-Abfragebeispiel zeigt, wie alle anonymen Anforderungen aus dem Server-Zugriffsprotokoll in Ihre S3-Buckets gelangen.

Example – Anzeigen aller anonymen Anforderer, die in einem bestimmten Zeitraum Anforderungen an einen Bucket richten

```
SELECT bucket_name, requester, remoteip, key, httpstatus, errorcode, requestdatetime
FROM s3_access_logs_db.mybucket_logs
WHERE requester IS NULL AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

Die folgende Amazon Athena Athena-Abfrage zeigt, wie Sie alle Anfragen an Ihre S3-Buckets identifizieren können, für die eine Zugriffskontrollliste (ACL) für die Autorisierung erforderlich war. Sie können diese Informationen verwenden, um diese ACL Berechtigungen in die entsprechenden Bucket-Richtlinien zu migrieren und zu deaktivieren. ACLs Nachdem Sie diese Bucket-Richtlinien erstellt haben, können Sie sie ACLs für diese Buckets deaktivieren. Weitere Informationen zur Deaktivierung ACLs finden Sie unter. Voraussetzungen für die Deaktivierung ACLs

Example — Identifizieren Sie alle Anfragen, ACL für die eine Autorisierung erforderlich war

```
SELECT bucket_name, requester, key, operation, aclrequired, requestdatetime
FROM s3_access_logs_db
WHERE aclrequired = 'Yes' AND
parse_datetime(requestdatetime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2022-05-10:00:00:00','yyyy-MM-dd:HH:mm:ss')
AND parse_datetime('2022-08-10:00:00:00','yyyy-MM-dd:HH:mm:ss')
```

Note

- Sie können den Datumsbereich nach Belieben an Ihre Anforderungen anpassen.
- Diese Abfragebeispiele können auch für die Sicherheitsüberwachung nützlich sein. Sie können die Ergebnisse auf `PutObject`- oder `GetObject`-Aufrufe von unerwarteten oder nicht autorisierten IP-Adressen oder Anforderern und zum Identifizieren anonymer Anforderungen an Ihre Buckets prüfen.
- Diese Abfrage ruft nur Informationen von der Zeit ab, zu der die Protokollierung aktiviert wurde.
- Wenn Sie AWS CloudTrail Protokolle verwenden, finden Sie weitere Informationen unterIdentifizieren des Zugriffs auf S3-Objekte mithilfe von CloudTrail.

Behebung von Fehlern bei der Server-Zugriffsprotokollierung

Die folgenden Themen können Ihnen bei der Behebung von Problemen helfen, die beim Einrichten der Protokollierung mit Amazon S3 auftreten können.

Themen

- Häufige Fehlermeldungen beim Einrichten der Protokollierung

- Behebung von Bereitstellungsfehlern

Häufige Fehlermeldungen beim Einrichten der Protokollierung

Die folgenden häufigen Fehlermeldungen können auftreten, wenn Sie die Protokollierung über AWS Command Line Interface (AWS CLI) und aktivieren AWS SDKs:

Fehler: S3-standortübergreifende Protokollierung nicht zulässig

Wenn sich der Ziel-Bucket in einer anderen Region als der Quell-Bucket befindet, tritt der Fehler S3-standortübergreifende Protokollierung nicht zulässig auf. Um diesen Fehler zu beheben, stellen Sie sicher, dass sich der Ziel-Bucket, der für den Empfang der Zugriffs-Logs konfiguriert ist, im selben AWS-Region und AWS-Konto wie der Quell-Bucket befindet.

Fehler: Die Eigentümer des zu protokollierenden Buckets und des Ziel-Buckets müssen identisch sein

Wenn Sie die Server-Zugriffsprotokollierung aktivieren, tritt dieser Fehler auf, wenn der angegebene Ziel-Bucket zu einem anderen Konto gehört. Um diesen Fehler zu beheben, stellen Sie sicher, dass sich der Ziel-Bucket im selben AWS-Konto wie der Quell-Bucket befindet.

> **Note**
>
> Wir empfehlen, einen anderen Ziel-Bucket als den Quell-Bucket auszuwählen. Wenn Quell-Bucket und Ziel-Bucket identisch sind, werden für die Protokolle, die in den Bucket geschrieben werden, zusätzliche Protokolle erstellt, wodurch die Speicherkosten steigen können. Diese zusätzlichen Protokolle zu den Protokollen können es zudem schwierig machen, die gesuchten Protokolle zu finden. Zur einfacheren Protokollverwaltung empfehlen wir, Zugriffsprotokolle in einem anderen Bucket zu speichern. Weitere Informationen finden Sie unter the section called “Wie aktiviere ich die Protokollzustellung?”.

Fehler: Der Ziel-Bucket für die Protokollierung ist nicht vorhanden

Der Ziel-Bucket muss vorhanden sein, bevor die Konfiguration festgelegt wird. Dieser Fehler weist darauf hin, dass der Ziel-Bucket nicht vorhanden ist oder nicht gefunden werden kann. Vergewissern Sie sich, dass der Bucket-Name richtig geschrieben ist, und wiederholen Sie den Vorgang.

Fehler: Ziel-Erteilungen sind für Buckets mit „Bucket-Eigentümer erzwungen“ nicht zulässig

Dieser Fehler weist darauf hin, dass der Ziel-Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für die S3-Objekt-Eigentümerschaft verwendet. Die Einstellung „Vom Bucket-Eigentümer erzwungen“ unterstützt keine Ziel-Erteilungen. Weitere Informationen finden Sie unter Berechtigungen für Protokollbereitstellung.

Behebung von Bereitstellungsfehlern

Halten Sie sich an die folgenden bewährten Methoden, um Probleme mit der Server-Zugriffsprotokollierung zu vermeiden:

- Die S3-Protokollbereitstellungsgruppe hat Schreibzugriff auf den Ziel-Bucket – Die S3-Protokollbereitstellungsgruppe liefert Serverzugriffsprotokolle an den Ziel-Bucket. Eine Bucket-Richtlinie oder eine Bucket-Zugriffskontrollliste (ACL) kann verwendet werden, um Schreibzugriff auf den Ziel-Bucket zu gewähren. Wir empfehlen jedoch, eine Bucket-Richtlinie anstelle einer zu verwendenACL. Weitere Informationen zum Gewähren von Schreibzugriff für Ihren Ziel-Bucket finden Sie unter Berechtigungen für Protokollbereitstellung.

  > **Note**
  >
  > Wenn der Ziel-Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für die Objekt-Eigentümerschaft verwendet, beachten Sie Folgendes:
  >
  > - ACLssind deaktiviert und wirken sich nicht mehr auf die Berechtigungen aus. Das bedeutet, dass Sie Ihren Bucket nicht aktualisieren könnenACL, um Zugriff auf die S3-Protokollbereitstellungsgruppe zu gewähren. Sie müssen vielmehr die Bucket-Richtlinie für den Ziel-Bucket aktualisieren, um Zugriff auf den Prinzipal des Protokollierungsservices zu gewähren.
  > - Sie können keine Ziel-Erteilungen in Ihre `PutBucketLogging`-Konfiguration einschließen.

- Die Bucket-Richtlinie für den Ziel-Bucket lässt den Zugriff auf die Protokolle zu – Überprüfen Sie die Bucket-Richtlinie des Ziel-Buckets. Suchen Sie in der Bucket-Richtlinie nach allen Anweisungen, die `"Effect": "Deny"` enthalten. Stellen Sie anschließend sicher, dass die `Deny`-Anweisung das Schreiben von Zugriffsprotokollen in den Bucket nicht verhindert.
- S3 Object Lock ist auf dem Ziel-Bucket nicht aktiviert – Prüfen Sie, ob Object Lock für den Ziel-Bucket aktiviert ist. Die Objektsperre blockiert die Bereitstellung von Server-Zugriffsprotokollen. Sie müssen einen Ziel-Bucket auswählen, für den Object Lock nicht aktiviert ist.

- Verwaltete Amazon S3 S3-Schlüssel (SSE-S3) ist ausgewählt, wenn die Standardverschlüsselung im Ziel-Bucket aktiviert ist — Sie können die Standard-Bucket-Verschlüsselung für den Ziel-Bucket nur verwenden, wenn Sie die serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) verwenden. Die serverseitige Standardverschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-KMS) wird für Ziel-Buckets zur Serverzugriffsprotokollierung nicht unterstützt. Weitere Informationen zum Aktivieren der Standardverschlüsselung finden Sie unter Konfigurieren der Standardverschlüsselung.
- Für den Ziel-Bucket ist „Zahlung durch den Anforderer“ nicht aktiviert – Die Verwendung eines Buckets mit „Zahlung durch den Anforderer“ als Ziel-Bucket für die Server-Zugriffsprotokollierung wird nicht unterstützt. Um die Bereitstellung von Serverzugriffsprotokollen zu ermöglichen, deaktivieren Sie die Option „Zahlung durch den Anforderer“ im Ziel-Bucket.
- Überprüfen Sie Ihre AWS Organizations Service Control Policies (SCPs) und Resource Control Policies (RCPs) — Überprüfen Sie bei der Nutzung die Service Control Policies und Resource Control Policies AWS Organizations, um sicherzustellen, dass der Amazon S3 S3-Zugriff erlaubt ist. Diese Richtlinien legen die maximalen Berechtigungen für Prinzipale und Ressourcen in den betroffenen Konten fest. Suchen Sie in den Richtlinien nach Anweisungen, die Aussagen enthalten, `"Effect": "Deny"` und stellen Sie sicher, dass diese `Deny` Anweisungen nicht verhindern, dass Zugriffsprotokolle in den Bucket geschrieben werden. Weitere Informationen finden Sie AWS Organizations im AWS Organizations Benutzerhandbuch unter Autorisierungsrichtlinien.
- Eine gewisse Zeit warten, bis die letzten Änderungen an der Protokollierungskonfiguration wirksam werden – Die erstmalige Aktivierung der Server-Zugriffsprotokollierung oder die Änderung des Ziel-Buckets für Protokolle wird erst nach einer gewissen Zeit vollständig wirksam. Es kann über eine Stunde dauern, bis alle Anforderungen ordnungsgemäß protokolliert und übermittelt wurden.

  Um nach Protokollzustellungsfehlern zu suchen, aktivieren Sie Anforderungsmetriken in Amazon CloudWatch. Wenn die Protokolle nicht innerhalb weniger Stunden bereitgestellt werden, suchen Sie nach der Metrik `4xxErrors`, die auf Protokollbereitstellungsfehler hinweisen kann. Weitere Informationen zur Aktivierung von Anforderungsmetriken finden Sie unter the section called “Erstellen einer Metrik-Konfiguration für alle Objekte”.

Metriken mit Amazon überwachen CloudWatch

CloudWatch Amazon-Metriken für Amazon S3 können Ihnen helfen, die Leistung von Anwendungen, die Amazon S3 verwenden, zu verstehen und zu verbessern. Es gibt mehrere Möglichkeiten, die Sie CloudWatch mit Amazon S3 verwenden können.

Tägliche Speichermetriken für Buckets

Überwachen Sie den Bucket-Speicher mithilfe CloudWatch, der Speicherdaten von Amazon S3 sammelt und zu lesbaren, täglichen Messwerten verarbeitet. Diese Speichermetriken für Amazon S3 werden einmal pro Tag gemeldet und allen Kunden ohne zusätzliche Kosten zur Verfügung gestellt.

Anforderungsmetriken

Überwachen von Amazon-S3-Anforderungen, um Probleme bei der Ausführung schnell zu identifizieren und zu beheben. Die Metriken stehen in 1-Minuten-Intervallen nach einer gewissen Latenz für die Verarbeitung zur Verfügung. Diese CloudWatch Metriken werden zum gleichen Tarif wie die CloudWatch benutzerdefinierten Amazon-Metriken in Rechnung gestellt. Informationen zu den CloudWatch Preisen finden Sie unter CloudWatch Amazon-Preise. Weitere Informationen zum optionalen Erhalt dieser Metriken finden Sie unter CloudWatch Metriken, Konfigurationen.

Wenn die Anforderungsmetriken aktiviert sind, werden sie für alle Objekt-Vorgänge gemeldet. Standardmäßig stehen diese 1-Minuten-Metriken auf Ebene der Amazon-S3-Buckets zur Verfügung. Sie können auch einen Filter für die Metriken definieren, die mit einem gemeinsamen Präfix oder Objekt-Tag oder Zugriffspunkt erfasst wurden.

- Zugriffspunkt – Zugriffspunkte sind benannte Netzwerkendpunkte, die Buckets zugeordnet sind und vereinfachen die skalierbare Verwaltung des Datenzugriffs auf freigegebene Datensätze in S3. Mit dem Zugriffspunkt-Filter erhalten Sie Erkenntnisse in Ihre Zugriffspunkt-Nutzung. Weitere Hinweise zu Zugangspunkten finden Sie unter Überwachen und Protokollieren von Zugriffspunkten.
- Präfix – Obwohl das Amazon-S3-Datenmodell eine flache Struktur aufweist, können Sie die Hierarchie durch die Verwendung eines Präfixes inferieren. Ein Präfix ähnelt einem Verzeichnisnamen, mit dem Sie ähnliche Objekte in einem Bucket gruppieren können. Die S3-Konsole unterstützt Präfixe mit dem Ordnerkonzept. Wenn Sie nach dem Präfix filtern, werden Objekte mit demselben Präfix in die Metrik-Konfiguration aufgenommen. Weitere Informationen zu Präfixen finden Sie unter Organisieren von Objekten mit Präfixen.
- Tag – Tags sind Schlüssel-Wert-Namenspaare, die Sie Objekten hinzufügen können. Mit Markierungen können Sie Objekte einfacher finden und organisieren. Tags können auch als Filter für Metrikkonfigurationen verwendet werden, sodass nur Objekte mit diesen Tags in die Metrikkonfiguration aufgenommen werden. Weitere Informationen über Objekt-Markierungen finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen.

Um diese Metriken an bestimmten Geschäftsanwendungen, Workflows oder internen Organisationen auszurichten, können Sie nach einem freigegebenen Präfix, einem Objekt-Tag oder einem Zugriffspunkt filtern.

Replikationsmetriken

Überwachen Sie die Gesamtzahl der API S3-Operationen, deren Replikation aussteht, die Gesamtgröße der Objekte, deren Replikation noch aussteht, die maximale Replizierungszeit bis zum Ziel und die Gesamtzahl der Operationen AWS-Region, bei denen die Replikation fehlgeschlagen ist. Replikationsregeln, für die S3 Replication Time Control (S3RTC) oder S3 Replication-Metriken aktiviert sind, veröffentlichen Replikationsmetriken.

Weitere Informationen finden Sie unter Überwachung der Replikation mit Metriken, Ereignisbenachrichtigungen und Status oder Erfüllung von Compliance-Anforderungen mit S3 Replication Time Control.

Metriken von Amazon S3 Storage Lens

Sie können die Nutzungs- und Aktivitätsmetriken von S3 Storage Lens CloudWatch auf Amazon veröffentlichen, um eine einheitliche Ansicht Ihres Betriebszustands in CloudWatch Dashboards zu erstellen. S3-Storage-Lens-Metriken sind im `AWS/S3/Storage-Lens`-Namespace verfügbar. Die CloudWatch Veröffentlichungsoption ist für S3 Storage Lens-Dashboards verfügbar, die auf erweiterte Metriken und Empfehlungen aktualisiert wurden. Sie können die CloudWatch Veröffentlichungsoption für eine neue oder bestehende Dashboard-Konfiguration in S3 Storage Lens aktivieren.

Weitere Informationen finden Sie unter Überwachen Sie S3 Storage Lens-Metriken in CloudWatch.

Alle CloudWatch Statistiken werden für einen Zeitraum von 15 Monaten aufbewahrt, sodass Sie auf historische Informationen zugreifen und sich einen besseren Überblick über die Leistung Ihrer Webanwendung oder Ihres Dienstes verschaffen können. Weitere Informationen zu CloudWatch finden Sie unter Was ist Amazon CloudWatch? im CloudWatch Amazon-Benutzerhandbuch. Abhängig von Ihren Anwendungsfällen benötigen Sie möglicherweise einige zusätzliche Konfigurationen für Ihre CloudWatch Alarme. Sie können beispielsweise einen mathematischen Metrik-Ausdruck verwenden, um einen Alarm zu erzeugen. Weitere Informationen finden Sie unter Verwenden von CloudWatch Metriken, Verwenden von metrischen Berechnungen, Verwenden von CloudWatch Amazon-Alarmen und Erstellen eines CloudWatch Alarms auf der Grundlage eines metrischen mathematischen Ausdrucks im CloudWatch Amazon-Benutzerhandbuch.

Bereitstellung von CloudWatch Kennzahlen nach bestem Wissen

CloudWatch Die Metriken werden nach bestem Wissen und Gewissen bereitgestellt. Die meisten Anfragen für ein Amazon S3 S3-Objekt mit Anforderungsmetriken führen dazu, dass ein Datenpunkt an gesendet wird CloudWatch.

Die Vollständigkeit und Rechtzeitigkeit der Metriken ist nicht garantiert. Der Datenpunkt für eine bestimmte Anforderung wird möglicherweise mit einem Zeitstempel zurückgegeben, der nach der tatsächlichen Anforderungsverarbeitung liegt. Der Datenpunkt kann sich um eine Minute verzögern, bevor er verfügbar ist CloudWatch, oder er wird möglicherweise gar nicht zugestellt. CloudWatch Anhand von Anforderungsmetriken erhalten Sie nahezu in Echtzeit eine Vorstellung von der Art des Datenverkehrs in Ihrem Bucket. Sie sind nicht als vollständige Abrechnung aller Anforderungen vorgesehen.

Aufgrund der Best-Effort-Natur dieser Funktion enthalten die im Fakturierungs- und Kostenverwaltungs-Dashboard verfügbaren Berichte möglicherweise eine oder mehrere Zugriffsanforderungen, die nicht in den Bucket-Metriken angezeigt werden.

Weitere Informationen finden Sie unter den folgenden Themen.

Themen

- Metriken und Dimensionen
- Zugriff auf Metriken CloudWatch
- CloudWatch Metriken, Konfigurationen

Metriken und Dimensionen

Die Speichermetriken und Dimensionen, die Amazon S3 an Amazon sendet, CloudWatch sind in den folgenden Tabellen aufgeführt.

Bereitstellung von CloudWatch Kennzahlen nach bestem Wissen

CloudWatch Die Metriken werden nach bestem Wissen und Gewissen bereitgestellt. Die meisten Anfragen für ein Amazon S3 S3-Objekt mit Anforderungsmetriken führen dazu, dass ein Datenpunkt an gesendet wird CloudWatch.

Die Vollständigkeit und Rechtzeitigkeit der Metriken ist nicht garantiert. Der Datenpunkt für eine bestimmte Anforderung wird möglicherweise mit einem Zeitstempel zurückgegeben, der nach der

tatsächlichen Anforderungsverarbeitung liegt. Der Datenpunkt kann sich um eine Minute verzögern, bevor er verfügbar ist CloudWatch, oder er wird möglicherweise gar nicht zugestellt. CloudWatch Anhand von Anforderungsmetriken erhalten Sie nahezu in Echtzeit eine Vorstellung von der Art des Datenverkehrs in Ihrem Bucket. Sie sind nicht als vollständige Abrechnung aller Anforderungen vorgesehen.

Aufgrund der Best-Effort-Natur dieser Funktion enthalten die im Fakturierungs- und Kostenverwaltungs-Dashboard verfügbaren Berichte möglicherweise eine oder mehrere Zugriffsanforderungen, die nicht in den Bucket-Metriken angezeigt werden.

Themen

- Tägliche Amazon S3 S3-Speichermetriken für Buckets in CloudWatch
- Amazon S3 S3-Anforderungsmetriken in CloudWatch
- Metriken zur S3-Replikation in CloudWatch
- S3 Storage Lens-Metriken in CloudWatch
- Lambda-Anforderungsmetriken für S3-Objekte in CloudWatch
- Amazon S3 S3-Abmessungen in CloudWatch
- Dimensionen der S3-Replikation in CloudWatch
- Abmessungen der S3-Speicherlinse in CloudWatch
- Abmessungen der Lambda-Anforderung von S3 Object in CloudWatch
- Amazon S3 S3-Nutzungsmetriken

Tägliche Amazon S3 S3-Speichermetriken für Buckets in CloudWatch

Der `AWS/S3`-Namespace enthält die folgenden täglichen Speichermetriken für Buckets.

Metrik	Beschreibung
`BucketSizeBytes`	Die Datenmenge in Byte, die in einem Bucket in den folgenden Speicherklassen gespeichert wird: • S3 Standard (`STANDARD`) • S3 Intelligent-Tiering (`INTELLIGENT_TIERING`) • S3 Standard-Infrequent Access (`STANDARD_IA`) • S3 One Zone — Seltener Zugriff () `ONEZONE_IA`

Metrik	Beschreibung
	• Speicher mit reduzierter Redundanz () () RRS `REDUCED_REDUNDANCY` • S3 Glacier Instant Retrieval (`GLACIER_IR`) • S3 Glacier Deep Archive (`DEEP_ARCHIVE`) • S3 Glacier Flexible Retrieval (`GLACIER`) • S3 Express One Zone (`EXPRESS_ONEZONE`) Dieser Wert wird berechnet, indem die Größe aller Objekte und Metadaten (wie Bucket-Namen) im Bucket (sowohl aktuelle als auch nicht aktuelle Objekte) summiert wird, einschließlich der Größe aller Teile für alle unvollständigen mehrteiligen Uploads in den Bucket. **Note** Die S3 Express One Zone-Speicherklasse ist nur für Verzeichnis-Buckets verfügbar. Gültige Speichertypfilter (siehe Dimension): `StorageType` • S3-Standard: `StandardStorage` • Intelligentes S3-Tiering:`IntelligentTieringFAStorage` ,,, `IntelligentTieringIAStorage` `IntelligentTieringAAStorage` `IntelligentTieringAIAStorage` `IntelligentTieringDAAStorage` • S3 Standard — Seltener Zugriff:,, `StandardIAStorage` `StandardIASizeOverhead` `StandardIAObjectOverhead` • S3 One Zone — Seltener Zugriff:, `OneZoneIAStorage` `OneZoneIASizeOverhead` • Speicher mit reduzierter Redundanz (): RRS `ReducedRedundancyStorage` • S3 Glacier Sofortiger Abruf:, `GlacierIRSizeOverhead` `GlacierInstantRetrievalStorage`

Metrik	Beschreibung
	• Flexibler S3 Glacier-Abruf:`GlacierStorage` ,,, `GlacierStagingStorage` `GlacierObjectOverhead` `GlacierS3ObjectOverhead` • S3 Glacier Deep Archive:`DeepArchiveStorage` ,`DeepArchiveObjectOverhead` ,`DeepArchiveS3ObjectOverhead` , `DeepArchiveStagingStorage` • S3 Express Eine Zone: `ExpressOneZoneStorage` Einheiten: Byte Gültige Statistiken: Durchschnitt Weitere Informationen zu den `StorageType` Abmessungen finden Sie unterthe section called “Amazon S3 S3-Abmessungen in CloudWatch”.
`NumberOfObjects`	Die Gesamtzahl der Objekte für alle Speicherklassen, die in einem Bucket für allgemeine Zwecke gespeichert sind. Zur Berechnung dieses Werts werden alle aktuellen und nicht aktuellen Objekte im Bucket, alle Löschmarkierungen sowie die Gesamtanzahl der Teile sämtliche r unvollständiger mehrteiliger Uploads in den Bucket gezählt. Bei Verzeichnis-Buckets mit Objekten in der S3 Express One Zone-Spei cherklasse wird dieser Wert berechnet, indem alle Objekte im Bucket gezählt werden. Unvollständige Mehrfach-Uploads in den Bucket werden jedoch nicht berücksichtigt. Gültige Speichertypfilter: nur `AllStorageTypes` (siehe `StorageType` -Dimension) Einheiten: Anzahl Gültige Statistiken: Durchschnitt

Amazon S3 S3-Anforderungsmetriken in CloudWatch

Der `AWS/S3`-Namespace enthält die folgenden Anforderungsmetriken. Zu diesen Metriken gehören Anfragen, die nicht abgerechnet werden können (im Fall von `GET` Anfragen von `CopyObject` und Replikation).

> **Note**
>
> Amazon S3 S3-Anforderungsmetriken in werden für Directory-Buckets CloudWatch nicht unterstützt.

Metrik	Beschreibung
`AllRequests`	Die Gesamtzahl der HTTP Anfragen an einen Amazon S3 S3-Bucket , unabhängig vom Typ. Wenn Sie eine Metrikkonfiguration mit einem Filter verwenden, gibt diese Metrik nur die HTTP Anfragen zurück, die die Anforderungen des Filters erfüllen. Einheiten: Anzahl Gültige Statistiken: Summe
`GetRequests`	Die Anzahl der HTTP `GET` Anfragen für Objekte in einem Amazon S3 S3-Bucket. Das Auflisten von Vorgängen ist hierin nicht inbegriffen. Diese Metrik wird für die Quelle jeder `CopyObject` Anfrage inkrementiert. Einheiten: Anzahl Gültige Statistiken: Summe **Note** Paginierte, listenorientierte Anfragen, wie ListMultipartUploads, ListParts, ListObjectVersionsund andere sind in dieser Metrik nicht enthalten.

Metrik	Beschreibung
`PutRequests`	Die Anzahl der HTTP `PUT` Anfragen für Objekte in einem Amazon S3 S3-Bucket. Diese Metrik wird für das Ziel jeder `CopyObject` Anfrage inkrementiert. Einheiten: Anzahl Gültige Statistiken: Summe
`DeleteRequests`	Die Anzahl der HTTP `DELETE` Anfragen für Objekte in einem Amazon S3 S3-Bucket. Diese Metrik umfasst auch DeleteObjectsAnfragen. Diese Metrik zeigt die Anzahl der gestellten Anfragen, nicht die Anzahl der gelöschten Objekte an. Einheiten: Anzahl Gültige Statistiken: Summe
`HeadRequests`	Die Anzahl der HTTP `HEAD` Anfragen an einen Amazon S3 S3-Bucket. Einheiten: Anzahl Gültige Statistiken: Summe
`PostRequests`	Die Anzahl der HTTP `POST` Anfragen an einen Amazon S3 S3-Bucket. Einheiten: Anzahl Gültige Statistiken: Summe **Note** DeleteObjects und SelectObjectContentAnfragen sind in dieser Metrik nicht enthalten.

Metrik	Beschreibung
`SelectRequests`	Die Anzahl der Amazon S3 SelectObjectContentAnfragen für Objekte in einem Amazon S3 S3-Bucket. Einheiten: Anzahl Gültige Statistiken: Summe
`SelectBytesScanned`	Die Anzahl der mit Amazon S3 gescannten Datenbytes SelectObjectContentAnfragen in einem Amazon S3 S3-Bucket. Einheiten: Byte Gültige Statistiken: Durchschnitt (Byte pro Anforderung), Summe (Byte pro Zeitraum), Stichprobenanzahl, Min, Max (entspricht p100), beliebiges Perzentil zwischen p0,0 und p99,9
`SelectBytesReturned`	Die Anzahl der mit Amazon S3 zurückgegebenen Datenbytes SelectObjectContentAnfragen in einem Amazon S3 S3-Bucket. Einheiten: Byte Gültige Statistiken: Durchschnitt (Byte pro Anforderung), Summe (Byte pro Zeitraum), Stichprobenanzahl, Min, Max (entspricht p100), beliebiges Perzentil zwischen p0,0 und p99,9
`ListRequests`	Die Anzahl der HTTP Anfragen, die den Inhalt eines Buckets auflisten. Einheiten: Anzahl Gültige Statistiken: Summe
`BytesDownloaded`	Anzahl der heruntergeladenen Bytes für Anfragen an einen Amazon-S3-Bucket, wobei die Antwort einen Textkörper enthält. Einheiten: Byte Gültige Statistiken: Durchschnitt (Byte pro Anforderung), Summe (Byte pro Zeitraum), Stichprobenanzahl, Min, Max (entspricht p100), beliebiges Perzentil zwischen p0,0 und p99,9

Metrik	Beschreibung
`BytesUploaded`	Anzahl der hochgeladenen Bytes für Anfragen an einen Amazon-S3-Bucket, wobei die Antwort einen Textkörper enthält. Einheiten: Byte Gültige Statistiken: Durchschnitt (Byte pro Anforderung), Summe (Byte pro Zeitraum), Stichprobenanzahl, Min, Max (entspricht p100), beliebiges Perzentil zwischen p0,0 und p99,9
`4xxErrors`	Die Anzahl der HTTP 4 x x Client-Fehlerstatuscode-Anfragen, die an einen Amazon S3 S3-Bucket mit einem Wert von entweder 0 oder 1 gestellt wurden. Die Durchschnittsstatistik zeigt die Fehlerrate und die Summenstatistik die Anzahl dieses Fehlertyps für den jeweiligen Zeitraum an. Einheiten: Anzahl Gültige Statistiken: Durchschnitt (Berichte pro Anforderung), Summe (Berichte pro Zeitraum), Min, Max, Stichprobenanzahl
`5xxErrors`	Die Anzahl von HTTP 5 xx-Serverfehlerstatuscode-Anfragen, die an einen Amazon S3 S3-Bucket mit einem Wert von entweder 0 oder 1 gestellt wurden. Die Durchschnittsstatistik zeigt die Fehlerrate und die Summenstatistik die Anzahl dieses Fehlertyps für den jeweiligen Zeitraum an. Einheiten: Anzahl Gültige Statistiken: Durchschnitt (Berichte pro Anforderung), Summe (Berichte pro Zeitraum), Min, Max, Stichprobenanzahl

Metrik	Beschreibung
`FirstByteLatency`	Zeit pro Anforderung vom Zeitpunkt des Eingangs der vollständigen Anfragen bei einem Amazon-S3-Bucket bis das Zurückgeben der Antwort beginnt. Einheiten: Millisekunden Gültige Statistiken: Durchschnitt, Summe, Min, Max (entspricht p100), Stichprobenanzahl, beliebiges Perzentil zwischen p0,0 und p100
`TotalRequestLatency`	Zeit pro Anforderung vom Zeitpunkt des Eingangs des ersten Bytes bei einem Amazon-S3-Bucket bis zum Senden des letzten Bytes. Diese Metrik umfasst die erforderliche Zeit, um den Textkörper zu erhalten und den Antworttextkörper zu senden, welche nicht in `FirstByteLatency` enthalten ist. Einheiten: Millisekunden Gültige Statistiken: Durchschnitt, Summe, Min, Max (entspricht p100), Stichprobenanzahl, beliebiges Perzentil zwischen p0,0 und p100

Metriken zur S3-Replikation in CloudWatch

Sie können den Fortschritt der Replikation mit S3-Replikationsmetriken überwachen, indem Sie ausstehende Bytes, ausstehende Operationen und die Replikationslatenz nachverfolgen. Weitere Informationen finden Sie unter Überwachen des Fortschritts mit Replikationsmetriken.

> **Note**
>
> Sie können Alarme für Ihre Replikationsmetriken in Amazon aktivieren CloudWatch. Wenn Sie Alarme für Ihre Replikationsmetriken einrichten, stellen Sie das Feld Missing data treatment (Behandlung fehlender Daten) auf Treat missing data as ignore (maintain the alarm state) (Fehlende Daten ignorieren) (Alarmstatus beibehalten) ein.

Metrik	Beschreibung
`ReplicationLatency`	Die maximale Anzahl von Sekunden, um die das Replikationsziel hinter der Quelle AWS-Region für eine bestimmte Replikationsregel AWS-Region liegt. Einheiten: Sekunden Gültige Statistiken: Max
`BytesPendingReplication`	Die Gesamtanzahl der Bytes von Objekten, die für eine bestimmte Replikationsregel ausstehen. Einheiten: Byte Gültige Statistiken: Max
`OperationsPendingReplication`	Die Anzahl der Vorgänge mit ausstehender Replikation für eine bestimmte Replikationsregel. Einheiten: Anzahl Gültige Statistiken: Max
`OperationsFailedReplication`	Die Anzahl der Operationen, deren Replikation für eine bestimmte Replikationsregel fehlgeschlagen ist. Einheiten: Anzahl Gültige Statistiken: Summe (Gesamtzahl der fehlgeschlagenen Operationen), Durchschnitt (Fehlerrate), Stichprobenanzahl (Gesamtzahl der Replikationsoperationen)

S3 Storage Lens-Metriken in CloudWatch

Sie können die Nutzungs- und Aktivitätsmetriken von S3 Storage Lens CloudWatch auf Amazon veröffentlichen, um eine einheitliche Ansicht Ihres Betriebszustands in CloudWatchDashboards zu erstellen. S3 Storage Lens-Metriken werden im `AWS/S3/Storage-Lens` Namespace in

veröffentlicht. CloudWatch Die CloudWatch Veröffentlichungsoption ist für S3 Storage Lens-Dashboards verfügbar, die auf erweiterte Metriken und Empfehlungen aktualisiert wurden.

Eine Liste der S3 Storage Lens-Metriken, die veröffentlicht wurden CloudWatch, finden Sie unterAmazon S3-Storage-Lens-Metrikglossar. Eine vollständige Liste der Dimensionen finden Sie unter Dimensionen.

Lambda-Anforderungsmetriken für S3-Objekte in CloudWatch

S3 Objekt Lambda enthält die folgenden Anforderungsmetriken.

Metrik	Beschreibung
`AllRequests`	Die Gesamtzahl der HTTP Anfragen, die mithilfe eines Object Lambda Access Points an einen Amazon S3 S3-Bucket gestellt wurden. Einheiten: Anzahl Gültige Statistiken: Summe
`GetRequests`	Die Anzahl der HTTP `GET` Anfragen für Objekte, die mithilfe eines Object Lambda Access Points gestellt wurden. Diese Metrik umfasst keine Operationen zum Auflisten. Einheiten: Anzahl Gültige Statistiken: Summe
`BytesUploaded`	Die Anzahl von Bytes, die mithilfe eines Object Lambda Access Point in einen Amazon-S3-Bucket hochgeladen wurden, wobei die Anforderung einen Textkörper enthält. Einheiten: Byte Gültige Statistiken: Durchschnitt (Byte pro Anforderung), Summe (Byte pro Zeitraum), Stichprobenanzahl, Min, Max (entspricht p100), beliebiges Perzentil zwischen p0,0 und p99,9
`PostRequests`	Die Anzahl der HTTP `POST` Anfragen, die mithilfe eines Object Lambda Access Points an einen Amazon S3 S3-Bucket gestellt wurden.

Metrik	Beschreibung
	Einheiten: Anzahl Gültige Statistiken: Summe
`PutRequests`	Die Anzahl der HTTP `PUT` Anfragen für Objekte in einem Amazon S3 S3-Bucket, die mithilfe eines Object Lambda Access Points gestellt wurden. Einheiten: Anzahl Gültige Statistiken: Summe
`DeleteRequests`	Die Anzahl der HTTP `DELETE` Anfragen für Objekte in einem Amazon S3 S3-Bucket, die mithilfe eines Object Lambda Access Points gestellt wurden. Diese Metrik umfasst DeleteObjectsAnfragen. Diese Metrik zeigt die Anzahl der gestellten Anfragen, nicht die Anzahl der gelöschten Objekte an. Einheiten: Anzahl Gültige Statistiken: Summe
`BytesDownloaded`	Die Anzahl der heruntergeladenen Bytes für Anforderungen an einen Amazon-S3-Bucket mithilfe eines Object Lambda Access Point, wobei die Antwort einen Textkörper enthält Einheiten: Byte Gültige Statistiken: Durchschnitt (Byte pro Anforderung), Summe (Byte pro Zeitraum), Stichprobenanzahl, Min, Max (entspricht p100), beliebiges Perzentil zwischen p0,0 und p99,9

Metrik	Beschreibung
`FirstByteLatency`	Zeit pro Anforderung vom Zeitpunkt des Eingangs der vollständigen Anfragen bei einem Amazon-S3-Bucket über einen Object Lambda Access Point bis zum Beginn des Zurückgebens der Antwort. Diese Metrik hängt von der Laufzeit der AWS Lambda -Funktion zum Transformieren des Objekts ab, bevor die Funktion die Bytes an den Object Lambda Access Point zurückgibt. Einheiten: Millisekunden Gültige Statistiken: Durchschnitt, Summe, Min, Max (entspricht p100), Stichprobenanzahl, beliebiges Perzentil zwischen p0,0 und p100
`TotalRequestLatency`	Zeit pro Anforderung vom Zeitpunkt des Eingangs des ersten Bytes bis zum Senden des letzten Bytes an einen Object Lambda Access Point. Diese Metrik umfasst die erforderliche Zeit, um den Textkörper zu erhalten und den Antworttextkörper zu senden, welche nicht in `FirstByteLatency` enthalten ist. Einheiten: Millisekunden Gültige Statistiken: Durchschnitt, Summe, Min, Max (entspricht p100), Stichprobenanzahl, beliebiges Perzentil zwischen p0,0 und p100
`HeadRequests`	Die Anzahl der HTTP `HEAD` Anfragen, die mithilfe eines Object Lambda Access Points an einen Amazon S3 S3-Bucket gestellt wurden. Einheiten: Anzahl Gültige Statistiken: Summe
`ListRequests`	Die Anzahl der HTTP `GET` Anfragen, die den Inhalt eines Amazon S3 S3-Buckets auflisten. Diese Metrik enthält sowohl `ListObjects` - als auch `ListObjectsV2` -Operationen. Einheiten: Anzahl Gültige Statistiken: Summe

Metrik	Beschreibung
`4xxErrors`	Die Anzahl der HTTP 4 xx Client-Fehlerstatuscode-Anfragen, die mithilfe eines Object Lambda Access Points mit einem Wert von entweder 0 oder 1 an einen Amazon S3 S3-Bucket gestellt wurden. Die Durchschnittsstatistik zeigt die Fehlerrate und die Summenstatistik die Anzahl dieses Fehlertyps für den jeweiligen Zeitraum an. Einheiten: Anzahl Gültige Statistiken: Durchschnitt (Berichte pro Anforderung), Summe (Berichte pro Zeitraum), Min, Max, Stichprobenanzahl
`5xxErrors`	Die Anzahl der HTTP 5 xx-Serverfehlerstatuscode-Anfragen, die mithilfe eines Object Lambda Access Points mit einem Wert von entweder 0 oder 1 an einen Amazon S3 S3-Bucket gestellt wurden. Die Durchschnittsstatistik zeigt die Fehlerrate und die Summenstatistik die Anzahl dieses Fehlertyps für den jeweiligen Zeitraum an. Einheiten: Anzahl Gültige Statistiken: Durchschnitt (Berichte pro Anforderung), Summe (Berichte pro Zeitraum), Min, Max, Stichprobenanzahl
`ProxiedRequests`	Die Anzahl der HTTP Anfragen an einen Object Lambda Access Point, die die API Standardantwort von Amazon S3 zurückgeben. (Für solche Anforderungen ist keine Lambda-Funktion konfiguriert.) Einheiten: Anzahl Gültige Statistiken: Summe
`InvokedLambda`	Die Anzahl der HTTP Anfragen an ein S3-Objekt, bei denen eine Lambda-Funktion aufgerufen wurde. Einheiten: Anzahl Gültige Statistiken: Summe

Metrik	Beschreibung
LambdaResponseRequests	Die Anzahl von WriteGetObjectResponse -Anforderungen, die von der Lambda-Funktion gestellt wurden. Diese Metrik gilt nur für GetObject -Anforderungen.
LambdaResponse4xx	Die Anzahl von HTTP 4 xx Client-Fehlern, die beim Aufrufen WriteGetObjectResponse von einer Lambda-Funktion aus auftreten. Diese Metrik liefert dieselben Informationen wie 4xxErrors , jedoch nur für WriteGetObjectResponse -Aufrufe.
LambdaResponse5xx	Die Anzahl der HTTP 5 xx-Serverfehler, die beim Aufrufen WriteGetObjectResponse von einer Lambda-Funktion auftreten. Diese Metrik liefert dieselben Informationen wie 5xxErrors , jedoch nur für WriteGetObjectResponse -Aufrufe.

Amazon S3 S3-Abmessungen in CloudWatch

Die nachstehenden Dimensionen werden verwendet, um Amazon-S3-Metriken zu filtern.

Dimension	Beschreibung
BucketName	Diese Dimension filtert die angeforderten Daten nur für den identifizierten Bucket.
StorageType	Diese Dimension filtert die Daten, die Sie in einem Bucket gespeichert haben, nach den folgenden Speichertypen: • StandardStorage – Die Gesamtzahl der Bytes für Objekte in der STANDARD-Speicherklasse. • IntelligentTieringAAStorage – Die Anzahl von Bytes für Objekte in Archive Access Tier der INTELLIGENT_TIERING -Speicherklasse. • IntelligentTieringAIAStorage – Die Anzahl von Bytes für Objekte in Stufe Archive Instant Access Tier der INTELLIGENT_TIERING -Speicherklasse.

Dimension	Beschreibung
	• `IntelligentTieringDAAStorage` – Die Anzahl von Bytes für Objekte in Deep Archive Access Tier der `INTELLIGENT_TIERING` -Speicherklasse. • `IntelligentTieringFAStorage` – Die Anzahl von Bytes für Objekte in Frequent Access Tier der `INTELLIGENT_TIERING` -Speicherklasse. • `IntelligentTieringIAStorage` – Die Anzahl von Bytes für Objekte in Infrequent Access Tier der `INTELLIGENT_TIERING` -Speicherklasse. • `StandardIAStorage` — Die Anzahl der Byte, die für Objekte in der Speicherklasse S3 Standard-Infrequent Access (`STANDARD_IA`) verwendet werden. • `StandardIASizeOverhead` – Die Anzahl von Bytes für Objekte mit einer Größe von weniger als 128 KB in der `STANDARD_IA` -Speicherklasse. • `IntAAObjectOverhead` – Für jedes Objekt in der `INTELLIGENT_TIERING` -Speicherklasse in Archive Access Tier fügt S3 Glacier 32 KB Speicher für Index und zugehörige Metadaten hinzu. Diese zusätzlichen Daten sind erforderlich, um Ihr Objekt zu identifizieren und wiederher zustellen. Für diesen zusätzlichen Speicherplatz werden Ihnen die Gebühren für S3 Glacier Flexible Retrieval in Rechnung gestellt. • `IntAAS3ObjectOverhead` – Für jedes Objekt in der `INTELLIGENT_TIERING` -Speicherklasse in Archive Access Tier verwendet Amazon S3 8 KB Speicher für den Namen des Objekts und andere Metadaten. Für diesen zusätzlichen Speicherplatz werden Ihnen -S3-Standardgebühren in Rechnung gestellt. • `IntDAAObjectOverhead` – Für jedes Objekt in der `INTELLIGENT_TIERING` -Speicherklasse in Deep Archive Access Tier fügt S3 Glacier 32 KB Speicher für Index und zugehörige Metadaten hinzu. Diese zusätzlichen Daten sind

Dimension	Beschreibung
	erforderlich, um Ihr Objekt zu identifizieren und wiederher zustellen. Für diesen zusätzlichen Speicherplatz werden Ihnen die S3 Glacier Deep Archive-Speichergebühren in Rechnung gestellt. • `IntDAAS3ObjectOverhead` – Für jedes Objekt in der `INTELLIGENT_TIERING` -Speicherklasse in Deep Archive Access Tier fügt Amazon S3 8 KB Speicher für Index und zugehörige Metadaten hinzu. Diese zusätzlichen Daten sind erforderlich, um Ihr Objekt zu identifizieren und wiederher zustellen. Für diesen zusätzlichen Speicherplatz werden Ihnen -S3-Standardgebühren in Rechnung gestellt. • `OneZoneIAStorage` – Die Anzahl von Bytes für Objekte in der Speicherklasse S3 One Zone-Infrequent Access (`ONEZONE_IA`). • `OneZoneIASizeOverhead` – Die Anzahl von Bytes für Objekte mit einer Größe von weniger als 128 KB in der `ONEZONE_IA` -Speicherklasse. • `ReducedRedundancyStorage` — Die Anzahl der Byte, die für Objekte in der Klasse Reduced Redundancy Storage () verwendet werden. RRS • `GlacierIRSizeOverhead` – Die Anzahl von Bytes für Objekte, die kleiner als 128 KB sind, in der Speicherklasse S3 Glacier Instant Retrieval. • `GlacierInstantRetrievalStorage` – Die Anzahl von Bytes für Objekte in der Speicherklasse S3 Glacier Instant Retrieval. • `GlacierStorage` – Die Anzahl von Bytes für Objekte in der Speicherklasse S3 Glacier Flexible Retrieval. • `GlacierStagingStorage` — Die Anzahl der Byte, die für Teile von mehrteiligen Upload-Objekten verwendet wurden, bevor die `CompleteMultipartUpload` Anforderung für Objekte in der S3 Glacier Flexible Retrieval-Speicherklasse abgeschlossen ist.

Dimension	Beschreibung
	• `GlacierObjectOverhead` – Für jedes archivierte Objekt fügt S3 Glacier 32 KB Speicher für den Index und zugehörige Metadaten hinzu. Diese zusätzlichen Daten sind erforderlich, um Ihr Objekt zu identifizieren und wiederherzustellen. Für diesen zusätzlichen Speicherplatz werden Ihnen die Gebühren für S3 Glacier Flexible Retrieval in Rechnung gestellt. • `GlacierS3ObjectOverhead` – Für jedes in S3 Glacier Flexible Retrieval archivierte Objekt verwendet Amazon S3 8 KB Speicher für den Namen des Objekts und andere Metadaten. Für diesen zusätzlichen Speicherplatz werden Ihnen -S3-Standardgebühren in Rechnung gestellt. • `DeepArchiveStorage` – Die Anzahl von Bytes für Objekte in der Speicherklasse S3 Glacier Deep Archive. • `DeepArchiveObjectOverhead` – Für jedes archivierte Objekt fügt S3 Glacier 32 KB Speicher für den Index und zugehörige Metadaten hinzu. Diese zusätzlichen Daten sind erforderlich, um Ihr Objekt zu identifizieren und wiederherzustellen. Für diesen zusätzlichen Speicherplatz werden Ihnen die S3 Glacier Deep Archive-Gebühren in Rechnung gestellt. • `DeepArchiveS3ObjectOverhead` – Für jedes in S3 Glacier Deep Archive archivierte Objekt verwendet Amazon S3 8 KB Speicher für den Namen des Objekts und andere Metadaten. Für diesen zusätzlichen Speicherplatz werden Ihnen -S3-Standardgebühren in Rechnung gestellt. • `DeepArchiveStagingStorage` — Die Anzahl der Byte, die für Teile von mehrteiligen Upload-Objekten verwendet werden, bevor die `CompleteMultipartUpload` Anforderung für Objekte in der S3 Glacier Deep Archive Archive-Speicherklasse abgeschlossen ist. • `ExpressOneZoneStorage` – Die Gesamtzahl der Bytes für Objekte in der S3-Express-One-Zone-Speicherklasse.

Dimension	Beschreibung
`FilterId`	Diese Dimension filtert Metrikkonfigurationen, die Sie für Anforderungsmetriken für einen Bucket angeben. Wenn Sie eine Metrikkonfiguration erstellen, geben Sie eine Filter-ID an (z. B. ein Präfix, ein Tag oder einen Zugriffspunkt). Weitere Informati onen finden Sie unter Erstellen einer Metrik-Konfiguration.

Dimensionen der S3-Replikation in CloudWatch

Die folgenden Dimensionen werden verwendet, um S3-Replikationsmetriken zu filtern.

Dimension	Beschreibung
`SourceBucket`	Der Name der Bucket-Objekte, aus denen repliziert wird.
`DestinationBucket`	Der Name der Bucket-Objekte, in die repliziert werden.
`RuleId`	Eine eindeutige Kennung für die Regel, die die Aktualisierung dieser Replikationsmetrik ausgelöst hat.

Abmessungen der S3-Speicherlinse in CloudWatch

Eine Liste der Dimensionen, die zum Filtern von S3 Storage Lens-Metriken verwendet werden CloudWatch, finden Sie unterDimensionen.

Abmessungen der Lambda-Anforderung von S3 Object in CloudWatch

Die folgenden Dimensionen werden verwendet, um Daten aus einem Object Lambda Access Point zu filtern.

Dimension	Beschreibung
`AccessPointName`	Der Name des Zugriffspunkts, von dem Anforderungen gestellt werden.
`DataSourceARN`	Die Quelle, aus der der Object Lambda Access Point die Daten abruft. Wenn die Anfrage eine Lambda-Funktion aufruft, bezieht sich diese auf

Dimension	Beschreibung
	den Lambda-Amazon-Ressourcennamen (). ARN Andernfalls bezieht sich dies auf den Access Point. ARN

Amazon S3 S3-Nutzungsmetriken

Sie können CloudWatch Nutzungsmetriken verwenden, um einen Überblick über die Ressourcennutzung Ihres Kontos zu erhalten. Verwenden Sie diese Metriken, um Ihre aktuelle Servicenutzung in CloudWatch Diagrammen und Dashboards zu visualisieren.

Die Nutzungsmetriken von Amazon S3 entsprechen den AWS Servicekontingenten. Sie können Alarme konfigurieren, mit denen Sie benachrichtigt werden, wenn sich Ihre Nutzung einem Servicekontingent nähert. Weitere Informationen zur CloudWatch Integration mit Servicekontingenten finden Sie unter AWS Nutzungsmetriken im CloudWatch Amazon-Benutzerhandbuch.

Amazon S3 veröffentlicht die folgenden Metriken im `AWS/Usage` Namespace.

Metrik	Beschreibung
`ResourceCount`	Die Anzahl der angegebenen Ressourcen, die in Ihrem Konto ausgeführt werden. Die Ressourcen werden durch die Dimensionen definiert, die der Metrik zugeordnet sind.

Die folgenden Dimensionen werden verwendet, um die von Amazon S3 veröffentlichten Nutzungsmetriken zu verfeinern.

Dimension	Beschreibung
`Service`	Der Name des AWS Dienstes, der die Ressource enthält. Für Amazon S3 S3-Nutzungsmetriken ist der Wert für diese Dimension`S3`.
`Type`	Der Typ von Entität, die gemeldet wird. Derzeit ist der einzig gültige Wert für Amazon S3 S3-Nutzungsmetriken`Resource`.
`Resource`	Der Typ der Ressource, die ausgeführt wird. Derzeit ist der einzig gültige Wert für Amazon S3 S3-Nutzungsmetrike

Dimension	Beschreibung
	nGeneralPurposeBuckets , der die Anzahl der Allzweck-Buckets in einem AWS-Konto zurückgibt. Allzweck-Buckets ermöglichen Objekte, die in allen Speicherklassen gespeichert werden, mit Ausnahme von S3 Express One Zone.

Zugriff auf Metriken CloudWatch

Sie können die folgenden Vorgehensweisen nutzen, um die Speichermetriken von Amazon S3 anzuzeigen. Damit Amazon-S3-Metriken berücksichtigt werden, müssen Sie Zeitstempel für Anfang und Ende angeben. Für Metriken für einen bestimmten 24-Stunden-Zeitraum setzen Sie das Zeitintervall auf 86400 Sekunden, die Anzahl der Sekunden eines Tages. Denken Sie daran, die Dimensionen `BucketName` und `StorageType` festzulegen.

Mit dem AWS CLI

Wenn Sie beispielsweise den verwenden möchten, AWS CLI um den Durchschnitt der Größe eines bestimmten Buckets in Byte zu ermitteln, könnten Sie den folgenden Befehl verwenden:

```
aws cloudwatch get-metric-statistics --metric-name BucketSizeBytes --namespace AWS/S3
 --start-time 2016-10-19T00:00:00Z --end-time 2016-10-20T00:00:00Z --statistics Average
 --unit Bytes --region us-west-2 --dimensions Name=BucketName,Value=amzn-s3-demo-bucket
 Name=StorageType,Value=StandardStorage --period 86400 --output json
```

Dieses Beispiel erzeugt die folgende Ausgabe.

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-10-19T00:00:00Z",
            "Average": 1025328.0,
            "Unit": "Bytes"
        }
    ],
    "Label": "BucketSizeBytes"
}
```

Verwenden der S3-Konsole

So zeigen Sie Metriken mit der CloudWatch Amazon-Konsole an

1. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.
2. Wählen Sie im linken Navigationsbereich Metrics (Metriken) aus.
3. Wählen Sie den Namespace S3 aus.
4. (Optional) Um eine Metrik anzuzeigen, geben Sie den Metriknamen in das Suchfeld ein.
5. (Optional) Um nach der StorageTypeDimension zu filtern, geben Sie den Namen der Speicherklasse in das Suchfeld ein.

Um eine Liste gültiger Messwerte anzuzeigen, die für Sie gespeichert AWS-Konto wurden, verwenden Sie AWS CLI

- Geben Sie als Eingabeaufforderung den folgenden Befehl ein.

```
aws cloudwatch list-metrics --namespace "AWS/S3"
```

Weitere Informationen zu den für den Zugriff auf CloudWatch Dashboards erforderlichen Berechtigungen finden Sie unter CloudWatch Amazon-Dashboard-Berechtigungen im CloudWatch Amazon-Benutzerhandbuch.

CloudWatch Metriken, Konfigurationen

Mit Amazon CloudWatch Request Metrics für Amazon S3 können Sie CloudWatch 1-Minuten-Metriken erhalten, CloudWatch Alarme einrichten und auf CloudWatch Dashboards zugreifen, um den near-real-time Betrieb und die Leistung Ihres Amazon S3 S3-Speichers zu überprüfen. Für Anwendungen, die von Cloud-Speicher abhängig sind, ermöglichen diese Metriken Ihnen, Betriebsprobleme schnell zu identifizieren und entsprechende Maßnahmen zu ergreifen. Wenn diese 1-Minuten-Metriken aktiviert sind, stehen sie standardmäßig auf Ebene der Amazon-S3-Buckets zur Verfügung.

Wenn Sie die CloudWatch Anforderungsmetriken für die Objekte in einem Bucket abrufen möchten, müssen Sie eine Metrikkonfiguration für den Bucket erstellen. Weitere Informationen finden Sie unter Erstellen einer CloudWatch Metrikkonfiguration für alle Objekte in Ihrem Bucket.

Sie können auch ein freigegebenes Präfix, Objekt-Tags oder einen Zugriffspunkt verwenden, um einen Filter für die gesammelten Metriken zu definieren. Mit dieser Methode zum Definieren eines Filters können Sie also Metrikfilter an bestimmte Geschäftsanwendungen, Workflows oder interne Organisationen anpassen. Weitere Informationen finden Sie unter Erstellen einer Metrik-Konfiguration, die nach dem Präfix, Objekt-Tag oder Zugriffspunkt filtert. Weitere Informationen zu den verfügbaren CloudWatch Metriken und den Unterschieden zwischen Speicher- und Anforderungsmetriken finden Sie unterMetriken mit Amazon überwachen CloudWatch.

Berücksichtigen Sie bei Verwendung von Metrik-Konfigurationen Folgendes:

- Sie können maximal 1 000 Metrik-Konfigurationen pro Bucket verwenden.
- Unter Verwendung von Filtern können Sie wählen, welche Objekte in einem Bucket Metrik-Konfigurationen enthalten sollen. Sie können nach einem freigegebenen Präfix, Objekt-Tag oder Zugriffspunkt filtern, um Metrikfilter an bestimmte Geschäftsanwendungen, Workflows oder interne Organisationen auszurichten. Um Metriken für den gesamten Bucket anzufordern, erstellen Sie eine Metrik-Konfiguration ohne Filter.
- Metrik-Konfigurationen sind nur erforderlich, um Anforderungsmetriken zu aktivieren. Tägliche Speichermetriken auf Bucket-Ebene sind immer aktiviert und werden allen Kunden ohne zusätzliche Kosten zur Verfügung gestellt. Derzeit ist es nicht möglich, tägliche Speichermetriken für eine gefilterte Untermenge von Objekten zu erhalten.
- Jede Metrik-Konfiguration unterstützt den gesamten Satz verfügbarer Anforderungsmetriken. Operationsspezifische Metriken (z. B. `PostRequests`) werden nur gemeldet, wenn es Anforderungen dieses Typs für den Bucket oder Filter gibt.
- Anforderungsmetriken werden für Vorgänge auf Objektebene gemeldet. Sie werden auch für Operationen gemeldet, die Bucket-Inhalte auflisten, wie GETBucket (List Objects), GETBucket Object Versions und List Multipart Uploads, aber sie werden nicht für andere Operationen mit Buckets gemeldet.
- Anforderungsmetriken unterstützen das Filtern mit Präfixen, Objekt-Tags oder Zugriffspunkten, Speichermetriken dagegen nicht.

Bereitstellung von Kennzahlen nach bestem Wissen CloudWatch

CloudWatch Die Metriken werden nach bestem Wissen und Gewissen bereitgestellt. Die meisten Anfragen für ein Amazon S3 S3-Objekt mit Anforderungsmetriken führen dazu, dass ein Datenpunkt an gesendet wird CloudWatch.

Die Vollständigkeit und Rechtzeitigkeit der Metriken ist nicht garantiert. Der Datenpunkt für eine bestimmte Anforderung wird möglicherweise mit einem Zeitstempel zurückgegeben, der nach der tatsächlichen Anforderungsverarbeitung liegt. Der Datenpunkt kann sich um eine Minute verzögern, bevor er verfügbar ist CloudWatch, oder er wird möglicherweise gar nicht zugestellt. CloudWatch Anhand von Anforderungsmetriken erhalten Sie nahezu in Echtzeit eine Vorstellung von der Art des Datenverkehrs in Ihrem Bucket. Sie sind nicht als vollständige Abrechnung aller Anforderungen vorgesehen.

Aufgrund der Best-Effort-Natur dieser Funktion enthalten die im Fakturierungs- und Kostenverwaltungs-Dashboard verfügbaren Berichte möglicherweise eine oder mehrere Zugriffsanforderungen, die nicht in den Bucket-Metriken angezeigt werden.

Weitere Informationen zur Arbeit mit CloudWatch Metriken in Amazon S3 finden Sie in den folgenden Themen.

Themen

- Erstellen einer CloudWatch Metrikkonfiguration für alle Objekte in Ihrem Bucket
- Erstellen einer Metrik-Konfiguration, die nach dem Präfix, Objekt-Tag oder Zugriffspunkt filtert
- Löschen eines Metrikfilters

Erstellen einer CloudWatch Metrikkonfiguration für alle Objekte in Ihrem Bucket

Wenn Sie Anforderungsmetriken konfigurieren, können Sie eine CloudWatch Metrikkonfiguration für alle Objekte in Ihrem Bucket erstellen oder nach Präfix, Objekt-Tag oder Access Point filtern. Die Verfahren in diesem Thema zeigen Ihnen, wie Sie eine Konfiguration für alle Objekte in Ihrem Bucket erstellen. Informationen zum Erstellen einer Konfiguration, die nach Objekt-Tag, Präfix oder Zugriffspunkt filtert, finden Sie unter Erstellen einer Metrik-Konfiguration, die nach dem Präfix, Objekt-Tag oder Zugriffspunkt filtert.

Es gibt drei Arten von CloudWatch Amazon-Metriken für Amazon S3: Speichermetriken, Anforderungsmetriken und Replikationsmetriken. Speichermetriken werden einmal pro Tag gemeldet und allen Kunden ohne zusätzliche Kosten zur Verfügung gestellt. Die Anforderungsmetriken sind in 1-Minuten-Intervallen nach einer gewissen Latenz für die Verarbeitung verfügbar. Anforderungsmetriken werden zum CloudWatch Standardtarif abgerechnet. Sie müssen sich dafür entscheiden, Metriken anzufordern, indem Sie sie in der Konsole konfigurieren oder Amazon S3 verwendenAPI. S3-Replikationsmetriken enthalten detaillierte Metriken für die Replikationsregeln in Ihrer Replikationskonfiguration. Mit Replikationsmetriken können Sie den minute-by-minute Fortschritt

überwachen, indem Sie ausstehende Bytes, ausstehende Operationen, Operationen, bei denen die Replikation fehlgeschlagen ist, und die Replikationslatenz verfolgen.

Weitere Informationen zu CloudWatch Metriken für Amazon S3 finden Sie unterMetriken mit Amazon überwachen CloudWatch.

Sie können Metrikkonfigurationen zu einem Bucket hinzufügen, indem Sie die Amazon S3-Konsole, die AWS Command Line Interface (AWS CLI) oder die Amazon S3 verwenden RESTAPI.

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets (Buckets) den Namen des Buckets mit den Objekte aus, für die Sie Anforderungsmetriken anfordern möchten.
3. Wählen Sie den Tab Metrics (Metriken).
4. Wählen Sie unter Bucket metrics (Bucket-Metriken) View additional charts (Zusätzliche Diagramme anzeigen) aus.
5. Wählen Sie den Tab Request metrics (Anforderungsmetriken).
6. Wählen Sie Create Filter (Filter erstellen).
7. Geben Sie im Feld Filter name (Filtername) Ihren Filternamen ein.

   Namen dürfen nur Buchstaben, Zahlen, Punkte, Bindestriche und Unterstriche enthalten. Wir empfehlen, den Namen `EntireBucket` für einen Filter zu verwenden, der für alle Objekte gilt.
8. Wählen Sie unter Filter scope (Filterbereich) die Option This filter applies to all objects in the bucket (Dieser Filter gilt für alle Objekte im Bucket).

   Sie können auch einen Filter definieren, sodass die Metriken nur für eine Untermenge von Objekten im Bucket erfasst und gemeldet werden. Weitere Informationen finden Sie unter Erstellen einer Metrik-Konfiguration, die nach dem Präfix, Objekt-Tag oder Zugriffspunkt filtert.
9. Wählen Sie Save Changes (Änderungen speichern).
10. Wählen Sie im Tab Request metrics (Anforderungsmetriken) unter Filters (Filter) den Filter aus, den Sie gerade erstellt haben.

    Nach etwa 15 Minuten CloudWatch beginnt die Erfassung dieser Anforderungsmetriken. Sie können sie im Tab Request metrics (Anforderungsmetriken) sehen. Sie können Diagramme für die Metriken auf Amazon S3 oder der CloudWatch Konsole sehen. Anforderungsmetriken

werden zum CloudWatch Standardtarif in Rechnung gestellt. Weitere Informationen finden Sie unter CloudWatch Amazon-Preise.

Mit dem REST API

Sie können Metrikkonfigurationen auch programmgesteuert mit dem Amazon S3 hinzufügen. REST API Weitere Informationen zum Hinzufügen und Arbeiten mit Metrikkonfigurationen finden Sie in den folgenden Themen in der Amazon Simple Storage Service API Reference:

- PUTKonfiguration von Bucket-Metriken
- GETKonfiguration von Bucket Metric
- List Bucket-Metrikkonfiguration
- DELETEKonfiguration von Bucket Metric

Unter Verwendung der AWS CLI

1. Installiere und richte das ein AWS CLI. Anleitungen finden Sie unter Installieren, Aktualisieren und Deinstallieren der AWS CLI im AWS Command Line Interface -Benutzerhandbuch.
2. Öffnen Sie ein Terminalfenster.
3. Führen Sie den folgenden Befehl aus, um eine Metrik-Konfiguration hinzuzufügen.

```
aws s3api put-bucket-metrics-configuration --endpoint https://s3.us-
west-2.amazonaws.com --bucket bucket-name --id metrics-config-id --metrics-
configuration '{"Id":"metrics-config-id"}'
```

Erstellen einer Metrik-Konfiguration, die nach dem Präfix, Objekt-Tag oder Zugriffspunkt filtert

Es gibt drei Arten von CloudWatch Amazon-Metriken für Amazon S3: Speichermetriken, Anforderungsmetriken und Replikationsmetriken. Speichermetriken werden einmal pro Tag gemeldet und allen Kunden ohne zusätzliche Kosten zur Verfügung gestellt. Die Anforderungsmetriken sind in 1-Minuten-Intervallen nach einer gewissen Latenz für die Verarbeitung verfügbar. Anforderungsmetriken werden zum CloudWatch Standardtarif abgerechnet. Sie müssen sich dafür entscheiden, Metriken anzufordern, indem Sie sie in der Konsole konfigurieren oder Amazon S3 verwendenAPI. S3-Replikationsmetriken enthalten detaillierte Metriken für die Replikationsregeln in Ihrer Replikationskonfiguration. Mit Replikationsmetriken können Sie den minute-by-minute Fortschritt

überwachen, indem Sie ausstehende Bytes, ausstehende Operationen, Operationen, bei denen die Replikation fehlgeschlagen ist, und die Replikationslatenz verfolgen.

Weitere Informationen zu CloudWatch Metriken für Amazon S3 finden Sie unterMetriken mit Amazon überwachen CloudWatch.

Wenn Sie CloudWatch Metriken konfigurieren, können Sie einen Filter für alle Objekte in Ihrem Bucket erstellen, oder Sie können die Konfiguration in Gruppen verwandter Objekte innerhalb eines einzelnen Buckets filtern. Sie können Objekte in einem Bucket zur Aufnahme in eine Metrik-Konfiguration abhängig von einem oder mehreren der folgenden Filtertypen filtern:

- Object key name prefix (Objekt-Schlüsselnamepräfix) – Obwohl das Amazon-S3-Datenmodell eine flache Struktur aufweist, können Sie eine Hierarchie durch die Verwendung eines Präfixes inferieren. Die Amazon-S3-Konsole unterstützt diese Präfixe mit dem Ordnerkonzept. Wenn Sie nach dem Präfix filtern, werden Objekte mit demselben Präfix in die Metrik-Konfiguration aufgenommen. Weitere Informationen zu Präfixen finden Sie unter Organisieren von Objekten mit Präfixen.
- Tag – Sie können Objekten Markierungen hinzufügen, d. h. Schlüsselwert-Namenspaare. Mit Markierungen können Sie Objekte einfacher finden und organisieren. Markierungen können auch als Filter für Metrik-Konfigurationen verwendet werden. Weitere Informationen über Objekt-Markierungen finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen.
- Zugriffspunkt – S3-Zugriffspunkte sind benannte Netzwerkendpunkte, die Buckets zugeordnet sind und vereinfachen die skalierbare Verwaltung des Datenzugriffs auf freigegebene Datensätze in S3. Wenn Sie einen Zugriffspunkt-Filter erstellen, enthält Amazon S3 Anforderungen an den Zugriffspunkt, den Sie in der Metrikkonfiguration angeben. Weitere Informationen finden Sie unter Überwachen und Protokollieren von Zugriffspunkten.

  > **Note**
  >
  > Wenn Sie eine Metrikkonfiguration erstellen, die nach Access Point filtert, müssen Sie den Access Point Amazon Resource Name (ARN) verwenden, nicht den Access Point-Alias. Stellen Sie sicher, dass Sie den ARN für den Access Point selbst verwenden, nicht den ARN für ein bestimmtes Objekt. Weitere Informationen zum Access Point ARNs finden Sie unterVerwenden von Zugriffspunkten.

Wenn Sie einen Filter angeben, können nur Anforderungen, die für einzelne Objekte ausgeführt werden, mit dem Filter übereinstimmen und in die gemeldeten Metriken aufgenommen werden.

Anfragen wie DeleteObjectsund `ListObjects` Anfragen geben keine Metriken für Konfigurationen mit Filtern zurück.

Um ein komplexeres Filtern anzufordern, wählen Sie zwei oder mehr Elemente aus. Nur Objekte, die alle diese Elemente besitzen, werden in die Metrik-Konfiguration aufgenommen. Wenn Sie keine Filter setzen, werden alle Objekte aus dem Bucket in die Metrikenkonfiguration aufgenommen.

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets (Buckets) den Namen des Buckets mit den Objekte aus, für die Sie Anforderungsmetriken anfordern möchten.
3. Wählen Sie den Tab Metrics.
4. Wählen Sie unter Bucket metrics (Bucket-Metriken) View additional charts (Zusätzliche Diagramme anzeigen) aus.
5. Wählen Sie den Tab Request metrics (Anforderungsmetriken).
6. Wählen Sie Create Filter (Filter erstellen).
7. Geben Sie im Feld Filter name (Filtername) Ihren Filternamen ein.

   Namen dürfen nur Buchstaben, Zahlen, Punkte, Bindestriche und Unterstriche enthalten.
8. Wählen Sie unter Filterumfang Begrenzen Sie den Bereich dieses Filters mit einem Präfix, Objekt-Tags und einem S3-Zugriffspunkt oder einer Kombination aus allen drei.
9. Wählen Sie unter Filtertyp mindestens einen Filtertyp aus: Präfix,Objekt-Tags, oder Zugriffspunkt.
10. Um einen Präfixfilter zu definieren und den Bereich des Filters auf einen einzelnen Pfad zu beschränken, geben Sie im Feld Präfix ein Präfix ein.
11. Um einen Objekt-Tags-Filter zu definieren, wählen Sie unterObjekt-Tags Hinzufügen eines Tags, und geben Sie dann ein Tag Schlüssel und Wert ein.
12. Um einen Access Point-Filter zu definieren, geben Sie im Feld S3 Access Point den Access Point ein oder wählen Sie Browse S3 ARN, um zum Access Point zu navigieren.

> **Important**
>
> Sie können keinen Zugriffspunkt-Alias eingeben. Sie müssen den ARN für den Access Point selbst eingeben, nicht den ARN für ein bestimmtes Objekt.

13. Wählen Sie Änderungen speichern.

    Amazon S3 erstellt einen Filter, der das Präfix, die Tags oder den Zugriffspunkt verwendet, den Sie angegeben haben.

14. Wählen Sie im Tab Request metrics (Anforderungsmetriken) unter Filters (Filter) den Filter aus, den Sie gerade erstellt haben.

    Sie haben nun einen Filter erstellt, der den Bereich der Anforderungsmetriken nach Präfix, Objekt-Tags oder Zugriffspunkt einschränkt. Ungefähr 15 Minuten nach CloudWatch Beginn der Verfolgung dieser Anforderungsmetriken können Sie Diagramme für die Metriken sowohl auf Amazon S3 als auch auf CloudWatch Konsolen sehen. Anforderungsmetriken werden zum CloudWatch Standardtarif abgerechnet. Weitere Informationen finden Sie unter CloudWatch Amazon-Preise.

    Sie können Anforderungsmetriken auch auf Bucket-Ebene konfigurieren. Weitere Informationen finden Sie unter Erstellen einer CloudWatch Metrikkonfiguration für alle Objekte in Ihrem Bucket.

Mit dem AWS CLI

1. Installiere und richte das ein AWS CLI. Anleitungen finden Sie unter Installieren, Aktualisieren und Deinstallieren der AWS CLI im AWS Command Line Interface -Benutzerhandbuch.
2. Öffnen Sie ein Terminalfenster.
3. Führen Sie einen der folgenden Befehle aus, um eine Metrik-Konfiguration hinzuzufügen:

    Example : So filtern Sie nach Präfix

    ```
    aws s3api put-bucket-metrics-configuration --bucket amzn-s3-demo-bucket --
    id metrics-config-id --metrics-configuration '{"Id":"metrics-config-id", "Filter":
    {"Prefix":"prefix1"}} '
    ```

    Example : So filtern Sie nach Tags

    ```
    aws s3api put-bucket-metrics-configuration --bucket amzn-s3-demo-bucket --
    id metrics-config-id --metrics-configuration '{"Id":"metrics-config-id", "Filter":
    {"Tag": {"Key": "string", "Value": "string"}} '
    ```

Example : So filtern Sie nach Zugriffspunkt

```
aws s3api put-bucket-metrics-configuration --bucket amzn-s3-demo-bucket --
id metrics-config-id --metrics-configuration '{"Id":"metrics-config-id", "Filter":
{"AccessPointArn":"arn:aws:s3:Region:account-id:accesspoint/access-point-name"}} '
```

Example : So filtern Sie nach Präfix, Tags und Zugriffspunkt

```
aws s3api put-bucket-metrics-configuration --endpoint https://
s3.Region.amazonaws.com --bucket amzn-s3-demo-bucket --id metrics-config-id --
metrics-configuration '
{
    "Id": "metrics-config-id",
    "Filter": {
        "And": {
            "Prefix": "string",
            "Tags": [
                {
                    "Key": "string",
                    "Value": "string"
                }
            ],
            "AccessPointArn": "arn:aws:s3:Region:account-id:accesspoint/access-
point-name"
        }
    }
}'
```

Mit dem REST API

Sie können Metrikkonfigurationen auch programmgesteuert mit dem Amazon S3 hinzufügen. REST API Weitere Informationen zum Hinzufügen und Arbeiten mit Metrikkonfigurationen finden Sie in den folgenden Themen in der Amazon Simple Storage Service API Reference:

- PUTKonfiguration von Bucket-Metriken
- GETKonfiguration von Bucket Metric
- List Bucket-Metrikkonfiguration
- DELETEKonfiguration von Bucket Metric

Löschen eines Metrikfilters

Sie können einen CloudWatch Amazon-Anforderungsmetrikfilter löschen, wenn Sie ihn nicht mehr benötigen. Wenn Sie einen Filter löschen, werden Ihnen keine Anforderungsmetriken mehr berechnet, die diesen spezifischen Filter verwenden. Ihnen werden jedoch weiter alle anderen vorhandenen Filterkonfigurationen berechnet.

Wenn Sie einen Filter löschen, können Sie den Filter nicht mehr für Anforderungsmetriken verwenden. Das Löschen eines Filters kann nicht rückgängig gemacht werden.

Informationen zum Erstellen eines Anforderungsmetrikfilters finden Sie in den folgenden Themen:

- Erstellen einer CloudWatch Metrikkonfiguration für alle Objekte in Ihrem Bucket
- Erstellen einer Metrik-Konfiguration, die nach dem Präfix, Objekt-Tag oder Zugriffspunkt filtert

Verwenden der S3-Konsole

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets (Buckets) den Namen Ihres Buckets.
3. Wählen Sie den Tab Metrics.
4. Wählen Sie unter Bucket metrics (Bucket-Metriken) View additional charts (Zusätzliche Diagramme anzeigen) aus.
5. Wählen Sie den Tab Request metrics (Anforderungsmetriken).
6. Wählen Sie Manage filters (Filter verwalten).
7. Wählen Sie Ihren Filter aus.

   > **Important**
   >
   > Das Löschen eines Filters kann nicht rückgängig gemacht werden.

8. Wählen Sie Delete (Löschen).

   Amazon S3 löscht Ihren Filter.

Verwenden Sie den REST API

Sie können Metrikkonfigurationen auch programmgesteuert mit dem Amazon S3 hinzufügen. REST API Weitere Informationen zum Hinzufügen und Arbeiten mit Metrikkonfigurationen finden Sie in den folgenden Themen in der Amazon Simple Storage Service API Reference:

- PUTKonfiguration von Bucket-Metriken
- GETKonfiguration von Bucket Metric
- List Bucket-Metrikkonfiguration
- DELETEKonfiguration von Bucket Metric

Amazon-S3-Ereignis-Benachrichtigungen

Sie können die Amazon-S3-Funktion für Ereignis-Benachrichtigungen verwenden, um Benachrichtigungen zu erhalten, wenn bestimmte Ereignisse in Ihrem S3-Bucket eintreten. Um Benachrichtigungen zu aktivieren, fügen Sie eine Benachrichtigungskonfiguration hinzu, die die Ereignisse angibt, die Amazon S3 veröffentlichen soll. Stellen Sie sicher, dass es auch die Ziele angibt, an die Amazon S3 die Benachrichtigungen senden soll. Sie speichern dieses Konfiguration in der einem Bucket zugeordneten Benachrichtigung-Unterressourcen. Weitere Informationen finden Sie unter Optionen für die Bucket-Konfiguration. Amazon S3 bietet Ihnen eine API Möglichkeit, diese Unterressource zu verwalten.

> **Important**
>
> Amazon-S3-Ereignis-Benachrichtigungen sind so konzipiert, dass sie mindestens einmal zugestellt werden. Ereignis-Benachrichtigungen stellen Ereignisse typischerweise in wenigen Sekunden bereit, manchmal kann dies aber auch eine Minute oder länger dauern.

Übersicht über Amazon-S3-Ereignis-Benachrichtigungen

Zurzeit kann Amazon S3 Benachrichtigungen für die folgenden Ereignisse veröffentlichen:

- Neues Objekt erstellte Ereignisse
- Ereignisse zum Entfernen von Objekten
- Wiederherstellen von Objekt-Ereignissen

- Reduzierte Redundanz: Ereignisse bei Verlust von Speicherobjekten (RRS)
- Replikationsereignisse
- S3-Lebenszyklus-Ablaufereignisse
- S3-Lebenszyklusübergangsereignisse
- Aktivieren der automatischen S3-Intelligent-Tiering-Archivierung
- Ereignismarkierungen von Objekten
- Objektereignisse ACL PUT

Eine vollständige Beschreibung aller unterstützten Ereignistypen finden Sie unter Unterstützte Ereignistypen für SQSSNS, und Lambda.

Amazon S3 kann Ereignis-Benachrichtigungsmeldungen an die folgenden Ziele senden. Sie geben den Amazon Resource Name (ARN) -Wert dieser Ziele in der Benachrichtigungskonfiguration an.

- Themen zu Amazon Simple Notification Service (AmazonSNS)
- Warteschlangen von Amazon Simple Queue Service (AmazonSQS)
- AWS Lambda Funktion
- Amazon EventBridge

Weitere Informationen finden Sie unter Unterstützte Ereignisziele.

> **Note**
>
> Amazon Simple Queue Service FIFO (First-In-First-Out) -Warteschlangen werden nicht als Ziel für Amazon S3 S3-Ereignisbenachrichtigungen unterstützt. Um eine Benachrichtigung für ein Amazon S3 S3-Ereignis an eine SQS FIFO Amazon-Warteschlange zu senden, können Sie Amazon verwenden EventBridge. Weitere Informationen finden Sie unter Amazon aktivieren EventBridge.

> **Warning**
>
> Wenn Ihre Benachrichtigung in den selben Bucket schreibt, der die Benachrichtigung auslöst, kann dies zu einer Ausführungsschleife führen. Wenn der Bucket z. B. eine Lambda-Funktion immer dann auslöst, wenn ein Objekt hochgeladen wird, und die Funktion ein Objekt in den

Bucket hochlädt, löst sich die Funktion indirekt selbst aus. Um dies zu vermeiden, verwenden Sie zwei Buckets oder konfigurieren Sie den Auslöser so, dass er nur für einen Präfix gilt, der für eingehende Objekte verwendet wird.
Weitere Informationen und ein Beispiel für die Verwendung von Amazon S3-Benachrichtigungen mit AWS Lambda finden Sie unter AWS Lambda Using with Amazon S3 im AWS Lambda Developer Guide.

Weitere Informationen zur Anzahl der Konfigurationen für Ereignisbenachrichtigungen, die Sie pro Bucket erstellen können, finden Sie unter Amazon-S3-Service-Quotas in der Allgemeinen AWS -Referenz.

Weitere Informationen zu Ereignis-Benachrichtigungen finden Sie in den folgenden Abschnitten.

Themen

- Ereignis-Benachrichtigungstypen und -ziele
- Verwendung von Amazon SQSSNS, Amazon und Lambda
- Verwenden EventBridge

Ereignis-Benachrichtigungstypen und -ziele

Amazon S3 unterstützt verschiedene Arten von Ereignis-Benachrichtigungen und Ziele, wo die Benachrichtigungen veröffentlicht werden können. Sie können den Ereignistyp und das Ziel angeben, wenn Sie Ihre Ereignis-Benachrichtigungen konfigurieren. Für jede Ereignisbenachrichtigung kann nur ein Ziel angegeben werden. Amazon-S3-Ereignisbenachrichtigungen senden für jede Benachrichtigungsmeldung einen Ereigniseintrag.

Themen

- Unterstützte Ereignisziele
- Unterstützte Ereignistypen für SQSSNS, und Lambda
- Unterstützte Ereignistypen für Amazon EventBridge
- Bestellung von Veranstaltungen und doppelte Ereignisse

Unterstützte Ereignisziele

Amazon S3 kann Ereignis-Benachrichtigungsmeldungen an die folgenden Ziele senden.

- Themen zu Amazon Simple Notification Service (AmazonSNS)
- Warteschlangen von Amazon Simple Queue Service (AmazonSQS)
- AWS Lambda
- Amazon EventBridge

Für jede Ereignisbenachrichtigung kann jedoch nur ein Zieltyp angegeben werden.

> **Note**
>
> Sie müssen Amazon S3 S3-Berechtigungen erteilen, um Nachrichten an ein SNS Amazon-Thema oder eine SQS Amazon-Warteschlange zu posten. Sie müssen Amazon S3 außerdem die Erlaubnis erteilen, eine AWS Lambda Funktion in Ihrem Namen aufzurufen. Anweisungen zum Erteilen dieser Berechtigungen finden Sie unter Erteilen von Berechtigungen zur Veröffentlichung von Ereignis-Benachrichtigungsmeldungen an einem Ziel.

SNSAmazon-Thema

Amazon SNS ist ein flexibler, vollständig verwalteter Push-Messaging-Dienst. Sie können diesen Dienst verwenden, um Nachrichten an mobile Geräte oder verteilte Services zu senden. Mit SNS können Sie eine Nachricht einmal veröffentlichen und sie ein- oder mehrmals zustellen. Derzeit SNS ist Standard nur als Ziel für S3-Ereignisbenachrichtigungen SNS FIFO zulässig, nicht jedoch.

Amazon koordiniert und verwaltet SNS sowohl den Versand als auch die Zustellung von Nachrichten an abonnierte Endpunkte oder Kunden. Sie können die SNS Amazon-Konsole verwenden, um ein SNS Amazon-Thema zu erstellen, an das Ihre Benachrichtigungen gesendet werden können.

Das Thema muss sich in demselben Bereich AWS-Region wie Ihr Amazon S3 S3-Bucket befinden. Anweisungen zum Erstellen eines SNS Amazon-Themas finden Sie unter Erste Schritte mit Amazon SNS im Amazon Simple Notification Service Developer Guide und im Amazon SNS FAQ.

Bevor Sie das von Ihnen erstellte SNS Amazon-Thema als Ziel für Ereignisbenachrichtigungen verwenden können, benötigen Sie Folgendes:

- Der Amazon-Ressourcenname (ARN) für das SNS Amazon-Thema
- Ein gültiges SNS Amazon-Themenabonnement. Damit werden Themenabonnenten benachrichtigt, wenn eine Nachricht zu Ihrem SNS Amazon-Thema veröffentlicht wird.

SQSAmazon-Warteschlange

Amazon SQS bietet zuverlässige und skalierbare gehostete Warteschlangen zum Speichern von Nachrichten, während sie zwischen Computern übertragen werden. Sie können Amazon verwendenSQS, um beliebige Datenmengen zu übertragen, ohne dass andere Dienste immer verfügbar sein müssen. Sie können die SQS Amazon-Konsole verwenden, um eine SQS Amazon-Warteschlange zu erstellen, an die Ihre Benachrichtigungen gesendet werden können.

Die SQS Amazon-Warteschlange muss sich in derselben Warteschlange befinden AWS-Region wie Ihr Amazon S3-Bucket. Anweisungen zum Erstellen einer SQS Amazon-Warteschlange finden Sie unter Was ist Amazon Simple Queue Service und Erste Schritte mit Amazon SQS im Amazon Simple Queue Service Developer Guide.

Bevor Sie die SQS Amazon-Warteschlange als Ziel für Ereignisbenachrichtigungen verwenden können, benötigen Sie Folgendes:

- Der Amazon-Ressourcenname (ARN) für die SQS Amazon-Warteschlange

> **Note**
>
> Amazon Simple Queue Service FIFO (First-In-First-Out) -Warteschlangen werden nicht als Ziel für Amazon S3 S3-Ereignisbenachrichtigungen unterstützt. Um eine Benachrichtigung für ein Amazon S3 S3-Ereignis an eine SQS FIFO Amazon-Warteschlange zu senden, können Sie Amazon verwenden EventBridge. Weitere Informationen finden Sie unter Amazon aktivieren EventBridge.

Lambda-Funktion

Sie können AWS Lambda damit andere AWS Services mit benutzerdefinierter Logik erweitern oder Ihr eigenes Backend erstellen, das AWS skalierbar, leistungsstark und sicher arbeitet. Mit Lambda können Sie diskrete ereignisabhängige Anwendungen erstellen, die nur ausgeführt werden, wenn sie benötigt werden. Sie können damit auch diese Anwendungen automatisch von ein paar Anfragen pro Tag auf Tausende pro Sekunde skalieren.

Lambda kann benutzerdefinierten Code in Reaktion auf Amazon-S3-Bucket-Ereignisse ausführen. Sie laden Ihren benutzerdefinierten Code auf Lambda hoch und erstellen eine sogenannte Lambda-Funktion. Wenn Amazon S3 ein Ereignis eines bestimmten Typs erkennt, kann es das Ereignis

in Lambda veröffentlichen AWS Lambda und Ihre Funktion in Lambda aufrufen. Lambda führt als Antwort Ihre Funktion aus. Ein Ereignistyp, den er möglicherweise erkennt, ist beispielsweise ein objekterstelltes Ereignis.

Sie können die AWS Lambda Konsole verwenden, um eine Lambda-Funktion zu erstellen, die die AWS Infrastruktur verwendet, um den Code in Ihrem Namen auszuführen. Die Lambda-Funktion muss sich in derselben Region wie Ihr S3-Bucket befinden. Sie müssen auch den Namen oder den ARN einer Lambda-Funktion haben, um die Lambda-Funktion als Ziel für Ereignisbenachrichtigungen einzurichten.

> **Warning**
>
> Wenn Ihre Benachrichtigung in den selben Bucket schreibt, der die Benachrichtigung auslöst, kann dies zu einer Ausführungsschleife führen. Wenn der Bucket z. B. eine Lambda-Funktion immer dann auslöst, wenn ein Objekt hochgeladen wird, und die Funktion ein Objekt in den Bucket hochlädt, löst sich die Funktion indirekt selbst aus. Um dies zu vermeiden, verwenden Sie zwei Buckets oder konfigurieren Sie den Auslöser so, dass er nur für einen Präfix gilt, der für eingehende Objekte verwendet wird.
> Weitere Informationen und ein Beispiel für die Verwendung von Amazon S3-Benachrichtigungen mit AWS Lambda finden Sie unter AWS Lambda Using with Amazon S3 im AWS Lambda Developer Guide.

Amazon EventBridge

Amazon EventBridge ist ein serverloser Event-Bus, der Ereignisse von AWS Diensten empfängt. Sie können Regeln einrichten, um Ereignisse zuzuordnen und sie an Ziele wie einen AWS Service oder einen HTTP Endpunkt weiterzuleiten. Weitere Informationen finden Sie unter Was steht EventBridge im EventBridge Amazon-Benutzerhandbuch.

Im Gegensatz zu anderen Zielen können Sie EventBridge für einen Bucket Ereignisse, an die gesendet werden, entweder aktivieren oder deaktivieren. Wenn Sie die Zustellung aktivieren, werden alle Ereignisse an gesendet EventBridge. Darüber hinaus können Sie EventBridge Regeln verwenden, um Ereignisse an weitere Ziele weiterzuleiten.

Unterstützte Ereignistypen für SQSSNS, und Lambda

Amazon S3 kann Ereignisse der folgenden Typen veröffentlichen. Sie spezifizieren diese Ereignistypen in der Benachrichtigungskonfiguration.

Ereignistypen	Beschreibung
`s3:TestEvent`	Wenn eine Benachrichtigung aktiviert ist, veröffentlicht Amazon S3 eine Testbenachrichtigung. Damit soll sichergestellt werden, dass das Thema vorhanden ist und dass der Bucket-Eigentümer über die Berechtigung verfügt, das angegebene Thema zu veröffentlichen. Wenn das Aktivieren der Benachrichtigung fehlschlägt, erhalten Sie keine Testbenachrichtigung.

`s3:ObjectCreated:*`
 	
`s3:ObjectCreated:Put`
 	
`s3:ObjectCreated:Post`
 	
`s3:ObjectCreated:Copy`
 	
`s3:ObjectCreated:CompleteMultipartUpload`
	Amazon S3 API S3-Operationen wie `PUT`, `POST`, und `COPY` können ein Objekt erstellen. Mit diesen Ereignistypen können Sie Benachrichtigungen aktivieren, wenn ein Objekt mithilfe eines bestimmten API Vorgangs erstellt wird. Alternativ können Sie den `s3:ObjectCreated:*` Ereignistyp verwenden, um Benachrichtigungen anzufordern, unabhängig davonAPI, mit welchem Ereignistyp ein Objekt erstellt wurde.
	
	`s3:ObjectCreated:CompleteMultipartUpload` schließt Objekte ein, die erstellt wurden mit UploadPartCopyfür Kopiervorgänge.

`s3:ObjectRemoved:*`
 	
`s3:ObjectRemoved:Delete`
 	
`s3:ObjectRemoved:DeleteMarkerCreated`
	Mithilfe der `ObjectRemoved` Ereignistypen können Sie Benachrichtigungen aktivieren, wenn ein Objekt oder ein Stapel von Objekten aus einem Bucket entfernt wird.
	
	Sie können eine Benachrichtigung anfordern, wenn ein Objekt gelöscht wird, oder wenn ein versionsfähiges Objekt permanent gelöscht wird. Dazu wird der Ereignistyp `s3:ObjectRemoved:Delete` verwendet. Alternativ können Sie eine Benachrichtigung anfordern, wenn eine Löschmarkierung für ein versioniertes Objekt mit `s3:ObjectRemoved:DeleteMarkerCreated` erstellt wird. Eine entsprechende Anleitung zum Löschen von versionierten Objekten finden Sie unter Löschen von Objekten aus einem versioning-fähigen Bucket. Sie können

Ereignistypen	Beschreibung
	auch eine Wildcard `s3:ObjectRemoved:*` verwenden , um bei jedem Löschen eines Objekts eine Benachric htigung anzufordern. Diese Ereignisbenachrichtigungen warnen Sie nicht vor automatischen Löschungen aufgrund von Lebenszyk luskonfigurationen oder fehlgeschlagenen Operationen.

`s3:ObjectRestore:*`
 	
`s3:ObjectRestore:Post`
 	
`s3:ObjectRestore:C ompleted`
 	
`s3:ObjectRestore:Delete`
	Mithilfe der `ObjectRestore` Ereignistypen können Sie Benachrichtigungen über die Initiierung und den Abschluss von Ereignissen erhalten, wenn Objekte aus der Speicherk lasse S3 Glacier Flexible Retrieval, der Speicherklasse S3 Glacier Deep Archive, der Stufe S3 Intelligent-Tiering Archive Access und der Stufe S3 Intelligent-Tiering Deep Archive Access wiederhergestellt werden. Sie können auch Benachrichtigungen erhalten, wann die wiederhergestellte Kopie eines Objekts abläuft.
	
	Der `s3:ObjectRestore:Post` -Ereignistyp informiert Sie über die Initiierung von Objektwiederherstellungen. Der `s3:ObjectRestore:Completed` -Ereignistyp informier t Sie über den Abschluss von Wiederherstellungen. Der `s3:ObjectRestore:Delete` -Ereignistyp benachric htigt Sie, wenn die temporäre Kopie eines wiederher gestellten Objekts abläuft.

`s3:ReducedRedundan cyLostObject`	Sie erhalten dieses Benachrichtigungsereignis, wenn Amazon S3 feststellt, dass ein Objekt der RRS Speicherk lasse verloren gegangen ist.

Ereignistypen	Beschreibung

`s3:Replication:*`
 	
`s3:Replication:OperationFailedReplication`
 	
`s3:Replication:OperationMissedThreshold`
 	
`s3:Replication:OperationReplicatedAfterThreshold`
 	
`s3:Replication:OperationNotTracked`
	Mithilfe der `Replication` Ereignistypen können Sie Benachrichtigungen für Replikationskonfigurationen erhalten, für die S3-Replikationsmetriken oder S3 Replication Time Control (S3RTC) aktiviert sind. Sie können den minute-by-minute Fortschritt von Replikationsereignissen überwachen, indem Sie ausstehende Byte, ausstehende Vorgänge und die Replikationslatenz verfolgen. Informationen zu Replikationsmetriken finden Sie unter Überwachung der Replikation mit Metriken, Ereignisbenachrichtigungen und Status.
	
	• Der `s3:Replication:OperationFailedReplication` -Ereignistyp benachrichtigt Sie, wenn ein Objekt, das für die Replikation berechtigt war, nicht repliziert werden konnte.
	• Der `s3:Replication:OperationMissedThreshold` Ereignistyp benachrichtigt Sie, wenn ein Objekt, das für die Replikation in Frage kam und S3 verwendet, den 15-minütigen Schwellenwert für die Replikation RTC überschreitet.
	• Der `s3:Replication:OperationReplicatedAfterThreshold` Ereignistyp benachrichtigt Sie, wenn ein Objekt, das für die Replikation in Frage kam und S3 verwendet, nach dem RTC 15-Minuten-Schwellenwert repliziert wird.
	• Der `s3:Replication:OperationNotTracked` Ereignistyp benachrichtigt Sie, wenn ein Objekt, das für die Live-Replikation in Frage kam (entweder Replikation in derselben Region [SRR] oder regionsübergreifende Replikation [CRR]), nicht mehr anhand von Replikationsmetriken verfolgt wird.

Ereignistypen	Beschreibung

`s3:LifecycleExpiration:*`
 	
`s3:LifecycleExpiration:Delete`
 	
`s3:LifecycleExpiration:DeleteMarkerCreated`
	Mithilfe der `LifecycleExpiration` Ereignistypen können Sie eine Benachrichtigung erhalten, wenn Amazon S3 ein Objekt auf der Grundlage Ihrer S3-Lifecycle-Konfiguration löscht.
	
	Der Ereignistyp `s3:LifecycleExpiration:Delete` benachrichtigt Sie, wenn ein Objekt in einem unversionierten Bucket gelöscht wird. Es benachrichtigt Sie auch, wenn eine Objektversion durch eine S3-Lebenszyklus-Konfiguration dauerhaft gelöscht wird. Der Ereignistyp `s3:LifecycleExpiration:DeleteMarkerCreated` benachrichtigt Sie, wenn S3 Lebenszyklus eine Löschmarke erstellt, wenn eine aktuelle Version eines Objekts im versionierten Bucket gelöscht wird.

`s3:LifecycleTransition`	Sie erhalten dieses Benachrichtigungsereignis, wenn ein Objekt von einer S3-Lebenszykluskonfiguration in eine andere Amazon-S3-Speicherklasse überführt wird.
`s3:IntelligentTiering`	Sie erhalten dieses Benachrichtigungsereignis, wenn ein Objekt innerhalb der Speicherklasse S3 Intelligent-Tiering in die Stufe Archive Access oder Deep Archive Access verschoben wird.

`s3:ObjectTagging:*`
 	
`s3:ObjectTagging:Put`
 	
`s3:ObjectTagging:Delete`
	Mithilfe der `ObjectTagging` Ereignistypen können Sie Benachrichtigungen aktivieren, wenn ein Objekt-Tag hinzugefügt oder aus einem Objekt gelöscht wird.
	
	Der `s3:ObjectTagging:Put` Ereignistyp benachrichtigt Sie, wenn sich ein Tag PUT auf einem Objekt befindet oder wenn ein vorhandenes Tag aktualisiert wird. Der `s3:ObjectTagging:Delete` -Ereignistyp benachrichtigt Sie, wenn ein Tag aus einem Objekt entfernt wird.

Ereignistypen	Beschreibung
`s3:ObjectAcl:Put`	Sie erhalten dieses Benachrichtigungsereignis, wenn ACL sich ein Objekt PUT auf einem Objekt befindet oder wenn ein vorhandenes geändert ACL wird. Ein Ereignis wird nicht generiert, wenn eine Anfrage zu keiner Änderung an einem Objekt führtACL.

Unterstützte Ereignistypen für Amazon EventBridge

Eine Liste der Ereignistypen, die Amazon S3 an Amazon sendet EventBridge, finden Sie unterVerwenden EventBridge.

Bestellung von Veranstaltungen und doppelte Ereignisse

Amazon S3 Event Notifications ist so konzipiert, dass Benachrichtigungen mindestens einmal zugestellt werden, aber es kann nicht garantiert werden, dass sie in derselben Reihenfolge ankommen, in der die Ereignisse eingetreten sind. In seltenen Fällen kann der Wiederholungsmechanismus von Amazon S3 zu doppelten S3-Ereignisbenachrichtigungen für dasselbe Objektereignis führen. Weitere Informationen zum Umgang mit doppelten oder nicht in der richtigen Reihenfolge befindlichen Ereignissen finden Sie unter Verwaltung der Bestellung von Veranstaltungen und doppelten Ereignissen mit Amazon S3 S3-Ereignisbenachrichtigungen im AWS Storage-Blog.

Verwendung von Amazon SQSSNS, Amazon und Lambda

Die Aktivierung von Benachrichtigungen erfolgt auf Bucket-Ebene. Sie speichern Informationen zur Benachrichtigungskonfiguration in der Benachrichtigungen-Unterressourcen die einem Bucket zugeordnet sind. Nachdem Sie die Bucket-Benachrichtigungskonfiguration erstellt oder geändert haben, dauert es normalerweise etwa fünf Minuten, bis die Änderungen wirksam werden. Ein `s3:TestEvent` tritt auf, wenn die Benachrichtigung zum ersten Mal aktiviert wird. Sie können jede der folgenden Methoden verwenden, um die Benachrichtigungskonfiguration zu verwalten:

- Verwenden der Amazon-S3-Konsole – Die Benutzeroberfläche der Konsole ermöglicht Ihnen, eine Benachrichtigungskonfiguration für einen Bucket einzurichten, ohne Code schreiben zu müssen. Weitere Informationen finden Sie unter Aktivieren und Konfigurieren von Ereignis-Benachrichtigungen mit der Amazon-S3-Konsole.

- Programmgesteuert mithilfe von AWS SDKs — Intern werden sowohl die Konsole als auch der Amazon S3 SDKs aufgerufen, um REST API Benachrichtigungs-Unterressourcen zu verwalten, die dem Bucket zugeordnet sind. Beispiele für Benachrichtigungskonfigurationen, die verwendet werden, finden Sie unter. AWS SDK Exemplarische Vorgehensweise: Konfiguration eines Buckets für Benachrichtigungen (SNSThema oder SQS Warteschlange)

  > **Note**
  >
  > Sie können die Amazon S3 REST API S3-Anrufe auch direkt von Ihrem Code aus tätigen. Dies kann jedoch umständlich sein, da Sie dazu Code schreiben müssen, um Ihre Anforderungen zu authentifizieren.

Unabhängig von der Methode, die Sie verwenden, speichert Amazon S3 die Benachrichtigungskonfiguration wie XML in der Benachrichtigungs-Subressource, die einem Bucket zugeordnet ist. Weitere Informationen zu Bucket-Subressourcen finden Sie unter Optionen für die Bucket-Konfiguration.

> **Note**
>
> Wenn Sie aufgrund gelöschter Ziele mehrere fehlgeschlagene Ereignisbenachrichtigungen haben, erhalten Sie möglicherweise die Meldung Die folgenden Zielkonfigurationen konnten nicht überprüft werden, wenn Sie versuchen, sie zu löschen. Sie können dieses Problem in der S3-Konsole beheben, indem Sie alle fehlgeschlagenen Benachrichtigungen gleichzeitig löschen.

Themen

- Erteilen von Berechtigungen zur Veröffentlichung von Ereignis-Benachrichtigungsmeldungen an einem Ziel
- Aktivieren und Konfigurieren von Ereignis-Benachrichtigungen mit der Amazon-S3-Konsole
- Programmgesteuerte Konfiguration von Ereignis-Benachrichtigungen
- Exemplarische Vorgehensweise: Konfiguration eines Buckets für Benachrichtigungen (SNSThema oder SQS Warteschlange)
- Konfigurieren von Ereignis-Benachrichtigungen mithilfe der Namensfilterung
- Struktur von Ereignismeldungen

Erteilen von Berechtigungen zur Veröffentlichung von Ereignis-Benachrichtigungsmeldungen an einem Ziel

Sie müssen dem Amazon S3 S3-Principal die erforderlichen Berechtigungen zum Aufrufen der relevanten Funktionen API zur Veröffentlichung von Nachrichten in einem SNS Thema, einer SQS Warteschlange oder einer Lambda-Funktion gewähren. So kann Amazon S3 Ereignisbenachrichtigungsmeldungen an einem Ziel veröffentlichen.

Informationen zur Fehlerbehebung beim Veröffentlichen von Ereignisbenachrichtigungen in einem Ziel finden Sie unter Fehlerbehebung bei der Veröffentlichung von Amazon-S3-Ereignisbenachrichtigungen in einem Thema von Amazon Simple Notification Service.

Themen

- Erteilen von Berechtigungen zum Aufrufen einer Funktion AWS Lambda
- Erteilen von Berechtigungen zum Veröffentlichen von Nachrichten zu einem SNS Thema oder einer SQS Warteschlange

Erteilen von Berechtigungen zum Aufrufen einer Funktion AWS Lambda

Amazon S3 veröffentlicht Ereignismeldungen, AWS Lambda indem es eine Lambda-Funktion aufruft und die Ereignisnachricht als Argument bereitstellt.

Wenn Sie die Amazon-S3-Konsole verwenden, um Ereignisbenachrichtigungen in einem Amazon-S3-Bucket für eine Lambda-Funktion zu konfigurieren, richtet die Konsole die erforderlichen Berechtigungen für die Lambda-Funktion ein. Dies ist so, dass Amazon S3 über Berechtigungen verfügt, die Funktion aus dem Bucket aufzurufen. Weitere Informationen finden Sie unter Aktivieren und Konfigurieren von Ereignis-Benachrichtigungen mit der Amazon-S3-Konsole.

Sie können Amazon S3 S3-Berechtigungen auch erteilen AWS Lambda , um Ihre Lambda-Funktion aufzurufen. Weitere Informationen finden Sie unter Tutorial: Using AWS Lambda with Amazon S3 im AWS Lambda Developer Guide.

Erteilen von Berechtigungen zum Veröffentlichen von Nachrichten zu einem SNS Thema oder einer SQS Warteschlange

Um Amazon S3 S3-Berechtigungen zur Veröffentlichung von Nachrichten im SNS Thema oder in der SQS Warteschlange zu gewähren, fügen Sie dem SNS Zielthema oder der SQS Warteschlange eine AWS Identity and Access Management (IAM) -Richtlinie hinzu.

Ein Beispiel dafür, wie Sie eine Richtlinie an ein SNS Thema oder eine SQS Warteschlange anhängen, finden Sie unterExemplarische Vorgehensweise: Konfiguration eines Buckets für Benachrichtigungen (SNSThema oder SQS Warteschlange). Weitere Informationen über Berechtigungen finden Sie in den folgenden Themen:

- Beispielfälle für Amazon SNS Access Control im Amazon Simple Notification Service Developer Guide
- Identitäts- und Zugriffsmanagement in Amazon SQS im Amazon Simple Queue Service Developer Guide

IAMRichtlinie für ein SNS Zielthema

Im Folgenden finden Sie ein Beispiel für eine Richtlinie AWS Identity and Access Management (IAM), die Sie dem SNS Zielthema zuordnen. Anweisungen dazu, wie Sie mithilfe dieser Richtlinie ein SNS Amazon-Zielthema für Ereignisbenachrichtigungen einrichten können, finden Sie unterExemplarische Vorgehensweise: Konfiguration eines Buckets für Benachrichtigungen (SNSThema oder SQS Warteschlange).

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "Example SNS topic policy",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "SNS:Publish"
            ],
            "Resource": "SNS-topic-ARN",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:*:*:amzn-s3-demo-bucket"
                },
                "StringEquals": {
                    "aws:SourceAccount": "bucket-owner-account-id"
                }
            }
        }
```

```
    ]
}
```

IAMRichtlinie für eine SQS Zielwarteschlange

Im Folgenden finden Sie ein Beispiel IAM für eine Richtlinie, die Sie an die SQS Zielwarteschlange anhängen. Anweisungen dazu, wie Sie diese Richtlinie verwenden können, um eine SQS Amazon-Zielwarteschlange für Ereignisbenachrichtigungen einzurichten, finden Sie unterExemplarische Vorgehensweise: Konfiguration eines Buckets für Benachrichtigungen (SNSThema oder SQS Warteschlange).

Um diese Richtlinie verwenden zu können, müssen Sie die SQS Amazon-WarteschlangeARN, den Bucket-Namen und die AWS-Konto ID des Bucket-Besitzers aktualisieren.

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "example-statement-ID",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "SQS:SendMessage"
            ],
            "Resource": "arn:aws:sqs:Region:account-id:queue-name",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:*:*:awsexamplebucket1"
                },
                "StringEquals": {
                    "aws:SourceAccount": "bucket-owner-account-id"
                }
            }
        }
    ]
}
```

Sowohl für die Amazon SNS - als auch für SQS IAM die Amazon-Richtlinien können Sie anstelle der `StringLike` Bedingung die `ArnLike` Bedingung in der Richtlinie angeben.

Wenn diese `ArnLike` Option verwendet wird, müssen die Partition, der Service, die Account-ID, der Ressourcentyp und der Teil der Ressourcen-ID exakt mit dem im ARN Anforderungskontext übereinstimmen. ARN Nur die Region und der Ressourcenpfad lassen einen teilweisen Abgleich zu.

Wenn anstelle von verwendet `StringLike` wird`ArnLike`, ignoriert der Abgleich die ARN Struktur und ermöglicht einen teilweisen Abgleich, unabhängig davon, welcher Teil mit einem Platzhalter versehen wurde. Weitere Informationen finden Sie unter IAMJSONRichtlinienelemente im IAM Benutzerhandbuch.

```
"Condition": {
  "StringLike": { "aws:SourceArn": "arn:aws:s3:*:*:amzn-s3-demo-bucket" }
  }
```

AWS KMS wichtige Richtlinie

Wenn die SQS Warteschlange oder SNS Themen mit einem AWS Key Management Service (AWS KMS) vom Kunden verwalteten Schlüssel verschlüsselt sind, müssen Sie dem Amazon S3 S3-Service Principal die Erlaubnis erteilen, mit den verschlüsselten Themen oder der Warteschlange zu arbeiten. Um dem Amazon-S3-Service-Prinzipal die Berechtigung zu erteilen, fügen Sie der Schlüsselrichtlinie für den vom Kunden verwalteten Schlüssel die folgende Anweisung hinzu.

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "example-statement-ID",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

Weitere Informationen zu AWS KMS wichtigen Richtlinien finden Sie unter Verwenden von Schlüsselrichtlinien AWS KMS im AWS Key Management Service Entwicklerhandbuch.

Weitere Informationen zur Verwendung der serverseitigen Verschlüsselung mit AWS KMS für Amazon SQS und Amazon SNS finden Sie im Folgenden:

- Schlüsselverwaltung im Amazon Simple Notification Service-Entwicklerhandbuch.
- Schlüsselverwaltung im Amazon Simple Queue Service-Entwicklerhandbuch.
- Verschlüsseln von Nachrichten, die auf Amazon veröffentlicht wurden, SNS mit AWS KMS im AWS Compute-Blog.

Aktivieren und Konfigurieren von Ereignis-Benachrichtigungen mit der Amazon-S3-Konsole

Sie können bestimmte Amazon-S3-Bucket-Ereignisse aktivieren, um eine Benachrichtigungsmeldung an ein Ziel zu senden, wenn das Ereignis auftritt. In diesem Abschnitt erfahren Sie, wie Sie mit der Amazon-S3-Konsole Ereignis-Benachrichtigungen aktivieren können. Informationen zur Verwendung von Ereignisbenachrichtigungen mit dem AWS SDKs und dem Amazon S3 REST APIs finden Sie unterProgrammgesteuerte Konfiguration von Ereignis-Benachrichtigungen.

Voraussetzungen: Bevor Sie Ereignis-Benachrichtigungen für Ihren Bucket aktivieren können, müssen Sie einen der Zieltypen einrichten und dann die Berechtigungen konfigurieren. Weitere Informationen erhalten Sie unter Unterstützte Ereignisziele und Erteilen von Berechtigungen zur Veröffentlichung von Ereignis-Benachrichtigungsmeldungen an einem Ziel.

> **Note**
>
> Amazon Simple Queue Service FIFO (First-In-First-Out) -Warteschlangen werden nicht als Ziel für Amazon S3 S3-Ereignisbenachrichtigungen unterstützt. Um eine Benachrichtigung für ein Amazon S3 S3-Ereignis an eine SQS FIFO Amazon-Warteschlange zu senden, können Sie Amazon verwenden EventBridge. Weitere Informationen finden Sie unter Amazon aktivieren EventBridge.

Themen

- Amazon- SNSSQS, Amazon- oder Lambda-Benachrichtigungen über die Amazon S3 S3-Konsole aktivieren

Amazon- SNSSQS, Amazon- oder Lambda-Benachrichtigungen über die Amazon S3 S3-Konsole aktivieren

Ereignis-Benachrichtigungen für einen S3-Bucket aktivieren und konfigurieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie Ereignisse aktivieren möchten.
3. Wählen Sie Properties (Eigenschaften).
4. Navigieren Sie zum Abschnitt Event Notifications (Ereignis-Benachrichtigungen) und wählen Sie Create event notification (Ereignis-Benachrichtigung erstellen)
5. Geben Sie im Abschnitt General configuration (Allgemeine Konfiguration) den beschreibenden Ereignisnamen für Ihre Ereignis-Benachrichtigung an. Optional können Sie auch ein Präfix und ein Suffix angeben, um die Benachrichtigungen auf Objekte mit Schlüsseln zu beschränken, die mit den angegebenen Zeichen enden.

   a. Geben Sie eine Beschreibung für den Namen der Veranstaltung ein.

      Wenn Sie keinen Namen eingeben, wird eine global eindeutige Kennung (GUID) generiert und für den Namen verwendet.

   b. (Optional) Um Ereignisbenachrichtigungen nach Präfix zu filtern, geben Sie ein Präfix ein.

      Sie können beispielsweise einen Präfix-Filter einrichten, damit Sie nur Benachrichtigungen erhalten, wenn Dateien zu einem bestimmten Ordner (z. B.) hinzugefügt wer, `images/`).

   c. (Optional) Um Ereignisbenachrichtigungen nach Suffix zu filtern, geben Sie ein Suffix ein.

      Weitere Informationen finden Sie unter Konfigurieren von Ereignis-Benachrichtigungen mithilfe der Namensfilterung.
6. Wählen Sie im Abschnitt Event types (Ereignistypen) einen oder mehrere Ereignistypen aus, für die Sie Benachrichtigungen erhalten möchten.

   Eine Liste der verschiedenen Ereignistypen finden Sie unter Unterstützte Ereignistypen für SQSSNS, und Lambda.
7. Wählen Sie im Abschnitt Destination (Ziel) das Ziel für Ereignis-Benachrichtigungen aus.

> **Note**
>
> Bevor Sie Ereignisbenachrichtigungen veröffentlichen können, müssen Sie dem Amazon S3-Principal die erforderlichen Berechtigungen erteilen, um die entsprechenden Personen aufzurufenAPI. Auf diese Weise können Benachrichtigungen für eine Lambda-Funktion, ein SNS Thema oder eine SQS Warteschlange veröffentlicht werden.

a. Wählen Sie den Zieltyp aus: Lambda-Funktion, SNSThema oder SQSWarteschlange.

b. Nachdem Sie Ihren Zieltyp ausgewählt haben, wählen Sie eine Funktion, ein Thema oder eine Warteschlange aus der Liste aus.

c. Oder, wenn Sie lieber einen Amazon-Ressourcennamen (ARN) angeben möchten, wählen Sie Enter ARN und geben Sie den einARN.

Weitere Informationen finden Sie unter Unterstützte Ereignisziele.

8. Wählen Sie Save changes (Änderungen speichern) und Amazon S3 sendet eine Testnachricht an das Ziel für Ereignis-Benachrichtigungen.

Programmgesteuerte Konfiguration von Ereignis-Benachrichtigungen

Standardmäßig sind für keinen Ereignistyp Benachrichtigungen aktiviert. Aus diesem Grund speichert die Benachrichtigung-Unterressourcen anfänglich eine leere Konfiguration.

```
<NotificationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
</NotificationConfiguration>
```

Um Benachrichtigungen für Ereignisse bestimmter Typen zu aktivieren, ersetzen Sie die durch die XML entsprechende Konfiguration, die die Ereignistypen, die Amazon S3 veröffentlichen soll, und das Ziel, an dem die Ereignisse veröffentlicht werden sollen, identifiziert. Für jedes Ziel fügen Sie eine entsprechende XML Konfiguration hinzu.

Um Ereignisnachrichten in einer SQS Warteschlange zu veröffentlichen

Um eine SQS Warteschlange als Benachrichtigungsziel für einen oder mehrere Ereignistypen festzulegen, fügen Sie die hinzu`QueueConfiguration`.

```
<NotificationConfiguration>
  <QueueConfiguration>
    <Id>optional-id-string</Id>
    <Queue>sqs-queue-arn</Queue>
    <Event>event-type</Event>
    <Event>event-type</Event>
     ...
  </QueueConfiguration>
   ...
</NotificationConfiguration>
```

Um Ereignisnachrichten zu einem SNS Thema zu veröffentlichen

Um ein SNS Thema als Benachrichtigungsziel für bestimmte Ereignistypen festzulegen, fügen Sie den hinzu`TopicConfiguration`.

```
<NotificationConfiguration>
  <TopicConfiguration>
     <Id>optional-id-string</Id>
     <Topic>sns-topic-arn</Topic>
     <Event>event-type</Event>
     <Event>event-type</Event>
      ...
  </TopicConfiguration>
   ...
</NotificationConfiguration>
```

Um die AWS Lambda Funktion aufzurufen und eine Ereignisnachricht als Argument bereitzustellen

Um eine Lambda-Funktion als Benachrichtigungsziel für bestimmte Ereignistypen festzulegen, fügen Sie `CloudFunctionConfiguration` hinzu.

```
<NotificationConfiguration>
  <CloudFunctionConfiguration>
     <Id>optional-id-string</Id>
     <CloudFunction>cloud-function-arn</CloudFunction>
     <Event>event-type</Event>
     <Event>event-type</Event>
      ...
  </CloudFunctionConfiguration>
   ...
</NotificationConfiguration>
```

So entfernen Sie alle in einem Bucket konfigurierten Benachrichtigungen

Um alle für einen Bucket konfigurierten Benachrichtigungen zu entfernen, speichern Sie ein leeres `<NotificationConfiguration/>`-Element in der notification-Subressource.

Wenn Amazon S3 ein Ereignis des spezifischen Typs erkennt, veröffentlicht es eine Meldung mit der Ereignisinformation. Weitere Informationen finden Sie unter Struktur von Ereignismeldungen.

Weitere Informationen zur Konfiguration von Ereignis-Benachrichtigungen finden Sie in den folgenden Themen:

- Exemplarische Vorgehensweise: Konfiguration eines Buckets für Benachrichtigungen (SNSThema oder SQS Warteschlange).
- Konfigurieren von Ereignis-Benachrichtigungen mithilfe der Namensfilterung

Exemplarische Vorgehensweise: Konfiguration eines Buckets für Benachrichtigungen (SNSThema oder SQS Warteschlange)

Sie können Amazon S3 S3-Benachrichtigungen mit Amazon Simple Notification Service (AmazonSNS) oder Amazon Simple Queue Service (AmazonSQS) erhalten. In dieser exemplarischen Vorgehensweise fügen Sie Ihrem Bucket mithilfe eines SNS Amazon-Themas und einer SQS Amazon-Warteschlange eine Benachrichtigungskonfiguration hinzu.

> **Note**
>
> Amazon Simple Queue Service FIFO (First-In-First-Out) -Warteschlangen werden nicht als Ziel für Amazon S3 S3-Ereignisbenachrichtigungen unterstützt. Um eine Benachrichtigung für ein Amazon S3 S3-Ereignis an eine SQS FIFO Amazon-Warteschlange zu senden, können Sie Amazon verwenden EventBridge. Weitere Informationen finden Sie unter Amazon aktivieren EventBridge.

Themen

- Walkthrough-Übersicht
- Schritt 1: Erstellen Sie eine SQS Amazon-Warteschlange
- Schritt 2: Erstellen Sie ein SNS Amazon-Thema
- Schritt 3: Hinzufügen einer Benachrichtigungskonfiguration zu Ihrem Bucket

- Schritt 4: Testen der Einrichtung

Walkthrough-Übersicht

Dieser Walkhrough hilft Ihnen bei folgenden Aufgaben:

- Veröffentlichen Sie Ereignisse dieses `s3:ObjectCreated:*` Typs in einer SQS Amazon-Warteschlange.
- Veröffentlichen Sie Ereignisse dieses `s3:ReducedRedundancyLostObject` Typs zu einem SNS Amazon-Thema.

Weitere Informationen zur Benachrichtigungskonfiguration finden Sie unter Verwendung von Amazon SQSSNS, Amazon und Lambda

Alle diese Schritte können Sie auf der Konsole erledigen, ohne Code schreiben zu müssen. Darüber hinaus Codebeispiele AWS SDKs für Java und. NETwerden ebenfalls bereitgestellt, damit Sie Benachrichtigungskonfigurationen programmgesteuert hinzufügen können.

In dieser Vorgehensweise werden die folgenden Schritte beschrieben:

1. Erstellen Sie eine SQS Amazon-Warteschlange.

   Erstellen Sie mit der SQS Amazon-Konsole eine SQS Warteschlange. Sie können auf alle Meldungen zugreifen, die Amazon S3 programmgesteuert an die Warteschlange sendet. Für diese schrittweise Anleitung überprüfen Sie die Benachrichtigungsmitteilungen in der Konsole.

   Sie ordnen der Warteschlange eine Zugriffsrichtlinie zu, um Amazon S3 die Berechtigung zu erteilen, Meldungen zu veröffentlichen.
2. Erstellen Sie ein SNS Amazon-Thema.

   Erstellen Sie mit der SNS Amazon-Konsole ein SNS Thema und abonnieren Sie das Thema. Auf diese Weise werden alle darin veröffentlichten Ereignisse an Sie weitergegeben. Sie geben als Kommunikationsprotokoll E-Mail an. Nachdem Sie ein Thema erstellt haben, SNS sendet Amazon eine E-Mail. Sie verwenden den Link in der E-Mail, um das Abonnement des Themas zu bestätigen.

   Sie ordnen dem Thema eine Zugriffsrichtlinie zu, um Amazon S3 die Berechtigung zu erteilen, Meldungen zu veröffentlichen.
3. Fügen Sie einem Bucket eine Benachrichtigungskonfiguration hinzu.

Schritt 1: Erstellen Sie eine SQS Amazon-Warteschlange

Folgen Sie den Schritten, um eine Amazon Simple Queue Service (AmazonSQS) -Warteschlange zu erstellen und zu abonnieren.

1. Erstellen Sie mit der SQS Amazon-Konsole eine Warteschlange. Anweisungen finden Sie unter Erste Schritte mit Amazon SQS im Amazon Simple Queue Service Developer Guide.
2. Ersetzen Sie die der Warteschlange zugeordnete Zugriffsrichtlinie durch die folgende Richtlinie.

   a. Wählen Sie in der SQS Amazon-Konsole in der Warteschlangenliste den Namen der Warteschlange aus.

   b. Wählen Sie im Tab Zugriffsrichtlinie die Option Bearbeiten aus.

   c. Ersetzen Sie die Zugriffsrichtlinie, die der Warteschlange angefügt ist. Geben Sie darin Ihren Amazon-Namen SQSARN, den Quell-Bucket-Namen und die Konto-ID des Bucket-Besitzers an.

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "example-statement-ID",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "SQS:SendMessage"
            ],
            "Resource": "SQS-queue-ARN",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:*:*:awsexamplebucket1"
                },
                "StringEquals": {
                    "aws:SourceAccount": "bucket-owner-account-id"
                }
            }
        }
    ]
}
```

d. Wählen Sie Save (Speichern) aus.

3. (Optional) Wenn für die SQS Amazon-Warteschlange oder das SNS Amazon-Thema serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) aktiviert ist, fügen Sie dem zugehörigen, vom Kunden verwalteten Schlüssel für symmetrische Verschlüsselung die folgende Richtlinie hinzu.

   Sie müssen die Richtlinie zu einem vom Kunden verwalteten Schlüssel hinzufügen, da Sie den AWS verwalteten Schlüssel für Amazon SQS oder Amazon nicht ändern könnenSNS.

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "example-statement-ID",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "kms:GenerateDataKey",
                "kms:Decrypt"
            ],
            "Resource": "*"
        }
    ]
}
```

   Weitere Informationen zur Verwendung von SSE for Amazon SQS und Amazon SNS with AWS KMS finden Sie im Folgenden:

   - Schlüsselverwaltung im Amazon Simple Notification Service-Entwicklerhandbuch.
   - Schlüsselverwaltung im Amazon Simple Queue Service-Entwicklerhandbuch.

4. Beachten Sie die WarteschlangeARN.

   Die SQS Warteschlange, die Sie erstellt haben, ist eine weitere Ressource in Ihrer AWS-Konto. Es hat einen eindeutigen Amazon-Ressourcennamen (ARN). Das benötigen Sie ARN im nächsten Schritt. ARNDas hat das folgende Format:

```
arn:aws:sqs:aws-region:account-id:queue-name
```

Schritt 2: Erstellen Sie ein SNS Amazon-Thema

Folgen Sie den Schritten, um ein SNS Amazon-Thema zu erstellen und zu abonnieren.

1. Erstellen Sie mit der SNS Amazon-Konsole ein Thema. Anweisungen finden Sie unter Erstellen eines SNS Amazon-Themas im Amazon Simple Notification Service Developer Guide.
2. Abonnieren Sie das Thema. Für diese Übung geben Sie email als Kommunikationsprotokoll an. Eine Anleitung finden Sie unter SNSAmazon-Thema abonnieren im Amazon Simple Notification Service Developer Guide.

   Sie erhalten eine E-Mail, in der Sie aufgefordert werden, das Abonnement des Themas zu bestätigen. Bestätigen Sie das Abonnement.
3. Ersetzen Sie die dem Thema zugeordnete Zugriffsrichtlinie durch die folgende Richtlinie. Geben Sie darin Ihr SNS ThemaARN, Ihren Bucket-Namen und die Konto-ID des Bucket-Besitzers ein.

```
{
    "Version": "2012-10-17",
    "Id": "example-ID",
    "Statement": [
        {
            "Sid": "Example SNS topic policy",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": [
                "SNS:Publish"
            ],
            "Resource": "SNS-topic-ARN",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:*:*:amzn-s3-demo-bucket"
                },
                "StringEquals": {
                    "aws:SourceAccount": "bucket-owner-account-id"
                }
            }
```

```
        }
    ]
}
```

4. Notieren Sie sich das ThemaARN.

   Das SNS Thema, das Sie erstellt haben, ist eine weitere Ressource in Ihrem AWS-Konto, und es hat eine einzigartigeARN. Sie werden dies ARN im nächsten Schritt benötigen. Das ARN wird das folgende Format haben:

   ```
   arn:aws:sns:aws-region:account-id:topic-name
   ```

Schritt 3: Hinzufügen einer Benachrichtigungskonfiguration zu Ihrem Bucket

Sie können Bucket-Benachrichtigungen entweder mithilfe der Amazon S3 S3-Konsole oder programmgesteuert mithilfe von aktivieren. AWS SDKs Wählen Sie eine der Optionen für die Konfiguration von Benachrichtigungen über Ihren Bucket. Dieser Abschnitt enthält Codebeispiele mit dem AWS SDKs für Java und. NET.

Option A: Aktivieren von Benachrichtigungen über einen Bucket unter Verwendung der Konsole

Fügen Sie mithilfe der Amazon-S3-Konsole eine Benachrichtigungskonfiguration hinzu, die Amazon S3 zu Folgendem auffordert:

- Veröffentlichen Sie Ereignisse des Typs „Alle Objekte erstellen“ in Ihrer SQS Amazon-Warteschlange.
- Veröffentlichen Sie Ereignisse des Typs Object in RRS Lost in Ihrem SNS Amazon-Thema.

Nachdem Sie die Benachrichtigungskonfiguration gespeichert haben, veröffentlicht Amazon S3 eine Testmeldung, die Sie per E-Mail erhalten.

Detaillierte Anweisungen finden Sie unter Aktivieren und Konfigurieren von Ereignis-Benachrichtigungen mit der Amazon-S3-Konsole.

Option B: Aktivieren Sie Benachrichtigungen für einen Bucket mithilfe der AWS SDKs

.NET

Das folgende C#-Codebeispiel zeigt ein vollständiges Listing, das einem Bucket eine Benachrichtigungskonfiguration hinzufügt. Sie müssen den Code aktualisieren und Ihren Bucket-

Namen und Ihr SNS Thema angebenARN. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem AWS SDK for. NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class EnableNotificationsTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string snsTopic = "*** SNS topic ARN ***";
        private const string sqsQueue = "*** SQS topic ARN ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;

        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            EnableNotificationAsync().Wait();
        }

        static async Task EnableNotificationAsync()
        {
            try
            {
               PutBucketNotificationRequest request = new
 PutBucketNotificationRequest
                {
                    BucketName = bucketName
                };

                TopicConfiguration c = new TopicConfiguration
                {
                    Events = new List<EventType> { EventType.ObjectCreatedCopy },
                    Topic = snsTopic
```

```
                };
                request.TopicConfigurations = new List<TopicConfiguration>();
                request.TopicConfigurations.Add(c);
                request.QueueConfigurations = new List<QueueConfiguration>();
                request.QueueConfigurations.Add(new QueueConfiguration()
                {
                    Events = new List<EventType> { EventType.ObjectCreatedPut },
                    Queue = sqsQueue
                });

                PutBucketNotificationResponse response = await
 client.PutBucketNotificationAsync(request);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' ",
 e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown error encountered on server.
 Message:'{0}' ", e.Message);
            }
        }
    }
}
```

Java

Die folgende Beispiel veranschaulicht, wie Sie einem Bucket eine Benachrichtigungskonfiguration hinzufügen. Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.io.IOException;
```

```
import java.util.EnumSet;

public class EnableNotificationOnABucket {

    public static void main(String[] args) throws IOException {
        String bucketName = "*** Bucket name ***";
        Regions clientRegion = Regions.DEFAULT_REGION;
        String snsTopicARN = "*** SNS Topic ARN ***";
        String sqsQueueARN = "*** SQS Queue ARN ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();
            BucketNotificationConfiguration notificationConfiguration = new
 BucketNotificationConfiguration();

            // Add an SNS topic notification.
            notificationConfiguration.addConfiguration("snsTopicConfig",
                    new TopicConfiguration(snsTopicARN,
 EnumSet.of(S3Event.ObjectCreated)));

            // Add an SQS queue notification.
            notificationConfiguration.addConfiguration("sqsQueueConfig",
                    new QueueConfiguration(sqsQueueARN,
 EnumSet.of(S3Event.ObjectCreated)));

            // Create the notification configuration request and set the bucket
 notification
            // configuration.
            SetBucketNotificationConfigurationRequest request = new
 SetBucketNotificationConfigurationRequest(
                    bucketName, notificationConfiguration);
            s3Client.setBucketNotificationConfiguration(request);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
```

```
    }
}
```

Schritt 4: Testen der Einrichtung

Jetzt können Sie das Setup testen, indem Sie ein Objekt in Ihren Bucket hochladen und die Ereignisbenachrichtigung in der SQS Amazon-Konsole überprüfen. Anweisungen finden Sie unter Empfangen einer Nachricht im Abschnitt "Erste Schritte" im Amazon Simple Queue Service-Entwicklerhandbuch.

Konfigurieren von Ereignis-Benachrichtigungen mithilfe der Namensfilterung

Bei der Konfiguration einer Amazon-S3-Ereignis-Benachrichtigung müssen Sie angeben, welche unterstützten Amazon-S3-Ereignistypen das Senden der Benachrichtigung durch Amazon S3 veranlassen. Wenn ein Ereignistyp, den Sie nicht angegeben haben, in Ihrem S3-Bucket auftritt, sendet Amazon S3 die Benachrichtigung nicht.

Sie können Benachrichtigungen konfigurieren, die nach dem Präfix und dem Suffix des Objektschlüsselnamens gefiltert werden. Sie können beispielsweise eine Konfiguration so einrichten, dass Sie nur eine Benachrichtigung erhalten, wenn einem Bucket Bilddateien mit der Erweiterung „`.jpg`“ hinzugefügt werden. Oder Sie können eine Konfiguration einrichten, die eine Benachrichtigung an ein SNS Amazon-Thema sendet, wenn ein Objekt mit dem Präfix `images/` "" zum Bucket hinzugefügt wird, während Benachrichtigungen für Objekte mit dem Präfix `logs/` "" im selben Bucket an eine AWS Lambda Funktion gesendet werden.

> **Note**
>
> Ein Platzhalterzeichen (*) kann in Filtern nicht als Präfix oder Suffix verwendet werden. Wenn Ihr Präfix oder Suffix ein Leerzeichen enthält, müssen Sie es durch das Zeichen „+“ ersetzen. Wenn Sie weitere Sonderzeichen im Wert des Präfix oder Suffixes verwenden, müssen Sie diese im Format URL-kodiert (prozentkodiert) eingeben. Eine vollständige Liste der Sonderzeichen, die in ein URL -kodiertes Format konvertiert werden müssen, wenn sie in einem Präfix oder Suffix für Ereignisbenachrichtigungen verwendet werden, finden Sie unter. Sichere Zeichen

Sie können Benachrichtigungskonfigurationen einrichten, die die Filterung von Objektschlüsselnamen in der Amazon-S3-Konsole verwenden. Sie können dies tun, indem Sie Amazon S3 APIs über das AWS SDKs oder REST APIs direkt verwenden. Informationen zur Verwendung der Konsolen-Benutzeroberfläche zum Festlegen einer Benachrichtigungskonfiguration für einen Bucket finden Sie unter Aktivieren und Konfigurieren von Ereignis-Benachrichtigungen mit der Amazon-S3-Konsole.

Amazon S3 speichert die Benachrichtigungskonfiguration wie XML in der Unterressource für Benachrichtigungen, die einem Bucket zugeordnet ist, wie unter beschrieben. Verwendung von Amazon SQSSNS, Amazon und Lambda Sie verwenden die `Filter` XML Struktur, um die Regeln für Benachrichtigungen zu definieren, die nach dem Präfix oder Suffix eines Objektschlüsselnamens gefiltert werden sollen. Informationen zur `Filter` XML Struktur finden Sie unter PUTBucket-Benachrichtigung in der Amazon Simple Storage Service API Reference.

Benachrichtigungskonfigurationen, die `Filter` verwenden, können keine Filterregeln mit überlappenden Präfixen, überlappenden Suffixen oder Präfix- und Suffix-Überlappung definieren. Die folgenden Abschnitte enthalten Beispiele für gültige Benachrichtigungskonfigurationen mit Objektschlüssel-Namensfilterung. Sie enthalten auch Beispiele für Benachrichtigungskonfigurationen, die wegen der Überlappung von Präfix und Suffix ungültig sind.

Themen

- Beispiele für gültige Benachrichtigungskonfigurationen mit Filterung nach dem Objektschlüsselnamen
- Beispiele für Benachrichtigungskonfigurationen mit ungültiger Präfix- und Suffix-Überlappung

Beispiele für gültige Benachrichtigungskonfigurationen mit Filterung nach dem Objektschlüsselnamen

Die folgende Benachrichtigungskonfiguration enthält eine Warteschlangenkonfiguration, die eine SQS Amazon-Warteschlange identifiziert, in der Amazon S3 Ereignisse dieses `s3:ObjectCreated:Put` Typs veröffentlichen kann. Die Ereignisse werden immer dann veröffentlicht, wenn sich ein Objekt mit dem Präfix `images/` und einem `jpg` Suffix in einem Bucket befindetPUT.

```
<NotificationConfiguration>
  <QueueConfiguration>
      <Id>1</Id>
      <Filter>
          <S3Key>
              <FilterRule>
                  <Name>prefix</Name>
```

```
                <Value>images/</Value>
            </FilterRule>
            <FilterRule>
                <Name>suffix</Name>
                <Value>jpg</Value>
            </FilterRule>
        </S3Key>
    </Filter>
    <Queue>arn:aws:sqs:us-west-2:444455556666:s3notificationqueue</Queue>
    <Event>s3:ObjectCreated:Put</Event>
  </QueueConfiguration>
</NotificationConfiguration>
```

Die folgende Benachrichtigungskonfigurationen hat mehrere nicht überlappende Präfixe. Die Konfiguration definiert, dass Benachrichtigungen für PUT Anfragen im `images/` Ordner in Warteschlange A und Benachrichtigungen für PUT Anfragen im `logs/` Ordner in Warteschlange B verschoben werden.

```
<NotificationConfiguration>
  <QueueConfiguration>
    <Id>1</Id>
    <Filter>
           <S3Key>
               <FilterRule>
                   <Name>prefix</Name>
                   <Value>images/</Value>
               </FilterRule>
           </S3Key>
    </Filter>
    <Queue>arn:aws:sqs:us-west-2:444455556666:sqs-queue-A</Queue>
    <Event>s3:ObjectCreated:Put</Event>
  </QueueConfiguration>
  <QueueConfiguration>
    <Id>2</Id>
    <Filter>
           <S3Key>
               <FilterRule>
                   <Name>prefix</Name>
                   <Value>logs/</Value>
               </FilterRule>
           </S3Key>
    </Filter>
    <Queue>arn:aws:sqs:us-west-2:444455556666:sqs-queue-B</Queue>
```

```
    <Event>s3:ObjectCreated:Put</Event>
  </QueueConfiguration>
</NotificationConfiguration>
```

Die folgende Benachrichtigungskonfigurationen hat mehrere nicht überlappende Suffixe. Die Konfiguration definiert, dass alle neu zum Bucket hinzugefügten `.jpg` Bilder von Lambda cloud-function-A verarbeitet werden und dass alle neu hinzugefügten `.png` Bilder von cloud-function-B verarbeitet werden. Die `.jpg` Suffixe `.png` und überschneiden sich nicht, obwohl sie denselben letzten Buchstaben haben. Wenn eine bestimmte Zeichenfolge mit beiden Suffixen enden kann, werden die beiden Suffixe als überlappend betrachtet. Da eine Zeichenfolge nicht mit `.png` und `.jpg` gleichzeitig enden kann, sind die Suffixe in der Beispielkonfiguration keine überlappenden Suffixe.

```
<NotificationConfiguration>
  <CloudFunctionConfiguration>
     <Id>1</Id>
     <Filter>
            <S3Key>
                <FilterRule>
                    <Name>suffix</Name>
                    <Value>.jpg</Value>
                </FilterRule>
            </S3Key>
     </Filter>
     <CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-A</
CloudFunction>
     <Event>s3:ObjectCreated:Put</Event>
  </CloudFunctionConfiguration>
  <CloudFunctionConfiguration>
     <Id>2</Id>
     <Filter>
            <S3Key>
                <FilterRule>
                    <Name>suffix</Name>
                    <Value>.png</Value>
                </FilterRule>
            </S3Key>
     </Filter>
     <CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-B</
CloudFunction>
     <Event>s3:ObjectCreated:Put</Event>
  </CloudFunctionConfiguration>
</NotificationConfiguration>
```

Ihre Benachrichtigungskonfigurationen, die `Filter` verwenden, können keine Filterregeln mit überlappenden Präfixen für dieselben Ereignistypen definieren. Sie können dies nur tun, wenn die sich überlappenden Präfixe, die mit Suffixen verwendet werden, die sich nicht überlappen. Die folgende Beispielkonfiguration zeigt, wie Objekte, die mit einem gemeinsamen Präfix, aber nicht überlappenden Suffixen erstellt werden, an unterschiedliche Ziele geliefert werden können.

```
<NotificationConfiguration>
  <CloudFunctionConfiguration>
     <Id>1</Id>
     <Filter>
            <S3Key>
                <FilterRule>
                    <Name>prefix</Name>
                    <Value>images</Value>
                </FilterRule>
                <FilterRule>
                    <Name>suffix</Name>
                    <Value>.jpg</Value>
                </FilterRule>
            </S3Key>
     </Filter>
     <CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-A</
CloudFunction>
     <Event>s3:ObjectCreated:Put</Event>
  </CloudFunctionConfiguration>
  <CloudFunctionConfiguration>
     <Id>2</Id>
     <Filter>
            <S3Key>
                <FilterRule>
                    <Name>prefix</Name>
                    <Value>images</Value>
                </FilterRule>
                <FilterRule>
                    <Name>suffix</Name>
                    <Value>.png</Value>
                </FilterRule>
            </S3Key>
     </Filter>
     <CloudFunction>arn:aws:lambda:us-west-2:444455556666:cloud-function-B</
CloudFunction>
     <Event>s3:ObjectCreated:Put</Event>
  </CloudFunctionConfiguration>
```

```
</NotificationConfiguration>
```

Beispiele für Benachrichtigungskonfigurationen mit ungültiger Präfix- und Suffix-Überlappung

Ihre Benachrichtigungskonfigurationen, die `Filter` verwenden, können größtenteils keine Filterregeln mit überlappenden Präfixen, überlappenden Suffixen oder überlappenden Präfix- und Suffix-Kombinationen für dieselben Ereignistypen definieren. Überlappende Präfixe sind möglich, solange sich die Suffixe nicht überlappen. Ein Beispiel finden Sie unter Konfigurieren von Ereignis-Benachrichtigungen mithilfe der Namensfilterung.

Sie können überlappende Objektschlüsselnamensfilter mit unterschiedlichen Ereignistypen verwenden. Beispielsweise könnten sie eine Benachrichtigungskonfiguration erstellen, die das Präfix `image/` für den Ereignistyp `ObjectCreated:Put` und das Präfix `image/` für den Ereignistyp `ObjectRemoved:*` verwendet.

Sie erhalten eine Fehlermeldung, wenn Sie versuchen, eine Benachrichtigungskonfiguration zu speichern, die ungültige überlappende Namensfilter für dieselben Ereignistypen enthält, wenn Sie die Amazon S3 S3-Konsole verwenden oderAPI. Dieser Abschnitt zeigt Beispiele für Benachrichtigungskonfigurationen, die aufgrund überlappender Namensfilter ungültig sind.

Es wird angenommen, dass eine vorhandene Benachrichtigungskonfigurationsregel ein Standardpräfix und -suffix hat, die mit allen anderen Präfixen bzw. Suffixen übereinstimmen. Die folgende Benachrichtigungskonfiguration ist ungültig, weil sie überlappende Präfixe hat. Insbesondere überlappt sich das Root-Präfix mit allen anderen Präfixen. Das gleiche gilt, wenn Sie in diesem Beispiel ein Suffix anstelle eines Präfixes verwenden. Das Root-Suffix überlappt mit allen anderen Suffixen.

```
<NotificationConfiguration>
     <TopicConfiguration>
         <Topic>arn:aws:sns:us-west-2:444455556666:sns-notification-one</Topic>
         <Event>s3:ObjectCreated:*</Event>
    </TopicConfiguration>
    <TopicConfiguration>
         <Topic>arn:aws:sns:us-west-2:444455556666:sns-notification-two</Topic>
         <Event>s3:ObjectCreated:*</Event>
         <Filter>
             <S3Key>
                 <FilterRule>
                     <Name>prefix</Name>
                     <Value>images</Value>
                 </FilterRule>
```

```
            </S3Key>
        </Filter>
    </TopicConfiguration>
</NotificationConfiguration>
```

Die folgende Benachrichtigungskonfiguration ist ungültig, weil sie überlappende Suffixe hat. Wenn eine bestimmte Zeichenfolge mit beiden Suffixen enden kann, werden die beiden Suffixe als überlappend betrachtet. Eine Zeichenfolge kann mit `jpg` und `pg` enden. Die Suffixe überschneiden sich also. Gleiches gilt für Präfixe. Wenn eine bestimmte Zeichenfolge mit beiden Präfixen beginnen kann, werden die beiden Präfixe als überlappend betrachtet.

```
 <NotificationConfiguration>
     <TopicConfiguration>
         <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-one</Topic>
         <Event>s3:ObjectCreated:*</Event>
         <Filter>
             <S3Key>
                 <FilterRule>
                     <Name>suffix</Name>
                     <Value>jpg</Value>
                 </FilterRule>
            </S3Key>
        </Filter>
    </TopicConfiguration>
    <TopicConfiguration>
         <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-two</Topic>
         <Event>s3:ObjectCreated:Put</Event>
         <Filter>
             <S3Key>
                 <FilterRule>
                     <Name>suffix</Name>
                     <Value>pg</Value>
                 </FilterRule>
            </S3Key>
        </Filter>
    </TopicConfiguration>
</NotificationConfiguration
```

Die folgende Benachrichtigungskonfiguration ist ungültig, weil sie überlappende Präfixe und Suffixe hat.

```
<NotificationConfiguration>
```

```
     <TopicConfiguration>
         <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-one</Topic>
         <Event>s3:ObjectCreated:*</Event>
         <Filter>
             <S3Key>
                 <FilterRule>
                     <Name>prefix</Name>
                     <Value>images</Value>
                 </FilterRule>
                 <FilterRule>
                     <Name>suffix</Name>
                     <Value>jpg</Value>
                 </FilterRule>
            </S3Key>
        </Filter>
    </TopicConfiguration>
    <TopicConfiguration>
         <Topic>arn:aws:sns:us-west-2:444455556666:sns-topic-two</Topic>
         <Event>s3:ObjectCreated:Put</Event>
         <Filter>
             <S3Key>
                 <FilterRule>
                     <Name>suffix</Name>
                     <Value>jpg</Value>
                 </FilterRule>
            </S3Key>
        </Filter>
    </TopicConfiguration>
</NotificationConfiguration>
```

Struktur von Ereignismeldungen

Die Benachrichtigungsnachricht, die Amazon S3 zur Veröffentlichung eines Ereignisses sendet, hat das folgende JSON Format.

Eine allgemeine Übersicht und Anweisungen zum Konfigurieren von Ereignisbenachrichtigungen finden Sie unter Amazon-S3-Ereignis-Benachrichtigungen.

Dieses Beispiel zeigt Version 2.2 der JSON Struktur der Ereignisbenachrichtigung. Amazon S3 verwendet Versionen 2.1 und 2.2 und 2.3 dieser Ereignisstruktur. Amazon S3 verwendet Version 2.2 für regionsübergreifende Replikationsereignisbenachrichtigungen. Es verwendet Version 2.3 für Ereignisse wie S3 Lifecycle, S3 Intelligent-TieringACL, Objekt- und Objekt-Tagging und

Objektwiederherstellung zum Löschen von Objekten. Diese Versionen enthalten zusätzliche Informationen, die für diese Vorgänge spezifisch sind. Die Versionen 2.2 und 2.3 sind ansonsten mit Version 2.1 kompatibel, die Amazon S3 derzeit für alle anderen Ereignisbenachrichtigungstypen verwendet.

```
{
   "Records":[
      {
         "eventVersion":"2.2",
         "eventSource":"aws:s3",
         "awsRegion":"us-west-2",
         "eventTime":"The time, in ISO-8601 format, for example,
 1970-01-01T00:00:00.000Z, when Amazon S3 finished processing the request",
         "eventName":"event-type",
         "userIdentity":{
            "principalId":"Amazon-customer-ID-of-the-user-who-caused-the-event"
         },
         "requestParameters":{
            "sourceIPAddress":"ip-address-where-request-came-from"
         },
         "responseElements":{
            "x-amz-request-id":"Amazon S3 generated request ID",
            "x-amz-id-2":"Amazon S3 host that processed the request"
         },
         "s3":{
            "s3SchemaVersion":"1.0",
            "configurationId":"ID found in the bucket notification configuration",
            "bucket":{
               "name":"amzn-s3-demo-bucket",
               "ownerIdentity":{
                  "principalId":"Amazon-customer-ID-of-the-bucket-owner"
               },
               "arn":"bucket-ARN"
            },
            "object":{
               "key":"object-key",
               "size":"object-size in bytes",
               "eTag":"object eTag",
               "versionId":"object version if bucket is versioning-enabled, otherwise
 null",
               "sequencer": "a string representation of a hexadecimal value used to
 determine event sequence, only used with PUTs and DELETEs"
            }
```

```
         },
         "glacierEventData": {
            "restoreEventData": {
               "lifecycleRestorationExpiryTime": "The time, in ISO-8601 format, for
 example, 1970-01-01T00:00:00.000Z, of Restore Expiry",
               "lifecycleRestoreStorageClass": "Source storage class for restore"
            }
         }
      }
   ]
}
```

Beachten Sie im Zusammenhang mit der Ereignistachrichtenstruktur Folgendes:

- Der `eventVersion`-Schlüsselwert enthält eine Haupt- und eine Nebenversion im Format `<major>.<minor>`.

  Die Hauptversion wird erhöht, wenn Amazon S3 eine Änderung an der Ereignisstruktur vornimmt, die nicht abwärtskompatibel ist. Dazu gehört das Entfernen eines bereits vorhandenen JSON Felds oder das Ändern der Darstellung des Feldinhalts (z. B. ein Datumsformat).

  Die Nebenversion wird erhöht, wenn Amazon S3 der Ereignisstruktur neue Felder hinzufügt. Dies kann auftreten, wenn neue Informationen für einige oder alle vorhandenen Ereignisse bereitgestellt werden. Dies kann auch geschehen, wenn neue Informationen nur zu neu eingeführten Ereignistypen bereitgestellt werden. Anwendungen sollten neue Felder ignorieren, um weiter aufwärtskompatibel mit neuen Nebenversionen der Ereignisstruktur zu sein.

  Falls neue Ereignistypen eingeführt werden, aber die Struktur des Ereignisses anderweitig unmodifiziert ist, ändert sich die Ereignisversion nicht.

  Um sicherzustellen, dass Ihre Anwendungen die Ereignisstruktur ordnungsgemäß analysieren können, empfehlen wir, dass Sie einen Vergleich mit der Hauptversionsnummer durchführen. Um sicherzustellen, dass die Felder, die von Ihrer Anwendung erwartet werden, vorhanden sind, empfehlen wir außerdem, mit der Nebenversion einen greater-than-or-equal -to-Vergleich durchzuführen.
- Die `eventName` verweist auf die Liste der Typen der Ereignisbenachrichtigung, enthält aber nicht die `s3:`-Präfix.
- Der `responseElements` Schlüsselwert ist nützlich, wenn Sie eine Anfrage nachverfolgen möchten, indem Sie mit AWS Support nachverfolgen. `x-amz-request-id` und `x-amz-id-2` helfen Amazon S3, eine einzelne Anfrage nachzuverfolgen. Diese Werte sind dieselben, die

Amazon S3 in der Antwort auf die Anforderung zurückgibt, die die Ereignisse initiiert. Auf diese Weise können sie verwendet werden, um das Ereignis der Anfrage zuzuordnen.

- Der `s3`-Schlüssel bietet Informationen über den Bucket und das Objekt, die an dem Ereignis beteiligt sind. Der Wert des Objektschlüsselnamens ist URL codiert. Beispielsweise wird "red flower.jpg" zu "red+flower.jpg" (Amazon S3 gibt als Inhaltstyp in der Antwort "`application/x-www-form-urlencoded`" zurück).
- Der `sequencer`-Schlüssel bietet eine Möglichkeit, die Reihenfolge von Ereignissen zu bestimmen. Ereignis-Benachrichtigungen kommen nicht garantiert in der Reihenfolge an, in der die Ereignisse aufgetreten sind. Benachrichtigungen von Ereignissen, die Objekte erstellen (PUT) und Objekte löschen, enthalten jedoch ein `sequencer`. Es kann verwendet werden, um die Reihenfolge der Ereignisse für einen bestimmten Objektschlüssel zu bestimmen.

  Wenn Sie die `sequencer`-Zeichenfolgen von zwei Ereignisbenachrichtigungen für denselben Objektschlüssel vergleichen, ist die Ereignisbenachrichtigung mit dem größeren hexadezimalen Wert von `sequencer` das später aufgetretene Ereignis. Wenn Sie Ereignisbenachrichtigungen verwenden, um eine separate Datenbank oder einen separaten Index Ihrer Amazon-S3-Objekte zu verwalten, empfehlen wir Ihnen, die `sequencer`-Werte während der Verarbeitung jeder Ereignisbenachrichtigung zu vergleichen und zu speichern.

  Beachten Sie Folgendes:

  - Sie können `sequencer` nicht verwenden, um die Reihenfolge von Ereignissen für unterschiedliche Objektschlüssel zu bestimmen.
  - Die Sequenzer können unterschiedliche Längen haben. Um diese Werte zu vergleichen, füllen Sie zuerst den kürzeren Wert links mit Nullen auf und führen dann einen alphabetischen Vergleich durch.
- Der `glacierEventData`-Schlüssel ist nur für `s3:ObjectRestore:Completed`-Ereignisse sichtbar.
- Der `restoreEventData`-Schlüssel enthält Attribute, die sich auf Ihre Wiederherstellungsanfrage beziehen.
- Der Schlüssel `replicationEventData` ist nur für Replikationsereignisse sichtbar.
- Der `intelligentTieringEventData`-Schlüssel ist nur für Ereignisse von S3 Intelligent Tiering sichtbar.
- Der `lifecycleEventData`-Schlüssel ist nur für S3-Lebenszyklus-Übergangsereignisse sichtbar.

Beispielnachrichten

Im Folgenden finden Sie Beispiele für Amazon-S3-Ereignis-Benachrichtigungen.

Amazon-S3-Testnachricht

Nachdem Sie eine Ereignis-Benachrichtigung für einen Bucket konfigurieren, sendet Amazon S3 die folgende Testnachricht.

```
{
   "Service":"Amazon S3",
   "Event":"s3:TestEvent",
   "Time":"2014-10-13T15:57:02.089Z",
   "Bucket":"amzn-s3-demo-bucket",
   "RequestId":"5582815E1AEA5ADF",
   "HostId":"8cLeGAmw098X5cv4Zkwcmo8vvZa3eH3eKxsPzbB9wrR+YstdA6Knx4Ip8EXAMPLE"
}
```

Beispielnachricht, wenn ein Objekt mithilfe einer PUT Anfrage erstellt wird

Die folgende Nachricht ist ein Beispiel für eine Nachricht, die Amazon S3 zum Veröffentlichen eines `s3:ObjectCreated:Put`-Ereignisses sendet.

```
{
   "Records":[
      {
         "eventVersion":"2.1",
         "eventSource":"aws:s3",
         "awsRegion":"us-west-2",
         "eventTime":"1970-01-01T00:00:00.000Z",
         "eventName":"ObjectCreated:Put",
         "userIdentity":{
            "principalId":"AIDAJDPLRKLG7UEXAMPLE"
         },
         "requestParameters":{
            "sourceIPAddress":"127.0.0.1"
         },
         "responseElements":{
            "x-amz-request-id":"C3D13FE58DE4C810",
            "x-amz-id-2":"FMyUVURIY8/IgAtTv8xRjskZQpcIZ9KG4V5Wp6S7S/
JRWeUWerMUE5JgHvANOjpD"
         },
         "s3":{
```

```
            "s3SchemaVersion":"1.0",
            "configurationId":"testConfigRule",
            "bucket":{
               "name":"amzn-s3-demo-bucket",
               "ownerIdentity":{
                  "principalId":"A3NL1KOZZKExample"
               },
               "arn":"arn:aws:s3:::amzn-s3-demo-bucket"
            },
            "object":{
               "key":"HappyFace.jpg",
               "size":1024,
               "eTag":"d41d8cd98f00b204e9800998ecf8427e",
               "versionId":"096fKKXTRTtl3on89fVO.nfljtsv6qko",
               "sequencer":"0055AED6DCD90281E5"
            }
         }
      }
   ]
}
```

Eine Definition der einzelnen IAM Identifikationspräfixe (z. B.AIDA,AROA,AGPA) finden Sie im IAMBenutzerhandbuch unter IAMIdentifikatoren.

Verwenden EventBridge

Amazon S3 kann Ereignisse an Amazon senden, EventBridge wenn bestimmte Ereignisse in Ihrem Bucket auftreten. Im Gegensatz zu anderen Zielen müssen Sie nicht auswählen, welche Ereignistypen Sie liefern möchten. Nach EventBridge der Aktivierung werden alle unten aufgeführten Ereignisse an gesendet EventBridge. Sie können EventBridge Regeln verwenden, um Ereignisse an weitere Ziele weiterzuleiten. Im Folgenden sind die Ereignisse aufgeführt, an die Amazon S3 sendet EventBridge.

Ereignistyp	Beschreibung
Objekt erstellt	Ein Objekt wurde erstellt. Das Feld „Grund“ in der Struktur der Ereignisnachricht gibt an, welches S3 zur Erstellung des Objekts verwendet API wurde: PutObject, POSTObject CopyObject, oder CompleteMultipartUpload.

Ereignistyp	Beschreibung
Objekt gelöscht (DeleteObject) Objekt gelöscht (Lebenszyklusablauf)	Ein Objekt wurde gelöscht. Wenn ein Objekt mithilfe eines API S3-Aufrufs gelöscht wird, ist das Feld für den Grund auf gesetzt DeleteObject. Wenn ein Objekt durch eine S3-Lebenszyklus-Ablaufregel gelöscht wird, wird das Grundfeld auf Lebenszyklus-Ablauf gesetzt. Weitere Informationen finden Sie unter Auslaufende Objekte. Wenn ein nicht versioniertes Objekt gelöscht wird oder ein versioniertes Objekt dauerhaft gelöscht wird, wird das Feld für den Löschtyp auf Dauerhaft gelöscht gesetzt. Wenn eine Löschmarkierung für ein versioniertes Objekt erstellt wird, wird das Feld für den Löschtyp auf Löschmarkierung erstellt gesetzt. Weitere Informationen finden Sie unter Löschen von Objekten aus einem versioning-fähigen Bucket.
Objektwiederherstellung eingeleitet	Eine Objektwiederherstellung wurde von der Speicherklasse S3 Glacier oder S3 Glacier Deep Archive oder von der Ebene S3 Intelligent-Tiering Archive Access oder Deep Archive Access initiiert. Weitere Informationen finden Sie unter Arbeiten mit archivierten Objekten.
Objektwiederherstellung abgeschlossen	Eine Objektwiederherstellung wurde abgeschlossen.
Wiederherstellen von Objekten ist abgelaufen	Die temporäre Kopie eines Objekts, das aus S3 Glacier oder S3 Glacier Deep Archive wiederhergestellt wurde, abgelaufen und wurde gelöscht.
Objektspeicherklasse geändert	Ein Objekt wurde auf eine andere Speicherklasse umgestellt. Weitere Informationen finden Sie unter Übergang von Objekten mit Amazon-S3-Lebenszyklus.

Ereignistyp	Beschreibung
Objektzugriffsebene wurde geändert	Ein Objekt wurde auf die Stufen S3 Intelligent-Tiering Archive Access oder Deep Archive Access umgestellt. Weitere Informationen finden Sie unter Verwaltung der Speicherkosten mit Amazon S3 Intelligent-Tiering.
Objekt ACL wurde aktualisiert	Die Zugriffskontrollliste (ACL) eines Objekts wurde mit gesetzt PutObjectACL. Ein Ereignis wird nicht generiert, wenn eine Anforderung zu keiner Änderung an der eines Objekts führtACL. Weitere Informationen finden Sie unter Übersicht über die Zugriffskontrollliste (ACL).
Objekt-Tags hinzugefügt	Ein Satz von Tags wurde einem Objekt hinzugefügt mit PutObjectTagging. Weitere Informationen finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen.
Gelöschte Objekt-Tags	Alle Tags wurden mithilfe von aus einem Objekt entfernt DeleteObjectTagging. Weitere Informationen finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen.

> **Note**
>
> Weitere Informationen darüber, wie Amazon S3 S3-Ereignistypen EventBridge Ereignistypen zugeordnet werden, finden Sie unter EventBridge Amazon-Kartenerstellung und Problembehebung.

Sie können Amazon S3 Event Notifications mit verwenden EventBridge , um Regeln zu schreiben, die Maßnahmen ergreifen, wenn ein Ereignis in Ihrem Bucket eintritt. Sie können sich beispielsweise eine Benachrichtigung senden lassen. Weitere Informationen finden Sie unter Was steht EventBridge im EventBridge Amazon-Benutzerhandbuch.

Weitere Informationen zu den Aktionen und Datentypen, mit denen Sie interagieren können EventBridge API, finden Sie in der EventBridge APIAmazon-Referenz in der EventBridge APIAmazon-Referenz.

Informationen zu den Preisen finden Sie unter EventBridge Amazon-Preise.

Themen

- EventBridge Amazon-Berechtigungen
- Amazon aktivieren EventBridge
- EventBridge Struktur der Ereignisnachricht
- EventBridge Amazon-Kartenerstellung und Problembehebung

EventBridge Amazon-Berechtigungen

Amazon S3 benötigt keine zusätzlichen Berechtigungen, um Ereignisse an Amazon zu übermitteln EventBridge.

Amazon aktivieren EventBridge

Sie können Amazon EventBridge über die S3-Konsole AWS Command Line Interface (AWS CLI) oder Amazon S3 aktivieren RESTAPI.

Verwenden der S3-Konsole

Um die EventBridge Ereigniszustellung in der S3-Konsole zu aktivieren.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie Ereignisse aktivieren möchten.
3. Wählen Sie Properties (Eigenschaften).
4. Gehen Sie zum Abschnitt Event-Benachrichtigungen und suchen Sie den EventBridgeAmazon-Unterabschnitt. Wählen Sie Edit (Bearbeiten) aus.
5. Wählen Sie unter Benachrichtigungen EventBridge für alle Ereignisse in diesem Bucket an Amazon senden die Option An.

> **Note**
>
> Nach der Aktivierung dauert es etwa fünf Minuten EventBridge, bis die Änderungen wirksam werden.

Mit dem AWS CLI

Im folgenden Beispiel wird eine Bucket-Benachrichtigungskonfiguration für einen Bucket `amzn-s3-demo-bucket1` mit EventBridge aktiviertem Amazon erstellt.

```
aws s3api put-bucket-notification-configuration --bucket amzn-s3-demo-bucket1 --
notification-configuration='{ "EventBridgeConfiguration": {} }'
```

Mit dem REST API

Sie können Amazon EventBridge programmgesteuert in einem Bucket aktivieren, indem Sie Amazon S3 aufrufen. REST API Weitere Informationen finden Sie unter PutBucketNotificationConfigurationAmazon Simple Storage Service API Reference.

Das folgende Beispiel zeigt die Konfiguration, die XML verwendet wird, um eine Bucket-Benachrichtigung zu erstellen, wenn Amazon EventBridge aktiviert ist.

```
<NotificationConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <EventBridgeConfiguration>
  </EventBridgeConfiguration>
</NotificationConfiguration>
```

EventBridge Regeln erstellen

Nach der Aktivierung können Sie EventBridge Amazon-Regeln für bestimmte Aufgaben erstellen. Beispielsweise können Sie E-Mail-Benachrichtigungen senden, wenn ein Objekt erstellt wird. Ein vollständiges Tutorial finden Sie unter Tutorial: Eine Benachrichtigung senden, wenn ein Amazon S3 S3-Objekt erstellt wird im EventBridge Amazon-Benutzerhandbuch.

EventBridge Struktur der Ereignisnachricht

Die Benachrichtigungsnachricht, die Amazon S3 zur Veröffentlichung eines Ereignisses sendet, hat das folgende JSON Format. Wenn Amazon S3 ein Ereignis an Amazon sendet EventBridge, sind die folgenden Felder vorhanden.

- Ausführung – Derzeit 0 (Null) für alle Ereignisse.
- id — Eine Version 4, die für jedes Ereignis UUID generiert wird.
- detail-type – Die Art des Ereignisses, das gesendet wird. Eine Liste der Ereignistypen finden Sie unter Verwenden EventBridge.

- Quelle – Gibt den Service an, aus dem das Ereignis stammt.
- Konto– Die 12-stellige AWS-Konto -ID des Bucket-Eigentümers.
- Zeit – Die Zeit, zu der das Ereignis aufgetreten ist.
- Region – Identifiziert den AWS-Region des Buckets.
- resources — Ein JSON Array, das den Amazon-Ressourcennamen (ARN) des Buckets enthält.
- detail — Ein JSON Objekt, das Informationen über das Ereignis enthält. Weitere Informationen dazu, was in diesem Feld enthalten sein kann, finden Sie unter Detailfeld für Ereignismeldung.

Beispiele für Ereignismeldungen

Im Folgenden finden Sie Beispiele für einige der Amazon S3 S3-Ereignisbenachrichtigungen, die an Amazon gesendet werden können EventBridge.

Objekt erstellt


```
{
  "version": "0",
  "id": "17793124-05d4-b198-2fde-7ededc63b103",
  "detail-type": "Object Created",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::amzn-s3-demo-bucket1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "amzn-s3-demo-bucket1"
    },
    "object": {
      "key": "example-key",
      "size": 5,
      "etag": "b1946ac92492d2347c6235b4d2611184",
      "version-id": "IYV3p45BT0ac8hjHg1houSdS1a.Mro8e",
      "sequencer": "617f08299329d189"
    },
    "request-id": "N4N7GDK58NMKJ12R",
    "requester": "123456789012",
```

```
    "source-ip-address": "1.2.3.4",
    "reason": "PutObject"
  }
}
```

Objekt wurde gelöscht (mithilfe von DeleteObject)

```
{
  "version": "0",
  "id": "2ee9cc15-d022-99ea-1fb8-1b1bac4850f9",
  "detail-type": "Object Deleted",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::amzn-s3-demo-bucket1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "amzn-s3-demo-bucket1"
    },
    "object": {
      "key": "example-key",
      "etag": "d41d8cd98f00b204e9800998ecf8427e",
      "version-id": "1QW9g1Z99LUNbvaaYVpW9xDlOLU.qxgF",
      "sequencer": "617f0837b476e463"
    },
    "request-id": "0BH729840619AG5K",
    "requester": "123456789012",
    "source-ip-address": "1.2.3.4",
    "reason": "DeleteObject",
    "deletion-type": "Delete Marker Created"
  }
}
```

Objekt wurde gelöscht (unter Verwendung des Lebenszyklusablaufs)

```
{
  "version": "0",
```

```
  "id": "ad1de317-e409-eba2-9552-30113f8d88e3",
  "detail-type": "Object Deleted",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::amzn-s3-demo-bucket1"
  ],
  "detail": {
    "version": "0",
    "bucket": {
      "name": "amzn-s3-demo-bucket1"
    },
    "object": {
      "key": "example-key",
      "etag": "d41d8cd98f00b204e9800998ecf8427e",
      "version-id": "mtB0cV.jejK63XkRNceanNMC.qXPWLeK",
      "sequencer": "617b398000000000"
    },
    "request-id": "20EB74C14654DC47",
    "requester": "s3.amazonaws.com",
    "reason": "Lifecycle Expiration",
    "deletion-type": "Delete Marker Created"
  }
}
```

Objektwiederherstellung abgeschlossen

```
{
  "version": "0",
  "id": "6924de0d-13e2-6bbf-c0c1-b903b753565e",
  "detail-type": "Object Restore Completed",
  "source": "aws.s3",
  "account": "111122223333",
  "time": "2021-11-12T00:00:00Z",
  "region": "ca-central-1",
  "resources": [
    "arn:aws:s3:::amzn-s3-demo-bucket1"
  ],
  "detail": {
    "version": "0",
```

```
    "bucket": {
      "name": "amzn-s3-demo-bucket1"
    },
    "object": {
      "key": "example-key",
      "size": 5,
      "etag": "b1946ac92492d2347c6235b4d2611184",
      "version-id": "KKsjUC1.6gIjqtvhfg5AdMI0eCePIiT3"
    },
    "request-id": "189F19CB7FB1B6A4",
    "requester": "s3.amazonaws.com",
    "restore-expiry-time": "2021-11-13T00:00:00Z",
    "source-storage-class": "GLACIER"
  }
}
```

Detailfeld für Ereignismeldung

Das Detailfeld enthält ein JSON Objekt mit Informationen über das Ereignis. Die folgenden Felder können im Detailfeld vorhanden sein.

- Ausführung – Derzeit 0 (Null) für alle Ereignisse.
- Bucket – Informationen über den Amazon-S3-Bucket, der an der Veranstaltung beteiligt ist.
- Objekt – Informationen über das an dem Ereignis beteiligte Amazon-S3-Objekt.
- request-id – Anforderungs-ID in S3 Antwort.
- Anforderer — AWS-Konto ID oder AWS Dienstleiter des Anforderers.
- source-ip-address— Quell-IP-Adresse der S3-Anfrage. Nur vorhanden für Ereignisse, die durch eine S3-Anfrage ausgelöst werden.
- Grund — Für Object Created-Ereignisse wurde das S3 API verwendet, um das Objekt zu erstellen: PutObject, POSTObject CopyObject, oder CompleteMultipartUpload. DeleteObjectBei Ereignissen mit gelöschtem Objekt ist dies so eingestellt, dass ein Objekt durch einen API S3-Aufruf gelöscht wird, oder Lifecycle Expiration, wenn ein Objekt durch eine S3-Lebenszyklus-Ablaufregel gelöscht wird. Weitere Informationen finden Sie unter Auslaufende Objekte.
- deletion-type – Wenn ein nicht versioniertes Objekt gelöscht wird oder ein versioniertes Objekt dauerhaft gelöscht wird, wird bei Object Deleted-Ereignissen dies auf Permanently Deleted festgelegt. Wenn ein Löschmarker für ein versioniertes Objekt erstellt wird, wird dieser auf Löschmarkierung erstellt gesetzt. Weitere Informationen finden Sie unter Löschen von Objekten aus einem versioning-fähigen Bucket.

> **Note**
>
> Einige Objektattribute (wie `etag` und`size`) sind nur vorhanden, wenn eine Löschmarke erstellt wird.

- restore-expiry-time— Bei Ereignissen mit Abschluss der Objektwiederherstellung ist dies der Zeitpunkt, zu dem die temporäre Kopie des Objekts aus S3 gelöscht wird. Weitere Informationen finden Sie unter Arbeiten mit archivierten Objekten.
- source-storage-class— Bei Ereignissen wie „Object Restore Initiated“ und „Object Restore Completed“ die Speicherklasse des wiederherzustellenden Objekts. Weitere Informationen finden Sie unter Arbeiten mit archivierten Objekten.
- destination-storage-class— Bei Ereignissen vom Typ Object Storage Class Changed die neue Speicherklasse des Objekts. Weitere Informationen finden Sie unter Übergang von Objekten mit Amazon-S3-Lebenszyklus.
- destination-access-tier— Bei Ereignissen mit geänderter Objektzugriffsebene die neue Zugriffsebene des Objekts. Weitere Informationen finden Sie unter Verwaltung der Speicherkosten mit Amazon S3 Intelligent-Tiering.

EventBridge Amazon-Kartenerstellung und Problembehebung

In der folgenden Tabelle wird beschrieben, wie Amazon S3 S3-Ereignistypen EventBridge Amazon-Ereignistypen zugeordnet werden.

S3-Ereignistyp	EventBridge Amazon-Detailtyp
ObjectCreatedAmazon-Detailtyp ----SEP----:PUT ObjectCreated:Put ----SEP----:Post ObjectCreated:Post ----SEP----:Kopieren ObjectCreated:CompleteMultipartUpload	Objekt erstellt

S3-Ereignistyp	EventBridge Amazon-Detailtyp
ObjectRemoved:Kopieren ----SEP----:Löschen ObjectRemoved:DeleteMarkerCreated LifecycleExpiration:Löschen ----SEP----:Löschen LifecycleExpiration:DeleteMarkerCreated	Objekt gelöscht
ObjectRestore:Löschen ----SEP----:Beitrag	Objektwiederherstellung eingeleitet
ObjectRestore:Beitrag ----SEP----:Abgeschlossen	Objektwiederherstellung abgeschlossen
ObjectRestore:Abgeschlossen ----SEP----:Löschen	Wiederherstellen von Objekten ist abgelaufen
LifecycleTransition	Objektspeicherklasse geändert
IntelligentTiering	Objektzugriffsebene wurde geändert
ObjectTagging:Löschen ----SEP----:PUT	Objekt-Tags hinzugefügt
ObjectTagging:Put ----SEP----:Löschen	Gelöschte Objekt-Tags
ObjectAcl:Löschen ----SEP----:PUT	Objekt wurde aktualisiert ACL

EventBridge Amazon-Problembehebung

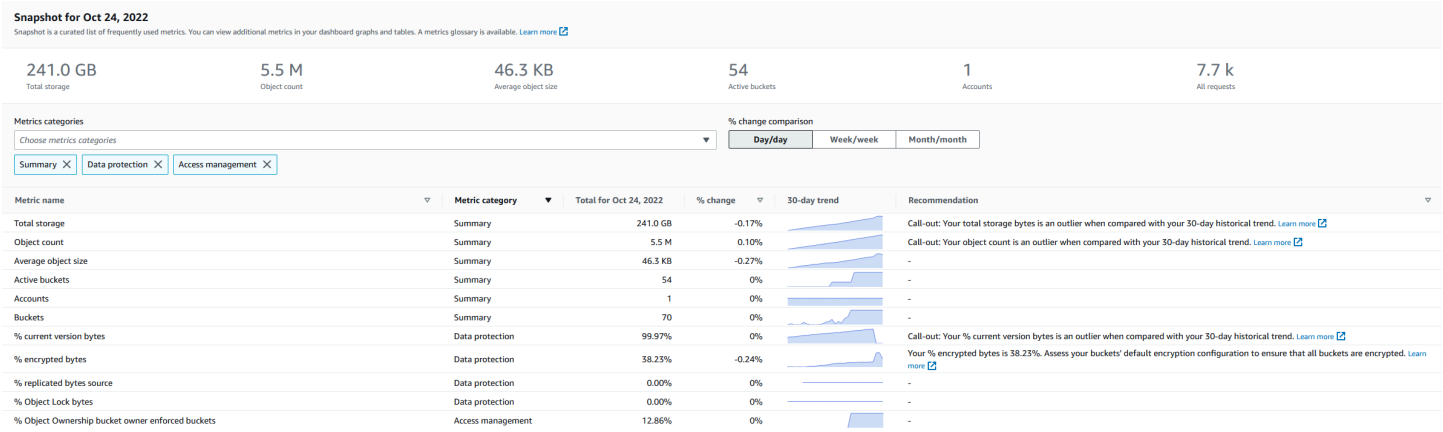
Informationen zur Fehlerbehebung EventBridge finden Sie unter Problembehandlung bei Amazon EventBridge im EventBridge Amazon-Benutzerhandbuch.

Bewerten Ihrer Speicheraktivität und -nutzung mit Amazon S3 Storage Lens

Amazon S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Erkenntnisse zu Objektspeichern und Aktivitäten erhalten können. S3 Storage Lens analysiert Metriken, um kontextbezogene Empfehlungen zur Optimierung der Speicherkosten und zur Anwendung bewährter Datenschutzmethoden zu geben.

Mithilfe der Metriken von S3 Storage Lens lässt sich eine Zusammenfassung der Erkenntnisse generieren. Beispielseise können Sie herausfinden, wie viel Speicher Sie in der gesamten Organisation haben oder welche Buckets und Präfixe am schnellsten wachsen. Außerdem können Sie anhand der Metriken von S3 Storage Lens umfassende Möglichkeiten zur Kostenoptimierung aufdecken, bewährte Methoden für den Datenschutz und die Zugriffsverwaltung implementieren und die Leistung von Anwendungs-Workloads verbessern. Sie können beispielsweise Buckets identifizieren, für die keine S3-Lebenszyklusregeln gelten, damit unvollständige mehrteilige Uploads, die älter als 7 Tage sind, abgebrochen werden. Sie können auch Buckets identifizieren, die nicht den bewährten Datenschutzmethoden entsprechen, z. B. die Verwendung von S3 Replication oder S3 Versionierung.

S3 Storage Lens aggregiert Ihre Metriken und zeigt die Informationen im Abschnitt Account snapshot (Konto-Snapshot) der Seite Buckets der Amazon-S3-Konsole an. S3 Storage Lens bietet außerdem ein interaktives Dashboard, mit dem Sie Erkenntnisse und Trends visualisieren, Ausreißer kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten erhalten und bewährte Datenschutzmethoden anwenden können. Das Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Organisations-, Konto-, AWS-Region-, Speicherklassen-, Bucket-, Präfix- oder Objekt- oder Storage-Lens-Gruppenebene zu generieren. Sie können auch einen täglichen Metrikexport in CSV oder senden Parquet in einen S3-Bucket formatieren.



Metriken und Funktionen von S3 Storage Lens

S3 Storage Lens bietet ein interaktives Standard-Dashboard, das täglich aktualisiert wird. Gemäß Vorkonfiguration von S3 Storage Lens werden in diesem Dashboard die zusammengefassten Erkenntnisse und Trends Ihres gesamten Kontos visualisiert und täglich in der S3-Konsole aktualisiert. Metriken aus diesem Dashboard werden auch in Ihrem Konto-Snapshot auf der Seite Buckets zusammengefasst. Weitere Informationen finden Sie unter Standard-Dashboard.

Um weitere Dashboards zu erstellen und sie nach AWS-Regionen S3-Buckets oder Konten (für AWS Organizations) zu skalieren, erstellen Sie eine S3 Storage Lens-Dashboard-Konfiguration. Sie können S3 Storage Lens-Dashboard-Konfigurationen mithilfe der Amazon S3-Konsole AWS Command Line Interface (AWS CLI) oder Amazon S3 erstellen und verwalten RESTAPI. AWS SDKs Wenn Sie ein S3-Storage-Lens-Dashboard erstellen oder bearbeiten, definieren Sie den Dashboard-Umfang und die Metrikauswahl.

S3 Storage Lens bietet neben kostenlosen Metriken auch erweiterte Metriken und Empfehlungen, auf die Sie gegen eine zusätzliche Gebühr upgraden können. Mit erweiterten Metriken und Empfehlungen können Sie auf zusätzliche Metriken und Funktionen zugreifen, um Erkenntnisse über Ihren Speicher zu erhalten. Zu diesen Funktionen gehören erweiterte Metrikkategorien, Präfixaggregation, kontextuelle Empfehlungen und Amazon-Veröffentlichungen. CloudWatch Präfixaggregation und kontextbezogene Empfehlungen sind nur in der Amazon-S3-Konsole verfügbar. Weitere Informationen zu den Preisen für S3 Storage Lens finden Sie unter Amazon S3-Preise.

Metrikkategorien

In den kostenlosen und erweiterten Stufen sind die Metriken in Kategorien unterteilt, die auf wichtige Anwendungsfälle wie Kostenoptimierung und Datenschutz abgestimmt sind. Zu den kostenlosen Metriken gehören Zusammenfassung, Kostenoptimierung, Datenschutz, Zugriffsverwaltung, Leistung und Ereignismetriken. Wenn Sie auf erweiterte Metriken und Empfehlungen upgraden, können Sie erweiterte Metriken zur Kostenoptimierung und zum Datenschutz aktivieren. Anhand dieser erweiterten Metriken können Sie Ihre S3-Speicherkosten weiter senken und Ihre Datenschutzpolitik verbessern. Sie können auch Aktivitätsmetriken und detaillierte Statuscode-Metriken aktivieren, um die Leistung von Anwendungs-Workloads zu verbessern, die auf Ihre S3-Buckets zugreifen. Weitere Informationen zu kostenlosen und erweiterten Metriken finden Sie unter Metrikauswahl.

Sie können Ihren Speicher auf der Grundlage bewährter Methoden von S3 bewerten, z. B. indem Sie den Prozentsatz der Buckets analysieren, für die die Verschlüsselung, S3 Object Lock oder S3

Versioning aktiviert ist. Sie können auch potenzielle Kosteneinsparungsmöglichkeiten identifizieren. Zum Beispiel können Sie Metriken zur Anzahl von S3-Lebenszyklusregeln verwenden, um Buckets zu identifizieren, in denen Lebenszyklusregeln zum Ablauf und Übergang fehlen. Sie können auch Ihre Anforderungsaktivität pro Bucket analysieren, um Buckets zu finden, in denen Objekte in eine kostengünstigere Speicherklasse überführt werden können. Weitere Informationen finden Sie unter Anwendungsfälle für Metriken von Amazon S3 Storage Lens.

Metrik-Export

Sie können nicht nur das Dashboard auf der S3-Konsole anzeigen, sondern auch Metriken in oder exportieren CSV Parquet zur weiteren Analyse mit dem Analysetool Ihrer Wahl in einen S3-Bucket formatieren. Weitere Informationen finden Sie unter Anzeigen von Amazon S3-Storage-Lens-Metriken mit einem Datenexport.

CloudWatch Amazon-Verlagswesen

Sie können die Nutzungs- und Aktivitätsmetriken von S3 Storage Lens CloudWatch auf Amazon veröffentlichen, um eine einheitliche Ansicht Ihres Betriebszustands in CloudWatch Dashboards zu erstellen. Sie können auch CloudWatch Funktionen wie Alarme und ausgelöste Aktionen, metrische Berechnungen und Anomalieerkennung verwenden, um S3 Storage Lens-Metriken zu überwachen und entsprechende Maßnahmen zu ergreifen. Darüber hinaus ermöglichen CloudWatch API Betriebsabläufe Anwendungen, einschließlich Drittanbietern, den Zugriff auf Ihre S3 Storage Lens-Metriken. Die CloudWatch Veröffentlichungsoption ist für Dashboards verfügbar, die auf erweiterte Messwerte und Empfehlungen von S3 Storage Lens aktualisiert wurden. Weitere Informationen zur Unterstützung von S3 Storage Lens-Metriken finden Sie CloudWatch unterÜberwachen Sie S3 Storage Lens-Metriken in CloudWatch.

Weitere Informationen zur Verwendung von S3 Storage Lens finden Sie in den folgenden Themen.

Themen

- Grundlegendes zu Amazon S3 Storage Lens
- Amazon S3-Storage-Lens-Metrikglossar
- Amazon S3 Storage Lens-Berechtigungen einrichten
- Arbeiten mit Amazon S3 Storage Lens mithilfe der Konsole und API
- Anzeigen von Metriken mit Amazon S3 Storage Lens
- Verwenden von Amazon S3 Storage Lens mit AWS Organizations
- Arbeiten mit S3 Storage Lens-Gruppen zum Filtern und Aggregieren von Metriken

Grundlegendes zu Amazon S3 Storage Lens

> **Important**
>
> Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3 an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist in den AWS CloudTrail Protokollen, im S3-Inventar, in der S3-Speicherlinse, in der Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header im AWS Command Line Interface und AWS SDKs verfügbar. Weitere Informationen finden Sie unter Standardverschlüsselung FAQ.

Amazon S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Sie können Metriken von S3 Storage Lens verwenden, um zusammenfassende Erkenntnisse zu gewinnen und z. B. herauszufinden, wie viel Speicher Sie in Ihrer gesamten Organisation haben oder welche Buckets und Präfixe am schnellsten wachsen. Außerdem können Sie anhand der Metriken von S3 Storage Lens umfassende Möglichkeiten zur Kostenoptimierung aufdecken, bewährte Methoden für den Datenschutz implementieren und die Leistung von Anwendungs-Workloads verbessern. Sie können beispielsweise Buckets identifizieren, für die keine S3-Lebenszyklusregeln gelten, damit unvollständige mehrteilige Uploads, die älter als 7 Tage sind, ablaufen. Sie können auch Buckets identifizieren, die nicht den bewährten Datenschutzmethoden entsprechen, z. B. die Verwendung von S3 Replication oder S3 Versionierung. S3 Storage Lens analysiert Metriken, um kontextbezogene Empfehlungen zur Optimierung der Speicherkosten und zur Anwendung bewährter Datenschutzmethoden zu geben.

S3 Storage Lens aggregiert Ihre Metriken und zeigt die Informationen im Abschnitt Account snapshot (Konto-Snapshot) der Seite Buckets der Amazon-S3-Konsole an. S3 Storage Lens bietet außerdem ein interaktives Dashboard, mit dem Sie Erkenntnisse und Trends visualisieren, Ausreißer kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten erhalten und bewährte Datenschutzmethoden anwenden können. Das Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Organisations-, Konto-, AWS-Region-, Speicherklassen-, Bucket-, Präfix- oder Objekt- oder Storage-Lens-Gruppenebene zu generieren. Sie können auch einen täglichen Metrikexport in CSV oder senden Parquet in einen S3-Bucket formatieren. Sie können S3 Storage

Lens-Dashboards mithilfe der Amazon S3-Konsole AWS Command Line Interface (AWS CLI) oder Amazon S3 REST API erstellen AWS SDKs und verwalten.

Konzepte und Terminologie von S3 Storage Lens

Dieser Abschnitt enthält die Terminologie und Konzepte, die für das Verständnis und die erfolgreiche Verwendung von Amazon S3 Storage Lens unerlässlich sind.

Themen

- Konfiguration des Dashboards
- Standard-Dashboard
- Dashboards
- Konto-Snapshot
- Metrik-Export
- Heimatregion
- Aufbewahrungszeitraum
- Metrikkategorien
- Empfehlungen
- Metrikauswahl
- S3 Storage Lens und AWS Organizations

Konfiguration des Dashboards

S3 Storage Lens erfordert eine Dashboard-Konfiguration, die die Eigenschaften enthält, mit denen Metriken in Ihrem Namen für ein einzelnes Dashboard oder einen Export aggregiert werden. Wenn Sie eine Konfiguration erstellen, wählen Sie den Dashboard-Namen und die Heimatregion aus, die Sie nach der Erstellung des Dashboards nicht mehr ändern können. Sie können optional Tags hinzufügen und einen Metrikexport in CSV oder konfigurieren Parquet .

In der Dashboard-Konfiguration definieren Sie auch den Dashboard-Umfang und die Metrikauswahl. Der Umfang kann den gesamten Speicherplatz für Ihr Organisationskonto oder Abschnitte umfassen, die nach Region, Bucket und Konto gefiltert sind. Wenn Sie die Metrikauswahl konfigurieren, wählen Sie zwischen kostenlosen Metriken und erweiterten Metriken und Empfehlungen, auf die Sie gegen eine zusätzliche Gebühr upgraden können. Mit erweiterten Metriken und Empfehlungen können Sie auf zusätzliche Metriken und Funktionen zugreifen. Zu diesen Funktionen gehören erweiterte Metrikkategorien, Aggregation auf Präfixebene, kontextuelle Empfehlungen und Amazon-

Veröffentlichungen. CloudWatch Weitere Informationen zu den Preisen für S3 Storage Lens finden Sie unter Amazon S3-Preise.

Standard-Dashboard

Das Standard-Dashboard von S3 Storage Lens auf der Konsole hat den Namen. default-account-dashboard Gemäß S3-Vorkonfiguration werden in diesem Dashboard die zusammengefassten Erkenntnisse und Trends Ihres gesamten Kontos visualisiert und täglich in der S3-Konsole aktualisiert. Sie können den Konfigurationsbereich des Standard-Dashboards zwar nicht ändern, haben aber die Möglichkeit, ein Upgrade für die Metrikauswahl durchzuführen und statt der kostenlosen Metriken die erweiterten Metriken und Empfehlungen zu verwenden. Sie können den optionalen Metrikexport konfigurieren oder sogar das Dashboard deaktivieren. Löschen können Sie das Standard-Dashboard jedoch nicht.

> **Note**
>
> Wenn Sie Ihr Standard-Dashboard deaktivieren, wird es nicht mehr aktualisiert. Sie erhalten auf Ihrem S3 Storage Lens-Dashboard, in Ihrem Metrikexport oder im Konto-Snapshot auf der S3-Seite Buckets keine neuen täglichen Metriken mehr. Wenn Ihr Dashboard erweiterte Metriken und Empfehlungen verwendet, werden Ihnen keine Gebühren mehr berechnet. Sie können im Dashboard weiterhin historische Daten anzeigen, bis der Zeitraum von 14 Tagen für Datenabfragen abläuft. Wenn Sie erweiterte Metriken und Empfehlungen aktiviert haben, beträgt dieser Zeitraum 15 Monate. Sie können das Dashboard innerhalb des Ablaufzeitraums wieder aktivieren, um auf historische Daten zuzugreifen.

Dashboards

Sie können zusätzliche S3 Storage Lens-Dashboards erstellen und deren Umfang nach AWS-Regionen S3-Buckets oder Konten (für AWS Organizations) festlegen. Wenn Sie ein S3-Storage-Lens-Dashboard erstellen oder bearbeiten, definieren Sie den Dashboard-Umfang und die Metrikauswahl. S3 Storage Lens bietet neben kostenlosen Metriken auch erweiterte Metriken und Empfehlungen, auf die Sie gegen eine zusätzliche Gebühr upgraden können. Mit erweiterten Metriken und Empfehlungen können Sie auf zusätzliche Metriken und Funktionen zugreifen, um Erkenntnisse über Ihren Speicher zu erhalten. Dazu gehören erweiterte Metrikkategorien, Aggregation auf Präfixebene, kontextuelle Empfehlungen und Amazon-Veröffentlichungen. CloudWatch Weitere Informationen zu den Preisen für S3 Storage Lens finden Sie unter Amazon S3-Preise.

Sie können das Dashboard auch deaktivieren oder löschen. Wenn Sie ein Dashboard deaktivieren, wird es nicht mehr aktualisiert und Sie erhalten keine neuen täglichen Metriken mehr. Sie können historische Daten dann noch bis zum Ende des Ablaufzeitraums von 14 Tagen einsehen. Wenn Sie erweiterte Metriken und Empfehlungen für das Dashboard aktiviert haben, beträgt dieser Zeitraum 15 Monate. Sie können das Dashboard innerhalb des Ablaufzeitraums wieder aktivieren, um auf historische Daten zuzugreifen.

Wenn Sie ein Dashboard löschen, gehen alle Dashboard-Konfigurationseinstellungen verloren. Sie erhalten dann keine neuen täglichen Metriken mehr und können nicht mehr auf die mit dem Dashboard verbundenen historischen Daten zugreifen. Wenn Sie auf die historischen Daten eines gelöschten Dashboards zugreifen möchten, müssen Sie ein neues Dashboard mit demselben Namen in derselben Heimatregion erstellen.

> **Note**
>
> - Sie können S3 Storage Lens verwenden, um bis zu 50 Dashboards pro Heimatregion zu erstellen.
> - Dashboards auf Organisationsebene können nur auf einen regionalen Bereich beschränkt werden.

Konto-Snapshot

Der Konto-Snapshot von S3 Storage Lens fasst Metriken aus Ihrem Standard-Dashboard zusammen und zeigt Ihren Gesamtspeicher, die Objektzahl und die durchschnittliche Objektgröße auf der Seite Buckets der S3-Konsole an. Mit diesem Konto-Snapshot haben Sie schnellen Zugriff auf Erkenntnisse über Ihren Speicher, ohne die Seite Buckets verlassen zu müssen. Der Konto-Snapshot bietet auch mit einem Klick Zugriff auf Ihr interaktives S3-Storage-Lens-Dashboard.

Sie können Ihr Dashboard verwenden, um Erkenntnisse und Trends zu visualisieren, Ausreißer zu kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten und zur Anwendung bewährter Datenschutzmethoden einzusehen. Ihr Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Organisations-, Konto-, Bucket-, Objekt- oder sogar Präfixebene zu generieren. Sie können auch einmal täglich einen Metrikexport an einen S3-Bucket in oder senden CSV Parquet .

Sie können den Dashboard-Umfang des default-account dashboard (Standardkonto-Dashboards) nicht ändern, da er mit dem Account snapshot (Konto-Snapshot) verknüpft ist. Sie können die Metrikauswahl in Ihrem jedoch default-account-dashboardvon kostenlosen Metriken auf

kostenpflichtige erweiterte Metriken und Empfehlungen umstellen. Nach dem Upgrade können Sie alle Anfragen, hochgeladenen Bytes und heruntergeladenen Bytes im Account snapshot (Konto-Snapshot) von S3 Storage Lens anzeigen.

> **Note**
>
> Wenn Sie Ihr Standard-Dashboard deaktivieren, wird Ihr Account snapshot (Konto-Snapshot) nicht mehr aktualisiert. Um weiterhin Metriken im Konto-Snapshot anzuzeigen, können Sie den default-account-dashboarderneut aktivieren.

Metrik-Export

Ein Metrik-Export ist eine Datei, die alle in Ihrer S3-Storage-Lens-Konfiguration erfassten Metriken enthält. Diese Informationen werden täglich in CSV oder generiert Parquet formatiert und wird an einen S3-Bucket gesendet. Sie können den Metrikexport für weitere Analysen verwenden, indem Sie das Metrik-Tool Ihrer Wahl verwenden. Der S3-Bucket für den Export Ihrer Metriken muss sich in derselben Region wie Ihre S3-Storage-Lens-Konfiguration befinden. Sie können einen Metrikexport von S3 Storage Lens über die S3-Konsole generieren, indem Sie Ihre Dashboard-Konfiguration bearbeiten. Sie können einen Metrikexport auch mithilfe von AWS CLI und konfigurieren AWS SDKs.

Heimatregion

In der Heimatregion werden alle AWS-Region S3 Storage Lens-Metriken für eine bestimmte Dashboard-Konfiguration gespeichert. Sie müssen eine Heimatregion auswählen, wenn Sie Ihr S3-Storage-Lens-Dashboard konfigurieren. Nachdem Sie eine Heimatregion ausgewählt haben, können Sie sie nicht mehr ändern. Wenn Sie eine Storage-Lens-Gruppe erstellen, empfehlen wir außerdem, dieselbe Heimatregion wie für das Storage-Lens-Dashboard auszuwählen.

> **Note**
>
> Sie können eine der folgenden Regionen als Heimatregion auswählen:
>
> - US East (N. Virginia) – `us-east-1`
> - US East (Ohio) – `us-east-2`
> - US West (N. California) – `us-west-1`
> - US West (Oregon) – `us-west-2`
> - Asia Pacific (Mumbai) – `ap-south-1`

- Asien-Pazifik (Seoul) – `ap-northeast-2`
- Asia Pacific (Singapore) – `ap-southeast-1`
- Asien-Pazifik (Sydney) – `ap-southeast-2`
- Asien-Pazifik (Tokio) – `ap-northeast-1`
- Canada (Central) – `ca-central-1`
- China (Peking) – `cn-north-1`
- China (Ningxia) – `cn-northwest-1`
- Europe (Frankfurt) – `eu-central-1`
- Europe (Ireland) – `eu-west-1`
- Europe (London) – `eu-west-2`
- Europe (Paris) – `eu-west-3`
- Europe (Stockholm) – `eu-north-1`
- South America (São Paulo) – `sa-east-1`

Aufbewahrungszeitraum

S3-Storage-Lens-Metriken werden beibehalten, damit Sie historische Trends erkennen und Unterschiede bei Speichernutzung und -aktivität im Zeitverlauf vergleichen können. Sie können Metriken von Amazon S3 Storage Lens für Abfragen verwenden, um historische Trends anzuzeigen und Unterschiede in Ihrer Speichernutzung und -aktivität im Zeitverlauf zu vergleichen.

Alle Metriken für S3 Storage Lens werden für einen Zeitraum von 15 Monaten aufbewahrt. Metriken sind jedoch nur für Abfragen für eine bestimmte Dauer verfügbar, die von Ihrer Metrikauswahl abhängt. Diese Dauer kann nicht geändert werden. Kostenlose Metriken stehen für Abfragen für einen Zeitraum von 14 Tagen zur Verfügung, und fortschrittliche Metriken stehen für Abfragen für 15 Monate zur Verfügung.

Metrikkategorien

In den kostenlosen und erweiterten Stufen sind die Metriken von S3 Storage Lens in Kategorien unterteilt, die auf wichtige Anwendungsfälle wie Kostenoptimierung und Datenschutz abgestimmt sind. Zu den kostenlosen Metriken gehören Zusammenfassung, Kostenoptimierung, Datenschutz, Zugriffsverwaltung, Leistung und Ereignismetriken. Wenn Sie auf erweiterte Metriken und Empfehlungen upgraden, können Sie zusätzliche Metriken zur Kostenoptimierung und zum Datenschutz aktivieren, mit denen Sie Ihre S3-Speicherkosten weiter senken und den Datenschutz

gewährleisten können. Sie können auch Aktivitätsmetriken und detaillierte Statuscode-Metriken aktivieren, um die Leistung von Anwendungs-Workloads zu verbessern.

Die folgende Liste enthält alle kostenlosen und erweiterten Metrikkategorien. Eine vollständige Liste der einzelnen Metriken, die in jeder Kategorie enthalten sind, finden Sie im Metrik-Glossar.

Zusammenfassende Metriken

Zusammenfassende Metriken bieten allgemeine Erkenntnisse über Ihren S3-Speicher, einschließlich Ihrer gesamten Speicherbytes und der Anzahl der Objekte.

Metriken zur Kostenoptimierung

Metriken zur Kostenoptimierung liefern Erkenntnisse, die Sie zur Verwaltung und Optimierung Ihrer Speicherkosten nutzen können. Sie können beispielsweise Buckets identifizieren, unvollständige mehrteilige Uploads enthalten, die älter als 7 Tage sind.

Mit erweiterten Metriken und Empfehlungen können Sie erweiterte Metriken zur Kostenoptimierung aktivieren. Zu diesen Metriken gehören Metriken für die Anzahl der S3-Lebenszyklusregeln, mit denen Sie die Anzahl der S3-Lebenszyklusregeln pro Bucket für den Ablauf und für den Übergang ermitteln können.

Metriken zum Datenschutz

Metriken zum Datenschutz liefern Erkenntnisse über Datenschutzfunktionen wie Verschlüsselung und S3 Versioning. Sie können diese Metriken verwenden, um Buckets zu identifizieren, die nicht den bewährten Datenschutzmethoden entsprechen. Sie können beispielsweise Buckets identifizieren, die keine Standardverschlüsselung mit AWS Key Management Service Schlüsseln (SSE-KMS) oder S3-Versionierung verwenden.

Mit erweiterten Metriken und Empfehlungen können Sie erweiterte Metriken zum Datenschutz aktivieren. Zu diesen Metriken gehören Metriken für die Anzahl der Replikationsregeln pro Bucket.

Metriken zur Zugriffsverwaltung

Metriken zur Zugriffsverwaltung liefern Erkenntnisse im Hinblick auf S3 Object Ownership. Sie können diese Metriken verwenden, um zu sehen, welche Einstellungen Ihre Buckets für die Objekteigentümerschaft verwenden.

Ereignismetriken

Ereignismetriken bieten Einblicke in S3-Ereignisbenachrichtigungen. Mithilfe von Ereignismetriken können Sie sehen, für welche Buckets S3-Ereignisbenachrichtigungen konfiguriert sind.

Leistungsmetriken

Leistungsmetriken liefern relevante Erkenntnisse für S3 Transfer Acceleration. Anhand von Leistungsmetriken können Sie sehen, für welche Buckets Transfer Acceleration aktiviert ist.

Aktivitätsmetriken (erweitert)

Wenn Sie das Dashboard auf erweiterte Metriken und Empfehlungen upgraden, können Sie Aktivitätsmetriken aktivieren. Aktivitätsmetriken liefern Details darüber, wie Ihr Speicherplatz angefordert wird (z. B. All-Anforderungen, Get-Anforderungen, Put-Anforderungen), sowie über hoch- oder heruntergeladene Bytes und Fehler.

Anhand von Aktivitätsmetriken auf Präfixebene können Sie ermitteln, welche Präfixe selten verwendet werden, sodass Sie im S3-Lebenszyklus zu einer optimaleren Speicherklasse wechseln können.

Detaillierte Statuscode-Metriken (erweitert)

Wenn Sie das Dashboard auf erweiterte Metriken und Empfehlungen upgraden, können Sie detaillierte Statuscode-Metriken aktivieren. Detaillierte Statuscode-Metriken bieten Einblicke in HTTP Statuscodes wie 403 Forbidden und 503 Service Unavailable, die Sie zur Behebung von Zugriffs- oder Leistungsproblemen verwenden können. Sie können sich beispielsweise die Metrik 403 Forbidden error count (Anzahl der Fehler 403 Forbidden) ansehen, um Workloads zu identifizieren, die auf Buckets zugreifen, ohne dass die korrekten Berechtigungen angewendet wurden.

Detaillierte Statuscode-Metriken auf Präfixebene können verwendet werden, um ein besseres Verständnis der Vorkommen von HTTP Statuscodes anhand von Präfixen zu erhalten. Mit den Metriken zur Anzahl der 503-Fehler können Sie beispielsweise Präfixe ermitteln, die während der Datenerfassung Drosselungsanfragen erhalten.

Empfehlungen

S3 Storage Lens bietet automatisierte Empfehlungen, die Sie bei der Optimierung Ihres Speichers unterstützen. Die Empfehlungen werden kontextbasiert neben relevanten Metriken im S3-Storage-Lens-Dashboard platziert. Historische Daten kommen nicht für Empfehlungen in Betracht, da Empfehlungen nur für aktuelle Geschehnisse relevant sind. Empfehlungen werden nur angezeigt, wenn sie relevant sind.

Es gibt folgende Arten von S3-Storage-Lens-Empfehlungen:

- Vorschläge

Vorschläge weisen Sie auf Trends beim Speicher und bei Aktivitäten hin, die auf Möglichkeiten zur Optimierung der Speicherkosten oder auf bewährte Datenschutzmethoden hindeuten könnten. Anhand der vorgeschlagenen Themen können Sie im Amazon-S3-Benutzerhandbuch und im S3-Storage-Lens-Dashboard nach weiteren Informationen zu den spezifischen Regionen, Buckets oder Präfixen suchen.

- Callouts

  Callouts sind Empfehlungen, die Sie auf interessante Anomalien in Bezug auf den Speicher und Aktivitäten aufmerksam machen, die über einen bestimmten Zeitraum auftreten und möglicherweise Maßnahmen erfordern oder überwacht werden sollten.

  - Ausreißer-Callouts

    S3 Storage Lens bietet Callouts für Metriken, bei denen es sich auf der Basis Ihres Trends der letzten 30 Tage um Ausreißer handelt. Der Ausreißer wird mit einem Standardwert berechnet, der auch als Z-Score bezeichnet wird. Bei diesem Score wird die Metrik des aktuellen Tages vom Durchschnitt der letzten 30 Tage für die Metrik subtrahiert. Die Metrik des aktuellen Tages wird dann durch die Standardabweichung für die Metrik in den letzten 30 Tagen dividiert. Der resultierende Score liegt normalerweise zwischen -3 und +3. Diese Zahl stellt die Anzahl der Standardabweichungen dar, die die Metrik des aktuellen Tages vom Mittelwert entfernt ist.

    Metriken mit einem Score von **>** 2 oder **<** -2 werden in S3 Storage Lens als Ausreißer erachtet, da sie über oder unter dem Wert von 95 Prozent der normal verteilten Daten liegen.

  - Callouts für signifikante Änderungen

    Das Callout für signifikante Änderungen gilt für Metriken, die sich erwartungsgemäß selten ändern. Daher ist er auf eine höhere Sensitivität eingestellt als die Ausreißerberechnung, die sich typischerweise im Bereich von +/-20 Prozent gegenüber dem Vortag, der Vorwoche oder dem Vormonat befindet.

    Bearbeiten von Callouts für den Speicher und Aktivitäten – Wenn Sie ein Callout für signifikante Änderungen erhalten, muss das nicht auf ein Problem hinweisen. Das Callout könnte das Ergebnis einer erwarteten Änderung des Speichers sein. Beispielsweise können Sie vor Kurzem eine große Anzahl neuer Objekte hinzugefügt, eine große Anzahl von Objekten gelöscht oder ähnliche geplante Änderungen vorgenommen haben.

    Wenn Sie in Ihrem Dashboard ein Callout für signifikante Änderungen sehen, sollten Sie zunächst bestimmen, ob es durch die jüngsten Umstände erklärt werden kann. Ist dies nicht

der Fall, können Sie über das S3-Storage-Lens-Dashboard weitere Details abrufen, um herauszufinden, welchen spezifischen Regionen, Buckets oder Präfixen die Fluktuation zugrunde liegt.

- Erinnerungen

  Erinnerungen bieten Einblicke in die Funktionsweise von Amazon S3. Sie können Ihnen helfen, mehr darüber zu erfahren, wie Sie S3-Funktionen nutzen können, um die Speicherkosten zu senken oder bewährte Datenschutzmethoden anzuwenden.

Metrikauswahl

S3 Storage Lens bietet zwei Arten von Metriken, die Sie für das Dashboard und den Export auswählen können: kostenlose Metriken und fortschrittliche Metriken und Empfehlungen.

- Kostenlose Metriken

  S3 Storage Lens bietet kostenlose Metriken für alle Dashboards und Konfigurationen. Kostenlose Metriken enthalten relevante Metriken für Ihre Speichernutzung, z. B. die Anzahl der Buckets und der Objekte in Ihrem Konto. Zu den kostenlosen Metriken gehören auch auf Anwendungsfällen basierende Metriken (z. B. Metriken zur Kostenoptimierung und zum Datenschutz), anhand derer Sie untersuchen können, ob Ihr Speicher gemäß den bewährten Methoden von S3 konfiguriert ist. Alle kostenlosen Metriken werden täglich gesammelt. Daten stehen 14 Tage für Abfragen zur Verfügung. Weitere Informationen darüber, welche Metriken kostenlos sind, finden Sie unter Amazon S3-Storage-Lens-Metrikglossar.

- Fortschrittliche Metriken und Empfehlungen

  S3 Storage Lens bietet kostenlose Metriken für alle Dashboards und Konfigurationen mit der Option zum Upgrade auf fortschrittliche Metriken und Empfehlungen. Es fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter Amazon S3 – Preise.

  Zu den erweiterten Metriken und Empfehlungen gehören neben den kostenlosen Metriken weitere Metriken, wie erweiterte Metriken zum Datenschutz und zur Kostenoptimierung, Aktivitätsmetriken und detaillierte Statuscode-Metriken. Erweiterte Metriken und Empfehlungen umfassen auch Empfehlungen zur Optimierung Ihrer Speichernutzung. Die Empfehlungen werden kontextbasiert neben relevanten Metriken im Dashboard platziert.

  Erweiterte Metriken und Empfehlungen umfassen die folgenden Funktionen:

- Erweiterte Metriken – Generieren Sie zusätzliche Metriken. Eine vollständige Liste der erweiterten Metrikkategorien finden Sie unter Metrikkategorien. Eine vollständige Liste der Metriken finden Sie unter Amazon S3-Storage-Lens-Metrikglossar.
- Amazon CloudWatch Publishing — Veröffentlicht S3 Storage Lens-Metriken CloudWatch , um einen einheitlichen Überblick über Ihren Betriebsstatus in CloudWatch Dashboards zu erhalten. Sie können auch CloudWatch API Betriebsabläufe und Funktionen wie Alarme und ausgelöste Aktionen, metrische Berechnungen und Anomalieerkennung verwenden, um S3 Storage Lens-Metriken zu überwachen und entsprechende Maßnahmen zu ergreifen. Weitere Informationen finden Sie unter Überwachen Sie S3 Storage Lens-Metriken in CloudWatch.
- Präfixaggregation – Metriken werden auf Präfixebene gesammelt. Durch die Aktivierung der Präfixaggregation werden alle in der Dashboard-Konfiguration enthaltenen Metriken auf Präfixebene erweitert. Metriken werden nur für Präfixe generiert, die den konfigurierten Schwellenwert erreichen. Beachten Sie, dass Metriken, die auf Präfixebene gelten, mit Präfixaggregation verfügbar sind. Ausgenommen sind Einstellungen auf Bucket-Ebene und Metriken zur Regelanzahl. Metriken auf Präfix-Ebene werden nicht veröffentlicht in. CloudWatch
- Aggregation von Storage-Lens-Gruppen – Metriken werden auf Storage-Lens-Gruppenebene gesammelt. Nachdem Sie Erweiterte Metriken und Empfehlungen und Aggregation von Storage-Lens-Gruppen aktiviert haben, können Sie angeben, welche Storage-Lens-Gruppen in das Storage-Lens-Dashboard aufgenommen oder daraus ausgeschlossen werden sollen. Es muss mindestens eine Storage-Lens-Gruppe angegeben werden. Die angegebenen Storage-Lens-Gruppen müssen sich in der angegebenen Heimatregion des Dashboard-Kontos befinden. Metriken auf Gruppenebene von Storage Lens werden nicht veröffentlicht. CloudWatch

Alle fortschrittlichen Metriken werden täglich gesammelt. Daten stehen für Abfragen 15 Monate lang zur Verfügung. Weitere Informationen zu den Speichermetriken, die von S3 Storage Lens aggregiert werden, finden Sie unter Amazon S3-Storage-Lens-Metrikglossar.

> **Note**
>
> Empfehlungen sind nur über das S3-Storage-Lens-Dashboard in der Amazon-S3-Konsole verfügbar.

S3 Storage Lens und AWS Organizations

AWS Organizations ist ein Tool AWS-Service , mit dem Sie alle Ihre Organisationshierarchie AWS-Konten unter einer einzigen Organisation zusammenfassen können. Amazon S3 Storage Lens bietet

AWS Organizations eine zentrale Ansicht des Objektspeichers und der Aktivitäten in Ihrem Amazon S3 S3-Speicher.

Weitere Informationen finden Sie unter Verwenden von Amazon S3 Storage Lens mit AWS Organizations.

- Vertrauenswürdiger Zugriff

  Sie müssen den vertrauenswürdigen Zugriff im Verwaltungskonto Ihrer Organisation aktivieren, damit S3 Storage Lens Speichermetriken und Nutzungsdaten für alle Mitgliedskonten in Ihrer Organisation aggregieren kann. Dann können Sie über das Verwaltungskonto Dashboards oder Exporte für Ihre Organisation erstellen oder delegierten Administratoren Zugriff auf andere Konten in der Organisation gewähren.

  Sie können den vertrauenswürdigen Zugriff für S3 Storage Lens jederzeit deaktivieren, um die Aggregation von Metriken für Ihre Organisation zu beenden.
- Delegated Administrator

  Sie können Dashboards und Metriken für S3 Storage Lens für Ihre Organisation erstellen, indem Sie Ihr AWS Organizations Verwaltungskonto verwenden oder indem Sie delegierten Administratorzugriff auf andere Konten in Ihrer Organisation gewähren. Sie können delegierte Administratoren jederzeit abmelden. Durch Abmelden eines delegierten Administrators wird die Aggregation neuer Speichermetriken durch alle Dashboards auf Organisationsebene, die von diesem delegierten Administrator erstellt wurden, automatisch beendet.

Weitere Informationen finden Sie unter Amazon S3 Storage Lens und AWS Organizations im AWS Organizations -Benutzerhandbuch.

Serviceverknüpfte Rollen für Amazon S3 Storage Lens

Neben dem AWS Organizations vertrauenswürdigen Zugriff verwendet Amazon S3 Storage Lens AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist ein einzigartiger IAM Rollentyp, der direkt mit S3 Storage Lens verknüpft ist. Serviceverknüpfte Rollen werden von S3 Storage Lens vordefiniert und enthalten alle Berechtigungen, die zum Erfassen der täglichen Speicher- und Aktivitätsmetriken aus Mitgliedskonten in Ihrer Organisation benötigt werden.

Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für Amazon S3 Storage Lens.

Amazon S3-Storage-Lens-Metrikglossar

Das Metrikglossar von Amazon S3 Storage Lens enthält eine vollständige Liste der kostenlosen und erweiterten Metriken für S3 Storage Lens.

S3 Storage Lens bietet kostenlose Metriken für alle Dashboards und Konfigurationen mit der Option zum Upgrade auf erweiterte Metriken.

- Kostenlose Metriken enthalten relevante Metriken für Ihre Speichernutzung, z. B. die Anzahl der Buckets und der Objekte in Ihrem Konto. Zu den kostenlosen Metriken gehören auch auf Anwendungsfällen basierende Metriken, etwa Metriken zur Kostenoptimierung und zum Datenschutz. Alle kostenlosen Metriken werden täglich erfasst und Daten stehen für Abfragen bis zu 14 Tage lang Daten zur Verfügung.
- Zu den erweiterten Metriken und Empfehlungen gehören neben den kostenlosen Metriken weitere Metriken, wie erweiterte Metriken zum Datenschutz und zur Kostenoptimierung. Zu den erweiterten Metriken gehören auch zusätzliche Metrikkategorien wie Aktivitätsmetriken und detaillierte Statuscode-Metriken. Erweiterte Metrikdaten stehen 15 Monate für Abfragen zur Verfügung.

  Für die Verwendung von S3 Storage Lens mit fortschrittlichen Metriken und Empfehlungen fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter Amazon S3 – Preise. Weitere Informationen über erweiterte Metriken und Empfehlungen finden Sie unter Metrikauswahl.

> **Note**
>
> Für Storage-Lens-Gruppen sind nur Speichermetriken des kostenlosen Kontingents verfügbar. Metriken für die Advanced-Stufe sind auf Storage-Lens-Gruppenebene nicht verfügbar.

Metriknamen

Die Spalte Metrikname in der folgenden Tabelle enthält den Namen jeder S3 Storage Lens in der S3-Konsole. Die Spalte CloudWatch und Export enthält den Namen jeder Metrik in Amazon CloudWatch und die Metrik-Exportdatei, die Sie in Ihrem S3 Storage Lens-Dashboard konfigurieren können.

Abgeleitete Metrikformeln

Abgeleitete Metriken sind für den Metrikexport und die CloudWatch Veröffentlichungsoption nicht verfügbar. Sie können jedoch die in der Spalte abgeleitete Metrikformeln angegebenen Metrikformeln verwenden, um sie zu berechnen.

Interpretieren der Präfix-Symbole von Amazon S3 Storage Lens für Vielfache von Metrikeinheiten (K, M, G usw.)

Metrikeinheiten von S3 Storage Lens werden mit mehreren Präfix-Symbolen geschrieben. Diese Präfixsymbole entsprechen den Symbolen des Internationalen Einheitensystems (SI), die vom Internationalen Büro für Maße und Gewichte (BIPM) standardisiert wurden. Diese Symbole werden auch im Einheitlichen Code für Maßeinheiten (UCUM) verwendet. Weitere Informationen finden Sie unter Liste der SI-Präfix-Symbole.

> **Note**
>
> - Die Maßeinheit für S3-Speicherbytes ist in binären Gigabyte (GB) angegeben, wobei 1 GB 2^{30} Byte, 1 TB 2^{40} Byte und 1 PB 2^{50} Byte entspricht. Diese Maßeinheit wird auch als Gibibyte (GiB) bezeichnet, wie von der Internationalen Elektrotechnischen Kommission () definiert. IEC
> - Wenn ein Objekt basierend auf seiner Lebenszykluskonfiguration das Ende seiner Lebensdauer erreicht hat, stellt Amazon S3 das Objekt zum Entfernen in eine Warteschlange und entfernt es asynchron. Daher kann es eine Verzögerung zwischen dem Ablaufdatum und dem Datum geben, an dem Amazon S3 ein Objekt entfernt. S3 Storage Lens enthält keine Metriken für Objekte, die abgelaufen sind, aber nicht entfernt wurden. Weitere Informationen über die Ablaufaktionen im S3-Lebenszyklus finden Sie unter Auslaufende Objekte.

Die folgende Tabelle zeigt das Glossar mit den Metriken von S3 Storage Lens.

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Alt	Abgeleitete Metrikmel
Gesamtspeicher	StorageBytes	Der Gesamtspeicher, einschlie	Kosti	Übe	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
		ßlich unvollstä ndiger mehrteili ger Uploads, Objektmetadaten und Löschmark ierungen				
Anzahl Objekte	ObjectCount	Die Gesamtzahl der Objekte	Kost i	Übe	N	-
Durchschn ittliche Objektgröße	-	Die durchschn ittliche Objektgrö ße	Kost i	Übe	Y	Summe (StorageB ytes) / Summe () ObjectCou nt
Aktive Buckets	-	Die Gesamtzah l der Buckets mit Speicher > 0 Byte	Kost i	Übe	Y	-
Buckets	-	Die Gesamtanzahl der Buckets	Kost i	Übe	Y	-
Konten	-	Die Anzahl der Konten, deren Speicher zum Bereich gehören	Kost i	Übe	Y	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
Aktuelle Versions-Bytes	CurrentVersionStorageBytes	Die Anzahl der Bytes, die eine aktuelle Version eines Objekts sind	Kost i	Kost imie	N	-
% Bytes der aktuellen Version	-	Der Prozentsatz der Bytes im Bereich, die eine aktuelle Version sind	Kost i	Kost imie	Y	sum (CurrentVersionStorageBytes) / sum () StorageBytes
Anzahl der Objekte der aktuellen Version	CurrentVersionObjectCount	Die Anzahl der aktuellen Versionsobjekte	Kost i	Date tz	N	-
% Objekte der aktuellen Version	-	Der Prozentsatz der Objekte im Bereich, die eine aktuelle Version sind	Kost i	Kost imie	Y	summe (CurrentVersionObjectCount) / summe () ObjectCount
Nicht aktuelle Versions-Bytes	NonCurrentVersionStorageBytes	Die Anzahl der Bytes der nicht aktuellen Version	Kost i	Kost imie	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
% Bytes der nicht aktuellen Version	-	Der Prozentsatz der Bytes im Bereich, die nicht aktuelle Versionen sind	Kost i	Kost imie	Y	summe (NonCurrentVersionStorageBytes) / summe () StorageBytes
Anzahl nicht aktueller Versionsobjekte	NonCurrentVersionObjectCount	Die Anzahl der Objekte der nicht aktuellen Versionen	Kost i	Kost imie	N	-
% Objekte der nicht aktuellen Version	-	Der Prozentsatz der Objekte im Bereich, die eine nicht aktuelle Version sind	Kost i	Kost imie	Y	summe (NonCurrentVersionObjectCount) / summe () ObjectCount
Löschmarkierungsbytes	DeleteMarkerStorageBytes	Die Anzahl der Bytes im Bereich, die Löschmarkierungen sind	Kost i	Kost imie	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
% Löschmark ierungsbytes	-	Der Prozentsa tz der Bytes im Bereich, die Löschmark ierungen sind	Kost i	Kost imie	Y	summe (DeleteMa rkerStora geBytes) / summe () StorageBy tes
Anzahl der Löschmark ierungsobjekte	DeleteMar kerObjectCount	Die Gesamtzah l der Objekte mit einer Löschmark ierung	Kost i	Kost imie	N	-
% Objekte mit Löschmark ierung	-	Der Prozentsatz der Objekte im Bereich mit einer Löschmarkierung	Kost i	Kost imie	Y	summe (DeleteMa rkerObjec tCount) / summe () ObjectCou nt
Unvollstä ndige Bytes für mehrteilige Uploads	Incomplet eMultipar tUploadSt orageBytes	Die Gesamtzah l der Bytes im Bereich für unvollständige mehrteilige Uploads	Kost i	Kost imie	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
% unvollstä ndige Bytes für mehrteilige Uploads	-	Der Prozentsa tz der Bytes im Bereich, die das Ergebnis unvollstä ndiger mehrteiliger Uploads sind	Kost i	Kost imie	Y	summe (Incomple teMultipa rtUploadS torageByt es) / summe () StorageBy tes
Anzahl unvollständiger mehrteiliger Uploads	Incomplet eMultipar tUploadOb jectCount	Die Anzahl der Objekte im Bereich, die unvollständige mehrteilige Uploads sind	Kost i	Kost imie	N	-
% unvollstä ndige mehrteili ge Uploads	-	Der Prozentsa tz der Objekte im Bereich, die unvollständige mehrteilige Uploads sind	Kost i	Kost imie	Y	summe (Incomple teMultipa rtUploadO bjectCoun t) / summe () ObjectCou nt

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
Unvollständige mehrteilige Upload-Speicherbytes, die älter als 7 Tage sind	In 7 ncomplete MPUStorage BytesOlderThan Tagen	Die Gesamtzahl der Bytes im Bereich für unvollständige mehrteilige Uploads, die älter als 7 Tage sind	Kosti	Kostimier	N	-
% unvollständige mehrteilige Upload-Speicherbytes, die älter als 7 Tage sind	-	Die Gesamtzahl der Bytes für unvollständige mehrteilige Uploads, die älter als 7 Tage sind	Kosti	Kostimier	Y	Summe (Incomplete MPUStorage BytesOlderThan 7 Tage) / Summe () StorageBytes
Anzahl der unvollständigen mehrteiligen Upload-Objekte, die älter als 7 Tage sind	l 7 Tage ncomplete MPUObject CountOlderThan	Die Anzahl der Objekte, die unvollständige mehrteilige Uploads und älter als 7 Tage sind	Kosti	Kostimier	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
% der Anzahl der unvollstä ndigen mehrteili gen Upload-Ob jekte, die älter als 7 Tage sind	-	Die Prozentsa tz der Objekte, die unvollstä ndige mehrteilige Uploads und älter als 7 Tage sind	Kost i	Kost imie	Y	Summe (I ncomplete MPUObject CountOlde rThan 7 Tage) / Summe () ObjectCou nt
Anzahl der Lebenszyk lusregeln für den Übergang	Transitio nLifecycl eRuleCount	Die Anzahl der Lebenszyk lusregeln für die Übertragung von Objekten in eine andere Speicherk lasse	Adva	Kost imie	N	-
Durchschn ittliche Lebenszyk lusregeln für den Übergang pro Bucket	-	Die durchschn ittliche Anzahl der Lebenszyk lusregeln für die Übertragung von Objekten in eine andere Speicherk lasse	Adva	Kost imie	Y	Summe (Transiti onLifecyc leRuleCou nt) / Summe () DistinctN umberOfBu ckets

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
Anzahl der Lebenszyk lusregeln für den Ablauf	Expiratio nLifecycl eRuleCount	Die Anzahl der Lebenszyk lusregeln für das Ablaufen von Objekten	Adva	Kost imie	N	-
Durchschn ittliche Lebenszyk lusregeln für den Ablauf pro Bucket	-	Die durchschn ittliche Anzahl der Lebenszyk lusregeln für das Ablaufen von Objekten	Adva	Kost imie	Y	summe (Expirati onLifecyc leRuleCou nt) / summe () DistinctN umberOfBu ckets
Anzahl der Lebenszyk lusregeln für den Übergang nicht aktueller Versionen	Noncurren tVersionT ransition Lifecycle RuleCount	Die Anzahl der Lebenszyk lusregeln für die Übertragung von nicht aktuellen Objektversionen in eine andere Speicherklasse	Adva	Kost imie	N	

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
Durchschn ittliche Lebenszyk lusregeln für den Übergang von nicht aktuellen Objektver sionen pro Bucket	-	Die durchschn ittliche Anzahl der Lebenszyk lusregeln für die Übertragung von nicht aktuellen Objektversionen in eine andere Speicherklasse	Adva	Kost imie	Y	summe (Noncurre ntVersion Transitio nLifecycl eRuleCoun t) / summe () DistinctN umberOfBu ckets
Anzahl der Lebenszyk lusregeln für den Ablauf nicht aktueller Versionen	Noncurren tVersionE xpiration Lifecycle RuleCount	Die Anzahl der Lebenszyk lusregeln für das Ablaufen von nicht aktuellen Objektversionen	Adva	Kost imie	N	-
Durchschn ittliche Lebenszyk lusregeln für den Ablauf von nicht aktuellen Objektver sionen pro Bucket	-	Die durchschn ittliche Anzahl der Lebenszyk lusregeln für das Ablaufen von nicht aktuellen Objektversionen	Adva	Kost imie	Y	summe (Noncurre ntVersion Expiratio nLifecycl eRuleCoun t) / summe () DistinctN umberOfBu ckets

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
Anzahl der Lebenszyk lusregeln zum Abbrechen unvollständiger mehrteiliger Uploads	AbortInco mpleteMPU Lifecycle RuleCount	Die Anzahl der Lebenszyk lusregeln zum Löschen unvollstä ndiger mehrteiliger Uploads	Adva	Kost imie	N	-
Die durchschn ittliche Anzahl von Lebenszyk lusregeln zum Abbruch unvollständiger mehrteiliger Uploads pro Bucket	-	Die durchschn ittliche Anzahl der Lebenszyk lusregeln zum Löschen unvollstä ndiger mehrteiliger Uploads	Adva	Kost imie	Y	summe (AbortInc ompleteMP ULifecycl eRuleCoun t) / summe () DistinctN umberOfBu ckets
Anzahl der Lebenszyk lusregeln für abgelaufene Objekte mit Löschmark ierung	ExpiredOb jectDelet eMarkerLi fecycleRu leCount	Anzahl der Lebenszyk lusregeln zum Entfernen abgelaufener Löschmark ierungen für Objekte	Adva	Kost imie	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
Durchschn ittliche Anzahl der Lebenszyk lusregeln zum Entfernen abgelaufener Löschmark ierungen für Objekte	-	Die durchschn ittliche Anzahl der Lebenszyk lusregeln zum Entfernen abgelaufener Löschmark ierungen für Objekte	Adva	Kost imie	Y	summe (ExpiredO bjectDele teMarkerL ifecycleR uleCount) / summe () DistinctN umberOfBu ckets
Anzahl der gesamten Lebenszyk lusregeln	TotalLife cycleRuleCount	Die Gesamtzah l der Lebenszyk lusregeln	Adva	Kost imie	N	-
Durchschn ittliche Anzahl von Lebenszyk lusregeln pro Bucket	-	Die durchschn ittliche Anzahl der Lebenszyk lusregeln	Adva	Kost imie	Y	summe (TotalLif ecycleRul eCount) / summe () DistinctN umberOfBu ckets
Verschlüsselte Bytes	Encrypted StorageBytes	Die Gesamtzah l der verschlüs selten Bytes	Kost i	Date tz	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
% verschlüsselte Bytes	-	Der Prozentsatz der Gesamtzahl von Bytes, die verschlüsselt sind	Kosti	Datetz	Y	summe (EncryptedObjectCount) / summe () StorageBytes
Anzahl der verschlüsselten Objekte	EncryptedObjectCount	Die Gesamtzahl der Objekte, die verschlüsselt sind	Kosti	Datetz	N	-
% verschlüsselte Objekte	-	Der Prozentsatz der Objekte, die verschlüsselt sind	Kosti	Datetz	Y	summe (EncryptedStorageBytes) / summe () ObjectCount
Unverschlüsselte Bytes	UnencryptedStorageBytes	Die Anzahl der unverschlüsselten Bytes	Kosti	Datetz	Y	summe (StorageBytes) - Summe () EncryptedStorageBytes

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
% unverschl üsselte Bytes	-	Der Prozentsa tz der Bytes, die unverschlüsselt sind	Kost i	Date tz	Y	Summe (UnencryptedStorageBytes) / Summe () StorageBytes
Unverschl üsselte Objektanzahl	UnencryptedObjectCount	Die Gesamtzahl der Objekte, die unverschlüsselt sind	Kost i	Date tz	Y	summe (ObjectCount) - Summe () EncryptedObjectCount
% unverschl üsselte Objekte	-	Der Prozentsa tz der unverschl üsselten Objekte	Kost i	Date tz	Y	Summe (UnencryptedStorageBytes) / Summe () ObjectCount
Replizierte Speicher-Bytes-Quelle	ReplicatedStorageBytesSource	Die Gesamtzahl der Bytes, die aus dem Quell-Bucket repliziert werden	Kost i	Date tz	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
% replizierte Bytes-Quelle	-	Der Prozentsatz der Gesamtzahl der Bytes, die aus dem Quell-Bucket repliziert werden	Kost i	Date tz	Y	summe (ReplicatedStorageBytesSource) / summe () StorageBytes
Quelle der Anzahl der replizierten Objekte	ReplicatedObjectCountSource	Die Anzahl der replizierten Objekte aus dem Quell-Bucket	Kost i	Date tz	N	-
% replizierte Objektquelle	-	Der Prozentsatz der Gesamtzahl der Objekte, die aus dem Quell-Bucket repliziert werden	Kost i	Date tz	Y	summe (ReplicatedStorageObjectCount) / summe () ObjectCount
Ziel der Replikationsspeicher-Bytes	ReplicatedStorageBytes	Die Gesamtzahl der Bytes, die in den Ziel-Bucket repliziert werden	Kost i	Date tz	Y	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
% repliziertes Bytes-Ziel	-	Der Prozentsatz der Gesamtzahl der Bytes, die in den Ziel-Bucket repliziert werden	Kost i	Date tz	Y	summe (ReplicatedStorageBytes) / summe () StorageBytes
Ziel der Anzahl der replizierten Objekte	ReplicatedObjectCount	Die Anzahl der Objekte, die in den Ziel-Bucket repliziert werden	Kost i	Date tz	Y	-
% Ziel replizierter Objekte	-	Der Prozentsatz der Gesamtzahl der Objekte, die in den Ziel-Bucket repliziert werden	Kost i	Date tz	Y	summe (ReplicatedObjectCount) / summe () ObjectCount

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
Bytes mit Objektsperre	ObjectLoc kEnabledS torageBytes	Die Gesamtzah l der Speicherb ytes mit aktivierter Objektsperre	Kost i	Date tz	Y	summe (Unencryp tedStorag eBytes) / summe (ObjectLo ckEnabled StorageCo unt) - summe () ObjectLoc kEnabledS torageByt es
% Bytes mit Objektsperre	-	Der Prozentsa tz der Speicherb ytes mit aktivierter Objektsperre	Kost i	Date tz	Y	Summe (ObjectLo ckEnabled StorageBy tes) / Summe () StorageBy tes
Anzahl der Objekte mit aktivierter Objektsperre	ObjectLoc kEnabledO bjectCount	Die Gesamtzah l der Objekte mit Objektsperre	Kost i	Date tz	Y	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
% Objekte mit Objektsperre	-	Der Prozentsatz der Gesamtzahl der Objekte mit aktivierter Objektsperre	Kost i	Date tz	Y	summe (ObjectLockEnabledObjectCount) / summe () ObjectCount
Bucket-Anzahl mit aktiviertem Versioning	VersioningEnabledBucketCount	Die Anzahl der Buckets, für die S3 Versioning aktiviert ist	Kost i	Date tz	N	-
% Buckets mit aktiviertem Versioning	-	Der Prozentsatz der Buckets, für die S3 Versioning aktiviert ist	Kost i	Date tz	Y	summe (VersioningEnabledBucketCount) / summe () DistinctNumberOfBuckets
MFAAnzahl der Buckets mit aktivierter Löschfunktion	MFADeleteEnabledBucketCount	Die Anzahl der Buckets, für die das Löschen MFA (Multi-Faktor-Authentifizierung) aktiviert ist	Kost i	Date tz	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
% MFA der Buckets, für die das Löschen aktiviert wurde	-	Der Prozentsatz der Buckets, für die das Löschen MFA (Multi-Faktor-Authentifizierung) aktiviert ist	Kost i	Date tz	Y	Summe (MFADelet eEnabledB ucketCoun t) / Summe () DistinctN umberOfBu ckets
SSE- Anzahl der Buckets KMS aktiviert	SSEKMSEna bledBucke tCount	Die Anzahl der Buckets, die serverseitige Verschlüsselung mit AWS Key Management Service Schlüssel n (SSE-KMS) für die Standard-Bucket-Ve rschlüsselung verwenden	Kost i	Date tz	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
% SSE — aktivierte Buckets KMS	-	Der Prozentsatz der Buckets, KMS für die SSE Standard-Bucket-Verschlüsselung	Kost i	Date tz	Y	sum (SSEKMSEnabledBucketCount) / sum () DistinctNumberOfBuckets
Alle nicht unterstützten Signaturanforderungen	AllUnsupportedSignatureRequests	Die Gesamtzahl der Anfragen, die nicht unterstützte Signaturversionen AWS verwenden	Adva	Date tz	N	-
% aller nicht unterstützten Signaturanforderungen	-	Der Prozentsatz der Anfragen, die nicht unterstützte Signaturversionen AWS verwenden	Adva	Date tz	Y	sum (AllUnsupportedSignatureRequests) / sum () AllRequests

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
Alle nicht unterstützten Anfragen TLS	AllUnsupp ortedTLSR equests	Die Anzahl der Anfragen, die nicht unterstüt zte Transport Layer Security () TLS -Versionen verwenden	Adva	Date tz	N	-
% aller nicht unterstützten Anfragen TLS	-	Der Prozentsatz der Anfragen, die nicht unterstützte TLS Versionen verwenden	Adva	Date tz	Y	sum (AllUnsup portedTLS Requests) / sum () AllReques ts
Alle SSE - Anfragen KMS	Ein llSSEKMSR equests	Die Gesamtzahl der Anfragen, die SSE - angeben KMS	Adva	Date tz	N	-
% aller SSE — KMS Anfragen	-	Der Prozentsatz der Anfragen, die SSE - angeben KMS	Adva	Date tz	Y	Summe (AllSSEKM SRequests) / Summe (AllReque sts)

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
Anzahl der Replikationsregeln für dieselbe Region	SameRegionReplicationRuleCount	Die Anzahl der Replikationsregeln für die Replikation in derselben Region () SRR	Adva	Datetz	N	-
Durchschnittliche Replikationsregeln für Replikation innerhalb derselben Region	-	Die durchschnittliche Anzahl von Replikationsregeln für SRR	Adva	Datetz	Y	sum (SameRegionReplicationRuleCount) / sum () DistinctNumberOfBuckets
Anzahl der regionsübergreifenden Replikationsregeln	CrossRegionReplicationRuleCount	Die Anzahl der Replikationsregeln für die regionsübergreifende Replikation () CRR	Adva	Datetz	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
Durchschn ittliche Replikati onsregeln für regionsüb ergreifende Replikation pro Bucket	-	Die durchschn ittliche Anzahl von Replikationsregeln für CRR	Adva	Date tz	Y	sum (CrossReg ionReplic ationRule Count) / sum () DistinctN umberOfBu ckets
Anzahl der Replikati onsregeln für dasselbe Konto	SameAccou ntReplica tionRuleCount	Die Anzahl der Replikationsregeln für die Replikati on innerhalb desselben Kontos	Adva	Date tz	N	-
Durchschn ittliche Replikati onsregeln für Replikati on innerhalb desselben Kontos pro Bucket	-	Die durchschn ittliche Anzahl der Replikationsregeln für die Replikati on innerhalb desselben Kontos	Adva	Date tz	Y	summe (SameAcco untReplic ationRule Count) / summe () DistinctN umberOfBu ckets

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
Kontenübergreifende Replikationsregelanzahl	CrossAccountReplicationRuleCount	Die Anzahl der Replikationsregeln für die kontoüber greifende Replikation	Adva	Datetz	N	-
Durchschnittliche Replikationsregeln für die kontoüber greifende Replikation pro Bucket	-	Die durchschnittliche Anzahl der Replikationsregeln für die kontoüber greifende Replikation	Adva	Datetz	Y	summe (CrossAccountReplicationRuleCount) / summe () DistinctNumberOfBuckets
Ungültige Anzahl der Zielreplikationsregeln	InvalidDestinationReplicationRuleCount	Die Anzahl der Replikationsregeln mit einem Replikationsziel, das nicht gültig ist	Adva	Datetz	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
Durchschn ittliche ungültige Anzahl der Zielrepli kationsregeln	-	Die durchschn ittliche Anzahl der Replikati onsregeln mit einem Replikati onsziel, das nicht gültig ist	Adva	Date tz	Y	summe (InvalidR eplicatio nRuleCoun t) / summe () DistinctN umberOfBu ckets
Gesamte Replikati onsregeln anzahl	-	Die Gesamtzah l der Replikati onsregeln	Adva	Date tz	Y	-
Durchschn ittliche Anzahl der Replikati onsregeln pro Bucket	-	Die durchschn ittliche Gesamtzah l der Replikati onsregeln	Adva	Date tz	Y	sum (alle Metriken zur Anzahl der Replikati onsregeln) / sum () DistinctN umberOfBu ckets

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
Anzahl der Buckets mit Bucket-Eigentümer erzwungen für Objekteigentümerschaft	ObjectOwnershipBucketOwnerEnforcedBucketCount	Die Gesamtzahl der Buckets, bei denen die Zugriffskontrolllisten (ACLs) mithilfe der vom Bucket-Besitzer erzwungenen Einstellung für Object Ownership deaktiviert wurden	Kosti	Zugr erwa	N	-
% Bucket-Eigentümer erzwungen für Objekteigentümerschaft	-	Der Prozentsatz der Buckets, die mithilfe der vom Bucket-Besitzer erzwungenen Einstellung für Objekteigentum ACLs deaktiviert wurden	Kosti	Zugr erwa	Y	Summe (ObjectOwnershipBucketOwnerEnforcedBucketCount) / Summe () DistinctNumberOfBuckets

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
Anzahl der Buckets mit Bucket-Eigentümer bevorzugt für Objekteigentümerschaft	ObjectOwnershipBucketOwnerPreferredBucketCount	Die Gesamtzahl der Buckets, die die Einstellung „Bucket-Eigentümer bevorzugt“ für die Objekteigentümerschaft verwenden	Kosti	Zugr erwa	N	-
% der Buckets mit Bucket-Eigentümer bevorzugt für Objekteigentümerschaft	-	Der Prozentsatz der Buckets, die die Einstellung „Bucket-Eigentümer bevorzugt“ für die Objekteigentümerschaft verwenden	Kosti	Zugr erwa	Y	summe (ObjectOwnershipBucketOwnerPreferredBucketCount) / summe () DistinctNumberOfBuckets
Objekt-Autor Bucket-Anzahl des Objekt-Eigentümers	ObjectOwnershipObjectWriterBucketCount	Die Gesamtzahl der Buckets, die die Einstellung „Objektschreiber“ für die Objekteigentümerschaft verwenden	Kosti	Zugr erwa	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
% der Buckets mit Objektsch reiber für Objekteig entümerschaft	-	Der Prozentsatz der Buckets, die die Einstellung „Objektschreiber“ für die Objekteig entümerschaft verwenden	Kost i	Zugr erwa	Y	summe (ObjectOw nershipOb jectWrite rBucketCo unt) / summe () DistinctN umberOfBu ckets
Anzahl aktiviert er Buckets für Transfer Acceleration	TransferA ccelerati onEnabled BucketCount	Die Gesamtzah l der Buckets, für die Transfer Acceleration aktiviert ist	Kost i	Leist	N	-
% der aktiviert en Buckets für Transfer Acceleration	-	Der Prozentsa tz der Buckets, für die Transfer Acceleration aktiviert ist	Kost i	Leist	Y	summe (Transfer Accelerat ionEnable dBucketCo unt) / summe () DistinctN umberOfBu ckets

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
Anzahl der aktivierten Buckets für Ereignisb enachrich tigungen	EventNoti ficationE nabledBuc ketCount	Die Gesamtzah l der Buckets, für die Ereignisb enachrichtigungen aktiviert sind	Kost i	Ereig e	N	
% der aktiviert en Buckets für Ereignisb enachrich tigungen	-	Der Prozentsa tz der Buckets, für die Ereignisb enachrichtigungen aktiviert sind	Kost i	Ereig e	Y	summe (EventNot ification EnabledBu cketCount) / summe () DistinctN umberOfBu ckets
Alle Anforderu ngen	AllRequests	Die Gesamtzahl der gesendeten - Anforderungen	Adva	Aktiv	N	-
GET-Anfor derungen	GetRequests	Die Gesamtzah l der gesendete n GET-Anforder ungen	Adva	Aktiv	N	-
PUT-Anfor derungen	PutRequests	Die Gesamtzah l der gesendete n PUT-Anforder ungen	Adva	Aktiv	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
HEAD-Anfo rderungen	HeadRequests	Die Gesamtzah l der gesendete n HEAD-Anforder ungen	Adva	Aktiv	N	-
DELETE-An forderungen	DeleteReq uests	Die Gesamtzah l der gesendeten DELETE-Anforder ungen	Adva	Aktiv	N	-
Listenanf orderungen	ListRequests	Die Gesamtzah l der gesendete n LIST-Anforder ungen	Adva	Aktiv	N	-
POST-Anfo rderungen	PostRequests	Die Gesamtzah l der gesendete n POST-Anforder ungen	Adva	Aktiv	N	-
SELECT-An forderungen	SelectRequests	Die Gesamtzahl der S3-SELECT-Anforderungen	Adva	Aktiv	N	-
Ausgewähl te gescannte Bytes	SelectSca nnedBytes	Die Anzahl der gescannten S3-SELECT-Bytes	Adva	Aktiv	N	-
Ausgewähl te zurückgeg ebene Bytes	SelectRet urnedBytes	Die Anzahl der zurückgegebenen S3-SELECT-Bytes	Adva	Aktiv	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
Heruntergeladene Bytes	BytesDownloaded	Die Anzahl der Bytes, die heruntergeladen wurden	Adva	Aktiv	N	-
% Abrufrate	-	Der Prozentsatz der Bytes, die heruntergeladen wurden	Adva	Aktiv	Y	summe (BytesDownloaded) / summe () StorageBytes
Hochgeladene Bytes	BytesUploaded	Die Anzahl der hochgeladenen Bytes	Adva	Aktiv	N	-
% Aufnahmeverhältnis	-	Der Prozentsatz der Bytes, die hochgeladen wurden	Adva	Aktiv	Y	summe (BytesUploaded) / summe () StorageBytes
4xx-Fehler	4 xxErrors	Die Gesamtzahl der HTTP 4xx-Statuscodes	Adva	Aktiv	N	-
5xx-Fehler	5 xxErrors	Die Gesamtzahl der HTTP 5xx-Statuscodes	Adva	Aktiv	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
Gesamtfehler	-	Die Summe aller 4xx- und 5xx-Fehler	Adva	Aktiv	Y	Summe (4xxErrors) + Summe (5xxErrors)
% Fehlerrate	-	Die Gesamtzahl der 4xx- und 5xx-Fehler als Prozentsatz der Gesamtanforderungen	Adva	Aktiv	Y	summe (TotalErrors) / summe () TotalRequests
200 OK Statusanzahl	200 OKStatusCount	Die Gesamtzahl der 200-OK-Statuscodes	Adva	Deta rter Statu e	N	-
% Status 200 OK	-	Die Gesamtzahl der 200-OK-Statuscodes als Prozentsatz der Gesamtzahl der Anforderungen	Adva	Deta rter Statu e	Y	Summe (200OKStatusCount) / Summe (AllRequests)
Statusanzahl 206 unvollständige Inhalte	206 PartialContentStatusCount	Die Gesamtzahl der Statuscodes 206 für unvollständige Inhalte	Adva	Deta rter Statu e	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
% Status 206 unvollständige Inhalte	-	Die Gesamtzahl der Statuscodes 206 für unvollständige Inhalte als Prozentsatz der Gesamtzahl der Anforderungen	Adva	Deta rter Statu e	Y	Summe (206PartialContentStatusCount) / Summe (AllRequests)
Anzahl der Fehler 400 Bad Request	400 BadRequestErrorCount	Die Gesamtzahl der Statuscodes 400 Bad Request	Adva	Deta rter Statu e	N	-
% Fehler 400 Bad Request	-	Die Gesamtzahl der Statuscodes 400 Bad Request als Prozentsatz der Gesamtzahl der Anforderungen	Adva	Deta rter Statu e	Y	Summe (400BadRequestErrorCount) / Summe (AllRequests)
Anzahl der Fehler 403 Forbidden	403 Forbidden ErrorCount	Die Gesamtzahl der Statuscodes 403 Forbidden	Adva	Deta rter Statu e	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
% Fehler 403 Forbidden	-	Die Gesamtzahl der Statuscodes 403 Forbidden als Prozentsatz der Gesamtzahl der Anforderungen	Adva	Deta rter Statu e	Y	Summe (403ForbiddenError Count) / Summe (AllRequests)
Anzahl der Fehler 404 Not Found	404 NotFoundErrorCount	Die Gesamtzahl der Statuscodes 404 Not Found	Adva	Deta rter Statu e	N	-
% Fehler 404 Not Found	-	Die Gesamtzahl der Statuscodes 404 Not Foundals Prozentsatz der Gesamtzahl der Anforderungen	Adva	Deta rter Statu e	Y	Summe (404NotFoundErrorCount) / Summe (AllRequests)
Fehleranzahl 500 Internal Server Error	500 InternalServerErrorCount	Die Gesamtzahl der Statuscodes 500 Internal Server Error	Adva	Deta rter Statu e	N	-

Metrikname	CloudWatch und exportieren	Beschreibung	Stufe [1]	Kate [2]	Ab t	Abge te Metri mel
% Fehler 500 Internal Server Error	-	Die Gesamtzahl der Statuscodes 500 Internal Server Error als Prozentsatz der Gesamtzahl der Anforderungen	Adva	Deta rter Statu e	Y	Summe (500InternalServerErrorCount) / Summe (AllRequests)
Anzahl der Fehler 503 Service Unavailable	503 ServiceUnavailableErrorCount	Die Gesamtzahl der Statuscodes 503 Service Unavailable	Adva	Deta rter Statu e	N	-
% Fehler 503 Service Unavailable	-	Die Gesamtzahl der Statuscodes 503 Service Unavailable als Prozentsatz der Gesamtzahl der Anforderungen	Adva	Deta rter Statu e	Y	Summe (503ServiceUnavailableErrorCount) / Summe (AllRequests)

[1] Alle Metriken für das kostenlose Speicherkontingent sind auf Storage-Lens-Gruppenebene verfügbar. Metriken für die Advanced-Stufe sind auf Storage-Lens-Gruppenebene nicht verfügbar.

[2] Metriken für die Anzahl der Regeln und Metriken für Bucket-Einstellungen sind auf Präfixebene nicht verfügbar.

Amazon S3 Storage Lens-Berechtigungen einrichten

Amazon S3 Storage Lens benötigt neue Berechtigungen in AWS Identity and Access Management (IAM), um den Zugriff auf S3 Storage Lens-Aktionen zu autorisieren. Um diese Berechtigungen zu gewähren, können Sie eine IAM identitätsbasierte Richtlinie verwenden. Sie können diese Richtlinie IAM Benutzern, Gruppen oder Rollen zuordnen, um ihnen Berechtigungen zu erteilen. Diese Berechtigungen können die Fähigkeit beinhalten, S3 Storage Lens zu aktivieren oder zu deaktivieren oder auf ein beliebiges Dashboard oder eine Konfiguration von S3 Storage Lens zuzugreifen.

Der IAM Benutzer oder die Rolle muss zu dem Konto gehören, das das Dashboard oder die Konfiguration erstellt hat oder besitzt, es sei denn, die beiden folgenden Bedingungen sind erfüllt:

- Ihr Konto ist Mitglied von AWS Organizations.
- Sie haben über Ihr Verwaltungskonto als delegierter Administrator Zugriff auf die Erstellung von Dashboards auf Organisationsebene erhalten.

> **Note**
>
> - Sie können Dashboards von Amazon S3 Storage Lens nicht mithilfe der Root–Benutzer-Anmeldeinformationen Ihres Kontos aufrufen. Um auf die S3 Storage Lens-Dashboards zugreifen zu können, müssen Sie einem neuen oder vorhandenen IAM Benutzer die erforderlichen IAM Berechtigungen erteilen. Anschließend können Sie sich mit diesen Anmeldeinformationen anmelden, um auf S3-Storage-Lens-Dashboards zuzugreifen. Weitere Informationen finden Sie unter Bewährte Sicherheitsmethoden IAM im IAM Benutzerhandbuch.
> - Die Verwendung von S3 Storage Lens in der Amazon S3-Konsole kann mehrere Berechtigungen erfordern. Zum Bearbeiten eines Dashboards in der Konsole benötigen Sie beispielsweise die folgenden Berechtigungen:
>   - `s3:ListStorageLensConfigurations`
>   - `s3:GetStorageLensConfiguration`
>   - `s3:PutStorageLensConfiguration`

Themen

- Festlegen von Kontoberechtigungen für die Verwendung von S3 Storage Lens

- Festlegen von Kontoberechtigungen für die Verwendung von S3-Storage-Lens-Gruppen
- Festlegen von Berechtigungen für die Verwendung von S3 Storage Lens mit AWS Organizations

Festlegen von Kontoberechtigungen für die Verwendung von S3 Storage Lens

Zum Erstellen und Verwalten von S3-Storage-Lens-Dashboards und Storage-Lens-Dashboard-Konfigurationen benötigen Sie je nachdem, welche Aktionen Sie ausführen möchten, die folgenden Berechtigungen:

Die folgende Tabelle zeigt die IAM Berechtigungen im Zusammenhang mit Amazon S3 Storage Lens.

Aktion	IAMBerechtigungen

Erstellen oder Aktualisieren eines S3-Storage-Lens-Dashboards in der Amazon S3-Konsole.	`s3:ListStorageLensConfigurations`
	`s3:GetStorageLensConfiguration`
	`s3:GetStorageLensConfigurationTagging`
	`s3:PutStorageLensConfiguration`
	`s3:PutStorageLensConfigurationTagging`

Abrufen der Tags eines S3-Storage-Lens-Dashboards in der Amazon-S3-Konsole.	`s3:ListStorageLensConfigurations` `s3:GetStorageLensConfigurationTagging`

Ansehen eines S3-Storage-Lens-Dashboards in der Amazon S3-Konsole.	`s3:ListStorageLensConfigurations`
	`s3:GetStorageLensConfiguration`
	`s3:GetStorageLensDashboard`

Löschen eines S3-Storage-Lens-Dashboards in der Amazon S3-Konsole.	`s3:ListStorageLensConfigurations` `s3:GetStorageLensConfiguration`

Aktion	IAMBerechtigungen
	`s3:DeleteStorageLensConfiguration`
Erstellen oder aktualisieren Sie eine S3-Storage-Lens-Konfiguration mit dem AWS CLI oder einem AWS SDK.	`s3:PutStorageLensConfiguration` `s3:PutStorageLensConfigurationTagging`
Rufen Sie die Tags einer S3 Storage Lens-Konfiguration ab, indem Sie AWS CLI oder ein verwenden AWS SDK.	`s3:GetStorageLensConfigurationTagging`
Zeigen Sie eine S3-Storage-Lens-Konfiguration an, indem Sie AWS CLI oder ein verwenden AWS SDK.	`s3:GetStorageLensConfiguration`
Löschen Sie eine S3-Storage-Lens-Konfiguration mithilfe von AWS CLI oder AWS SDK.	`s3:DeleteStorageLensConfiguration`

Note

- Sie können Ressourcen-Tags in einer IAM Richtlinie verwenden, um Berechtigungen zu verwalten.
- Ein IAM Benutzer oder eine Rolle mit diesen Berechtigungen kann Metriken aus Buckets und Präfixen sehen, für die er möglicherweise keine direkte Berechtigung hat, Objekte zu lesen oder aufzulisten.
- Wenn bei S3-Storage-Lens-Dashboards mit aktivierten Metriken auf Präfixebene ein ausgewählter Präfixpfad mit einem Objektschlüssel übereinstimmt, zeigt das Dashboard den Objektschlüssel möglicherweise als anderes Präfix an.
- Für Metrikexporte, die in einem Bucket in Ihrem Konto gespeichert sind, werden Berechtigungen mithilfe der in der IAM Richtlinie vorhandenen `s3:GetObject` Berechtigungen erteilt. Ähnlich können bei einer AWS Organizations Entität das Verwaltungskonto der Organisation oder delegierte Administratorkonten IAM Richtlinien

verwenden, um Zugriffsberechtigungen für Dashboards und Konfigurationen auf Organisationsebene zu verwalten.

Festlegen von Kontoberechtigungen für die Verwendung von S3-Storage-Lens-Gruppen

Mithilfe von S3-Storage-Lens-Gruppen können Sie die Speicherverteilung innerhalb von Buckets anhand von Präfix, Suffix, Objekt-Tag, Objektgröße oder Objektalter nachvollziehen. Sie können Storage-Lens-Gruppe an Dashboards anhängen, um ihre aggregierten Metriken anzuzeigen.

Für die Verwendung von Storage-Lens-Gruppen benötigen Sie bestimmte Berechtigungen. Weitere Informationen finden Sie unter the section called “Berechtigungen für Storage-Lens-Gruppen”.

Festlegen von Berechtigungen für die Verwendung von S3 Storage Lens mit AWS Organizations

Sie können Amazon S3 Storage Lens verwenden, um Speichermetriken und Nutzungsdaten für alle Konten zu sammeln, die Teil Ihrer AWS Organizations Hierarchie sind. Die folgende Tabelle zeigt die Aktionen und Berechtigungen im Zusammenhang mit der Verwendung von S3 Storage Lens mit Organizations.

Aktion	IAMBerechtigungen
Aktivieren des vertrauenswürdigen Zugriffs für S3 Storage Lens für Ihre Organisation.	`organizations:EnableAWSServiceAccess`
Deaktivieren des vertrauenswürdigen Zugriffs für S3 Storage Lens für Ihre Organisation.	`organizations:DisableAWSServiceAccess`
Registrieren eines delegierten Administrators für die Erstellung von S3-Storage-Lens-Dashboards oder -Konfigurationen für Ihre Organisation.	`organizations:RegisterDelegatedAdministrator`
Abmelden eines delegierten Administrators, damit dieser keine S3 Storage Lens-Dash	`organizations:DeregisterDelegatedAdministrator`

Aktion	IAMBerechtigungen
boards oder -Konfigurationen mehr für Ihre Organisation erstellen kann.	

Zusätzliche Berechtigungen zum Erstellen von organisationsweiten S3-Storage-Lens-Konfigurationen	`organizations:DescribeOrganization`
	`organizations:ListAccounts`
	`organizations:ListAWSServiceAccessForOrganization`
	`organizations:ListDelegatedAdministrators`
	`iam:CreateServiceLinkedRole`

Arbeiten mit Amazon S3 Storage Lens mithilfe der Konsole und API

Amazon S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Sie können Metriken von S3 Storage Lens verwenden, um zusammenfassende Erkenntnisse zu gewinnen und z. B. herauszufinden, wie viel Speicher Sie in Ihrer gesamten Organisation haben oder welche Buckets und Präfixe am schnellsten wachsen. Außerdem können Sie anhand der Metriken von S3 Storage Lens umfassende Möglichkeiten zur Kostenoptimierung aufdecken, bewährte Methoden für den Datenschutz implementieren und die Leistung von Anwendungs-Workloads verbessern. Sie können beispielsweise Buckets identifizieren, für die keine S3-Lebenszyklusregeln gelten, damit unvollständige mehrteilige Uploads, die älter als 7 Tage sind, ablaufen. Sie können auch Buckets identifizieren, die nicht den bewährten Datenschutzmethoden entsprechen, z. B. die Verwendung von S3 Replication oder S3 Versionierung. S3 Storage Lens analysiert Metriken, um kontextbezogene Empfehlungen zur Optimierung der Speicherkosten und zur Anwendung bewährter Datenschutzmethoden zu geben.

S3 Storage Lens aggregiert Ihre Metriken und zeigt die Informationen im Abschnitt Account snapshot (Konto-Snapshot) der Seite Buckets der Amazon-S3-Konsole an. S3 Storage Lens bietet außerdem ein interaktives Dashboard, mit dem Sie Erkenntnisse und Trends visualisieren, Ausreißer kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten erhalten und bewährte

Datenschutzmethoden anwenden können. Das Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Organisations-, Konto-, AWS-Region-, Speicherklassen-, Bucket-, Präfix- oder Objekt- oder Storage-Lens-Gruppenebene zu generieren. Sie können auch einen täglichen Metrikexport in CSV oder senden Parquet in einen S3-Bucket formatieren.

Dieser Abschnitt enthält Beispiele für das Erstellen, Aktualisieren und Anzeigen von S3-Storage-Lens-Konfigurationen und das Ausführen von funktionsbezogenen Vorgängen. Wenn Sie S3 Storage Lens mit verwenden AWS Organizations, decken diese Beispiele auch diese Anwendungsfälle ab. Ersetzen Sie alle Variablenwerte in den Beispielen durch Ihre eigenen Werte.

Themen

- Erstellen Sie ein Amazon S3 Storage Lens-Dashboard
- Aktualisieren Sie ein Amazon S3 Storage Lens-Dashboard
- Deaktivieren Sie ein Amazon S3 Storage Lens-Dashboard
- Löschen Sie ein Amazon S3 Storage Lens-Dashboard
- Amazon S3 Storage Lens-Dashboards auflisten
- Konfigurationsdetails eines Amazon S3 Storage Lens-Dashboards anzeigen
- Verwaltung von AWS Ressourcen-Tags mit S3 Storage Lens
- Hilfsdateien für die Verwendung von Amazon S3 Storage Lens

Erstellen Sie ein Amazon S3 Storage Lens-Dashboard

Sie können zusätzliche benutzerdefinierte S3 Storage Lens-Dashboards erstellen, die auf Ihr Unternehmen AWS Organizations oder auf bestimmte AWS-Regionen Bereiche innerhalb eines Kontos zugeschnitten werden können.

> **Note**
>
> Es kann bis zu 48 Stunden dauern, bis Änderungen an der Dashboard-Konfiguration genau angezeigt bzw. visualisiert werden.

Verwenden der S3-Konsole

Verwenden Sie die folgenden Schritte, um ein Amazon S3-Storage-Lens-Dashboard auf der Amazon S3-Konsole zu erstellen.

Schritt 1: Festlegen des Dashboard-Umfangs

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen der aktuell angezeigten AWS Region aus. Wählen Sie als Nächstes die Region aus, zu der Sie wechseln möchten.
3. Wählen Sie im linken Navigationsbereich unter S3 Storage Lens die Option Dashboards aus.
4. Klicken Sie auf Create dashboard (Dashboard erstellen).
5. Führen Sie auf der Seite Dashboard im Abschnitt Allgemein die folgenden Schritte aus:

   a. Sehen Sie sich die Heimatregion für Ihr Dashboard an. In der Heimatregion werden die AWS-Region Konfiguration und die Metriken für dieses Storage Lens-Dashboard gespeichert.

   b. Geben Sie einen Dashboard-Namen ein.

      Dashboard-Namen müssen weniger als 65 Zeichen lang sein und dürfen keine Sonderzeichen oder Leerzeichen enthalten.

      > **Note**
      >
      > Nach dem Erstellen des Dashboards kann der Name nicht mehr geändert werden.

   c. Sie haben die Wahl, Ihrem Dashboard Markierungen hinzuzufügen. Sie können Markierungen verwenden, um Berechtigungen für Ihr Dashboard zu verwalten und die Kosten für S3 Storage Lens nachzuverfolgen.

      Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Steuern des Zugriffs mithilfe von Ressourcen-Tags und unter AWS Generierte Kostenzuordnungs-Tags im AWS Billing Benutzerhandbuch.

      > **Note**
      >
      > Sie können Ihrer Dashboard-Konfiguration bis zu 50 Markierungen hinzufügen.

6. Führen Sie im Abschnitt Dashboard-Bereich die folgenden Schritte aus:

   a. Wählen Sie die Regionen und Buckets aus, die S3 Storage Lens im Dashboard ein- oder ausschließen soll.

b. Wählen Sie die Buckets in den ausgewählten Regionen aus, die S3 Storage Lens ein- oder ausschließen soll. Sie können Buckets entweder ein- oder ausschließen. Beides ist nicht möglich. Diese Option ist nicht verfügbar, wenn Sie Dashboards auf Organisationsebene erstellen.

> **Note**
>
> - Sie können Regionen und Buckets entweder ein- oder ausschließen. Diese Option ist nur auf Regionen beschränkt, wenn Sie Dashboards auf Organisationsebene für alle Mitgliedskonten in Ihrer Organisation erstellen.
> - Sie können bis zu 50 Buckets auswählen, die Sie einschließen oder ausschließen möchten.

Schritt 2: Konfigurieren der Metrikauswahl

1. Wählen Sie im Abschnitt Metrikauswahl den Metriktyp aus, den Sie für dieses Dashboard aggregieren möchten.

   - Wählen Sie Free metrics (Kostenlose Metriken) aus, um kostenlose Metriken einzuschließen, die auf Bucket-Ebene aggregiert und 14 Tage lang für Abfragen verfügbar sind.
   - Wählen Sie Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) aus, um erweiterte Metriken und andere erweiterte Optionen zu aktivieren. Zu diesen Optionen gehören erweiterte Präfixaggregation, CloudWatch Amazon-Veröffentlichung und kontextuelle Empfehlungen. Daten stehen für Abfragen für 15 Monate zur Verfügung. Erweiterte Metriken und Empfehlungen sind mit zusätzlichen Kosten verbunden. Weitere Informationen finden Sie unter Amazon S3 – Preise.

     Weitere Informationen zu erweiterten und zu kostenlosen Metriken finden Sie unter Metrikauswahl.

2. Wählen Sie unter Advanced metrics and recommendations features (Erweiterte Metrik- und Empfehlungsfunktionen) die Optionen aus, Sie aktivieren möchten:

   - Advanced metrics (Erweiterte Metriken)
   - CloudWatch veröffentlichen
   - Präfixzusammenfassung

> **Important**
>
> Wenn Sie die Präfixaggregation für Ihre S3 Storage Lens-Konfiguration aktivieren, werden Metriken auf Präfixebene nicht veröffentlicht. CloudWatch Nur S3 Storage Lens-Metriken auf Bucket-, Konto- und Organisationsebene werden veröffentlicht. CloudWatch

3. Wenn Sie Advanced Metrics (Erweiterte Metriken) aktiviert haben, wählen Sie die Advanced metrics categories (Erweiterte Metrikkategorien) aus, die Sie in Ihrem S3-Storage-Lens-Dashboard anzeigen möchten:

   - Metriken für Aktivitäten
   - Detailed status code metrics (Detaillierte Statuscode-Metriken)
   - Advanced cost optimization metrics (Erweiterte Kostenoptimierungsmetriken)
   - Advanced data protection metrics (Erweiterte Datensicherheitsmetriken)

   Weitere Informationen zu den Metrikkategorien finden Sie unter Metrikkategorien. Eine vollständige Liste der Metriken finden Sie unter Amazon S3-Storage-Lens-Metrikglossar.

4. Wenn Sie die Präfixaggregation aktivieren möchten, konfigurieren Sie Folgendes:

   a. Wählen Sie die Mindestgröße für den Präfixschwellenwert für dieses Dashboard aus.

      Ein Präfixschwellenwert von 5 Prozent gibt beispielsweise an, dass Präfixe, die einen Anteil des gesamten Bucket-Speichers von 5 Prozent oder mehr ausmachen, aggregiert werden.

   b. Wählen Sie die Präfixtiefe aus.

      Diese Einstellung gibt die maximale Anzahl von Ebenen an, bis zu denen die Präfixe ausgewertet werden. Die Präfixtiefe muss weniger als 10 betragen.

   c. Geben Sie ein Präfixtrennzeichen ein.

      Mit diesem Wert werden die einzelnen Präfixebenen identifiziert. Der Standardwert in Amazon S3 ist das Zeichen `/`, in Ihrer Speicherstruktur können jedoch ggf. andere Trennzeichen verwendet werden.

(Optional) Schritt 3: Exportieren von Metriken für das Dashboard

1. Wählen Sie im Abschnitt Metrics export (Metrik-Export) die Option Enable (Aktivieren) aus, um einen Metrik-Export zu erstellen, der jeden Tag in einem Ziel-Bucket Ihrer Wahl abgelegt wird.

   Der Metrikexport erfolgt in oder CSV Apache Parquet . Er enthält den gleichen Datenumfang wie Ihr S3-Storage-Lens-Dashboard ohne die Empfehlungen.
2. Wenn Sie den Metrikexport aktiviert haben, wählen Sie das Ausgabeformat für Ihren täglichen Metrikexport: CSVoder Apache Parquet.

   Parquet ist ein Open-Source-Dateiformat für Hadoop, das verschachtelte Daten in einem flachen Spaltenformat speichert.
3. Wählen Sie den S3-Ziel-Bucket für den Export Ihrer Metriken aus.

   Sie können einen Bucket im aktuellen Konto des S3-Storage-Lens-Dashboards auswählen. Sie können auch einen anderen auswählen, AWS-Konto wenn Sie über die Berechtigungen für den Ziel-Bucket und die Konto-ID des Besitzers des Ziel-Buckets verfügen.
4. Wählen Sie den S3-Ziel-Bucket (Format: `s3://bucket-name/prefix`) aus.

   Der Bucket muss sich in der Heimatregion Ihres S3-Storage-Lens-Dashboards befinden. In der S3-Konsole wird das Feld Destination bucket permission (Ziel-Bucket-Berechtigung) mit der Berechtigung angezeigt, die der Ziel-Bucket-Richtlinie von Amazon S3 hinzugefügt wird. Amazon S3 aktualisiert die Bucket-Richtlinie für den Ziel-Bucket, damit S3 Daten darin ablegen kann.
5. (Optional) Wenn Sie die serverseitige Verschlüsselung für Ihren Metriken aktivieren möchten, wählen Sie Specify an encryption key (Einen Verschlüsselungsschlüssel angeben) aus. Wählen Sie dann den Verschlüsselungstyp: Amazon S3 S3-verwaltete Schlüssel (SSE-S3) oder AWS Key Management Service Schlüssel (SSE-KMS).

   Sie können zwischen einem von Amazon S3 verwalteten Schlüssel (SSE-S3) und einem AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) wählen.
6. (Optional) Um einen AWS KMS Schlüssel anzugeben, müssen Sie einen KMS Schlüssel auswählen oder einen Schlüssel eingeben Amazon Resource Name (ARN).

   Wenn Sie einen kundenverwalteten Schlüssel auswählen, müssen Sie S3 Storage Lens in der AWS KMS -Schlüsselrichtlinie die Berechtigung zum Verschlüsseln erteilen. Weitere Informationen finden Sie unter Verwenden Sie einen AWS KMS key , um Ihre Metrikexporte zu verschlüsseln.
7. Klicken Sie auf Dashboard erstellen.

Mit dem AWS CLI

Example

Der folgende Beispielbefehl erstellt eine Amazon S3 Storage Lens-Konfiguration mit Tags. Wenn Sie diese Beispiele verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre Informationen.

```
aws s3control put-storage-lens-configuration --account-id=111122223333 --
config-id=example-dashboard-configuration-id --region=us-east-1 --storage-lens-
configuration=file://./config.json --tags=file://./tags.json
```

Example

Der folgende Beispielbefehl erstellt eine Amazon S3 Storage Lens-Konfiguration ohne Tags. Wenn Sie diese Beispiele verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre Informationen.

```
aws s3control put-storage-lens-configuration --account-id=222222222222 --config-
id=your-configuration-id --region=us-east-1 --storage-lens-configuration=file://./
config.json
```

Verwendung des AWS SDK for Java

Example — Erstellen und aktualisieren Sie eine Amazon S3 Storage Lens-Konfiguration

Das folgende Beispiel erstellt und aktualisiert eine Amazon S3 Storage Lens-Konfiguration in SDK für Java:

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.ActivityMetrics;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.CloudWatchMetrics;
import com.amazonaws.services.s3control.model.Format;
import com.amazonaws.services.s3control.model.Include;
```

```
import com.amazonaws.services.s3control.model.OutputSchemaVersion;
import com.amazonaws.services.s3control.model.PrefixLevel;
import com.amazonaws.services.s3control.model.PrefixLevelStorageMetrics;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.S3BucketDestination;
import com.amazonaws.services.s3control.model.SSES3;
import com.amazonaws.services.s3control.model.SelectionCriteria;
import com.amazonaws.services.s3control.model.StorageLensAwsOrg;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensDataExport;
import com.amazonaws.services.s3control.model.StorageLensDataExportEncryption;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateAndUpdateDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "111122223333";
        String exportAccountId = "Destination Account ID";
        String exportBucketArn = "arn:aws:s3:::destBucketName"; // The destination
 bucket for your metrics export must be in the same Region as your S3 Storage Lens
 configuration.
        String awsOrgARN = "arn:aws:organizations::123456789012:organization/o-
abcdefgh";
        Format exportFormat = Format.CSV;

        try {
            SelectionCriteria selectionCriteria = new SelectionCriteria()
                    .withDelimiter("/")
                    .withMaxDepth(5)
                    .withMinStorageBytesPercentage(10.0);
            PrefixLevelStorageMetrics prefixStorageMetrics = new
 PrefixLevelStorageMetrics()
                    .withIsEnabled(true)
                    .withSelectionCriteria(selectionCriteria);
            BucketLevel bucketLevel = new BucketLevel()
                    .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                    .withAdvancedCostOptimizationMetrics(new
 AdvancedCostOptimizationMetrics().withIsEnabled(true))
```

```
                    .withAdvancedDataProtectionMetrics(new
AdvancedDataProtectionMetrics().withIsEnabled(true))
                    .withDetailedStatusCodesMetrics(new
DetailedStatusCodesMetrics().withIsEnabled(true))
                    .withPrefixLevel(new
PrefixLevel().withStorageMetrics(prefixStorageMetrics));
            AccountLevel accountLevel = new AccountLevel()
                    .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                    .withAdvancedCostOptimizationMetrics(new
AdvancedCostOptimizationMetrics().withIsEnabled(true))
                    .withAdvancedDataProtectionMetrics(new
AdvancedDataProtectionMetrics().withIsEnabled(true))
                    .withDetailedStatusCodesMetrics(new
DetailedStatusCodesMetrics().withIsEnabled(true))
                    .withBucketLevel(bucketLevel);

            Include include = new Include()
                    .withBuckets(Arrays.asList("arn:aws:s3:::bucketName"))
                    .withRegions(Arrays.asList("us-west-2"));

            StorageLensDataExportEncryption exportEncryption = new
StorageLensDataExportEncryption()
                    .withSSES3(new SSES3());
            S3BucketDestination s3BucketDestination = new S3BucketDestination()
                    .withAccountId(exportAccountId)
                    .withArn(exportBucketArn)
                    .withEncryption(exportEncryption)
                    .withFormat(exportFormat)
                    .withOutputSchemaVersion(OutputSchemaVersion.V_1)
                    .withPrefix("Prefix");
            CloudWatchMetrics cloudWatchMetrics = new CloudWatchMetrics()
                    .withIsEnabled(true);
            StorageLensDataExport dataExport = new StorageLensDataExport()
                    .withCloudWatchMetrics(cloudWatchMetrics)
                    .withS3BucketDestination(s3BucketDestination);

            StorageLensAwsOrg awsOrg = new StorageLensAwsOrg()
                    .withArn(awsOrgARN);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                    .withId(configurationId)
                    .withAccountLevel(accountLevel)
                    .withInclude(include)
                    .withDataExport(dataExport)
```

```
                    .withAwsOrg(awsOrg)
                    .withIsEnabled(true);

            List<StorageLensTag> tags = Arrays.asList(
                    new StorageLensTag().withKey("key-1").withValue("value-1"),
                    new StorageLensTag().withKey("key-2").withValue("value-2")
            );

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfiguration(new
 PutStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withStorageLensConfiguration(configuration)
                    .withTags(tags)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Um sich einen besseren Überblick über Ihren Speicher zu verschaffen, können Sie eine oder mehrere S3-Storage-Lens-Gruppen erstellen und diese an das Dashboard anhängen. Eine Storage-Lens-Gruppe ist ein benutzerdefinierter Filter für Objekte, der auf Präfixen, Suffixen, Objekt-Tags, Objektgröße, Objektalter oder einer Kombination dieser Filter basiert.

Sie können S3-Storage-Lens-Gruppen verwenden, um detaillierte Einblicke in große freigegebene Buckets wie Data Lakes zu erhalten und fundiertere Geschäftsentscheidungen zu treffen. Sie können beispielsweise die Speicherzuweisung rationalisieren und die Erstellung von Kostenberichten

optimieren, indem Sie die Speichernutzung für einzelne Projekte und Kostenstellen innerhalb eines oder mehrerer Buckets nach bestimmten Objektgruppen aufschlüsseln.

Zur Nutzung von S3-Storage-Lens-Gruppen müssen Sie das Dashboard aktualisieren, um erweiterte Metriken und Empfehlungen zu verwenden. Weitere Informationen zur Verwendung von S3-Storage-Lens-Gruppen finden Sie unter the section called “Arbeiten mit Storage Lens-Gruppen”.

Aktualisieren Sie ein Amazon S3 Storage Lens-Dashboard

Das Standard-Dashboard von Amazon S3 Storage Lens ist`default-account-dashboard`. Dieses Dashboard ist von Amazon S3 vorkonfiguriert, damit Sie zusammengefasste Erkenntnisse und Trends der aggregierten kostenlosen und erweiterten Metriken Ihres gesamten Kontos in der Konsole visualisieren können. Sie können den Konfigurationsbereich des Standard-Dashboards nicht ändern, können aber die Metrikauswahl von den kostenlosen Metriken auf die kostenpflichtigen erweiterten Metriken und Empfehlungen upgraden, den optionalen Metrikexport konfigurieren oder sogar das Standard-Dashboard deaktivieren. Das Standard-Dashboard kann nicht gelöscht und nur deaktiviert werden. Weitere Informationen finden Sie unter Verwenden der S3-Konsole.

Verwenden der S3-Konsole

Verwenden Sie die folgenden Schritte, um ein Amazon S3-Storage-Lens-Dashboard auf der Amazon S3-Konsole zu aktualisieren.

Schritt 1: Aktualisieren des Dashboard-Umfangs

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens, Dashboards aus.
3. Wählen Sie das Dashboard aus, das Sie bearbeiten möchten, und wählen Sie dann Bearbeiten.

   Die Seite Edit dashboard (Dashboard bearbeiten) wird geöffnet.

> **Note**
>
> Folgendes können Sie nicht ändern:
>
> - Den Dashboard-Namen
> - Die Heimatregion

- Der Bereich des Standard-Dashboards, das für den Speicher Ihres gesamten Kontos eingerichtet ist

4. (Optional) Auf der Dashboard-Konfigurationsseite im Abschnitt General (Allgemein) können Sie Ihr Dashboard aktualisieren und Markierungen hinzufügen.

   Sie können Markierungen verwenden, um Berechtigungen für Ihr Dashboard zu verwalten und die Kosten für S3 Storage Lens nachzuverfolgen. Weitere Informationen finden Sie unter Steuern des Zugriffs mithilfe von Ressourcen-Tags im IAMBenutzerhandbuch und unter AWS Generierte Kostenzuweisungs-Tags im AWS Billing Benutzerhandbuch.

   > **Note**
   >
   > Sie können Ihrer Dashboard-Konfiguration bis zu 50 Markierungen hinzufügen.

5. Führen Sie im Abschnitt Dashboard-Bereich die folgenden Schritte aus:

   a. Aktualisieren Sie die Regionen und Buckets, die S3 Storage Lens im Dashboard ein- oder ausschließen soll.

      > **Note**
      >
      > - Sie können Regionen und Buckets entweder ein- oder ausschließen. Diese Option ist nur auf Regionen beschränkt, wenn Sie Dashboards auf Organisationsebene für alle Mitgliedskonten in Ihrer Organisation erstellen.
      > - Sie können bis zu 50 Buckets auswählen, die Sie einschließen oder ausschließen möchten.

   b. Aktualisieren Sie die Buckets in den ausgewählten Regionen, die S3 Storage Lens ein- oder ausschließen soll. Sie können Buckets entweder ein- oder ausschließen. Beides ist nicht möglich. Diese Option ist nicht vorhanden, wenn Sie Dashboards auf Organisationsebene erstellen.

Schritt 2: Aktualisieren der Metrikauswahl

1. Wählen Sie im Abschnitt Metrikauswahl den Metriktyp aus, den Sie für dieses Dashboard aggregieren möchten.

- Wählen Sie Free metrics (Kostenlose Metriken) aus, um kostenlose Metriken einzuschließen, die auf Bucket-Ebene aggregiert und 14 Tage lang für Abfragen verfügbar sind.
- Wählen Sie Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) aus, um erweiterte Metriken und andere erweiterte Optionen zu aktivieren. Zu diesen Optionen gehören erweiterte Präfixaggregation, CloudWatch Amazon-Veröffentlichung und kontextuelle Empfehlungen. Daten stehen für Abfragen für 15 Monate zur Verfügung. Erweiterte Metriken und Empfehlungen sind mit zusätzlichen Kosten verbunden. Weitere Informationen finden Sie unter Amazon S3 – Preise.

  Weitere Informationen zu erweiterten und zu kostenlosen Metriken finden Sie unter Metrikauswahl.

2. Wählen Sie unter Advanced metrics and recommendations features (Erweiterte Metrik- und Empfehlungsfunktionen) die Optionen aus, Sie aktivieren möchten:

   - Advanced metrics (Erweiterte Metriken)
   - CloudWatch veröffentlichen
   - Präfixzusammenfassung

   > **Important**
   >
   > Wenn Sie die Präfixaggregation für Ihre S3 Storage Lens-Konfiguration aktivieren, werden Metriken auf Präfixebene nicht veröffentlicht. CloudWatch Nur S3 Storage Lens-Metriken auf Bucket-, Konto- und Organisationsebene werden veröffentlicht. CloudWatch

3. Wenn Sie Advanced Metrics (Erweiterte Metriken) aktiviert haben, wählen Sie die Advanced metrics categories (Erweiterte Metrikkategorien) aus, die Sie in Ihrem S3-Storage-Lens-Dashboard anzeigen möchten:

   - Metriken für Aktivitäten
   - Detailed status code metrics (Detaillierte Statuscode-Metriken)
   - Advanced cost optimization metrics (Erweiterte Kostenoptimierungsmetriken)
   - Advanced data protection metrics (Erweiterte Datensicherheitsmetriken)

   Weitere Informationen zu Metrikkategorien finden Sie unter Metrikkategorien. Eine vollständige Liste der Metriken finden Sie unter Amazon S3-Storage-Lens-Metrikglossar.

4. Wenn Sie die Präfixaggregation aktivieren möchten, konfigurieren Sie Folgendes:

   a. Wählen Sie die Mindestgröße für den Präfixschwellenwert für dieses Dashboard aus.

      Ein Präfixschwellenwert von 5 Prozent gibt beispielsweise an, dass Präfixe, die einen Anteil des gesamten Bucket-Speichers von 5 Prozent oder mehr ausmachen, aggregiert werden.

   b. Wählen Sie die Präfixtiefe aus.

      Diese Einstellung gibt die maximale Anzahl von Ebenen an, bis zu denen die Präfixe ausgewertet werden. Die Präfixtiefe muss weniger als 10 betragen.

   c. Geben Sie ein Präfixtrennzeichen ein.

      Dies ist der Wert, mit dem die einzelnen Präfixebenen identifiziert werden. Der Standardwert in Amazon S3 ist das Zeichen `/`, in Ihrer Speicherstruktur können jedoch ggf. andere Trennzeichen verwendet werden.

(Optional) Schritt 3: Exportieren von Metriken für das Dashboard

1. Wählen Sie im Abschnitt Metrics export (Metrik-Export) die Option Enable (Aktivieren) aus, um einen Metrik-Export zu erstellen, der jeden Tag in einem Ziel-Bucket Ihrer Wahl abgelegt wird. Wenn Sie den Metrik-Export deaktivieren möchten, wählen Sie Disable (Deaktivieren) aus.

   Der Metrikexport erfolgt in oder CSV Apache Parquet . Er enthält den gleichen Datenumfang wie Ihr S3-Storage-Lens-Dashboard ohne die Empfehlungen.

2. Falls aktiviert, wählen Sie das Ausgabeformat für Ihren täglichen Metrikexport: CSVoder Apache Parquet.

   Parquet ist ein Open-Source-Dateiformat für Hadoop, das verschachtelte Daten in einem flachen Spaltenformat speichert.

3. Wählen Sie den S3-Ziel-Bucket für den Export Ihrer Metriken aus.

   Sie können einen Bucket im aktuellen Konto des S3-Storage-Lens-Dashboards auswählen. Sie können auch einen anderen auswählen, AWS-Konto wenn Sie über die Berechtigungen für den Ziel-Bucket und die Konto-ID des Besitzers des Ziel-Buckets verfügen.

4. Wählen Sie den S3-Ziel-Bucket (Format: `s3://`*`bucket-name`*`/`*`prefix`*) aus.

   Der Bucket muss sich in der Heimatregion Ihres S3-Storage-Lens-Dashboards befinden. In der S3-Konsole wird das Feld Destination bucket permission (Ziel-Bucket-Berechtigung) mit der

Berechtigung angezeigt, die der Ziel-Bucket-Richtlinie von Amazon S3 hinzugefügt wird. Amazon S3 aktualisiert die Bucket-Richtlinie für den Ziel-Bucket, damit S3 Daten darin ablegen kann.

5. (Optional) Wenn Sie die serverseitige Verschlüsselung für Ihren Metriken aktivieren möchten, wählen Sie Specify an encryption key (Einen Verschlüsselungsschlüssel angeben) aus. Wählen Sie dann den Verschlüsselungstyp: Amazon S3 S3-verwaltete Schlüssel (SSE-S3) oder AWS Key Management Service Schlüssel (SSE-KMS).

   Sie können zwischen einem von Amazon S3 verwalteten Schlüssel (SSE-S3) und einem AWS Key Management Service (AWS KMS) Schlüssel (SSE-KMS) wählen.

6. (Optional) Um einen AWS KMS Schlüssel anzugeben, müssen Sie einen KMS Schlüssel auswählen oder einen Schlüssel eingeben Amazon Resource Name (ARN). Geben Sie unter AWS KMS Schlüssel Ihren KMS Schlüssel auf eine der folgenden Arten an:

   - Um aus einer Liste verfügbarer KMS Schlüssel zu wählen, wählen Sie Wählen Sie aus Ihrem AWS KMS keys und wählen Sie Ihren KMSSchlüssel aus der Liste der verfügbaren Schlüssel aus.

     Sowohl der Von AWS verwalteter Schlüssel (`aws/s3`) als auch Ihr vom Kunden verwalteter Schlüssel werden in dieser Liste angezeigt. Weitere Informationen über vom Kunden verwaltete Schlüssel finden Sie unter Kundenschlüssel und AWS -Schlüssel im Entwicklerhandbuch zu AWS Key Management Service .

     > **Note**
     >
     > Das Von AWS verwalteter Schlüssel (`aws/S3`) wird für die SSE KMS Verschlüsselung mit S3 Storage Lens nicht unterstützt.

   - Um den KMS Schlüssel einzugebenARN, wählen Sie Enter AWS KMS key ARN und geben Sie Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
   - Um einen neuen, vom Kunden verwalteten Schlüssel in der AWS KMS Konsole zu erstellen, wählen Sie Create a KMS key aus.

     Wenn Sie einen kundenverwalteten Schlüssel auswählen, müssen Sie S3 Storage Lens in der AWS KMS -Schlüsselrichtlinie die Berechtigung zum Verschlüsseln erteilen. Weitere Informationen finden Sie unter Verwenden Sie einen AWS KMS key , um Ihre Metrikexporte zu verschlüsseln.

Weitere Informationen zum Erstellen eines finden Sie AWS KMS key unter Creating Keys im AWS Key Management Service Developer Guide.

7. Wählen Sie Änderungen speichern.

Um sich einen besseren Überblick über Ihren Speicher zu verschaffen, können Sie eine oder mehrere S3-Storage-Lens-Gruppen erstellen und diese an das Dashboard anhängen. Eine Storage-Lens-Gruppe ist ein benutzerdefinierter Filter für Objekte, der auf Präfixen, Suffixen, Objekt-Tags, Objektgröße, Objektalter oder einer Kombination dieser Filter basiert.

Sie können S3-Storage-Lens-Gruppen verwenden, um detaillierte Einblicke in große freigegebene Buckets wie Data Lakes zu erhalten und fundiertere Geschäftsentscheidungen zu treffen. Sie können beispielsweise die Speicherzuweisung rationalisieren und die Erstellung von Kostenberichten optimieren, indem Sie die Speichernutzung für einzelne Projekte und Kostenstellen innerhalb eines oder mehrerer Buckets nach bestimmten Objektgruppen aufschlüsseln.

Zur Nutzung von S3-Storage-Lens-Gruppen müssen Sie das Dashboard aktualisieren, um erweiterte Metriken und Empfehlungen zu verwenden. Weitere Informationen zur Verwendung von S3-Storage-Lens-Gruppen finden Sie unter the section called “Arbeiten mit Storage Lens-Gruppen”.

Mit dem AWS CLI

Example

Der folgende Beispielbefehl aktualisiert eine Amazon S3 Storage Lens-Dashboard-Konfiguration. Wenn Sie diese Beispiele verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre Informationen.

```
aws s3control put-storage-lens-configuration --account-id=111122223333 --
config-id=example-dashboard-configuration-id --region=us-east-1 --storage-lens-
configuration=file://./config.json --tags=file://./tags.json
```

Verwendung des AWS SDK for Java

Example — Aktualisieren Sie eine Amazon S3 Storage Lens-Konfiguration mit erweiterten Metriken und Empfehlungen

Die folgenden Beispiele zeigen Ihnen, wie Sie die Standardkonfiguration von S3 Storage Lens mit erweiterten Metriken und Empfehlungen SDK für Java aktualisieren können:

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.ActivityMetrics;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.Format;
import com.amazonaws.services.s3control.model.Include;
import com.amazonaws.services.s3control.model.OutputSchemaVersion;
import com.amazonaws.services.s3control.model.PrefixLevel;
import com.amazonaws.services.s3control.model.PrefixLevelStorageMetrics;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.S3BucketDestination;
import com.amazonaws.services.s3control.model.SSES3;
import com.amazonaws.services.s3control.model.SelectionCriteria;
import com.amazonaws.services.s3control.model.StorageLensAwsOrg;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensDataExport;
import com.amazonaws.services.s3control.model.StorageLensDataExportEncryption;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class UpdateDefaultConfigWithPaidFeatures {

    public static void main(String[] args) {
        String configurationId = "default-account-dashboard"; // This configuration ID
 cannot be modified.
        String sourceAccountId = "111122223333";

        try {
            SelectionCriteria selectionCriteria = new SelectionCriteria()
                    .withDelimiter("/")
                    .withMaxDepth(5)
                    .withMinStorageBytesPercentage(10.0);
```

```
            PrefixLevelStorageMetrics prefixStorageMetrics = new
 PrefixLevelStorageMetrics()
                    .withIsEnabled(true)
                    .withSelectionCriteria(selectionCriteria);
            BucketLevel bucketLevel = new BucketLevel()
                    .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                    .withPrefixLevel(new
 PrefixLevel().withStorageMetrics(prefixStorageMetrics));
            AccountLevel accountLevel = new AccountLevel()
                    .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                    .withBucketLevel(bucketLevel);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                    .withId(configurationId)
                    .withAccountLevel(accountLevel)
                    .withIsEnabled(true);

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfiguration(new
 PutStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withStorageLensConfiguration(configuration)
            );

        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

> **Note**
>
> Für erweiterte Metriken und Empfehlungen fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter Erweiterte Metriken und Empfehlungen.

Deaktivieren Sie ein Amazon S3 Storage Lens-Dashboard

Sie können ein Amazon S3 Storage Lens-Dashboard von der Amazon S3 S3-Konsole aus deaktivieren. Die Deaktivierung eines Dashboards verhindert, dass es in future Metriken generiert. Ein deaktiviertes Dashboard behält seine Konfigurationsinformationen, sodass es bei erneuter Aktivierung einfach wieder genutzt werden kann. Ein deaktiviertes Dashboard behält seine Verlaufsdaten bei, bis es nicht mehr für Abfragen verfügbar ist.

Verwenden der S3-Konsole

Gehen Sie wie folgt vor, um ein Amazon S3 Storage Lens-Dashboard auf der Amazon S3 S3-Konsole zu deaktivieren.

So deaktivieren Sie ein Amazon S3 Storage Lens-Dashboard

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie deaktivieren möchten, und gehen Sie dann oben in der Liste auf Deaktivieren.
4. Bestätigen Sie auf der Bestätigungsseite, dass Sie das Dashboard deaktivieren möchten, indem Sie den Namen des Dashboards in das Textfeld eingeben und dann Bestätigen auswählen.

Löschen Sie ein Amazon S3 Storage Lens-Dashboard

Das Standard-Dashboard können Sie nicht löschen. Eine Deaktivierung ist jedoch möglich. Bevor Sie ein selbst erstelltes Dashboard löschen, sollten Sie Folgendes beachten:

- Als Alternative zum Löschen können Sie das Dashboard deaktivieren, damit es in Zukunft wieder aktiviert werden kann. Weitere Informationen finden Sie unter Verwenden der S3-Konsole.
- Durch das Löschen eines Dashboards werden alle damit verbundenen Konfigurationseinstellungen gelöscht.

- Sobald Sie ein Dashboard löschen, können die historischen Metrikdaten nicht mehr abgerufen werden. Diese historischen Daten werden weiterhin für eine Dauer von 15 Monaten aufbewahrt. Wenn Sie wieder auf diese Daten zugreifen möchten, müssen Sie ein neues Dashboard mit dem gleichen Namen erstellen, dass sich in derselben Heimatregion befindet wie das gelöschte Dashboard.

Verwenden der S3-Konsole

Sie können ein Amazon S3 Storage Lens-Dashboard von der Amazon S3 S3-Konsole löschen. Das Löschen eines Dashboards verhindert jedoch, dass es in future Metriken generiert.

Löschen eines Amazon S3-Storage-Lens-Dashboards

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie löschen möchten, und gehen Sie dann oben in der Liste auf Löschen.
4. Bestätigen Sie auf der Seite Dashboards löschen, dass Sie das Dashboard löschen möchten, indem Sie den Namen des Dashboards in das Textfeld eingeben. Wählen Sie dann Confirm (Bestätigen) aus.

Verwenden Sie den AWS CLI

Example

Das folgende Beispiel löscht eine S3 Storage Lens-Konfiguration. Wenn Sie diese Beispiele verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre Informationen.

```
aws s3control delete-storage-lens-configuration --account-id=222222222222 --region=us-east-1 --config-id=your-configuration-id
```

Verwendung des AWS SDK for Java

Example — Löschen Sie eine Amazon S3 Storage Lens-Dashboard-Konfiguration

Das folgende Beispiel zeigt Ihnen, wie Sie eine S3 Storage Lens-Konfiguration mithilfe von SDK Java löschen:

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.DeleteStorageLensConfigurationRequest;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class DeleteDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "111122223333";
        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.deleteStorageLensConfiguration(new
 DeleteStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Amazon S3 Storage Lens-Dashboards auflisten

Verwenden der S3-Konsole

Um S3 Storage Lens-Dashboards aufzulisten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Navigieren Sie im linken Navigationsbereich zu Storage Lens.
3. Wählen Sie Dashboards. Sie können jetzt die Dashboards in Ihrem anzeigen. AWS-Konto

Mit dem AWS CLI

Example

Der folgende Beispielbefehl listet die S3 Storage Lens-Dashboards in Ihrem AWS-Konto auf. Wenn Sie diese Beispiele verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre Informationen.

```
aws s3control list-storage-lens-configurations --account-id=222222222222 --region=us-east-1 --next-token=abcdefghij1234
```

Example

Das folgende Beispiel listet S3 Storage Lens-Konfigurationen ohne ein nächstes Token auf. Wenn Sie diese Beispiele verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre Informationen.

```
aws s3control list-storage-lens-configurations --account-id=222222222222 --region=us-east-1
```

Verwendung des AWS SDK for Java

Example — Listet die Dashboard-Konfigurationen von S3 Storage Lens auf

Die folgenden Beispiele zeigen Ihnen, wie Sie S3 Storage Lens-Konfigurationen SDK für Java auflisten. Um dieses Beispiel zu verwenden, ersetzen Sie das *user input placeholders* durch Ihre eigenen Informationen.“ zu jeder Beispielbeschreibung.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
```

```
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.ListStorageLensConfigurationEntry;
import com.amazonaws.services.s3control.model.ListStorageLensConfigurationsRequest;

import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class ListDashboard {

    public static void main(String[] args) {
        String sourceAccountId = "111122223333";
        String nextToken = "nextToken";

        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            final List<ListStorageLensConfigurationEntry> configurations =
                    s3ControlClient.listStorageLensConfigurations(new
 ListStorageLensConfigurationsRequest()
                            .withAccountId(sourceAccountId)
                            .withNextToken(nextToken)
                    ).getStorageLensConfigurationList();

            System.out.println(configurations.toString());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Konfigurationsdetails eines Amazon S3 Storage Lens-Dashboards anzeigen

Sie können ein Amazon S3 Storage Lens-Dashboard von der Amazon S3 S3-Konsole aus und SDK für Java aufrufen. AWS CLI

Verwenden der S3-Konsole

Um die Konfigurationsdetails des S3 Storage Lens-Dashboards anzuzeigen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Navigieren Sie im linken Navigationsbereich zu Storage Lens.
3. Wählen Sie Dashboards aus.
4. Klicken Sie in der Liste der Dashboards auf das Dashboard, das Sie anzeigen möchten. Sie können jetzt die Details Ihres Storage Lens-Dashboards einsehen.

Mit dem AWS CLI

Example

Im folgenden Beispiel wird eine S3 Storage Lens-Konfiguration abgerufen, sodass Sie die Konfigurationsdetails anzeigen können. Wenn Sie diese Beispiele verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre Informationen.

```
aws s3control get-storage-lens-configuration --account-id=222222222222 --config-
id=your-configuration-id --region=us-east-1
```

Verwendung des AWS SDK for Java

Example — Rufen Sie eine S3 Storage Lens-Konfiguration ab und zeigen Sie sie an

Das folgende Beispiel zeigt Ihnen, wie Sie eine S3 Storage Lens-Konfiguration SDK für Java abrufen, sodass Sie die Konfigurationsdetails anzeigen können. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
```

```
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.GetStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.GetStorageLensConfigurationResult;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class GetDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "111122223333";

        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            final StorageLensConfiguration configuration =
                    s3ControlClient.getStorageLensConfiguration(new
 GetStorageLensConfigurationRequest()
                            .withAccountId(sourceAccountId)
                            .withConfigId(configurationId)
                    ).getStorageLensConfiguration();

            System.out.println(configuration.toString());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Verwaltung von AWS Ressourcen-Tags mit S3 Storage Lens

Jedes Amazon S3 Storage Lens-Dashboard wird als AWS Ressource mit eigenem Amazon-Ressourcennamen (ARN) gezählt. Daher können Sie bei der Konfiguration Ihres Storage Lens-Dashboards optional AWS Ressourcen-Tags zum Dashboard hinzufügen. Sie können bis zu 50 Tags für jedes Storage Lens-Dashboard hinzufügen. Um ein Storage Lens-Dashboard mit Tags zu erstellen, benötigen Sie die folgenden S3 Storage Lens-Berechtigungen:

- `s3:ListStorageLensConfigurations`
- `s3:GetStorageLensConfiguration`
- `s3:GetStorageLensConfigurationTagging`
- `s3:PutStorageLensConfiguration`
- `s3:PutStorageLensConfigurationTagging`

Sie können AWS Ressourcen-Tags verwenden, um Ressourcen nach Abteilung, Geschäftsbereich oder Projekt zu kategorisieren. Dies ist nützlich, wenn Sie viele Ressourcen desselben Typs haben. Durch das Anwenden von Tags können Sie anhand der Tags, die Sie ihm zugewiesen haben, schnell ein bestimmtes S3 Storage Lens-Dashboard identifizieren. Sie können Tags auch verwenden, um Ihre Kosten zu kategorisieren und zu verfolgen.

Wenn Sie Ihrem Storage Lens-Dashboard ein AWS Ressourcen-Tag hinzufügen, aktivieren Sie außerdem die attributbasierte Zugriffskontrolle (). ABAC ABACist eine Autorisierungsstrategie, die Berechtigungen auf der Grundlage von Attributen wie Tags definiert. Sie können in Ihren IAM Richtlinien auch Bedingungen verwenden, die Ressourcen-Tags angeben, um den Zugriff auf AWS Ressourcen zu kontrollieren.

Sie können Tag (Markierung)-Schlüssel und -Werte bearbeiten und Tags (Markierungen) jederzeit von einer Ressource entfernen. Beachten Sie folgende Einschränkungen:

- Bei Tag-Schlüsseln und Tag-Werten muss die Groß- und Kleinschreibung beachtet werden.
- Wenn Sie ein Tag (Markierung) mit demselben Schlüssel wie ein vorhandener Tag (Markierung) für die Ressource hinzufügen, wird der alte Wert mit dem neuen überschrieben.
- Wenn Sie eine Ressource löschen, werden alle Tags (Markierungen) der Ressource ebenfalls gelöscht.
- Nehmen Sie keine privaten oder vertraulichen Daten in Ihre AWS Ressourcen-Tags auf.
- System-Tags (mit Tag-Schlüsseln, die mit `aws:` beginnen) werden nicht unterstützt.

- Die einzelnen Tag-Schlüssel dürfen maximal 128 Zeichen lang sein. Die einzelnen Tag-Werte dürfen maximal 256 Zeichen lang sein.

Die folgenden Beispiele zeigen, wie Sie AWS Ressourcen-Tags mit dem Storage Lens-Dashboard verwenden.

Themen

- Fügen Sie AWS Ressourcen-Tags zu einem Storage Lens-Dashboard hinzu
- Rufen Sie AWS Ressourcen-Tags für ein Storage Lens-Dashboard ab
- Aktualisierung der Storage Lens-Dashboard-Tags
- Löschen von AWS Ressourcen-Tags aus einem S3 Storage Lens-Dashboard

Fügen Sie AWS Ressourcen-Tags zu einem Storage Lens-Dashboard hinzu

Die folgenden Beispiele zeigen, wie Sie einem S3 Storage Lens-Dashboard AWS Ressourcen-Tags hinzufügen. Sie können Ressourcen-Tags mithilfe der Amazon S3 S3-Konsole hinzufügen, AWS Command Line Interface (AWS CLI) und AWS SDK for Java.

Verwenden der S3-Konsole

Um AWS Ressourcen-Tags zu einem Storage Lens-Dashboard hinzuzufügen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Navigieren Sie im linken Navigationsbereich zu Storage Lens im linken Navigationsbereich.
3. Wählen Sie Dashboards.
4. Wählen Sie das Optionsfeld für das Storage Lens-Dashboard, das Sie aktualisieren möchten. Wählen Sie dann Edit (Bearbeiten) aus.
5. Wählen Sie unter Allgemein die Option Tag hinzufügen aus.
6. Fügen Sie auf der Seite Tag hinzufügen das neue Schlüssel-Wert-Paar hinzu.

> **Note**
>
> Wenn Sie ein neues Tag hinzufügen, dessen Schlüssel mit dem eines vorhandenen Tags identisch ist, wird der vorherige Tag-Wert überschrieben.

7. (Optional) Um mehrere neue Tags hinzuzufügen, wählen Sie erneut Tag hinzufügen aus und fügen Sie weiterhin neue Einträge hinzu. Sie können Ihrem Storage Lens-Dashboard bis zu 50 AWS Ressourcen-Tags hinzufügen.
8. (Optional) Wenn Sie einen neu hinzugefügten Eintrag entfernen möchten, wählen Sie neben dem zu entfernenden Tag Entfernen aus.
9. Wählen Sie Änderungen speichern.

Verwenden Sie den AWS CLI

Example

Der folgende Beispielbefehl fügt einer S3 Storage Lens-Dashboard-Konfiguration Tags hinzu. Wenn Sie diese Beispiele verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre Informationen.

```
aws s3control put-storage-lens-configuration-tagging --account-id=222222222222 --
region=us-east-1 --config-id=your-configuration-id --tags=file://./tags.json
```

Verwendung des AWS SDK for Java

Das folgende Beispiel fügt einer Amazon S3 Storage Lens-Konfiguration in SDK für Java Tags hinzu. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

Example — Fügen Sie einer S3 Storage Lens-Konfiguration Tags hinzu

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import
 com.amazonaws.services.s3control.model.PutStorageLensConfigurationTaggingRequest;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;
```

```
import static com.amazonaws.regions.Regions.US_WEST_2;

public class PutDashboardTagging {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "111122223333";

        try {
            List<StorageLensTag> tags = Arrays.asList(
                    new StorageLensTag().withKey("key-1").withValue("value-1"),
                    new StorageLensTag().withKey("key-2").withValue("value-2")
            );

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfigurationTagging(new
 PutStorageLensConfigurationTaggingRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withTags(tags)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Rufen Sie AWS Ressourcen-Tags für ein Storage Lens-Dashboard ab

Die folgenden Beispiele zeigen, wie Sie AWS Ressourcen-Tags für ein S3 Storage Lens-Dashboard abrufen. Sie können Ressourcen-Tags mithilfe der Amazon S3 S3-Konsole AWS Command Line Interface (AWS CLI) und abrufen AWS SDK for Java.

Verwenden der S3-Konsole

Um die AWS Ressourcen-Tags für ein Storage Lens-Dashboard abzurufen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Navigieren Sie im linken Navigationsbereich zu Storage Lens.
3. Wählen Sie Dashboards.
4. Wählen Sie das Optionsfeld für die Storage Lens-Dashboard-Konfiguration, die Sie anzeigen möchten. Wählen Sie dann „Dashboard-Konfiguration anzeigen“.
5. Überprüfen Sie unter Tags die mit dem Dashboard verknüpften Tags.
6. (Optional) Wenn Sie ein neues Tag hinzufügen möchten, wählen Sie Bearbeiten aus. Wählen Sie dann Tag hinzufügen aus. Fügen Sie auf der Seite „Tag hinzufügen“ das neue Schlüssel-Wert-Paar hinzu.

   > **Note**
   >
   > Wenn Sie ein neues Tag hinzufügen, dessen Schlüssel mit dem eines vorhandenen Tags identisch ist, wird der vorherige Tag-Wert überschrieben.

7. (Optional) Wenn Sie einen neu hinzugefügten Eintrag entfernen möchten, wählen Sie neben dem zu entfernenden Tag Entfernen aus.
8. Wählen Sie Änderungen speichern.

Unter Verwendung des AWS CLI

Example

Mit dem folgenden Beispielbefehl werden Tags für eine S3 Storage Lens-Dashboard-Konfiguration abgerufen. Wenn Sie diese Beispiele verwenden möchten, ersetzen Sie die *`user input placeholders`* durch Ihre Informationen.

```
aws s3control get-storage-lens-configuration-tagging --account-id=222222222222 --
region=us-east-1 --config-id=your-configuration-id --tags=file://./tags.json
```

Verwendung des AWS SDK for Java

Example — Ruft Tags für eine S3 Storage Lens-Dashboard-Konfiguration ab

Das folgende Beispiel zeigt Ihnen, wie Sie Tags für eine S3 Storage Lens-Dashboard-Konfiguration in SDK Java abrufen. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.DeleteStorageLensConfigurationRequest;
import
 com.amazonaws.services.s3control.model.GetStorageLensConfigurationTaggingRequest;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class GetDashboardTagging {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "111122223333";
        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            final List<StorageLensTag> s3Tags = s3ControlClient
                    .getStorageLensConfigurationTagging(new
 GetStorageLensConfigurationTaggingRequest()
                            .withAccountId(sourceAccountId)
                            .withConfigId(configurationId)
                    ).getTags();

            System.out.println(s3Tags.toString());
```

```
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Aktualisierung der Storage Lens-Dashboard-Tags

Die folgenden Beispiele zeigen, wie Sie die Dashboard-Tags von Storage Lens mithilfe der Amazon S3 S3-Konsole, AWS Command Line Interface (AWS CLI) und aktualisieren AWS SDK for Java.

Verwenden der S3-Konsole

Um ein AWS Ressourcen-Tag für ein Storage Lens-Dashboard zu aktualisieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Navigieren Sie im linken Navigationsbereich zu Storage Lens.
3. Wählen Sie Dashboards.
4. Wählen Sie das Optionsfeld für die Storage Lens-Dashboard-Konfiguration, die Sie anzeigen möchten. Wählen Sie dann „Dashboard-Konfiguration anzeigen“.
5. Überprüfen Sie unter Tags die mit dem Dashboard verknüpften Tags.
6. (Optional) Wenn Sie ein neues Tag hinzufügen möchten, wählen Sie Bearbeiten aus. Wählen Sie dann Tag hinzufügen aus. Fügen Sie auf der Seite „Tag hinzufügen“ das neue Schlüssel-Wert-Paar hinzu.

   > **Note**
   >
   > Wenn Sie ein neues Tag hinzufügen, dessen Schlüssel mit dem eines vorhandenen Tags identisch ist, wird der vorherige Tag-Wert überschrieben.

7. (Optional) Wenn Sie einen neu hinzugefügten Eintrag entfernen möchten, wählen Sie neben dem zu entfernenden Tag Entfernen aus.

8. Wählen Sie Änderungen speichern.

Unter Verwendung des AWS CLI

Example

Der folgende Beispielbefehl fügt Tags zu einer vorhandenen Amazon S3 Storage Lens-Dashboard-Konfiguration hinzu oder ersetzt sie. Wenn Sie diese Beispiele verwenden möchten, ersetzen Sie die *user input placeholders* durch Ihre Informationen.

```
aws s3control put-storage-lens-configuration-tagging --account-id=111122223333 --
config-id=example-dashboard-configuration-id --region=us-east-1 --config-id=your-
configuration-id
```

Verwendung des AWS SDK for Java

Das folgende AWS SDK for Java Beispiel aktualisiert die AWS Ressourcen-Tags in einem vorhandenen Storage Lens-Dashboard. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

Example — Aktualisieren Sie die Tags in einer vorhandenen Storage Lens-Dashboard-Konfiguration

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import
 com.amazonaws.services.s3control.model.PutStorageLensConfigurationTaggingRequest;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class PutDashboardTagging {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
```

```
        String sourceAccountId = "111122223333";

        try {
            List<StorageLensTag> tags = Arrays.asList(
                    new StorageLensTag().withKey("key-1").withValue("value-1"),
                    new StorageLensTag().withKey("key-2").withValue("value-2")
            );

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfigurationTagging(new
 PutStorageLensConfigurationTaggingRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withTags(tags)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Löschen von AWS Ressourcen-Tags aus einem S3 Storage Lens-Dashboard

Die folgenden Beispiele zeigen, wie Sie AWS Ressourcen-Tags aus einem vorhandenen Storage Lens-Dashboard löschen. Sie können Tags mithilfe der Amazon S3 S3-Konsole AWS Command Line Interface (AWS CLI) und löschen AWS SDK for Java.

Verwenden der S3-Konsole

Um AWS Ressourcen-Tags aus einem vorhandenen Storage Lens-Dashboard zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

2. Navigieren Sie im linken Navigationsbereich zu Storage Lens.
3. Wählen Sie Dashboards.
4. Wählen Sie das Optionsfeld für die Storage Lens-Dashboard-Konfiguration, die Sie anzeigen möchten. Wählen Sie dann „Dashboard-Konfiguration anzeigen“.
5. Überprüfen Sie unter Tags die mit dem Dashboard verknüpften Tags.
6. Wählen Sie neben dem Tag, das Sie entfernen möchten, die Option Entfernen aus.
7. Wählen Sie Änderungen speichern.

Verwenden Sie den AWS CLI

Mit dem folgenden AWS CLI Befehl werden AWS Ressourcen-Tags aus einem vorhandenen Storage Lens-Dashboard gelöscht. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

Example

```
aws s3control delete-storage-lens-configuration-tagging --account-id=222222222222 --
config-id=your-configuration-id --region=us-east-1
```

Verwendung des AWS SDK for Java

Im folgenden AWS SDK for Java Beispiel wird mithilfe des AWS Amazon-Ressourcennamens (ARN), den Sie im Konto *111122223333* angeben, ein Ressourcen-Tag aus dem Storage Lens-Dashboard gelöscht. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

Example — Löscht Tags für eine S3 Storage Lens-Dashboard-Konfiguration

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import
 com.amazonaws.services.s3control.model.DeleteStorageLensConfigurationTaggingRequest;
```

```
import static com.amazonaws.regions.Regions.US_WEST_2;

public class DeleteDashboardTagging {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "111122223333";
        try {
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.deleteStorageLensConfigurationTagging(new
 DeleteStorageLensConfigurationTaggingRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Hilfsdateien für die Verwendung von Amazon S3 Storage Lens

Verwenden Sie die folgenden JSON Dateien und ihre wichtigsten Eingaben für Ihre Beispiele.

Beispielkonfiguration für S3 Storage Lens in JSON

Example **`config.json`**

Die Datei `config.json` enthält Details einer S3-Storage-Lens-Konfiguration auf Organisationsebene mit erweiterten Metriken und Empfehlungen. Wenn Sie das folgende Beispiel verwenden möchten, ersetzen Sie die *`user input placeholders`* durch Ihre eigenen Informationen.

> **Note**
>
> Für erweiterte Metriken und Empfehlungen fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter Erweiterte Metriken und Empfehlungen.

```
{
  "Id": "SampleS3StorageLensConfiguration", //Use this property to identify your S3
 Storage Lens configuration.
  "AwsOrg": { //Use this property when enabling S3 Storage Lens for AWS Organizations.
    "Arn": "arn:aws:organizations::123456789012:organization/o-abcdefgh"
  },
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "DetailedStatusCodesMetrics": {
      "IsEnabled":true
    },
    "BucketLevel": {
      "ActivityMetrics": {
        "IsEnabled":true
      },
      "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
      },
      "AdvancedCostOptimizationMetrics": {
        "IsEnabled":true
      },
      "DetailedStatusCodesMetrics": {
        "IsEnabled":true
      },
      "PrefixLevel":{
        "StorageMetrics":{
          "IsEnabled":true,
          "SelectionCriteria":{
```

```
            "MaxDepth":5,
            "MinStorageBytesPercentage":1.25,
            "Delimiter":"/"
          }
        }
      }
    }
  },
  "Exclude": { //Replace with "Include" if you prefer to include Regions.
    "Regions": [
      "eu-west-1"
    ],
    "Buckets": [ //This attribute is not supported for AWS Organizations-level
 configurations.
      "arn:aws:s3:::amzn-s3-demo-source-bucket"
    ]
  },
  "IsEnabled": true, //Whether the configuration is enabled
  "DataExport": { //Details about the metrics export
    "S3BucketDestination": {
      "OutputSchemaVersion": "V_1",
      "Format": "CSV", //You can add "Parquet" if you prefer.
      "AccountId": "111122223333",
      "Arn": "arn:aws:s3:::
amzn-s3-demo-destination-bucket", // The destination bucket for your metrics export
 must be in the same Region as your S3 Storage Lens configuration.
      "Prefix": "prefix-for-your-export-destination",
      "Encryption": {
        "SSES3": {}
      }
    },
    "CloudWatchMetrics": {
      "IsEnabled": true
    }
  }
}
```

Beispielkonfiguration für S3 Storage Lens mit Storage Lens-Gruppen in JSON

Example **`config.json`**

Die Datei `config.json` enthält die Details, die Sie bei Verwendung von Storage-Lens-Gruppen auf die Storage-Lens-Konfiguration anwenden möchten. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie die *`user input placeholders`* durch Ihre Informationen.

Um alle Storage-Lens-Gruppen an das Dashboard anzuhängen, aktualisieren Sie die Storage-Lens-Konfiguration mit der folgenden Syntax:

```
{
  "Id": "ExampleS3StorageLensConfiguration",
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "BucketLevel": {
      "ActivityMetrics": {
      "IsEnabled":true
      },
    "StorageLensGroupLevel": {},
  "IsEnabled": true
}
```

Um nur zwei Storage Lens-Gruppen in Ihre Storage Lens-Dashboard-Konfiguration aufzunehmen (*`slg-1`* and *`slg-2`*), verwenden Sie die folgende Syntax:

```
{
  "Id": "ExampleS3StorageLensConfiguration",
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
```

```
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "BucketLevel": {
      "ActivityMetrics": {
      "IsEnabled":true
      },
   "StorageLensGroupLevel": {
        "SelectionCriteria": {
            "Include": [
                "arn:aws:s3:us-east-1:111122223333:storage-lens-group/slg-1",
                "arn:aws:s3:us-east-1:444455556666:storage-lens-group/slg-2"
            ]
    },
  "IsEnabled": true
}
```

Verwenden Sie die folgende Syntax, um bestimmte Storage-Lens-Gruppen vom Anhängen an die Dashboard-Konfiguration auszuschließen:

```
{
  "Id": "ExampleS3StorageLensConfiguration",
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
    },
    "BucketLevel": {
      "ActivityMetrics": {
      "IsEnabled":true
      },
   "StorageLensGroupLevel": {
        "SelectionCriteria": {
            "Exclude": [
                "arn:aws:s3:us-east-1:111122223333:storage-lens-group/slg-1",
                "arn:aws:s3:us-east-1:444455556666:storage-lens-group/slg-2"
            ]
```

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

Beispiel für die Konfiguration von S3-Storage-Lens-Tags in JSON

Example **`tags.json`**

Die Datei `tags.json` enthält die Tags, die Sie auf Ihre S3-Storage-Lens-Konfiguration anwenden möchten. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
[
    {
        "Key": "key1",
        "Value": "value1"
    },
    {
        "Key": "key2",
        "Value": "value2"
    }
]
```

IAMKonfigurationsberechtigungen für ein Beispiel für S3 Storage Lens

Example **`permissions.json`** – Spezifischer Dashboard-Name

Diese Beispielrichtlinie zeigt eine S3 Storage IAM `permissions.json` Lens-Datei mit einem bestimmten angegebenen Dashboard-Namen. Ersetzen *`Wert1`*, *`us-east-1your-dashboard-name`*, und *`example-account-id`* mit deinen eigenen Werten.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetStorageLensConfiguration",
                "s3:DeleteStorageLensConfiguration",
                "s3:PutStorageLensConfiguration"
            ],
```

```
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/key1": "value1"
                }
            },
            "Resource": "arn:aws:s3:us-east-1:example-account-id:storage-lens/your-
dashboard-name"
        }
    ]
}
```

Example **`permissions.json`** – Kein spezifischer Dashboard-Name

Diese Beispielrichtlinie zeigt eine S3 Storage IAM `permissions.json` Lens-Datei ohne Angabe eines bestimmten Dashboard-Namens. Ersetzen *`Wert1`*,*`us-east-1`*, und *`example-account-id`* mit deinen eigenen Werten.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetStorageLensConfiguration",
                "s3:DeleteStorageLensConfiguration",
                "s3:PutStorageLensConfiguration"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/key1": "value1"
                }
            },
            "Resource": "arn:aws:s3:us-east-1:example-account-id:storage-lens/*"
        }
    ]
}
```

Anzeigen von Metriken mit Amazon S3 Storage Lens

S3 Storage Lens aggregiert Ihre Metriken und zeigt die Informationen im Abschnitt Account snapshot (Konto-Snapshot) der Seite Buckets der Amazon-S3-Konsole an. S3 Storage Lens bietet

außerdem ein interaktives Dashboard, mit dem Sie Erkenntnisse und Trends visualisieren, Ausreißer kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten erhalten und bewährte Datenschutzmethoden anwenden können. Das Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Organisations-, Konto-, AWS-Region-, Speicherklassen-, Bucket-, Präfix- oder Objekt- oder Storage-Lens-Gruppenebene zu generieren. Sie können auch einen täglichen Metrikexport in CSV oder senden Parquet in einen S3-Bucket formatieren.

Standardmäßig sind alle Dashboards mit kostenlosen Metriken konfiguriert. Dazu gehören Metriken, anhand derer Sie die Nutzung und Aktivität in Ihrem S3-Speicher nachvollziehen, Ihre Speicherkosten optimieren und bewährte Methoden für Datenschutz und Zugriffsverwaltung implementieren können. Kostenlose Metriken werden bis auf Bucket-Ebene aggregiert. Bei kostenlosen Metriken stehen Daten bis zu 14 Tage lang für Abfragen zur Verfügung.

Zu den erweiterten Metriken und Empfehlungen gehören die folgenden zusätzlichen Funktionen, mit denen Sie weitere Einblicke in die Nutzung und Aktivität in Ihrem Speicher sowie bewährte Methoden zur Speicheroptimierung erhalten können:

- Kontextbezogene Empfehlungen (nur im Dashboard verfügbar)
- Erweiterte Metriken (einschließlich nach Buckets aggregierter Aktivitätsmetriken)
- Präfixzusammenfassung
- Aggregation von Storage-Lens-Gruppen
- Aggregation von Storage-Lens-Gruppen
- CloudWatch Amazon-Verlagswesen

Erweiterte Metrikdaten stehen 15 Monate für Abfragen zur Verfügung. Für die Verwendung von S3 Storage Lens mit fortschrittlichen Metriken fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter Amazon S3 – Preise. Weitere Informationen über kostenlose und erweiterte Metriken finden Sie unter Metrikauswahl.

Themen

- Anzeigen von S3-Storage-Lens-Metriken in den Dashboards
- Anzeigen von Amazon S3-Storage-Lens-Metriken mit einem Datenexport
- Überwachen Sie S3 Storage Lens-Metriken in CloudWatch
- Anwendungsfälle für Metriken von Amazon S3 Storage Lens

Anzeigen von S3-Storage-Lens-Metriken in den Dashboards

In der Amazon-S3-Konsole bietet S3 Storage Lens ein interaktives Standard-Dashboard, mit dem Sie Erkenntnisse zu Daten sowie entsprechende Trends visualisieren können. Zudem lässt sich das Dashboard verwenden, um Ausreißer zu kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten sowie zur Anwendung bewährter Datenschutzmethoden einzusehen. Das Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Konto-, Bucket-, AWS-Region-, Präfix- oder Storage-Lens-Gruppenebene zu generieren. Wenn Sie die Verwendung von S3 Storage Lens aktiviert haben AWS Organizations, können Sie auch Erkenntnisse auf Organisationsebene generieren (z. B. Daten für alle Konten, die Teil Ihrer AWS Organizations Hierarchie sind). Das Dashboard wird immer für das letzte Datum geladen, für das Metriken verfügbar sind.

Das Standard-Dashboard von S3 Storage Lens auf der Konsole hat den Namen default-account-dashboard. Gemäß Amazon-S3-Vorkonfiguration werden in diesem Dashboard die zusammengefassten Erkenntnisse und Trends Ihres gesamten Kontos visualisiert und täglich in der S3-Konsole aktualisiert. Sie können den Konfigurationsbereich des Standard-Dashboards nicht ändern, aber Sie haben die Möglichkeit, statt der kostenlosen Metriken die kostenpflichtigen erweiterten Metriken und Empfehlungen auszuwählen. Mit erweiterten Metriken und Empfehlungen können Sie auf zusätzliche Metriken und Funktionen zugreifen. Zu diesen Funktionen gehören erweiterte Metrikkategorien, Aggregation auf Präfixebene, kontextuelle Empfehlungen und Amazon-Veröffentlichungen. CloudWatch

Sie können das Standard-Dashboard deaktivieren, aber nicht löschen. Wenn Sie Ihr Standard-Dashboard deaktivieren, wird es nicht mehr aktualisiert. Sie erhalten keine neuen täglichen Metriken mehr in S3 Storage Lens oder im Konto-Snapshot auf der Seite Buckets. Sie können im Standard-Dashboard weiterhin historische Daten anzeigen, bis der Zeitraum von 14 Tagen für Datenabfragen abläuft. Wenn Sie erweiterte Metriken und Empfehlungen aktiviert haben, beträgt dieser Zeitraum 15 Monate. Sie können das Standard-Dashboard innerhalb des Ablaufzeitraums wieder aktivieren, um auf diese Daten zuzugreifen.

Sie können zusätzliche S3 Storage Lens-Dashboards erstellen und sie nach AWS-Regionen S3-Buckets oder Konten eingrenzen. Sie können Ihre Dashboards auch nach Organisation eingrenzen, wenn Sie die Verwendung von Storage Lens aktiviert haben. AWS Organizations Wenn Sie ein S3-Storage-Lens-Dashboard erstellen oder bearbeiten, definieren Sie den Dashboard-Umfang und die Metrikauswahl.

Sie können zusätzliche Dashboards, die Sie erstellen, deaktivieren oder löschen.

- Wenn Sie ein Dashboard deaktivieren, wird es nicht mehr aktualisiert und Sie erhalten keine neuen täglichen Metriken mehr. Sie können historische Daten für kostenlose Metriken dann noch bis zum Ende des Ablaufzeitraums von 14 Tagen einsehen. Wenn Sie erweiterte Metriken und Empfehlungen für das Dashboard aktiviert haben, beträgt dieser Zeitraum 15 Monate. Sie können das Dashboard innerhalb des Ablaufzeitraums wieder aktivieren, um auf diese Daten zuzugreifen.
- Wenn Sie ein Dashboard löschen, gehen alle Dashboard-Konfigurationseinstellungen verloren. Sie erhalten dann keine neuen täglichen Metriken mehr und können nicht mehr auf die mit dem Dashboard verbundenen historischen Daten zugreifen. Wenn Sie auf die historischen Daten eines gelöschten Dashboards zugreifen möchten, müssen Sie ein neues Dashboard mit demselben Namen in derselben Heimatregion erstellen.

Themen

- Anzeigen eines Amazon S3-Storage-Lens-Dashboards
- Grundlegendes zum S3-Storage-Lens-Dashboard

Anzeigen eines Amazon S3-Storage-Lens-Dashboards

Das folgende Verfahren zeigt, wie Sie ein S3-Storage-Lens-Dashboard in der S3-Konsole ansehen können. Anwendungsfallbasierte Anleitungen, die zeigen, wie Sie das Dashboard verwenden können, um Ihre Kosten zu optimieren, bewährte Verfahren zu implementieren und die Leistung von Anwendungen zu verbessern, die auf S3-Buckets zugreifen, finden Sie unter Anwendungsfälle für Metriken von Amazon S3 Storage Lens.

> **Note**
>
> Sie können Dashboards von Amazon S3 Storage Lens nicht mithilfe der Root–Benutzer-Anmeldeinformationen Ihres Kontos aufrufen. Um auf S3 Storage Lens-Dashboards zugreifen zu können, müssen Sie einem neuen oder vorhandenen IAM Benutzer die erforderlichen AWS Identity and Access Management (IAM) Berechtigungen erteilen. Anschließend können Sie sich mit diesen Anmeldeinformationen anmelden, um auf S3-Storage-Lens-Dashboards zuzugreifen. Weitere Informationen finden Sie unter Amazon S3 Storage Lens-Berechtigungen einrichten Bewährte Sicherheitsmethoden IAM im IAM Benutzerhandbuch.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie anzeigen möchten.

   Ihr Dashboard wird in S3 Storage Lens geöffnet. Der Abschnitt Snapshot for date (Snapshot für Datum) zeigt das letzte Datum an, an dem S3 Storage Lens Metriken gesammelt hat. Ihr Dashboard wird immer für das letzte Datum geladen, für das Metriken verfügbar sind.
4. (Optional) Wenn Sie das Datum für Ihr S3-Storage-Lens-Dashboard ändern möchten, wählen Sie in der oberen rechten Datumsauswahl ein neues Datum aus.
5. (Optional) Gehen Sie wie folgt vor, wenn Sie temporäre Filter anwenden möchten, mit denen Sie den Umfang Ihrer Dashboard-Daten weiter einschränken können:

   a. Erweitern Sie den Bereich Filter.
   b. Um nach bestimmten Konten, Speicherklassen, Buckets AWS-Regionen, Präfixen oder Storage Lens-Gruppen zu filtern, wählen Sie die Optionen aus, nach denen gefiltert werden soll.

      > **Note**
      >
      > Der Filter Präfixe und der Filter Storage-Lens-Gruppen können nicht gleichzeitig angewendet werden.

   c. Wenn Sie einen Filter aktualisieren möchten, wählen Sie Apply (Anwenden) aus.
   d. Zum Entfernen eines Filters klicken Sie auf das X neben dem Filter.
6. Wählen Sie zum Anzeigen von Daten für eine bestimmte Metrik in einem beliebigen Abschnitt Ihres S3-Storage-Lens-Dashboards im Feld Metric (Metrik) den Namen der entsprechenden Metrik aus.
7. In jedem Diagramm oder jeder Visualisierung des S3-Storage-Lens-Dashboards können Sie mithilfe der Registerkarten Konto, AWS-Regionen, Speicherklassen, Buckets, Präfixe oder Storage-Lens-Gruppen tiefere Aggregationsebenen aufrufen. Ein Beispiel finden Sie unter Entdecken Sie kalte Amazon S3-Buckets.

Grundlegendes zum S3-Storage-Lens-Dashboard

Ihr S3-Storage-Lens-Dashboard enthält eine primäre Registerkarte Overview (Übersicht) und bis zu fünf zusätzlichen Registerkarten für die einzelnen Aggregationsebenen:

- Konten

- AWS-Regionen
- Speicherklassen
- Buckets
- Präfixe
- Storage-Lens-Gruppen

Auf der Registerkarte Overview (Übersicht) werden Ihre Dashboard-Daten in drei verschiedene Abschnitte zusammengefasst: Snapshot for date (Snapshot für Datum), Trends and distributions (Trends und Verteilungen) und Top N overview (Top N-Übersicht).

Weitere Informationen zur Verwendung Ihres S3-Storage-Lens-Dashboards finden Sie in den folgenden Abschnitten.

Snapshot

Der Abschnitt Snapshot for date (Snapshot für Datum) zeigt zusammenfassende Metriken an, die S3 Storage Lens für das ausgewählte Datum gesammelt hat. Diese zusammenfassenden Metriken umfassen die folgenden Metriken:

- Gesamtspeicher – Die Gesamtmenge des verwendeten Speichers in Byte
- Anzahl der Objekte – Die Gesamtanzahl der Objekte im AWS-Konto
- Durchschnittliche Objektgröße – Die durchschnittliche Objektgröße
- Aktive Buckets – Die Gesamtanzahl der aktiven Buckets in aktiver Nutzung mit Speicher > 0 Byte im Konto
- Konten – Die Anzahl der Konten, deren Speicher zum Umfang gehören. Dieser Wert ist 1, es sei denn, Sie verwenden S3 Storage Lens AWS Organizations und Ihr S3 Storage Lens verfügt über einen vertrauenswürdigen Zugriff mit einer gültigen serviceverknüpften Rolle. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für Amazon S3 Storage Lens.
- Buckets – Die Gesamtanzahl der Buckets im Konto

Metrik-Daten

Für jede Metrik, die im Snapshot angezeigt wird, können Sie die folgenden Daten sehen:

- Metrikname – Der Name der Metrik

- Metrikkategorie – Die Kategorie, in der die Metrik organisiert ist
- Gesamtsumme für Datum – Die Gesamtzahl für das ausgewählte Datum
- % Änderung – Die prozentuale Änderung seit dem letzten Snapshot-Datum
- 30-Tage-Trend – Eine Trendlinie, die die Änderungen der Metrik über einen Zeitraum von 30 Tagen zeigt
- Empfehlung – Eine kontextbezogene Empfehlung, die auf den Daten basiert, die im Snapshot bereitgestellt werden. Empfehlungen sind mit erweiterten Metriken und Empfehlungen verfügbar. Weitere Informationen finden Sie unter Empfehlungen.

Metrikkategorien

Sie können optional Ihren Dashboard-Abschnitt Snapshot for date (Snapshot für Datum) aktualisieren, um Metriken für andere Kategorien anzuzeigen. Wenn Sie Snapshot-Daten für zusätzliche Metriken sehen möchten, können Sie aus den folgenden Metrics categories (Metrikkategorien) wählen:

- Kostenoptimierung
- Datenschutz
- Aktivität (mit erweiterten Metriken verfügbar)
- Zugriffsverwaltung
- Leistung
- Ereignisse

Im Abschnitt Snapshot for date (Snapshot für Datum) wird nur eine Auswahl von Metriken für jede Kategorie angezeigt. Um alle Metriken für eine bestimmte Kategorie anzuzeigen, wählen Sie die Metrik in den Abschnitten Trends and distributions (Trends und Verteilungen) oder Top N overview (Top N-Übersicht) aus. Weitere Informationen zu den Metrikkategorien finden Sie unter Metrikkategorien. Eine vollständige Liste der Metriken von S3 Storage Lens finden Sie unter Amazon S3-Storage-Lens-Metrikglossar.

Trends und Verteilungen

Der zweite Abschnitt der Registerkarte Overview (Übersicht) lautet Trends and distributions (Trends und Verteilungen). Im Abschnitt Trends and distributions (Trends und Verteilungen) können Sie zwei Metriken auswählen, um sie über einen von Ihnen definierten Zeitraum zu vergleichen. Der Abschnitt

Trends and distributions (Trends und Verteilungen) zeigt die Beziehung zwischen zwei Metriken im Zeitverlauf. Dieser Abschnitt enthält Diagramme, mit denen Sie die Verteilung von Storage class (Speicherklasse) und Region zwischen den beiden Trends sehen können, die Sie beobachten. Sie können optional einen Datenpunkt in einem der Diagramme aufschlüsseln, um ihn eingehender zu analysieren.

Eine Anleitung, die den Abschnitt Trends and distribution (Trends und Verteilungen) verwendet, finden Sie unter Identifizieren Sie Buckets, die keine serverseitige Verschlüsselung verwenden, mit AWS KMS der Standardverschlüsselung (-) SSE KMS.

Top N-Übersicht

Der dritte Abschnitt des S3-Storage-Lens-Dashboards ist die Top-N-Übersicht (in aufsteigender oder absteigender Reihenfolge sortiert). In diesem Abschnitt werden Ihre ausgewählten Metriken für die höchste Anzahl von Konten, Buckets AWS-Regionen, Präfixen oder Storage Lens-Gruppen angezeigt. Wenn Sie die Verwendung von S3 Storage Lens aktiviert haben AWS Organizations, können Sie auch Ihre ausgewählten Metriken in Ihrer gesamten Organisation sehen.

Eine Anleitung, die den Abschnitt Top N overview (Top N-Übersicht) verwendet, finden Sie unter Identifizieren Sie Ihre größten S3-Buckets.

Aufschlüsselung und Analyse nach Optionen

Um für eine reibungslose Analyse zu sorgen, bietet das S3-Storage-Lens-Dashboard ein Menü mit Aktionen, das bei Auswahl eines Diagrammwerts angezeigt wird. Wenn Sie dieses Menü verwenden möchten, wählen Sie einen beliebigen Diagrammwert aus, um die zugehörigen Metrikwerte zu sehen. Es stehen zwei Optionen zur Auswahl:

- Mit der Aktion Drill down (Aufgliedern) wird der ausgewählte Wert als Filter auf alle Registerkarten Ihres Dashboards angewendet. Sie können dann einen Drilldown für eine ausführlichere Analyse durchführen.
- Bei Auswahl der Aktion Analysieren nach wird die ausgewählte Registerkarte Dimension im Dashboard geöffnet und der Wert wird als Filter angewendet. Zu diesen Registerkarten gehören Konten, AWS-Regionen, Speicherklassen, Buckets, Präfixe (für Dashboards mit aktivierten Optionen Erweiterte Metriken und Präfixaggregation) sowie Storage-Lens-Gruppen (für Dashboards mit aktivierten Optionen Erweiterte Metriken und Aggregation von Storage-Lens-Gruppen). Mit Analysieren nach lassen sich die Daten im Kontext der neuen Dimension anzeigen und noch ausführlicher analysieren.

Die Aktionen Aufgliedern und Analysieren nach sind möglicherweise deaktiviert, wenn das Ergebnis unlogische Ergebnisse liefern würde oder keinen Wert hätte. Beide Aktionen Aufgliedern und Analysieren nach führen dazu, dass Filter auf alle Registerkarten des Dashboards angewendet werden (zusätzlich zu den vorhandenen Filtern). Bei Bedarf lassen sich Filter auch entfernen.

Registerkarten

Die Registerkarten auf Dimensionsebene bieten eine detaillierte Ansicht aller Werte innerhalb einer bestimmten Dimension. Die Registerkarte AWS-Regionen zeigt beispielsweise Metriken für alle AWS-Regionen an und die Registerkarte Buckets Metriken für alle Buckets. Jeder Dimensions-Tab hat das gleiche Layout, das aus vier Abschnitten besteht:

- Im Trenddiagramm werden Ihre Top-N-Elemente innerhalb der Dimension in den letzten 30 Tagen für die ausgewählte Metrik angezeigt. Standardmäßig enthält dieses Diagramm die Top 10 Elemente. Sie können die Anzahl auf minimal 3 Elemente verringern oder auf maximal 50 Elemente erhöhen.
- Das Histogramm enthält ein vertikales Balkendiagramm für die ausgewählte Kombination aus Datum und Metrik. Wenn Sie eine große Anzahl von Elementen in diesem Diagramm anzeigen möchten, müssen Sie ggf. horizontal scrollen.
- Im Blasen-Analysediagramm werden alle Elemente innerhalb der Dimension dargestellt. In diesem Diagramm wird die erste Metrik auf der X-Achse und die zweite Metrik auf der Y-Achse dargestellt. Die dritte Metrik wird durch die Größe der Blase repräsentiert.
- Die metrische Rasteransicht enthält jedes Element in der Dimension. Die Werte werden in Zeilen aufgeführt. Die Spalten stellen jede verfügbare Metrik dar, die zur einfacheren Navigation nach Tabs mit Metrikkategorien angeordnet sind.

Anzeigen von Amazon S3-Storage-Lens-Metriken mit einem Datenexport

Amazon S3 Storage Lens-Metriken werden täglich generiert in CSV oder Apache Parquet-formatierte Metriken exportieren Dateien und werden in einem S3-Bucket in Ihrem Konto platziert. Von dort aus können Sie den Metrikexport in die Analysetools Ihrer Wahl wie Amazon QuickSight und Amazon Athena aufnehmen, wo Sie die Speichernutzung und die Aktivitätstrends analysieren können.

Themen

- Verwenden Sie einen AWS KMS key , um Ihre Metrikexporte zu verschlüsseln
- Was ist ein S3-Storage-Lens-Exportmanifest?
- Grundlegendes zum Amazon S3-Storage-Lens-Exportschema

Verwenden Sie einen AWS KMS key , um Ihre Metrikexporte zu verschlüsseln

Um Amazon S3 Storage Lens die Berechtigung zum Verschlüsseln Ihrer Metrikexporte mit einem kundenverwalteten Schlüssel zu erteilen, müssen Sie eine Schlüsselrichtlinie verwenden. Gehen Sie wie folgt vor, um Ihre Schlüsselrichtlinie so zu aktualisieren, dass Sie einen KMS Schlüssel verwenden können, um Ihre S3 Storage Lens-Metrikexporte zu verschlüsseln.

Um S3 Storage Lens die Erlaubnis zu erteilen, Daten mithilfe Ihres Schlüssels zu verschlüsseln KMS

1. Melden Sie sich mit dem AWS Management Console vom Kunden verwalteten Schlüssel an AWS-Konto , dem der Kunde gehört.
2. Öffnen Sie die AWS KMS Konsole unter https://console.aws.amazon.com/kms.
3. Um das zu ändern AWS-Region, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
4. Klicken Sie im linken Navigationsbereich auf Customer managed keys (Vom Kunden verwaltete Schlüssel).
5. Wählen Sie unter Vom Kunden verwaltete Schlüssel den Schlüssel aus, den Sie zur Verschlüsselung der Metrikexporte verwenden möchten. AWS KMS keys sind regionsspezifisch und müssen sich in derselben Region befinden wie der S3-Bucket, in dem der Metrikexport ausgeführt wird.
6. Wählen Sie unter Key policy (Schlüsselrichtlinie) die Option Switch to policy view (Zur Richtlinienansicht wechseln) aus.
7. Um die Schlüsselrichtlinie zu aktualisieren, wählen Sie Edit (Bearbeiten).
8. Fügen Sie unter Edit key policy (Schlüsselrichtlinie bearbeiten) die folgende Schlüsselrichtlinie zu der vorhandenen Schlüsselrichtlinie hinzu. Wenn Sie diese Richtlinie verwenden möchten, ersetzen Sie *`user input placeholders`* durch eigene Informationen.

```
{
    "Sid": "Allow Amazon S3 Storage Lens use of the KMS key",
     "Effect": "Allow",
    "Principal": {
        "Service": "storage-lens.s3.amazonaws.com"
    },
    "Action": [
        "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition": {
```

```
        "StringEquals": {
            "aws:SourceArn": "arn:aws:s3:us-east-1:source-account-id:storage-
lens/your-dashboard-name",
            "aws:SourceAccount": "source-account-id"
         }
      }
}
```

9. Wählen Sie Änderungen speichern.

Weitere Informationen zum Erstellen von kundenverwalteten Schlüsseln und zum Verwenden von Schlüsselrichtlinien finden Sie unter den folgenden Themen im AWS Key Management Service Entwicklerhandbuch:

- Erste Schritte
- Verwenden der wichtigsten Richtlinien in AWS KMS

Sie können auch den API Vorgang für AWS KMS PUT wichtige Richtlinien verwenden (PutKeyPolicy), um die Schlüsselrichtlinie auf die vom Kunden verwalteten Schlüssel zu kopieren, die Sie zur Verschlüsselung der Metrikexporte verwenden möchten, indem Sie RESTAPI, AWS CLI, und SDKs verwenden.

Was ist ein S3-Storage-Lens-Exportmanifest?

Angesichts der großen Menge von aggregierten Daten kann der tägliche Metrik-Export von S3 Storage Lens in mehrere Dateien aufgeteilt werden. Die Manifestdatei `manifest.json` beschreibt, wo sich die Metrik-Exportdateien für den jeweiligen Tag befinden. Immer wenn ein neuer Export bereitgestellt wird, wird ein neues zugehöriges Manifest erstellt. Jedes in der `manifest.json`-Datei enthaltene Manifest bietet Metadaten und andere grundlegende Informationen zum Export.

Die Manifestinformationen umfassen die folgenden Eigenschaften:

- `sourceAccountId` – Die Konto-ID des Konfigurationsinhabers.
- `configId` – Eine eindeutige ID für das Dashboard.
- `destinationBucket`— Der Ziel-Bucket Amazon Resource Name (ARN), in den die Metriken exportiert werden.
- `reportVersion` – Die Version des Exports.
- `reportDate` – Das Datum des Berichts.

- `reportFormat` – Das Format des Berichts.
- `reportSchema` – Das Schema des Berichts.
- `reportFiles` – Die tatsächliche Liste der Exportberichtsdateien, die sich im Ziel-Bucket befinden.

Im Folgenden finden Sie ein Beispiel für ein Manifest in einer `manifest.json` Datei für einen Export im CSV -Format.

```
{
   "sourceAccountId":"123456789012",
   "configId":"my-dashboard-configuration-id",
   "destinationBucket":"arn:aws:s3:::amzn-s3-demo-destination-bucket",
   "reportVersion":"V_1",
   "reportDate":"2020-11-03",
   "reportFormat":"CSV",

 "reportSchema":"version_number,configuration_id,report_date,aws_account_number,aws_region,stor
   "reportFiles":[
        {
            "key":"DestinationPrefix/StorageLens/123456789012/my-dashboard-
configuration-id/V_1/reports/dt=2020-11-03/a38f6bc4-2e3d-4355-ac8a-e2fdcf3de158.csv",
            "size":1603959,
            "md5Checksum":"2177e775870def72b8d84febe1ad3574"
        }
      ]
}
```

Das Folgende ist ein Beispiel für ein Manifest in einer `manifest.json` Datei für ein Parquet-formatierter Export.

```
{
   "sourceAccountId":"123456789012",
   "configId":"my-dashboard-configuration-id",
   "destinationBucket":"arn:aws:s3:::amzn-s3-demo-destination-bucket",
   "reportVersion":"V_1",
   "reportDate":"2020-11-03",
   "reportFormat":"Parquet",
   "reportSchema":"message s3.storage.lens { required string version_number;
 required string configuration_id; required string report_date; required string
 aws_account_number; required string aws_region; required string storage_class;
```

```
 required string record_type; required string record_value; required string
 bucket_name; required string metric_name; required long metric_value; }",
   "reportFiles":[
      {
         "key":"DestinationPrefix/StorageLens/123456789012/my-dashboard-configuration-
id/V_1/reports/dt=2020-11-03/bd23de7c-b46a-4cf4-bcc5-b21aac5be0f5.par",
         "size":14714,
         "md5Checksum":"b5c741ee0251cd99b90b3e8eff50b944"
      }
}
```

Sie können Ihren Metrikexport so konfigurieren, dass er als Teil Ihrer Dashboard-Konfiguration in der Amazon S3-Konsole oder mithilfe von Amazon S3 RESTAPI, AWS CLI, und generiert wirdSDKs.

Grundlegendes zum Amazon S3-Storage-Lens-Exportschema

Die folgende Tabelle enthält das Schema Ihres S3-Storage-Lens-Metrik-Exports.

Attributname	Datentyp	Spaltenname	Beschreibung
`VersionNumber`	Zeichenfolge	`version_number`	Die Version der verwendeten S3-Storage-Lens-Metriken.
`ConfigurationId`	String	`configuration_id`	Die `configuration_id` Ihrer S3-Storage-Lens-Konfiguration.
`ReportDate`	String	`report_date`	Das Datum, an dem die Metriken erfasst wurden.
`AwsAccountNumber`	String	`aws_account_number`	Ihre AWS-Konto Nummer.
`AwsRegion`	String	`aws_region`	Die, AWS-Region für die die Metriken verfolgt werden.

Attributname	Datentyp	Spaltenname	Beschreibung
`StorageClass`	String	`storage_class`	Die Speicherklasse des betreffenden Buckets.
`RecordType`	ENUM	`record_type`	Die Art des Artefakts , das gemeldet wird (ACCOUNTBUCKET, oderPREFIX).
`RecordValue`	String	`record_value`	Der Wert des `RecordType` - Artefakts. **Note** Der `record_value` ist URL -kodiert.
`BucketName`	String	`bucket_name`	Der Name des Buckets, der gemeldet wird.
`MetricName`	Zeichenfolge	`metric_name`	Der Name der Metrik, die gemeldet wird.
`MetricValue`	Long	`metric_value`	Der Wert der Metrik, die gemeldet wird.

Beispiel für einen S3-Storage-Lens-Metrik-Export

Im Folgenden sehen Sie ein Beispiel für einen auf diesem Schema basierenden S3-Storage-Lens-Metrik-Export.

> **Note**
>
> Sie können Metriken für Storage-Lens-Gruppen ermitteln, indem Sie in der Spalte `record_type` nach dem Wert `STORAGE_LENS_GROUP_BUCKET` oder `STORAGE_LENS_GROUP_ACCOUNT` suchen. `record_value`In der Spalte wird der Amazon-Ressourcenname (ARN) für die Storage Lens-Gruppe angezeigt, zum Beispiel`arn:aws:s3:us-east-1:123456789012:storage-lens-group/slg-1`.

version_r	configuration_id	report_date	aws_account_number	aws_region	storage_class	record_type	record_value	bucket_name	metric_name	metric_value
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			StorageBytes	2478830621
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ObjectCount	1598962
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ReplicatedStorageBytes	20000
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ReplicatedObjectCount	20
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			EncryptedStorageBytes	2478828742
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			EncryptedObjectCount	1598961
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			DeleteMarkerObjectCount	1500
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ObjectLockEnabledStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			ObjectLockEnabledObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			CurrentVersionStorageBytes	2478830621
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			CurrentVersionObjectCount	1598962
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			NonCurrentVersionStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			NonCurrentVersionObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			IncompleteMultipartUploadStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	eu-west-1	STANDARD	ACCOUNT			IncompleteMultipartUploadObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	StorageBytes	29996800
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	ObjectCount	12264
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	ReplicatedStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	ReplicatedObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	EncryptedStorageBytes	29996800
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	EncryptedObjectCount	12264
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	DeleteMarkerObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	ObjectLockEnabledStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	ObjectLockEnabledObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	CurrentVersionStorageBytes	29996800
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	CurrentVersionObjectCount	12264
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	NonCurrentVersionStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	NonCurrentVersionObjectCount	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	IncompleteMultipartUploadStorageBytes	0
V_1	sample-cmh-exclude	11/3/2020	546264889236	us-west-1	STANDARD	PREFIX	AWSLogs%2F546264889236%2FCloudTrail%2Fu	cloudtrail-log-sf	IncompleteMultipartUploadObjectCount	0

Im Folgenden finden Sie ein Beispiel für einen S3-Storage-Lens-Metrikexport mit Storage-Lens-Gruppendaten.

version_numbe	configuration_id	report_date	aws_account_num	aws_region	storage_class	record_type	record_value	bucket_name	metric_name	metric_value
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		StorageBytes	3128548856
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ObjectCount	4440
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		EncryptedStorageBytes	3128548856
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		EncryptedObjectCount	4440
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		DeleteMarkerObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ObjectLockEnabledStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ObjectLockEnabledObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		CurrentVersionStorageBytes	3128548856
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		CurrentVersionObjectCount	4440
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		NonCurrentVersionStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		NonCurrentVersionObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMultipartUploadStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMultipartUploadObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		DeleteMarkerStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedStorageBytesSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		ReplicatedObjectCountSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMPUStorageBytesOlderThan7Days	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_ACCOUNT	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180		IncompleteMPUObjectCountOlderThan7Days	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	StorageBytes	676863200
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ObjectCount	3000
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	EncryptedStorageBytes	676863200
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	EncryptedObjectCount	3000
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	DeleteMarkerObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ObjectLockEnabledStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ObjectLockEnabledObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	CurrentVersionStorageBytes	676863200
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	CurrentVersionObjectCount	3000
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	NonCurrentVersionStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	NonCurrentVersionObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMultipartUploadStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMultipartUploadObjectCount	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	DeleteMarkerStorageBytes	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedStorageBytesSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	ReplicatedObjectCountSource	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMPUStorageBytesOlderThan7Days	0
V_1	sample-sfo-exclude	11/15/23	546264889236	us-west-1	STANDARD	STORAGE_LENS_GROUP_BUCKET	arn:aws:s3:us-west-1:546264889236:storage-lens-group/age-more-than-180	cloud-trail-log-sfo	IncompleteMPUObjectCountOlderThan7Days	0

Überwachen Sie S3 Storage Lens-Metriken in CloudWatch

Sie können S3 Storage Lens-Metriken auf Amazon veröffentlichen CloudWatch , um eine einheitliche Ansicht Ihres Betriebszustands in CloudWatchDashboards zu erstellen. Sie können auch CloudWatch Funktionen wie Alarme und ausgelöste Aktionen, metrische Berechnungen und Anomalieerkennung verwenden, um S3 Storage Lens-Metriken zu überwachen und entsprechende Maßnahmen zu ergreifen. Darüber hinaus ermöglichen CloudWatch API Betriebsabläufe Anwendungen, einschließlich Drittanbietern, den Zugriff auf Ihre S3 Storage Lens-Metriken. Weitere Informationen zu den CloudWatch Funktionen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

Sie können die CloudWatch Veröffentlichungsoption für neue oder bestehende Dashboard-Konfigurationen aktivieren, indem Sie die Amazon S3-Konsole, Amazon S3 REST API AWS CLI, und verwenden AWS SDKs. Dashboards, die auf erweiterte Metriken und Empfehlungen von S3 Storage Lens aktualisiert wurden, können die CloudWatch Veröffentlichungsoption verwenden. Informationen zu Preisen für erweiterte Metriken und Empfehlungen von S3 Storage Lens finden Sie unter Amazon-S3-Preise. Es fallen keine zusätzlichen Gebühren für die Veröffentlichung von CloudWatch Kennzahlen an. Es fallen jedoch andere CloudWatch Gebühren an, z. B. für Dashboards, Alarme und API Anrufe. Weitere Informationen finden Sie unter CloudWatchAmazon-Preise.

S3 Storage Lens-Metriken werden CloudWatch in dem Konto veröffentlicht, dem die S3 Storage Lens-Konfiguration gehört. Nachdem Sie die CloudWatch Veröffentlichungsoption in den erweiterten Metriken und Empfehlungen aktiviert haben, können Sie unter auf Kennzahlen auf Organisations-,

Konto- und Bucket-Ebene zugreifen. CloudWatch Metriken auf Präfix-Ebene sind in nicht verfügbar. CloudWatch

> **Note**
>
> S3 Storage Lens-Metriken sind Tagesmetriken und werden CloudWatch nur einmal täglich veröffentlicht. Wenn Sie S3 Storage Lens-Metriken in abfragen CloudWatch, muss der Zeitraum für die Abfrage 1 Tag (86400 Sekunden) betragen. Nachdem Ihre täglichen S3 Storage Lens-Metriken in Ihrem S3 Storage Lens-Dashboard in der Amazon S3 S3-Konsole angezeigt werden, kann es einige Stunden dauern, bis dieselben Messwerte angezeigt werden CloudWatch. Wenn Sie die CloudWatch Veröffentlichungsoption für S3 Storage Lens-Metriken zum ersten Mal aktivieren, kann es bis zu 24 Stunden dauern, bis Ihre Metriken veröffentlicht CloudWatch werden.

Nachdem Sie die CloudWatch Veröffentlichungsoption aktiviert haben, können Sie die folgenden CloudWatch Funktionen verwenden, um Ihre S3 Storage LensStorage Lens-Daten zu überwachen und zu analysieren:

- Dashboards — Verwenden Sie CloudWatch Dashboards, um benutzerdefinierte S3 Storage Lens-Dashboards zu erstellen. Teilen Sie Ihr CloudWatch Dashboard mit Personen, die keinen direkten Zugriff auf Ihr Dashboard haben AWS-Konto, teamübergreifend, mit Stakeholdern und mit Personen außerhalb Ihrer Organisation.
- Alarme und ausgelöste Aktionen – Konfigurieren Sie Alarme, die Metriken überwachen und Maßnahmen ergreifen, wenn ein Schwellenwert überschritten wird. Sie können beispielsweise einen Alarm konfigurieren, der eine SNS Amazon-Benachrichtigung sendet, wenn die Metrik Incomplete Multipart Upload Bytes an drei aufeinanderfolgenden Tagen 1 GB überschreitet.
- Anomalieerkennung – Ermöglichen Sie die Anomalieerkennung, um Metriken kontinuierlich zu analysieren, normale Baselines und Oberflächenanomalien zu ermitteln. Sie können einen Anomalieerkennungsalarm basierend auf dem erwarteten Wert einer Metrik erstellen. Sie können beispielsweise Anomalien für die Metrik Object Lock Enabled Bytes (Bytes mit aktivierter Objektsperre), um das unbefugte Entfernen von Objektsperreinstellungen zu erkennen.
- Mithilfe von Metrikberechnungen können Sie mehrere -Metriken mit S3-Storage-Lens-Metriken abfragen und mathematische Ausdrücke verwenden, um neue Zeitreihen basierend auf diesen Metriken zu erstellen. Sie können beispielsweise eine neue Metrik erstellen, um die durchschnittliche Objektgröße zu erhalten, indem Sie `StorageBytes` durch `ObjectCount` dividieren.

Weitere Informationen zur CloudWatch Veröffentlichungsoption für S3 Storage Lens-Metriken finden Sie in den folgenden Themen.

Themen

- Metriken und Dimensionen von S3 Storage Lens
- CloudWatch Veröffentlichung für S3 Storage Lens aktivieren
- Arbeiten mit S3 Storage Lens-Metriken in CloudWatch

Metriken und Dimensionen von S3 Storage Lens

Um S3 Storage Lens-Metriken an zu senden CloudWatch, müssen Sie die CloudWatch Veröffentlichungsoption in den erweiterten Metriken und Empfehlungen von S3 Storage Lens aktivieren. Nachdem die erweiterten Metriken aktiviert wurden, können Sie CloudWatchDashboards verwenden, um S3 Storage Lens-Metriken zusammen mit anderen Anwendungsmetriken zu überwachen und einen einheitlichen Überblick über Ihren Betriebsstatus zu erhalten. Sie können Dimensionen verwenden, um Ihre S3 Storage Lens-Metriken CloudWatch nach Organisation, Konto, Bereich, Speicherklasse, Region und Konfigurations-ID der Metriken zu filtern.

S3 Storage Lens-Metriken werden CloudWatch in dem Konto veröffentlicht, dem die S3 Storage Lens-Konfiguration gehört. Nachdem Sie die CloudWatch Veröffentlichungsoption in den erweiterten Metriken und Empfehlungen aktiviert haben, können Sie unter auf Kennzahlen auf Organisations-, Konto- und Bucket-Ebene zugreifen. CloudWatch Metriken auf Präfix-Ebene sind in nicht verfügbar. CloudWatch

> **Note**
>
> S3 Storage Lens-Metriken sind Tagesmetriken und werden CloudWatch nur einmal täglich veröffentlicht. Wenn Sie S3 Storage Lens-Metriken in abfragen CloudWatch, muss der Zeitraum für die Abfrage 1 Tag (86400 Sekunden) betragen. Nachdem Ihre täglichen S3 Storage Lens-Metriken in Ihrem S3 Storage Lens-Dashboard in der Amazon S3 S3-Konsole angezeigt werden, kann es einige Stunden dauern, bis dieselben Messwerte angezeigt werden CloudWatch. Wenn Sie die CloudWatch Veröffentlichungsoption für S3 Storage Lens-Metriken zum ersten Mal aktivieren, kann es bis zu 24 Stunden dauern, bis Ihre Metriken veröffentlicht CloudWatch werden.

Weitere Informationen zu S3 Storage Lens-Metriken und -Dimensionen finden Sie in CloudWatch den folgenden Themen.

Themen

- Metriken
- Dimensionen

Metriken

S3 Storage Lens-Metriken sind als Metriken innerhalb verfügbar CloudWatch. S3-Storage-Lens-Metriken werden auf dem `AWS/S3/Storage-Lens`-Namespace veröffentlicht. Dieser Namespace ist nur für S3-Storage-Lens-Metriken vorgesehen. Amazon-S3-Bucket, Anforderungs- und Replikations-Metriken werden im `AWS/S3`-Namespace veröffentlicht.

S3 Storage Lens-Metriken werden CloudWatch in dem Konto veröffentlicht, dem die S3 Storage Lens-Konfiguration gehört. Nachdem Sie die CloudWatch Veröffentlichungsoption in den erweiterten Metriken und Empfehlungen aktiviert haben, können Sie unter auf Kennzahlen auf Organisations-, Konto- und Bucket-Ebene zugreifen. CloudWatch Metriken auf Präfix-Ebene sind in nicht verfügbar. CloudWatch

In S3 Storage Lens werden Metriken aggregiert und nur in der dafür vorgesehenen Heimatregion gespeichert. S3 Storage Lens-Metriken werden auch CloudWatch in der Heimatregion veröffentlicht, die Sie in der S3 Storage Lens-Konfiguration angeben.

Eine vollständige Liste der S3 Storage Lens-Metriken, einschließlich einer Liste der in verfügbaren Metriken CloudWatch, finden Sie unterAmazon S3-Storage-Lens-Metrikglossar.

> **Note**
>
> Die gültige Statistik für S3 Storage Lens-Metriken in CloudWatch lautet Durchschnitt. Weitere Informationen zu Statistiken finden Sie CloudWatch unter CloudWatch Statistikdefinitionen im CloudWatch Amazon-Benutzerhandbuch.

Granularität der S3 Storage Lens-Metriken in CloudWatch

S3 Storage Lens bietet Metriken für Organisation, Konto, Bucket und Präfixgranularität. S3 Storage Lens veröffentlicht S3 Storage Lens-Metriken auf Organisations-, Konto- und Bucket-Ebene für. CloudWatch S3 Storage Lens-Metriken auf Präfix-Ebene sind in nicht verfügbar. CloudWatch

Weitere Informationen zur Granularität der S3 Storage Lens-Metriken, die in verfügbar sind CloudWatch, finden Sie in der folgenden Liste:

- Organisation – In den Mitgliedskonten Ihrer Organisation aggregierte Metriken. S3 Storage Lens veröffentlicht Metriken für Mitgliedskonten CloudWatch im Verwaltungskonto.
  - Organisation und Konto – Metriken für die Mitgliedskonten Ihrer Organisation.
  - Organisation und Bucket – Metriken für Amazon-S3-Buckets in den Mitgliedskonten Ihrer Organisation.
- Konto (Nicht-Organisationsebene) – Metriken, die in den Buckets Ihres Kontos zusammengefasst sind.
- Bucket (Nicht-Organisationsebene) – Metriken für einen bestimmten Bucket. In CloudWatch veröffentlicht S3 Storage Lens diese Metriken in der Datei AWS-Konto , die die S3 Storage Lens-Konfiguration erstellt hat. S3 Storage Lens veröffentlicht diese Metriken nur für nicht organisatorische Konfigurationen.

Dimensionen

Wenn S3 Storage Lens Daten an sendet CloudWatch, werden jeder Metrik Dimensionen zugewiesen. Dimensionen sind Kategorien, die die Eigenschaften von Metriken beschreiben. Sie können Dimensionen verwenden, um die zurückgegebenen Ergebnisse zu filtern. CloudWatch

Beispielsweise CloudWatch haben alle S3 Storage Lens-Metriken in die `configuration_id` Dimension. Sie können diese Dimension verwenden, um zwischen Metriken zu unterscheiden, die mit einer bestimmten S3-Storage-Lens-Konfiguration verknüpft sind. Die `organization_id` identifiziert Metriken auf Organisationsebene. Weitere Informationen zu Dimensionen in CloudWatch finden Sie unter Dimensionen im CloudWatch Benutzerhandbuch.

Abhängig von der Granularität der Metriken sind für S3-Storage-Lens-Metriken unterschiedliche Dimensionen verfügbar. Sie können die `organization_id` Dimension beispielsweise verwenden, um Metriken auf Organisationsebene nach der AWS Organizations ID zu filtern. Sie können diese Dimension jedoch nicht für Metriken auf Bucket- und Kontoebene verwenden. Weitere Informationen finden Sie unter Metriken mit Dimensionen filtern.

Auf der folgenden Tabelle können Sie erkennen, welche Dimensionen für Ihre S3-Storage-Lens-Konfiguration verfügbar sind.

Dimension	Beschreibung	Bucket	Account	Organization	Organisation und Bucket	Organisation und Konto
`configuration_id`	Der in den Metriken gemeldete Dashboard-Name für die S3-Storage-Lens-Konfiguration					
`metrics_version`	Die Version der S3-Storage-Lens-Metriken. Die Metrikversion hat einen festen Wert von `1.0`.					
`organization_id`	Die AWS Organizations ID für die Metriken					
`aws_account_number`	Die AWS-Konto , die mit den Metriken verknüpft ist					
`aws_region`	Das AWS-Region für die Metriken					
`bucket_name`	Der Name des in den Metriken gemeldeten S3-Buckets					
`storage_class`	Die Speicherklasse für den Bucket, die in den Metriken gemeldet wird					
`record_type`	Die Granularität der Metriken: ORGANIZATION,, ACCOUNT BUCKET	BUCKET	ACCOUNT	ORGANIZATION		

CloudWatch Veröffentlichung für S3 Storage Lens aktivieren

Sie können S3 Storage Lens-Metriken auf Amazon veröffentlichen CloudWatch , um eine einheitliche Ansicht Ihres Betriebszustands in CloudWatchDashboards zu erstellen. Sie können auch CloudWatch Funktionen wie Alarme und ausgelöste Aktionen, metrische Berechnungen und Anomalieerkennung verwenden, um S3 Storage Lens-Metriken zu überwachen und entsprechende Maßnahmen zu ergreifen. Darüber hinaus ermöglichen CloudWatch API Betriebsabläufe Anwendungen,

einschließlich Drittanbietern, den Zugriff auf Ihre S3 Storage Lens-Metriken. Weitere Informationen zu den CloudWatch Funktionen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

S3 Storage Lens-Metriken werden CloudWatch in dem Konto veröffentlicht, dem die S3 Storage Lens-Konfiguration gehört. Nachdem Sie die CloudWatch Veröffentlichungsoption in den erweiterten Metriken und Empfehlungen aktiviert haben, können Sie unter auf Kennzahlen auf Organisations-, Konto- und Bucket-Ebene zugreifen. CloudWatch Metriken auf Präfix-Ebene sind in nicht verfügbar. CloudWatch

Sie können die CloudWatch Unterstützung für neue oder bestehende Dashboard-Konfigurationen aktivieren, indem Sie die S3-Konsole REST APIs AWS CLI, Amazon S3 und verwenden AWS SDKs. Die CloudWatch Veröffentlichungsoption ist für Dashboards verfügbar, die auf erweiterte Metriken und Empfehlungen von S3 Storage Lens aktualisiert wurden. Informationen zu Preisen für erweiterte Metriken und Empfehlungen von S3 Storage Lens finden Sie unter Amazon-S3-Preise. Es fallen keine zusätzlichen Gebühren für die Veröffentlichung von CloudWatch Kennzahlen an. Es fallen jedoch andere CloudWatch Gebühren an, z. B. für Dashboards, Alarme und API Anrufe.

Informationen zur Aktivierung der CloudWatch Veröffentlichungsoption für S3 Storage Lens-Metriken finden Sie in den folgenden Themen.

> **Note**
>
> S3 Storage Lens-Metriken sind Tagesmetriken und werden bis zu CloudWatch einmal täglich veröffentlicht. Wenn Sie S3 Storage Lens-Metriken in abfragen CloudWatch, muss der Zeitraum für die Abfrage 1 Tag (86400 Sekunden) betragen. Nachdem Ihre täglichen S3 Storage Lens-Metriken in Ihrem S3 Storage Lens-Dashboard in der Amazon S3 S3-Konsole angezeigt werden, kann es einige Stunden dauern, bis dieselben Messwerte angezeigt werden CloudWatch. Wenn Sie die CloudWatch Veröffentlichungsoption für S3 Storage Lens-Metriken zum ersten Mal aktivieren, kann es bis zu 24 Stunden dauern, bis Ihre Metriken veröffentlicht CloudWatch werden.
> Derzeit können S3 Storage Lens-Metriken nicht über CloudWatch Streams abgerufen werden.

Verwenden der S3-Konsole

Wenn Sie ein S3-Storage-Lens-Dashboard aktualisieren, können Sie den Dashboard-Namen oder die Heimatregion nicht ändern. Sie können auch den Umfang des Standard-Dashboards nicht ändern, das auf den gesamten Speicher Ihres Kontos abgestimmt ist.

Um ein S3 Storage Lens-Dashboard zu aktualisieren, um die CloudWatch Veröffentlichung zu ermöglichen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich S3 Storage Lens und dann Dashboards aus.
3. Wählen Sie das Dashboard aus, das Sie bearbeiten möchten, und wählen Sie dann Bearbeiten.
4. Unter Metrikauswahl, wählen Sie Fortschrittliche Metriken und Empfehlungen aus.

   Fortschrittliche Metriken und Empfehlungen sind gegen Aufpreis erhältlich. Zu den erweiterten Kennzahlen und Empfehlungen gehören ein Zeitraum von 15 Monaten für Datenabfragen, auf Präfix-Ebene aggregierte Nutzungsmetriken, nach Bucket aggregierte Aktivitätsmetriken, die CloudWatch Veröffentlichungsoption und kontextuelle Empfehlungen, die Ihnen helfen, die Speicherkosten zu optimieren und bewährte Datenschutzpraktiken anzuwenden. Weitere Informationen finden Sie unter Amazon S3 – Preise.
5. Wählen Sie unter Erweiterte Funktionen für Metriken und Empfehlungen auswählen die Option Veröffentlichung aus. CloudWatch

   > **Important**
   >
   > Wenn Ihre Konfiguration die Präfixaggregation für Nutzungsmetriken aktiviert, werden Metriken auf Präfixebene nicht veröffentlicht. CloudWatch Nur S3 Storage Lens-Metriken auf Bucket-, Konto- und Organisationsebene werden veröffentlicht. CloudWatch
6. Wählen Sie Änderungen speichern.

Um ein neues S3 Storage Lens-Dashboard zu erstellen, das Support ermöglicht CloudWatch

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Klicken Sie auf Dashboard erstellen.
4. Definieren Sie unter General (Allgemein) die folgenden Konfigurationsoptionen:

   a. Geben Sie im Feld Dashboard name (Dashboard-Name) Ihren Dashboard-Namen ein.

Dashboard-Namen müssen weniger als 65 Zeichen lang sein und dürfen keine Sonderzeichen oder Leerzeichen enthalten. Sie können den Dashboard-Namen nicht mehr ändern, nachdem Sie Ihr Dashboard erstellt haben.

b. Wählen Sie die Heimatregion für Ihr Dashboard aus.

Metriken für alle Regionen, die in diesem Dashboard-Umfang enthalten sind, werden zentral in der angegebenen Heimatregion gespeichert. In CloudWatch, S3 Storage Lens-Metriken sind auch in der Heimatregion verfügbar. Sie können die Heimatregion nicht ändern, nachdem Sie Ihr Dashboard erstellt haben.

5. (Optional) Sie fügen ein Tag hinzu, indem Sie Add tag (Tag hinzufügen) auswählen und den Schlüssel und den Wert für den Tag eingeben.

> **Note**
>
> Sie können Ihrer Dashboard-Konfiguration bis zu 50 Markierungen hinzufügen.

6. Definieren Sie den Umfang für Ihre Konfiguration:

a. Wenn Sie eine Konfiguration auf Organisationsebene erstellen, wählen Sie die Konten aus, die in die Konfiguration aufgenommen werden sollen: Include all accounts in your configuration (Alle Konten in die Konfiguration einschließen) oder Limit the scope to your signed-in account (Umfang auf das angemeldete Konto beschränken).

> **Note**
>
> Wenn Sie eine Konfiguration auf Organisationsebene erstellen, die alle Konten enthält, können Sie nur Regionen und keine Buckets ein- oder ausschließen.

b. Wählen Sie die Regionen und Buckets aus, die S3 Storage Lens im Dashboard einschließen soll, indem Sie die folgenden Schritte ausführen:

- Um alle Regionen einzuschließen, wählen Sie Regionen und Buckets einschließen aus.
- Um bestimmte Regionen einzubeziehen, löschen Sie Alle Regionen einschließen. Unter Wählen Sie die einzuschließenden Regionen wählen Sie die Regionen aus, die S3 Storage Lens im Dashboard enthalten soll.

- Um bestimmte Buckets einzubeziehen, löschen Sie Alle Buckets einschließen. Wählen Sie unter Einzuschließende Buckets auswählen die Buckets aus, die S3 Storage Lens in das Dashboard aufnehmen soll.

> **Note**
>
> Sie können bis zu 50 Buckets auswählen.

7. Unter Metrics selection (Metrikauswahl) wählen Sie Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) aus.

   Weitere Informationen zu fortschrittlichen Metriken und Preisempfehlungen finden Sie unter Amazon-S3-Preise.

8. Wählen Sie unter Advanced metrics and recommendations features (Erweiterte Metrik- und Empfehlungsfunktionen) die Optionen aus, Sie aktivieren möchten:

   - Advanced metrics (Erweiterte Metriken)
   - CloudWatch publizieren

   > **Important**
   >
   > Wenn Sie die Präfixaggregation für Ihre S3 Storage Lens-Konfiguration aktivieren, werden Metriken auf Präfixebene nicht veröffentlicht. CloudWatch Nur S3 Storage Lens-Metriken auf Bucket-, Konto- und Organisationsebene werden veröffentlicht. CloudWatch

   - Präfixzusammenfassung

   > **Note**
   >
   > Weitere Informationen über erweiterte Metriken und Empfehlungen finden Sie unter Metrikauswahl.

9. Wenn Sie Advanced Metrics (Erweiterte Metriken) aktiviert haben, wählen Sie die Advanced metrics categories (Erweiterte Metrikkategorien) aus, die Sie in Ihrem S3-Storage-Lens-Dashboard anzeigen möchten:

   - Metriken für Aktivitäten

- Detailed status code metrics (Detaillierte Statuscode-Metriken)
- Advanced cost optimization metrics (Erweiterte Kostenoptimierungsmetriken)
- Advanced data protection metrics (Erweiterte Datensicherheitsmetriken)

Weitere Informationen zu den Metrikkategorien finden Sie unter Metrikkategorien. Eine vollständige Liste der Metriken finden Sie unter Amazon S3-Storage-Lens-Metrikglossar.

10. (Optional) Konfigurieren Sie den Metrikexport.

    Weitere Informationen zum Konfigurieren eines Metrikexports finden Sie unter Schritt Verwenden der S3-Konsole.

11. Klicken Sie auf Dashboard erstellen.

Unter Verwendung der AWS CLI

Im folgenden AWS CLI Beispiel wird die CloudWatch Veröffentlichungsoption mithilfe einer Konfiguration für erweiterte Metriken und Empfehlungen von S3 Storage Lens auf Organisationsebene aktiviert. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control put-storage-lens-configuration --account-id=555555555555 --config-
id=your-configuration-id --region=us-east-1 --storage-lens-configuration=file://./
config.json

config.json
{
  "Id": "SampleS3StorageLensConfiguration", //Use this property to identify your S3
 Storage Lens configuration.
  "AwsOrg": { //Use this property when enabling S3 Storage Lens for AWS Organizations.
    "Arn": "arn:aws:organizations::123456789012:organization/o-abcdefgh"
  },
  "AccountLevel": {
    "ActivityMetrics": {
      "IsEnabled":true
    },
    "AdvancedCostOptimizationMetrics": {
      "IsEnabled":true
    },
    "AdvancedDataProtectionMetrics": {
      "IsEnabled":true
```

```
    },
    "DetailedStatusCodesMetrics": {
      "IsEnabled":true
    },
    "BucketLevel": {
      "ActivityMetrics": {
        "IsEnabled":true //Mark this as false if you want only free metrics.
      },
      "ActivityMetrics": {
        "IsEnabled":true //Mark this as false if you want only free metrics.
      },
      "AdvancedCostOptimizationMetrics": {
        "IsEnabled":true //Mark this as false if you want only free metrics.
      },
      "DetailedStatusCodesMetrics": {
        "IsEnabled":true //Mark this as false if you want only free metrics.
      },
      "PrefixLevel":{
        "StorageMetrics":{
          "IsEnabled":true, //Mark this as false if you want only free metrics.
          "SelectionCriteria":{
            "MaxDepth":5,
            "MinStorageBytesPercentage":1.25,
            "Delimiter":"/"
          }
        }
      }
    }
  },
  "Exclude": { //Replace with "Include" if you prefer to include Regions.
    "Regions": [
      "eu-west-1"
    ],
    "Buckets": [ //This attribute is not supported for AWS Organizations-level
configurations.
      "arn:aws:s3:::amzn-s3-demo-source-bucket "
    ]
  },
  "IsEnabled": true, //Whether the configuration is enabled
  "DataExport": { //Details about the metrics export
    "S3BucketDestination": {
      "OutputSchemaVersion": "V_1",
      "Format": "CSV", //You can add "Parquet" if you prefer.
      "AccountId": "111122223333",
```

```
      "Arn": "arn:aws:s3:::amzn-s3-demo-destination-bucket", // The destination
 bucket for your metrics export must be in the same Region as your S3 Storage Lens
 configuration.
      "Prefix": "prefix-for-your-export-destination",
      "Encryption": {
        "SSES3": {}
      }
    },
    "CloudWatchMetrics": {
      "IsEnabled": true //Mark this as false if you want to export only free metrics.
    }
  }
}
```

Verwendung des AWS SDK for Java

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.ActivityMetrics;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.CloudWatchMetrics;
import com.amazonaws.services.s3control.model.Format;
import com.amazonaws.services.s3control.model.Include;
import com.amazonaws.services.s3control.model.OutputSchemaVersion;
import com.amazonaws.services.s3control.model.PrefixLevel;
import com.amazonaws.services.s3control.model.PrefixLevelStorageMetrics;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.S3BucketDestination;
import com.amazonaws.services.s3control.model.SSES3;
import com.amazonaws.services.s3control.model.SelectionCriteria;
import com.amazonaws.services.s3control.model.StorageLensAwsOrg;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensDataExport;
import com.amazonaws.services.s3control.model.StorageLensDataExportEncryption;
import com.amazonaws.services.s3control.model.StorageLensTag;

import java.util.Arrays;
```

```
import java.util.List;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateAndUpdateDashboard {

    public static void main(String[] args) {
        String configurationId = "ConfigurationId";
        String sourceAccountId = "Source Account ID";
        String exportAccountId = "Destination Account ID";
        String exportBucketArn = "arn:aws:s3:::amzn-s3-demo-destination-bucket"; //
 The destination bucket for your metrics export must be in the same Region as your S3
 Storage Lens configuration.
        String awsOrgARN = "arn:aws:organizations::123456789012:organization/o-
abcdefgh";
        Format exportFormat = Format.CSV;

        try {
            SelectionCriteria selectionCriteria = new SelectionCriteria()
                    .withDelimiter("/")
                    .withMaxDepth(5)
                    .withMinStorageBytesPercentage(10.0);
            PrefixLevelStorageMetrics prefixStorageMetrics = new
 PrefixLevelStorageMetrics()
                    .withIsEnabled(true)
                    .withSelectionCriteria(selectionCriteria);
            BucketLevel bucketLevel = new BucketLevel()
                    .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                    .withAdvancedCostOptimizationMetrics(new
 AdvancedCostOptimizationMetrics().withIsEnabled(true))
                    .withAdvancedDataProtectionMetrics(new
 AdvancedDataProtectionMetrics().withIsEnabled(true))
                    .withDetailedStatusCodesMetrics(new
 DetailedStatusCodesMetrics().withIsEnabled(true))
                    .withPrefixLevel(new
 PrefixLevel().withStorageMetrics(prefixStorageMetrics));
            AccountLevel accountLevel = new AccountLevel()
                    .withActivityMetrics(new ActivityMetrics().withIsEnabled(true))
                    .withAdvancedCostOptimizationMetrics(new
 AdvancedCostOptimizationMetrics().withIsEnabled(true))
                    .withAdvancedDataProtectionMetrics(new
 AdvancedDataProtectionMetrics().withIsEnabled(true))
                    .withDetailedStatusCodesMetrics(new
 DetailedStatusCodesMetrics().withIsEnabled(true))
```

```
                .withBucketLevel(bucketLevel);

        Include include = new Include()
                .withBuckets(Arrays.asList("arn:aws:s3:::amzn-s3-demo-bucket"))
                .withRegions(Arrays.asList("us-west-2"));

        StorageLensDataExportEncryption exportEncryption = new
StorageLensDataExportEncryption()
                .withSSES3(new SSES3());
        S3BucketDestination s3BucketDestination = new S3BucketDestination()
                .withAccountId(exportAccountId)
                .withArn(exportBucketArn)
                .withEncryption(exportEncryption)
                .withFormat(exportFormat)
                .withOutputSchemaVersion(OutputSchemaVersion.V_1)
                .withPrefix("Prefix");
        CloudWatchMetrics cloudWatchMetrics = new CloudWatchMetrics()
                .withIsEnabled(true);
        StorageLensDataExport dataExport = new StorageLensDataExport()
                .withCloudWatchMetrics(cloudWatchMetrics)
                .withS3BucketDestination(s3BucketDestination);

        StorageLensAwsOrg awsOrg = new StorageLensAwsOrg()
                .withArn(awsOrgARN);

        StorageLensConfiguration configuration = new StorageLensConfiguration()
                .withId(configurationId)
                .withAccountLevel(accountLevel)
                .withInclude(include)
                .withDataExport(dataExport)
                .withAwsOrg(awsOrg)
                .withIsEnabled(true);

        List<StorageLensTag> tags = Arrays.asList(
                new StorageLensTag().withKey("key-1").withValue("value-1"),
                new StorageLensTag().withKey("key-2").withValue("value-2")
        );

        AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(US_WEST_2)
                .build();
```

```
            s3ControlClient.putStorageLensConfiguration(new
 PutStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withStorageLensConfiguration(configuration)
                    .withTags(tags)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Unter Verwendung der REST API

Um die CloudWatch Veröffentlichungsoption mithilfe von Amazon S3 zu aktivieren RESTAPI, können Sie PutStorageLensConfiguration.

Nächste Schritte

Nachdem Sie die CloudWatch Veröffentlichungsoption aktiviert haben, können Sie unter auf Ihre S3 Storage Lens-Metriken zugreifen CloudWatch. Sie können auch CloudWatch Funktionen zur Überwachung und Analyse Ihrer S3 Storage Lens-Daten in nutzen CloudWatch. Weitere Informationen finden Sie unter den folgenden Themen:

- Metriken und Dimensionen von S3 Storage Lens
- Arbeiten mit S3 Storage Lens-Metriken in CloudWatch

Arbeiten mit S3 Storage Lens-Metriken in CloudWatch

Sie können S3 Storage Lens-Metriken auf Amazon veröffentlichen CloudWatch , um eine einheitliche Ansicht Ihres Betriebszustands in CloudWatchDashboards zu erstellen. Sie können auch CloudWatch Funktionen wie Alarme und ausgelöste Aktionen, metrische Berechnungen und Anomalieerkennung verwenden, um S3 Storage Lens-Metriken zu überwachen und entsprechende Maßnahmen zu ergreifen. Darüber hinaus ermöglichen CloudWatch API Betriebsabläufe Anwendungen,

einschließlich Drittanbietern, den Zugriff auf Ihre S3 Storage Lens-Metriken. Weitere Informationen zu den CloudWatch Funktionen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

Sie können die CloudWatch Veröffentlichungsoption für neue oder bestehende Dashboard-Konfigurationen aktivieren, indem Sie die Amazon S3-Konsole, Amazon S3 REST APIs AWS CLI, und verwenden AWS SDKs. Die CloudWatch Veröffentlichungsoption ist für Dashboards verfügbar, die auf erweiterte Metriken und Empfehlungen von S3 Storage Lens aktualisiert wurden. Informationen zu Preisen für erweiterte Metriken und Empfehlungen von S3 Storage Lens finden Sie unter Amazon-S3-Preise. Es fallen keine zusätzlichen Gebühren für die Veröffentlichung von CloudWatch Kennzahlen an. Es fallen jedoch andere CloudWatch Gebühren an, z. B. für Dashboards, Alarme und API Anrufe. Weitere Informationen finden Sie unter CloudWatch Amazon-Preise.

S3 Storage Lens-Metriken werden CloudWatch in dem Konto veröffentlicht, dem die S3 Storage Lens-Konfiguration gehört. Nachdem Sie die CloudWatch Veröffentlichungsoption in den erweiterten Metriken und Empfehlungen aktiviert haben, können Sie unter auf Kennzahlen auf Organisations-, Konto- und Bucket-Ebene zugreifen. CloudWatch Metriken auf Präfix-Ebene sind in nicht verfügbar. CloudWatch

> **Note**
>
> S3 Storage Lens-Metriken sind Tagesmetriken und werden CloudWatch nur einmal täglich veröffentlicht. Wenn Sie S3 Storage Lens-Metriken in abfragen CloudWatch, muss der Zeitraum für die Abfrage 1 Tag (86400 Sekunden) betragen. Nachdem Ihre täglichen S3 Storage Lens-Metriken in Ihrem S3 Storage Lens-Dashboard in der Amazon S3 S3-Konsole angezeigt werden, kann es einige Stunden dauern, bis dieselben Messwerte angezeigt werden CloudWatch. Wenn Sie die CloudWatch Veröffentlichungsoption für S3 Storage Lens-Metriken zum ersten Mal aktivieren, kann es bis zu 24 Stunden dauern, bis Ihre Metriken veröffentlicht CloudWatch werden.
> Derzeit können S3 Storage Lens-Metriken nicht über CloudWatch Streams abgerufen werden.

Weitere Informationen zur Arbeit mit S3 Storage Lens-Metriken finden Sie in CloudWatch den folgenden Themen.

Themen

- Arbeiten mit CloudWatch Dashboards

- Alarme einstellen, Aktionen auslösen und Anomalieerkennung verwenden
- Metriken mit Dimensionen filtern
- Berechnen neuer Metriken mit Metrikmathematik
- Verwenden von Suchausdrücken in Diagrammen

Arbeiten mit CloudWatch Dashboards

Sie können CloudWatch Dashboards verwenden, um S3 Storage Lens-Metriken zusammen mit anderen Anwendungsmetriken zu überwachen und einen einheitlichen Überblick über Ihren Betriebsstatus zu erhalten. Dashboards sind anpassbare Homepages in der CloudWatch Konsole, mit denen Sie Ihre Ressourcen in einer einzigen Ansicht überwachen können.

CloudWatch verfügt über eine umfassende Berechtigungssteuerung, die die Beschränkung des Zugriffs auf einen bestimmten Satz von Metriken oder Dimensionen nicht unterstützt. Benutzer in Ihrem Konto oder Ihrer Organisation, die Zugriff auf haben, haben Zugriff auf Metriken für alle S3 Storage Lens-Konfigurationen, für die die CloudWatch Support-Option aktiviert ist. CloudWatch Sie können Berechtigungen für bestimmte Dashboards nicht wie in S3 Storage Lens verwalten. Weitere Informationen zu CloudWatch Berechtigungen finden Sie unter Verwaltung von Zugriffsberechtigungen für Ihre CloudWatch Ressourcen im CloudWatch Amazon-Benutzerhandbuch.

Weitere Informationen zur Verwendung von CloudWatch Dashboards und zur Konfiguration von Berechtigungen finden Sie unter CloudWatch Amazon-Dashboards verwenden und CloudWatch Dashboards teilen im CloudWatch Amazon-Benutzerhandbuch.

Alarme einstellen, Aktionen auslösen und Anomalieerkennung verwenden

Sie können CloudWatch Alarme konfigurieren, die die Messwerte von S3 Storage Lens überwachen CloudWatch und Maßnahmen ergreifen, wenn ein Schwellenwert überschritten wird. Sie können beispielsweise einen Alarm konfigurieren, der eine SNS Amazon-Benachrichtigung sendet, wenn die Metrik Incomplete Multipart Upload Bytes an drei aufeinanderfolgenden Tagen 1 GB überschreitet.

Sie können auch die Anomalieerkennung aktivieren, um Ihre S3-Storage-Lens-Metriken kontinuierlich zu analysieren und normale Baselines und Oberflächenanomalien zu ermitteln. Sie können Anomalieerkennungsalarme basierend auf dem erwarteten Wert einer Metrik erstellen. Sie können beispielsweise Anomalien für die Metrik Object Lock Enabled Bytes (Bytes mit aktivierter Objektsperre), um das unbefugte Entfernen von Objektsperreinstellungen zu erkennen.

Weitere Informationen und Beispiele finden Sie unter Verwenden von CloudWatch Amazon-Alarmen und Erstellen eines Alarms anhand einer Metrik in einem Diagramm im CloudWatch Amazon-Benutzerhandbuch.

Metriken mit Dimensionen filtern

Sie können Dimensionen verwenden, um S3 Storage Lens-Metriken in der CloudWatch Konsole zu filtern. Beispielsweise können Sie nach `configuration_id`, `aws_account_number`, `aws_region`, `bucket_name` und mehr filtern.

S3 Storage Lens unterstützt mehrere Dashboard-Konfigurationen pro Konto. Dies bedeutet, dass verschiedene Konfigurationen denselben Bucket enthalten können. Wenn diese Metriken veröffentlicht werden CloudWatch, enthält der Bucket doppelte Metriken CloudWatch. Um nur Metriken für eine bestimmte S3 Storage Lens-Konfiguration in anzuzeigen CloudWatch, können Sie die `configuration_id` Dimension verwenden. Wenn Sie nach `configuration_id` filtern, sehen Sie nur Metriken, die mit der Konfiguration verknüpft sind, die Sie identifizieren.

Weitere Informationen zum Filtern nach Konfigurations-ID finden Sie unter Suchen nach verfügbaren Metriken im CloudWatch Amazon-Benutzerhandbuch.

Berechnen neuer Metriken mit Metrikmathematik

Mithilfe von Metrikberechnungen können Sie mehrere Metriken mit S3 Storage Lens abfragen und mathematische Ausdrücke verwenden, um neue Zeitreihen basierend auf diesen Metriken zu erstellen. Sie können beispielsweise eine neue Metrik für unverschlüsselte Objekte erstellen, indem Sie verschlüsselte Objekte von der Objektanzahl subtrahieren. Sie können auch eine Metrik erstellen, um die durchschnittliche Objektgröße zu erhalten, indem Sie `StorageBytes` durch `ObjectCount` oder die Anzahl der Bytes dividieren, auf die an einem Tag zugegriffen wird, indem Sie `BytesDownloaded` durch `StorageBytes` dividieren.

Weitere Informationen finden Sie unter Verwenden von metrischer Mathematik im CloudWatch Amazon-Benutzerhandbuch.

Verwenden von Suchausdrücken in Diagrammen

Mit S3-Storage-Lens-Metriken können Sie einen Suchausdruck erstellen. Sie können beispielsweise einen Suchausdruck für alle benannten Metriken erstellen IncompleteMultipartUploadStorageBytesund `SUM` dem Ausdruck hinzufügen. Mit diesem Suchausdruck können Sie Ihre gesamten unvollständigen mehrteiligen Upload-Bytes über alle Dimensionen Ihres Speichers in einer einzigen Metrik sehen.

Dieses Beispiel zeigt die Syntax, die Sie verwenden würden, um einen Suchausdruck für alle genannten Metriken zu erstellen IncompleteMultipartUploadStorageBytes.

```
SUM(SEARCH('{AWS/S3/Storage-
Lens,aws_account_number,aws_region,configuration_id,metrics_version,record_type,storage_class}
 MetricName="IncompleteMultipartUploadStorageBytes"', 'Average',86400))
```

Weitere Informationen zu dieser Syntax finden Sie unter Syntax für CloudWatch Suchausdrücke im CloudWatch Amazon-Benutzerhandbuch. Informationen zum Erstellen eines CloudWatch Diagramms mit einem Suchausdruck finden Sie unter Erstellen eines CloudWatch Diagramms mit einem Suchausdruck im CloudWatch Amazon-Benutzerhandbuch.

Anwendungsfälle für Metriken von Amazon S3 Storage Lens

Sie können Ihr Dashboard von Amazon S3 Storage Lens verwenden, um Erkenntnisse und Trends zu visualisieren, Ausreißer zu markieren und Empfehlungen einzusehen. Die Metriken von S3 Storage Lens sind in Kategorien unterteilt, die den wichtigsten Anwendungsfällen entsprechen. Diese Metriken bieten Ihnen folgende Möglichkeiten:

- Identifizieren von Möglichkeiten zur Kostenoptimierung
- Anwenden von bewährten Datenschutzmethoden
- Anwenden von bewährten Zugriffsverwaltungsmethoden
- Verbessern der Leistung von Anwendungs-Workloads

Mithilfe der Metriken zur Kostenoptimierung können Sie Möglichkeiten ermitteln, wie Sie Ihre Amazon-S3-Speicherkosten senken können. Sie können Buckets mit mehrteiligen Uploads identifizieren, die älter als 7 Tage sind, oder Buckets, in denen sich nicht aktuelle Versionen ansammeln.

In ähnlicher Weise können Sie Datenschutzmetriken verwenden, um Buckets zu identifizieren, die nicht den bewährten Datenschutzmethoden Ihres Unternehmens entsprechen. Sie können beispielsweise Buckets identifizieren, die keine AWS Key Management Service Schlüssel (SSE-KMS) für die Standardverschlüsselung verwenden oder für die die S3-Versionierung nicht aktiviert ist.

Mit den Zugriffsmanagement-Metriken von S3 Storage Lens können Sie Bucket-Einstellungen für den Besitz von S3-Objekten identifizieren, sodass Sie Zugriffssteuerungslisten (ACL) in Bucket-Richtlinien migrieren und deaktivieren können. ACLs

Wenn Sie die Option S3 Storage Lens advanced metrics (Erweiterte Metriken von S3 Storage Lens) aktiviert haben, können Sie die detaillierten Statuscode-Metriken verwenden, um die Anzahl von erfolgreichen oder fehlgeschlagenen Anforderungen zu erhalten. Anhand dieser Informationen können Sie Zugriffs- oder Leistungsprobleme beheben.

Mithilfe erweiterter Metriken können Sie auch auf zusätzliche Metriken zur Kostenoptimierung und zum Datenschutz zugreifen, anhand derer Sie Möglichkeiten zur weiteren Senkung Ihrer Gesamtkosten für den S3-Speicher identifizieren und besser an bewährte Methoden für den Schutz Ihrer Daten anpassen können. Erweiterte Metriken zur Kostenoptimierung umfassen beispielsweise die Anzahl von Lebenszyklusregeln, mit der Sie Buckets identifizieren können, für die keine Lebenszyklusregeln festgelegt sind, um unvollständige mehrteilige Uploads, die älter als 7 Tage sind, ablaufen zu lassen. Zu den erweiterten Datenschutzmetriken gehört die Anzahl der Replikationsregeln.

Weitere Informationen zu den Metrikkategorien finden Sie unter Metrikkategorien. Eine vollständige Liste der Metriken von S3 Storage Lens finden Sie unter Amazon S3-Storage-Lens-Metrikglossar.

Themen

- Verwenden von Amazon S3 Storage Lens zur Optimierung Ihrer Speicherkosten
- Verwenden von S3 Storage Lens zum Schutz Ihrer Daten
- Verwenden von S3 Storage Lens zur Überwachung der Einstellungen für die Objekteigentümerschaft
- Verwenden der Metriken von S3 Storage Lens zur Verbesserung der Leistung

Verwenden von Amazon S3 Storage Lens zur Optimierung Ihrer Speicherkosten

Sie können die Kostenoptimierungsmetriken von S3 Storage Lens verwenden, um die Gesamtkosten Ihres S3-Speichers zu senken. Mithilfe von Metriken zur Kostenoptimierung können Sie überprüfen, dass Sie Amazon S3 kostengünstig und gemäß den bewährten Methoden konfiguriert haben. Sie können beispielsweise die folgenden Möglichkeiten zur Kostenoptimierung identifizieren:

- Buckets mit unvollständigen mehrteiligen Uploads, die älter als 7 Tage sind
- Buckets, in denen sich zahlreiche nicht aktuelle Versionen ansammeln
- Buckets, die keine Lebenszyklusregeln zum Abbruch unvollständiger mehrteiliger Uploads haben
- Buckets, die keine Lebenszyklusregeln für das Ablaufen von Objekten mit nicht aktuellen Versionen haben

- Buckets, die keine Lebenszyklusregeln für den Übergang von Objekten in eine andere Speicherklasse haben

Sie können diese Daten dann verwenden, um Ihren Buckets zusätzliche Lebenszyklusregeln hinzuzufügen.

In den folgenden Beispielen wird gezeigt, wie Sie Metriken zur Kostenoptimierung in Ihrem S3-Storage-Lens-Dashboard verwenden, um Ihre Speicherkosten zu optimieren.

Themen

- Identifizieren Sie Ihre größten S3-Buckets
- Entdecken Sie kalte Amazon S3-Buckets
- Suchen von unvollständigen mehrteiligen Uploads
- Reduzieren Sie die Anzahl der beibehaltenen nicht aktuellen Versionen
- Identifizieren von Buckets ohne Lebenszyklusregeln und Überprüfen der Anzahl von Lebenszyklusregeln

Identifizieren Sie Ihre größten S3-Buckets

Sie zahlen für das Speichern von Objekten in S3-Buckets. Der Ihnen in Rechnung gestellte Preis hängt von der Größe Ihrer Objekte, der Speicherdauer der Objekte und ihren Speicherklassen ab. Mit S3 Storage Lens erhalten Sie einen zentralen Überblick über alle Buckets in Ihrem Konto. Um alle Buckets in allen Konten Ihres Unternehmens zu sehen, können Sie ein S3 Storage AWS Organizations Lens-Dashboard auf der gleichen Ebene konfigurieren. In dieser Dashboard-Ansicht können Sie Ihre größten Buckets identifizieren.

Schritt 1: Identifizieren Ihrer größten Buckets

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie anzeigen möchten.

   Wenn das Dashboard geöffnet wird, können Sie das Datum sehen, für das S3 Storage Lens zuletzt Metriken gesammelt hat. Ihr Dashboard wird immer für das letzte Datum geladen, für das Metriken verfügbar sind.

4. Scrollen Sie zum Abschnitt Top N overview for date (Top-N-Übersicht für Datum), um eine Rangfolge Ihrer größten Buckets nach der Metrik Total storage (Gesamtspeicher) für einen ausgewählten Zeitraum anzuzeigen.

   Sie können die Sortierreihenfolge ändern, um die kleinsten Buckets anzuzeigen. Sie können auch die Auswahl Metric (Metrik) anpassen, um Ihre Buckets nach einer der verfügbaren Metriken zu ordnen. Der Abschnitt Top N overview for date (Top-N-Übersicht für Datum) zeigt auch die prozentuale Veränderung gegenüber dem Vortag oder der Vorwoche sowie eine Sparkline zur Veranschaulichung des Trends an. Dieser Trend ist ein 14-Tage-Trend für kostenlose Metriken und ein 30-Tage-Trend für erweiterte Metriken und Empfehlungen.

   > **Note**
   >
   > Mit den erweiterten Metriken und Empfehlungen von S3 Storage Lens stehen Metriken 15 Monate für Abfragen zur Verfügung. Weitere Informationen finden Sie unter Metrikauswahl.

5. Für detailliertere Informationen zu Ihren Buckets scrollen Sie zum Seitenanfang und wählen Sie dann die Registerkarte Bucket.

   Auf der Registerkarte Buckets können Sie Details wie die aktuelle Wachstumsrate, die durchschnittliche Objektgröße, die größten Präfixe und die Anzahl der Objekte anzeigen.

Schritt 2: Navigieren zu Ihren Buckets und Untersuchen von Daten

Nachdem Sie Ihre größten S3-Buckets identifiziert haben, können Sie innerhalb der S3-Konsole zu jedem Bucket navigieren, um seine Objekte anzuzeigen, die damit verbundene Workload zu verstehen und interne Eigentümer des Buckets zu identifizieren. Sie können die Bucket-Eigentümer kontaktieren, um herauszufinden, ob das Wachstum erwartet wird oder ob es einer weiteren Überwachung und Kontrolle bedarf.

Entdecken Sie kalte Amazon S3-Buckets

Wenn Sie die fortschrittlichen Metriken von S3 Storage Lens aktiviert haben, können Sie Aktivitätsmetriken verwenden, um zu verstehen, wie kalt Ihre S3-Buckets sind. Ein „kalter“ Bucket ist ein Bucket, auf dessen Speicher nicht mehr (oder sehr selten) zugegriffen wird. Dieser Mangel an Aktivität weist normalerweise darauf hin, dass auf die Objekte des Buckets nicht häufig zugegriffen wird.

Aktivitätskennzahlen wie GETAnfragen und Download-Bytes geben an, wie oft täglich auf Ihre Buckets zugegriffen wird. Um die Konsistenz des Zugriffsmusters zu verstehen und Buckets zu erkennen, auf die überhaupt nicht mehr zugegriffen wird, können Sie diese Daten über mehrere Monate hinweg trenden. Die Metrik der Abrufrate, die als Download-Bytes/Gesamtspeicher berechnet wird, gibt den Anteil des Speichers in einem Bucket an, auf den täglich zugegriffen wird.

> **Note**
>
> Download-Bytes werden dupliziert, wenn das gleiche Objekt mehrmals am Tag heruntergeladen wird.

Voraussetzung

Wenn Sie Aktivitätsmetriken in Ihrem S3-Storage-Lens-Dashboard anzeigen möchten, müssen Sie die Option Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) von S3 Storage Lens aktivieren und dann Activity metrics (Aktivitätsmetriken) auswählen. Weitere Informationen finden Sie unter Verwenden der S3-Konsole.

Schritt 1: Identifizieren aktiver Buckets

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie anzeigen möchten.
4. Wählen Sie die Registerkarte Bucket aus und scrollen Sie dann nach unten zum Abschnitt Bubble analysis by buckets for date (Blasenanalyse nach Buckets für Datum).

   Im Abschnitt Bubble analysis by buckets for date (Blasenanalyse nach Buckets für Datum) können Sie Ihre Buckets in mehreren Dimensionen darstellen, indem Sie drei beliebige Metriken verwenden, um die X-axis (x-Achse), Y-axis (y-Achse) und die Size (Größe) der Blase darzustellen.
5. Wählen Sie für X-axis (x-Achse) Y-axis (y-Achse) und Size (Größe) die Metriken Total storage (Gesamtspeicher), % retrieval rate (% Abrufrate) und Average object size (Durchschnittliche Objektgröße) aus, um „kalte“ Buckets zu finden.
6. Suchen Sie im Abschnitt Bubble analysis by buckets for date (Blasenanalyse nach Buckets für Datum) nach Buckets mit Abrufraten von null (oder nahe null) und einer größeren relativen Speichergröße. Wählen Sie dann die Blase aus, die den Bucket darstellt.

Es erscheint ein Feld mit Auswahlmöglichkeiten für detailliertere Einblicke. Führen Sie eine der folgenden Aktionen aus:

a. Damit die Registerkarte Bucket aktualisiert wird und nur Metriken für den ausgewählten Bucket anzeigt, wählen Sie Drill down (Aufgliedern) und dann Apply (Anwenden) aus.

b. Um Ihre Daten auf Bucket-Ebene nach Konto, AWS-Region Speicherklasse oder Bucket zu aggregieren, wählen Sie Analysieren nach und treffen Sie dann eine Auswahl für Dimension. Wenn Sie beispielsweise nach Speicherklasse aggregieren möchten, wählen Sie Storage class (Speicherklasse) für Dimension aus.

Um Buckets zu finden, die kalt geworden sind, führen Sie eine Blasenanalyse mit den Metriken Gesamtspeicher, Abrufrate in % und Durchschnittliche Objektgröße durch. Suchen Sie nach Buckets mit Abrufraten von null (oder nahe null) und einer größeren relativen Speichergröße.

Die Registerkarte Bucket Ihres Dashboards wird aktualisiert und zeigt Daten für die von Ihnen ausgewählte Aggregation oder den ausgewählten Filter an. Wenn Sie nach Speicherklasse oder einer anderen Dimension aggregiert haben, wird diese neue Registerkarte in Ihrem Dashboard geöffnet (z. B. die Registerkarte Storage class (Speicherklasse)).

Schritt 2: Untersuchen kalter Buckets

Von hier aus können Sie die Eigentümer von kalten Buckets in Ihrem Konto oder Ihrer Organisation identifizieren und herausfinden, ob dieser Speicher noch benötigt wird. Anschließend können Sie die Kosten optimieren, indem Sie Ablaufkonfigurationen für den Lebenszyklus für diese Buckets konfigurieren oder die Daten in einer der Amazon S3 Glacier -Speicherklassen archivieren.

Um das Problem von kalten Buckets in Zukunft zu vermeiden, können Sie Ihre Daten mithilfe von S3-Lifecycle-Konfigurationen für Ihre Buckets automatisch umstellen oder die automatische Archivierung mit S3 Intelligent-Tiering aktivieren.

Sie können Schritt 1 auch verwenden, um „heiße“ Buckets zu identifizieren. Anschließend können Sie sich vergewissern, dass diese Buckets die richtige S3-Speicherklasse verwenden, um sicherzustellen, dass sie ihre Anforderungen in Bezug auf Leistung und Kosten am effektivsten bearbeiten.

Suchen von unvollständigen mehrteiligen Uploads

Sie können mehrteilige Uploads verwenden, um sehr große Objekte (bis zu 5 TB) als einen Satz von Teilen hochzuladen und so den Durchsatz zu verbessern und Netzwerkprobleme schneller zu beheben. In Fällen, in denen der mehrteilige Upload-Prozess nicht abgeschlossen wird, verbleiben die unvollständigen Teile im Bucket (in einem unbrauchbaren Zustand). Für diese unvollständigen Teile fallen Speicherkosten an, bis der Upload-Prozess abgeschlossen ist oder die unvollständigen Teile entfernt werden. Weitere Informationen finden Sie unter Hochladen und Kopieren von Objekten mit mehrteiligen Uploads.

Mit S3 Storage Lens können Sie die Anzahl der unvollständigen mehrteiligen Upload-Bytes in Ihrem Konto oder in Ihrem gesamten Unternehmen ermitteln. Dies umfasst auch mehrteilige Uploads, die älter sind als 7 Tage. Eine vollständige Liste der Metriken für unvollständige mehrteilige Uploads finden Sie unter Amazon S3-Storage-Lens-Metrikglossar.

Als bewährte Methode empfehlen wir, Lebenszyklusregeln so zu konfigurieren, dass unvollständige mehrteilige Uploads, die älter als eine bestimmte Anzahl von Tagen sind, ablaufen. Wenn Sie Ihre Lebenszyklusregel so erstellen, dass unvollständige mehrteilige Uploads ablaufen, empfehlen wir 7 Tage als geeigneten Ausgangspunkt.

Schritt 1: Überprüfen allgemeiner Trends für unvollständige mehrteilige Uploads

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie anzeigen möchten.
4. Wählen Sie im Abschnitt Snapshot for date (Snapshot für Datum) unter Metrics categories (Metrikkategorien) die Option Cost optimization (Kostenoptimierung) aus.

   Der Abschnitt Snapshot for date (Snapshot für Datum) wird aktualisiert und zeigt nun Metriken zu Cost optimization (Kostenoptimierung) an. Dazu gehören Incomplete multipart upload bytes greater than 7 days old (Unvollständige mehrteilige Upload-Bytes, die älter als 7 Tage sind).

   In jedem Diagramm Ihres S3-Storage-Lens-Dashboards können Sie Metriken für unvollständige mehrteilige Uploads sehen. Sie können diese Metriken verwenden, um die Auswirkungen unvollständiger mehrteiliger Upload-Bytes auf Ihren Speicher weiter zu bewerten, einschließlich deren Beitrag zu den allgemeinen Wachstumstrends. Sie können auch eine detaillierte Aufschlüsselung tieferer Aggregationsebenen aufrufen. Verwenden Sie dazu die Registerkarten

Account (Konto), AWS-Region, Bucket oder Storage class (Speicherklasse), um Ihre Daten eingehender zu analysieren. Ein Beispiel finden Sie unter Entdecken Sie kalte Amazon S3-Buckets.

Schritt 2: Identifizieren von Buckets, die die unvollständigsten mehrteiligen Upload-Bytes, aber keine Lebenszyklusregeln zum Abbruch von unvollständigen mehrteiligen Uploads haben

Voraussetzung

Wenn Sie in Ihrem S3-Storage-Lens-Dashboard die Metrik Abort incomplete multipart upload lifecycle rule count (Anzahl der Lebenszyklusregeln zum Abbrechen unvollständiger mehrteiliger Uploads) sehen möchten, müssen Sie Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) von S3 Storage Lens aktivieren und dann Advanced cost optimization metrics (Erweiterte Kostenoptimierungsmetriken) auswählen. Weitere Informationen finden Sie unter Verwenden der S3-Konsole.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie anzeigen möchten.
4. Wenn Sie bestimmte Buckets identifizieren möchten, in denen sich unvollständige mehrteilige Uploads ansammeln, die älter als 7 Tage sind, gehen Sie zum Abschnitt Top N overview for date (Top-N-Übersicht für Datum).

   Standardmäßig werden im Abschnitt Top N overview for date (Top-N-Übersicht für Datum) Metriken für die drei wichtigsten Buckets angezeigt. Sie können die Anzahl der Buckets im Feld Top N erhöhen oder verringern. Der Abschnitt Top N overview for date (Top-N-Übersicht für Datum) zeigt auch die prozentuale Veränderung gegenüber dem Vortag oder der Vorwoche sowie eine Sparkline zur Veranschaulichung des Trends an. (Dieser Trend ist ein 14-Tage-Trend für kostenlose Metriken und ein 30-Tage-Trend für erweiterte Metriken und Empfehlungen.)

   > **Note**
   >
   > Mit den erweiterten Metriken und Empfehlungen von S3 Storage Lens stehen Metriken 15 Monate für Abfragen zur Verfügung. Weitere Informationen finden Sie unter Metrikauswahl.

5. Wählen Sie für Metric (Metrik) in der Kategorie Cost optimization (Kostenoptimierung) die Option Incomplete multipart upload bytes greater than 7 days old (Unvollständige mehrteilige Upload-Bytes, die älter als 7 Tage sind) aus.

   Unter Top number buckets (Top N-Buckets) sehen Sie die Buckets mit den unvollständigsten mehrteiligen Upload-Speicherbytes, die älter als 7 Tage sind.

6. Wenn Sie detailliertere Metriken auf Bucket-Ebene für unvollständige mehrteilige Uploads anzeigen möchten, scrollen Sie zum Seitenanfang und wählen Sie dann die Registerkarte Bucket aus.

7. Scrollen Sie nach unten zum Abschnitt Buckets. Wählen Sie für Metrics categories (Metrikkategorien) die Option Cost optimization (Kostenoptimierung) aus. Löschen Sie dann Summary (Zusammenfassung).

   Die Liste Buckets wird aktualisiert und enthält alle verfügbaren Metriken für Cost optimization (Kostenoptimierung) für die angezeigten Buckets.

8. Wenn Sie die Liste Buckets so filtern möchten, dass nur bestimmte Kostenoptimierungsmetriken angezeigt werden, wählen Sie das Präferenzensymbol ( ) aus.

9. Deaktivieren Sie die Schalter für alle Metriken zur Kostenoptimierung, bis nur noch Incomplete multipart upload bytes greater than 7 days old (Unvollständige mehrteilige Upload-Bytes, die älter als 7 Tage sind) und Abort incomplete multipart upload lifecycle rule count (Anzahl der Lebenszyklusregeln zum Abbrechen unvollständiger mehrteiliger Uploads) ausgewählt sind.

10. (Optional) Wählen Sie unter Page size (Seitengröße) die Anzahl der Buckets aus, die in der Liste angezeigt werden sollen.

11. Wählen Sie Bestätigen aus.

    Die Liste Buckets wird aktualisiert und zeigt Metriken auf Bucket-Ebene für unvollständige mehrteilige Uploads und die Anzahl der Lebenszyklusregeln an. Sie können diese Daten verwenden, um Buckets zu identifizieren, die die unvollständigsten mehrteiligen Upload-Bytes enthalten, die älter als 7 Tage sind und denen Lebenszyklusregeln fehlen, um unvollständige mehrteilige Uploads abzubrechen. Anschließend können Sie in der S3-Konsole zu diesen Buckets navigieren und Lebenszyklusregeln hinzufügen, um abgebrochene, unvollständige mehrteilige Uploads zu löschen.

Schritt 3: Hinzufügen einer Lebenszyklusregel, um unvollständige mehrteilige Uploads nach 7 Tagen zu löschen

Um unvollständige mehrteilige Uploads automatisch zu verwalten, können Sie die S3-Konsole verwenden, um eine Lebenszykluskonfiguration zu erstellen und unvollständige mehrteilige Upload-Bytes aus einem Bucket nach einer bestimmten Anzahl von Tagen ablaufen zu lassen. Weitere Informationen finden Sie unter Konfigurieren einer Bucket-Lebenszykluskonfiguration zum Löschen unvollständiger mehrteiliger Uploads.

Reduzieren Sie die Anzahl der beibehaltenen nicht aktuellen Versionen

Wenn die S3 Versioning aktiviert ist, werden mehrere Versionen desselben Objekts beibehalten, die zur schnellen Wiederherstellung von Daten verwendet werden können, wenn ein Objekt versehentlich gelöscht oder überschrieben wird. Wenn Sie S3 Versioning aktiviert haben, ohne Lebenszyklusregeln für die Umstellung oder den Ablauf nicht aktueller Versionen zu konfigurieren, kann sich eine große Anzahl nicht aktueller Versionen ansammeln, was sich auf die Speicherkosten auswirken kann. Weitere Informationen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

Schritt 1: Identifizieren von Buckets mit den meisten nicht aktuellen Objektversionen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie anzeigen möchten.
4. Wählen Sie im Abschnitt Snapshot for date (Snapshot für Datum) unter Metrics categories (Metrikkategorien) die Option Cost optimization (Kostenoptimierung) aus.

   Der Abschnitt Snapshot for date (Snapshot für Datum) wird aktualisiert und zeigt nun Metriken für Cost optimization (Kostenoptimierung) an, zu denen auch die Metrik % noncurrent version bytes (% Bytes der nicht aktuellen Version) gehört. Die Metrik % noncurrent version bytes (% Bytes der nicht aktuellen Version) stellt den Anteil Ihrer gesamten Speicherbytes dar, der im Rahmen des Dashboards und für das ausgewählte Datum auf nicht aktuelle Versionen zurückzuführen ist.

> **Note**
>
> Wenn der Wert % noncurrent version bytes (% Bytes der nicht aktuellen Version) mehr als 10 % Ihres Speicherplatzes auf Kontoebene beträgt, könnte dies ein Hinweis darauf sein, dass Sie zu viele Versionen speichern.

5. So identifizieren Sie bestimmte Buckets, in denen sich eine große Anzahl nicht aktueller Versionen ansammelt:

   a. Scrollen Sie nach unten zum Abschnitt Top N overview for date (Top-N-Übersicht für Datum). Geben Sie für Top N die Anzahl der Buckets ein, deren Daten Sie sehen möchten.

   b. Wählen Sie für Metric (Metrik) die Option % noncurrent version bytes (% Bytes der nicht aktuellen Version) aus.

      Unter Top number buckets (Top N-Buckets) werden die Buckets (für die von Ihnen angegebene Anzahl) mit dem höchsten Wert für % noncurrent version bytes (% Bytes der nicht aktuellen Version) angezeigt. Der Abschnitt Top N overview for date (Top-N-Übersicht für Datum) zeigt auch die prozentuale Veränderung gegenüber dem Vortag oder der Vorwoche sowie eine Sparkline zur Veranschaulichung des Trends an. Dieser Trend ist ein 14-Tage-Trend für kostenlose Metriken und ein 30-Tage-Trend für erweiterte Metriken und Empfehlungen.

      > **Note**
      >
      > Mit den erweiterten Metriken und Empfehlungen von S3 Storage Lens stehen Metriken 15 Monate für Abfragen zur Verfügung. Weitere Informationen finden Sie unter Metrikauswahl.

   c. Wenn Sie detailliertere Metriken auf Bucket-Ebene für nicht aktuelle Objektversionen anzeigen möchten, scrollen Sie zum Seitenanfang und wählen Sie dann die Registerkarte Bucket aus.

      In jedem Diagramm oder jeder Visualisierung Ihres S3-Storage-Lens-Dashboards können Sie mithilfe der Registerkarten Account (Konto), AWS-Region, Storage classe (Speicherklasse) oder Bucket tiefere Aggregationsebenen aufrufen. Ein Beispiel finden Sie unter Entdecken Sie kalte Amazon S3-Buckets.

d. Wählen Sie im Abschnitt Buckets für Metric categories (Metrikkategorien) die Option Cost optimization (Kostenoptimierung) aus. Löschen Sie dann Summary (Zusammenfassung).

   Sie können jetzt die Metrik % noncurrent version bytes (% Bytes der nicht aktuellen Version) zusammen mit anderen Metriken sehen, die sich auf nicht aktuelle Versionen beziehen.

Schritt 2: Identifizieren von Buckets, für die Lebenszyklusregeln für Übergang und Ablauf nicht aktueller Versionen fehlen

Voraussetzung

Wenn Sie in Ihrem S3-Storage-Lens-Dashboard die Metriken Noncurrent version transition lifecycle rule count (Anzahl der Lebenszyklusregeln für den Übergang nicht aktueller Versionen) und Noncurrent version expiration lifecycle rule count (Anzahl der Lebenszyklusregeln für den Ablauf nicht aktueller Versionen) zu sehen, müssen Sie Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) von S3 Storage Lens aktivieren und dann Advanced cost optimization metrics (Erweiterte Kostenoptimierungsmetriken) auswählen. Weitere Informationen finden Sie unter Verwenden der S3-Konsole.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie anzeigen möchten.
4. Wählen Sie in Ihrem Storage-Lens-Dashboard die Registerkarte Bucket aus.
5. Scrollen Sie nach unten zum Abschnitt Buckets. Wählen Sie für Metrics categories (Metrikkategorien) die Option Cost optimization (Kostenoptimierung) aus. Löschen Sie dann Summary (Zusammenfassung).

   Die Liste Buckets wird aktualisiert und enthält alle verfügbaren Metriken für Cost optimization (Kostenoptimierung) für die angezeigten Buckets.
6. Wenn Sie die Liste Buckets so filtern möchten, dass nur bestimmte Kostenoptimierungsmetriken angezeigt werden, wählen Sie das Präferenzensymbol ( ) aus.
7. Deaktivieren Sie die Schalter für alle Metriken zur Kostenoptimierung, bis nur noch die folgenden Metriken ausgewählt sind:

- % noncurrent version bytes (% Bytes der nicht aktuellen Version)
- Noncurrent version transition lifecycle rule count (Anzahl der Lebenszyklusregeln für den Übergang nicht aktueller Versionen)
- Noncurrent version expiration lifecycle rule count (Anzahl der Lebenszyklusregeln für den Ablauf nicht aktueller Versionen)

8. (Optional) Wählen Sie unter Page size (Seitengröße) die Anzahl der Buckets aus, die in der Liste angezeigt werden sollen.
9. Wählen Sie Bestätigen aus.

   Die Liste Buckets wird aktualisiert und zeigt Metriken für die Anzahl der Bytes nicht aktueller Versionen und die Anzahl der Lebenszyklusregeln für nicht aktuelle Versionen an. Sie können diese Daten verwenden, um Buckets zu identifizieren, die einen hohen Prozentsatz an Bytes nicht aktueller Versionen enthalten, für die jedoch Übergangs- und Ablauflebenszyklusregeln fehlen. Anschließend können Sie in der S3-Konsole zu diesen Buckets navigieren und ihnen Lebenszyklusregeln hinzufügen.

Schritt 3: Hinzufügen von Lebenszyklusregeln für den Übergang oder den Ablauf nicht aktueller Objektversionen

Nachdem Sie festgestellt haben, welche Buckets eine weitere Untersuchung erfordern, können Sie in der S3-Konsole zu den Buckets navigieren und eine Lebenszyklusregel hinzufügen, um nicht aktuelle Versionen nach einer bestimmten Anzahl von Tagen ablaufen zu lassen. Um die Kosten zu senken und gleichzeitig ältere Versionen beizubehalten, können Sie alternativ eine Lebenszyklusregel konfigurieren, um ältere Versionen auf eine der Amazon S3 Glacier Speicherklassen umzustellen. Weitere Informationen finden Sie unter Angabe einer Lebenszyklusregel für einen Bucket mit aktivierter Versionierung.

Identifizieren von Buckets ohne Lebenszyklusregeln und Überprüfen der Anzahl von Lebenszyklusregeln

S3 Storage Lens bietet Metriken zur Anzahl der S3-Lebenszyklusregeln, anhand derer Sie Buckets identifizieren können, denen Lebenszyklusregeln fehlen. Um Buckets zu finden, die keine Lebenszyklusregeln haben, können Sie die Metrik Total buckets without lifecycle rules (Buckets insgesamt ohne Lebenszyklusregeln) verwenden. Ein Bucket ohne S3-Lifecycle-Konfiguration verfügt möglicherweise über Speicher, den Sie nicht mehr benötigen oder den Sie zu einer kostengünstigeren Speicherklasse migrieren können. Sie können auch Metriken zur Anzahl von

Lebenszyklusregeln verwenden, um Buckets zu identifizieren, in denen bestimmte Arten von Lebenszyklusregeln fehlen, z. B. Ablauf- oder Übergangsregeln.

Voraussetzung

Wenn Sie in Ihrem S3-Storage-Lens-Dashboard die Metrik Total buckets without lifecycle rules (Buckets insgesamt ohne Lebenszyklusregeln) sehen möchten, müssen Sie Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) von S3 Storage Lens aktivieren und dann Advanced cost optimization metrics (Erweiterte Kostenoptimierungsmetriken) auswählen. Weitere Informationen finden Sie unter Verwenden der S3-Konsole.

Schritt 1: Identifizieren von Buckets ohne Lebenszyklusregeln

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie anzeigen möchten.
4. Wenn Sie bestimmte Buckets ohne Lebenszyklusregeln identifizieren möchten, scrollen Sie nach unten zum Abschnitt Top N overview for date (Top-N-Übersicht für Datum).

   Standardmäßig werden im Abschnitt Top N overview for date (Top-N-Übersicht für Datum) Metriken für die drei wichtigsten Buckets angezeigt. Im Feld Top N können Sie die Anzahl der Buckets erhöhen. Der Abschnitt Top N overview for date (Top-N-Übersicht für Datum) zeigt auch die prozentuale Veränderung gegenüber dem Vortag oder der Vorwoche sowie eine Sparkline zur Veranschaulichung des Trends an. Dieser Trend ist ein 14-Tage-Trend für kostenlose Metriken und ein 30-Tage-Trend für erweiterte Metriken und Empfehlungen.

   > **Note**
   >
   > Mit den erweiterten Metriken und Empfehlungen von S3 Storage Lens stehen Metriken 15 Monate für Abfragen zur Verfügung. Weitere Informationen finden Sie unter Metrikauswahl.

5. Wählen Sie für Metric (Metrik) in der Kategorie Cost optimization (Kostenoptimierung) die Option Total buckets without lifecycle rules (Buckets insgesamt ohne Lebenszyklusregeln) aus.
6. Überprüfen Sie die folgenden Daten für Total buckets without lifecycle rules (Buckets insgesamt ohne Lebenszyklusregeln):

- Top number accounts (Top N-Konten) – Sehen Sie sich an, welche Konten die meisten Buckets ohne Lebenszyklusregeln haben.
- Top number Regions (Top N-Regionen) – Sehen Sie sich eine Aufschlüsselung der Buckets ohne Lebenszyklusregeln nach Region an.
- Top number buckets (Top N-Buckets) – Finden Sie heraus, für welche Buckets keine Lebenszyklusregeln festgelegt sind.

In jedem Diagramm oder jeder Visualisierung Ihres S3-Storage-Lens-Dashboards können Sie mithilfe der Registerkarten Account (Konto), AWS-Region, Storage classe (Speicherklasse) oder Bucket tiefere Aggregationsebenen aufrufen. Ein Beispiel finden Sie unter Entdecken Sie kalte Amazon S3-Buckets.

Nachdem Sie ermittelt haben, für welche Buckets keine Lebenszyklusregeln festgelegt sind, können Sie auch die Anzahl der spezifischen Lebenszyklusregeln für Ihre Buckets überprüfen.

Schritt 2: Überprüfen der Anzahl der Lebenszyklusregeln für Ihre Buckets

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards das Dashboard aus, das Sie anzeigen möchten.
4. Wählen Sie in Ihrem S3-Storage-Lens-Dashboard die Registerkarte Bucket aus.
5. Scrollen Sie nach unten zum Abschnitt Buckets. Wählen Sie unter Metrics categories (Metrikkategorien) die Option Cost optimization (Kostenoptimierung) aus. Löschen Sie dann Summary (Zusammenfassung).

   Die Liste Buckets wird aktualisiert und enthält alle verfügbaren Metriken für Cost optimization (Kostenoptimierung) für die angezeigten Buckets.
6. Wenn Sie die Liste Buckets so filtern möchten, dass nur bestimmte Kostenoptimierungsmetriken angezeigt werden, wählen Sie das Präferenzensymbol ( ) aus.
7. Deaktivieren Sie die Schalter für alle Metriken zur Kostenoptimierung, bis nur noch die folgenden Metriken ausgewählt sind:

- Transition lifecycle rule count (Anzahl der Lebenszyklusregeln für den Übergang)
- Expiration lifecycle rule count (Anzahl der Lebenszyklusregeln für den Ablauf)
- Noncurrent version transition lifecycle rule count (Anzahl der Lebenszyklusregeln für den Übergang nicht aktueller Versionen)
- Noncurrent version expiration lifecycle rule count (Anzahl der Lebenszyklusregeln für den Ablauf nicht aktueller Versionen)
- Abort incomplete multipart upload lifecycle rule count (Anzahl der Lebenszyklusregeln zum Abbrechen unvollständiger mehrteiliger Uploads)
- Total lifecycle rule count (Anzahl der gesamten Lebenszyklusregeln)

8. (Optional) Wählen Sie unter Page size (Seitengröße) die Anzahl der Buckets aus, die in der Liste angezeigt werden sollen.
9. Wählen Sie Bestätigen aus.

   Die Liste Buckets wird aktualisiert und zeigt die die Metriken für die Anzahl der Lebenszyklusregeln für Ihre Buckets an. Sie können diese Daten verwenden, um Buckets ohne Lebenszyklusregeln oder Buckets zu identifizieren, für die bestimmte Arten von Lebenszyklusregeln fehlen, z. B. Ablaufregeln oder Übergangsregeln. Anschließend können Sie in der S3-Konsole zu diesen Buckets navigieren und ihnen Lebenszyklusregeln hinzufügen.

Schritt 3: Hinzufügen von Lebenszyklusregeln

Nachdem Sie Buckets ohne Lebenszyklusregeln identifiziert haben, können Sie Lebenszyklusregeln hinzufügen. Weitere Informationen erhalten Sie unter Einstellung einer S3-Lifecycle-Konfiguration für einen Bucket und Beispiele für S3-Lifecycle-Konfigurationen.

Verwenden von S3 Storage Lens zum Schutz Ihrer Daten

Sie können die Datenschutzmetriken von Amazon S3 Storage Lens verwenden, um Buckets zu identifizieren, in denen die bewährten Datenschutzmethoden nicht angewendet wurden. Sie können diese Metriken verwenden, um Maßnahmen zu ergreifen und Standardeinstellungen anzuwenden, die den bewährten Methoden zum Schutz Ihrer Daten in allen Buckets Ihres Kontos oder Ihrer Organisation entsprechen. Sie können beispielsweise Datenschutzmetriken verwenden, um Buckets zu identifizieren, die keine () -Schlüssel AWS Key Management Service (SSE-AWS KMS KMS) für die Standardverschlüsselung verwenden, oder Anfragen, die AWS Signature Version 2 (SigV2) verwenden.

Die folgenden Anwendungsfälle bieten Strategien zur Verwendung Ihres S3-Storage-Lens-Dashboards, um Ausreißer zu ermitteln und bewährte Datenschutzmethoden auf Ihre S3-Buckets anzuwenden.

Themen

- Identifizieren Sie Buckets, die keine serverseitige Verschlüsselung verwenden, mit AWS KMS der Standardverschlüsselung (-) SSE KMS
- Identifizieren von Buckets, für die S3 Versioning aktiviert ist
- Identifizieren von Anforderungen, die AWS Signature Version 2 (SigV2) verwenden
- Zählen der Gesamtzahl der Replikationsregeln für jeden Bucket
- Identifizieren des Prozentsatz der Bytes mit Objektsperre

Identifizieren Sie Buckets, die keine serverseitige Verschlüsselung verwenden, mit AWS KMS der Standardverschlüsselung (-) SSE KMS

Mit der Amazon-S3-Standardverschlüsselung können Sie das Verhalten der Standardverschlüsselung für einen S3-Bucket festlegen. Weitere Informationen finden Sie unter the section called “Festlegen der Standard-Bucket-Verschlüsselung”.

Sie können die Metriken SSE- KMS enabled bucket count und% SSE - KMS enabled Buckets verwenden, um Buckets zu identifizieren, die serverseitige Verschlüsselung mit AWS KMS Schlüsseln (SSE-KMS) für die Standardverschlüsselung verwenden. S3 Storage Lens bietet auch Metriken für unverschlüsselte Bytes, unverschlüsselte Objekte, verschlüsselte Bytes und verschlüsselte Objekte. Eine vollständige Liste der Metriken finden Sie unter Amazon S3-Storage-Lens-Metrikglossar.

Sie können die KMS Verschlüsselungsmetriken im Zusammenhang mit allgemeinen Verschlüsselungsmetriken analysieren, um Buckets zu identifizieren, die SSE - nicht verwenden. SSE KMS Wenn Sie SSE — KMS für alle Buckets in Ihrem Konto oder Ihrer Organisation verwenden möchten, können Sie anschließend die Standardverschlüsselungseinstellungen für diese Buckets aktualisieren, sodass Sie — verwenden können. SSE KMS Zusätzlich zu SSE - KMS können Sie serverseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) oder vom Kunden bereitgestellten Schlüsseln (SSE-C) verwenden. Weitere Informationen finden Sie unter Datenschutz durch Verschlüsselung.

Schritt 1: Identifizieren Sie, welche Buckets — für die Standardverschlüsselung verwenden SSE KMS

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.
4. Wählen Sie im Bereich Trends und Verteilungen die Option% SSE — KMS enabled bucket count für die primäre Metrik und% verschlüsselte Bytes für die sekundäre Metrik aus.

   Das Diagramm „Trend für Datumsdaten“ wird aktualisiert und zeigt nun Trends für SSE - KMS und verschlüsselte Byte an.
5. Um detailliertere Einblicke auf Bucketebene für - zu erhalten: SSE KMS

   a. Wählen Sie einen Punkt im Diagramm aus. Es erscheint ein Feld mit Auswahlmöglichkeiten für detailliertere Einblicke.

   b. Wählen Sie die Dimension Buckets aus. Wählen Sie dann Apply (Anwenden).
6. Wählen Sie im Diagramm Verteilung nach Zeiträumen für das Datum die Metrik „Anzahl der Buckets“ aus, die SSEKMSaktiviert ist.
7. Sie können jetzt sehen, für welche Buckets SSE — KMS aktiviert ist und für welche nicht.

Schritt 2: Aktualisieren der Standardverschlüsselungseinstellungen für den Bucket

Nachdem Sie nun ermittelt haben, welche Buckets SSE — KMS im Kontext Ihrer% verschlüsselten Byte — verwenden, können Sie Buckets identifizieren, die — nicht verwenden. SSE KMS Sie können dann optional in der S3-Konsole zu diesen Buckets navigieren und ihre Standardverschlüsselungseinstellungen so aktualisieren, dass sie SSE - KMS oder -S3 verwenden. SSE Weitere Informationen finden Sie unter Konfigurieren der Standardverschlüsselung.

Identifizieren von Buckets, für die S3 Versioning aktiviert ist

Wenn die S3-Versioningfunktion aktiviert ist, behält sie mehrere Versionen desselben Objekts bei, die zur schnellen Wiederherstellung von Daten verwendet werden können, wenn ein Objekt versehentlich gelöscht oder überschrieben wird. Sie können die Metrik Versioning-enabled bucket count (Bucket-Anzahl mit aktiviertem Versioning) verwenden, um zu sehen, welche Buckets S3 Versioning verwenden. Anschließend können Sie in der S3-Konsole Maßnahmen ergreifen, um S3 Versioning für andere Buckets zu aktivieren.

Schritt 1: Identifizieren von Buckets, für die S3 Versioning aktiviert ist

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Gehen Sie im Navigationsbereich auf Storage Lens und dann auf Dashboards.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.
4. Wählen Sie im Abschnitt Trends and distributions (Trends und Verteilungen) die Metrik Versioning-enabled bucket count (Bucket-Anzahl mit aktiviertem Versioning) als primäre Metrik und Buckets als sekundäre Metrik aus.

   Das Diagramm Trend for date (Trend für Datum) wird aktualisiert, um Trends für Buckets mit aktiviertem S3 Versioning anzuzeigen. Direkt unter der Trendlinie sehen Sie die Unterabschnitte Storage class distribution (Speicherklassenverteilung) und Region distribution (Regionale Verteilung).
5. Gehen Sie wie folgt vor, um detailliertere Einblicke für einen der Buckets im Diagramm Trend for date (Trend für Datum) zu sehen und eine eingehendere Analyse durchführen zu können:

   a. Wählen Sie einen Punkt im Diagramm aus. Es erscheint ein Feld mit Auswahlmöglichkeiten für detailliertere Einblicke.

   b. Wählen Sie eine Dimension aus, die Sie für eine eingehendere Analyse auf Ihre Daten anwenden möchten: Account (Konto), AWS-Region, Storage class (Speicherklasse) oder Bucket. Wählen Sie dann Apply (Anwenden).
6. Wählen Sie im Bubble analysis by buckets for date (Blasenanalyse nach Buckets für Datum) die Metriken Versioning-enabled bucket count (Bucket-Anzahl mit aktiviertem Versioning), Buckets und Active Buckets (Aktive Buckets) aus.

   Der Abschnitt Bubble analysis by buckets for date (Blasenanalyse nach Buckets für Datum) wird aktualisiert und zeigt nun Daten für die ausgewählten Metriken an. Sie können diese Daten verwenden, um zu sehen, für welche Buckets S3 Versioning im Kontext Ihrer gesamten Bucket-Anzahl aktiviert ist. Im Abschnitt Bubble analysis by buckets for date (Blasenanalyse nach Buckets für Datum) können Sie Ihre Buckets in mehreren Dimensionen darstellen, indem Sie drei beliebige Metriken verwenden, um die X-axis (x-Achse), Y-axis (y-Achse) und die Size (Größe) der Blase darzustellen.

Schritt 2: Aktivieren von S3 Versioning

Nachdem Sie Buckets identifiziert haben, für die S3 Versioning aktiviert ist, können Sie Buckets identifizieren, für die S3 Versioning noch nie aktiviert war oder deren Versionsverwaltung ausgesetzt ist. Anschließend können Sie die Versionsverwaltung für diese Buckets in der S3-Konsole optional aktivieren. Weitere Informationen finden Sie unter Aktivieren des Versioning für Buckets.

Identifizieren von Anforderungen, die AWS Signature Version 2 (SigV2) verwenden

Sie können die Metrik Alle nicht unterstützten Signaturanfragen verwenden, um Anfragen zu identifizieren, die AWS Signature Version 2 (SigV2) verwenden. Diese Daten können Ihnen helfen, bestimmte Anwendungen zu identifizieren, die SigV2 verwenden. Anschließend können Sie diese Anwendungen auf AWS Signature Version 4 (Sigv4) migrieren.

SigV4 ist die empfohlene Signaturmethode für alle neuen S3-Anwendungen. SigV4 bietet verbesserte Sicherheit und wird von allen unterstützt. AWS-Regionen Weitere Informationen finden Sie unter Amazon S3 Update – SigV2 Deprecation Period Extended and Modified.

Voraussetzung

Zum Anzeigen von All unsupported signature requests (Alle nicht unterstützten Signaturanforderungen) in Ihrem S3-Storage-Lens-Dashboard müssen Sie die Option Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) von S3 Storage Lens aktivieren und dann Advanced data protection metrics (Erweiterte Datensicherheitsmetriken) auswählen. Weitere Informationen finden Sie unter Verwenden der S3-Konsole.

Schritt 1: Untersuchen Sie die Sigv2-Signierungstrends nach AWS-Konto Region und Bereich

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.
4. So identifizieren Sie bestimmte Buckets, Konten und Regionen mit Anforderungen, die SigV2 verwenden:

    a. Geben Sie unter Top N overview for date (Top-N-Übersicht für Datum) im Feld Top N die Anzahl der Buckets ein, deren Daten Sie sehen möchten.

    b. Wählen Sie für Metric (Metrik) die Option All unsupported signature requests (Alle nicht unterstützten Signaturanforderungen) in der Kategorie Data protection (Datenschutz) aus.

Die Top-N-Übersicht für Datumsaktualisierungen zur Anzeige von Daten für SIGv2-Anfragen nach Konto, AWS-Region, und Bucket. Der Abschnitt Top N overview for date (Top-N-Übersicht für Datum) zeigt auch die prozentuale Veränderung gegenüber dem Vortag oder der Vorwoche sowie eine Sparkline zur Veranschaulichung des Trends an. Dieser Trend ist ein 14-Tage-Trend für kostenlose Metriken und ein 30-Tage-Trend für erweiterte Metriken und Empfehlungen.

> **Note**
>
> Mit den erweiterten Metriken und Empfehlungen von S3 Storage Lens stehen Metriken 15 Monate für Abfragen zur Verfügung. Weitere Informationen finden Sie unter Metrikauswahl.

Schritt 2: Identifizieren von Buckets, auf die Anwendungen über SigV2-Anforderungen zugreifen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.
4. Wählen Sie in Ihrem Storage-Lens-Dashboard die Registerkarte Bucket aus.
5. Scrollen Sie nach unten zum Abschnitt Buckets. Wählen Sie unter Metrics categories (Metrikkategorien) die Option Data protection (Datenschutz) aus. Löschen Sie dann Summary (Zusammenfassung).

   Die Liste Buckets wird aktualisiert und enthält alle verfügbaren Metriken für Data protection (Datenschutz) für die angezeigten Buckets.
6. Wenn Sie die Liste Buckets so filtern möchten, dass nur bestimmte Datenschutzmetriken angezeigt werden, wählen Sie das Präferenzensymbol ( ) aus.
7. Deaktivieren Sie die Schalter für alle Datenschutzmetriken, bis nur noch die folgenden Metriken ausgewählt sind:

   - All unsupported signature requests (Alle nicht unterstützten Signaturanforderungen)

- % all unsupported signature requests (% aller nicht unterstützten Signaturanforderungen)

8. (Optional) Wählen Sie unter Page size (Seitengröße) die Anzahl der Buckets aus, die in der Liste angezeigt werden sollen.
9. Wählen Sie Bestätigen aus.

   Die Liste Buckets wird aktualisiert, um Metriken auf Bucket-Ebene für SigV2-Anforderungen anzuzeigen. Sie können diese Daten verwenden, um bestimmte Buckets zu identifizieren, die SigV2-Anforderungen haben. Anschließend können Sie diese Informationen verwenden, um Ihre Anwendungen zu SigV4 zu migrieren. Weitere Informationen finden Sie unter Authentifizieren von Anfragen (AWS Signature Version 4) in der Amazon Simple Storage Service API Reference.

Zählen der Gesamtzahl der Replikationsregeln für jeden Bucket

S3 Replication ermöglicht das automatische, asynchrone Kopieren von Objekten in Amazon-S3-Buckets. Buckets, die für die Objektreplikation konfiguriert sind, können sich im Besitz desselben AWS-Konto oder unterschiedlicher Konten befinden. Weitere Informationen finden Sie unter Objekte innerhalb und zwischen Regionen replizieren.

Sie können die Metriken zur Anzahl der Replikationsregeln von S3 Storage Lens verwenden, um detaillierte Informationen pro Bucket zu Ihren Buckets zu erhalten, die für die Replikation konfiguriert sind. Zu diesen Informationen gehören Replikationsregeln innerhalb von Buckets und Regionen sowie Bucket- und regionsübergreifende Replikationsregeln.

Voraussetzung

Zum Anzeigen von Metriken zur Anzahl der Replikationsregeln in Ihrem S3-Storage-Lens-Dashboard müssen Sie die Option Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) von S3 Storage Lens aktivieren und dann Advanced data protection metrics (Erweiterte Datensicherheitsmetriken) auswählen. Weitere Informationen finden Sie unter Verwenden der S3-Konsole.

Schritt 1: Zählen der Gesamtzahl der Replikationsregeln für jeden Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.

4. Wählen Sie in Ihrem Storage-Lens-Dashboard die Registerkarte Bucket aus.
5. Scrollen Sie nach unten zum Abschnitt Buckets. Wählen Sie unter Metrics categories (Metrikkategorien) die Option Data protection (Datenschutz) aus. Löschen Sie dann Summary (Zusammenfassung).
6. Wenn Sie die Liste Buckets so filtern möchten, dass nur Metriken zur Anzahl der Replikationsregeln angezeigt werden, wählen Sie das Präferenzensymbol ( ) aus.
7. Deaktivieren Sie die Schalter für alle Datenschutzmetriken, bis nur noch die Metriken zur Anzahl der Replikationsregeln ausgewählt sind:
   - Same-Region Replication rule count (Anzahl der Replikationsregeln für dieselbe Region)
   - Cross-Region Replication rule count (Anzahl der regionsübergreifenden Replikationsregeln)
   - Same-account replication rule count (Anzahl der Replikationsregeln für dasselbe Konto)
   - Cross-account replication rule count (Kontenübergreifende Replikationsregelanzahl)
   - Total replication rule count (Gesamte Replikationsregelnanzahl)
8. (Optional) Wählen Sie unter Page size (Seitengröße) die Anzahl der Buckets aus, die in der Liste angezeigt werden sollen.
9. Wählen Sie Bestätigen aus.

Schritt 2: Hinzufügen von Replikationsregeln

Nachdem Sie die Anzahl der Replikationsregeln pro Bucket festgelegt haben, können Sie optional weitere Replikationsregeln erstellen. Weitere Informationen finden Sie unter Beispiele für die Konfiguration der Live-Replikation.

Identifizieren des Prozentsatz der Bytes mit Objektsperre

Mit S3 Object Lock können Sie Objekte mithilfe eines write-once-read-many (WORM) -Modells speichern. Mit der Objektsperre können Sie verhindern, dass Objekte für einen bestimmten Zeitraum oder auf unbestimmte Zeit gelöscht oder überschrieben werden. Sie können die Objektsperre nur aktivieren, wenn Sie einen Bucket erstellen und S3 Versioning aktivieren. Sie können jedoch den Aufbewahrungszeitraum für einzelne Objektversionen bearbeiten oder gesetzliche Aufbewahrungsfristen für Buckets anwenden, für die die Objektsperre aktiviert ist. Weitere Informationen finden Sie unter Sperren von Objekten mit Object Lock.

Sie können Metriken für die Objektsperre in S3 Storage Lens verwenden, um die Metrik % Object Lock bytes (% Bytes mit Objektsperre) für Ihr Konto oder Ihre Organisation anzuzeigen. Sie können diese Informationen verwenden, um Buckets in Ihrem Konto oder Ihrer Organisation zu identifizieren, die Ihre bewährten Datenschutzmethoden nicht einhalten.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.
4. Wählen Sie im Abschnitt Snapshot unter Metrics categories (Metrikkategorien) die Option Data protection (Datenschutz) aus.

   Der Abschnitt Snapshot wird aktualisiert und zeigt nun Datenschutzmetriken an, einschließlich der Metrik % Object Lock bytes (% Bytes mit Objektsperre). Sie können den Gesamtprozentsatz der Bytes mit Objektsperre für Ihr Konto oder Ihre Organisation sehen.
5. Wenn Sie die Metrik % Object Lock bytes (% Bytes mit Objektsperre) pro Bucket sehen möchten, scrollen Sie nach unten zum Abschnitt Top N overview (Top-N-Übersicht).

   Wenn Sie Daten mit Objektsperre auf Objektebene abrufen möchten, können Sie auch die Metriken Object Lock object count (Objektsperren-Objektanzahl) und % Object Lock objects (% Objekte mit Objektsperre) verwenden.
6. Wählen Sie für Metric (Metrik) die Option % Object Lock Bytes (% Bytes mit Objektsperre) aus der Kategorie Data protection (Datenschutz) aus.

   Standardmäßig werden im Abschnitt Top N overview for date (Top-N-Übersicht für Datum) Metriken für die drei wichtigsten Buckets angezeigt. Im Feld Top N können Sie die Anzahl der Buckets erhöhen. Der Abschnitt Top N overview for date (Top-N-Übersicht für Datum) zeigt auch die prozentuale Veränderung gegenüber dem Vortag oder der Vorwoche sowie eine Sparkline zur Veranschaulichung des Trends an. Dieser Trend ist ein 14-Tage-Trend für kostenlose Metriken und ein 30-Tage-Trend für erweiterte Metriken und Empfehlungen.

> **Note**
>
> Mit den erweiterten Metriken und Empfehlungen von S3 Storage Lens stehen Metriken 15 Monate für Abfragen zur Verfügung. Weitere Informationen finden Sie unter Metrikauswahl.

7. Überprüfen Sie die folgenden Daten für % Object Lock bytes (% Objekte mit Objektsperre):

   - Top number accounts (Top N-Konten) – Sehen Sie sich an, welche Konten den höchsten und den niedrigsten Wert für % Object Lock bytes (% Bytes mit Objektsperre) haben.
   - Top number Regions (Top N-Regionen) – Zeigt eine Aufschlüsselung der % Object Lock bytes (% Bytes mit Objektsperre) nach Region an.
   - Top number buckets (Top N-Buckets) – Sehen Sie sich an, welche Buckets den höchsten und den niedrigsten Wert für % Object Lock bytes (% Bytes mit Objektsperre) haben.

Verwenden von S3 Storage Lens zur Überwachung der Einstellungen für die Objekteigentümerschaft

Amazon S3 Object Ownership ist eine S3-Einstellung auf Bucket-Ebene, mit der Sie Zugriffskontrolllisten (ACLs) deaktivieren und den Besitz der Objekte in Ihrem Bucket kontrollieren können. Wenn Sie Object Ownership auf Bucket Owner erforced setzen, können Sie die Zugriffskontrolllisten (ACLs) deaktivieren und die Verantwortung für jedes Objekt in Ihrem Bucket übernehmen. Dieser Ansatz vereinfacht die Zugriffsverwaltung für in Amazon S3 gespeicherte Daten.

Wenn ein anderer Benutzer ein Objekt in Ihren S3-Bucket AWS-Konto hochlädt, besitzt dieses Konto (der Objekt-Writer) standardmäßig das Objekt, hat Zugriff darauf und kann anderen Benutzern Zugriff darauf gewähren. ACLs Sie können Object Ownership verwenden, um dieses Standardverhalten zu ändern.

Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs. Wir empfehlen Ihnen daher, die Option zu deaktivierenACLs, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Wenn Sie „Objektbesitz“ auf „Bucket Owner erforced“ setzen, können Sie Richtlinien für die Zugriffskontrolle deaktivieren ACLs und sich auf diese verlassen. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

Mit den Zugriffsmanagement-Metriken von S3 Storage Lens können Sie Buckets identifizieren, die nicht deaktiviert wurden. ACLs Nachdem Sie diese Buckets identifiziert haben, können Sie ACL Berechtigungen auf Richtlinien migrieren und für diese Buckets deaktivierenACLs.

Themen

- Schritt 1: Identifizieren allgemeiner Trends für die Einstellungen der Objekteigentümerschaft
- Schritt 2: Identifizieren von Trends auf Bucket-Ebene für die Einstellungen der Objekteigentümerschaft
- Schritt 3: Ändern Sie Ihre Einstellung „Object Ownership“ auf „Bucket Owner erforced“, um sie zu deaktivieren ACLs

Schritt 1: Identifizieren allgemeiner Trends für die Einstellungen der Objekteigentümerschaft

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.
4. Wählen Sie im Abschnitt Snapshot for date (Snapshot für Datum) unter Metrics categories (Metrikkategorien) die Option Access management (Zugriffsverwaltung) aus.

   Der Abschnitt Snapshot for date (Snapshot für Datum) wird aktualisiert und zeigt nun die Metrik % Object Ownership bucket owner enforced (% Bucket-Eigentümer erzwungen für Objekteigentümerschaft) an. Sie können den Gesamtanteil der Buckets in Ihrem Konto oder Ihrer Organisation sehen, bei denen die vom Bucket Ownership erzwungene Einstellung für Object Ownership zur Deaktivierung verwendet wurde. ACLs

Schritt 2: Identifizieren von Trends auf Bucket-Ebene für die Einstellungen der Objekteigentümerschaft

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.

4. Wenn Sie detailliertere Metriken auf Bucket-Ebene anzeigen möchten, wählen Sie die Registerkarte Bucket aus.
5. Wählen Sie im Abschnitt Distribution by buckets for date (Verteilung nach Buckets für Datum) die Metrik % Object Ownership bucket owner enforced (% Bucket-Eigentümer erzwungen für Objekteigentümerschaft) aus.

   Das Diagramm wird aktualisiert und zeigt nun eine Aufschlüsselung je Bucket für % Object Ownership bucket owner enforced (% Bucket-Eigentümer erzwungen für Objekteigentümerschaft) an. Sie können sehen, in welchen Buckets die vom Bucket-Besitzer erzwungene Einstellung für Object Ownership zur Deaktivierung verwendet wird. ACLs
6. Wenn Sie die Einstellungen „Bucket-Eigentümer erzwungen“ im Kontext sehen möchten, scrollen Sie nach unten zum Abschnitt Buckets. Wählen Sie für Metrics categories (Metrikkategorien) die Option Access management (Zugriffsverwaltung) aus. Löschen Sie dann Summary (Zusammenfassung).

   In der Liste Buckets werden Daten für alle drei Einstellungen für die Objekteigentümerschaft angezeigt: „Bucket owner enforced“ (Bucket-Eigentümer erzwungen), „Bucket owner preferred (Bucket-Eigentümer bevorzugt und „Object writer“ (Objektschreiber).
7. Wenn Sie die Liste Buckets so filtern möchten, dass Metriken nur für eine bestimmte Einstellung der Objekteigentümerschaft angezeigt werden, wählen Sie das Präferenzensymbol ( ).
8. Löschen Sie die Metriken, die Sie nicht anzeigen möchten.
9. (Optional) Wählen Sie unter Page size (Seitengröße) die Anzahl der Buckets aus, die in der Liste angezeigt werden sollen.
10. Wählen Sie Bestätigen aus.

Schritt 3: Ändern Sie Ihre Einstellung „Object Ownership“ auf „Bucket Owner erforced“, um sie zu deaktivieren ACLs

Nachdem Sie Buckets identifiziert haben, die die bevorzugte Einstellung Object Writer und Bucket Owner für Object Ownership verwenden, können Sie Ihre ACL Berechtigungen auf Bucket-Richtlinien migrieren. Wenn Sie mit der Migration Ihrer ACL Berechtigungen fertig sind, können Sie Ihre Einstellungen für den Objekteigentum auf „Bucket Owner Enforced“ ändern, um sie zu deaktivieren. ACLs Weitere Informationen finden Sie unter Voraussetzungen für die Deaktivierung ACLs.

Verwenden der Metriken von S3 Storage Lens zur Verbesserung der Leistung

Wenn Sie die Option S3 Storage Lens advanced metrics (Erweiterte Metriken von S3 Storage Lens) aktiviert haben, können Sie die detaillierten Statuscode-Metriken verwenden, um die Anzahl von erfolgreichen oder fehlgeschlagenen Anforderungen zu erhalten. Sie können diese Informationen verwenden, um Probleme mit dem Zugriff oder der Leistung zu beheben. Detaillierte Statuscode-Metriken zeigen die Anzahl der HTTP Statuscodes wie 403 Forbidden und 503 Service Unavailable. Sie können allgemeine Trends anhand detaillierter Statuscode-Metriken für S3-Buckets, Konten und Organisationen untersuchen. Anschließend können Sie sich Metriken auf Bucket-Ebene genauer ansehen, um Workloads zu identifizieren, die derzeit auf diese Buckets zugreifen und Fehler verursachen.

Sie können sich beispielsweise die Metrik 403 Forbidden error count (Anzahl der Fehler 403 Forbidden) ansehen, um Workloads zu identifizieren, die auf Buckets zugreifen, ohne dass die korrekten Berechtigungen angewendet wurden. Nachdem Sie diese Workloads identifiziert haben, können Sie außerhalb von S3 Storage Lens detaillierte Einblicke erhalten, um Ihre „403 Forbidden“-Fehler zu beheben.

Dieses Beispiel zeigt Ihnen, wie Sie eine Trendanalyse für den Fehler „403 Forbidden“ durchführen, indem Sie die Metriken 403 Forbidden error count (Anzahl der Fehler 403 Forbidden) und % 403 Forbidden errors (% Fehler 403 Forbidden) verwenden. Anhand dieser Metriken können Sie Workloads identifizieren, die auf Buckets zugreifen, ohne dass die korrekten Berechtigungen angewendet wurden. Eine ähnliche Trendanalyse können Sie für jede der anderen Detailed status code metrics (Detaillierte Statuscode-Metriken) durchführen. Weitere Informationen finden Sie unter Amazon S3-Storage-Lens-Metrikglossar.

Voraussetzung

Wenn Sie Detailed status code metrics (Detaillierte Statuscode-Metriken) in Ihrem S3-Storage-Lens-Dashboard anzeigen möchten, müssen Sie die Option Advanced metrics and recommendations (Erweiterte Metriken und Empfehlungen) von S3 Storage Lens aktivieren und dann Detailed status code metrics (Detaillierte Statuscode-Metriken) auswählen. Weitere Informationen finden Sie unter Verwenden der S3-Konsole.

Themen

- Schritt 1: Führen Sie eine Trendanalyse für einen einzelnen HTTP Statuscode durch
- Schritt 2: Analysieren der Fehleranzahl nach Bucket
- Schritt 3: Fehlerbehebung

Schritt 1: Führen Sie eine Trendanalyse für einen einzelnen HTTP Statuscode durch

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.
4. Wählen Sie im Abschnitt Trends and distributions (Trends und Verteilungen) für Primary metric (Primäre Metrik) in der Kategorie Detailed status codes (Detaillierte Statuscodes) die Option 403 Forbidden error count (Anzahl der Fehler 403 Forbidden) aus. Wählen Sie für Secondary metric (Sekundäre Metrik) die Option % 403 Forbidden errors (% Fehler 403 Forbidden) aus.
5. Scrollen Sie nach unten zum Abschnitt Top N overview for date (Top-N-Übersicht für Datum). Wählen Sie für Metrics (Metriken) die Option 403 Forbidden error count (Anzahl der Fehler 403 Forbidden) oder % 403 Forbidden errors (% Fehler 403 Forbidden) in der Kategorie Detailed status codes (Detaillierte Statuscodes) aus.

   Der Bereich Top-N-Übersicht für das Datum wurde aktualisiert und zeigt nun die Anzahl der häufigsten 403 verbotenen Fehler nach Konto AWS-Region, und Bucket an.

Schritt 2: Analysieren der Fehleranzahl nach Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Storage Lens und dann Dashboards aus.
3. Wählen Sie in der Liste Dashboards den Namen des Dashboards aus, das Sie anzeigen möchten.
4. Wählen Sie in Ihrem Storage-Lens-Dashboard die Registerkarte Bucket aus.
5. Scrollen Sie nach unten zum Abschnitt Buckets. Wählen Sie für Metrics categories (Metrikkategorien) die Option Detailed status code metrics (Detaillierte Statuscode-Metriken) aus. Löschen Sie dann Summary (Zusammenfassung).

   Die Liste Buckets wird aktualisiert und zeigt alle verfügbaren detaillierten Statuscode-Metriken an. Anhand dieser Informationen können Sie herausfinden, welche Buckets einen großen Anteil an bestimmten HTTP Statuscodes haben und welche Statuscodes in allen Buckets üblich sind.

6. Wenn Sie die Liste Buckets so filtern möchten, dass nur bestimmte detaillierte Statuscode-Metriken angezeigt werden, wählen Sie das Präferenzensymbol ( ).
7. Deaktivieren Sie die Schalter für alle detaillierten Statuscode-Metriken, die Sie nicht in der Liste Buckets anzeigen möchten.
8. (Optional) Wählen Sie unter Page size (Seitengröße) die Anzahl der Buckets aus, die in der Liste angezeigt werden sollen.
9. Wählen Sie Bestätigen aus.

   In der Liste Buckets werden Metriken zur Fehleranzahl für die von Ihnen angegebene Anzahl von Buckets angezeigt. Sie können diese Informationen verwenden, um bestimmte Buckets zu identifizieren, in denen viele Fehler auftreten, und um Fehler nach Bucket zu beheben.

Schritt 3: Fehlerbehebung

Nachdem Sie Buckets mit einem hohen Anteil an bestimmten HTTP Statuscodes identifiziert haben, können Sie diese Fehler beheben. Weitere Informationen finden Sie hier:

- Warum erhalte ich die Fehlermeldung „403 Forbidden“, wenn ich versuche, Dateien in Amazon S3 hochzuladen?
- Warum erhalte ich die Fehlermeldung „403 Forbidden“, wenn ich versuche, eine Bucket-Richtlinie in Amazon S3 zu ändern?
- Wie behebe ich „403 Forbidden“-Fehler in meinem Amazon-S3-Bucket, bei dem alle Ressourcen aus denselben AWS-Konto stammen?
- Wie behebe ich einen HTTP 500- oder 503-Fehler von Amazon S3?

Verwenden von Amazon S3 Storage Lens mit AWS Organizations

Amazon S3 Storage Lens ist eine Cloud-Speicheranalysefunktion, mit der Sie unternehmensweite Einblicke in die Nutzung und Aktivität von Objektspeichern erhalten können. Sie können Metriken von S3 Storage Lens verwenden, um zusammenfassende Erkenntnisse zu gewinnen und z. B. herauszufinden, wie viel Speicher Sie in Ihrer gesamten Organisation haben oder welche Buckets und Präfixe am schnellsten wachsen. Sie können Amazon S3 Storage Lens auch verwenden, um Speichermetriken und Nutzungsdaten für alle zu sammeln AWS-Konten , die Teil Ihrer AWS Organizations Hierarchie sind. Dazu müssen Sie den vertrauenswürdigen Zugriff von AWS

Organizations S3 Storage Lens mithilfe Ihres AWS Organizations Verwaltungskontos verwenden und aktivieren.

Nachdem Sie den vertrauenswürdigen Zugriff aktiviert haben, fügen Sie delegierten Administratorzugriff auf Konten in Ihrer Organisation hinzu. Die delegierten Administratorkonten werden verwendet, um Konfigurationen und Dashboards für S3 Storage Lens zu erstellen, in denen unternehmensweite Speichermetriken und Benutzerdaten erfasst werden. Weitere Informationen zum Aktivieren des vertrauenswürdigen Zugriffs finden Sie unter Amazon S3 Storage Lens und AWS Organizations im AWS Organizations -Benutzerhandbuch.

Themen

- Aktivieren des vertrauenswürdigen Zugriffs für S3 Storage Lens
- Deaktivieren des vertrauenswürdigen Zugriffs für S3 Storage Lens
- Registrieren eines delegierten Administrators für S3 Storage Lens
- Aufheben der Registrierung eines delegierten Administrators für S3 Storage Lens

Aktivieren des vertrauenswürdigen Zugriffs für S3 Storage Lens

Durch die Aktivierung des vertrauenswürdigen Zugriffs ermöglichen Sie Amazon S3 Storage Lens, über AWS Organizations API Operationen auf Ihre AWS Organizations Hierarchie, Mitgliedschaft und Struktur zuzugreifen. S3 Storage Lens wird dann zu einem vertrauenswürdigen Service für die Struktur Ihrer gesamten Organisation.

Immer wenn eine Dashboard-Konfiguration definiert wird, erstellt S3 Storage Lens serviceverknüpfte Rollen in den Verwaltungskonten oder den delegierten Administratorkonten Ihrer Organisation. Die mit dem Service verknüpfte Rolle erteilt S3 Storage Lens die Erlaubnis, die folgenden Aktionen auszuführen:

- Beschreiben von Organisationen
- Auflisten von Konten
- Überprüfen Sie eine AWS-Service Zugriffsliste für die Organisationen
- Abrufen delegierter Administratoren für die Organisationen

S3 Storage Lens kann dann sicherstellen, dass es Zugriff auf die Erfassung der kontoübergreifenden Kennzahlen für die Konten in Ihrer Organisation hat. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für Amazon S3 Storage Lens.

Nachdem Sie den vertrauenswürdigen Zugriff aktiviert haben, können Sie anderen Konten in Ihrer Organisation den delegierten Administratorzugriff zuweisen. Wenn ein Konto als delegierter Administrator für einen Dienst gekennzeichnet ist, erhält das Konto die Autorisierung für den Zugriff auf alle schreibgeschützten Organisationsvorgänge. API Dieser Zugriff macht die Mitglieder und Strukturen Ihrer Organisation für den delegierten Administrator sichtbar, sodass dieser ebenfalls S3-Storage-Lens-Dashboards erstellen können.

> **Note**
>
> - Vertrauenswürdiger Zugriff kann nur im Verwaltungskonto aktiviert werden.
> - S3-Storage-Lens-Dashboards oder -Konfigurationen für Ihre Organisation können nur über das Verwaltungskonto und von delegierten Administratoren erstellt werden.

Verwenden der S3-Konsole

Um S3 Storage Lens einen AWS Organizations vertrauenswürdigen Zugriff zu ermöglichen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Navigieren Sie im linken Navigationsbereich zu Storage Lens.
3. Wählen Sie AWS Organizations Einstellungen aus. Die Seite „AWS Organizations Zugriff auf Storage Lens“ wird angezeigt.
4. Wählen Sie unter AWS Organizations Vertrauenswürdiger Zugriff die Option Bearbeiten aus.

   Die AWS Organizations Zugriffsseite wird angezeigt.
5. Wählen Sie Aktivieren, um den vertrauenswürdigen Zugriff für Ihr S3 Storage Lens-Dashboard zu aktivieren.
6. Wählen Sie Änderungen speichern.

Verwenden Sie den AWS CLI

Example

Das folgende Beispiel zeigt Ihnen, wie Sie den AWS Organizations vertrauenswürdigen Zugriff für S3 Storage Lens in aktivieren AWS CLI.

```
aws organizations enable-aws-service-access --service-principal storage-
lens.s3.amazonaws.com
```

Verwendung des AWS SDK for Java

Example — Aktivieren Sie den AWS Organizations vertrauenswürdigen Zugriff für S3 Storage Lens mithilfe von SDK Java

Das folgende Beispiel zeigt Ihnen, wie Sie den vertrauenswürdigen Zugriff für S3 Storage Lens in SDK für Java aktivieren. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import com.amazonaws.services.organizations.model.EnableAWSServiceAccessRequest;

public class EnableOrganizationsTrustedAccess {
 private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

 public static void main(String[] args) {
  try {
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            organizationsClient.enableAWSServiceAccess(new
 EnableAWSServiceAccessRequest()
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but AWS Organizations couldn't
 process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // AWS Organizations couldn't be contacted for a response, or the client
            // couldn't parse the response from AWS Organizations.
```

```
            e.printStackTrace();
        }
 }
}
```

Deaktivieren des vertrauenswürdigen Zugriffs für S3 Storage Lens

Wenn Sie ein Konto als delegierter Administrator entfernen oder den vertrauenswürdigen Zugriff deaktivieren, werden die S3 Storage Lens-Dashboard-Metriken des Kontoinhabers so eingeschränkt, dass sie nur auf Kontoebene funktionieren. Jeder Kontoinhaber kann dann die Vorteile von S3 Storage Lens nur für den begrenzten Umfang seines Kontos nutzen, nicht aber für sein gesamtes Unternehmen.

Wenn Sie den vertrauenswürdigen Zugriff in S3 Storage Lens deaktivieren, werden alle Dashboards, die vertrauenswürdigen Zugriff erfordern, nicht mehr aktualisiert. Alle erstellten organisatorischen Dashboards werden ebenfalls nicht mehr aktualisiert. Stattdessen können Sie nur historische Daten für das S3 Storage Lens-Dashboard abfragen, solange die Daten noch verfügbar sind.

> **Note**
>
> - Durch Deaktivieren des vertrauenswürdigen Zugriffs für S3 Storage Lens wird außerdem die Erfassung und Aggregation von Speichermetriken durch alle Dashboards auf Organisationsebene automatisch beendet. Dies liegt daran, dass S3 Storage Lens keinen vertrauenswürdigen Zugriff mehr auf die Unternehmenskonten hat.
> - Ihre Verwaltungs- und Delegiertenadministratorkonten können weiterhin die historischen Daten aller deaktivierten Dashboards einsehen. Sie können diese historischen Daten auch abfragen, solange sie noch verfügbar sind.

Verwenden der S3-Konsole

So deaktivieren Sie den vertrauenswürdigen Zugriff für S3 Storage Lens

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Navigieren Sie im linken Navigationsbereich zu Storage Lens.
3. Wählen Sie AWS Organizations Einstellungen aus. Die Seite „AWS Organizations Zugriff auf Storage Lens“ wird angezeigt.

4. Wählen Sie unter AWS Organizations Vertrauenswürdiger Zugriff die Option Bearbeiten aus.

   Die AWS Organizations Zugriffsseite wird angezeigt.

5. Wählen Sie Deaktivieren, um den vertrauenswürdigen Zugriff für Ihr S3 Storage Lens-Dashboard zu deaktivieren.
6. Wählen Sie Änderungen speichern.

Verwenden Sie den AWS CLI

Example

Das folgende Beispiel deaktiviert den vertrauenswürdigen Zugriff für S3 Storage Lens mithilfe von. AWS CLI

```
aws organizations disable-aws-service-access --service-principal storage-
lens.s3.amazonaws.com
```

Verwendung des AWS SDK for Java

Example — Deaktiviert den AWS Organizations vertrauenswürdigen Zugriff für S3 Storage Lens

Das folgende Beispiel zeigt Ihnen, wie Sie den AWS Organizations vertrauenswürdigen Zugriff für S3 Storage Lens in SDK Java deaktivieren. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import com.amazonaws.services.organizations.model.DisableAWSServiceAccessRequest;

public class DisableOrganizationsTrustedAccess {
 private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

 public static void main(String[] args) {
  try {
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
```

```
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            // Make sure to remove any existing delegated administrator for S3 Storage
 Lens
            // before disabling access; otherwise, the request will fail.
            organizationsClient.disableAWSServiceAccess(new
 DisableAWSServiceAccessRequest()
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but AWS Organizations couldn't
 process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // AWS Organizations couldn't be contacted for a response, or the client
            // couldn't parse the response from AWS Organizations.
            e.printStackTrace();
        }
 }
}
```

Registrieren eines delegierten Administrators für S3 Storage Lens

Dashboards auf Organisationsebene können mithilfe des Verwaltungskontos Ihrer Organisation oder mit delegierten Administratorkonten erstellt werden. Der delegierte Administratorzugriff ermöglicht anderen Konten neben dem Verwaltungskonto, Dashboards auf Organisationsebene zu erstellen. Das Registrieren und Aufheben der Registrierung anderer Konten als delegierte Administratoren für eine Organisation kann lediglich über das Verwaltungskonto erfolgen.

Nachdem Sie den vertrauenswürdigen Zugriff aktiviert haben, können Sie den delegierten Administratorzugriff auf Konten in Ihrer Organisation registrieren, indem Sie das AWS Organizations RESTAPI, AWS CLI, oder SDKs vom Verwaltungskonto aus verwenden. (Weitere Informationen finden Sie unter RegisterDelegatedAdministratorin der AWS Organizations APIReferenz.) Wenn ein Konto als delegierter Administrator registriert wird, erhält das Konto die Autorisierung für den Zugriff auf alle schreibgeschützten Operationen AWS Organizations API. Das macht die Mitglieder und Strukturen Ihrer Organisation sichtbar, sodass die delegierten Administratoren in Ihrem Namen S3-Storage-Lens-Dashboards erstellen können.

> **Note**
>
> Bevor Sie mithilfe von, oder SDKs einen delegierten Administrator bestimmen können AWS Organizations REST API AWS CLI, müssen Sie den EnableAWSOrganizationsAccessVorgang.

Verwenden der S3-Konsole

So registrieren Sie delegierte Administratoren für S3 Storage Lens

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Navigieren Sie im linken Navigationsbereich zu Storage Lens.
3. Wählen Sie AWS Organizations Einstellungen aus.
4. Wählen Sie unter Delegierte Administratoren die Option Konto registrieren aus.
5. Fügen Sie eine AWS-Konto ID hinzu, um das Konto als delegierten Administrator zu registrieren. Der delegierte Administrator kann Dashboards auf Organisationsebene für alle Konten und Speicher in Ihrer Organisation erstellen.
6. Wählen Sie Konto registrieren.

Mit dem AWS CLI

Example

Das folgende Beispiel zeigt Ihnen, wie Sie delegierte Administratoren von Organizations für S3 Storage Lens mithilfe der AWS CLI registrieren. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws organizations register-delegated-administrator --service-principal storage-
lens.s3.amazonaws.com --account-id 111122223333
```

Verwendung des AWS SDK for Java

Example — Delegierte Administratoren von Organizations für S3 Storage Lens registrieren

Das folgende Beispiel zeigt Ihnen, wie Sie AWS Organizations delegierte Administratoren für S3 Storage Lens in SDK für Java registrieren. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import
 com.amazonaws.services.organizations.model.RegisterDelegatedAdministratorRequest;

public class RegisterOrganizationsDelegatedAdministrator {
 private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

 public static void main(String[] args) {
  try {
            String delegatedAdminAccountId = "111122223333"; // Account Id for the
 delegated administrator.
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            organizationsClient.registerDelegatedAdministrator(new
 RegisterDelegatedAdministratorRequest()
                .withAccountId(delegatedAdminAccountId)
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but AWS Organizations couldn't
 process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // AWS Organizations couldn't be contacted for a response, or the client
            // couldn't parse the response from AWS Organizations.
            e.printStackTrace();
```

```
        }
 }
}
```

Aufheben der Registrierung eines delegierten Administrators für S3 Storage Lens

Nachdem Sie den vertrauenswürdigen Zugriff aktiviert haben, können Sie auch den delegierten Administratorzugriff auf Konten in Ihrer Organisation abmelden. Delegierte Administratorkonten ermöglichen es anderen Konten als Ihrem Verwaltungskonto, Dashboards auf Organisationsebene zu erstellen. Konten, die als delegierte Administratoren festgelegt wurden, können nur über das Verwaltungskonto einer Organisation abgemeldet werden.

Sie können einen delegierten Administrator abmelden, indem Sie das AWS Organizations AWS Management Console, RESTAPI, AWS CLI oder vom Verwaltungskonto aus verwenden. AWS SDKS Weitere Informationen finden Sie unter DeregisterDelegatedAdministratorin der Referenz.AWS Organizations API

Wenn die Registrierung eines Kontos als delegierter Administrator aufgehoben wird, verliert das Konto den Zugriff auf Folgendes:

- Alle schreibgeschützten AWS Organizations API Operationen, die den Mitgliedern und Strukturen Ihrer Organisation Einblick gewähren.
- Alle Dashboards auf Organisationsebene, die vom delegierten Administrator erstellt wurden. Durch Abmelden eines delegierten Administrators wird die Aggregation neuer Speichermetriken durch alle Dashboards auf Organisationsebene, die von diesem delegierten Administrator erstellt wurden, automatisch beendet.

> **Note**
>
> Der delegierte Administrator, dessen Registrierung aufgehoben wurde, kann weiterhin die historischen Daten für die deaktivierten Dashboards sehen, die er erstellt hat, sofern noch Daten für Abfragen verfügbar sind.

Verwenden der S3-Konsole

So melden Sie delegierte Administratoren für S3 Storage Lens ab

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Navigieren Sie im linken Navigationsbereich zu Storage Lens.
3. Wählen Sie AWS Organizations Einstellungen aus.
4. Wählen Sie unter Delegierte Administratoren das Konto aus, dessen Registrierung Sie aufheben möchten.
5. Wählen Sie Konto abmelden. Das abgemeldete Konto ist kein delegierter Administrator mehr und kann jetzt keine Dashboards auf Organisationsebene für alle Konten und Speicher in Ihrer Organisation erstellen.
6. Wählen Sie Konto registrieren.

Mit dem AWS CLI

Example

Das folgende Beispiel zeigt Ihnen, wie Sie delegierte Administratoren von Organizations für S3 Storage Lens mit dem abmelden. AWS CLI Um dieses Beispiel zu verwenden, ersetzen Sie es *111122223333* durch Ihre eigene ID. AWS-Konto

```
aws organizations deregister-delegated-administrator --service-principal storage-
lens.s3.amazonaws.com --account-id 111122223333
```

Verwendung des AWS SDK for Java

Example — Delegierte Administratoren von Organizations für S3 Storage Lens abmelden

Das folgende Beispiel zeigt Ihnen, wie Sie delegierte Administratoren von Organizations für S3 Storage Lens mithilfe von SDK Java abmelden. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.organizations.AWSOrganizations;
```

```
import com.amazonaws.services.organizations.AWSOrganizationsClient;
import
 com.amazonaws.services.organizations.model.DeregisterDelegatedAdministratorRequest;

public class DeregisterOrganizationsDelegatedAdministrator {
 private static final String S3_STORAGE_LENS_SERVICE_PRINCIPAL = "storage-
lens.s3.amazonaws.com";

 public static void main(String[] args) {
  try {
            String delegatedAdminAccountId = "111122223333"; // Account Id for the
 delegated administrator.
            AWSOrganizations organizationsClient = AWSOrganizationsClient.builder()
                .withCredentials(new ProfileCredentialsProvider())
                .withRegion(Regions.US_EAST_1)
                .build();

            organizationsClient.deregisterDelegatedAdministrator(new
 DeregisterDelegatedAdministratorRequest()
                .withAccountId(delegatedAdminAccountId)
                .withServicePrincipal(S3_STORAGE_LENS_SERVICE_PRINCIPAL));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but AWS Organizations couldn't
 process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // AWS Organizations couldn't be contacted for a response, or the client
            // couldn't parse the response from AWS Organizations.
            e.printStackTrace();
        }
 }
}
```

Arbeiten mit S3 Storage Lens-Gruppen zum Filtern und Aggregieren von Metriken

Eine Amazon-S3-Storage-Lens-Gruppe aggregiert Metriken mithilfe benutzerdefinierter Filter auf der Grundlage von Objektmetadaten. Mit Storage-Lens-Gruppen können Sie Details zu den Eigenschaften Ihrer Daten anzeigen, etwa die Objektverteilung nach Alter, die gängigsten Dateitypen usw. Sie können Metriken beispielsweise nach Objekt-Tag filtern, um die am schnellsten wachsenden Datensätze zu ermitteln, oder den Speicher auf Grundlage der Objektgröße und des Objektalters

visualisieren, um die Speicherarchivierungsstrategie festzulegen. Folglich vermitteln Amazon-S3-Storage-Lens-Gruppen Ihnen ein besseres Verständnis des S3-Speichers und unterstützen Sie bei seiner Optimierung.

Wenn Sie Storage-Lens-Gruppen verwenden, können Sie S3-Storage-Lens-Metriken anhand von Objektmetadaten wie Präfixen, Suffixen, Objekt-Tags, Objektgröße oder Objektalter analysieren und filtern. Diese Filter lassen sich auch kombinieren. Nachdem Sie eine Storage-Lens-Gruppe an das S3-Storage-Lens-Dashboard angehängt haben, können Sie die nach Amazon-S3-Storage-Lens-Gruppen aggregierten S3-Storage-Lens-Metriken direkt im Dashboard anzeigen.

Filtern Sie Messwerte beispielsweise auch nach Objektgröße oder Altersklassen, um festzustellen, welcher Speicheranteil aus kleinen Objekten besteht. Sie können diese Informationen dann mit S3 Intelligent-Tiering oder im S3-Lebenszyklus verwenden, um kleine Objekte zur Kosten- und Speicheroptimierung in verschiedene Speicherklassen zu überführen.

Themen

- Funktionsweise von S3-Storage-Lens-Gruppen
- Verwenden von Storage-Lens-Gruppen

Funktionsweise von S3-Storage-Lens-Gruppen

Mit Storage-Lens-Gruppen lassen sich Metriken mithilfe benutzerdefinierter Filter auf der Grundlage von Objektmetadaten aggregieren. Wenn Sie einen benutzerdefinierten Filter definieren, können Sie Präfixe, Suffixe, Objekt-Tags, Objektgrößen, Objektalter oder eine Kombination dieser benutzerdefinierten Filter verwenden. Bei der Erstellung einer Storage-Lens-Gruppe können Sie zudem einen einzelnen Filter oder mehrere Filterbedingungen angeben. Zur Angabe mehrerer Filterbedingungen verwenden Sie den logischen Operator `And` oder `Or`.

Wenn Sie eine Storage-Lens-Gruppe erstellen und konfigurieren, dient die Storage-Lens-Gruppe selbst als benutzerdefinierter Filter im Dashboard, an das Sie die Gruppe anhängen. Im Dashboard können Sie dann den Storage-Lens-Gruppenfilter verwenden, um Speichermetriken auf Grundlage des benutzerdefinierten Filters abzurufen, den Sie in der Gruppe definiert haben.

Um die Daten für die Storage-Lens-Gruppe im S3-Storage-Lens-Dashboard anzuzeigen, müssen Sie die Gruppe nach der Erstellung an das Dashboard anhängen. Nachdem die Storage-Lens-Gruppe an das Storage-Lens-Dashboard angehängt wurde, erfasst das Dashboard innerhalb von 48 Stunden Metriken zur Speichernutzung. Sie können diese Daten dann im Storage-Lens-Dashboard visualisieren oder sie über einen Metrikexport exportieren. Falls Sie vergessen, eine Storage-Lens-

Gruppe an ein Dashboard anzuhängen, werden die Daten der Storage-Lens-Gruppe weder erfasst noch angezeigt.

> **Note**
>
> - Wenn Sie eine S3 Storage Lens-Gruppe erstellen, erstellen Sie eine AWS Ressource. Daher hat jede Storage Lens-Gruppe ihren eigenen Amazon-Ressourcennamen (ARN), den Sie angeben können, wenn Sie sie an ein S3 Storage Lens-Dashboard anhängen oder daraus ausschließen.
> - Wenn eine Storage-Lens-Gruppe nicht an ein Dashboard angehängt ist, fallen für Sie keine zusätzlichen Gebühren für die Erstellung einer Storage-Lens-Gruppe an.
> - S3 Storage Lens aggregiert Nutzungsmetriken für ein Objekt aus allen passenden Storage-Lens-Gruppen. Wenn also ein Objekt den Filterbedingungen für zwei oder mehr Storage-Lens-Gruppen entspricht, werden Sie in Ihrer Speichernutzung wiederholte Angaben für dasselbe Objekt sehen.

Sie können eine Storage Lens-Gruppe auf Kontoebene in einer bestimmten Heimatregion (aus der Liste der unterstützten AWS-Regionen) erstellen. Anschließend können Sie die Storage-Lens-Gruppe mehreren Storage-Lens-Dashboards zuordnen, sofern sich die Dashboards in demselben AWS-Konto und in derselben Heimatregion befinden. Sie sind in der Lage, pro Heimatregion in jedem AWS-Konto bis zu 50 Storage-Lens-Gruppen zu erstellen.

Sie können S3 Storage Lens-Gruppen mithilfe der Amazon S3-Konsole AWS Command Line Interface (AWS CLI) oder Amazon S3 erstellen und verwalten RESTAPI. AWS SDKs

Themen

- Anzeigen aggregierter Metriken von Storage-Lens-Gruppen
- Berechtigungen für Storage-Lens-Gruppen
- Konfiguration von Storage-Lens-Gruppen
- AWS Ressourcen-Tags
- Metrikexport für Storage-Lens-Gruppen

Anzeigen aggregierter Metriken von Storage-Lens-Gruppen

Sie können die aggregierten Metriken für Storage-Lens-Gruppen anzeigen, indem Sie die Gruppen an ein Dashboard anhängen. Die anzuhängenden Storage-Lens-Gruppen müssen sich in der angegebenen Heimatregion des Dashboard-Kontos befinden.

Um eine Storage-Lens-Gruppe an ein Dashboard anzuhängen, müssen Sie die Gruppe im Abschnitt Aggregation von Storage-Lens-Gruppen der Dashboard-Konfiguration angeben. Sind mehrere Storage-Lens-Gruppen vorhanden, können Sie die Ergebnisse der Aggregation von Storage-Lens-Gruppen filtern und bestimmte Gruppen ein- oder ausschließen. Weitere Informationen zum Anhängen von Gruppen an Dashboards finden Sie unter the section called “Anhängen oder Entfernen einer Storage-Lens-Gruppe”.

Nachdem Sie die Gruppen angehängt haben, erhalten Sie innerhalb von 48 Stunden die zusätzlichen Aggregationsdaten der Storage-Lens-Gruppen im Dashboard.

> **Note**
>
> Wenn Sie aggregierte Metriken für die Storage-Lens-Gruppe einsehen möchten, müssen Sie die Gruppe an das S3-Storage-Lens-Dashboard anhängen.

Berechtigungen für Storage-Lens-Gruppen

Storage Lens-Gruppen benötigen bestimmte Berechtigungen in AWS Identity and Access Management (IAM), um den Zugriff auf S3 Storage Lens-Gruppenaktionen zu autorisieren. Um diese Berechtigungen zu gewähren, können Sie eine IAM identitätsbasierte Richtlinie verwenden. Sie können diese Richtlinie IAM Benutzern, Gruppen oder Rollen zuordnen, um ihnen Berechtigungen zu erteilen. Solche Berechtigungen können die Fähigkeit beinhalten, Storage-Lens-Gruppen zu erstellen oder zu löschen, ihre Konfigurationen einzusehen oder ihre Tags zu verwalten.

Der IAM Benutzer oder die Rolle, dem/der Sie Berechtigungen gewähren, muss zu dem Konto gehören, das die Storage Lens-Gruppe erstellt hat oder ihr gehört.

Zur Nutzung von Storage-Lens-Gruppen und zur Anzeige der Metriken von Storage-Lens-Gruppen benötigen Sie zunächst die entsprechenden Berechtigungen für die Verwendung von S3 Storage Lens. Weitere Informationen finden Sie unter the section called “Festlegung von -Berechtigungen”.

Um S3 Storage Lens-Gruppen zu erstellen und zu verwalten, benötigen Sie je nachdem, welche Aktionen Sie ausführen möchten, die folgenden IAM Berechtigungen:

Aktion	IAMBerechtigungen
Erstellen einer neuen Storage-Lens-Gruppe	`s3:CreateStorageLensGroup`
Erstellen einer neuen Storage-Lens-Gruppe	`s3:CreateStorageLensGroup` , `s3:TagResource`
Aktualisieren einer vorhandenen Storage-Lens-Gruppe	`s3:UpdateStorageLensGroup`
Zurückgeben der Konfigurationsdetails einer Storage-Lens-Gruppe	`s3:GetStorageLensGroup`
Auflisten aller Storage-Lens-Gruppen in der Heimatregion	`s3:ListStorageLensGroups`
Löschen einer Storage-Lens-Gruppe	`s3:DeleteStorageLensGroup`
Auflisten der Tags, die der Storage-Lens-Gruppe hinzugefügt wurden	`s3:ListTagsForResource`
Hinzufügen oder Aktualisieren eines Storage-Lens-Gruppen-Tags für eine vorhandene Storage-Lens-Gruppe	`s3:TagResource`
Löschen eines Tags aus einer Storage-Lens-Gruppe	`s3:UntagResource`

Hier ist ein Beispiel dafür, wie Sie Ihre IAM Richtlinie in dem Konto konfigurieren, das die Storage Lens-Gruppe erstellt. Zur Verwendung dieser Richtlinie ersetzen Sie *`us-east-1`* durch die Heimatregion, in der sich die Storage-Lens-Gruppe befindet. Ersetzen Sie *`111122223333`* durch Ihre AWS-Konto -ID und *`example-storage-lens-group`* durch den Namen der Storage-Lens-Gruppe. Wenn Sie diese Berechtigungen auf alle Storage-Lens-Gruppen anwenden möchten, ersetzen Sie *`example-storage-lens-group`* durch ein *`*`*.

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

```
            "Sid": "EXAMPLE-Statement-ID",
            "Effect": "Allow",
            "Action": [
                "s3:CreateStorageLensGroup",
                "s3:UpdateStorageLensGroup",
                "s3:GetStorageLensGroup",
                "s3:ListStorageLensGroups",
                "s3:DeleteStorageLensGroup,
                "s3:TagResource",
                "s3:UntagResource",
                "s3:ListTagsForResource"
                ],
            "Resource": "arn:aws:s3:us-east-1:111122223333:storage-lens-group/example-
storage-lens-group"
        }
    ]
}
```

Weitere Informationen zur Verwendung von S3-Storage-Lens-Berechtigungen finden Sie unter Amazon S3 Storage Lens-Berechtigungen einrichten. Weitere Informationen zur Sprache der IAM Richtlinie finden Sie unterRichtlinien und Berechtigungen in Amazon S3.

Konfiguration von Storage-Lens-Gruppen

Namen von S3-Storage-Lens-Gruppen

Wir empfehlen, Storage-Lens-Gruppen Namen zu geben, die auf ihren Zweck hinweisen. Auf diese Weise können Sie leicht feststellen, an welche Gruppen Dashboards angehängt werden sollen. Um eine Storage-Lens-Gruppe an ein Dashboard anzuhängen, müssen Sie die Gruppe im Abschnitt Aggregation von Storage-Lens-Gruppen der Dashboard-Konfiguration angeben.

Namen von Storage-Lens-Gruppen müssen innerhalb des Kontos eindeutig sein. Sie dürfen nicht länger als 64 Zeichen sein. Nur Buchstaben (a–z, A–Z), Zahlen (0–9), Bindestriche (-) und Unterstriche (_) sind zulässig.

Heimatregion

In der Heimatregion wird AWS-Region Ihre Storage Lens-Gruppe erstellt und verwaltet. Eine Storage-Lens-Gruppe wird in derselben Heimatregion wie das Amazon-S3-Storage-Lens-Dashboard erstellt. Die Konfiguration und die Metriken der Storage-Lens-Gruppe werden ebenfalls in dieser Region gespeichert. Sie sind in der Lage, in einer Heimatregion bis zu 50 Storage-Lens-Gruppen zu erstellen.

Nachdem Sie eine Storage-Lens-Gruppe erstellt haben, können Sie die Heimatregion nicht bearbeiten.

Scope

Um Objekte in eine Storage-Lens-Gruppe aufzunehmen, müssen sie sich im Umfang des Amazon-S3-Storage-Lens-Dashboards befinden. Der Umfang des Storage-Lens-Dashboards wird durch die Buckets bestimmt, die Sie in den Dashboard-Umfang der S3-Storage-Lens-Dashboard-Konfiguration aufgenommen haben.

Sie können verschiedene Filter für Ihre Objekte verwenden, um den Umfang einer Storage-Lens-Gruppe zu definieren. Um diese Storage-Lens-Gruppenmetriken im S3-Storage-Lens-Dashboard anzuzeigen, müssen Objekte den Filtern entsprechen, die Sie in die Storage-Lens-Gruppen aufnehmen. Angenommen, in einer Storage-Lens-Gruppe werden Objekte mit dem Präfix `marketing` und dem Suffix `.png` berücksichtigt, aber keine Objekte entsprechen diesen Kriterien. In diesem Fall werden in dem täglichen Metrikexport keine Metriken für diese Storage-Lens-Gruppe generiert und im Dashboard sind keine Metriken für diese Gruppe sichtbar.

Filter

Sie können die folgenden Filter in einer S3-Storage-Lens-Gruppe verwenden:

- Präfixe – Gibt das Präfix der berücksichtigten Objekte an. Es handelt sich um eine Zeichenfolge am Anfang des Objektschlüsselnamens. Der Wert `images` für den Filter Präfixe schließt beispielsweise Objekte mit einem der folgenden Präfixe ein:`images/`, `images-marketing` und `images/production`. Die maximale Länge eines Präfixes beträgt 512 Byte.
- Suffixe – Gibt das Suffix der berücksichtigten Objekte an (z. B. `.png`, `.jpeg` oder `.csv`). Die maximale Länge eines Suffixes beträgt 1 024 Byte.
- Objekt-Tags – Gibt die Liste der Objekt-Tags an, nach denen gefiltert werden soll. Ein Tag-Schlüssel darf nicht mehr als 128 Unicode-Zeichen beinhalten, ein Tag-Wert maximal 256 Unicode-Zeichen. Beachten Sie, dass, wenn das Feld für den Objekt-Tag-Wert leer gelassen wird, die S3 Storage Lens-Gruppen das Objekt nur anderen Objekten zuordnen, die ebenfalls leere Tag-Werte haben.
- Alter – Gibt den Altersbereich der enthaltenen Objekte in Tagen an. Es werden nur Ganzzahlen unterstützt.
- Größe – Gibt den Objektgrößenbereich der berücksichtigten Objekte in Byte an. Es werden nur Ganzzahlen unterstützt. Der maximal zulässige Wert ist 5 TB.

Objekt-Tags von Storage-Lens-Gruppen

Sie können eine Storage-Lens-Gruppe erstellen, die bis zu 10 Objekt-Tag-Filter enthält. Das folgende Beispiel enthält zwei Schlüssel-Wert-Paare für Objekt-Tags als Filter für eine Storage-Lens-Gruppe namens *Marketing-Department*. Zur Verwendung dieses Beispiels ersetzen Sie *Marketing-Department* durch den Namen Ihrer Gruppe und *object-tag-key-1*, *object-tag-value-1* usw. durch die Schlüssel-Wert-Paare der Objekt-Tags, nach denen Sie filtern möchten.

```
{
    "Name": "Marketing-Department",
    "Filter": {
     "MatchAnyTag":[
                {
                    "Key": "object-tag-key-1",
                    "Value": "object-tag-value-1"
                },
                {
                    "Key": "object-tag-key-2",
                    "Value": "object-tag-value-2"
                }
            ]
    }
}
```

Logische Operatoren (**And** oder **Or**)

Um mehrere Filterbedingungen in eine Storage-Lens-Gruppe aufzunehmen, können Sie logische Operatoren (entweder `And` oder `Or`) einsetzen. Im folgenden Beispiel besitzt die Storage-Lens-Gruppe *Marketing-Department* einen `And`-Operator, der die Filter `ObjectSize`, `Prefix` und `ObjectAge` enthält. Aufgrund des `And`-Operators werden nur Objekte, die allen Filterbedingungen entsprechen, in den Umfang der Storage-Lens-Gruppe aufgenommen.

Zur Verwendung dieses Beispiels ersetzen Sie die *user input placeholders* durch die Werte, nach denen Sie filtern möchten.

```
{
    "Name": "Marketing-Department",
    "Filter": {
        "And": {
            "MatchAnyPrefix": [
                "prefix-1",
                "prefix-2",
```

```
                "prefix-3/sub-prefix-1"
            ],
             "MatchObjectAge": {
                "DaysGreaterThan": 10,
                "DaysLessThan": 60
            },
            "MatchObjectSize": {
                "BytesGreaterThan": 10,
                "BytesLessThan": 60
            }
        }
    }
}
```

> **Note**
>
> Wenn Sie Objekte berücksichtigen möchten, die nur einer Filterbedingung entsprechen, ersetzen Sie in diesem Beispiel den logischen Operator `And` durch den logischen Operator `Or`.

AWS Ressourcen-Tags

Jede S3 Storage Lens-Gruppe wird als AWS Ressource mit eigenem Amazon-Ressourcennamen (ARN) gezählt. Daher können Sie bei der Konfiguration einer Storage-Lens-Gruppe optional AWS -Ressourcen-Tags zur Gruppe hinzufügen. Es ist möglich, für jede Storage-Lens-Gruppe bis zu 50 Tags hinzuzufügen. Zum Erstellen einer Storage-Lens-Gruppe mit Tags benötigen Sie die Berechtigungen `s3:TagResource` und `s3:CreateStorageLensGroup`.

Sie können AWS Ressourcen-Tags verwenden, um Ressourcen nach Abteilung, Geschäftsbereich oder Projekt zu kategorisieren. Dies ist sinnvoll, wenn viele Ressourcen desselben Typs vorliegen. Durch die Anwendung von Tags können Sie eine bestimmte Storage-Lens-Gruppe auf Grundlage der ihr zugewiesenen Tags schnell identifizieren. Sie können Tags auch verwenden, um Ihre Kosten zu kategorisieren und zu verfolgen.

Wenn Sie Ihrer Storage Lens-Gruppe ein AWS Ressourcen-Tag hinzufügen, aktivieren Sie außerdem die attributbasierte Zugriffskontrolle (). ABAC ABACist eine Autorisierungsstrategie, die Berechtigungen auf der Grundlage von Attributen, in diesem Fall Tags, definiert. Sie können in Ihren IAM Richtlinien auch Bedingungen verwenden, die Ressourcen-Tags angeben, um den Zugriff auf AWS Ressourcen zu kontrollieren.

Sie können Tag (Markierung)-Schlüssel und -Werte bearbeiten und Tags (Markierungen) jederzeit von einer Ressource entfernen. Beachten Sie folgende Einschränkungen:

- Bei Tag-Schlüsseln und Tag-Werten muss die Groß- und Kleinschreibung beachtet werden.
- Wenn Sie ein Tag (Markierung) mit demselben Schlüssel wie ein vorhandener Tag (Markierung) für die Ressource hinzufügen, wird der alte Wert mit dem neuen überschrieben.
- Wenn Sie eine Ressource löschen, werden alle Tags (Markierungen) der Ressource ebenfalls gelöscht.
- Nehmen Sie keine privaten oder vertraulichen Daten in Ihre AWS Ressourcen-Tags auf.
- Systemtags (oder Tags mit Tag-Schlüsseln, die mit `aws:` beginnen) werden nicht unterstützt.
- Die einzelnen Tag-Schlüssel dürfen maximal 128 Zeichen lang sein. Die einzelnen Tag-Werte dürfen maximal 256 Zeichen lang sein.

Metrikexport für Storage-Lens-Gruppen

Die Metriken einer S3-Storage-Lens-Gruppe sind im Amazon-S3-Storage-Lens-Metrikexport für das Dashboard enthalten, an das die Storage-Lens-Gruppe angehängt wird. Allgemeine Informationen zur Exportfunktion für Storage-Lens-Metriken finden Sie unter Anzeigen von Amazon S3-Storage-Lens-Metriken mit einem Datenexport.

Der Metrikexport für Storage-Lens-Gruppen umfasst alle S3-Storage-Lens-Metriken, die im Umfang für das Dashboard enthalten sind, an das die Storage-Lens-Gruppe angehängt wurde. Der Export beinhaltet zudem zusätzliche Metrikdaten für Storage-Lens-Gruppen.

Nach Erstellung der Storage-Lens-Gruppe wird der Metrikexport täglich an den Bucket gesendet, den Sie bei der Konfiguration des Metrikexports für das Dashboard ausgewählt haben, an das die Gruppe angehängt wird. Es kann bis zu 48 Stunden dauern, bis Sie den ersten Metrikexport erhalten.

Damit Metriken im täglichen Export generiert werden, müssen Objekte den Filtern entsprechen, die Sie in die Storage-Lens-Gruppen aufnehmen. Wenn den Filtern, die Sie in die Storage-Lens-Gruppe eingebunden haben, keine Objekte entsprechen, werden keine Metriken generiert. Entspricht ein Objekt jedoch zwei oder mehr Storage-Lens-Gruppen, wird das Objekt im Metrikexport für jede Gruppe separat aufgeführt.

Sie können Metriken für Storage-Lens-Gruppen ermitteln, indem Sie in der Spalte `record_type` des Metrikexports für das Dashboard nach einem der folgenden Werte suchen:

- `STORAGE_LENS_GROUP_BUCKET`

- `STORAGE_LENS_GROUP_ACCOUNT`

In der `record_value` Spalte wird die Ressource ARN für die Storage Lens-Gruppe angezeigt (z. B.`arn:aws:s3:us-east-1:111122223333:storage-lens-group/Marketing-Department`).

Verwenden von Storage-Lens-Gruppen

Amazon-S3-Storage-Lens-Gruppen aggregieren Metriken mithilfe benutzerdefinierter Filter auf der Grundlage von Objektmetadaten. Sie können S3-Storage-Lens-Metriken anhand der Präfixe, der Suffixe, der Objekt-Tags, der Objektgröße oder des Objektalters analysieren und filtern. Zudem bieten Amazon-S3-Storage-Lens-Gruppen die Möglichkeit, die Nutzung innerhalb und zwischen Amazon-S3-Buckets zu kategorisieren. Auf diese Weise erhalten Sie mehr Informationen zum S3-Speicher und können ihn besser optimieren.

Für die Visualisierung der Daten für eine Storage-Lens-Gruppe müssen Sie zunächst Ihre Storage-Lens-Gruppe an ein S3-Storage-Lens-Dashboard anhängen. Wenn Sie Storage-Lens-Gruppen im Dashboard verwalten müssen, können Sie die Dashboard-Konfiguration bearbeiten. Um festzustellen, welche Storage-Lens-Gruppen sich in Ihrem Konto befinden, können Sie sie auflisten. Auf der Registerkarte Storage-Lens-Gruppen im Dashboard können Sie jederzeit überprüfen, welche Storage-Lens-Gruppen an das Dashboard angefügt sind. Wenn Sie den Umfang einer vorhandenen Storage-Lens-Gruppe überprüfen oder aktualisieren möchten, können Sie deren Details einsehen. Storage-Lens-Gruppen können auch dauerhaft gelöscht werden.

Um Berechtigungen zu verwalten, können Sie benutzerdefinierte AWS Ressourcen-Tags erstellen und zu Ihren Storage Lens-Gruppen hinzufügen. Mithilfe von AWS Ressourcen-Tags können Sie Ressourcen nach Abteilung, Geschäftsbereich oder Projekt kategorisieren. Dies ist sinnvoll, wenn viele Ressourcen desselben Typs vorliegen. Durch die Anwendung von Tags können Sie eine bestimmte Storage-Lens-Gruppe auf Grundlage der ihr zugewiesenen Tags schnell identifizieren.

Wenn Sie Ihrer Storage Lens-Gruppe ein AWS Ressourcen-Tag hinzufügen, aktivieren Sie außerdem die attributbasierte Zugriffskontrolle (). ABAC ABACist eine Autorisierungsstrategie, die Berechtigungen auf der Grundlage von Attributen, in diesem Fall Tags, definiert. Sie können in Ihren IAM Richtlinien auch Bedingungen verwenden, die Ressourcen-Tags angeben, um den Zugriff auf AWS Ressourcen zu kontrollieren.

Themen

- Erstellen einer Storage-Lens-Gruppe

- Anhängen von S3-Storage-Lens-Gruppen an das Dashboard und Entfernen der Gruppen
- Visualisieren von Storage Lens-Gruppendaten
- Aktualisieren einer Storage-Lens-Gruppe
- Verwaltung von AWS Ressourcen-Tags mit Storage Lens-Gruppen
- Auflisten aller Storage-Lens-Gruppen
- Anzeigen von Details zu Storage-Lens-Gruppen
- Löschen einer Storage-Lens-Gruppe

Erstellen einer Storage-Lens-Gruppe

Die folgenden Beispiele zeigen, wie Sie eine Amazon S3 Storage Lens-Gruppe mithilfe der Amazon S3 S3-Konsole, AWS Command Line Interface (AWS CLI) und erstellen AWS SDK for Java.

Verwenden der S3-Konsole

So erstellen Sie eine Storage-Lens-Gruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Navigationsleiste oben auf der Seite den Namen der aktuell angezeigten AWS Region aus. Wählen Sie als Nächstes die Region aus, zu der Sie wechseln möchten.
3. Wählen Sie im linken Navigationsbereich die Option Storage-Lens-Gruppen aus.
4. Wählen Sie Storage-Lens-Gruppe erstellen aus.
5. Sehen Sie sich unter Allgemein Ihre Heimatregion an und geben Sie den Namen Ihrer Storage Lens-Gruppe ein.
6. Wählen Sie unter Umfang den Filter aus, der auf die Storage-Lens-Gruppe angewendet werden soll. Um mehrere Filter anzuwenden, wählen Sie Ihre Filter und dann den ANDlogischen Operator OR aus.

    - Wählen Sie für den Filter Präfixe die Option Präfixe aus und geben Sie eine Präfixzeichenfolge ein. Zum Hinzufügen mehrerer Präfixe wählen Sie Präfix hinzuzufügen aus. Zum Entfernen eines Präfixes wählen Sie neben dem zu entfernenden Präfix Entfernen aus.
    - Wählen Sie für den Filter Objekt-Tags die Option Objekt-Tags aus und geben Sie das Schlüssel-Wert-Paar für das Objekt ein. Wählen Sie dann Tag hinzufügen aus. Zum Entfernen eines vorhandenen Tags wählen Sie neben dem zu entfernenden Tag Entfernen aus.

- Wählen Sie für den Filter Suffixe die Option Suffixe aus und geben Sie eine Suffixzeichenfolge ein. Um mehrere Suffixe hinzuzufügen, wählen Sie Suffix hinzufügen aus. Zum Entfernen eines Suffixes wählen Sie neben dem zu entfernenden Suffix Entfernen aus.
- Geben Sie für den Filter Alter den Altersbereich des Objekts in Tagen an. Wählen Sie Mindestobjektalter festlegen aus und geben Sie das Mindestalter des Objekts ein. Wählen Sie dann Das maximale Objektalter festlegen aus und geben Sie das maximale Objektalter ein.
- Geben Sie für den Filter Größe den Objektgrößenbereich und die Maßeinheit an. Wählen Sie Mindestobjektgröße festlegen aus und geben Sie die Mindestobjektgröße ein. Wählen Sie Die maximale Objektgröße festlegen aus und geben Sie die maximale Objektgröße ein.

7. (Optional) Fügen Sie für AWS Ressourcen-Tags das Schlüssel-Wert-Paar hinzu und wählen Sie dann Tag hinzufügen aus.
8. Wählen Sie Storage-Lens-Gruppe erstellen aus.

Verwenden Sie AWS CLI

Mit dem folgenden AWS CLI Beispielbefehl wird eine Storage Lens-Gruppe erstellt. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control create-storage-lens-group --account-id 111122223333 \
--region us-east-1 --storage-lens-group=file://./marketing-department.json
```

Der folgende AWS CLI Beispielbefehl erstellt eine Storage Lens-Gruppe mit zwei AWS Ressourcen-Tags. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control create-storage-lens-group --account-id 111122223333 \
--region us-east-1 --storage-lens-group=file://./marketing-department.json \
--tags Key=k1,Value=v1 Key=k2,Value=v2
```

JSONBeispielkonfigurationen finden Sie unterKonfiguration von Storage-Lens-Gruppen.

Verwendung des AWS SDK for Java

Im folgenden AWS SDK for Java Beispiel wird eine Storage Lens-Gruppe erstellt. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

Example – Erstellen einer Storage-Lens-Gruppe mit einem einzigen Filter

In dem folgenden Beispiel wird eine Storage-Lens-Gruppe namens *Marketing-Department* erstellt. Diese Gruppe besitzt einen Filter für das Objektalter, der einen Altersbereich von *30* bis *90* Tagen festlegt. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.MatchObjectAge;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;

public class CreateStorageLensGroupWithObjectAge {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            StorageLensGroupFilter objectAgeFilter = StorageLensGroupFilter.builder()
                    .matchObjectAge(MatchObjectAge.builder()
                            .daysGreaterThan(30)
                            .daysLessThan(90)
                            .build())
                    .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                    .name(storageLensGroupName)
                    .filter(objectAgeFilter)
                    .build();

            CreateStorageLensGroupRequest createStorageLensGroupRequest =
 CreateStorageLensGroupRequest.builder()
                    .storageLensGroup(storageLensGroup)
                    .accountId(accountId).build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
```

```
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example — Erstellen einer Storage-Lens-Gruppe mit einem **AND**-Operator, der mehrere Filter umfasst

In dem folgenden Beispiel wird eine Storage-Lens-Gruppe namens *Marketing-Department* erstellt. Diese Gruppe verwendet den AND-Operator, um anzugeben, dass Objekte allen Filterbedingungen entsprechen müssen. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.MatchObjectAge;
import software.amazon.awssdk.services.s3control.model.MatchObjectSize;
import software.amazon.awssdk.services.s3control.model.S3Tag;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupAndOperator;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;


public class CreateStorageLensGroupWithAndFilter {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
```

```
        String accountId = "111122223333";

        try {
            // Create object tags.
            S3Tag tag1 = S3Tag.builder()
                    .key("object-tag-key-1")
                    .value("object-tag-value-1")
                    .build();
            S3Tag tag2 = S3Tag.builder()
                    .key("object-tag-key-2")
                    .value("object-tag-value-2")
                    .build();

            StorageLensGroupAndOperator andOperator =
StorageLensGroupAndOperator.builder()
                    .matchAnyPrefix("prefix-1", "prefix-2", "prefix-3/sub-prefix-1")
                    .matchAnySuffix(".png", ".gif", ".jpg")
                    .matchAnyTag(tag1, tag2)
                    .matchObjectAge(MatchObjectAge.builder()
                            .daysGreaterThan(30)
                            .daysLessThan(90).build())
                    .matchObjectSize(MatchObjectSize.builder()
                            .bytesGreaterThan(1000L)
                            .bytesLessThan(6000L).build())
                    .build();

            StorageLensGroupFilter andFilter = StorageLensGroupFilter.builder()
                    .and(andOperator)
                    .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                    .name(storageLensGroupName)
                    .filter(andFilter)
                    .build();

            CreateStorageLensGroupRequest createStorageLensGroupRequest =
CreateStorageLensGroupRequest.builder()
                    .storageLensGroup(storageLensGroup)
                    .accountId(accountId).build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
```

```
            s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example — Erstellen einer Storage-Lens-Gruppe mit einem **OR**-Operator, der mehrere Filter umfasst

In dem folgenden Beispiel wird eine Storage-Lens-Gruppe namens *Marketing-Department* erstellt. Für diese Gruppe wird ein OR-Operator verwendet, um einen Präfixfilter (*prefix-1*, *prefix-2*, *prefix3/sub-prefix-1*) oder einen Objektgrößenfilter mit einem Größenbereich zwischen *1000* Byte und *6000* Byte anzuwenden. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.MatchObjectSize;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupOrOperator;

public class CreateStorageLensGroupWithOrFilter {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            StorageLensGroupOrOperator orOperator =
 StorageLensGroupOrOperator.builder()
                    .matchAnyPrefix("prefix-1", "prefix-2", "prefix-3/sub-prefix-1")
```

```
                    .matchObjectSize(MatchObjectSize.builder()
                            .bytesGreaterThan(1000L)
                            .bytesLessThan(6000L)
                            .build())
                    .build();

            StorageLensGroupFilter orFilter = StorageLensGroupFilter.builder()
                    .or(orOperator)
                    .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                    .name(storageLensGroupName)
                    .filter(orFilter)
                    .build();

            CreateStorageLensGroupRequest createStorageLensGroupRequest =
 CreateStorageLensGroupRequest.builder()
                    .storageLensGroup(storageLensGroup)
                    .accountId(accountId).build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example — Erstellen Sie eine Storage Lens-Gruppe mit einem einzigen Filter und zwei AWS Ressourcen-Tags

In dem folgenden Beispiel wird eine Storage-Lens-Gruppe namens *Marketing-Department* erstellt, die einen Suffixfilter aufweist. In diesem Beispiel werden der Storage Lens-Gruppe außerdem

zwei AWS Ressourcen-Tags hinzugefügt. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.CreateStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;
import software.amazon.awssdk.services.s3control.model.Tag;

public class CreateStorageLensGroupWithResourceTags {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            // Create AWS resource tags.
            Tag resourceTag1 = Tag.builder()
                    .key("resource-tag-key-1")
                    .value("resource-tag-value-1")
                    .build();
            Tag resourceTag2 = Tag.builder()
                    .key("resource-tag-key-2")
                    .value("resource-tag-value-2")
                    .build();

            StorageLensGroupFilter suffixFilter = StorageLensGroupFilter.builder()
                    .matchAnySuffix(".png", ".gif", ".jpg")
                    .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                    .name(storageLensGroupName)
                    .filter(suffixFilter)
                    .build();

            CreateStorageLensGroupRequest createStorageLensGroupRequest =
 CreateStorageLensGroupRequest.builder()
                    .storageLensGroup(storageLensGroup)
```

```
                    .tags(resourceTag1, resourceTag2)
                    .accountId(accountId).build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.createStorageLensGroup(createStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

JSONBeispielkonfigurationen finden Sie unterKonfiguration von Storage-Lens-Gruppen.

Anhängen von S3-Storage-Lens-Gruppen an das Dashboard und Entfernen der Gruppen

Nachdem Sie in Amazon S3 Storage Lens ein Upgrade auf das erweiterte Kontingent durchgeführt haben, können Sie eine Storage-Lens-Gruppe an das Dashboard anhängen. Sind mehrere Storage-Lens-Gruppen vorhanden, lassen sich die gewünschten Gruppen ein- oder ausschließen.

Die Storage-Lens-Gruppen müssen sich in der angegebenen Heimatregion im Dashboard-Konto befinden. Nachdem Sie eine Storage-Lens-Gruppe an das Dashboard angehängt haben, erhalten Sie innerhalb von 48 Stunden die Aggregationsdaten der zusätzlichen Storage-Lens-Gruppe im Metrikexport.

> **Note**
>
> Wenn Sie aggregierte Metriken für die Storage Lens-Gruppe einsehen möchten, müssen Sie sie an das Storage-Lens-Dashboard anhängen. Beispiele für JSON Konfigurationsdateien der Storage Lens-Gruppe finden Sie unterBeispielkonfiguration für S3 Storage Lens mit Storage Lens-Gruppen in JSON.

Verwenden der S3-Konsole

So hängen Sie eine Storage-Lens-Gruppe an ein Storage-Lens-Dashboard an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich unter Storage Lens die Option Dashboards aus.
3. Aktivieren Sie das Optionsfeld für das Storage-Lens-Dashboard, an das Sie eine Storage-Lens-Gruppe anhängen möchten.
4. Wählen Sie Edit (Bearbeiten) aus.
5. Unter Metrikauswahl, wählen Sie Fortschrittliche Metriken und Empfehlungen aus.
6. Wählen Sie Aggregation von Storage-Lens-Gruppen aus.

   > **Note**
   >
   > Standardmäßig ist die Option Erweiterte Metriken auch ausgewählt. Sie können diese Einstellung jedoch auch deaktivieren, da sie nicht erforderlich ist, um Daten von Storage-Lens-Gruppen zu aggregieren.

7. Scrollen Sie nach unten zu Aggregation von Storage-Lens-Gruppen und geben Sie die Storage-Lens-Gruppe(n) an, die Sie in der Datenaggregation berücksichtigen oder daraus ausschließen möchten. Sie können die folgenden Filteroptionen verwenden:

   - Wenn Sie bestimmte Storage-Lens-Gruppen berücksichtigen möchten, wählen Sie Storage-Lens-Gruppen einbeziehen aus. Wählen Sie unter Zu berücksichtigende Storage-Lens-Gruppen Ihre Storage-Lens-Gruppen aus.
   - Wenn Sie alle Storage-Lens-Gruppen berücksichtigen möchten, wählen Sie Alle Storage-Lens-Gruppen in der Heimatregion in diesem Konto berücksichtigen aus.
   - Möchten Sie bestimmte Storage-Lens-Gruppen ausschließen, wählen Sie Storage-Lens-Gruppen ausschließen aus. Wählen Sie unter Auszuschließende Storage-Lens-Gruppen die Storage-Lens-Gruppen aus, die Sie ausschließen möchten.

8. Wählen Sie Änderungen speichern. Nachdem Sie eine Storage-Lens-Gruppe an das Dashboard angehängt haben, erhalten Sie innerhalb von 48 Stunden die Aggregationsdaten der zusätzlichen Storage-Lens-Gruppe im Metrikexport.

So entfernen Sie eine Storage-Lens-Gruppe aus einem S3-Storage-Lens-Dashboard

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich unter Storage Lens die Option Dashboards aus.
3. Aktivieren Sie das Optionsfeld für das Storage-Lens-Dashboard, aus dem Sie eine Storage-Lens-Gruppe entfernen möchten.
4. Wählen Sie Dashboard-Konfigurationen anzeigen aus.
5. Wählen Sie Edit (Bearbeiten) aus.
6. Scrollen Sie nach unten bis zum Abschnitt Auswahl von Metriken.
7. Wählen Sie unter Aggregation von Storage-Lens-Gruppen das X neben der Storage-Lens-Gruppe aus, die Sie entfernen möchten. Dadurch wird die Storage-Lens-Gruppe entfernt.

   Wenn Sie alle Storage-Lens-Gruppen in das Dashboard aufgenommen haben, deaktivieren Sie das Kontrollkästchen neben Alle Storage-Lens-Gruppen in der Heimatregion in diesem Konto berücksichtigen.
8. Wählen Sie Änderungen speichern.

> **Note**
>
> Es dauert bis zu 48 Stunden, bis die Konfigurationsänderungen im Dashboard angezeigt werden.

Verwendung des AWS SDK for Java

Example — Ordnen Sie alle Storage Lens-Gruppen einem Dashboard zu

Im folgenden SDK Java-Beispiel werden alle Storage Lens-Gruppen im Konto angehängt *`111122223333`* an das *`DashBoardConfigurationId`* Armaturenbrett:

```
package aws.example.s3control;


import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
```

```
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensGroupLevel;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateDashboardWithStorageLensGroups {
    public static void main(String[] args) {
        String configurationId = "ExampleDashboardConfigurationId";
        String sourceAccountId = "111122223333";

        try {
            StorageLensGroupLevel storageLensGroupLevel = new StorageLensGroupLevel();

            AccountLevel accountLevel = new AccountLevel()
                    .withBucketLevel(new BucketLevel())
                    .withStorageLensGroupLevel(storageLensGroupLevel);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                    .withId(configurationId)
                    .withAccountLevel(accountLevel)
                    .withIsEnabled(true);

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfiguration(new
 PutStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withStorageLensConfiguration(configuration)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
```

```
            e.printStackTrace();
        }
    }
}
```

Example — Hängen Sie zwei Storage Lens-Gruppen an ein Dashboard an

Im folgenden AWS SDK for Java Beispiel werden zwei Storage Lens-Gruppen angehängt (*StorageLensGroupName1* and *StorageLensGroupName2*) an den *ExampleDashboardConfigurationId* Armaturenbrett.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensGroupLevel;
import com.amazonaws.services.s3control.model.StorageLensGroupLevelSelectionCriteria;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateDashboardWith2StorageLensGroups {
    public static void main(String[] args) {
        String configurationId = "ExampleDashboardConfigurationId";
        String storageLensGroupName1 = "StorageLensGroupName1";
        String storageLensGroupName2 = "StorageLensGroupName2";
        String sourceAccountId = "111122223333";

        try {
            StorageLensGroupLevelSelectionCriteria selectionCriteria = new
 StorageLensGroupLevelSelectionCriteria()
                    .withInclude(
                            "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
 + ":storage-lens-group/" + storageLensGroupName1,
                            "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
 + ":storage-lens-group/" + storageLensGroupName2);
```

```
            System.out.println(selectionCriteria);
            StorageLensGroupLevel storageLensGroupLevel = new StorageLensGroupLevel()
                    .withSelectionCriteria(selectionCriteria);

            AccountLevel accountLevel = new AccountLevel()
                    .withBucketLevel(new BucketLevel())
                    .withStorageLensGroupLevel(storageLensGroupLevel);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                    .withId(configurationId)
                    .withAccountLevel(accountLevel)
                    .withIsEnabled(true);

            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfiguration(new
 PutStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withStorageLensConfiguration(configuration)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Example — Ordnen Sie alle Storage Lens-Gruppen mit Ausschlüssen zu

Im folgenden Beispiel SDK für Java werden alle Storage Lens-Gruppen an das angehängt *ExampleDashboardConfigurationId* Dashboard, mit Ausnahme der beiden angegebenen (*StorageLensGroupName1* and *StorageLensGroupName2*):

```
package aws.example.s3control;
```

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;
import com.amazonaws.services.s3control.model.AccountLevel;
import com.amazonaws.services.s3control.model.BucketLevel;
import com.amazonaws.services.s3control.model.PutStorageLensConfigurationRequest;
import com.amazonaws.services.s3control.model.StorageLensConfiguration;
import com.amazonaws.services.s3control.model.StorageLensGroupLevel;
import com.amazonaws.services.s3control.model.StorageLensGroupLevelSelectionCriteria;

import static com.amazonaws.regions.Regions.US_WEST_2;

public class CreateDashboardWith2StorageLensGroupsExcluded {
    public static void main(String[] args) {
        String configurationId = "ExampleDashboardConfigurationId";
        String storageLensGroupName1 = "StorageLensGroupName1";
        String storageLensGroupName2 = "StorageLensGroupName2";
        String sourceAccountId = "111122223333";

        try {
            StorageLensGroupLevelSelectionCriteria selectionCriteria = new
 StorageLensGroupLevelSelectionCriteria()
                    .withInclude(
                            "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
 + ":storage-lens-group/" + storageLensGroupName1,
                            "arn:aws:s3:" + US_WEST_2.getName() + ":" + sourceAccountId
 + ":storage-lens-group/" + storageLensGroupName2);

            System.out.println(selectionCriteria);
            StorageLensGroupLevel storageLensGroupLevel = new StorageLensGroupLevel()
                    .withSelectionCriteria(selectionCriteria);

            AccountLevel accountLevel = new AccountLevel()
                    .withBucketLevel(new BucketLevel())
                    .withStorageLensGroupLevel(storageLensGroupLevel);

            StorageLensConfiguration configuration = new StorageLensConfiguration()
                    .withId(configurationId)
                    .withAccountLevel(accountLevel)
                    .withIsEnabled(true);
```

```
            AWSS3Control s3ControlClient = AWSS3ControlClient.builder()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(US_WEST_2)
                    .build();

            s3ControlClient.putStorageLensConfiguration(new
 PutStorageLensConfigurationRequest()
                    .withAccountId(sourceAccountId)
                    .withConfigId(configurationId)
                    .withStorageLensConfiguration(configuration)
            );
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Visualisieren von Storage Lens-Gruppendaten

Sie können Storage-Lens-Gruppendaten visualisieren, indem Sie die Gruppe an das Amazon-S3-Storage-Lens-Dashboard anhängen. Nachdem Sie die Storage-Lens-Gruppe in die Aggregation von Storage-Lens-Gruppen in der Dashboard-Konfiguration aufgenommen haben, kann es bis zu 48 Stunden dauern, bis die Daten der Storage-Lens-Gruppe im Dashboard angezeigt werden.

Nach Aktualisierung der Dashboard-Konfiguration werden alle neu hinzugefügten Storage-Lens-Gruppen in der Liste der verfügbaren Ressourcen auf der Registerkarte Storage-Lens-Gruppen angezeigt. Zudem können Sie die Speichernutzung auf der Registerkarte Übersicht weiter analysieren, indem Sie die Daten nach einer anderen Dimension aufteilen. Wählen Sie z. B. eines der Elemente aus, die in den drei wichtigsten Kategorien aufgeführt sind, und dann Analysieren nach, um die Daten nach einer anderen Dimension zu aufzuteilen. Sie können nicht dieselbe Dimension wie der Filter selbst anwenden.

> **Note**
>
> Es ist nicht möglich, einen Storage-Lens-Gruppenfilter zusammen mit einem Präfixfilter anzuwenden oder umgekehrt. Sie können eine Storage-Lens-Gruppe auch nicht weiter analysieren, indem Sie einen Präfixfilter verwenden.

Mithilfe der Registerkarte Storage-Lens-Gruppen im Amazon-S3-Storage-Lens-Dashboard lässt sich die Datenvisualisierung für die Storage-Lens-Gruppen anzupassen, die mit dem Dashboard verknüpft sind. Sie können entweder die Daten für einige Storage-Lens-Gruppen, die an das Dashboard angehängt sind, oder für alle Gruppen visualisieren.

Beachten Sie bei der Visualisierung von Storage-Lens-Gruppendaten im S3-Storage-Lens-Dashboard Folgendes:

- S3 Storage Lens aggregiert Nutzungsmetriken für ein Objekt aus allen passenden Storage-Lens-Gruppen. Wenn also ein Objekt den Filterbedingungen für zwei oder mehr Storage-Lens-Gruppen entspricht, werden Sie in Ihrer Speichernutzung wiederholte Zählungen für dasselbe Objekt sehen.
- Objekte müssen den Filtern entsprechen, die Sie in die Storage-Lens-Gruppen einbinden. Wenn den Filtern, die Sie in die Storage-Lens-Gruppe eingebunden haben, keine Objekte entsprechen, werden keine Metriken generiert. Um festzustellen, ob es nicht zugewiesene Objekte gibt, überprüfen Sie die Gesamtzahl der Objekte im Dashboard auf Konto- und Bucket-Ebene.

Aktualisieren einer Storage-Lens-Gruppe

In den folgenden Beispielen wird veranschaulicht, wie eine Amazon-S3-Storage-Lens-Gruppe aktualisiert wird. Sie können eine Storage Lens-Gruppe mithilfe der Amazon S3 S3-Konsole, AWS Command Line Interface (AWS CLI) und aktualisieren AWS SDK for Java.

Verwenden der S3-Konsole

So aktualisieren Sie eine Storage-Lens-Gruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Storage-Lens-Gruppen aus.
3. Wählen Sie unter Storage-Lens-Gruppen die zu aktualisierende Storage-Lens-Gruppe aus.
4. Wählen Sie unter Umfang die Option Bearbeiten aus.

5. Wählen Sie auf der Seite Umfang den Filter aus, der auf die Storage-Lens-Gruppe angewendet werden soll. Um mehrere Filter anzuwenden, wählen Sie Ihre Filter aus und wählen Sie den ANDlogischen Operator ODER ODER.

   - Wählen Sie für den Filter Präfixe die Option Präfixe aus und geben Sie eine Präfixzeichenfolge ein. Zum Hinzufügen mehrerer Präfixe wählen Sie Präfix hinzufügen aus. Zum Entfernen eines Präfixes wählen Sie neben dem zu entfernenden Präfix Entfernen aus.
   - Geben Sie für den Filter Objekt-Tags das Schlüssel-Wert-Paar für das Objekt ein. Wählen Sie dann Tag hinzufügen aus. Zum Entfernen eines vorhandenen Tags wählen Sie neben dem zu entfernenden Tag Entfernen aus.
   - Wählen Sie für den Filter Suffixe die Option Suffixe aus und geben Sie eine Suffixzeichenfolge ein. Um mehrere Suffixe hinzuzufügen, wählen Sie Suffix hinzufügen aus. Zum Entfernen eines Suffixes wählen Sie neben dem zu entfernenden Suffix Entfernen aus.
   - Geben Sie für den Filter Alter den Altersbereich des Objekts in Tagen an. Wählen Sie Mindestobjektalter festlegen aus und geben Sie das Mindestalter des Objekts ein. Geben Sie für Das maximale Objektalter festlegen das maximale Objektalter ein.
   - Geben Sie für den Filter Größe den Objektgrößenbereich und die Maßeinheit an. Wählen Sie Mindestobjektgröße festlegen aus und geben Sie die Mindestobjektgröße ein. Geben Sie für Die maximale Objektgröße festlegen die maximale Objektgröße ein.

6. Wählen Sie Änderungen speichern. Die Detailseite für die Storage-Lens-Gruppe wird angezeigt.
7. (Optional) Wenn Sie ein neues AWS Ressourcen-Tag hinzufügen möchten, scrollen Sie zum Abschnitt mit den AWS Ressourcen-Tags und wählen Sie dann Tags hinzufügen aus. Die Seite Add tags (Tags hinzufügen) wird angezeigt.

   Fügen Sie das neue Schlüssel-Wert-Paar hinzu und wählen Sie dann Änderungen speichern aus. Die Detailseite für die Storage-Lens-Gruppe wird angezeigt.
8. (Optional) Wenn Sie ein vorhandenes AWS Ressourcen-Tag entfernen möchten, scrollen Sie zum Abschnitt mit den AWS Ressourcen-Tags und wählen Sie das Ressourcen-Tag aus. Wählen Sie dann Löschen aus. Das Dialogfeld AWS -Ressourcen-Tags löschen wird angezeigt.

   Wählen Sie erneut Löschen aus, um das AWS -Ressourcen-Tag dauerhaft zu löschen.

   > **Note**
   >
   > Nachdem Sie ein AWS Ressourcen-Tag dauerhaft gelöscht haben, kann es nicht wiederhergestellt werden.

Mit dem AWS CLI

Der folgende AWS CLI Beispielbefehl gibt die Konfigurationsdetails für eine Storage Lens-Gruppe mit dem Namen zurück*marketing-department*. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control get-storage-lens-group --account-id 111122223333 \
--region us-east-1 --name marketing-department
```

Im folgenden AWS CLI Beispiel wird eine Storage Lens-Gruppe aktualisiert. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control update-storage-lens-group --account-id 111122223333 \
--region us-east-1 --storage-lens-group=file://./marketing-department.json
```

JSONBeispielkonfigurationen finden Sie unterKonfiguration von Storage-Lens-Gruppen.

Verwendung des AWS SDK for Java

Im folgenden AWS SDK for Java Beispiel werden die Konfigurationsdetails für die *Marketing-Department* Storage Lens-Gruppe im Konto zurückgegeben*111122223333*. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupResponse;

public class GetStorageLensGroup {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";
```

```
        try {
            GetStorageLensGroupRequest getRequest =
 GetStorageLensGroupRequest.builder()
                    .name(storageLensGroupName)
                    .accountId(accountId).build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            GetStorageLensGroupResponse response =
 s3ControlClient.getStorageLensGroup(getRequest);
            System.out.println(response);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

In dem folgenden Beispiel wird die Storage-Lens-Gruppe *Marketing-Department* im Konto *111122223333* aktualisiert. In diesem Beispiel wird der Dashboard-Umfang so aktualisiert, dass die Objekte berücksichtigt werden, die einem der folgenden Suffixe entsprechen: *.png*, *.gif*, *.jpg* oder *.jpeg*. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.StorageLensGroup;
import software.amazon.awssdk.services.s3control.model.StorageLensGroupFilter;
import software.amazon.awssdk.services.s3control.model.UpdateStorageLensGroupRequest;

public class UpdateStorageLensGroup {
    public static void main(String[] args) {
```

```
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            // Create updated filter.
            StorageLensGroupFilter suffixFilter = StorageLensGroupFilter.builder()
                    .matchAnySuffix(".png", ".gif", ".jpg", ".jpeg")
                    .build();

            StorageLensGroup storageLensGroup = StorageLensGroup.builder()
                    .name(storageLensGroupName)
                    .filter(suffixFilter)
                    .build();

            UpdateStorageLensGroupRequest updateStorageLensGroupRequest =
 UpdateStorageLensGroupRequest.builder()
                    .name(storageLensGroupName)
                    .storageLensGroup(storageLensGroup)
                    .accountId(accountId)
                    .build();

            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.updateStorageLensGroup(updateStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

JSONBeispielkonfigurationen finden Sie unterKonfiguration von Storage-Lens-Gruppen.

Verwaltung von AWS Ressourcen-Tags mit Storage Lens-Gruppen

Jede Amazon S3 Storage Lens-Gruppe wird als AWS Ressource mit eigenem Amazon-Ressourcennamen (ARN) gezählt. Daher können Sie bei der Konfiguration einer Storage-Lens-

Gruppe optional AWS -Ressourcen-Tags zur Gruppe hinzufügen. Es ist möglich, für jede Storage-Lens-Gruppe bis zu 50 Tags hinzuzufügen. Zum Erstellen einer Storage-Lens-Gruppe mit Tags benötigen Sie die Berechtigungen `s3:TagResource` und `s3:CreateStorageLensGroup`.

Sie können AWS Ressourcen-Tags verwenden, um Ressourcen nach Abteilung, Geschäftsbereich oder Projekt zu kategorisieren. Dies ist sinnvoll, wenn viele Ressourcen desselben Typs vorliegen. Durch die Anwendung von Tags können Sie eine bestimmte Storage-Lens-Gruppe auf Grundlage der ihr zugewiesenen Tags schnell identifizieren. Sie können Tags auch verwenden, um Ihre Kosten zu kategorisieren und zu verfolgen.

Wenn Sie Ihrer Storage Lens-Gruppe ein AWS Ressourcen-Tag hinzufügen, aktivieren Sie außerdem die attributbasierte Zugriffskontrolle (). ABAC ABACist eine Autorisierungsstrategie, die Berechtigungen auf der Grundlage von Attributen, in diesem Fall Tags, definiert. Sie können in Ihren IAM Richtlinien auch Bedingungen verwenden, die Ressourcen-Tags angeben, um den Zugriff auf AWS Ressourcen zu kontrollieren.

Sie können Tag (Markierung)-Schlüssel und -Werte bearbeiten und Tags (Markierungen) jederzeit von einer Ressource entfernen. Beachten Sie folgende Einschränkungen:

- Bei Tag-Schlüsseln und Tag-Werten muss die Groß- und Kleinschreibung beachtet werden.
- Wenn Sie ein Tag (Markierung) mit demselben Schlüssel wie ein vorhandener Tag (Markierung) für die Ressource hinzufügen, wird der alte Wert mit dem neuen überschrieben.
- Wenn Sie eine Ressource löschen, werden alle Tags (Markierungen) der Ressource ebenfalls gelöscht.
- Nehmen Sie keine privaten oder vertraulichen Daten in Ihre AWS Ressourcen-Tags auf.
- System-Tags (mit Tag-Schlüsseln, die mit `aws:` beginnen) werden nicht unterstützt.
- Die einzelnen Tag-Schlüssel dürfen maximal 128 Zeichen lang sein. Die einzelnen Tag-Werte dürfen maximal 256 Zeichen lang sein.

Die folgenden Beispiele zeigen, wie Sie AWS Ressourcen-Tags mit Storage Lens-Gruppen verwenden.

Themen

- Hinzufügen eines AWS Ressourcen-Tags zu einer Storage Lens-Gruppe
- Aktualisieren von Tag-Werten einer Storage-Lens-Gruppe
- Löschen eines AWS Ressourcen-Tags aus einer Storage Lens-Gruppe

- Auflisten von Storage-Lens-Gruppen-Tags

Hinzufügen eines AWS Ressourcen-Tags zu einer Storage Lens-Gruppe

Die folgenden Beispiele zeigen, wie AWS Ressourcen-Tags zu einer Amazon S3 Storage Lens-Gruppe hinzugefügt werden. Sie können Ressourcen-Tags mithilfe der Amazon S3 S3-Konsole hinzufügen, AWS Command Line Interface (AWS CLI) und AWS SDK for Java.

Verwenden der S3-Konsole

Um ein AWS Ressourcen-Tag zu einer Storage Lens-Gruppe hinzuzufügen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Storage-Lens-Gruppen aus.
3. Wählen Sie unter Storage-Lens-Gruppen die zu aktualisierende Storage-Lens-Gruppe aus.
4. Wählen Sie unter AWS -Ressourcen-Tags die Option Tags hinzufügen aus.
5. Fügen Sie auf der Seite Tags hinzufügen das neue Schlüssel-Wert-Paar hinzu.

   > **Note**
   >
   > Wenn Sie ein neues Tag hinzufügen, dessen Schlüssel mit dem eines vorhandenen Tags identisch ist, wird der vorherige Tag-Wert überschrieben.

6. (Optional) Um mehrere neue Tags hinzuzufügen, wählen Sie erneut Tag hinzufügen aus und fügen Sie weiterhin neue Einträge hinzu. Sie können Ihrer Storage Lens-Gruppe bis zu 50 AWS Ressourcen-Tags hinzufügen.
7. (Optional) Wenn Sie einen neu hinzugefügten Eintrag entfernen möchten, wählen Sie neben dem zu entfernenden Tag Entfernen aus.
8. Wählen Sie Änderungen speichern.

Unter Verwendung der AWS CLI

Mit dem folgenden AWS CLI Beispielbefehl werden zwei Ressourcentags zu einer vorhandenen Storage Lens-Gruppe mit dem Namen hinzugefügt*`marketing-department`*. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws s3control tag-resource --account-id 111122223333 \
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/marketing-
department \
--region us-east-1 --tags Key=k1,Value=v1 Key=k2,Value=v2
```

Verwendung des AWS SDK for Java

Im folgenden AWS SDK for Java Beispiel werden einer vorhandenen Storage Lens-Gruppe zwei AWS Ressourcentags hinzugefügt. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.Tag;
import software.amazon.awssdk.services.s3control.model.TagResourceRequest;

public class TagResource {
    public static void main(String[] args) {
        String resourceARN = "Resource_ARN";
        String accountId = "111122223333";

        try {
            Tag resourceTag1 = Tag.builder()
                .key("resource-tag-key-1")
                .value("resource-tag-value-1")
                .build();
            Tag resourceTag2 = Tag.builder()
                    .key("resource-tag-key-2")
                    .value("resource-tag-value-2")
                    .build();
            TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
                    .resourceArn(resourceARN)
                    .tags(resourceTag1, resourceTag2)
                    .accountId(accountId)
                    .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
```

```
                    .build();
            s3ControlClient.tagResource(tagResourceRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Aktualisieren von Tag-Werten einer Storage-Lens-Gruppe

Die folgenden Beispiele zeigen, wie Sie Storage Lens-Gruppen-Tag-Werte mithilfe der Amazon S3 S3-Konsole, AWS Command Line Interface (AWS CLI) und aktualisieren AWS SDK for Java.

Verwenden der S3-Konsole

Um ein AWS Ressourcen-Tag für eine Storage Lens-Gruppe zu aktualisieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Storage-Lens-Gruppen aus.
3. Wählen Sie unter Storage-Lens-Gruppen die zu aktualisierende Storage-Lens-Gruppe aus.
4. Wählen Sie unter AWS -Ressourcen-Tags das zu aktualisierende Tag aus.
5. Fügen Sie den neuen Tag-Wert hinzu. Verwenden Sie hierbei den Schlüssel, der dem Schlüssel des zu aktualisierenden Schlüssel-Wert-Paares entspricht. Klicken Sie auf das Häkchen-Symbol, um den Tag-Wert zu aktualisieren.

   > **Note**
   >
   > Wenn Sie ein neues Tag hinzufügen, dessen Schlüssel mit dem eines vorhandenen Tags identisch ist, wird der vorherige Tag-Wert überschrieben.

6. (Optional) Wenn Sie neue Tags hinzufügen möchten, wählen Sie Tag hinzufügen aus, um neue Einträge hinzuzufügen. Die Seite Add tags (Tags hinzufügen) wird angezeigt.

Sie können bis zu 50 AWS Ressourcen-Tags für Ihre Storage Lens-Gruppe hinzufügen. Wenn Sie das Hinzufügen der neuen Tags beendet haben, klicken Sie auf Änderungen speichern.

7. (Optional) Wenn Sie einen neu hinzugefügten Eintrag entfernen möchten, wählen Sie neben dem zu entfernenden Tag Entfernen aus. Wenn Sie mit dem Entfernen der Tags fertig sind, wählen Sie Speichern aus.

Unter Verwendung der AWS CLI

Mit dem folgenden AWS CLI Beispielbefehl werden zwei Tagwerte für die Storage Lens-Gruppe mit dem Namen aktualisiert*marketing-department*. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control tag-resource --account-id 111122223333 \
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/marketing-
department \
--region us-east-1 --tags Key=k1,Value=v3 Key=k2,Value=v4
```

Verwendung des AWS SDK for Java

Im folgenden AWS SDK for Java Beispiel werden zwei Tagwerte der Storage Lens-Gruppe aktualisiert. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.Tag;
import software.amazon.awssdk.services.s3control.model.TagResourceRequest;

public class UpdateTagsForResource {
    public static void main(String[] args) {
        String resourceARN = "Resource_ARN";
        String accountId = "111122223333";

        try {
```

```
            Tag updatedResourceTag1 = Tag.builder()
                .key("resource-tag-key-1")
                .value("resource-tag-updated-value-1")
                .build();
            Tag updatedResourceTag2 = Tag.builder()
                    .key("resource-tag-key-2")
                    .value("resource-tag-updated-value-2")
                    .build();
            TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
                    .resourceArn(resourceARN)
                    .tags(updatedResourceTag1, updatedResourceTag2)
                    .accountId(accountId)
                    .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.tagResource(tagResourceRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Löschen eines AWS Ressourcen-Tags aus einer Storage Lens-Gruppe

Die folgenden Beispiele zeigen, wie Sie ein AWS Ressourcen-Tag aus einer Storage Lens-Gruppe löschen. Sie können Tags mithilfe der Amazon S3 S3-Konsole AWS Command Line Interface (AWS CLI) und löschen AWS SDK for Java.

Verwenden der S3-Konsole

Um ein AWS Ressourcen-Tag aus einer Storage Lens-Gruppe zu löschen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Storage-Lens-Gruppen aus.

3. Wählen Sie unter Storage-Lens-Gruppen die zu aktualisierende Storage-Lens-Gruppe aus.
4. Wählen Sie unter AWS -Ressourcen-Tags das zu löschende Schlüssel-Wert-Paar aus.
5. Wählen Sie Löschen. Das Dialogfeld AWS -Ressourcen-Tags löschen wird angezeigt.

> **Note**
>
> Wenn Tags für die Zugriffskontrolle verwendet werden, kann die Durchführung dieser Aktion Auswirkungen auf zugehörige Ressourcen haben. Nachdem Sie ein Tag dauerhaft gelöscht haben, kann es nicht wiederhergestellt werden.

6. Wählen Sie Löschen aus, um das Schlüssel-Wert-Paar dauerhaft zu löschen.

Verwenden Sie den AWS CLI

Mit dem folgenden AWS CLI Befehl werden zwei AWS Ressourcen-Tags aus einer vorhandenen Storage Lens-Gruppe gelöscht: Um diesen Beispielbefehl zu verwenden, ersetzen Sie die *user input placeholders* durch Ihre eigenen Informationen.

```
aws s3control untag-resource --account-id 111122223333 \
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/Marketing-Department \
--region us-east-1 --tag-keys k1 k2
```

Verwendung des AWS SDK for Java

Im folgenden AWS SDK for Java Beispiel werden zwei AWS Ressourcen-Tags aus der Storage Lens-Gruppe Amazon Resource Name (ARN) gelöscht, die Sie im Konto *111122223333* angeben. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.UntagResourceRequest;
```

```
public class UntagResource {
    public static void main(String[] args) {
        String resourceARN = "Resource_ARN";
        String accountId = "111122223333";

        try {
            String tagKey1 = "resource-tag-key-1";
            String tagKey2 = "resource-tag-key-2";
            UntagResourceRequest untagResourceRequest = UntagResourceRequest.builder()
                    .resourceArn(resourceARN)
                    .tagKeys(tagKey1, tagKey2)
                    .accountId(accountId)
                    .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.untagResource(untagResourceRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Auflisten von Storage-Lens-Gruppen-Tags

Die folgenden Beispiele zeigen, wie Sie die AWS Ressourcen-Tags auflisten, die einer Storage Lens-Gruppe zugeordnet sind. Sie können Tags mithilfe der Amazon S3 S3-Konsole AWS Command Line Interface (AWS CLI) und auflisten AWS SDK for Java.

Verwenden der S3-Konsole

So überprüfen Sie die Liste der Tags und Tag-Werte für eine Storage-Lens-Gruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Storage-Lens-Gruppen aus.

3. Wählen Sie unter Storage-Lens-Gruppen die betreffende Storage-Lens-Gruppe aus.
4. Scrollen Sie nach unten zum Abschnitt AWS -Ressourcen-Tags. Alle benutzerdefinierten AWS Ressourcen-Tags, die zu Ihrer Storage Lens-Gruppe hinzugefügt wurden, werden zusammen mit ihren Tag-Werten aufgelistet.

Mit dem AWS CLI

Der folgende AWS CLI Beispielbefehl listet alle Tag-Werte der Storage Lens-Gruppe für die Storage Lens-Gruppe mit dem Namen auf*marketing-department*. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control list-tags-for-resource --account-id 111122223333 \
--resource-arn arn:aws:s3:us-east-1:111122223333:storage-lens-group/marketing-
department \
--region us-east-1
```

Verwendung des AWS SDK for Java

Das folgende AWS SDK for Java Beispiel listet die Storage Lens-Gruppen-Tag-Werte für die Storage Lens-Gruppe Amazon Resource Name (ARN) auf, die Sie angeben. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.ListTagsForResourceRequest;
import software.amazon.awssdk.services.s3control.model.ListTagsForResourceResponse;

public class ListTagsForResource {
    public static void main(String[] args) {
        String resourceARN = "Resource_ARN";
        String accountId = "111122223333";

        try {
            ListTagsForResourceRequest listTagsForResourceRequest =
 ListTagsForResourceRequest.builder()
```

```
                    .resourceArn(resourceARN)
                    .accountId(accountId)
                    .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            ListTagsForResourceResponse response =
 s3ControlClient.listTagsForResource(listTagsForResourceRequest);
            System.out.println(response);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Auflisten aller Storage-Lens-Gruppen

In den folgenden Beispielen wird veranschaulicht, wie alle Amazon-S3-Storage-Lens-Gruppen in einem AWS-Konto und in einer Heimatregion aufgelistet werden. Diese Beispiele zeigen, wie alle Storage Lens-Gruppen mithilfe der Amazon S3 S3-Konsole, AWS Command Line Interface (AWS CLI) und aufgelistet AWS SDK for Java werden.

Verwenden der S3-Konsole

So listen Sie alle Storage-Lens-Gruppen in einem Konto und einer Heimatregion auf

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Storage-Lens-Gruppen aus.
3. Unter Storage-Lens-Gruppen wird die Liste der Storage-Lens-Gruppen in Ihrem Konto angezeigt.

Verwenden Sie den AWS CLI

Im folgenden AWS CLI Beispiel werden alle Storage Lens-Gruppen für Ihr Konto aufgeführt. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control list-storage-lens-groups --account-id 111122223333 \
--region us-east-1
```

Verwendung des AWS SDK for Java

Im folgenden AWS SDK for Java Beispiel werden die Storage Lens-Gruppen für das Konto aufgeführt*111122223333*. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.ListStorageLensGroupsRequest;
import software.amazon.awssdk.services.s3control.model.ListStorageLensGroupsResponse;

public class ListStorageLensGroups {
    public static void main(String[] args) {
        String accountId = "111122223333";

        try {
            ListStorageLensGroupsRequest listStorageLensGroupsRequest =
 ListStorageLensGroupsRequest.builder()
                    .accountId(accountId)
                    .build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            ListStorageLensGroupsResponse response =
 s3ControlClient.listStorageLensGroups(listStorageLensGroupsRequest);
            System.out.println(response);
        } catch (AmazonServiceException e) {
```

```
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Anzeigen von Details zu Storage-Lens-Gruppen

In den folgenden Beispielen wird veranschaulicht, wie die Konfigurationsdetails einer Amazon-S3-Storage-Lens-Gruppe angezeigt werden. Sie können diese Details mithilfe der Amazon S3 S3-Konsole AWS Command Line Interface (AWS CLI) und anzeigen AWS SDK for Java.

Verwenden der S3-Konsole

So zeigen Sie die Konfigurationsdetails einer Storage-Lens-Gruppe an

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Storage-Lens-Gruppen aus.
3. Wählen Sie unter Storage-Lens-Gruppen die Optionsschaltfläche neben der betreffenden Storage-Lens-Gruppe aus.
4. Wählen Sie die Option View details aus. Jetzt können Sie die Details der Storage-Lens-Gruppe überprüfen.

Verwenden Sie den AWS CLI

Im folgenden AWS CLI Beispiel werden die Konfigurationsdetails für eine Storage Lens-Gruppe zurückgegeben. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *`user input placeholders`* durch Ihre Informationen.

```
aws s3control get-storage-lens-group --account-id 111122223333 \
--region us-east-1 --name marketing-department
```

Verwendung des AWS SDK for Java

Im folgenden AWS SDK for Java Beispiel werden die Konfigurationsdetails für die *Marketing-Department* im Konto angegebene Storage Lens-Gruppe zurückgegeben*111122223333*. Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupRequest;
import software.amazon.awssdk.services.s3control.model.GetStorageLensGroupResponse;

public class GetStorageLensGroup {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            GetStorageLensGroupRequest getRequest =
 GetStorageLensGroupRequest.builder()
                    .name(storageLensGroupName)
                    .accountId(accountId).build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            GetStorageLensGroupResponse response =
 s3ControlClient.getStorageLensGroup(getRequest);
            System.out.println(response);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
```

```
    }
}
```

Löschen einer Storage-Lens-Gruppe

Die folgenden Beispiele zeigen, wie Sie eine Amazon S3 Storage Lens-Gruppe mithilfe der Amazon S3 S3-Konsole, AWS Command Line Interface (AWS CLI) und löschen AWS SDK for Java.

Verwenden der S3-Konsole

So löschen Sie eine Storage-Lens-Gruppe

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Storage-Lens-Gruppen aus.
3. Aktivieren Sie unter Storage-Lens-Gruppen das Optionsfeld neben der zu löschenden Storage-Lens-Gruppe.
4. Wählen Sie Löschen. Das Dialogfeld Storage-Lens-Gruppe löschen wird angezeigt.
5. Wählen Sie erneut Löschen aus, um die Storage-Lens-Gruppe dauerhaft zu löschen.

> **Note**
>
> Nachdem Sie eine Storage-Lens-Gruppe gelöscht haben, kann sie nicht wiederhergestellt werden.

Verwenden Sie den AWS CLI

Im folgenden AWS CLI Beispiel wird die Storage Lens-Gruppe mit dem Namen *marketing-department* gelöscht. Wenn Sie diesen Beispielbefehl verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
aws s3control delete-storage-lens-group --account-id 111122223333 \
--region us-east-1 --name marketing-department
```

Verwendung des AWS SDK for Java

Im folgenden AWS SDK for Java Beispiel wird die Storage Lens-Gruppe mit dem Namen account *Marketing-Department* gelöscht. *111122223333* Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie *user input placeholders* durch Ihre Informationen.

```
package aws.example.s3control;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.DeleteStorageLensGroupRequest;

public class DeleteStorageLensGroup {
    public static void main(String[] args) {
        String storageLensGroupName = "Marketing-Department";
        String accountId = "111122223333";

        try {
            DeleteStorageLensGroupRequest deleteStorageLensGroupRequest =
 DeleteStorageLensGroupRequest.builder()
                    .name(storageLensGroupName)
                    .accountId(accountId).build();
            S3ControlClient s3ControlClient = S3ControlClient.builder()
                    .region(Region.US_WEST_2)
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .build();
            s3ControlClient.deleteStorageLensGroup(deleteStorageLensGroupRequest);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it and returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory

Sie können Amazon S3 Inventory verwenden, um Ihren Speicher zu verwalten. Sie können es beispielsweise für die Prüfung und Meldung des Replikations- und Verschlüsselungsstatus Ihrer Objekte für Unternehmens-, Compliance- und regulatorische Anforderungen verwenden. Sie können auch Geschäftsabläufe und Big-Data-Jobs vereinfachen und beschleunigen, indem Sie Amazon S3 Inventory verwenden, das eine geplante Alternative zu den synchronen `List` API Vorgängen von Amazon S3 darstellt. Amazon S3 Inventory verwendet die `List` API Operationen nicht zur Prüfung Ihrer Objekte und hat keinen Einfluss auf die Anforderungsrate Ihres Buckets.

Amazon S3 Inventory bietet kommagetrennte Werte ()CSV, Apache optimierte Zeile spaltenförmig () oder ORC Apache ParquetAusgabedateien, die Ihre Objekte und die entsprechenden Metadaten täglich oder wöchentlich für einen S3-Bucket oder Objekte mit einem gemeinsamen Präfix auflisten (d. h. Objekte, deren Namen mit einer gemeinsamen Zeichenfolge beginnen). Wenn Sie eine wöchentliche Bestandsaufnahme einrichten, wird jeden Sonntag (UTCZeitzone) nach dem ersten Bericht ein Bericht generiert. Informationen zu den Preisen für Amazon S3 Inventory finden Sie unter Amazon-S3-Preise.

Sie können mehrere Bestandslisten für einen Bucket konfigurieren. Wenn Sie eine Bestandsliste konfigurieren, können Sie Folgendes angeben:

- welche Objektmetadaten in das Inventar aufgenommen werden sollen
- ob alle Objektversionen oder nur aktuelle Versionen aufgelistet werden sollen
- wo die Ausgabe der Bestandslistendatei gespeichert werden soll
- ob der Bestandsbericht täglich oder wöchentlich generiert werden soll
- ob die Bestandslistendatei verschlüsselt werden soll

Sie können Amazon S3 S3-Inventar mit SQL Standardabfragen abfragen, indem Sie Amazon Athena, Amazon Redshift Spectrum und andere Tools verwenden, wie Presto, Apache Hive, und Apache Spark. Weitere Informationen zur Verwendung von Athena zur Abfrage Ihrer Lagerbestandsdateien finden Sie unterthe section called “Bestandsabfrage mit Athena”.

> **Note**
>
> Es könnte bis zu 48 Stunden dauern, bis Amazon S3 den ersten Inventarbericht bereitstellt.

Quell- und Ziel-Buckets

Der Bucket, dessen Objekte die Bestandserfassung auflistet, wird als Quell-Bucket bezeichnet. Der Bucket, in dem die Datei mit der Bestandsliste gespeichert ist, wird als Ziel-Bucket bezeichnet.

Quell-Bucket

Der Bestand listet die Objekte auf, die im Quell-Bucket gespeichert sind. Sie können eine Bestandsliste für einen ganzen Bucket erhalten oder die Liste nach dem Präfix eines Objektschlüsselnamens filtern.

Der Quell-Bucket:

- Enthält die Objekte, die im Bestand aufgelistet sind.
- Enthält die Konfiguration für den Bestand.

Ziel-Bucket

Amazon S3 Inventory listet Dateien auf, die in den Ziel-Bucket geschrieben werden. Um alle Bestandslisten-Dateien an einem gemeinsamen Speicherort im Ziel-Bucket zu gruppieren, können Sie ein Zielpräfix in der Bestands-Konfiguration angeben.

Der Ziel-Bucket:

- Enthält die Dateilisten für den Bestand.
- Enthält die Manifestdateien, die alle Bestandslistendateien enthalten, die im Ziel-Bucket gespeichert sind. Weitere Informationen finden Sie unter Bestandsmanifest.
- Erfordert eine Bucket-Richtlinie, um Amazon S3 die Berechtigung zu erteilen, die Eigentümerschaft an dem Bucket zu überprüfen, ebenso wie die Berechtigung, Dateien in den Bucket zu schreiben.
- Muss sich im selben Bucket befinden AWS-Region wie der Quell-Bucket.
- Kann gleich dem Quell-Bucket sein.
- Kann einem anderen Konto gehören AWS-Konto als dem Konto, dem der Quell-Bucket gehört.

Amazon-S3-Inventory-Liste

Eine Datei mit einer Bestandsliste enthält eine Liste der Objekte im Quell-Bucket sowie die Metadaten für jedes Objekt. Eine Bestandslistendatei wird im Ziel-Bucket in einem der folgenden Formate gespeichert:

- Als komprimierte CSV Datei mit GZIP
- Als Apache optimierte row columnar (ORC) -Datei, komprimiert mit ZLIB
- Als Apache Parquet mit Snappy komprimierte Datei

> **Note**
>
> Es kann nicht garantiert werden, dass Objekte in Berichten von Amazon S3 Inventory auf bestimmte Weise sortiert werden.

Eine Datei mit einer Bestandsliste enthält eine Liste der Objekte im Quell-Bucket sowie die Metadaten für jedes aufgelistete Objekt:

- Bucket name (Bucket-Name) – Der Name des Buckets, für den der Bestand gilt.
- Key name (Schlüsselname) – Name des Objektschlüssels (oder Schlüssel), der das Objekt in dem Bucket eindeutig identifiziert. Wenn Sie das CSV Dateiformat verwenden, ist der Schlüsselname URL -kodiert und muss dekodiert werden, bevor Sie ihn verwenden können.
- Version ID (Versions-ID) – ID der Objektversion. Wenn Sie das Versioning für einen Bucket aktivieren, weist Amazon S3 allen Objekten, die dem Bucket hinzugefügt werden, eine Versionsnummer zu. Weitere Informationen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning. (Dieses Feld ist nicht enthalten, wenn die Liste nur für die aktuelle Version der Objekte konfiguriert ist.)
- IsLatest— Legt fest, `True` ob es sich bei dem Objekt um die aktuelle Version des Objekts handelt. (Dieses Feld ist nicht enthalten, wenn die Liste nur für die aktuelle Version der Objekte konfiguriert ist.)
- Delete marker (Löschmarkierung) – Auf `True` gesetzt, wenn es sich bei dem Objekt um eine Löschmarkierung handelt. Weitere Informationen finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning. (Dieses Feld wird dem Bericht automatisch hinzugefügt, wenn Sie den Bericht so konfiguriert haben, dass alle Versionen der Objekte aufgenommen werden.)
- Größe – Die Objektgröße in Byte, ohne die Größe von unvollständigen mehrteiligen Uploads, Objektmetadaten und Löschmarkierungen.
- Last modified date (Letztes Änderungsdatum) – Datum der Erstellung oder der letzten Änderung des Objekts, je nachdem, welches neuer ist.

- ETag— Das Entity-Tag (ETag) ist ein Hash des Objekts. Das ETag spiegelt nur Änderungen am Inhalt eines Objekts wider, nicht an seinen Metadaten. Das ETag kann eine MD5 Zusammenfassung der Objektdaten sein. Dies hängt davon ab, wie das Objekt erstellt und verschlüsselt wurde. Weitere Informationen finden Sie unter Objectin der Amazon Simple Storage Service API Reference.
- Storage class (Speicherklasse) – Die für die Speicherung des Objekts verwendete Speicherklasse. Auf `STANDARD`, `REDUCED_REDUNDANCY`, `STANDARD_IA`, `ONEZONE_IA`, `INTELLIGENT_TIERING`, `GLACIER`, `DEEP_ARCHIVE`, `OUTPOSTS`, `GLACIER_IR` oder `SNOW` festgelegt. Weitere Informationen finden Sie unter Amazon S3 S3-Speicherklassen verstehen und verwalten.

  > **Note**
  >
  > S3 Inventory unterstützt S3 Express One Zone nicht.

- Multipart upload flag (Markierung für mehrteiligen Upload) – Auf `True` gesetzt, wenn das Objekt als mehrteiliger Upload hochgeladen wurde. Weitere Informationen finden Sie unter Hochladen und Kopieren von Objekten mit mehrteiligen Uploads.
- Replikationsstatus – Auf `PENDING`, `COMPLETED`, `FAILED`, oder `REPLICA` gesetzt. Weitere Informationen finden Sie unter Abrufen von Replikationsstatusinformationen.
- Verschlüsselungsstatus — Der serverseitige Verschlüsselungsstatus, abhängig von der Art des verwendeten Verschlüsselungsschlüssels — serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3), serverseitige Verschlüsselung mit AWS Key Management Service () Schlüsseln (SSE-AWS KMS), zweischichtige serverseitige Verschlüsselung mit Schlüsseln (-KMS) oder serverseitige Verschlüsselung mit vom Kunden bereitgestellten AWS KMS Schlüsseln (DSSE-CKMS). SSE Setzen Sie den Wert auf `SSE-S3`, `SSE-KMS`, `DSSE-KMS`, `SSE-C` oder `NOT-SSE`. Der Status `NOT-SSE` bedeutet, dass das Objekt nicht mit serverseitiger Verschlüsselung verschlüsselt ist. Weitere Informationen finden Sie unter Datenschutz durch Verschlüsselung.
- S3 Object Lock retain until date (Beibehaltungsfrist für S3-Objektsperre) – Das Datum, bis zu dem das gesperrte Objekt nicht gelöscht werden kann. Weitere Informationen finden Sie unter Sperren von Objekten mit Object Lock.
- S3 Object Lock retention mode (Aufbewahrungsmodus für S3-Objekt) – Auf `Governance` oder `Compliance` gesetzt für Objekte, die gesperrt sind. Weitere Informationen finden Sie unter Sperren von Objekten mit Object Lock.

- S3 Object Lock legal hold status (Rechtlicher Aufbewahrungsstatus der S3-Objektsperre) – Auf `On` gesetzt, wenn für ein Objekt eine rechtliche Aufbewahrungsfrist gilt. Andernfalls lautet der Wert `Off`. Weitere Informationen finden Sie unter Sperren von Objekten mit Object Lock.
- S3 Intelligent-Tiering access tier (S3-Intelligent-Tiering-Zugriffsebene) – Zugriffsebene (häufig oder selten) des Objekts bei Speicherung in der Speicherklasse S3 Intelligent-Tiering. Setzen Sie den Wert auf `FREQUENT`, `INFREQUENT`, `ARCHIVE_INSTANT_ACCESS`, `ARCHIVE` oder `DEEP_ARCHIVE`. Weitere Informationen finden Sie unter Speicherklasse zur automatischen Optimierung von Daten mit sich ändernden oder unbekannten Zugriffsmustern.
- S3 Buckt Key status (Status des S3-Bucket-Schlüssels) – Auf `ENABLED` oder `DISABLED` gesetzt. Gibt an, ob das Objekt einen S3-Bucket-Schlüssel für - verwendet. SSE KMS Weitere Informationen finden Sie unter Verwenden von Amazon-S3-Bucket-Schlüssel.
- Prüfsummen-Algorithmus – Gibt den Algorithmus an, mit dem die Prüfsumme für das Objekt erstellt wurde. Weitere Informationen finden Sie unter Verwenden unterstützter Prüfsummenalgorithmen.
- Objektzugriffskontrollliste — Eine Zugriffskontrollliste (ACL) für jedes Objekt, die definiert, welchen AWS-Konten Gruppen Zugriff auf dieses Objekt gewährt wird und welche Art von Zugriff gewährt wird. Das ACL Objektfeld ist im JSON Format definiert. Ein S3-Inventarbericht enthält ObjekteACLs, die Objekten in Ihrem Quell-Bucket zugeordnet ACLs sind, auch wenn sie für den Bucket deaktiviert sind. Weitere Informationen erhalten Sie unter Mit dem ACL Objektfeld arbeiten und Übersicht über die Zugriffskontrollliste (ACL).

> **Note**
>
> Das ACL Objektfeld ist im JSON Format definiert. In einem Inventarbericht wird der Wert für das ACL Objektfeld als Base64-kodierte Zeichenfolge angezeigt.
> Nehmen wir beispielsweise an, Sie haben das folgende ACL Objektfeld im Format: JSON
>
> ```
> {
>         "version": "2022-11-10",
>         "status": "AVAILABLE",
>         "grants": [{
>             "canonicalId": "example-canonical-user-ID",
>             "type": "CanonicalUser",
>             "permission": "READ"
>         }]
> }
> ```

Das ACL Objektfeld ist codiert und wird als folgende Base64-kodierte Zeichenfolge angezeigt:

```
eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IkFWQUlMQUJMRSIsImdyYW50cyI6W3siY2Fub25pY2Fs
```

Um den dekodierten Wert JSON für das ACL Objektfeld zu erhalten, können Sie dieses Feld in Amazon Athena abfragen. Abfragebeispiele finden Sie unter Abfragen von Amazon S3 Inventory mit Amazon Athena.

- Objekteigentümer — Die kanonische Benutzer-ID des Besitzers des Objekts. Weitere Informationen finden Sie unter Finden Sie die kanonische Benutzer-ID für Ihr AWS Konto im Referenzhandbuch zur KontoverwaltungAWS .

> **Note**
>
> Wenn ein Objekt basierend auf seiner Lebenszykluskonfiguration das Ende seiner Lebensdauer erreicht hat, stellt Amazon S3 das Objekt zum Entfernen in eine Warteschlange und entfernt es asynchron. Daher kann es eine Verzögerung zwischen dem Ablaufdatum und dem Datum geben, an dem Amazon S3 ein Objekt entfernt. Der Bestandsbericht enthält die Objekte, die abgelaufen sind, aber noch nicht entfernt wurden. Weitere Informationen über die Ablaufaktionen im S3-Lebenszyklus finden Sie unter Auslaufende Objekte.

Im Folgenden finden Sie ein Beispiel für einen Inventarbericht mit zusätzlichen Metadatenfeldern, der aus vier Datensätzen besteht.

```
amzn-s3-demo-bucket1    example-object-1    EXAMPLEDC8l.XJCENlF7LePaNIIvs001    TRUE
     1500    2024-08-15T15:28:26.0004    EXAMPLE21e1518b92f3d92773570f600    STANDARD
  FALSE    COMPLETED    SSE-KMS    2025-01-25T15:28:26.000Z    COMPLIANCE    Off
 ENABLED
 eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IkFWQUlMQUJMRSIsImdyYW50cyI6W3sicGVybWlzc2lvbiI6Ik
    EXAMPLE766e8f6b115d93d41df2eac420aa4a465d177c1398bc6a088c76b7000
amzn-s3-demo-bucket1    example-object-2    EXAMPLEDC8l.XJCENlF7LePaNIIvs002
 TRUE        200    2024-08-21T15:28:26.000Z    EXAMPLE21e1518b92f3d92773570f601
  INTELLIGENT_TIERING    FALSE    COMPLETED    SSE-KMS    2025-01-25T15:28:26.000Z
    COMPLIANCE    Off    INFREQUENT    ENABLED    SHA-256
 eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IkFWQUlMQUJMRSIsImdyYW50cyI6W3sicGVybWlzc2lvbiI6Ik
    EXAMPLE766e8f6b115d93d41df2eac420aa4a465d177c1398bc6a088c76b7001
```

```
amzn-s3-demo-bucket1    example-object-3    EXAMPLEDC8l.XJCENlF7LePaNIIvs003    TRUE
      12500    2023-01-15T15:28:30.000Z    EXAMPLE21e1518b92f3d92773570f602    STANDARD
    FALSE    REPLICA    SSE-KMS    2025-01-25T15:28:26.000Z    GOVERNANCE    On
 ENABLED
 eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IkFWQUlMQUJMRSIsImdyYW50cyI6W3sicGVybWlzc2lvbiI6Ik
    EXAMPLE766e8f6b115d93d41df2eac420aa4a465d177c1398bc6a088c76b7002
amzn-s3-demo-bucket1    example-object-4    EXAMPLEDC8l.XJCENlF7LePaNIIvs004    TRUE
      100    2021-02-15T15:28:27.000Z    EXAMPLE21e1518b92f3d92773570f603    STANDARD
  FALSE    COMPLETED    SSE-KMS    2025-01-25T15:28:26.000Z    COMPLIANCE    Off
 ENABLED
 eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IkFWQUlMQUJMRSIsImdyYW50cyI6W3sicGVybWlzc2lvbiI6Ik
    EXAMPLE766e8f6b115d93d41df2eac420aa4a465d177c1398bc6a088c76b7003
```

Wir empfehlen, eine Lebenszyklusrichtlinie einzurichten, die alte Bestandslisten löscht. Weitere Informationen finden Sie unter Verwaltung des Lebenszyklus von Objekten.

Die `s3:PutInventoryConfiguration`-Berechtigung ermöglicht es einem Benutzer, beim Konfigurieren einer Bestandsliste sowohl alle Metadatenfelder auszuwählen, die zuvor für jedes Objekt aufgelistet wurden, als auch den Ziel-Bucket zum Speichern des Bestands anzugeben. Ein Benutzer mit Lesezugriff auf Objekte im Ziel-Bucket kann auf alle Objektmetadatenfelder zugreifen, die in der Bestandsliste verfügbar sind. Informationen zum Einschränken des Zugriffs auf Bestandsberichte finden Sie unter Gewähren von Berechtigungen für S3 Inventory und S3 Analytics.

Bestandskonsistenz

Möglicherweise erscheinen nicht alle Ihre Objekte in jeder Bestandsliste. Die Bestandsliste bietet letztendliche Datenkonsistenz für PUT-Anforderungen (sowohl neuer als auch überschriebener Objekte) sowie für `DELETE`-Anforderungen. Jede Bestandsliste für einen Bucket ist ein Snapshot der Bucket-Elemente. Diese Listen sind letztendlich konsistent (das heißt, eine Liste enthält möglicherweise keine kürzlich hinzugefügten oder gelöschten Objekte).

Um den Status eines Objekts zu überprüfen, bevor Sie Maßnahmen für das Objekt ergreifen, empfehlen wir Ihnen, eine `HeadObject` REST API Anfrage zum Abrufen von Metadaten für das Objekt durchzuführen oder die Eigenschaften des Objekts in der Amazon S3 S3-Konsole zu überprüfen. Sie können Objektmetadaten auch mit dem AWS CLI oder dem überprüfen AWS SDKS. Weitere Informationen finden Sie `HeadObject`in der Amazon Simple Storage Service API Reference.

Weitere Informationen zur Arbeit mit Amazon S3 Inventory finden Sie in den folgenden Themen.

Themen

- Konfigurieren von Amazon S3 Inventory
- Lokalisieren Ihrer Bestandsliste
- Einrichten von Amazon-S3-Ereignis-Benachrichtigungen für den Bestandsabschluss
- Abfragen von Amazon S3 Inventory mit Amazon Athena
- Konvertieren leerer Versions-ID-Strings in Amazon-S3-Inventory-Berichten in Null-Zeichenfolgen
- Mit dem ACL Objektfeld arbeiten

Konfigurieren von Amazon S3 Inventory

Amazon S3 Inventory stellt eine Flat-File-Liste Ihrer Objekte und Metadaten nach einem von Ihnen definierten Zeitplan bereit. Sie können S3 Inventory als geplante Alternative zum synchronen Amazon S3 `List` API S3-Betrieb verwenden. S3 Inventory bietet kommagetrennte Werte (), CSV Apache optimierte Zeile spaltenförmig () oder ORC Apache Parquet (Parquet)Ausgabedateien, die Ihre Objekte und die entsprechenden Metadaten auflisten.

Sie können S3 Inventory so konfigurieren, dass täglich oder wöchentlich Bestandslisten für einen S3-Bucket oder für Objekte mit gemeinsamem Präfix (Objekte, deren Namen mit derselben Zeichenfolge beginnen) erstellt werden. Weitere Informationen finden Sie unter Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory.

In diesem Abschnitt wird beschrieben, wie Sie einen Bestand, einschließlich Details über die Quell- und Ziel-Buckets des Bestands, einrichten.

Themen

- Übersicht
- Erstellen einer Ziel-Bucket-Richtlinie
- Erteilen der Berechtigung an Amazon S3 zur Verwendung Ihres vom Kunden verwalteten Schlüssels für die Verschlüsselung
- Konfigurieren des Bestands mit der S3-Konsole
- Verwenden von RESTAPI, um mit S3 Inventory zu arbeiten

Übersicht

Amazon S3 Inventory hilft Ihnen, Ihren Speicher zu verwalten, indem nach einem definierten Zeitplan Listen der Objekte in einem S3-Bucket erstellt werden. Sie können mehrere Bestandslisten für einen

Bucket konfigurieren. Die Inventarlisten werden veröffentlicht inCSV,ORC, oder Parquet Dateien in einem Ziel-Bucket.

Der einfachste Weg, ein Inventar einzurichten, ist die Verwendung der Amazon S3-Konsole, aber Sie können auch Amazon S3 RESTAPI, AWS Command Line Interface (AWS CLI) oder verwenden AWS SDKs. Die Konsole führt den ersten Schritt des folgenden Verfahrens für Sie durch: Hinzufügen einer Bucket-Richtlinie zum Ziel-Bucket.

So richten Sie Amazon S3 Inventory für einen S3-Bucket ein

1. Fügen Sie eine Bucket-Richtlinie für den Ziel-Bucket hinzu.

   Sie müssen eine Bucket-Richtlinie für den Ziel-Bucket erstellen, die Amazon S3 die Erlaubnis erteilt, Objekte in den Bucket am definierten Speicherort zu schreiben. Eine Beispielrichtlinie finden Sie unter Gewähren von Berechtigungen für S3 Inventory und S3 Analytics.
2. Konfigurieren Sie eine Bestandsliste, um die Objekte in einem Quell-Bucket aufzulisten und die Liste in einem Ziel-Bucket zu veröffentlichen.

   Wenn Sie eine Bestandsliste für einen Quell-Bucket konfigurieren, geben Sie den Ziel-Bucket an, in dem die Liste gespeichert werden soll, und ob Sie möchten, dass die Liste täglich oder wöchentlich generiert werden soll. Sie können auch konfigurieren, ob alle Objektversionen oder nur aktuelle Versionen aufgeführt werden sollen und welche Objektmetadaten aufgenommen werden sollen.

   Einige Objektmetadatenfelder in Konfigurationen für S3-Inventarberichte sind optional, d. h. sie sind standardmäßig verfügbar, können aber eingeschränkt werden, wenn Sie einem Benutzer die `s3:PutInventoryConfiguration` entsprechende Berechtigung erteilen. Mithilfe des `s3:InventoryAccessibleOptionalFields` Bedingungsschlüssels können Sie steuern, ob Benutzer diese optionalen Metadatenfelder in ihre Berichte aufnehmen können.

   Weitere Informationen zu den optionalen Metadatenfeldern, die in S3 Inventory verfügbar sind, finden Sie unter OptionalFieldsin der Amazon Simple Storage Service API Reference. Weitere Informationen zur Beschränkung des Zugriffs auf bestimmte optionale Metadatenfelder in einer Inventarkonfiguration finden Sie unterSteuern Sie die Erstellung der Konfiguration des S3-Inventarberichts.

   Sie können angeben, dass die Inventarlistendatei verschlüsselt werden soll, indem Sie serverseitige Verschlüsselung mit einem von Amazon S3 verwalteten Schlüssel (SSE-S3) oder

einem AWS Key Management Service (AWS KMS) vom Kunden verwalteten Schlüssel (SSE-KMS) verwenden.

> **Note**
>
> Von AWS verwalteter Schlüssel (`aws/s3`) wird für die KMS Verschlüsselung mit S3 SSE Inventory nicht unterstützt.

Weitere Hinweise zu SSE -S3 und SSE - finden Sie KMS unterSchützen von Daten mit serverseitiger Verschlüsselung. Wenn Sie beabsichtigen, die SSE KMS -Verschlüsselung zu verwenden, lesen Sie Schritt 3.

- Informationen zur Verwendung der Konsole zum Konfigurieren einer Bestandsliste finden Sie unter Konfigurieren des Bestands mit der S3-Konsole.
- Um Amazon S3 API zur Konfiguration einer Inventarliste zu verwenden, verwenden Sie den PutBucketInventoryConfigurationRESTAPIOperation oder das Äquivalent von AWS CLI oder AWS SDKs.

3. Um die Inventarlistendatei mit SSE - zu verschlüsselnKMS, gewähren Sie Amazon S3 die Erlaubnis, den AWS KMS key zu verwenden.

   Sie können die Verschlüsselung für die Inventarlistendatei mithilfe der Amazon S3-Konsole, Amazon S3 REST API AWS CLI, oder konfigurieren AWS SDKs. Unabhängig davon, welche Möglichkeit Sie wählen: Sie müssen Amazon S3 die Berechtigung gewähren, den vom Kunden verwalteten Schlüssel zum Verschlüsseln der Bestandsdatei zu verwenden. Sie erteilen Amazon S3 die Berechtigung, indem Sie die Schlüsselrichtlinie für den kundenverwalteten Schlüssel ändern, den Sie zur Verschlüsselung der Bestandsdatei verwenden möchten. Weitere Informationen finden Sie unter Erteilen der Berechtigung an Amazon S3 zur Verwendung Ihres vom Kunden verwalteten Schlüssels für die Verschlüsselung.

   Der Ziel-Bucket, in dem die Bestandslistendatei gespeichert wird, kann zu einem anderen AWS-Konto als zu dem Konto gehören, zu dem der Quell-Bucket gehört. Wenn Sie SSE — KMS Verschlüsselung für den kontoübergreifenden Betrieb von Amazon S3 S3-Inventory verwenden, empfehlen wir Ihnen, ARN bei der Konfiguration des S3-Inventars einen vollqualifizierten KMS Schlüssel zu verwenden. Weitere Informationen finden Sie unter Verwendung von SSE — KMS Verschlüsselung für kontoübergreifende Operationen und ServerSideEncryptionByDefaultin der Amazon Simple Storage Service API Reference.

Erstellen einer Ziel-Bucket-Richtlinie

Wenn Sie Ihre Bestandskonfiguration über die S3-Konsole erstellen, erstellt Amazon S3 automatisch eine Bucket-Richtlinie für den Ziel-Bucket, die ihm Amazon-S3-Schreibberechtigung gewährt. Wenn Sie Ihre Inventarkonfiguration jedoch über AWS CLI, oder Amazon S3 REST API erstellen AWS SDKs, müssen Sie dem Ziel-Bucket manuell eine Bucket-Richtlinie hinzufügen. Die Richtlinie für S3-Inventar-Ziel-Buckets ermöglicht es Amazon S3, Daten für die Inventarberichte in den Bucket zu schreiben.

Im Folgenden finden Sie ein Beispiel für eine Bucket-Richtlinie.

```
{
      "Version": "2012-10-17",
      "Statement": [
        {
            "Sid": "InventoryExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-DESTINATION-BUCKET/*"
            ],
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:s3:::DOC-EXAMPLE-SOURCE-BUCKET"
                },
                "StringEquals": {
                    "aws:SourceAccount": "source-account-id",
                    "s3:x-amz-acl": "bucket-owner-full-control"
                }
            }
        }
    ]
}
```

Weitere Informationen finden Sie unter Gewähren von Berechtigungen für S3 Inventory und S3 Analytics.

Wenn beim Erstellen der Bucket-Richtlinie ein Problem auftritt, erhalten Sie Anweisungen zur Fehlerbehebung. Wenn Sie beispielsweise einen Ziel-Bucket in einem anderen auswählen AWS-

Konto und keine Lese- und Schreibberechtigungen für die Bucket-Richtlinie haben, wird eine Fehlermeldung angezeigt.

In diesem Fall muss der Besitzer des Ziel-Buckets die Bucket-Richtlinie zum Ziel-Bucket hinzufügen. Wird die Richtlinie dem Ziel-Bucket nicht hinzugefügt, erhalten Sie keinen Bestandsbericht, da Amazon S3 nicht über die Berechtigung verfügt, in den Ziel-Bucket zu schreiben. Wenn der Quell-Bucket nicht dem Konto des aktuellen Benutzers gehört, muss die richtige Konto-ID des Quell-Bucket-Eigentümers in der Richtlinie ersetzt werden.

> **Note**
>
> Stellen Sie sicher, dass der Ziel-Bucket-Richtlinie keine Deny-Anweisungen hinzugefügt wurden, die die Übermittlung von Inventarberichten in diesen Bucket verhindern würden. Weitere Informationen finden Sie unter Warum kann ich keinen Amazon S3 S3-Inventarbericht erstellen? .

Erteilen der Berechtigung an Amazon S3 zur Verwendung Ihres vom Kunden verwalteten Schlüssels für die Verschlüsselung

Um Amazon S3 die Erlaubnis zu erteilen, Ihren AWS Key Management Service (AWS KMS) vom Kunden verwalteten Schlüssel für die serverseitige Verschlüsselung zu verwenden, müssen Sie eine Schlüsselrichtlinie verwenden. Gehen Sie wie folgt vor, um Ihre Schlüsselrichtlinie zu aktualisieren, damit Sie einen vom Kunden verwalteten Schlüssel verwenden können.

Um Amazon S3 Berechtigungen zur Verschlüsselung mithilfe Ihres vom Kunden verwalteten Schlüssels zu gewähren

1. Melden Sie AWS-Konto sich mit dem Schlüssel, der dem Kunden gehört, bei der AWS Management Console an.
2. Öffnen Sie die AWS KMS Konsole unter https://console.aws.amazon.com/kms.
3. Um das zu ändern AWS-Region, verwenden Sie die Regionsauswahl in der oberen rechten Ecke der Seite.
4. Klicken Sie im linken Navigationsbereich auf Customer managed keys (Vom Kunden verwaltete Schlüssel).
5. Wählen Sie unter Kundenverwaltete Schlüssel den vom Kunden verwalteten Schlüssel aus, den Sie zum Verschlüsseln der Bestandsdateien verwenden möchten.

6. Wählen Sie im Abschnitt Key policy (Schlüsselrichtlinie) die Option Switch to policy view (Zur Richtlinienansicht wechseln) aus.
7. Um die Schlüsselrichtlinie zu aktualisieren, wählen Sie Edit (Bearbeiten).
8. Fügen Sie auf der Seite Schlüsselrichtlinie bearbeiten die folgenden Zeilen zu der vorhandenen Schlüsselrichtlinie hinzu. Geben Sie für *source-account-id* und *amzn-s3-demo-source-bucket* die entsprechenden Werte für Ihren Anwendungsfall an.

```
{
    "Sid": "Allow Amazon S3 use of the customer managed key",
    "Effect": "Allow",
    "Principal": {
        "Service": "s3.amazonaws.com"
    },
    "Action": [
        "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition":{
      "StringEquals":{
         "aws:SourceAccount":"source-account-id"
     },
      "ArnLike":{
        "aws:SourceARN": "arn:aws:s3:::amzn-s3-demo-source-bucket"
     }
   }
}
```

9. Wählen Sie Änderungen speichern.

Weitere Informationen zum Erstellen von kundenverwalteten Schlüsseln in und zum Verwenden von Schlüsselrichtlinien finden Sie unter den folgenden Links im AWS Key Management Service Benutzerhandbuch:

- Schlüssel verwalten
- Wichtige Richtlinien in AWS KMS

> **Note**
>
> Stellen Sie sicher, dass der Ziel-Bucket-Richtlinie keine Deny-Anweisungen hinzugefügt wurden, die die Übermittlung von Inventarberichten in diesen Bucket verhindern würden. Weitere Informationen finden Sie unter Warum kann ich keinen Amazon S3 S3-Inventarbericht erstellen? .

Konfigurieren des Bestands mit der S3-Konsole

Verwenden Sie diese Anweisungen, um den Bestand mit der S3-Konsole zu konfigurieren.

> **Note**
>
> Es könnte bis zu 48 Stunden dauern, bis Amazon S3 den ersten Inventarbericht bereitstellt.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus. Wählen Sie in der Liste Bucket den Namen des Buckets aus, für den Sie Amazon S3 Inventory konfigurieren wollen.
3. Wählen Sie den Tab Management.
4. Wählen Sie unter Inventory configurations (Bestands-Konfigurationen) die Option Create inventory configuration (Bestands-Konfiguration erstellen).
5. Geben Sie unter Name der Bestandskonfiguration einen Namen ein.
6. Gehen Sie für Umfang für den Bestand wie folgt vor:
   - Geben Sie ein optionales Präfix ein.
   - Wählen Sie aus, welche Objektversionen eingeschlossen werden sollen: Nur aktuelle Versionen oder Alle Versionen einschließen.
7. Wählen Sie unter Berichtsdetails den Speicherort des AWS-Konto aus, in dem Sie die Berichte speichern möchten: Dieses Konto oder ein anderes Konto.
8. Wählen Sie unter Destination (Ziel) den Ziel-Bucket aus, in dem die Bestandsberichte gespeichert werden sollen.

Der Ziel-Bucket muss sich im selben Bucket befinden AWS-Region wie der Bucket, für den Sie das Inventar einrichten. Der Ziel-Bucket kann sich in einem anderem AWS-Konto befinden. Beim Festlegen des Ziel-Buckets können Sie auch ein optionales Präfix angeben, um Ihre Bestandsberichte zu gruppieren.

Unter dem Bucket-Feld Destination (Ziel) sehen Sie die Anweisung Ziel-Bucket-Berechtigung, die der Ziel-Bucket-Richtlinie hinzugefügt wird, damit Amazon S3 Daten in diesen Bucket platzieren kann. Weitere Informationen finden Sie unter Erstellen einer Ziel-Bucket-Richtlinie.

9. Wählen Sie unter Häufigkeit aus, wie oft der Bericht erstellt wird: Täglich oder Wöchentlich.
10. Wählen Sie als Ausgabeformat eines der folgenden Formate für den Bericht aus:

    - CSV— Wenn Sie diesen Inventarbericht mit S3 Batch Operations verwenden möchten oder wenn Sie diesen Bericht in einem anderen Tool wie Microsoft Excel analysieren möchten, wählen Sie CSV.
    - Apache ORC
    - Apache Parquet
11. Wählen Sie unter Status die Option Enable (Aktivieren) oder Disable (Deaktivieren) aus.
12. Führen Sie unter Verschlüsselung des Bestandsberichts die folgenden Schritte aus, um die serverseitige Verschlüsselung zu konfigurieren:

    a. Wählen Sie unter Serverseitige Verschlüsselung entweder Keinen Verschlüsselungsschlüssel angeben oder Einen Verschlüsselungsschlüssel zum Verschlüsseln von Daten angeben aus.

        - Um die Bucket-Einstellungen für die serverseitige Standardverschlüsselung von Objekten beizubehalten, wenn sie in Amazon S3 gespeichert werden, wählen Sie Keinen Verschlüsselungsschlüssel angeben aus. Solange S3-Bucket-Schlüssel für den Ziel-Bucket aktiviert sind, wendet der Kopiervorgang S3-Bucket-Schlüssel auf den Ziel-Bucket an.

        > **Note**
        >
        > Wenn die Bucket-Richtlinie für das angegebene Ziel vorschreibt, dass Objekte verschlüsselt werden müssen, bevor sie in Amazon S3 gespeichert werden, müssen Sie „Verschlüsselungsschlüssel angeben“ wählen. Andernfalls schlägt das Kopieren von Objekten in das Ziel fehl.

- Um Objekte zu verschlüsseln, bevor sie in Amazon S3 gespeichert werden, wählen Sie Verschlüsselungsschlüssel angeben aus.

b. Wenn Sie unter Verschlüsselungstyp die Option Verschlüsselungsschlüssel angeben ausgewählt haben, müssen Sie entweder den von Amazon S3 verwalteten Schlüssel (SSE-S3) oder den AWS Key Management Service Schlüssel (SSE-KMS) auswählen.

SSE-S3 verwendet eine der stärksten Blockchiffren — den 256-Bit-Advanced Encryption Standard (-256), um jedes Objekt zu verschlüsseln. AES SSE- bietet Ihnen mehr Kontrolle KMS über Ihren Schlüssel. Weitere Hinweise zu SSE -S3 finden Sie unterServerseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln verwenden (SSE-S3). Weitere Hinweise zu SSE - finden Sie KMS unterServerseitige Verschlüsselung mit AWS KMS Schlüsseln verwenden (SSE-) KMS.

> **Note**
>
> Um die Inventarlistendatei mit SSE - zu verschlüsselnKMS, müssen Sie Amazon S3 die Erlaubnis erteilen, den vom Kunden verwalteten Schlüssel zu verwenden. Eine Anleitung finden Sie unter Amazon S3 Permission to Encrypt Using Your KMS Keys gewähren.

c. Wenn Sie unter AWS Key Management Service Schlüssel (SSE-KMS) ausgewählt haben AWS KMS key, können Sie Ihren AWS KMS Schlüssel über eine der folgenden Optionen angeben.

> **Note**
>
> Wenn der Ziel-Bucket, in dem die Inventarlistendatei gespeichert ist, einem anderen gehört AWS-Konto, stellen Sie sicher, dass Sie einen vollqualifizierten KMS Schlüssel verwendenARN, um Ihren KMS Schlüssel anzugeben.

- Um aus einer Liste verfügbarer KMS Schlüssel auszuwählen, wählen Sie Aus Ihren AWS KMS Schlüsseln auswählen und wählen Sie einen symmetrischen KMS Verschlüsselungsschlüssel aus der Liste der verfügbaren Schlüssel aus. Stellen Sie sicher, dass sich der KMS Schlüssel in derselben Region wie Ihr Bucket befindet.

> **Note**
>
> Sowohl der Von AWS verwalteter Schlüssel (`aws/s3`) als auch Ihr vom Kunden verwalteter Schlüssel werden in der Liste angezeigt. Von AWS verwalteter Schlüssel (`aws/s3`) wird jedoch nicht für die KMS Verschlüsselung SSE mit S3 Inventory unterstützt.

- Um den KMS Schlüssel einzugebenARN, wählen Sie AWS KMS Schlüssel ARN eingeben und geben Sie Ihren KMS Schlüssel ARN in das angezeigte Feld ein.
- Um einen neuen, vom Kunden verwalteten Schlüssel in der AWS KMS Konsole zu erstellen, wählen Sie KMS Schlüssel erstellen.

13. Wählen Sie für Zusätzliche Metadatenfelder eine oder mehrere der folgenden Optionen für das Hinzufügen zum Bestandsbericht aus:

    - Größe – Die Objektgröße in Byte, ohne die Größe von unvollständigen mehrteiligen Uploads, Objektmetadaten und Löschmarkierungen.
    - Last modified date (Letztes Änderungsdatum) – Datum der Erstellung oder der letzten Änderung des Objekts, je nachdem, welches neuer ist.
    - Multipart upload (Mehrteiliger Upload) – Gibt an, dass das Objekt als mehrteiliger Upload hochgeladen wurde. Weitere Informationen finden Sie unter Hochladen und Kopieren von Objekten mit mehrteiligen Uploads.
    - Replication status (Replikationsstatus) – Der Replikationsstatus des Objekts. Weitere Informationen finden Sie unter Abrufen von Replikationsstatusinformationen.
    - Encryption status (Verschlüsselungsstatus) – Der serverseitige Verschlüsselungstyp, der für die Verschlüsselung des Objekts verwendet wird. Weitere Informationen finden Sie unter Schützen von Daten mit serverseitiger Verschlüsselung.
    - Bucket-Key-Status — Gibt an, ob ein von generierter Schlüssel auf Bucket-Ebene für das Objekt AWS KMS gilt. Weitere Informationen finden Sie unter Senkung der Kosten von SSE — KMS mit Amazon S3 Bucket Keys.
    - Objektzugriffskontrollliste — Eine Zugriffskontrollliste (ACL) für jedes Objekt, die definiert, welchen AWS-Konten oder welchen Gruppen Zugriff auf dieses Objekt gewährt wird und welche Art von Zugriff gewährt wird. Weitere Hinweise zu diesem Feld finden Sie unter Mit dem ACL Objektfeld arbeiten. Weitere Informationen zu finden ACLs Sie unterÜbersicht über die Zugriffskontrollliste (ACL).

- Object owner (Besitzer des Objekts) – Der Besitzer des Objekts.
- Storage class (Speicherklasse) – Die für die Speicherung des Objekts verwendete Speicherklasse.
- Intelligent-Tiering: Access tier (Zugriffsebene) – Zugriffsebene (häufig oder selten) des Objekts, wenn es im S3-Intelligent-Tiering-Speicher gespeichert wurde. Weitere Informationen finden Sie unter Speicherklasse zur automatischen Optimierung von Daten mit sich ändernden oder unbekannten Zugriffsmustern.
- ETag— Das Entity-Tag (ETag) ist ein Hash des Objekts. Das ETag spiegelt nur Änderungen am Inhalt eines Objekts wider, nicht an seinen Metadaten. Dies ETag kann eine MD5 Zusammenfassung der Objektdaten sein oder auch nicht. Dies hängt davon ab, wie das Objekt erstellt und verschlüsselt wurde. Weitere Informationen finden Sie unter Objectin der Amazon Simple Storage Service API Reference.
- Prüfsummen-Algorithmus – Gibt den Algorithmus an, mit dem die Prüfsumme für das Objekt erstellt wurde. Weitere Informationen finden Sie unter Verwenden unterstützter Prüfsummenalgorithmen.
- All Object lock configurations (Alle Objektsperre-Konfigurationen) – Der Objektsperrenstatus des Objekts, einschließlich der folgenden Einstellungen:
  - Object Lock: Retention mode (Aufbewahrungsmodus der Objektsperre) – Die auf das Objekt angewendete Schutzebene, entweder Governance oder Compliance.
  - Object Lock: Retain until date (Beibehaltungsfrist für Objektsperre) – Das Datum, bis zu dem das gesperrte Objekt nicht gelöscht werden kann.
  - Object Lock: Legal hold status (Status der gesetzlichen Sperrfrist des Objekts) – Der Status der gesetzlichen Sperrfrist des gesperrten Objekts.

  Weitere Informationen zur S3-Objektsperre finden Sie unter So funktioniert die S3-Objektsperre.

Weitere Informationen zum Inhalt eines Bestandsberichts finden Sie unter Amazon-S3-Inventory-Liste.

Weitere Informationen zur Beschränkung des Zugriffs auf bestimmte optionale Metadatenfelder in einer Inventarkonfiguration finden Sie unterSteuern Sie die Erstellung der Konfiguration des S3-Inventarberichts.

14. Wählen Sie Create (Erstellen) aus.

Verwenden von RESTAPI, um mit S3 Inventory zu arbeiten

Im Folgenden sind die REST Operationen aufgeführt, die Sie verwenden können, um mit Amazon S3 Inventory zu arbeiten.

- DeleteBucketInventoryConfiguration
- GetBucketInventoryConfiguration
- ListBucketInventoryConfigurations
- PutBucketInventoryConfiguration

Lokalisieren Ihrer Bestandsliste

Wenn eine Bestandsliste veröffentlicht wird, werden die Manifestdateien am folgenden Standort im Ziel-Bucket veröffentlicht.

```
destination-prefix/amzn-s3-demo-source-bucket/config-ID/YYYY-MM-DDTHH-MMZ/manifest.json
 destination-prefix/amzn-s3-demo-source-bucket/config-ID/YYYY-MM-DDTHH-MMZ/
manifest.checksum
 destination-prefix/amzn-s3-demo-source-bucket/config-ID/hive/dt=YYYY-MM-DD-HH-MM/
symlink.txt
```

- *destination-prefix* ist das Präfix des Objektschlüsselnamens, das optional in der Bestandskonfiguration festgelegt wird. Sie können dieses Präfix verwenden, um alle Bestandslisten-Dateien an einem gemeinsamen Standort im Ziel-Bucket zu gruppieren.
- *amzn-s3-demo-source-bucket* ist der Quell-Bucket, für den die Bestandsliste erstellt wird. Der Quell-Bucket-Name wird hinzugefügt, um Kollisionen zu vermeiden, wenn mehrere Bestandsberichte von mehreren Quell-Buckets an denselben Ziel-Bucket gesendet werden.
- *config-ID* wird hinzugefügt, um Kollisionen zu vermeiden, wenn mehrere Bestandsberichte vom selben Quell-Bucket an denselben Ziel-Bucket gesendet werden. Die *config-ID* kommt aus der Bestandsbericht-Konfiguration und ist der Name des Berichts, der bei der Einrichtung festgelegt wurde.
- *YYYY-MM-DDTHH-MMZ* ist der Zeitstempel, der sich aus der Startzeit und dem Datum zusammensetzt, an dem die Bestandsberichtserstellung mit dem Scannen des Buckets beginnt, z. B. `2016-11-06T21-32Z`.
- `manifest.json` ist die Manifestdatei.
- `manifest.checksum`ist der MD5 Hash des Inhalts der `manifest.json` Datei.

- `symlink.txt`ist der Apache Hive-kompatible Manifestdatei.

Die Bestandslisten werden täglich oder wöchentlich am folgenden Standort im Ziel-Bucket veröffentlicht.

```
destination-prefix/amzn-s3-demo-source-bucket/config-ID/data/example-file-name.csv.gz
...
destination-prefix/amzn-s3-demo-source-bucket/config-ID/data/example-file-name-1.csv.gz
```

- *`destination-prefix`* ist das Präfix des Objektschlüsselnamens, das optional in der Bestandskonfiguration festgelegt wird. Es kann verwendet werden, um alle Bestandslisten-Dateien an einem gemeinsamen Standort innerhalb des Ziel-Buckets zu gruppieren.
- *`amzn-s3-demo-source-bucket`* ist der Quell-Bucket, für den die Bestandsliste erstellt wird. Der Quell-Bucket-Name wird hinzugefügt, um Kollisionen zu vermeiden, wenn mehrere Bestandsberichte von mehreren Quell-Buckets an denselben Ziel-Bucket gesendet werden.
- *`example-file-name`*`.csv.gz`ist eine der CSV Inventardateien. ORCInventarnamen enden mit der Dateinamenerweiterung `.orc` und Parquet Inventarnamen enden mit der Dateinamenerweiterung`.parquet`.

Bestandsmanifest

Die Manifest-Dateien `manifest.json` und `symlink.txt` beschreiben, wo sich die Bestandsdateien befinden. Wenn eine neue Bestandsliste geliefert wird, wird sie durch eine neue Reihe von Manifestdateien begleitet. Diese Dateien könnten sich gegenseitig überschreiben. In versionierungsfähigen Buckets erstellt Amazon S3 neue Versionen der Manifestdateien.

Jedes in der `manifest.json`-Datei enthaltene Manifest bietet Metadaten und andere grundlegende Informationen zu einem Bestand. Diese Informationen beinhalten Folgendes:

- Name des Quell-Buckets
- Name des Ziel-Buckets
- Bestandsversion
- Erstellungszeitstempel im Epochen-Datumsformat, der aus der Startzeit und dem Datum besteht, an dem die Bestandberichtserstellung beginnt, den Bucket zu scannen
- Format und Schema der Bestandsdateien
- Liste der Bestandsdateien, die im Ziel-Bucket enthalten sind

Immer wenn eine `manifest.json` Datei geschrieben wird, wird sie von einer `manifest.checksum` Datei begleitet, die den MD5 Hash des `manifest.json` Dateiinhalts darstellt.

Example Bestandsmanifest in einer **`manifest.json`**-Datei

Die folgenden Beispiele zeigen ein Inventarmanifest in einer `manifest.json` Datei für CSVORC, und Parquet-formatierte Inventare.

CSV

Im Folgenden finden Sie ein Beispiel für ein Manifest in einer `manifest.json` Datei für ein CSV Inventar im -Format.

```
{
    "sourceBucket": "amzn-s3-demo-source-bucket",
    "destinationBucket": "arn:aws:s3:::example-inventory-destination-bucket",
    "version": "2016-11-30",
    "creationTimestamp" : "1514944800000",
    "fileFormat": "CSV",
    "fileSchema": "Bucket, Key, VersionId, IsLatest, IsDeleteMarker,
 Size, LastModifiedDate, ETag, StorageClass, IsMultipartUploaded,
 ReplicationStatus, EncryptionStatus, ObjectLockRetainUntilDate, ObjectLockMode,
 ObjectLockLegalHoldStatus, IntelligentTieringAccessTier, BucketKeyStatus,
 ChecksumAlgorithm, ObjectAccessControlList, ObjectOwner",
    "files": [
        {
            "key": "Inventory/amzn-s3-demo-source-bucket/2016-11-06T21-32Z/
files/939c6d46-85a9-4ba8-87bd-9db705a579ce.csv.gz",
            "size": 2147483647,
            "MD5checksum": "f11166069f1990abeb9c97ace9cdfabc"
        }
    ]
}
```

ORC

Das Folgende ist ein Beispiel für ein Manifest in einer `manifest.json` Datei für ein Inventar im ORC -Format.

```
{
    "sourceBucket": "amzn-s3-demo-source-bucket",
    "destinationBucket": "arn:aws:s3:::example-destination-bucket",
```

```
    "version": "2016-11-30",
    "creationTimestamp" : "1514944800000",
    "fileFormat": "ORC",
    "fileSchema":
 "struct<bucket:string,key:string,version_id:string,is_latest:boolean,is_delete_marker:boole
    "files": [
        {
            "key": "inventory/amzn-s3-demo-source-bucket/data/
d794c570-95bb-4271-9128-26023c8b4900.orc",
            "size": 56291,
            "MD5checksum": "5925f4e78e1695c2d020b9f6eexample"
        }
    ]
}
```

Parquet

Das Folgende ist ein Beispiel für ein Manifest in einer `manifest.json` Datei für Parquet-formatiertes Inventar.

```
{
    "sourceBucket": "amzn-s3-demo-source-bucket",
    "destinationBucket": "arn:aws:s3:::example-destination-bucket",
    "version": "2016-11-30",
    "creationTimestamp" : "1514944800000",
    "fileFormat": "Parquet",
    "fileSchema": "message s3.inventory { required binary bucket (UTF8);
 required binary key (UTF8); optional binary version_id (UTF8); optional boolean
 is_latest; optional boolean is_delete_marker; optional int64 size; optional
 int64 last_modified_date (TIMESTAMP_MILLIS); optional binary e_tag (UTF8);
 optional binary storage_class (UTF8); optional boolean is_multipart_uploaded;
 optional binary replication_status (UTF8); optional binary encryption_status
 (UTF8); optional int64 object_lock_retain_until_date (TIMESTAMP_MILLIS); optional
 binary object_lock_mode (UTF8); optional binary object_lock_legal_hold_status
 (UTF8); optional binary intelligent_tiering_access_tier (UTF8); optional binary
 bucket_key_status (UTF8); optional binary checksum_algorithm (UTF8); optional
 binary object_access_control_list (UTF8); optional binary object_owner (UTF8);}",
    "files": [
        {
           "key": "inventory/amzn-s3-demo-source-bucket/data/
d754c470-85bb-4255-9218-47023c8b4910.parquet",
            "size": 56291,
            "MD5checksum": "5825f2e18e1695c2d030b9f6eexample"
```

```
        }
    ]
}
```

Die `symlink.txt` Datei ist eine Apache Hive-kompatible Manifest-Datei, die Folgendes ermöglicht Hive um Inventardateien und die zugehörigen Datendateien automatisch zu erkennen. Das Tool Hive-kompatibles Manifest funktioniert mit Hive-kompatible Dienste Athena und Amazon Redshift Spectrum. Es funktioniert auch mit Hive-kompatible Anwendungen, einschließlich Presto, Apache Hive, Apache Sparkund viele andere.

> **Important**
>
> Die `symlink.txt` Apache Hive-kompatible Manifestdatei funktioniert derzeit nicht mit AWS Glue.
> Die `symlink.txt` Datei wird gelesen mit Apache Hive und Apache Sparkwird nicht unterstützt für ORC und Parquet-formatierte Inventardateien.

Einrichten von Amazon-S3-Ereignis-Benachrichtigungen für den Bestandsabschluss

Sie können eine Amazon S3-Ereignis-Benachrichtigung einrichten, um informiert zu werden, wenn die Prüfsummendatei für das Manifest erstellt wird, woran zu erkennen ist, dass dem Ziel-Bucket eine Bestandsliste hinzugefügt wurde. Das Manifest ist eine up-to-date Liste aller Inventarlisten am Zielort.

Amazon S3 kann Ereignisse in einem Amazon Simple Notification Service (AmazonSNS) -Thema, einer Amazon Simple Queue Service (AmazonSQS) -Warteschlange oder einer AWS Lambda Funktion veröffentlichen. Weitere Informationen finden Sie unter Amazon-S3-Ereignis-Benachrichtigungen.

Die folgende Benachrichtigungskonfiguration definiert, dass alle `manifest.checksum`-Dateien, die dem Ziel-Bucket hinzugefügt werden, von der AWS Lambda `cloud-function-list-write`. verarbeitet werden.

```
<NotificationConfiguration>
  <QueueConfiguration>
      <Id>1</Id>
      <Filter>
```

```
          <S3Key>
              <FilterRule>
                  <Name>prefix</Name>
                  <Value>destination-prefix/source-bucket</Value>
              </FilterRule>
              <FilterRule>
                  <Name>suffix</Name>
                  <Value>checksum</Value>
              </FilterRule>
          </S3Key>
      </Filter>
      <CloudFunction>arn:aws:lambda:us-west-2:222233334444:cloud-function-list-write</
CloudFunction>
      <Event>s3:ObjectCreated:*</Event>
    </QueueConfiguration>
    </NotificationConfiguration>
```

Weitere Informationen finden Sie unter Using AWS Lambda with Amazon S3 im AWS Lambda Developer Guide.

Abfragen von Amazon S3 Inventory mit Amazon Athena

Sie können Amazon S3 S3-Inventardateien mit SQL Standardabfragen abfragen, indem Sie Amazon Athena in allen Regionen verwenden, in denen Athena verfügbar ist. Informationen zur AWS-Region Verfügbarkeit finden Sie in der AWS-Region Tabelle.

Athena kann Amazon S3 S3-Inventardateien abfragen in Apache optimierte Zeile spaltenförmig ()ORC, Apache Parquet, oder Format für kommagetrennte Werte ()CSV. Wenn Sie Athena verwenden, um Inventardateien abzufragen, empfehlen wir die Verwendung von ORC -formatted oder Parquet-formatierte Inventardateien. Das und ORC Parquet Formate bieten eine schnellere Abfrageleistung und niedrigere Abfragekosten. ORCund Parquet sind selbsterklärende, typspezifische spaltenorientierte Dateiformate, die für Apache Hadoop. Das Spaltenformat ermöglicht es dem Leser, nur die Spalten zu lesen, zu dekomprimieren und zu verarbeiten, die für die aktuelle Abfrage erforderlich sind. Das und ORC Parquet Formate für Amazon S3 Inventory sind in allen verfügbar AWS-Regionen.

So fragen Sie Amazon-S3-Inventory-Dateien mit Athena ab

1. Erstellen Sie eine Athena-Tabelle. Informationen zum Erstellen einer Tabelle finden Sie unter Erstellen von Tabellen in Amazon Athena im Amazon Athena-Benutzerhandbuch.

2. Erstellen Sie Ihre Anfrage anhand einer der folgenden Beispielabfragevorlagen, je nachdem, ob Sie einen ORC Lagerbestandsbericht im -Format, im Parquet-Format oder im -Format abfragen. CSV

   - Wenn Sie Athena verwenden, um einen Inventarbericht im ORC -Format abzufragen, verwenden Sie die folgende Beispielabfrage als Vorlage.

     Die folgende Beispielabfrage umfasst alle optionalen Felder in einem ORC -formatierten Inventarbericht.

     Gehen Sie wie folgt vor, um diese Beispielabfrage zu verwenden:

     - Ersetzen Sie *your_table_name* durch den Namen der Athena-Tabelle, die Sie erstellt haben.
     - Entfernen Sie alle optionalen Felder, die Sie für Ihren Bestand nicht ausgewählt haben, sodass die Abfrage den für Ihren Bestand gewählten Feldern entspricht.
     - Ersetzen Sie den folgenden Bucket-Namen und den Bestandsort (die Konfigurations-ID) entsprechend Ihrer Konfiguration.

       `s3://`*amzn-s3-demo-bucket*`/`*config-ID*`/hive/`

     - Ersetzen Sie das Datum *2022-01-01-00-00* unter `projection.dt.range` mit dem ersten Tag des Zeitbereichs, innerhalb dessen Sie die Daten in Athena partitionieren. Weitere Informationen finden Sie unter Partitionieren von Daten in Athena.

```
CREATE EXTERNAL TABLE your_table_name(
         bucket string,
         key string,
         version_id string,
         is_latest boolean,
         is_delete_marker boolean,
         size bigint,
         last_modified_date timestamp,
         e_tag string,
         storage_class string,
         is_multipart_uploaded boolean,
         replication_status string,
         encryption_status string,
         object_lock_retain_until_date bigint,
         object_lock_mode string,
         object_lock_legal_hold_status string,
         intelligent_tiering_access_tier string,
```

```
         bucket_key_status string,
         checksum_algorithm string,
         object_access_control_list string,
         object_owner string
) PARTITIONED BY (
        dt string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.orc.OrcSerde'
  STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
  OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat'
  LOCATION 's3://source-bucket/config-ID/hive/'
  TBLPROPERTIES (
    "projection.enabled" = "true",
    "projection.dt.type" = "date",
    "projection.dt.format" = "yyyy-MM-dd-HH-mm",
    "projection.dt.range" = "2022-01-01-00-00,NOW",
    "projection.dt.interval" = "1",
    "projection.dt.interval.unit" = "HOURS"
  );
```

- Wenn Sie Athena verwenden, um eine abzufragen Parquet-formatierter Inventarbericht, verwenden Sie die Beispielabfrage für einen ORC -formatierten Bericht. Verwenden Sie jedoch Folgendes Parquet SerDe anstelle von ORC SerDe in der `ROW FORMAT SERDE` Erklärung.

  ```
  ROW FORMAT SERDE 'org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'
  ```

- Wenn Sie Athena verwenden, um einen Inventarbericht im CSV -Format abzufragen, verwenden Sie die folgende Beispielabfrage als Vorlage.

  Die folgende Beispielabfrage umfasst alle optionalen Felder in einem CSV -formatierten Inventarbericht.

  Gehen Sie wie folgt vor, um diese Beispielabfrage zu verwenden:

  - Ersetzen Sie *`your_table_name`* durch den Namen der Athena-Tabelle, die Sie erstellt haben.
  - Entfernen Sie alle optionalen Felder, die Sie für Ihren Bestand nicht ausgewählt haben, sodass die Abfrage den für Ihren Bestand gewählten Feldern entspricht.
  - Ersetzen Sie den folgenden Bucket-Namen und den Bestandsort (die Konfigurations-ID) entsprechend Ihrer Konfiguration.

    `s3://amzn-s3-demo-bucket/config-ID/hive/`

- Ersetzen Sie das Datum *2022-01-01-00-00* unter `projection.dt.range` mit dem ersten Tag des Zeitbereichs, innerhalb dessen Sie die Daten in Athena partitionieren. Weitere Informationen finden Sie unter Partitionieren von Daten in Athena.

```
CREATE EXTERNAL TABLE your_table_name(
         bucket string,
         key string,
         version_id string,
         is_latest boolean,
         is_delete_marker boolean,
         size string,
         last_modified_date string,
         e_tag string,
         storage_class string,
         is_multipart_uploaded boolean,
         replication_status string,
         encryption_status string,
         object_lock_retain_until_date string,
         object_lock_mode string,
         object_lock_legal_hold_status string,
         intelligent_tiering_access_tier string,
         bucket_key_status string,
         checksum_algorithm string,
         object_access_control_list string,
         object_owner string
) PARTITIONED BY (
        dt string
)
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
  STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat'
  OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat'
  LOCATION 's3://source-bucket/config-ID/hive/'
  TBLPROPERTIES (
    "projection.enabled" = "true",
    "projection.dt.type" = "date",
    "projection.dt.format" = "yyyy-MM-dd-HH-mm",
    "projection.dt.range" = "2022-01-01-00-00,NOW",
    "projection.dt.interval" = "1",
    "projection.dt.interval.unit" = "HOURS"
  );
```

3. Sie können jetzt verschiedene Bestandsabfragen ausführen, wie in den folgenden Beispielen gezeigt. Ersetzen Sie jeden *user input placeholder* durch Ihre Informationen.

```
# Get a list of the latest inventory report dates available.
SELECT DISTINCT dt FROM your_table_name ORDER BY 1 DESC limit 10;

# Get the encryption status for a provided report date.
SELECT encryption_status, count(*) FROM your_table_name WHERE dt = 'YYYY-MM-DD-HH-
MM' GROUP BY encryption_status;

# Get the encryption status for inventory report dates in the provided range.
SELECT dt, encryption_status, count(*) FROM your_table_name
WHERE dt > 'YYYY-MM-DD-HH-MM' AND dt < 'YYYY-MM-DD-HH-MM' GROUP BY dt,
 encryption_status;
```

Wenn Sie S3 Inventory so konfigurieren, dass das Feld Object Access Control List (ObjectACL) zu einem Inventarbericht hinzugefügt wird, zeigt der Bericht den Wert für das ACL Objektfeld als Base64-kodierte Zeichenfolge an. Um den dekodierten Wert JSON für das ACL Objektfeld zu erhalten, können Sie dieses Feld mit Athena abfragen. Sehen Sie sich die folgenden Abfragebeispiele an. Weitere Informationen zum ACL Objektfeld finden Sie unter. Mit dem ACL Objektfeld arbeiten

```
# Get the S3 keys that have Object ACL grants with public access.
WITH grants AS (
    SELECT key,
        CAST(
            json_extract(from_utf8(from_base64(object_access_control_list)),
 '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))
        ) AS grants_array
    FROM your_table_name
)
SELECT key,
       grants_array,
       grant
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE element_at(grant, 'uri') = 'http://acs.amazonaws.com/groups/global/AllUsers'
```

```
# Get the S3 keys that have Object ACL grantees in addition to the object owner.
WITH grants AS
    (SELECT key,
    from_utf8(from_base64(object_access_control_list)) AS
 object_access_control_list,
```

```
         object_owner,
         CAST(json_extract(from_utf8(from_base64(object_access_control_list)),
         '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))) AS grants_array
    FROM your_table_name)
SELECT key,
       grant,
       objectowner
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE cardinality(grants_array) > 1 AND element_at(grant, 'canonicalId') !=
 object_owner;
```

```
# Get the S3 keys with READ permission that is granted in the Object ACL.
WITH grants AS (
    SELECT key,
        CAST(
            json_extract(from_utf8(from_base64(object_access_control_list)),
 '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))
        ) AS grants_array
    FROM your_table_name
)
SELECT key,
       grants_array,
       grant
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE element_at(grant, 'permission') = 'READ';
```

```
# Get the S3 keys that have Object ACL grants to a specific canonical user ID.
WITH grants AS (
    SELECT key,
        CAST(
            json_extract(from_utf8(from_base64(object_access_control_list)),
 '$.grants') AS ARRAY(MAP(VARCHAR, VARCHAR))
        ) AS grants_array
    FROM your_table_name
)
SELECT key,
       grants_array,
       grant
FROM grants, UNNEST(grants_array) AS t(grant)
WHERE element_at(grant, 'canonicalId') = 'user-canonical-id';
```

```
# Get the number of grantees on the Object ACL.
SELECT key,
       object_access_control_list,
       json_array_length(json_extract(object_access_control_list,'$.grants')) AS
 grants_count
FROM your_table_name;
```

Weitere Informationen zur Verwendung von Athena finden Sie im Amazon Athena-Benutzerhandbuch.

Konvertieren leerer Versions-ID-Strings in Amazon-S3-Inventory-Berichten in Null-Zeichenfolgen

> **Note**
>
> Das folgende Verfahren gilt nur für Amazon S3-Lagerbestandsberichte, die alle Versionen enthalten, und nur, wenn die Berichte „Alle Versionen“ als Manifeste für S3-Batch-Operationen bei Buckets verwendet werden, bei denen S3-Versionierung aktiviert ist. Sie müssen keine Zeichenfolgen für S3-Inventory-Berichte konvertieren, die nur die aktuelle Version angeben.

Sie können S3-Inventory-Berichte als Manifeste für S3-Batchvorgänge verwenden. Wenn jedoch die S3-Versionsverwaltung für einen Bucket aktiviert ist, markieren S3-Inventory-Berichte, die alle Versionen enthalten, alle nullversionierten Objekte mit leeren Zeichenfolgen im Versions-ID-Feld. Wenn ein Inventarbericht alle Objektversionen enthältIDs, erkennt Batch Operations `null` Zeichenfolgen als VersionIDs, aber keine leeren Zeichenfolgen.

Wenn ein S3-Batchvorgangauftrag einen S3-Inventory-Bericht „alle Versionen“ als Manifest verwendet, schlägt er alle Aufgaben für Objekte fehl, die eine leere Zeichenfolge im Feld Versions-ID haben. Gehen Sie wie folgt vor, um leere Zeichenfolgen im Versions-ID-Feld des S3-Inventory-Berichts in `null`-Zeichenfolgen für Batchvorgänge zu konvertieren.

Aktualisieren eines Amazon-S3-Inventory-Berichts zur Verwendung mit Batchvorgängen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Navigieren Sie zu Ihrem S3-Inventory-Bericht. Der Inventory-Bericht befindet sich im Ziel-Bucket, den Sie beim Konfigurieren Ihres Inventory-Berichts angegeben haben. Weitere Informationen zum Auffinden von Inventory-Berichten finden Sie unter Lokalisieren Ihrer Bestandsliste.

   a. Wählen Sie den Ziel-Bucket aus.

   b. Wählen Sie den Ordner aus. Der Ordner ist nach dem ursprünglichen Quell-Bucket benannt.

   c. Wählen Sie den nach der Inventory-Konfiguration benannten Ordner aus.

   d. Aktivieren Sie das Kontrollkästchen neben dem Ordner namens Hive aus. Wählen Sie oben auf der Seite Copy S3 aus, URI um das S3 URI für den Ordner zu kopieren.
3. Öffnen Sie die Amazon Athena Athena-Konsole unter https://console.aws.amazon.com/athena/.
4. Wählen Sie im Abfrage-Editor Einstellungen und anschließend Verwalten aus. Wählen Sie auf der Seite Einstellungen verwalten für Standort des Abfrageergebnisses einen S3-Bucket aus, um Ihre Abfrageergebnisse zu speichern.
5. Erstellen Sie im Abfrage-Editor eine Athena-Tabelle, um die Daten mit dem folgenden Befehl im Inventory-Bericht zu speichern. *table_name*Ersetzen Sie es durch einen Namen Ihrer Wahl und fügen Sie in die `LOCATION` Klausel das S3 ein, URI das Sie zuvor kopiert haben. Wählen Sie dann Ausführen aus, um die Abfrage auszuführen.

   ```
   CREATE EXTERNAL TABLE table_name(bucket string, key string,
    version_id string) PARTITIONED BY (dt string)ROW FORMAT SERDE
    'org.apache.hadoop.hive.serde2.OpenCSVSerde'STORED AS INPUTFORMAT
    'org.apache.hadoop.hive.ql.io.SymlinkTextInputFormat' OUTPUTFORMAT
    'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat' LOCATION 'Copied S3 URI';
   ```
6. Um den Abfrage-Editor zu löschen, wählen Sie Löschen. Laden Sie den Inventory-Bericht mit dem folgenden Befehl in die Tabelle. Ersetzen Sie *table_name* durch denjenigen, den Sie im vorherigen Schritt ausgewählt haben. Wählen Sie dann Ausführen aus, um die Abfrage auszuführen.

   ```
   MSCK REPAIR TABLE table_name;
   ```
7. Um den Abfrage-Editor zu löschen, wählen Sie Löschen. Führen Sie die folgende `SELECT` Abfrage aus, um alle Einträge im ursprünglichen Inventarbericht abzurufen und alle leeren

Versionen IDs durch `null` Zeichenfolgen zu ersetzen. Ersetzen Sie *`table_name`* durch das zuvor gewählte, und ersetzen Sie *`YYYY-MM-DD-HH-MM`* in der `WHERE`-Klausel durch das Datum des Inventory-Berichts, auf dem dieses Tool ausgeführt werden soll. Wählen Sie dann Ausführen aus, um die Abfrage auszuführen.

```
SELECT bucket as Bucket, key as Key, CASE WHEN version_id = '' THEN 'null' ELSE
 version_id END as VersionId FROM table_name WHERE dt = 'YYYY-MM-DD-HH-MM';
```

8. Kehren Sie zur Amazon S3 S3-Konsole (https://console.aws.amazon.com/s3/) zurück und navigieren Sie zu dem S3-Bucket, den Sie zuvor für Speicherort des Abfrageergebnisses ausgewählt haben. Im Inneren sollte es eine Reihe von Ordnern geben, die mit dem Datum enden.

   Sie sollten beispielsweise so etwas wie s3://sehen***amzn-s3-demo-bucket*/*query-result-location***/Nicht gespeichert/2021/10/07/. Sie sollten `.csv`-Dateien sehen, die die Ergebnisse der ausgeführten `SELECT`-Abfrage enthalten.

   Wählen Sie die Datei mit dem letzten Änderungsdatum aus. CSV Laden Sie diese Datei für den nächsten Schritt auf Ihren lokalen Computer herunter.

9. Die generierte CSV Datei enthält eine Kopfzeile. Um diese CSV Datei als Eingabe für einen S3 Batch Operations-Job zu verwenden, müssen Sie die Kopfzeile entfernen, da Batch Operations keine Kopfzeilen in CSV Manifesten unterstützt.

   Um den Header zu entfernen, können Sie einen der folgenden Befehle für die Datei ausführen. Ersetzen *`file.csv`* mit dem Namen Ihrer CSV Datei.

   Führen Sie auf macOS- und Linux-Computern den `tail`-Befehl in einem Terminalfenster aus.

   ```
   tail -n +2 file.csv > tmp.csv && mv tmp.csv file.csv
   ```

   Führen Sie für Windows-Computer das folgende Skript in einem PowerShell Windows-Fenster aus. Ersetzen Sie *`File-location`* durch den Dateipfad und *`file.csv`* durch den Namen Ihrer Datei.

   ```
   $ins = New-Object System.IO.StreamReader File-location\file.csv
   $outs = New-Object System.IO.StreamWriter File-location\temp.csv
   try {
       $skip = 0
       while ( !$ins.EndOfStream ) {
   ```

```
        $line = $ins.ReadLine();
        if ( $skip -ne 0 ) {
            $outs.WriteLine($line);
        } else {
            $skip = 1
        }
    }
} finally {
    $outs.Close();
    $ins.Close();
}
Move-Item File-location\temp.csv File-location\file.csv -Force
```

10. Nachdem Sie die Kopfzeile aus der CSV Datei entfernt haben, können Sie sie als Manifest in einem S3-Batch Operations-Job verwenden. Laden Sie die CSV Datei in einen S3-Bucket oder an einen Speicherort Ihrer Wahl hoch und erstellen Sie dann einen Batch Operations-Job, der die CSV Datei als Manifest verwendet.

    Weitere Informationen zum Erstellen eines S3-Batchvorgang-Auftrags finden Sie unter Erstellen eines S3-Batch-Vorgangsauftrags.

Mit dem ACL Objektfeld arbeiten

Eine Amazon-S3-Inventory-Bericht enthält eine Liste der Objekte im S3-Quell-Bucket sowie die Metadaten für jedes Objekt. Das Feld Object Access Control List (ACL) ist ein Metadatenfeld, das im Amazon S3 S3-Inventar verfügbar ist. Insbesondere enthält das ACL Feld Objekt die Zugriffskontrollliste (ACL) für jedes Objekt. Das ACL Feld für ein Objekt definiert, welchen AWS-Konten Gruppen Zugriff auf dieses Objekt gewährt wird und welche Art von Zugriff gewährt wird. Weitere Informationen erhalten Sie unter Übersicht über die Zugriffskontrollliste (ACL) und Amazon-S3-Inventory-Liste.

Das ACL Objektfeld in Amazon S3 S3-Inventarberichten ist im JSON Format definiert. Die JSON Daten umfassen die folgenden Felder:

- `version`— Die Version des ACL Objektfeldformats in den Inventarberichten. Das Datumsformat lautet `yyyy-mm-dd`.
- `status`— Mögliche Werte sind `AVAILABLE` oder geben `UNAVAILABLE` an, ob ein Objekt für ein Objekt verfügbar ACL ist. Wenn der Status für das Objekt ACL lautet`UNAVAILABLE`, ist auch der Wert des Felds Objekteigentümer im Inventarbericht`UNAVAILABLE`.

- `grants`— Berechtigungspaare, die den Berechtigungsstatus jedes Empfängers auflisten, der vom Objekt erteilt wurde. ACL Die verfügbaren Werte für einen Berechtigungsempfänger sind `CanonicalUser` und `Group`. Weitere Informationen zu den Berechtigungsempfängern finden Sie unter Berechtigungsempfänger in Zugriffssteuerungslisten.

  Für einen Berechtigungsempfänger des Typs `Group` umfasst ein Berechtigung-Berechtigungsempfänger-Paar die folgenden Attribute:

  - `uri` – Eine vordefinierte Amazon-S3-Gruppe.
  - `permission`— Die ACL Berechtigungen, die für das Objekt erteilt wurden. Weitere Informationen finden Sie unter ACL Berechtigungen für ein Objekt.
  - `type` – Den Typ `Group`, was bedeutet, dass es sich bei dem Berechtigungsempfänger um eine Gruppe handelt.

  Für einen Berechtigungsempfänger des Typs `CanonicalUser` umfasst ein Berechtigung-Berechtigungsempfänger-Paar die folgenden Attribute:

  - `canonicalId` – Eine verschleierte Form der AWS-Konto -ID. Die kanonische Benutzer-ID für ein AWS-Konto ist spezifisch für dieses Konto. Sie können die kanonische Benutzer-ID abrufen. Weitere Informationen finden Sie unter Finden Sie die kanonische Benutzer-ID für Sie AWS-Konto im Referenzhandbuch zur AWS Kontoverwaltung.

    > **Note**
    >
    > Wenn es sich bei einem Empfänger um die E-Mail-Adresse eines ACL handelt AWS-Konto, verwendet S3 Inventory diese E-Mail-Adresse AWS-Konto und den `canonicalId` `CanonicalUser` Typ, um diesen Empfänger anzugeben. Weitere Informationen finden Sie unter Berechtigungsempfänger in Zugriffssteuerungslisten.

  - `permission`— Die ACL Berechtigungen, die für das Objekt gewährt wurden. Weitere Informationen finden Sie unter ACL Berechtigungen für ein Objekt.
  - `type`— Der Typ`CanonicalUser`, der angibt, dass es sich bei dem Empfänger um einen handelt. AWS-Konto

Das folgende Beispiel zeigt mögliche Werte für das ACL Objektfeld im JSON Format:

```
{
    "version": "2022-11-10",
    "status": "AVAILABLE",
```

```
    "grants": [{
        "uri": "http://acs.amazonaws.com/groups/global/AllUsers",
        "permission": "READ",
        "type": "Group"
    }, {
        "canonicalId": "example-canonical-id",
        "permission": "FULL_CONTROL",
        "type": "CanonicalUser"
    }]
}
```

Note

Das ACL Objektfeld ist im JSON Format definiert. In einem Inventarbericht wird der Wert für das ACL Objektfeld als Base64-kodierte Zeichenfolge angezeigt.
Nehmen wir beispielsweise an, Sie haben das folgende ACL Objektfeld im Format: JSON

```
{
        "version": "2022-11-10",
        "status": "AVAILABLE",
        "grants": [{
            "canonicalId": "example-canonical-user-ID",
            "type": "CanonicalUser",
            "permission": "READ"
        }]
}
```

Das ACL Objektfeld ist codiert und wird als folgende Base64-kodierte Zeichenfolge angezeigt:

```
eyJ2ZXJzaW9uIjoiMjAyMi0xMS0xMCIsInN0YXR1cyI6IkFWQUlMQUJMRSIsImdyYW50cyI6W3siY2Fub25pY2FsSW
```

Um den dekodierten Wert JSON für das ACL Objektfeld zu erhalten, können Sie dieses Feld in Amazon Athena abfragen. Abfragebeispiele finden Sie unter Abfragen von Amazon S3 Inventory mit Amazon Athena.

Bewährte Methoden für Designmuster: Optimieren der Leistung von Amazon S3

Ihre Anwendungen erreichen schnell Tausende von Transaktionen pro Sekunde bei der Anfrageleistung, wenn Speicherinhalte zu Amazon S3 hochgeladen oder von dort abgerufen werden. Amazon S3 wird automatisch auf hohe Anfrageraten skaliert. Ihre Anwendung kann beispielsweise mindestens 3.500 PUT/COPY/POST/DELETE or 5,500 GET/HEAD Anfragen pro Sekunde pro partitioniertem Amazon S3 S3-Präfix erreichen. Es gibt keine Einschränkungen für die Anzahl der Präfixe in einem Bucket. Sie können Ihre Lese- und Schreibleistung steigern, indem Sie Parallelisierung durchführen. Wenn Sie beispielsweise 10 Präfixe in einem Amazon S3-Bucket für parallele Lesevorgänge einrichten, können Sie damit die Leseleistung auf 55 000 Leseanfragen pro Sekunde skalieren. Auf ähnliche Weise können Sie Schreibvorgänge skalieren, indem Sie auf mehrere Präfixe schreiben. Die Skalierung erfolgt sowohl bei Lese- als auch bei Schreiboperationen schrittweise und nicht sofort. Während Amazon S3 auf Ihre neue höhere Anforderungsrate skaliert, können einige 503 (Slow Down)-Fehler auftreten. Diese Fehler werden nicht mehr angezeigt, sobald die Skalierung abgeschlossen ist. Weitere Informationen zum Erstellen und Verwenden von Präfixen finden Sie unter Organisieren von Objekten mit Präfixen.

So prüfen etwa manche Data Lake-Anwendungen in Amazon S3 Millionen oder Milliarden von Objekten auf Anfragen, die über Petabytes von Daten ausgeführt werden. Diese Data-Lake-Anwendungen erzielen Single-Instance-Übertragungsraten, die die Netzwerkschnittstellennutzung für ihre EC2Amazon-Instance maximieren. Diese kann bis zu 100 GB/s auf einer einzelnen Instance betragen. Anschließend aggregieren diese Anwendungen den Durchsatz über mehrere Instances hinweg, um mehrere Terabit pro Sekunde zu erreichen.

Andere Anwendungen sind latenzempfindlich, wie etwa Social-Media-Messaginganwendungen. Diese Anwendungen können konsistente Latenzen für kleine Objekte (und first-byte-out Latenzen für größere Objekte) von etwa 100 bis 200 Millisekunden erreichen.

Andere AWS Services können ebenfalls dazu beitragen, die Leistung verschiedener Anwendungsarchitekturen zu beschleunigen. Wenn Sie beispielsweise höhere Übertragungsraten über eine einzelne HTTP Verbindung oder Latenzen im einstelligen Millisekundenbereich wünschen, verwenden Sie Amazon CloudFront oder Amazon ElastiCache für das Caching mit Amazon S3.

Und wenn Sie eine schnelle Datenübertragung über große Distanzen zwischen einem Client und einem S3-Bucket wünschen, verwenden Sie Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration. Transfer Acceleration nutzt die global verteilten Edge-Standorte,

um den Datentransport über geografische CloudFront Entfernungen zu beschleunigen. Wenn Ihr Amazon S3 S3-Workload serverseitige Verschlüsselung mit verwendet AWS KMS, finden Sie unter AWS KMS Limits im AWS Key Management Service Developer Guide Informationen zu den für Ihren Anwendungsfall unterstützten Anforderungsraten.

Die folgenden Themen beschreiben bewährte Verfahren und Designmuster zur Optimierung der Leistung von Anwendungen, die Amazon S3 verwenden. Die neuesten Informationen zur Leistungsoptimierung für Amazon S3 finden Sie unter Leistungsrichtlinien für Amazon S3 und Leistungsentwurfsmuster für Amazon S3.

> **Note**
>
> Weitere Informationen zur Verwendung der Speicherklasse Amazon S3 Express One Zone mit Verzeichnis-Buckets finden Sie unter S3 Express One Zone und Arbeiten mit Verzeichnis-Buckets.

Themen

- Leistungsrichtlinien für Amazon S3
- Leistungsentwurfsmuster für Amazon S3

Leistungsrichtlinien für Amazon S3

Bei der Erstellung von Anwendungen, die Objekte zu Amazon S3 hochladen und davon abrufen, sollten Sie unsere bewährten Methoden befolgen, um die Leistung zu optimieren. Wir bieten auch detailliertere Leistungsentwurfsmuster für Amazon S3 .

Zur Erzielung der besten Leistung für Ihre Anwendung auf Amazon S3 empfehlen wir die folgenden Vorgehensweisen.

Themen

- Messen der Leistung
- Skalieren Sie Speicherverbindungen horizontal
- Verwenden Sie Abrufe im Bytebereich
- Wiederholung von Anforderungen für latenzsensitive Anwendungen

- Kombinieren Sie Amazon S3 (Storage) und Amazon EC2 (Compute) in einem AWS-Region
- Verwenden Sie Amazon S3 Transfer Acceleration, um die durch Entfernung verursachte Latenz zu minimieren
- Verwenden Sie die neueste Version von AWS SDKs

Messen der Leistung

Achten Sie bei der Leistungsoptimierung auf den Netzwerkdurchsatz und die DRAM Anforderungen. CPU Abhängig von der Mischung der Anforderungen an diese verschiedenen Ressourcen kann es sich lohnen, verschiedene EC2Amazon-Instance-Typen zu evaluieren. Weitere Informationen zu Instance-Typen finden Sie unter Instance-Typen im EC2Amazon-Benutzerhandbuch.

Es ist auch hilfreich, bei der Leistungsmessung die DNS Suchzeit, die Latenz und die Datenübertragungsgeschwindigkeit mithilfe von HTTP Analysetools zu berücksichtigen.

Um die Leistungsanforderungen zu verstehen und die Leistung Ihrer Anwendung zu optimieren, können Sie auch die 503-Fehlerantworten überwachen, die Sie erhalten. Die Überwachung bestimmter Leistungsmetriken kann mit zusätzlichen Kosten verbunden sein. Weitere Informationen finden Sie unter Amazon S3 – Preise.

Überwachen der Anzahl der Antworten auf 503 (Slow Down)-Statusfehler

Um die Anzahl der Antworten auf 503-Statusfehler zu überwachen, die Sie erhalten, können Sie eine der folgenden Optionen verwenden:

- Verwenden Sie CloudWatch Amazon-Anforderungsmetriken für Amazon S3. Die CloudWatch Anforderungsmetriken beinhalten eine Metrik für Antworten mit dem Status 5xx. Weitere Informationen zu CloudWatch Anforderungsmetriken finden Sie unterMetriken mit Amazon überwachen CloudWatch.
- Verwenden Sie die Anzahl der 503 (Service Unavailable)-Fehler, die im Abschnitt „Erweiterte Metriken“ von Amazon S3 Storage Lens verfügbar ist. Weitere Informationen finden Sie unter Verwenden der Metriken von S3 Storage Lens zur Verbesserung der Leistung.
- Verwenden Sie die Amazon-S3-Serverzugriffsprotokollierung. Mit der Serverzugriffsprotokollierung können Sie alle Anforderungen filtern und überprüfen, die 503 (Internal Error)-Antworten erhalten. Sie können auch Amazon Athena verwenden, um Protokolle zu analysieren. Weitere Informationen zu Server-Zugriffsprotokollen finden Sie unter Protokollieren von Anfragen mit Server-Zugriffsprotokollierung.

Wenn Sie die Anzahl der Statusfehlercodes HTTP 503 überwachen, können Sie häufig wertvolle Erkenntnisse darüber gewinnen, welche Präfixe, Schlüssel oder Buckets die Anfragen am stärksten einschränken.

Skalieren Sie Speicherverbindungen horizontal

Die Verteilung von Anfragen über zahlreiche Verbindungen ist ein übliches Designmuster für die horizontale Skalierung der Leistung. Wenn Sie hochleistungsfähige Anwendungen erstellen, stellen Sie sich Amazon S3 als sehr großes verteiltes System vor, nicht als einen einzelnen Netzwerkendpunkt wie ein herkömmlicher Speicherserver. Sie erreichen die beste Leistung durch die Ausgabe mehrerer gleichzeitiger Anfragen an Amazon S3. Verteilen Sie diese Anfragen über separate Verbindungen, um die von Amazon S3 aus zugängliche Bandbreite zu maximieren. Amazon S3 hat keine Beschränkungen für die Anzahl der Verbindungen, die mit Ihrem Bucket hergestellt werden.

Verwenden Sie Abrufe im Bytebereich

Mithilfe des `Range` HTTP Headers in einer GETObjektanforderung können Sie einen Bytebereich aus einem Objekt abrufen und dabei nur den angegebenen Teil übertragen. Sie können gleichzeitige Verbindungen zu Amazon S3 verwenden, um verschiedene Bytebereiche aus demselben Objekt abzurufen. Dies hilft beim Erreichen eines höheren aggregierten Durchsatzes als bei einer einzelnen Anforderung eines ganzen Objekts. Der Abruf kleinerer Bereiche eines größeren Objekts ermöglicht Ihrer Anwendung auch die Verbesserung der Zeiten für Wiederholungsversuche, wenn Anforderungen unterbrochen werden. Weitere Informationen finden Sie unter Herunterladen von Objekten.

Typische Größen für Bytebereichanforderungen sind 8 oder 16 MB. Wenn Objekte einen mehrteiligen Upload PUT verwenden, empfiehlt es sich, GET sie in den gleichen Teilgrößen (oder zumindest an den Teilgrenzen) zu platzieren, um eine optimale Leistung zu erzielen. GETAnfragen können sich direkt auf einzelne Teile beziehen, zum Beispiel `GET ?partNumber=N.`

Wiederholung von Anforderungen für latenzsensitive Anwendungen

Aggressive Timeouts und Wiederholungsversuche sorgen für konsistente Latenz. Aufgrund des Umfangs von Amazon S3 gilt: Wenn die erste Anforderung langsam ist, nimmt eine alte Anfrage wahrscheinlich einen anderen Pfad und ist schnell erfolgreich. Sie AWS SDKs verfügen über konfigurierbare Werte für Timeout und Wiederholungsversuche, die Sie an die Toleranzen Ihrer spezifischen Anwendung anpassen können.

Kombinieren Sie Amazon S3 (Storage) und Amazon EC2 (Compute) in einem AWS-Region

Obwohl Namen von S3-Buckets global eindeutig sind, wird jeder Bucket in einer Region gespeichert, die Sie bei dessen Erstellung auswählen. Weitere Informationen zu den Richtlinien zur Benennung von Buckets finden Sie unter Übersicht über Buckets und Regeln zur Benennung von Buckets. Um die Leistung zu optimieren, empfehlen wir, dass Sie nach Möglichkeit auf dieselbe AWS-Region Weise von EC2 Amazon-Instances aus auf den Bucket zugreifen. Dies hilft bei der Reduzierung der Netzwerklatenz und der Datenübertragungskosten.

Weitere Informationen zu den Kosten von Datenübertragungen finden Sie unter Amazon S3 – Preise.

Verwenden Sie Amazon S3 Transfer Acceleration, um die durch Entfernung verursachte Latenz zu minimieren

Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration ermöglicht die schnelle, einfache und sichere Übertragung von Dateien über größere Entfernungen zwischen Ihrem Client und einem S3-Bucket. Transfer Acceleration nutzt die weltweit verteilten Edge-Standorte von Amazon CloudFront. Sobald die Daten an einem Edge-Standort eingehen, werden sie über einen optimierten Netzwerkpfad an Amazon S3 weitergeleitet. Transfer Acceleration ist ideal für die regelmäßige Übertragung von Daten im Gigabyte- bis Terabyte-Bereich von Kontinent zu Kontinent geeignet. Die Funktion ist auch für Kunden nützlich, die Uploads in einen zentralen Bucket aus der ganzen Welt vornehmen.

Sie können das Vergleichstool Amazon S3 Transfer Acceleration Speed verwenden, um beschleunigte und nicht beschleunigte Upload-Geschwindigkeiten in Amazon S3-Regionen zu vergleichen. Das Speed Comparison-Tool verwendet mehrteilige Uploads, um eine Datei von Ihrem Browser in verschiedene Amazon S3-Regionen mit und ohne Amazon S3 Transfer Acceleration zu übertragen.

Verwenden Sie die neueste Version von AWS SDKs

AWS SDKsSie bieten integrierte Unterstützung für viele der empfohlenen Richtlinien zur Optimierung der Amazon S3 S3-Leistung. SDKsSie bieten eine API einfachere Nutzung der Vorteile von Amazon S3 innerhalb einer Anwendung und werden regelmäßig aktualisiert, um den neuesten Best Practices zu entsprechen. Sie SDKs enthalten beispielsweise eine Logik, mit der Anfragen bei HTTP 503-Fehlern automatisch wiederholt werden, und investieren in Code, um auf langsame Verbindungen zu reagieren und sich an diese anzupassen.

Sie bieten SDKs auch den Transfer Manager, der die horizontale Skalierung von Verbindungen automatisiert, um Tausende von Anfragen pro Sekunde zu erreichen, wobei gegebenenfalls Anfragen im Bytebereich verwendet werden. Es ist wichtig, die neueste Version von zu verwenden, AWS SDKs um die neuesten Funktionen zur Leistungsoptimierung zu erhalten.

Sie können die Leistung auch optimieren, wenn Sie HTTP REST API Anfragen verwenden. Bei der REST API Verwendung von sollten Sie dieselben bewährten Methoden befolgen, die Teil von sindSDKs. Lassen Sie bei langsamen Anforderungen Timeouts und Wiederholungsversuche zu, sowie mehrere Verbindung, um den parallelen Abruf von Objektdaten zu ermöglichen. Informationen zur Verwendung von finden Sie in der Amazon Simple Storage Service API Reference. REST API

Leistungsentwurfsmuster für Amazon S3

Beim Entwurf von Anwendungen zum Upload und Abruf von Objekten von Amazon S3 sollten Sie unsere bewährten Designmuster verwenden, um eine optimale Leistung für Ihre Anwendung zu erzielen. Dazu bieten wir Ihnen Leistungsrichtlinien für Amazon S3 für Überlegungen zur Planung Ihrer Anwendungsarchitektur an.

Zur Optimierung der Leistung können Sie die folgenden Designmuster verwenden.

Themen

- Verwenden von Caching für Inhalte, auf die häufig zugegriffen wird
- Timeouts und Wiederholungsversuche für latenzempfindliche Anwendungen
- Horizontale Skalierung und Parallelisierung von Anfragen für hohen Durchsatz
- Verwenden von Amazon S3 Transfer Acceleration zur Beschleunigung geografisch verteilter Datenübertragungen

Verwenden von Caching für Inhalte, auf die häufig zugegriffen wird

Viele Anwendungen, die Daten in Amazon S3 speichern, stellen einen „Arbeitssatz“ der Daten bereit, der von Benutzern häufig angefragt wird. Wenn ein Workload wiederholte GET Anfragen für eine gemeinsame Gruppe von Objekten sendet, können Sie einen Cache wie Amazon CloudFront, Amazon oder verwenden ElastiCache, um die Leistung AWS Elemental MediaStorezu optimieren. Die erfolgreiche Cache-Nutzung kann zu niedriger Latenz und zu hohen Datenübertragungsraten führen. Anwendungen, die die Zwischenspeicherung verwenden, senden auch weniger direkte Anforderungen an Amazon S3, was zur Senkung der Anfragekosten beitragen kann.

Amazon CloudFront ist ein schnelles Netzwerk zur Bereitstellung von Inhalten (CDN), das Daten aus Amazon S3 transparent in einer Vielzahl von geografisch verteilten Präsenzpunkten () zwischenspeichert. PoPs Wenn auf Objekte aus mehreren Regionen oder über das Internet zugegriffen werden kann, CloudFront können Daten in der Nähe der Benutzer zwischengespeichert werden, die auf die Objekte zugreifen. Dies kann zu hoher Leistung bei der Bereitstellung beliebter Amazon S3-Inhalte führen. Informationen dazu CloudFront finden Sie im Amazon CloudFront Developer Guide.

Amazon ElastiCache ist ein verwalteter In-Memory-Cache. Mit ElastiCache können Sie EC2 Amazon-Instances bereitstellen, die Objekte im Speicher zwischenspeichern. Dieses Caching führt zu einer Verringerung der GET Latenz um ein Vielfaches und zu einer erheblichen Steigerung des Download-Durchsatzes. Um dies zu verwenden ElastiCache, ändern Sie die Anwendungslogik so, dass sowohl der Cache mit Hot-Objekten gefüllt wird als auch der Cache auf Hot-Objekte überprüft wird, bevor Sie sie von Amazon S3 anfordern. Beispiele für die Verwendung ElastiCache zur Verbesserung der GET Leistung von Amazon S3 finden Sie im Blogbeitrag Turbocharge Amazon S3 with Amazon ElastiCache for Redis.

AWS Elemental MediaStore ist ein Caching- und Inhaltsverteilungssystem, das speziell für Video-Workflows und Medienbereitstellung von Amazon S3 entwickelt wurde. MediaStore bietet end-to-end Speicher APIs speziell für Videos und wird für leistungsabhängige Video-Workloads empfohlen. Informationen zu MediaStore finden Sie im AWS Elemental MediaStore Benutzerhandbuch.

Timeouts und Wiederholungsversuche für latenzempfindliche Anwendungen

Es gibt bestimmte Situationen, in denen eine Anwendung eine Antwort von Amazon S3 erhält, die darauf hinweist, dass ein Wiederholungsversuch erforderlich ist. Amazon S3 ordnet Bucket- und Objektnamen den damit verbundenen Objektdaten zu. Wenn eine Anwendung hohe Anforderungsraten generiert (in der Regel anhaltende Raten von über 5.000 Anfragen pro Sekunde für eine kleine Anzahl von Objekten), erhält sie möglicherweise 503 verlangsamte Antworten. HTTP Wenn diese Fehler auftreten, AWS SDK implementiert jede Methode eine automatische Wiederholungslogik mit exponentiellem Backoff. Wenn Sie keine verwenden AWS SDK, sollten Sie beim Empfang des 503-Fehlers eine Wiederholungslogik implementieren. HTTP Informationen zu Back-Off-Techniken finden Sie unter Verhalten bei Wiederholungsversuchen im Referenzhandbuch AWS SDKsund im Tools-Referenzhandbuch.

Amazon S3 wird automatisch in Reaktion auf andauernde neue Anforderungsraten skaliert und optimiert so die Leistung in dynamischer Weise. Während Amazon S3 intern für eine neue Anforderungsrate optimiert, erhalten Sie vorübergehend HTTP 503 Antworten auf Anfragen, bis die

Optimierung abgeschlossen ist. Nachdem Amazon S3 die Leistung intern für die neue Anfragerate optimiert hat, werden alle Anfragen generell ohne Wiederholungsversuche bereitgestellt.

Für latenzsensitive Anwendungen empfiehlt Amazon S3 Nachverfolgung und aggressive Wiederholungsversuche bei langsameren Vorgängen. Wenn Sie eine Anfrage erneut versuchen, empfehlen wir, eine neue Verbindung zu Amazon S3 zu verwenden und eine neue DNS Suche durchzuführen.

Wenn Sie sehr große Anforderungen unterschiedlicher Größe (beispielsweise mit mehr als 128 MB) durchführen, sollten Sie den erreichten Durchsatz nachverfolgen und für die langsamsten 5 Prozent der Anforderungen Wiederholungsversuche durchführen. Wenn Sie kleinere Anfragen stellen (z. B. weniger als 512 KB), bei denen die mittleren Latenzen oft im Bereich von mehreren zehn Millisekunden liegen, ist es eine gute Richtlinie, einen OR-Vorgang nach 2 Sekunden erneut zu versuchen. GET PUT Wenn weitere Wiederholungsversuche erforderlich sind, ist ein Backoff die beste Lösung. So empfehlen wir beispielsweise die Ausgabe eines Wiederholungsversuchs nach zwei Sekunden und eines zweiten Versuchs nach weiteren vier Sekunden.

Wenn Ihre Anwendung Anfragen mit fester Größe an Amazon S3 sendet, sollten Sie konsistentere Reaktionszeiten für diese einzelnen Anfragen erwarten. In diesem Fall ist es eine einfache Strategie, das langsamste Prozent der Anforderungen zu identifizieren und diese zu wiederholen. Selbst ein einziger Wiederholungsversuch ist oft erfolgreich für die Reduzierung der Latenz.

Wenn Sie AWS Key Management Service (AWS KMS) für die serverseitige Verschlüsselung verwenden, finden Sie unter Kontingente im AWS Key Management Service Entwicklerhandbuch Informationen zu den Anforderungsraten, die für Ihren Anwendungsfall unterstützt werden.

Horizontale Skalierung und Parallelisierung von Anfragen für hohen Durchsatz

Amazon S3 ist ein sehr großes verteiltes System. Um diese Größe zu nutzen, sollten Sie parallele Anforderungen horizontal zu den Amazon S3-Service-Endpunkten skalieren. Zusätzlich zur Verteilung der Anforderungen in Amazon S3 hilft dieses Skalierungskonzept dabei, die Last über mehrere Pfade im Netzwerk zu verteilen.

Für Übertragungen mit hohem Durchsatz empfiehlt Amazon S3 die Verwendung von Anwendungen, die mehrere Verbindungen zu GET oder PUT Daten parallel verwenden. Dies wird beispielsweise von Amazon S3 Transfer Manager in AWS Java unterstütztSDK, und die meisten anderen AWS SDKs bieten ähnliche Konstrukte. Für manche Anwendungen können Sie parallele Verbindungen

dadurch erreichen, dass Sie Anforderungen gleichzeitig in verschiedenen Anwendungsthreads oder in verschiedenen Anwendungsinstances starten. Das beste Konzept hängt von Ihrer Anwendung und der Struktur der Objekte ab, auf die Sie zugreifen.

Sie können das verwenden AWS SDKs, um PUT Anfragen direkt GET auszustellen, anstatt die Verwaltung von Übertragungen in der zu verwenden. AWS SDK Mit diesem Ansatz können Sie Ihre Arbeitslast direkter anpassen und gleichzeitig von der SDK Unterstützung für Wiederholungsversuche und der Bearbeitung aller HTTP 503 Antworten profitieren, die möglicherweise auftreten. Wenn Sie große Objekte innerhalb einer Region von Amazon S3 nach Amazon herunterladen, empfehlen wir generellEC2, gleichzeitige Anfragen für Bytebereiche eines Objekts mit einer Granularität von 8—16 MB zu stellen. Stellen Sie eine gleichzeitige Anfrage für jeweils 85—90. MB/s of desired network throughput. To saturate a 10 Gb/s network interface card (NIC), you might use about 15 concurrent requests over separate connections. You can scale up the concurrent requests over more connections to saturate faster NICs, such as 25 Gb/s or 100 Gb/s NICs

Die Messung der Leistung ist wichtig für die Einstellung der Anzahl der gleichzeitig auszugebenden Anforderungen. Wir empfehlen, mit jeweils einer einzigen Anforderung zu beginnen. Messen Sie die erreichte Netzwerkbandbreite und die Nutzung weiterer Ressourcen durch Ihre Anwendung bei der Verarbeitung der Daten. Sie können dann die Engpassressource (die Ressource mit der höchsten Nutzung) identifizieren und so die Anzahl der Anforderungen ermitteln, die wahrscheinlich nützlich waren. Wenn beispielsweise die Bearbeitung einer Anfrage nach der anderen zu einer CPU Auslastung von 25 Prozent führt, deutet dies darauf hin, dass bis zu vier gleichzeitige Anfragen berücksichtigt werden können. Messungen sind sehr wichtig, und es lohnt sich, die Ressourcennutzung zu ermitteln, wenn die Anforderungsrate erhöht wird.

Wenn Ihre Anwendung Anfragen direkt an Amazon S3 sendet, empfehlen wir RESTAPI, einen Pool von HTTP Verbindungen zu verwenden und jede Verbindung für eine Reihe von Anfragen wiederzuverwenden. Durch die Vermeidung einer Verbindungseinrichtung pro Anfrage entfällt die Notwendigkeit, bei jeder Anfrage TCP Slow-Start- und Secure Sockets Layer (SSL) -Handshakes durchzuführen. Informationen zur Verwendung von finden Sie in der Amazon Simple Storage Service API Reference. REST API

Schließlich lohnt es sich, darauf zu achten DNS und zu überprüfen, ob Anfragen über einen großen Pool von Amazon S3 S3-IP-Adressen verteilt werden. DNSAbfragen für Amazon S3 durchlaufen eine große Liste von IP-Endpunkten. Das Caching von Resolvern oder Anwendungscode, der eine einzelne IP-Adresse wiederverwendet, profitiert nicht von der Adressendiversität und dem daraus resultierenden Lastenausgleich. Netzwerk-Utility-Tools wie das `netstat` Befehlszeilentool können die IP-Adressen anzeigen, die für die Kommunikation mit Amazon S3 verwendet werden, und wir

stellen Richtlinien für die zu verwendenden DNS Konfigurationen bereit. Weitere Informationen zu diesen Richtlinien finden Sie unter Anfragen stellen in der Amazon S3 API S3-Referenz.

Verwenden von Amazon S3 Transfer Acceleration zur Beschleunigung geografisch verteilter Datenübertragungen

Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration ist effektiv bei der Minimierung oder Beseitigung der durch geographische Entfernung zwischen global verteilten Clients und einer regionalen Anwendung mit Amazon S3 verursachten Latenz. Transfer Acceleration nutzt die global verteilten Edge-Standorte CloudFront für den Datentransport. Das AWS Edge-Netzwerk verfügt über Präsenzpunkte an mehr als 50 Standorten. Heute wird es verwendet, um Inhalte über Amazon Route 53 zu verteilen CloudFront und schnelle Antworten auf DNS Anfragen zu geben.

Dazu hilft das Edge-Netzwerk bei der Beschleunigung von Datenübertragungen zu und aus Amazon S3. Es ist ideal für Anwendungen, die Daten über oder zwischen Kontinenten übertragen, eine schnelle Internetverbindung nutzen, große Objekte verwenden oder zahlreiche Inhalte hochladen müssen. Sobald die Daten an einem Edge-Standort eingehen, werden sie über einen optimierten Netzwerkpfad an Ihren Amazon S3-Bucket weitergeleitet. Allgemein gilt: Je weiter Sie von einer Amazon S3-Region entfernt sind, um so größer ist die Geschwindigkeits-Verbesserung, die Sie von Transfer Acceleration erwarten können.

Sie können Transfer Acceleration auf neuen oder bestehenden Buckets einrichten. Sie können einen separaten Amazon S3 Transfer Acceleration Acceleration-Endpunkt verwenden, um die AWS Edge-Standorte zu verwenden. Die beste Möglichkeit zur Prüfung, ob Transfer Acceleration die Client-Anfrageleistung erhöht, ist die Verwendung des Amazon S3-Transfer-Acceleration-Speed-Comparison-Tools. Netzwerkkonfigurationen und Bedingungen variieren von zeit zu Zeit und von Standort zu Standort. Sie werden daher nur für Übertragungen belastet, bei denen Amazon S3 Transfer Acceleration Ihre Upload-Leistung potentiell verbessern kann. Informationen zur Verwendung von Transfer Acceleration mit anderen AWS SDKs finden Sie unterAktivieren und Verwenden von S3 Transfer Acceleration.

Hosten einer statischen Website mit Amazon S3

Sie können Amazon S3 verwenden, um eine statische Website zu hosten. Auf einer statischen Website enthalten einzelne Webseiten statischen Inhalt. Sie könnten auch clientseitige Skripts enthalten.

> **Note**
>
> Wir empfehlen Ihnen, AWS Amplify Hosting zu verwenden, um statische Website-Inhalte zu hosten, die auf S3 gespeichert sind. Amplify Hosting ist ein vollständig verwalteter Service, mit dem Sie Ihre Websites auf einfache Weise in einem weltweit verfügbaren Content Delivery Network (CDN) bereitstellen können CloudFront, das von Amazon betrieben wird und sicheres statisches Website-Hosting ermöglicht. Mit AWS Amplify Hosting können Sie den Speicherort Ihrer Objekte in Ihrem Allzweck-Bucket auswählen, Ihre Inhalte in einem verwalteten Bereich bereitstellen und eine Öffentlichkeit HTTPS URL generierenCDN, damit Ihre Website von überall aus zugänglich ist. Weitere Informationen zu Amplify Hosting finden Sie unter Bereitstellen einer statischen Website auf AWS Amplify Hosting aus einem S3-Allzweck-Bucket und Bereitstellen einer statischen Website von S3 mit der AWS Amplify Amplify-Konsole im Konsolen-Benutzerhandbuch.

Weitere Informationen zum Hosten einer statischen Website auf Amazon S3, einschließlich Anleitungen und step-by-step exemplarischer Vorgehensweisen, finden Sie in den folgenden Themen.

Themen

- Website-Endpunkte
- Aktivieren des Website-Hostings
- Konfigurieren eines Indexdokuments
- Konfigurieren eines benutzerdefinierten Fehlerdokuments
- Festlegen von Berechtigungen für den Website-Zugriff
- (Optional) Protokollieren des Webdatenverkehrs
- (Optional) Konfigurieren einer Webseitenumleitung
- Verwendung von ursprungsübergreifendem Ressourcenaustausch () CORS
- Tutorials für statische Websites

Website-Endpunkte

Wenn Sie Ihren Bucket als statische Website konfigurieren, steht die Website an dem für die AWS-Region-spezifischen Website-Endpunkt des Buckets zur Verfügung. Website-Endpunkte unterscheiden sich von den Endpunkten, an die Sie Anfragen senden. REST API Weitere Informationen zu den Unterschieden zwischen den Endpunkten finden Sie unter Hauptunterschiede zwischen einem Website-Endpunkt und einem REST API Endpunkt.

Je nach Region weisen Ihre Amazon-S3-Website-Endpunkte eines der beiden folgenden Formate auf.

- s3-website dash (-) Region - `http://bucket-name.s3-website-Region.amazonaws.com`
- s3-website dot (.) Region - `http://bucket-name.s3-website.Region.amazonaws.com`

Diese URLs geben das Standard-Indexdokument zurück, das Sie für die Website konfigurieren. Eine vollständige Liste der Amazon-S3-Website-Endpunkte finden Sie unter Amazon-S3-Website-Endpunkte.

> **Note**
>
> Um die Sicherheit Ihrer statischen Amazon S3 S3-Websites zu erhöhen, sind die Endpunkt-Domains der Amazon S3 S3-Website (z. B. s3- website-us-east -1.amazonaws.com oder s3-website.ap-south-1.amazonaws.com) in der Liste der öffentlichen Suffixe () registriert. PSL Aus Sicherheitsgründen empfehlen wir Ihnen, Cookies mit einem `__Host-`-Präfix zu verwenden, falls Sie jemals sensible Cookies im Domain-Namen für statische Amazon-S3-Webistes einrichten müssen. Diese Vorgehensweise hilft Ihnen dabei, Ihre Domain vor Cross-Site-Request-Forgery-Versuchen zu schützen (). CSRF Weitere Informationen finden Sie auf der Set-Cookie-Seite im Mozilla Developer Network.

Wenn Ihre Website öffentlich sein soll, müssen Sie Ihren gesamten Inhalt öffentlich lesbar machen, damit Ihre Kunden am Website-Endpunkt darauf zugreifen können. Weitere Informationen finden Sie unter Festlegen von Berechtigungen für den Website-Zugriff.

> **Important**
>
> Endpunkte der Amazon S3 S3-Website unterstützen keine HTTPS Zugriffspunkte. Wenn Sie verwenden möchtenHTTPS, können Sie Amazon verwenden, CloudFront um eine statische

Website bereitzustellen, die auf Amazon S3 gehostet wird. Weitere Informationen finden Sie unter Wie verwende ich, CloudFront um HTTPS Anfragen für meinen Amazon S3 S3-Bucket zu bearbeiten? Informationen zur Verwendung HTTPS mit einer benutzerdefinierten Domain finden Sie unter Konfiguration einer statischen Website mithilfe einer bei Route 53 registrierten benutzerdefinierten Domain.

Buckets mit Zahlung durch den Auftraggeber erlauben keinen Zugriff über den Website-Endpunkt. Jede Anforderung an einen solchen Bucket erhält die Antwort 403 Access Denied (403 Zugriff verweigert). Weitere Informationen finden Sie unter Nutzen von Buckets mit Zahlung durch den Anforderer für Speicherübertragungen und Nutzung.

Note

Wir empfehlen Ihnen, AWS Amplify Hosting zu verwenden, um statische Website-Inhalte zu hosten, die auf S3 gespeichert sind. Amplify Hosting ist ein vollständig verwalteter Service, mit dem Sie Ihre Websites auf einfache Weise in einem weltweit verfügbaren Content Delivery Network (CDN) bereitstellen können CloudFront, das von Amazon betrieben wird und sicheres statisches Website-Hosting ermöglicht. Mit AWS Amplify Hosting können Sie den Speicherort Ihrer Objekte in Ihrem Allzweck-Bucket auswählen, Ihre Inhalte in einem verwalteten Bereich bereitstellen und eine Öffentlichkeit HTTPS URL generierenCDN, damit Ihre Website von überall aus zugänglich ist. Weitere Informationen zu Amplify Hosting finden Sie unter Bereitstellen einer statischen Website auf AWS Amplify Hosting aus einem S3-Allzweck-Bucket und Bereitstellen einer statischen Website von S3 mit der AWS Amplify Amplify-Konsole im Konsolen-Benutzerhandbuch.

Themen

- Website-Endpunkt-Beispiele
- Hinzufügen eines DNS CNAME
- Verwenden einer benutzerdefinierten Domäne mit Route 53
- Hauptunterschiede zwischen einem Website-Endpunkt und einem REST API Endpunkt

Website-Endpunkt-Beispiele

Die folgenden Beispiele zeigen, wie Sie auf einen Amazon-S3-Bucket zugreifen können, der als statische Website konfiguriert ist.

Example – Anfordern eines Objekts auf Stammebene

Verwenden Sie die folgende Struktur, um ein bestimmtes Objekt anzufordern, das auf der Stammebene im Bucket gespeichert ist. URL

```
http://bucket-name.s3-website.Region.amazonaws.com/object-name
```

Im Folgenden wird beispielsweise das `photo.jpg` Objekt URL angefordert, das auf der Stammebene im Bucket gespeichert ist.

```
http://example-bucket.s3-website.us-west-2.amazonaws.com/photo.jpg
```

Example – Anfordern eines Objekts in einem Präfix

Verwenden Sie diese URL Struktur, um ein Objekt anzufordern, das in einem Ordner in Ihrem Bucket gespeichert ist.

```
http://bucket-name.s3-website.Region.amazonaws.com/folder-name/object-name
```

Im Folgenden wird URL das `docs/doc1.html` Objekt in Ihrem Bucket angefordert.

```
http://example-bucket.s3-website.us-west-2.amazonaws.com/docs/doc1.html
```

Hinzufügen eines DNS CNAME

Wenn Sie eine registrierte Domain haben, können Sie einen DNS CNAME Eintrag hinzufügen, der auf den Endpunkt der Amazon S3 S3-Website verweist. Wenn Sie die Domain beispielsweise registriert haben`www.example-bucket.com`, könnten Sie einen Bucket erstellen und einen DNS CNAME Datensatz hinzufügen`www.example-bucket.com`, der auf Folgendes verweist`www.example-bucket.com.s3-website.Region.amazonaws.com`. Alle Anfragen an `http://www.example-bucket.com` werden an `www.example-bucket.com.s3-website.Region.amazonaws.com` weitergeleitet.

Weitere Informationen finden Sie unter Amazon S3 URLs mit CNAME Datensätzen anpassen.

Verwenden einer benutzerdefinierten Domäne mit Route 53

Anstatt über einen Amazon-S3-Website-Endpunkt auf die Website zuzugreifen, können Sie beispielsweise Ihre eigene Domäne verwenden, die bei Amazon Route 53 registriert ist, um Ihre Inhalte bereitzustellen, z. B, `example.com`. Sie können Amazon S3 mit Route 53 verwenden,

um eine Website in der Root-Domäne zu hosten. Wenn Sie beispielsweise die Root-Domäne `example.com` haben und Ihre Website auf Amazon S3 hosten, können Ihre Website-Besucher von ihrem Browser aus auf die Seite zugreifen, indem sie `http://www.example.com` oder `http://example.com` eingeben.

Ein Beispiel-Walkthrough finden Sie unte Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne.

Hauptunterschiede zwischen einem Website-Endpunkt und einem REST API Endpunkt

Der Amazon-S3-Website-Endpunkt ist auf den Zugriff über einen Webbrowser optimiert. In der folgenden Tabelle werden die wichtigsten Unterschiede zwischen einem REST API Endpunkt und einem Website-Endpunkt zusammengefasst.

Wichtiger Unterschied	RESTAPIEndpunkt	Website-Endpunkt
Zugriffskontrolle	Unterstützt öffentlichen und privaten Inhalt	Unterstützt nur öffentlich lesbaren Inhalt
Verarbeiten von Fehlermeldungen	Gibt eine XML -formatierte Fehlerant wort zurück	Gibt ein Dokument zurück HTML
Unterstützung einer Umleitung	Nicht zutreffend	Unterstützt Umleitungen auf Objekt- und auf Bucket-Ebene
Unterstützte Anfragen	Unterstützt alle Bucket- und Objekt-Vorgänge	Unterstützt nur GET und HEAD Anfragen für Objekte
Antworten auf GET und HEAD Anfragen im Stammverz eichnis eines Buckets	Gibt eine Liste der Objektschlüssel im Bucket zurück.	Gibt das in der Website-Konfiguration angegebene Indexdokument zurück.

Wichtiger Unterschied	RESTAPIEndpunkt	Website-Endpunkt
Unterstützung für Secure Sockets Layer (SSL)	Unterstützt SSL Verbindungen	Unterstützt keine SSL Verbindungen

Eine vollständige Liste der Amazon-S3-Endpunkte finden Sie unter Amazon-S3-Endpunkte und -Kontingente in der Allgemeine AWS-Referenz.

Aktivieren des Website-Hostings

Wenn Sie einen Bucket als statische Website konfigurieren, müssen Sie das statische Website-Hosting aktivieren, ein Indexdokument konfigurieren und Berechtigungen festlegen.

Sie können statisches Website-Hosting mit der Amazon S3 S3-Konsole RESTAPI, AWS SDKs, dem AWS CLI, oder aktivieren AWS CloudFormation.

Informationen zum Konfigurieren Ihrer Website mit einer benutzerdefinierten Domäne finden Sie unter Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne.

Verwenden der S3-Konsole

So aktivieren Sie das statische Website-Hosting

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie das Hosting statischer Websites aktivieren wollen.
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie unter Static website hosting (Hosting statischer Websites) Edit (Bearbeiten) aus.
5. Wählen Sie Use this bucket to host a website (Diesen Bucket zum Hosten einer Website verwenden).
6. Wählen Sie unter Static website hosting (Hosting statischer Websites) die Option Enable (Aktivieren) aus

7. Geben Sie unter Index document (Index-Dokument) den Dateinamen des Index-Dokuments ein, der typischerweise `index.html` ist.

   Beim Namen des Indexdokuments wird zwischen Groß- und Kleinschreibung unterschieden und er muss exakt mit dem Dateinamen des HTML Indexdokuments übereinstimmen, das Sie in Ihren S3-Bucket hochladen möchten. Wenn Sie Ihren Bucket für das Hosting von Websites konfigurieren, müssen Sie ein Indexdokument angeben. Amazon S3 gibt dieses Indexdokument zurück, wenn Anfragen an die Root-Domäne oder einen der Unterordner gestellt werden. Weitere Informationen finden Sie unter Konfigurieren eines Indexdokuments.

8. Um ein eigenes benutzerdefiniertes Fehlerdokument für Fehler der Klasse 4XX bereitzustellen, geben Sie unter Fehlerdokument den Dateinamen des benutzerdefinierten Fehlerdokuments ein.

   Beim Namen des Fehlerdokuments wird zwischen Groß- und Kleinschreibung unterschieden und er muss genau mit dem Dateinamen des HTML Fehlerdokuments übereinstimmen, das Sie in Ihren S3-Bucket hochladen möchten. Wenn Sie kein benutzerdefiniertes Fehlerdokument angeben und ein Fehler auftritt, gibt Amazon S3 ein HTML Standard-Fehlerdokument zurück. Weitere Informationen finden Sie unter Konfigurieren eines benutzerdefinierten Fehlerdokuments.

9. (Optional) Wenn Sie erweiterte Umleitungsregeln angeben möchten, geben Sie im Feld JSONUmleitungsregeln eine Beschreibung der Regeln ein.

   Beispielsweise können Sie bedingt Anfragen abhängig von bestimmten Objektschlüsselnamen oder Präfixen in der Anfrage weiterleiten. Weitere Informationen finden Sie unter Konfigurieren von Umleitungsregeln für die Verwendung von fortschrittliche bedingten Umleitungen.

10. Wählen Sie Save Changes (Änderungen speichern).

    Amazon S3 ermöglicht statisches Website-Hosting für Ihren Bucket. Unten auf der Seite sehen Sie unter Static website hosting (Hosting statischer Websites) den Website-Endpunkt für Ihren Bucket.

11. Notieren Sie unter Static website hosting (Statisches Website-Hosting) den Wert für Endpoint (Endpunkt).

    Der Endpoint (Endpunkt) ist der Amazon-S3-Website-Endpunkt für Ihren Bucket. Nachdem Sie den Bucket als statische Website konfiguriert haben, können Sie diesen Endpunkt verwenden, um Ihre Website zu testen.

Verwenden von REST API

Weitere Informationen zum direkten Senden von REST Anfragen zur Aktivierung des statischen Website-Hostings finden Sie in den folgenden Abschnitten der Amazon Simple Storage Service API Reference:

- PUTBucket-Website
- GETBucket-Webseite
- DELETEBucket-Webseite

Mit der AWS SDKs

Um eine statische Website auf Amazon S3 zu hosten, konfigurieren Sie einen Amazon-S3-Bucket für ein Website-Hosting und laden dann Ihren Website-Inhalt in den Bucket hoch. Sie können den auch verwenden, AWS SDKs um die Website-Konfiguration programmgesteuert zu erstellen, zu aktualisieren und zu löschen. SDKsSie bieten Wrapper-Klassen rund um den Amazon S3 RESTAPI. Wenn Ihre Anwendung dies erfordert, können Sie REST API Anfragen direkt von Ihrer Anwendung aus senden.

.NET

Das folgende Beispiel zeigt, wie Sie AWS SDK for .NET die Website-Konfiguration für einen Bucket verwalten. Um einem Bucket eine Website-Konfiguration hinzuzufügen, geben Sie einen Bucket-Namen und eine Website-Konfiguration an. Die Website-Konfiguration muss ein Indexdokument enthalten und kann optional ein Fehlerdokument enthalten. Diese Dokumente müssen im Bucket gespeichert sein. Weitere Informationen finden Sie auf der PUTBucket-Website. Weitere Informationen zur Amazon-S3-Website-Funktion finden Sie unter Hosten einer statischen Website mit Amazon S3.

Das folgende C#-Code-Beispiel fügt dem angegebenen Bucket eine Website-Konfiguration hinzu. Die Konfiguration gibt die Namen für das Indexdokument und das Fehlerdokument an. Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem AWS SDK for. NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Threading.Tasks;
```

```
namespace Amazon.DocSamples.S3
{
    class WebsiteConfigTest
    {
        private const string bucketName = "*** bucket name ***";
        private const string indexDocumentSuffix = "*** index object key ***"; //
 For example, index.html.
        private const string errorDocument = "*** error object key ***"; // For
 example, error.html.
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 client;
        public static void Main()
        {
            client = new AmazonS3Client(bucketRegion);
            AddWebsiteConfigurationAsync(bucketName, indexDocumentSuffix,
 errorDocument).Wait();
        }

        static async Task AddWebsiteConfigurationAsync(string bucketName,
                                            string indexDocumentSuffix,
                                            string errorDocument)
        {
            try
            {
                // 1. Put the website configuration.
                PutBucketWebsiteRequest putRequest = new PutBucketWebsiteRequest()
                {
                    BucketName = bucketName,
                    WebsiteConfiguration = new WebsiteConfiguration()
                    {
                        IndexDocumentSuffix = indexDocumentSuffix,
                        ErrorDocument = errorDocument
                    }
                };
                PutBucketWebsiteResponse response = await
 client.PutBucketWebsiteAsync(putRequest);

                // 2. Get the website configuration.
                GetBucketWebsiteRequest getRequest = new GetBucketWebsiteRequest()
                {
                    BucketName = bucketName
```

```
                };
                GetBucketWebsiteResponse getResponse = await
 client.GetBucketWebsiteAsync(getRequest);
                Console.WriteLine("Index document: {0}",
 getResponse.WebsiteConfiguration.IndexDocumentSuffix);
                Console.WriteLine("Error document: {0}",
 getResponse.WebsiteConfiguration.ErrorDocument);
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
 writing an object", e.Message);
            }
        }
    }
}
```

PHP

Das folgende PHP Beispiel fügt dem angegebenen Bucket eine Website-Konfiguration hinzu. Die Methode `create_website_config` stellt explizit die Namen des Indexdokuments und des Fehlerdokuments bereit. Das Beispiel ruft auch die Website-Konfiguration ab und gibt die Antwort aus. Weitere Informationen zur Amazon-S3-Website-Funktion finden Sie unter Hosten einer statischen Website mit Amazon S3.

Weitere Informationen zu AWS SDK for Ruby API finden Sie unter AWS SDKfor Ruby — Version 2.

```
 require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucket = '*** Your Bucket Name ***';

$s3 = new S3Client([
    'version' => 'latest',
    'region'  => 'us-east-1'
]);
```

```
// Add the website configuration.
$s3->putBucketWebsite([
    'Bucket'                => $bucket,
    'WebsiteConfiguration'  => [
        'IndexDocument' => ['Suffix' => 'index.html'],
        'ErrorDocument' => ['Key' => 'error.html']
    ]
]);

// Retrieve the website configuration.
$result = $s3->getBucketWebsite([
    'Bucket' => $bucket
]);
echo $result->getPath('IndexDocument/Suffix');

// Delete the website configuration.
$s3->deleteBucketWebsite([
    'Bucket' => $bucket
]);
```

Mit dem AWS CLI

Weitere Informationen zur Verwendung von AWS CLI , um einen S3-Bucket als statische Website zu konfigurieren, finden Sie unter website in der AWS CLI Befehlsreferenz.

Als Nächstes müssen Sie das Indexdokument konfigurieren und Berechtigungen festlegen. Weitere Informationen finden Sie unter Konfigurieren eines Indexdokuments und Festlegen von Berechtigungen für den Website-Zugriff.

Optional können Sie auch ein Fehlerdokument, eineWebdatenverkehrs-Protokollierung oder eine Weiterleitung konfigurieren.

Konfigurieren eines Indexdokuments

Wenn Sie Website-Hosting aktivieren, müssen Sie auch ein Indexdokument konfigurieren und hochladen. Ein Indexdokument ist eine Webseite, die Amazon S3 zurückgibt, wenn eine Anforderung für die Root einer Website oder einen Unterordner erfolgt. Wenn ein Benutzer beispielsweise `http://www.example.com` in den Browser eingibt, fordert er keine spezifische Seite an. In diesem

Fall stellt Amazon S3 das Indexdokument bereit, das manchmal auch als Standardseite bezeichnet wird.

Wenn Sie das statische Website-Hosting für Ihren Bucket aktivieren, geben Sie den Namen des Indexdokuments ein (z. B, `index.html`). Nachdem Sie das statische Website-Hosting für Ihren Bucket aktiviert haben, laden Sie eine HTML Datei mit dem Namen des Indexdokuments in Ihren Bucket hoch.

Der abschließende Schrägstrich auf Stammebene ist optional. URL Wenn Sie Ihre Website beispielsweise `index.html` als Indexdokument konfigurieren, wird eine der folgenden Optionen zurückgegeben. URLs `index.html`

```
http://example-bucket.s3-website.Region.amazonaws.com/
http://example-bucket.s3-website.Region.amazonaws.com
```

Weitere Informationen zu Amazon-S3-Website-Endpunkten finden Sie unter Website-Endpunkte.

Indexdokument und Ordner

In Amazon S3 ist ein Bucket ein eindimensionaler Container mit Objekten. Er bietet keine hierarchische Organisation wie das Dateisystem auf Ihrem Computer. Sie können jedoch mit Objektschlüsselnamen, die eine Ordnerstruktur implizieren, eine logische Hierarchie erstellen.

Betrachten wir beispielsweise einen Bucket mit drei Objekten, die die folgenden Schlüsselnamen besitzen. Obwohl diese nicht in einer physischen Hierarchie gespeichert sind, können Sie aus den Schlüsselnamen die folgende logische Ordnerstruktur ableiten:

- `sample1.jpg` – Das Objekt befindet sich im Stammverzeichnis des Buckets.
- `photos/2006/Jan/sample2.jpg` – Das Objekt befindet sich im Unterordner `photos/2006/Jan`.
- `photos/2006/Feb/sample3.jpg` – Das Objekt befindet sich im Unterordner `photos/2006/Feb`.

In der Amazon-S3-Konsole können Sie auch einen Ordner in einem Bucket erstellen. Sie können beispielsweise einen Ordner mit dem Namen erstelle `photos`. Sie können Objekte in den Bucket hochladen, oder in den `photos`-Ordner im Bucket. Wenn Sie dem Bucket das Objekt `sample.jpg` hinzufügen, ist der Schlüsselname `sample.jpg`. Wenn Sie das Objekt in den Ordner `photos` hochladen, ist der Objektschlüsselname `photos/sample.jpg`.

Wenn Sie eine Ordnerstruktur in Ihrem Bucket erstellen, muss es auf jeder Ebene ein Indexdokument geben. In jedem Ordner muss das Indexdokument den gleichen Namen haben, `index.html`. Wenn ein Benutzer ein angibtURL, das einer Ordnersuche ähnelt, bestimmt das Vorhandensein oder Fehlen eines abschließenden Schrägstrichs das Verhalten der Website. Im Folgenden URL wird beispielsweise mit einem abschließenden Schrägstrich das Indexdokument zurückgegeben. `photos/index.html`

```
http://bucket-name.s3-website.Region.amazonaws.com/photos/
```

Wenn Sie jedoch den abschließenden Schrägstrich aus dem vorherigen ausschließenURL, sucht Amazon S3 zunächst nach einem Objekt `photos` im Bucket. Wird das Objekt `photos` nicht gefunden, sucht es nach einem Indexdokument, `photos/index.html`. Wird dieses Dokument gefunden, gibt Amazon S3 eine `302 Found`-Meldung zurück und verweist auf den `photos/`-Schlüssel. Für nachfolgende Anforderungen für `photos/` gibt Amazon S3 `photos/index.html` zurück. Wird das Indexdokument nicht gefunden, gibt Amazon S3 einen Fehler zurück.

Konfigurieren eines Indexdokuments

Gehen Sie wie folgt vor, um ein Indexdokument mit der S3-Konsole zu konfigurieren. Sie können ein Indexdokument auch mit dem RESTAPI, dem AWS SDKs AWS CLI, dem oder konfigurieren. AWS CloudFormation

> **Note**
>
> In einem Bucket mit aktivierter Versionsverwaltung können Sie mehrere Kopien der Datei „`index.html`“ hochladen, es wird jedoch nur die neueste Version verwendet. Weitere Informationen zur Verwendung der S3-Versionsverwaltung finden Sie unter Beibehaltung mehrerer Versionen von Objekten mit S3 Versioning.

Wenn Sie das statische Website-Hosting für Ihren Bucket aktivieren, geben Sie den Namen des Indexdokuments ein (z. B, **`index.html`**). Nachdem Sie das statische Website-Hosting für den Bucket aktiviert haben, laden Sie eine HTML Datei mit diesem Indexdokumentnamen in Ihren Bucket hoch.

So konfigurieren Sie das Indexdokument

1. Erstellen Sie eine Datei `index.html`.

Wenn Sie keine `index.html` Datei haben, können Sie wie folgt HTML eine Datei erstellen:

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
    <title>My Website Home Page</title>
</head>
<body>
  <h1>Welcome to my website</h1>
  <p>Now hosted on Amazon S3!</p>
</body>
</html>
```

2. Speichern Sie die Indexdatei lokal.

   Der Dateiname des Indexdokuments muss genau mit dem Namen des Indexdokuments übereinstimmen, den Sie im Dialogfeld Static website hosting (Statisches Website-Hosting) eingeben. Beim Namen des Indexdokuments wird die Groß- und Kleinschreibung berücksichtigt. Wenn Sie beispielsweise im Dialogfeld Static website hosting (Statisches Website-Hosting) `index.html` als den Namen des Index document (Indexdokuments) eingeben, muss der Dateiname des Indexdokuments ebenfalls `index.html` und nicht `Index.html` lauten.

3. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

4. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie zum Hosten einer statischen Website verwenden möchten.

5. Aktivieren Sie das Hosting statischer Websites für Ihren Bucket und geben Sie den exakten Namen Ihres Indexdokuments ein (z. B. `index.html`). Weitere Informationen finden Sie unter Aktivieren des Website-Hostings.

   Fahren Sie mit Schritt 6 fort, nachdem Sie das Hosting statischer Websites aktiviert haben.

6. Führen Sie einen der folgenden Schritte aus, um das Indexdokument in Ihren Bucket hochzuladen:

   - Ziehen Sie die Indexdatei per Drag & Drop in das Konsolen-Bucket-Verzeichnis.
   - Wählen Sie Upload (Hochladen) und folgen Sie den Anweisungen zur Auswahl und zum Hochladen der Indexdatei.

   step-by-stepAnweisungen finden Sie unterObjekte hochladen.

7. (Optional) Laden Sie andere Website-Inhalte in Ihren Bucket hoch.

Als Nächstes müssen Sie Berechtigungen für den Websitezugriff festlegen. Weitere Informationen finden Sie unter Festlegen von Berechtigungen für den Website-Zugriff.

Optional können Sie auch ein Fehlerdokument, eineWebdatenverkehrs-Protokollierung oder eine Weiterleitung konfigurieren.

Konfigurieren eines benutzerdefinierten Fehlerdokuments

Nachdem Sie Ihren Bucket als statische Website konfiguriert haben und ein Fehler auftritt, gibt Amazon S3 ein HTML Fehlerdokument zurück. Sie können Ihren Bucket optional mit einem benutzerdefinierten Fehlerdokument konfigurieren, sodass Amazon S3 bei Auftreten eines Fehlers dieses Dokument zurückgibt.

> **Note**
>
> Tritt ein Fehler auf, zeigen einige Browser beim Auftreten eines Fehlers ihre eigene Fehlermeldung an und ignorieren das Fehlerdokument, das Amazon S3 zurückgibt. Wenn beispielsweise der Fehler HTTP404 Not Found auftritt, ignoriert Google Chrome möglicherweise das von Amazon S3 zurückgegebene Fehlerdokument und zeigt einen eigenen Fehler an.

Themen

- Amazon S3 HTTP S3-Antwortcodes
- Konfigurieren eines benutzerdefinierten Fehlerdokuments

Amazon S3 HTTP S3-Antwortcodes

In der folgenden Tabelle ist die Teilmenge der HTTP Antwortcodes aufgeführt, die Amazon S3 zurückgibt, wenn ein Fehler auftritt.

HTTPFehlercode	Beschreibung
301 Moved Permanently	Wenn ein Benutzer eine Anforderung direkt an den Amazon-S3-Website-Endpunkt sendet (`http://s3-website.` *`Region`*`.amazonaws.com/`),

HTTPFehlercode	Beschreibung
(301 Dauerhaft verschoben)	gibt Amazon S3 die Antwort 301 Moved Permanently zurück und leitet diese Anforderungen auf `https://aws.amazon.com/s3/` um.
302 Found (302 Gefunden)	Wenn Amazon S3 eine Anforderung für einen Schlüssel `x`, `http://`*`bucket-name`*`.s3-website.`*`Region`*`.amazonaws.com/x` ohne nachfolgenden Schrägstrich erhält, sucht es zuerst nach dem Objekt mit dem Schlüsselnamen `x`. Wenn das Objekt nicht gefunden wird, stellt Amazon S3 fest, dass die Anfrage für den Unterordner `x` vorgesehen ist, und leitet die Anfrage um, indem es einen Schrägstrich am Ende einfügt. Es gibt 302 Found zurück.
304 Not Modified (304 Nicht verändert)	Amazon S3 verwendet die Anforderungs-Header `If-Modified-Since`, `If-Unmodified-Since`, `If-Match` und/oder `If-None-Match`, um festzustellen, ob das angeforderte Objekt dasselbe ist, wie das, das der Client im Cache vorhält. Ist das Objekt dasselbe, gibt der Website-Endpunkt die Antwort 304 Not Modified (304 Nicht verändert) zurück.
400 Malformed Request (400 Falsch formatierte Anfrage)	Der Website-Endpunkt antwortet mit 400 Malformed Request (400 Falsch formatierte Anfrage), wenn ein Benutzer versucht, über einen inkorrekten regionalen Endpunkt auf einen Bucket zuzugreifen.
403 Forbidden	Der Website-Endpunkt antwortet mit 403 Forbidden (403 Verboten), wenn ein Benutzer auf ein Objekt zugreifen will, das nicht öffentlich lesbar ist. Der Objekteigentümer muss das Objekt mithilfe einer Bucket-Richtlinie oder einer öffentlich lesbar machenACL.

HTTPFehlercode	Beschreibung
404 Not Found (404 Nicht gefunden)	Der Website-Endpunkt gibt aus den folgenden Gründen 404 Not Found (404 Nicht gefunden) zurück: • Amazon S3 stellt fest, dass sich die Website auf einen Objektschlüssel bezieht, der nicht existiert. URL • Amazon S3 geht davon aus, dass die Anforderung für ein Indexdokument gilt, das nicht existiert. • Ein in der angegebener Bucket ist URL nicht vorhanden. • Ein in der angegebener Bucket URL existiert, ist aber nicht als Website konfiguriert. Sie können ein benutzerdefiniertes Objekt erstellen, das für 404 Not Found (404 Nicht gefunden) zurückgegeben wird. Stellen Sie sicher, dass das Dokument in den als Website konfigurierten Bucket hochgeladen ist, und dass die Konfiguration für das Website-Hosting darauf ausgelegt ist, das Dokument zu verwenden. Informationen darüber, wie Amazon S3 das URL als Anfrage für ein Objekt oder ein Indexdokument interpretiert, finden Sie unterKonfigurieren eines Indexdokuments.
500 Service Error (500 Servicefe hler)	Der Website-Endpunkt reagiert mit 500 Service Error (500 Servicefehler), wenn ein interner Serverfehler auftritt.
503 Service Unavailable (503 Service nicht verfügbar)	Der Website-Endpunkt antwortet mit 503 Service Unavailable, wenn Amazon S3 feststellt, dass Sie Ihre Anforderungsrate reduzieren müssen.

Für jeden dieser Fehler gibt Amazon S3 eine vordefinierte HTML Meldung zurück. Im Folgenden finden Sie eine HTML Beispielnachricht, die bei einer Antwort vom Typ 403 Forbidden zurückgegeben wird.

403 Forbidden

- Code: AccessDenied
- Message: Access Denied
- RequestId: 873CA367A51F7EC7
- HostId: DdQezl9vkuw5luD5HKsFaTDm9KH4PZzCPRkW3igimILbTu1DiYlvXjgyd7pVxq32

An Error Occurred While Attempting to Retrieve a Custom Error Document

- Code: AccessDenied
- Message: Access Denied

Konfigurieren eines benutzerdefinierten Fehlerdokuments

Wenn Sie Ihren Bucket als statische Website konfigurieren, können Sie ein benutzerdefiniertes Fehlerdokument bereitstellen, das eine benutzerfreundliche Fehlermeldung und zusätzliche Hilfe enthält. Amazon S3 gibt Ihr benutzerdefiniertes Fehlerdokument nur für die HTTP 4XX-Klasse von Fehlercodes zurück.

Um ein benutzerdefiniertes Fehlerdokument über die S3-Konsole zu konfigurieren, führen Sie die folgenden Schritte aus. Sie können ein Fehlerdokument auch mit dem RESTAPI, dem AWS SDKs AWS CLI, dem oder AWS CloudFormation konfigurieren. Weitere Informationen finden Sie hier:

- PutBucketWebsitein der Amazon Simple Storage API Service-Referenz
- AWS: :S3: :Bucket WebsiteConfiguration im Benutzerhandbuch AWS CloudFormation
- put-bucket-website in der AWS CLI Befehlsreferenz

Wenn Sie das Hosting statischer Websites für Ihren Bucket aktivieren, geben Sie den Namen des Fehlerdokuments ein (z. B, **`404.html`**). Nachdem Sie das statische Website-Hosting für den Bucket aktiviert haben, laden Sie eine HTML Datei mit diesem Fehlerdokumentnamen in Ihren Bucket hoch.

So konfigurieren Sie ein Fehlerdokument

1. Erstellen Sie ein Fehlerdokument, z. B. `404.html`.

2. Speichern Sie die Fehlerdokumentdatei lokal.

   Der Name des Fehlerdokuments unterscheidet zwischen Groß- und Kleinschreibung und muss genau mit dem Namen übereinstimmen, den Sie beim Aktivieren des statischen Website-Hostings eingeben. Wenn Sie beispielsweise `404.html` im Dialogfeld Hosten einer statischen Website als Namen des Fehlerdokuments eingeben, muss der Dateiname des Fehlerdokuments ebenfalls `404.html` lauten.
3. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
4. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie zum Hosten einer statischen Website verwenden möchten.
5. Aktivieren Sie das Hosting statischer Websites für Ihren Bucket und geben Sie den exakten Namen Ihres Fehlerdokuments ein (z. B, `404.html`). Weitere Informationen erhalten Sie unter Aktivieren des Website-Hostings und Konfigurieren eines benutzerdefinierten Fehlerdokuments.

   Fahren Sie mit Schritt 6 fort, nachdem Sie das Hosting statischer Websites aktiviert haben.
6. Führen Sie einen der folgenden Schritte aus, um das Fehlerdokument in Ihren Bucket hochzuladen:

   - Ziehen Sie die Fehlerdokumentdatei in das Konsolen-Bucket-Verzeichnis.
   - Wählen Sie Upload (Hochladen) und folgen Sie den Anweisungen zur Auswahl und zum Hochladen der Indexdatei.

   step-by-stepAnweisungen finden Sie unterObjekte hochladen.

Festlegen von Berechtigungen für den Website-Zugriff

Wenn Sie einen Bucket als statische Website konfigurieren und Ihre Website öffentlich sein soll, können Sie öffentlichen Lesezugriff gewähren. Um Ihren Bucket öffentlich lesbar zu machen, müssen Sie die Block-Einstellungen für den öffentlichen Zugriff für den Bucket deaktivieren und eine Bucket-Richtlinie schreiben, die öffentlichen Lesezugriff gewährt. Wenn Ihr Bucket Objekte enthält, die nicht dem Bucket-Besitzer gehören, müssen Sie möglicherweise auch eine Objektzugriffskontrollliste (ACL) hinzufügen, die allen Benutzern Lesezugriff gewährt.

Wenn Sie die Einstellungen zum Blockieren des öffentlichen Zugriffs für Ihren Bucket nicht deaktivieren möchten, Ihre Website aber trotzdem öffentlich zugänglich sein soll, können Sie eine

CloudFront Amazon-Distribution erstellen, um Ihre statische Website bereitzustellen. Weitere Informationen finden Sie unter Beschleunigen Sie Ihre Website mit Amazon CloudFront oder Verwenden einer CloudFront Amazon-Distribution zur Bereitstellung einer statischen Website im Amazon Route 53-Entwicklerhandbuch.

> **Note**
>
> Wenn ein Benutzer auf dem Endpunkt der Website ein Objekt anfordert, das nicht existiert, gibt Amazon S3 einen HTTP Antwortcode zurück`404 (Not Found)`. Wenn das Objekt existiert, Sie aber keine Leseberechtigung dafür erteilt haben, gibt der Website-Endpunkt den HTTP Antwortcode zurück`403 (Access Denied)`. Der Benutzer kann aus dem Antwortcode ableiten, ob ein bestimmtes Objekt existiert. Wenn Sie dieses Verhalten nicht wünschen, sollten Sie den Website-Support für Ihren Bucket nicht aktivieren.

Themen

- Schritt 1: Bearbeiten der S3 Block Public Access-Einstellungen
- Schritt 2: Hinzufügen einer Bucket-Richtlinie
- Objektzugriffskontrolllisten

Schritt 1: Bearbeiten der S3 Block Public Access-Einstellungen

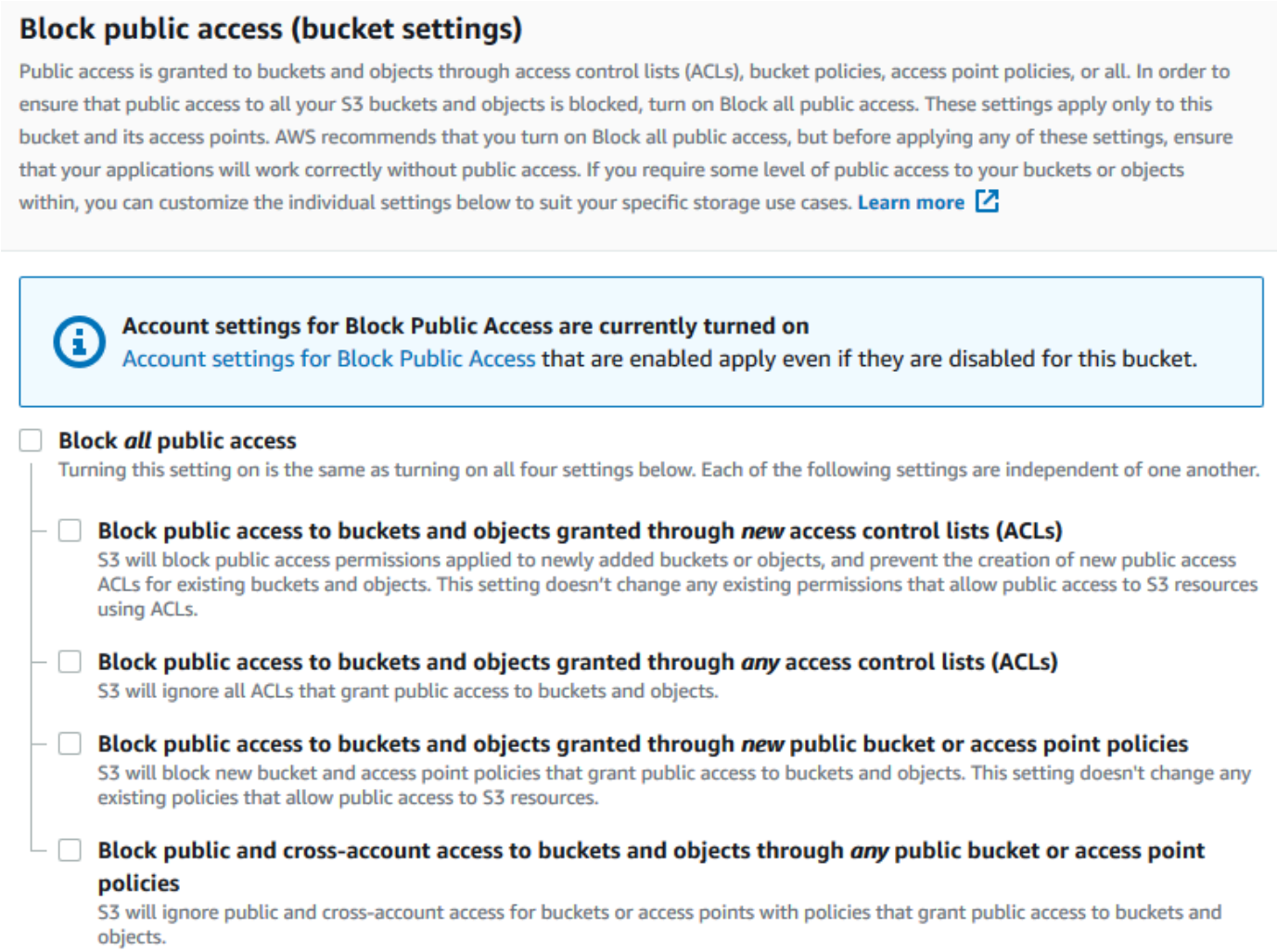
Wenn Sie einen bestehenden Bucket als statische Website konfigurieren möchten, die einen öffentlichen Zugriff bietet, müssen Sie die Block Public Access-Einstellungen für diesen Bucket bearbeiten. Möglicherweise müssen Sie auch Ihre Einstellungen für Block Public Access auf Kontoebene bearbeiten. Amazon S3 wendet die restriktivste Kombination der Block Public Access-Einstellungen auf Bucket- und Kontoebene an.

Wenn Sie beispielsweise den öffentlichen Zugriff für einen Bucket erlauben, aber den gesamten öffentlichen Zugriff auf Kontoebene blockieren, blockiert Amazon S3 den öffentlichen Zugriff auf den Bucket weiter. In diesem Szenario müssten Sie Ihre Block Public Access-Einstellungen auf Bucket- und Kontoebene bearbeiten. Weitere Informationen finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

Standardmäßig blockiert Amazon S3 den öffentlichen Zugriff auf Ihr Konto und Ihre Buckets. Wenn Sie einen Bucket verwenden möchten, um eine statische Website zu hosten, können Sie diese Schritte verwenden, um Ihre Einstellungen für Block Public Access zu bearbeiten:

> **Warning**
>
> Bevor Sie diese Schritte ausführen, sollten Sie sicherstellen, Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher dass Sie die Risiken verstehen und akzeptieren, die mit der Gewährung eines öffentlichen Zugriffs verbunden sind. Wenn Sie die Einstellungen für Block Public Access deaktivieren, um Ihren Bucket öffentlich zu machen, kann jeder im Internet auf Ihren Bucket zugreifen. Wir empfehlen Ihnen, den gesamten öffentlichen Zugriff auf Ihre Buckets zu blockieren.

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie den Namen des Buckets aus, den Sie als statische Website konfiguriert haben.
3. Wählen Sie Permissions (Berechtigungen).
4. Wählen Sie unter Block public access (bucket settings) (Öffentlichen Zugriff blockieren (Bucket-Einstellungen)), die Option Edit (Bearbeiten).
5. Löschen Sie Block all public access (Gesamten öffentlichen Zugriff blockieren) und wählen Sie Save (Speichern).



Amazon S3 deaktiviert die Einstellungen für den Block Public Access für Ihren Bucket. Um eine öffentliche statische Website zu erstellen, müssen Sie möglicherweise auch die Einstellungen zum Blockieren des öffentlichen Zugriffs für Ihr Konto bearbeiten, bevor Sie eine Bucket-Richtlinie hinzufügen. Wenn die Einstellungen „Öffentlichen Zugriff blockieren“ für Ihr Konto derzeit aktiviert sind, wird unter „Öffentlichen Zugriff blockieren“ (Bucket-Einstellungen) ein Hinweis angezeigt.

Schritt 2: Hinzufügen einer Bucket-Richtlinie

Um die Objekte in Ihrem Bucket öffentlich lesbar zu machen, müssen Sie eine Bucket-Richtlinie schreiben, die jedem die `s3:GetObject`-Berechtigung erteilt.

Nachdem Sie die Einstellungen für S3 Block Public Access bearbeitet haben, können Sie eine Bucket-Richtlinie hinzufügen, um öffentlichen Lesezugriff auf den Bucket zu gewähren. Wenn Sie öffentlichen Lesezugriff gewähren, kann jeder im Internet auf Ihren Bucket zugreifen.

> **Important**
>
> Die zuvor genannte Richtlinie ist nur ein Beispiel und erlaubt Vollzugriff auf die Inhalte Ihres Buckets. Bevor Sie mit diesem Schritt fortfahren, lesen Sie den Abschnitt Wie kann ich die Dateien in meinem Amazon-S3-Bucket sichern?, um sicherzustellen, dass Sie die bewährten Methoden zum Sichern der Dateien in Ihrem S3-Bucket und die Risiken in Zusammenhang mit der Gewährung von öffentlichem Zugriff kennen.

1. Wählen Sie unter Buckets den Namen Ihres Buckets aus.
2. Wählen Sie Permissions (Berechtigungen).
3. Wählen Sie unter Bucket Policy (Bucket-Richtlinie) Edit (Bearbeiten).
4. Um öffentlichen Lesezugriff auf Ihre Website zu gewähren, kopieren Sie die folgende Bucket-Richtlinie und fügen Sie sie in den Bucket policy editor (Bucket-Richtlinieneditor) ein.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::Bucket-Name/*"
            ]
        }
    ]
}
```

5. Aktualisieren Sie den `Resource` zu Ihrem Bucket-Namen.

Im obigen Beispiel für eine Bucket-Richtlinie *Bucket-Name* ist ein Platzhalter für den Bucket-Namen. Um diese Bucket-Richtlinie mit Ihrem eigenen Bucket zu verwenden, müssen Sie diesen Namen so aktualisieren, dass er mit Ihrem Bucket übereinstimmt.

6. Wählen Sie Save Changes (Änderungen speichern).

   Es wird eine Meldung angezeigt, die darauf hinweist, dass die Bucket-Richtlinie erfolgreich hinzugefügt wurde.

   Wenn die Fehlermeldung `Policy has invalid resource` angezeigt wird, bestätigen Sie, dass der Bucket-Name in der Bucket-Richtlinie mit Ihrem Bucket-Namen übereinstimmt. Informationen zum Hinzufügen einer Bucket-Richtlinie finden Sie unter Wie füge ich eine S3-Bucket-Richtlinie hinzu?

   Wenn Sie eine Fehlermeldung erhalten und die Bucket-Richtlinie nicht speichern können, überprüfen Sie Ihr Konto und die Bucket-Einstellungen für Block Public Access, um zu bestätigen, dass Sie den öffentlichen Zugriff auf den Bucket zulassen.

Objektzugriffskontrolllisten

Sie können eine Bucket-Richtlinie verwenden, um Ihren Objekten öffentliche Leserechte zu erteilen. Die Bucket-Richtlinie gilt jedoch nur für Objekte, die sich im Besitz des Bucket-Eigentümers befinden. Wenn Ihr Bucket Objekte enthält, die nicht dem Bucket-Besitzer gehören, sollte der Bucket-Besitzer die Objektzugriffskontrollliste (ACL) verwenden, um öffentliche READ Berechtigungen für diese Objekte zu erteilen.

S3 Object Ownership ist eine Einstellung auf Amazon S3 S3-Bucket-Ebene, mit der Sie sowohl das Eigentum an den Objekten kontrollieren können, die in Ihren Bucket hochgeladen werden, als auch um sie zu deaktivieren oder zu aktivieren. ACLs Standardmäßig ist für Object Ownership die erzwungene Einstellung des Bucket-Besitzers festgelegt, und alle sind ACLs deaktiviert. Wenn ACLs diese Option deaktiviert ist, besitzt der Bucket-Besitzer alle Objekte im Bucket und verwaltet den Zugriff darauf ausschließlich mithilfe von Zugriffsverwaltungsrichtlinien.

Für die meisten modernen Anwendungsfälle in Amazon S3 ist die Verwendung von nicht mehr erforderlichACLs. Wir empfehlen, die ACLs Option deaktiviert zu lassen, außer in Ausnahmefällen, in denen Sie den Zugriff für jedes Objekt einzeln steuern müssen. Wenn diese ACLs Option deaktiviert ist, können Sie mithilfe von Richtlinien den Zugriff auf alle Objekte in Ihrem Bucket steuern,

unabhängig davon, wer die Objekte in Ihren Bucket hochgeladen hat. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

> **Important**
>
> Wenn Ihr Bucket die Einstellung „Vom Bucket-Eigentümer erzwungen“ für S3 Object Ownership verwendet, müssen Sie Richtlinien verwenden, um Zugriff auf Ihren Bucket und die darin enthaltenen Objekte zu gewähren. Wenn die Einstellung „Bucket Owner erforced“ aktiviert ist, schlagen Anfragen zur Einrichtung von Zugriffskontrolllisten (ACLs) oder zur Aktualisierung ACLs fehl und geben den `AccessControlListNotSupported` Fehlercode zurück. Leseanfragen ACLs werden weiterhin unterstützt.

Um ein Objekt mit einem öffentlich lesbar zu machenACL, READ erteilen Sie der `AllUsers` Gruppe die entsprechenden Berechtigungen, wie im folgenden Grant-Element dargestellt. Fügen Sie dem Objekt dieses Grant-Element hinzuACL. Informationen zur Verwaltung finden ACLs Sie unterÜbersicht über die Zugriffskontrollliste (ACL).

```
<Grant>
  <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:type="Group">
    <URI>http://acs.amazonaws.com/groups/global/AllUsers</URI>
  </Grantee>
  <Permission>READ</Permission>
</Grant>
```

(Optional) Protokollieren des Webdatenverkehrs

Sie können optional die Amazon-S3-Server-Zugriffsprotokollierung für einen Bucket aktivieren, der als statische Website konfiguriert ist. Die Server-Zugriffsprotokollierung bietet detaillierte Aufzeichnungen über die Anfragen, die an Ihren Bucket gestellt werden. Weitere Informationen finden Sie unter Protokollieren von Anfragen mit Server-Zugriffsprotokollierung. Wenn Sie Amazon verwenden möchten, CloudFront um Ihre Website zu beschleunigen, können Sie auch die CloudFront Protokollierung verwenden. Weitere Informationen finden Sie unter Configuring and Using Access Logs im Amazon CloudFront Developer Guide.

So aktivieren Sie die Server-Zugriffsprotokollierung für Ihren statischen Website-Bucket

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

2. Erstellen Sie in derselben Region, in der Sie den Bucket erstellt haben, der als statische Website konfiguriert ist, einen Bucket für die Protokollierung, z. B. `logs.example.com`.
3. Erstellen Sie einen Ordner für die Protokolldateien der Server-Zugriffsprotokollierung (z. B, `logs`).
4. (Optional) Wenn Sie die Leistung Ihrer Website verbessern CloudFront möchten, erstellen Sie einen Ordner für die CloudFront Protokolldateien (z. B.`cdn`).

   Weitere Informationen finden Sie unter Beschleunigen Sie Ihre Website mit Amazon CloudFront.
5. Wählen Sie in der Liste Buckets den Namen Ihres Buckets.
6. Wählen Sie Properties (Eigenschaften).
7. Wählen Sie unter Server access logging (Server-Zugriffsprotokollierung) Edit (Bearbeiten).
8. Wählen Sie Enable aus.
9. Wählen Sie im Ziel-Bucket den Bucket und das Ordnerziel für die Server-Zugriffsprotokolle aus:

   - Navigieren Sie zum Ordner- und Bucket-Speicherort:
     1. Wählen Sie Browse S3 (S3 durchsuchen).
     2. Wählen Sie den Bucket-Namen und dann den Ordner mit den Protokollen aus.
     3. Wählen Sie Choose path (Pfad wählen).
   - Geben Sie den S3-Bucket-Pfad ein, z. B, **`s3://logs.example.com/logs/`**.
10. Wählen Sie Save Changes (Änderungen speichern).

    In Ihrem Protokoll-Bucket können Sie jetzt auf Ihre Protokolle zugreifen. Amazon S3 schreibt Website-Zugriffsprotokolle alle zwei Stunden in Ihren Bucket zur Protokollierung.

(Optional) Konfigurieren einer Webseitenumleitung

Wenn Ihr Amazon-S3-Bucket für das statische Website-Hosting konfiguriert ist, können Sie Umleitungsregeln für Ihren Bucket oder die darin enthaltenen Objekte konfigurieren. Sie haben die folgenden Optionen für die Konfigurierung einer Umleitung.

Themen

- Umleiten von Anforderungen für den Website-Endpunkt Ihres Buckets an einen anderen Bucket oder eine andere Domäne
- Konfigurieren von Umleitungsregeln für die Verwendung von fortschrittliche bedingten Umleitungen

- So leiten Sie Anforderungen für ein Objekt um

Umleiten von Anforderungen für den Website-Endpunkt Ihres Buckets an einen anderen Bucket oder eine andere Domäne

Sie können alle Anforderungen für einen Website-Endpunkt eines Buckets an einen anderen Host umleiten. Wenn Sie alle Anforderungen umleiten, werden alle Anforderungen an den Website-Endpunkt an den angegebenen Bucket oder die Domäne umgeleitet.

Wenn Ihre Root-Domäne zum Beispiel `example.com` ist, und Sie Anfragen sowohl für `http://example.com`, als auch für `http://www.example.com` bereitstellen möchten, müssen Sie zwei Buckets mit den Namen `example.com` und `www.example.com` erstellen. Pflegen Sie dann den Inhalt im Bucket `example.com` und konfigurieren Sie den anderen Bucket `www.example.com` so, dass alle Anforderungen an den Bucket `example.com` umgeleitet werden. Weitere Informationen finden Sie unter Konfigurieren einer statischen Website mit einem benutzerdefinierten Domänennamen.

So leiten Sie Anforderungen für einen Bucket-Website-Endpunkt um

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie unter Buckets den Namen des Buckets aus, von dem Sie Anfragen umleiten möchten (z. B. `www.example.com`).
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie unter Static website hosting (Hosting statischer Websites) Edit (Bearbeiten) aus.
5. Wählen Sie Redirect requests for an object (Anfragen für ein Objekt umleiten).
6. Geben Sie im Feld Host name (Host-Name) den Website-Endpunkt für Ihren Bucket oder Ihre benutzerdefinierte Domäne ein.

   Wenn Sie Anforderungen beispielsweise zu einer Root-Domänenadresse umleiten, geben Sie ei **`example.com`**.
7. Wählen Sie unter Protocol (Protokoll) das Protokoll für die umgeleiteten Anfragen (none (kein), http oder https).

   Wenn Sie kein Protokoll angeben, ist die Standardoption none (kein).
8. Wählen Sie Save Changes (Änderungen speichern).

Konfigurieren von Umleitungsregeln für die Verwendung von fortschrittliche bedingten Umleitungen

Unter Verwendung fortschrittliche Umleitungsregeln können Sie bedingt Anfragen abhängig von bestimmten Objektschlüsselnamen oder Präfixen in der Anfrage weiterleiten, oder abhängig von Antwortcodes. Angenommen, Sie löschen ein Objekt in Ihrem Bucket oder benennen es um. Sie können eine Weiterleitungsregel hinzufügen, die die Anfrage an ein anderes Objekt weiterleitet. Wenn Sie einen Ordner nicht zur Verfügung stellen wollen, können Sie eine Umleitungsregel hinzufügen, um die Anfrage an eine andere Webseite umzuleiten. Sie können auch eine Weiterleitungsregel hinzufügen, um Fehlerbedingungen zu verarbeiten, indem Sie Anfragen, die den Fehler zurückgeben, an eine andere Domäne weiterleiten, wenn der Fehler verarbeitet wird.

Wenn Sie das statische Website-Hosting für Ihren Bucket aktivieren, können Sie optional fortschrittliche Umleitungsregeln angeben. Amazon S3 hat eine Begrenzung von 50 Routingregeln pro Websitekonfiguration. Wenn Sie mehr als 50 Routingregeln benötigen, können Sie die Objektumleitung verwenden. Weitere Informationen finden Sie unter Verwenden der S3-Konsole.

Weitere Informationen zur Konfiguration von Routing-Regeln mithilfe von finden Sie PutBucketWebsitein der Amazon Simple Storage Service API Reference. REST API

> **Important**
>
> Um Umleitungsregeln in der neuen Amazon S3 S3-Konsole zu erstellen, müssen Sie JSON JSONBeispiele finden Sie unterBeispiele für Umleitungsregeln.

So konfigurieren Sie Umleitungsregeln für eine statische Website

Gehen Sie folgendermaßen vor, um Umleitungsregeln für einen Bucket hinzuzufügen, für den das statische Website-Hosting bereits aktiviert ist.

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets (Buckets) den Namen eines Buckets aus, den Sie als statische Website konfiguriert haben.
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie unter Static website hosting (Hosting statischer Websites) Edit (Bearbeiten) aus.
5. Geben Sie im Feld Umleitungsregeln Ihre Umleitungsregeln ein. JSON

In der S3-Konsole beschreiben Sie die Regeln mithilfe von. JSON JSONBeispiele finden Sie unterBeispiele für Umleitungsregeln. Amazon S3 hat eine Begrenzung von 50 Routingregeln pro Websitekonfiguration.

6. Wählen Sie Save Changes (Änderungen speichern).

Routingregel-Elemente

Im Folgenden finden Sie die allgemeine Syntax für die Definition der Routing-Regeln in einer Website-Konfiguration unter JSON und XML Um Umleitungsregeln in der neuen S3-Konsole zu konfigurieren, müssen Sie Folgendes verwendenJSON. JSONBeispiele finden Sie unterBeispiele für Umleitungsregeln.

JSON

```
[
    {
      "Condition": {
        "HttpErrorCodeReturnedEquals": "string",
        "KeyPrefixEquals": "string"
      },
      "Redirect": {
        "HostName": "string",
        "HttpRedirectCode": "string",
        "Protocol": "http"|"https",
        "ReplaceKeyPrefixWith": "string",
        "ReplaceKeyWith": "string"
      }
    }
  ]

Note: Redirect must each have at least one child element. You can have either
 ReplaceKeyPrefix with or ReplaceKeyWith but not both.
```

XML

```
<RoutingRules> =
    <RoutingRules>
         <RoutingRule>...</RoutingRule>
         [<RoutingRule>...</RoutingRule>
         ...]
```

```
    </RoutingRules>

<RoutingRule> =
   <RoutingRule>
      [ <Condition>...</Condition> ]
      <Redirect>...</Redirect>
   </RoutingRule>

<Condition> =
   <Condition>
      [ <KeyPrefixEquals>...</KeyPrefixEquals> ]
      [ <HttpErrorCodeReturnedEquals>...</HttpErrorCodeReturnedEquals> ]
   </Condition>
    Note: <Condition> must have at least one child element.

<Redirect> =
   <Redirect>
      [ <HostName>...</HostName> ]
      [ <Protocol>...</Protocol> ]
      [ <ReplaceKeyPrefixWith>...</ReplaceKeyPrefixWith>  ]
      [ <ReplaceKeyWith>...</ReplaceKeyWith> ]
      [ <HttpRedirectCode>...</HttpRedirectCode> ]
   </Redirect>

Note: <Redirect> must have at least one child element. You can have either
 ReplaceKeyPrefix with or ReplaceKeyWith but not both.
```

In der folgenden Tabelle werden die Elemente in der Weiterleitungsregel beschrieben.

Name	Beschreibung
`RoutingRules`	Container für eine Sammlung von `RoutingRule` -Elementen.
`RoutingRule`	Eine Regel, die eine Bedingung identifiziert, sowie die Weiterlei tung, die angewendet wird, wenn die Bedingung erfüllt ist. Bedingung: • Ein `RoutingRules` -Container muss mindestens eine Weiterleitungsregel enthalten.

Name	Beschreibung
`Condition`	Container für die Beschreibung einer Bedingung, die für die angegebene Weiterleitung erfüllt sein muss, damit sie angewende t wird. Wenn die Weiterleitungsregel keine Bedingung enthält, wird die Regel auf alle Anfragen angewendet.
`KeyPrefixEquals`	Das Präfix des Objektschlüsselnamens, von dem die Anfragen weitergeleitet werden. `KeyPrefixEquals` ist erforderliche, wenn `HttpError CodeReturnedEquals` nicht angegeben ist. Wenn `KeyPrefixEquals` und `HttpErrorCodeReturnedEquals` angegeben sind, müssen beide true sein, damit die Bedingung erfüllt ist.
`HttpErrorCodeRetur nedEquals`	Der HTTP Fehlercode, der übereinstimmen muss, damit die Umleitung angewendet wird. Wenn ein Fehler auftritt und der Fehlercode mit diesem Wert übereinstimmt, gilt die angegebene Weiterleitung. `HttpErrorCodeReturnedEquals` ist erforderliche, wenn `KeyPrefixEquals` nicht angegeben ist. Wenn `KeyPrefix Equals` und `HttpErrorCodeReturnedEquals` angegeben sind, müssen beide true sein, damit die Bedingung erfüllt ist.
`Redirect`	Container-Element, das Anweisungen für die Weiterleitung der Anfrage enthält. Sie können Anfragen an einen anderen Host oder eine andere Seite umleiten, oder ein anderes Protokoll vorgeben. Eine `RoutingRule` muss ein `Redirect`-Element besitzen. Ein `Redirect`-Element muss mindestens eines der folgenden zugeordneten Elemente enthalten: `Protocol`, `HostName`, `ReplaceKeyPrefixWith` , `ReplaceKeyWith` oder `HttpRedirectCode` .

Name	Beschreibung
`Protocol`	Das Protokoll, `http` oder `https`, das im `Location`-Header verwendet werden soll, der in der Antwort zurückgegeben wird. Wenn eines der zugeordneten Elemente vorhanden ist, ist `Protocol` nicht erforderlich.
`HostName`	Der Hostname, der im `Location`-Header verwendet werden soll, der in der Antwort zurückgegeben wird. Wenn eines der zugeordneten Elemente vorhanden ist, ist `HostName` nicht erforderlich.
`ReplaceKeyPrefixWith`	Das Präfix des Objektschlüsselnamens, der den Wert von `KeyPrefixEquals` in der Umleitungsanforderung ersetzt. Wenn eines der zugeordneten Elemente vorhanden ist, ist `ReplaceKeyPrefixWith` nicht erforderlich. Es kann nur bereitgestellt werden, wenn `ReplaceKeyWith` nicht bereitgestellt wird.
`ReplaceKeyWith`	Der Objektschlüssel, der im `Location`-Header verwendet werden soll und in der Antwort zurückgegeben wird. Wenn eines der zugeordneten Elemente vorhanden ist, ist `ReplaceKeyWith` nicht erforderlich. Es kann nur bereitgestellt werden, wenn `ReplaceKeyPrefixWith` nicht bereitgestellt wird.
`HttpRedirectCode`	Der HTTP Umleitungscode, der in dem `Location` Header verwendet werden soll, der in der Antwort zurückgegeben wird. Wenn eines der zugeordneten Elemente vorhanden ist, ist `HttpRedirectCode` nicht erforderlich.

Beispiele für Umleitungsregeln

Die folgenden Beispiele erklären häufige Weiterleitungsfälle:

> **Important**
>
> Um Umleitungsregeln in der neuen Amazon S3 S3-Konsole zu erstellen, müssen Sie JSON

Example 1: Weiterleitung nach der Umbenennung eines Schlüsselpräfix

Angenommen, Ihr Bucket enthält die folgenden Objekte:

- index.html
- docs/article1.html
- docs/article2.html

Jetzt wollen Sie den Ordner von `docs/` in `documents/` umbenennen. Nach dieser Änderung müssen Sie Anfragen für das Präfix `docs/` in `documents/` weiterleiten. Beispielsweise muss die Anfragen für `docs/article1.html` an `documents/article1.html` weitergeleitet werden.

In diesem Fall fügen Sie der Konfiguration der Website die folgende Routingregel hinzu.

JSON

```
[
    {
        "Condition": {
            "KeyPrefixEquals": "docs/"
        },
        "Redirect": {
            "ReplaceKeyPrefixWith": "documents/"
        }
    }
]
```

XML

```
  <RoutingRules>
```

```
    <RoutingRule>
    <Condition>
      <KeyPrefixEquals>docs/</KeyPrefixEquals>
    </Condition>
    <Redirect>
      <ReplaceKeyPrefixWith>documents/</ReplaceKeyPrefixWith>
    </Redirect>
    </RoutingRule>
  </RoutingRules>
```

Example 2: Weiterleitung von Anfragen für einen gelöschten Ordner auf eine Seite

Angenommen, Sie löschen den Ordner `images/` (d. h. Sie löschen alle Objekte mit dem Schlüsselpräfix `images/`). Sie können eine Weiterleitungsregel einrichten, die Anfrage für jedes Objekt mit dem Schlüsselpräfix `images/` auf eine Seite namens `folderdeleted.html` weiterleitet.

JSON

```
[
    {
        "Condition": {
            "KeyPrefixEquals": "images/"
        },
        "Redirect": {
            "ReplaceKeyWith": "folderdeleted.html"
        }
    }
]
```

XML

```
  <RoutingRules>
    <RoutingRule>
    <Condition>
       <KeyPrefixEquals>images/</KeyPrefixEquals>
    </Condition>
    <Redirect>
      <ReplaceKeyWith>folderdeleted.html</ReplaceKeyWith>
    </Redirect>
    </RoutingRule>
  </RoutingRules>
```

Example 3: Bei einem HTTP Fehler weiterleiten

Nehmen wir an, wenn ein angefordertes Objekt nicht gefunden wird, möchten Sie Anfragen an eine Amazon Elastic Compute Cloud (AmazonEC2) -Instance umleiten. Fügen Sie eine Umleitungsregel hinzu, sodass der Besucher der Website, wenn der HTTP Statuscode 404 (Not Found) zurückgegeben wird, zu einer EC2 Amazon-Instance weitergeleitet wird, die die Anfrage bearbeitet.

Im folgenden Beispiel wird auch das Objektschlüsselpräfix `report-404/` in die Weiterleitung eingefügt. Wenn Sie beispielsweise eine Seite anfordern `ExamplePage.html` und dies zu einem HTTP 404-Fehler führt, wird die Anfrage auf eine Seite `report-404/ExamplePage.html` auf der angegebenen EC2 Amazon-Instance umgeleitet. Wenn es keine Routing-Regel gibt und der HTTP Fehler 404 auftritt, wird das in der Konfiguration angegebene Fehlerdokument zurückgegeben.

JSON

```
[
    {
        "Condition": {
            "HttpErrorCodeReturnedEquals": "404"
        },
        "Redirect": {
            "HostName": "ec2-11-22-333-44.compute-1.amazonaws.com",
            "ReplaceKeyPrefixWith": "report-404/"
        }
    }
]
```

XML

```
  <RoutingRules>
    <RoutingRule>
    <Condition>
      <HttpErrorCodeReturnedEquals>404</HttpErrorCodeReturnedEquals >
    </Condition>
    <Redirect>
      <HostName>ec2-11-22-333-44.compute-1.amazonaws.com</HostName>
      <ReplaceKeyPrefixWith>report-404/</ReplaceKeyPrefixWith>
    </Redirect>
    </RoutingRule>
  </RoutingRules>
```

So leiten Sie Anforderungen für ein Objekt um

Sie können Anfragen für ein Objekt an ein anderes Objekt umleiten oder URL indem Sie den Ort der Website-Weiterleitung in den Metadaten des Objekts festlegen. Sie richten die Umleitung ein, indem Sie den Objekt-Metadaten die `x-amz-website-redirect-location`-Eigenschaft hinzufügen. In der Amazon-S3-Konsole legen Sie den Wert für Website Redirect Location (Websiteumleitungsort) in den Metadaten des Objekts fest. Wenn Sie Amazon S3 verwendenAPI, legen Sie fest`x-amz-website-redirect-location`. Die Website interpretiert anschließend das Objekt als eine 301-Umleitung.

Um eine Anfrage an ein anderes Objekt umzuleiten, richten Sie den Umleitungsstandort auf den Schlüssel für das Zielobjekt ein. Um eine Anfrage an eine externe Adresse weiterzuleitenURL, legen Sie den Umleitungsort auf den gewünschten Wert URL fest. Weitere Informationen zu Objekt-Metadaten erhalten Sie unter Systemdefinierte Objektmetadaten.

Wenn Sie eine Seitenumleitung einrichten, können Sie den Inhalt des Quellobjekts beibehalten oder löschen. Wenn es in Ihrem Bucket beispielsweise das Objekt `page1.html` gibt, können Sie alle Anforderungen für diese Seite an ein anderes Objekt umleiten, `page2.html`. Sie haben hierfür zwei Möglichkeiten:

- Behalten Sie den Inhalt des Objekts `page1.html` und leiten Sie Seitenanforderungen um.
- Löschen Sie den Inhalt von `page1.html` und laden Sie ein Null-Byte-Objekt mit dem Namen `page1.html` hoch, um das vorhandene Objekt zu ersetzen und Seitenanforderungen umzuleiten.

Verwenden der S3-Konsole

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie als statische Website konfiguriert haben (z. B. `example.com`).
3. Wählen Sie unter Objects (Objekte) Ihr Objekt aus.
4. Wählen Sie Actions (Aktionen) und dann Edit metadata (Metadaten bearbeiten).
5. Wählen Sie Metadata (Metadaten) aus.
6. Wählen Sie Add Metadata (Metadaten hinzufügen) aus.
7. Wählen Sie unter Type (Typ) die Option System Defined (Systemdefiniert) aus.
8. Wählen Sie unter Key die Option x-amz-website-redirect-location aus.

9. Geben Sie in Value (Wert) den Schlüsselnamen des Objekts ein, zu dem Sie umleiten möchten, z. B. `/page2.html`.

   Für ein anderes Objekt im selben Bucket ist das Präfix `/` im Wert erforderlich. Sie können den Wert auch auf einen externen Wert festlegenURL, `http://www.example.com` z. B.
10. Wählen Sie Metadaten bearbeiten.

Mit dem REST API

Die folgenden Amazon S3 API S3-Aktionen unterstützen den `x-amz-website-redirect-location` Header in der Anfrage. Amazon S3 speichert den Header-Wert in den Objekt-Metadaten als `x-amz-website-redirect-location`.

- PUTObjekt
- Initiieren eines mehrteiligen Uploads
- POSTObjekt
- PUTObjekt — Kopieren

Ein für das Hosting von Websites konfigurierter Bucket hat sowohl den Website-Endpunkt als auch den REST Endpunkt. Eine Anfrage für eine Seite, die als 301-Umleitung konfiguriert ist, erzeugt die folgenden möglichen Ergebnisse, abhängig vom Endpunkt der Anfrage:

- Regionsspezifischer Website-Endpunkt – Amazon S3 leitet die Seitenanforderung in Übereinstimmung mit dem Wert der `x-amz-website-redirect-location`-Eigenschaft um.
- RESTEndpunkt — Amazon S3 leitet die Seitenanfrage nicht weiter. Es gibt das angefragte Objekt zurück.

Weitere Informationen zu den Endpunkten finden Sie unter Hauptunterschiede zwischen einem Website-Endpunkt und einem REST API Endpunkt.

Wenn Sie eine Seitenumleitung einrichten, können Sie den Inhalt des Objekts beibehalten oder löschen. Nehmen wir z. B. an, dass Sie ein `page1.html`-Objekt in Ihrem Bucket haben.

- Um den Inhalt von Seitenanfragen beizubehalten `page1.html` und nur Seitenanfragen umzuleiten, können Sie eine PUTObject-Copy-Anfrage einreichen, um ein neues `page1.html` Objekt zu erstellen, das das bestehende `page1.html` Objekt als Quelle verwendet. In Ihrer Anfrage richten Sie den `x-amz-website-redirect-location`-Header ein. Nachdem die

Anforderung abgeschlossen ist, ist die Originalseite mit ihrem Inhalt unverändert, aber Amazon S3 leitet alle Anforderungen für die Seite an den von Ihnen angegebenen Umleitungsstandort um.

- Um den Inhalt des `page1.html` Objekts zu löschen und Anfragen für die Seite umzuleiten, können Sie eine PUT Objektanforderung senden, um ein Null-Byte-Objekt hochzuladen, das denselben Objektschlüssel hat:. `page1.html` In der PUT Anfrage legen Sie `x-amz-website-redirect-location` für das neue `page1.html` Objekt fest. Nachdem die Anfrage abgeschlossen ist, hat `page1.html` keinen Inhalt mehr, und Anfragen werden an den Standort umgeleitet, der in `x-amz-website-redirect-location` angegeben ist.

Wenn Sie das Objekt mithilfe der GETObjekt-Aktion zusammen mit anderen Objektmetadaten abrufen, gibt Amazon S3 den `x-amz-website-redirect-location` Header in der Antwort zurück.

Verwendung von ursprungsübergreifendem Ressourcenaustausch () CORS

Cross-Origin Resource Sharing (CORS) definiert eine Möglichkeit, wie Client-Webanwendungen, die in eine Domain geladen sind, mit Ressourcen in einer anderen Domain interagieren können. Mit CORS Support können Sie umfangreiche clientseitige Webanwendungen mit Amazon S3 erstellen und selektiv den ursprungsübergreifenden Zugriff auf Ihre Amazon S3 S3-Ressourcen zulassen.

Dieser Abschnitt bietet einen Überblick über. CORS In den Unterthemen wird beschrieben, wie Sie die Aktivierung CORS mithilfe der Amazon S3-Konsole oder programmgesteuert mithilfe von Amazon S3 REST API und dem durchführen können. AWS SDKs

Cross-Origin Resource Sharing: Szenarien in Anwendungsfällen

Im Folgenden finden Sie Beispielszenarien für die Verwendung von. CORS

Szenario 1

Szenario 1: Angenommen, Sie hosten eine Website in einem Amazon-S3-Bucket mit dem Namen `website`, wie in Hosten einer statischen Website mit Amazon S3 beschrieben. Ihre Benutzer laden den Website-Endpunkt:

```
http://website.s3-website.us-east-1.amazonaws.com
```

Jetzt möchten Sie JavaScript auf den Webseiten, die in diesem Bucket gespeichert sind, verwenden, um authentifizierte PUT Anfragen GET und Anfragen für denselben Bucket stellen zu können, indem Sie den Amazon S3 API S3-Endpunkt für den Bucket verwenden. `website.s3.us-east-1.amazonaws.com` Ein Browser würde diese Anfragen normalerweise blockieren, aber mit können CORS Sie Ihren Bucket so konfigurieren, dass Anfragen JavaScript aus verschiedenen Quellen explizit aktiviert werden. `website.s3-website.us-east-1.amazonaws.com`

Szenario 2

Angenommen, Sie möchten eine Web-Schriftart aus Ihrem S3-Bucket hosten. Auch hier benötigen Browser eine CORS Prüfung (auch Preflight-Prüfung genannt), um Webfonts laden zu können. Sie würden den Bucket, der die Web-Schriftart hostet, deshalb so konfigurieren, dass jeder Ursprung diese Anfragen machen kann.

Wie bewertet Amazon S3 die CORS Konfiguration eines Buckets?

Wenn Amazon S3 eine Preflight-Anfrage von einem Browser empfängt, bewertet es die CORS Konfiguration für den Bucket und verwendet die erste `CORSRule` Regel, die der eingehenden Browseranfrage entspricht, um eine ursprungsübergreifende Anfrage zu aktivieren. Für die Übereinstimmung mit einer Regel müssen die folgenden Bedingungen erfüllt sei:

- Der `Origin` Header in einer CORS Anfrage an Ihren Bucket muss mit den Ursprüngen im `AllowedOrigins` Element in Ihrer Konfiguration übereinstimmen. CORS
- Die HTTP Methoden, die `Access-Control-Request-Method` in einer CORS Anfrage an Ihren Bucket angegeben sind, müssen mit der oder den Methoden übereinstimmen, die in dem `AllowedMethods` Element in Ihrer CORS Konfiguration aufgeführt sind.
- Die im Header einer Pre-Flight-Anfrage aufgeführten `Access-Control-Request-Headers` Header müssen mit den Headern im `AllowedHeaders` Element in Ihrer Konfiguration übereinstimmen. CORS

> **Note**
>
> Die Richtlinien ACLs und gelten weiterhin, wenn Sie sie in Ihrem Bucket aktivierenCORS.

So unterstützt Object Lambda Access Point CORS

Wenn S3 Object Lambda eine Anforderung von einem Browser empfängt oder die Anforderung einen `Origin`-Header enthält, fügt S3 Object Lambda immer das Header-Feld `“AllowedOrigins":"*"` hinzu.

Weitere Informationen zur Verwendung CORS finden Sie in den folgenden Themen.

Themen

- Elemente einer CORS Konfiguration
- Konfiguration der ursprungsübergreifenden gemeinsamen Nutzung von Ressourcen () CORS
- Testen CORS
- Fehlerbehebung für CORS

Elemente einer CORS Konfiguration

Um Ihren Bucket so zu konfigurieren, dass ursprungsübergreifende Anfragen zulässig sind, erstellen Sie eine CORS Konfiguration. Bei der CORS Konfiguration handelt es sich um ein Dokument mit Elementen, die die Ursprünge angeben, denen Sie den Zugriff auf Ihren Bucket gestatten, die Operationen (HTTPMethoden), die Sie für jeden Ursprung unterstützen werden, und andere vorgangsspezifische Informationen. Sie können der Konfiguration bis zu 100 Regeln hinzufügen. Sie können die CORS Konfiguration als `cors` Unterressource zum Bucket hinzufügen.

Wenn Sie CORS in der S3-Konsole konfigurieren, müssen Sie diese verwenden, um eine CORS Konfiguration JSON zu erstellen. Die neue S3-Konsole unterstützt nur JSON CORS Konfigurationen.

Weitere Informationen zur CORS Konfiguration und den darin enthaltenen Elementen finden Sie in den folgenden Themen. Anweisungen zum Hinzufügen einer CORS Konfiguration finden Sie unterKonfiguration der ursprungsübergreifenden gemeinsamen Nutzung von Ressourcen () CORS.

> **Important**
>
> In der S3-Konsole muss die CORS Konfiguration seinJSON.

Themen

- AllowedMethods-Element
- AllowedOrigins-Element

- AllowedHeaders-Element
- ExposeHeaders-Element
- MaxAgeSeconds-Element
- Beispiele für Konfigurationen CORS

`AllowedMethods`-Element

In der CORS Konfiguration können Sie die folgenden Werte für das `AllowedMethods` Element angeben.

- GET
- PUT
- POST
- DELETE
- HEAD

`AllowedOrigins`-Element

Im `AllowedOrigins`-Element geben Sie die Ursprünge an, über die Sie domänenübergreifende Anfragen erlauben möchten, z. B. `http://www.example.com`. Die Ursprungszeichenfolge darf nur ein *-Platzhalterzeichen enthalten, wie beispielsweise `http://*.example.com`. Optional können Sie * als Ursprung angeben, sodass alle Ursprünge ursprungsübergreifende Anfragen senden dürfen. Sie können auch `https` angeben, um nur sichere Ursprünge zuzulassen.

`AllowedHeaders`-Element

Das `AllowedHeaders`-Element gibt an, welche Header in einer Preflight-Anfrage durch den `Access-Control-Request-Headers`-Header erlaubt sind. Jeder Header-Name in der `Access-Control-Request-Headers` Kopfzeile muss mit einem entsprechenden Eintrag im Element übereinstimmen. Amazon S3 sendet nur die zulässigen angeforderten Header in einer Antwort. Eine Beispielliste von Headern, die in Anfragen an Amazon S3 verwendet werden können, finden Sie im APIReferenzhandbuch zu Amazon Simple Storage Service unter Allgemeine Anforderungsheader.

Jede AllowedHeaders Zeichenfolge in Ihrer Konfiguration kann höchstens ein Platzhalterzeichen (*) enthalten. Beispielsweise aktiviert `<AllowedHeader>x-amz-*</AllowedHeader>` alle für Amazon spezifischen Header.

`ExposeHeaders`-Element

Jedes `ExposeHeader` Element identifiziert einen Header in der Antwort, auf den Kunden von ihren Anwendungen (z. B. von einem JavaScript `XMLHttpRequest` Objekt) aus zugreifen können sollen. Eine Liste gängiger Amazon S3 S3-Antwortheader finden Sie unter Common Response Headers im APIReferenzhandbuch zu Amazon Simple Storage Service.

`MaxAgeSeconds`-Element

Das `MaxAgeSeconds` Element gibt die Zeit in Sekunden an, in der Ihr Browser die Antwort für eine Preflight-Anfrage zwischenspeichern kann, wie durch die Ressource, die HTTP Methode und den Ursprung identifiziert.

Beispiele für Konfigurationen CORS

Statt über einen Amazon-S3-Website-Endpunkt auf eine Website zuzugreifen, können Sie Ihre eigen Domäne verwenden, wie beispielsweise `example1.com`, um Ihren Inhalt bereitzustellen. Weitere Informationen zur Verwendung Ihrer eigenen Domäne finden Sie unter Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne.

Die folgende CORS Beispielkonfiguration hat drei Regeln, die als `CORSRule` Elemente angegeben sind:

- Die erste Regel erlaubt ursprungsübergreifendePUT,POST, und DELETE Anfragen vom `http://www.example1.com` Ursprung aus. Die Regel erlaubt auch, dass alle Header in einer OPTIONS Preflight-Anfrage den Header verwenden. `Access-Control-Request-Headers` Als Antwort auf OPTIONS Preflight-Anfragen gibt Amazon S3 angeforderte Header zurück.
- Die zweite Regel gestattet dieselben ursprungsübergreifenden Anfragen wie die erste Regel, aber sie bezieht sich auf einen anderen Ursprung, `http://www.example2.com`.
- Die dritte Regel erlaubt ursprungsübergreifende GET Anfragen aller Herkunft. Das Platzhalterzeichen `*` bezieht sich auf alle Ursprünge.

JSON

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
```

```
        "AllowedMethods": [
            "PUT",
            "POST",
            "DELETE"
        ],
        "AllowedOrigins": [
            "http://www.example1.com"
        ],
        "ExposeHeaders": []
    },
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "PUT",
            "POST",
            "DELETE"
        ],
        "AllowedOrigins": [
            "http://www.example2.com"
        ],
        "ExposeHeaders": []
    },
    {
        "AllowedHeaders": [],
        "AllowedMethods": [
            "GET"
        ],
        "AllowedOrigins": [
            "*"
        ],
        "ExposeHeaders": []
    }
]
```

XML

```
<CORSConfiguration>
 <CORSRule>
   <AllowedOrigin>http://www.example1.com</AllowedOrigin>

   <AllowedMethod>PUT</AllowedMethod>
```

```
    <AllowedMethod>POST</AllowedMethod>
    <AllowedMethod>DELETE</AllowedMethod>

    <AllowedHeader>*</AllowedHeader>
  </CORSRule>
  <CORSRule>
    <AllowedOrigin>http://www.example2.com</AllowedOrigin>

    <AllowedMethod>PUT</AllowedMethod>
    <AllowedMethod>POST</AllowedMethod>
    <AllowedMethod>DELETE</AllowedMethod>

    <AllowedHeader>*</AllowedHeader>
  </CORSRule>
  <CORSRule>
    <AllowedOrigin>*</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
  </CORSRule>
</CORSConfiguration>
```

Die CORS Konfiguration erlaubt auch optionale Konfigurationsparameter, wie in der folgenden CORS Konfiguration gezeigt. In diesem Beispiel erlaubt die CORS Konfiguration ursprungsübergreifendePUT,POST, und DELETE Anfragen vom `http://www.example.com` Ursprung aus.

JSON

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "PUT",
            "POST",
            "DELETE"
        ],
        "AllowedOrigins": [
            "http://www.example.com"
        ],
        "ExposeHeaders": [
            "x-amz-server-side-encryption",
```

```
            "x-amz-request-id",
            "x-amz-id-2"
        ],
        "MaxAgeSeconds": 3000
    }
]
```

XML

```
<CORSConfiguration>
 <CORSRule>
   <AllowedOrigin>http://www.example.com</AllowedOrigin>
   <AllowedMethod>PUT</AllowedMethod>
   <AllowedMethod>POST</AllowedMethod>
   <AllowedMethod>DELETE</AllowedMethod>
   <AllowedHeader>*</AllowedHeader>
  <MaxAgeSeconds>3000</MaxAgeSeconds>
  <ExposeHeader>x-amz-server-side-encryption</
ExposeHeader>
  <ExposeHeader>x-amz-request-id</
ExposeHeader>
  <ExposeHeader>x-amz-id-2</ExposeHeader>
 </CORSRule>
</CORSConfiguration>
```

Das `CORSRule`-Element in der obigen Konfiguration beinhaltet die folgenden optionalen Elemente:

- `MaxAgeSeconds`— Gibt den Zeitraum in Sekunden (in diesem Beispiel 3000) an, für den der Browser eine Amazon S3 S3-Antwort auf eine OPTIONS Preflight-Anfrage für die angegebene Ressource zwischenspeichert. Durch die Zwischenspeicherung der Antwort muss der Browser keine Preflight-Anfragen an Amazon S3 senden, wenn die ursprüngliche Anfrage wiederholt werden soll.
- `ExposeHeaders`— Identifiziert die Antwort-Header (in diesem Beispiel, und`x-amz-id-2`) `x-amz-server-side-encryption``x-amz-request-id`, auf die Kunden von ihren Anwendungen aus zugreifen können (z. B. von einem Objekt aus). JavaScript `XMLHttpRequest`

Konfiguration der ursprungsübergreifenden gemeinsamen Nutzung von Ressourcen () CORS

Die ursprungsübergreifende gemeinsame Nutzung von Ressourcen (CORS) definiert eine Möglichkeit, wie Client-Webanwendungen, die in eine Domain geladen sind, mit Ressourcen in einer anderen Domain interagieren können. Mit CORS Support können Sie umfangreiche clientseitige Webanwendungen mit Amazon S3 erstellen und selektiv den ursprungsübergreifenden Zugriff auf Ihre Amazon S3 S3-Ressourcen zulassen.

In diesem Abschnitt erfahren Sie, wie Sie CORS mithilfe der Amazon S3-Konsole, Amazon S3 REST API und der aktivieren AWS SDKs. Um Ihren Bucket so zu konfigurieren, dass er ursprungsübergreifende Anfragen zulässt, fügen Sie dem Bucket eine CORS Konfiguration hinzu. Eine CORS Konfiguration ist ein Dokument, das Regeln definiert, die die Ursprünge, denen Sie Zugriff auf Ihren Bucket gewähren, die für jeden Ursprung unterstützten Operationen (HTTPMethoden) und andere vorgangsspezifische Informationen festlegen. In der S3-Konsole muss es sich bei der CORS Konfiguration um ein JSON Dokument handeln.

CORSKonfigurationen in JSON und finden Sie XML beispielsweise unterElemente einer CORS Konfiguration.

Verwenden der S3-Konsole

In diesem Abschnitt wird erklärt, wie Sie die Amazon S3 S3-Konsole verwenden, um einem S3-Bucket eine Cross-Origin-Konfiguration für die gemeinsame Nutzung von Ressourcen (CORS) hinzuzufügen.

Wenn Sie die Option für CORS den Bucket aktivieren, gelten weiterhin die Zugriffskontrolllisten (ACLs) und andere Richtlinien für Zugriffsberechtigungen.

> **Important**
>
> In der S3-Konsole muss die CORS Konfiguration seinJSON. Beispiele für CORS Konfigurationen in JSON und XML finden Sie unterElemente einer CORS Konfiguration.

Um eine CORS Konfiguration zu einem S3-Bucket hinzuzufügen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie eine Bucket-Richtlinie erstellen wollen.
3. Wählen Sie Permissions (Berechtigungen).
4. Wählen Sie im Abschnitt Cross-Origin Resource Sharing (CORS) die Option Bearbeiten aus.
5. Geben Sie im Textfeld des CORSKonfigurationseditors eine neue CORS Konfiguration ein oder kopieren Sie sie und fügen Sie sie ein, oder bearbeiten Sie eine vorhandene Konfiguration.

   Die CORS Konfiguration ist eine JSON Datei. Der Text, den Sie in den Editor eingeben, muss gültig seinJSON. Weitere Informationen finden Sie unter Elemente einer CORS Konfiguration.
6. Wählen Sie Save Changes (Änderungen speichern).

> **Note**
>
> Amazon S3 zeigt den Amazon-Ressourcennamen (ARN) für den Bucket neben dem Titel des CORSKonfigurationseditors an. Weitere Informationen zu ARNs finden Sie unter Amazon Resource Names (ARNs) und AWS Service Namespaces in der. Allgemeine Amazon Web Services-Referenz

Mit dem AWS SDKs

Sie können den verwenden AWS SDK, um die ursprungsübergreifende gemeinsame Nutzung von Ressourcen (CORS) für einen Bucket zu verwalten. Weitere Informationen zu finden Sie CORS unterVerwendung von ursprungsübergreifendem Ressourcenaustausch () CORS.

Im Folgenden sind einige Beispiele aufgeführt:

- Erstellt eine CORS Konfiguration und legt die Konfiguration für einen Bucket fest
- Ruft die Konfiguration ab und ändert sie durch Hinzufügen einer Regel ab
- Fügt die abgeänderte Konfiguration dem Bucket hinzu
- Löscht die Konfiguration

Java

Example

Example

Anweisungen zum Erstellen und Testen eines funktionierenden Beispiels finden Sie unter Erste Schritte im AWS SDK for Java Entwicklerhandbuch.

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.BucketCrossOriginConfiguration;
import com.amazonaws.services.s3.model.CORSRule;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class CORS {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String bucketName = "*** Bucket name ***";

        // Create two CORS rules.
        List<CORSRule.AllowedMethods> rule1AM = new
 ArrayList<CORSRule.AllowedMethods>();
        rule1AM.add(CORSRule.AllowedMethods.PUT);
        rule1AM.add(CORSRule.AllowedMethods.POST);
        rule1AM.add(CORSRule.AllowedMethods.DELETE);
        CORSRule rule1 = new
 CORSRule().withId("CORSRule1").withAllowedMethods(rule1AM)
                .withAllowedOrigins(Arrays.asList("http://*.example.com"));

        List<CORSRule.AllowedMethods> rule2AM = new
 ArrayList<CORSRule.AllowedMethods>();
        rule2AM.add(CORSRule.AllowedMethods.GET);
```

```
        CORSRule rule2 = new
CORSRule().withId("CORSRule2").withAllowedMethods(rule2AM)
                .withAllowedOrigins(Arrays.asList("*")).withMaxAgeSeconds(3000)
                .withExposedHeaders(Arrays.asList("x-amz-server-side-encryption"));

        List<CORSRule> rules = new ArrayList<CORSRule>();
        rules.add(rule1);
        rules.add(rule2);

        // Add the rules to a new CORS configuration.
        BucketCrossOriginConfiguration configuration = new
BucketCrossOriginConfiguration();
        configuration.setRules(rules);

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                    .withCredentials(new ProfileCredentialsProvider())
                    .withRegion(clientRegion)
                    .build();

            // Add the configuration to the bucket.
            s3Client.setBucketCrossOriginConfiguration(bucketName, configuration);

            // Retrieve and display the configuration.
            configuration = s3Client.getBucketCrossOriginConfiguration(bucketName);
            printCORSConfiguration(configuration);

            // Add another new rule.
            List<CORSRule.AllowedMethods> rule3AM = new
ArrayList<CORSRule.AllowedMethods>();
            rule3AM.add(CORSRule.AllowedMethods.HEAD);
            CORSRule rule3 = new
CORSRule().withId("CORSRule3").withAllowedMethods(rule3AM)
                    .withAllowedOrigins(Arrays.asList("http://www.example.com"));

            rules = configuration.getRules();
            rules.add(rule3);
            configuration.setRules(rules);
            s3Client.setBucketCrossOriginConfiguration(bucketName, configuration);

            // Verify that the new rule was added by checking the number of rules in
the
            // configuration.
            configuration = s3Client.getBucketCrossOriginConfiguration(bucketName);
```

```
            System.out.println("Expected # of rules = 3, found " +
 configuration.getRules().size());

            // Delete the configuration.
            s3Client.deleteBucketCrossOriginConfiguration(bucketName);
            System.out.println("Removed CORS configuration.");

            // Retrieve and display the configuration to verify that it was
            // successfully deleted.
            configuration = s3Client.getBucketCrossOriginConfiguration(bucketName);
            printCORSConfiguration(configuration);
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }

    private static void printCORSConfiguration(BucketCrossOriginConfiguration
 configuration) {
        if (configuration == null) {
            System.out.println("Configuration is null.");
        } else {
            System.out.println("Configuration has " +
 configuration.getRules().size() + " rules\n");

            for (CORSRule rule : configuration.getRules()) {
                System.out.println("Rule ID: " + rule.getId());
                System.out.println("MaxAgeSeconds: " + rule.getMaxAgeSeconds());
                System.out.println("AllowedMethod: " + rule.getAllowedMethods());
                System.out.println("AllowedOrigins: " + rule.getAllowedOrigins());
                System.out.println("AllowedHeaders: " + rule.getAllowedHeaders());
                System.out.println("ExposeHeader: " + rule.getExposedHeaders());
                System.out.println();
            }
        }
    }
}
```

.NET

Example

Informationen zum Einrichten und Ausführen der Codebeispiele finden Sie unter Erste Schritte mit dem AWS SDK for. NETim AWS SDKfür. NETLeitfaden für Entwickler.

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace Amazon.DocSamples.S3
{
    class CORSTest
    {
        private const string bucketName = "*** bucket name ***";
        // Specify your bucket region (an example region is shown).
        private static readonly RegionEndpoint bucketRegion =
 RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
            s3Client = new AmazonS3Client(bucketRegion);
            CORSConfigTestAsync().Wait();
        }
        private static async Task CORSConfigTestAsync()
        {
            try
            {
                // Create a new configuration request and add two rules
                CORSConfiguration configuration = new CORSConfiguration
                {
                    Rules = new System.Collections.Generic.List<CORSRule>
                        {
                          new CORSRule
                          {
                            Id = "CORSRule1",
                            AllowedMethods = new List<string> {"PUT", "POST",
 "DELETE"},
```

```
                            AllowedOrigins = new List<string> {"http://
*.example.com"}
                        },
                        new CORSRule
                        {
                          Id = "CORSRule2",
                          AllowedMethods = new List<string> {"GET"},
                          AllowedOrigins = new List<string> {"*"},
                          MaxAgeSeconds = 3000,
                          ExposeHeaders = new List<string> {"x-amz-server-side-
encryption"}
                        }
                      }
                };

                // Add the configuration to the bucket.
                await PutCORSConfigurationAsync(configuration);

                // Retrieve an existing configuration.
                configuration = await RetrieveCORSConfigurationAsync();

                // Add a new rule.
                configuration.Rules.Add(new CORSRule
                {
                    Id = "CORSRule3",
                    AllowedMethods = new List<string> { "HEAD" },
                    AllowedOrigins = new List<string> { "http://www.example.com" }
                });

                // Add the configuration to the bucket.
                await PutCORSConfigurationAsync(configuration);

                // Verify that there are now three rules.
                configuration = await RetrieveCORSConfigurationAsync();
                Console.WriteLine();
                Console.WriteLine("Expected # of rulest=3; found:{0}",
 configuration.Rules.Count);
                Console.WriteLine();
                Console.WriteLine("Pause before configuration delete. To continue,
 click Enter...");
                Console.ReadKey();

                // Delete the configuration.
                await DeleteCORSConfigurationAsync();
```

```
                // Retrieve a nonexistent configuration.
                configuration = await RetrieveCORSConfigurationAsync();
            }
            catch (AmazonS3Exception e)
            {
                Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
            catch (Exception e)
            {
                Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
            }
        }

        static async Task PutCORSConfigurationAsync(CORSConfiguration configuration)
        {

            PutCORSConfigurationRequest request = new PutCORSConfigurationRequest
            {
                BucketName = bucketName,
                Configuration = configuration
            };

            var response = await s3Client.PutCORSConfigurationAsync(request);
        }

        static async Task<CORSConfiguration> RetrieveCORSConfigurationAsync()
        {
            GetCORSConfigurationRequest request = new GetCORSConfigurationRequest
            {
                BucketName = bucketName

            };
            var response = await s3Client.GetCORSConfigurationAsync(request);
            var configuration = response.Configuration;
            PrintCORSRules(configuration);
            return configuration;
        }

        static async Task DeleteCORSConfigurationAsync()
        {
```

```
            DeleteCORSConfigurationRequest request = new
 DeleteCORSConfigurationRequest
            {
                BucketName = bucketName
            };
            await s3Client.DeleteCORSConfigurationAsync(request);
        }

        static void PrintCORSRules(CORSConfiguration configuration)
        {
            Console.WriteLine();

            if (configuration == null)
            {
                Console.WriteLine("\nConfiguration is null");
                return;
            }

            Console.WriteLine("Configuration has {0} rules:",
 configuration.Rules.Count);
            foreach (CORSRule rule in configuration.Rules)
            {
                Console.WriteLine("Rule ID: {0}", rule.Id);
                Console.WriteLine("MaxAgeSeconds: {0}", rule.MaxAgeSeconds);
                Console.WriteLine("AllowedMethod: {0}", string.Join(", ",
 rule.AllowedMethods.ToArray()));
                Console.WriteLine("AllowedOrigins: {0}", string.Join(", ",
 rule.AllowedOrigins.ToArray()));
                Console.WriteLine("AllowedHeaders: {0}", string.Join(", ",
 rule.AllowedHeaders.ToArray()));
                Console.WriteLine("ExposeHeader: {0}", string.Join(", ",
 rule.ExposeHeaders.ToArray()));
            }
        }
    }
}
```

Mit dem REST API

Um eine CORS Konfiguration für Ihren Bucket festzulegen, können Sie den AWS Management Console verwenden. Wenn Ihre Anwendung dies erfordert, können Sie REST Anfragen auch direkt

senden. In den folgenden Abschnitten der Amazon Simple Storage Service API Reference werden die REST API Aktionen im Zusammenhang mit der CORS Konfiguration beschrieben:

- PutBucketCors
- GetBucketCors
- DeleteBucketCors
- OPTIONSObjekt

Testen CORS

Um Ihre CORS Konfiguration zu testen, kann mit dieser `OPTIONS` Methode eine CORS Preflight-Anfrage gesendet werden, sodass der Server antworten kann, ob das Senden der Anfrage akzeptabel ist. Wenn Amazon S3 eine Preflight-Anfrage erhält, bewertet S3 die CORS Konfiguration für den Bucket und verwendet die erste `CORSRule` Regel, die der eingehenden Anfrage entspricht, um eine ursprungsübergreifende Anfrage zu aktivieren. Für die Übereinstimmung mit einer Regel müssen die folgenden Bedingungen erfüllt sei:

- Der `Origin` Header in einer CORS Anfrage an Ihren Bucket muss mit den Ursprüngen im `AllowedOrigins` Element in Ihrer Konfiguration übereinstimmen. CORS
- Die HTTP Methoden, die `Access-Control-Request-Method` in einer CORS Anfrage an Ihren Bucket angegeben sind, müssen mit der oder den Methoden übereinstimmen, die in dem `AllowedMethods` Element in Ihrer CORS Konfiguration aufgeführt sind.
- Die im Header einer Preflight-Anfrage aufgeführten `Access-Control-Request-Headers` Header müssen mit den Headern im `AllowedHeaders` Element in Ihrer Konfiguration übereinstimmen. CORS

Das Folgende ist ein Beispiel für eine Konfiguration. CORS Informationen zum Erstellen einer CORS Konfiguration finden Sie unter Konfiguration CORS. Weitere Beispiele für eine CORS Konfiguration finden Sie unter Elemente einer CORS Konfiguration.

JSON

```
[
    {
        "AllowedHeaders": [
            "Authorization"
        ],
```

```
        "AllowedMethods": [
            "GET"
            "PUT",
            "POST",
            "DELETE"
        ],
        "AllowedOrigins": [
            "http://www.example1.com"
        ],
        "ExposeHeaders":  [
             "x-amz-meta-custom-header"
        ]

    }
]
```

Um die CORS Konfiguration zu testen, können Sie mit dem folgenden CURL Befehl einen `OPTIONS` Preflight-Check senden. CURList ein Befehlszeilentool, das für die Interaktion mit S3 verwendet werden kann. Weitere Informationen finden Sie unter. CURL

```
curl -v -X OPTIONS \
 -H "Origin: http://www.example1.com" \
 -H "Access-Control-Request-Method: PUT" \
 -H "Access-Control-Request-Headers: Authorization" \
 -H "Access-Control-Expose-Headers: x-amz-meta-custom-header"\
    "http://bucket_name.s3.amazonaws.com/object_prefix_name"
```

Im obigen Beispiel wird der `curl -v -x OPTIONS` Befehl verwendet, um eine Preflight-Anfrage an S3 zu senden, um abzufragen, ob S3 erlaubt ist, eine `PUT` Anfrage für ein Objekt vom Cross-Origin zu senden. `http://www.example1.com` Die Header `Access-Control-Request-Headers` und `Access-Control-Expose-Headers` sind optional.

- Als Antwort auf den `Access-Control-Request-Method` Header in der `OPTIONS` Preflight-Anfrage gibt Amazon S3 die Liste der zulässigen Methoden zurück, wenn die angeforderten Methoden übereinstimmen.
- Als Antwort auf den `Access-Control-Request-Headers` Header in der `OPTIONS` Preflight-Anfrage gibt Amazon S3 die Liste der zulässigen Header zurück, wenn die angeforderten Header übereinstimmen.

- Als Antwort auf den `Access-Control-Expose-Headers` Header in der `OPTIONS` Preflight-Anfrage gibt Amazon S3 eine Liste der zulässigen Header zurück, wenn die angeforderten Header mit den erlaubten Headern übereinstimmen, auf die von Skripten zugegriffen werden kann, die im Browser ausgeführt werden.

> **Note**
>
> Wenn beim Senden einer Preflight-Anfrage einer der CORS Anforderungsheader nicht zulässig ist, wird keiner der Antwortheader zurückgegeben. CORS

Als Antwort auf diese `OPTIONS` Preflight-Anfrage erhalten Sie eine Antwort. `200 OK` Allgemeine Fehlercodes, die beim Testen aufgetreten sind, CORS und weitere Informationen zur Lösung CORS verwandter Probleme finden Sie unter Problembehandlung CORS.

```
< HTTP/1.1 200 OK
< Date: Fri, 12 Jul 2024 00:23:51 GMT
< Access-Control-Allow-Origin: http://www.example1.com
< Access-Control-Allow-Methods: GET, PUT, POST, DELETE
< Access-Control-Allow-Headers: Authorization
< Access-Control-Expose-Headers: x-amz-meta-custom-header
< Access-Control-Allow-Credentials: true
< Vary: Origin, Access-Control-Request-Headers, Access-Control-Request-Method
< Server: AmazonS3
< Content-Length: 0
```

Fehlerbehebung für CORS

Die folgenden Themen können Ihnen helfen, einige häufig auftretende CORS Probleme im Zusammenhang mit S3 zu beheben.

Themen

- 403 Unerlaubter Fehler: CORS ist für diesen Bucket nicht aktiviert
- 403 Unzulässiger Fehler: Diese CORS Anfrage ist nicht zulässig
- Header wurden als Antwort nicht gefunden CORS
- Überlegungen zu CORS S3-Proxy-Integrationen

403 Unerlaubter Fehler: CORS ist für diesen Bucket nicht aktiviert

Der folgende `403 Forbidden` Fehler tritt auf, wenn eine ursprungsübergreifende Anfrage an Amazon S3 gesendet, aber nicht in Ihrem S3-Bucket konfiguriert CORS ist.

Fehler: HTTP/1.1 403 Forbidden CORS Response: CORS ist für diesen Bucket nicht aktiviert.

Bei der CORS Konfiguration handelt es sich um ein Dokument oder eine Richtlinie mit Regeln, die die Ursprünge, denen Sie den Zugriff auf Ihren Bucket erlauben, die Operationen (HTTPMethoden), die Sie für jeden Ursprung unterstützen, und andere vorgangsspezifische Informationen festlegen. Erfahren Sie, wie Sie mithilfe der Amazon S3-Konsole CORS auf S3 konfigurieren AWS SDKs, und RESTAPI. Weitere Informationen CORS und Beispiele für eine CORS Konfiguration finden Sie unter Elements of CORS.

403 Unzulässiger Fehler: Diese CORS Anfrage ist nicht zulässig

Der folgende `403 Forbidden` Fehler wird angezeigt, wenn eine CORS Regel in Ihrer CORS Konfiguration nicht mit den Daten in Ihrer Anfrage übereinstimmt.

Fehler: HTTP/1.1 403 Verbotene CORS Antwort: Diese CORS Anfrage ist nicht zulässig.

Daher kann dieser `403 Forbidden` Fehler aus mehreren Gründen auftreten:

- Herkunft ist nicht erlaubt.
- Methoden sind nicht erlaubt.
- Angeforderte Header sind nicht erlaubt.

Für jede Anfrage, die Amazon S3 empfängt, müssen Sie eine CORS Regel in Ihrer CORS Konfiguration haben, die den Daten in Ihrer Anfrage entspricht.

Origin ist nicht erlaubt

Der `Origin` Header in einer CORS Anfrage an Ihren Bucket muss mit den Ursprüngen im `AllowedOrigins` Element in Ihrer CORS Konfiguration übereinstimmen. Ein Platzhalterzeichen (`"*"`) im `AllowedOrigins` Element würde allen HTTP Methoden entsprechen. Weitere Informationen zum Aktualisieren des `AllowedOrigins` Elements finden Sie unter Konfiguration der gemeinsamen Nutzung von Ressourcen zwischen verschiedenen Quellen () CORS.

Wenn beispielsweise nur die `http://www.example1.com` Domäne im `AllowedOrigins` Element enthalten ist, würde bei einer von der `http://www.example2.com` Domäne gesendeten CORS Anfrage der `403 Forbidden` Fehler angezeigt.

Das folgende Beispiel zeigt einen Teil einer CORS Konfiguration, die die `http://www.example1.com` Domain in das `AllowedOrigins` Element einbezieht.

```
"AllowedOrigins":[
   "http://www.example1.com"
]
```

Damit eine von der `http://www.example2.com` Domain gesendete CORS Anfrage erfolgreich ist, muss die `http://www.example2.com` Domain in das `AllowedOrigins` CORS Konfigurationselement aufgenommen werden.

```
"AllowedOrigins":[
   "http://www.example1.com"
   "http://www.example2.com"
]
```

Methoden sind nicht erlaubt

Die HTTP Methoden, die `Access-Control-Request-Method` in einer CORS Anfrage an Ihren Bucket angegeben sind, müssen mit der oder den Methoden übereinstimmen, die in dem `AllowedMethods` Element in Ihrer CORS Konfiguration aufgeführt sind. Ein Platzhalterzeichen (`"*"`) in `AllowedMethods` würde allen HTTP Methoden entsprechen. Weitere Informationen zum Aktualisieren des `AllowedOrigins` Elements finden Sie unter Cross-Origin Resource Sharing () CORS konfigurieren.

In einer CORS Konfiguration können Sie die folgenden Methoden im `AllowedMethods` Element angeben:

- `GET`
- `PUT`
- `POST`
- `DELETE`
- `HEAD`

Das folgende Beispiel zeigt einen Teil einer CORS Konfiguration, die die GET Methode im AllowedMethods Element enthält. Nur Anfragen, die die GET Methode enthalten, würden erfolgreich sein.

```
"AllowedMethods":[
   "GET"
]
```

Wenn eine HTTP Methode (z. B.PUT) in einer CORS Anfrage verwendet oder in einer CORS Preflight-Anfrage an Ihren Bucket enthalten war, die Methode jedoch nicht in Ihrer CORS Konfiguration vorhanden ist, würde die Anfrage zu einem 403 Forbidden Fehler führen. Um diese CORS Anfrage oder CORS Pre-Flight-Anfrage zuzulassen, muss die PUT Methode zu Ihrer CORS Konfiguration hinzugefügt werden.

```
"AllowedMethods":[
   "GET"
   "PUT"
]
```

Angeforderte Header sind nicht zulässig

Die im Header einer Pre-Flight-Anfrage aufgeführten Access-Control-Request-Headers Header müssen mit den Headern im AllowedHeaders Element in Ihrer Konfiguration übereinstimmen. CORS Eine Liste gängiger Header, die in Anfragen an Amazon S3 verwendet werden können, finden Sie unter Allgemeine Anforderungsheader. Weitere Informationen zur Aktualisierung des AllowedHeaders Elements finden Sie unter Konfiguration der ursprungsübergreifenden gemeinsamen Nutzung von Ressourcen (). CORS

Das folgende Beispiel zeigt einen Teil einer CORS Konfiguration, die den Authorization Header im AllowedHeaders Element enthält. Nur Anfragen für den Authorization Header wären erfolgreich.

```
"AllowedHeaders":  [
    "Authorization"
]
```

Wenn ein Header (zum Beispiel) in einer CORS Anfrage enthalten Content-MD5 war, der Header jedoch nicht in Ihrer CORS Konfiguration vorhanden ist, würde die Anfrage zu einem 403 Forbidden Fehler führen. Um diese CORS Anfrage zuzulassen, muss der Content-MD5 Header

zu Ihrer CORS Konfiguration hinzugefügt werden. Wenn Sie `Authorization` sowohl `Content-MD5` Header als auch Header in einer CORS Anfrage an Ihren Bucket übergeben möchten, stellen Sie sicher, dass beide Header im `AllowedHeaders` Element in Ihrer CORS Konfiguration enthalten sind.

```
"AllowedHeaders":  [
    "Authorization"
    "Content-MD5"
]
```

Header wurden als Antwort nicht gefunden CORS

Das `ExposeHeaders` Element in Ihrer CORS Konfiguration identifiziert, welche Antwortheader Sie Skripten und Anwendungen, die in Browsern ausgeführt werden, als Antwort auf eine CORS Anfrage zugänglich machen möchten.

Wenn Ihre in Ihrem S3-Bucket gespeicherten Objekte neben den Antwortdaten auch benutzerdefinierte Metadaten (z. B.`x-amz-meta-custom-header`) enthalten, könnte dieser benutzerdefinierte Header zusätzliche Metadaten oder Informationen enthalten, auf die Sie über Ihren JavaScript clientseitigen Code zugreifen möchten. Standardmäßig blockieren Browser jedoch aus Sicherheitsgründen den Zugriff auf benutzerdefinierte Header. Um Ihrem Client den Zugriff auf benutzerdefinierte Header JavaScript zu ermöglichen, müssen Sie den Header in Ihre Konfiguration aufnehmen. CORS

Im folgenden Beispiel ist der `x-amz-meta-custom-header1` Header im Element enthalten. `ExposeHeaders` Das `x-amz-meta-custom-header2` ist nicht im `ExposeHeaders` Element enthalten und fehlt in der CORS Konfiguration. In der Antwort würden nur die im `ExposeHeaders` Element enthaltenen Werte zurückgegeben. Wenn die Anfrage den `x-amz-meta-custom-header2` Header im `Access-Control-Expose-Headers` Header enthalten würde, würde die Antwort trotzdem a zurückgeben`200 OK`. Es `x-amz-meta-custom-header` würde jedoch zum Beispiel nur der zulässige Header zurückgegeben und in der Antwort angezeigt.

```
"ExposeHeaders":  [
    "x-amz-meta-custom-header1"
]
```

Um sicherzustellen, dass alle Header in der Antwort erscheinen, fügen Sie dem `ExposeHeaders` Element in Ihrer CORS Konfiguration alle zulässigen Header hinzu, wie unten gezeigt.

```
"ExposeHeaders":  [
    "x-amz-meta-custom-header1",
    "x-amz-meta-custom-header2"
]
```

Überlegungen zu CORS S3-Proxy-Integrationen

Wenn bei Ihnen Fehler auftreten und Sie die CORS Konfiguration Ihres S3-Buckets bereits überprüft haben und die Cross-Origin-Anfrage an Proxys wie gesendet wird AWS CloudFront, versuchen Sie Folgendes:

- Konfigurieren Sie die Einstellungen, um die `OPTIONS` Methode für Anfragen zuzulassen. HTTP
- Konfigurieren Sie den Proxy so, dass er die folgenden Header weiterleitet: `OriginAccess-Control-Request-Headers`, und`Access-Control-Request-Method`.

Einige Proxys bieten vordefinierte Funktionen für Anfragen. CORS In können Sie beispielsweise eine Richtlinie konfigurieren CloudFront, die die Header enthält, die CORS Anfragen ermöglichen, wenn der Ursprung ein S3-Bucket ist. Weitere Informationen finden Sie unter Steuern von Ursprungsanfragen mit einer Richtlinie und Verwenden von Richtlinien für verwaltete Ursprungsanfragen im CloudFront Entwicklerhandbuch.

Tutorials für statische Websites

Die folgenden Tutorials oder exemplarischen Vorgehensweisen enthalten vollständige Verfahren zum Erstellen und Konfigurieren eines Amazon S3 S3-Allzweck-Buckets für das Hosten statischer Websites und das Hosten von On-Demand-Videostreaming. Der Zweck dieser Tutorials besteht darin, allgemeine Anleitungen zu geben. Diese Tutorials sind für eine Labor-Umgebung konzipiert und verwenden Beispiel-Bucket-Namen, Benutzernamen usw. Ohne die sorgfältige Prüfung und Anpassung an die besonderen Gegebenheiten der Umgebung Ihrer Organisation sind sie nicht zur direkten Verwendung in einer Produktionsumgebung bestimmt.

- Hosten von On-Demand-Streaming-Videos mit Amazon S3 CloudFront, Amazon und Amazon Route 53 — Sie können Amazon S3 mit Amazon verwenden, CloudFront um Videos für die On-Demand-Ansicht auf sichere und skalierbare Weise zu hosten. Nachdem Ihr Video in die richtigen Formate verpackt wurde, können Sie es auf einem Server oder in einem S3-Allzweckcontainer speichern und es dann auf Wunsch der Zuschauer bereitstellen. CloudFront In diesem Tutorial erfahren Sie, wie Sie Ihren Allzweck-Bucket CloudFront für das Hosten

von On-Demand-Videostreaming mithilfe von Amazon Route 53 for Domain Name System (DNS) und benutzerdefinierter Domainverwaltung konfigurieren. CloudFront stellt das Video aus seinem Cache bereit und ruft es nur dann aus Ihrem Allzweck-Bucket ab, wenn es noch nicht zwischengespeichert ist. Diese Caching-Verwaltungsfunktion beschleunigt die Bereitstellung Ihres Videos für Viewer weltweit mit geringer Latenz, hohem Durchsatz und hohen Übertragungsgeschwindigkeiten. Weitere Informationen zur CloudFront Cachingverwaltung finden Sie unter Optimieren von Caching und Verfügbarkeit im Amazon CloudFront Developer Guide.

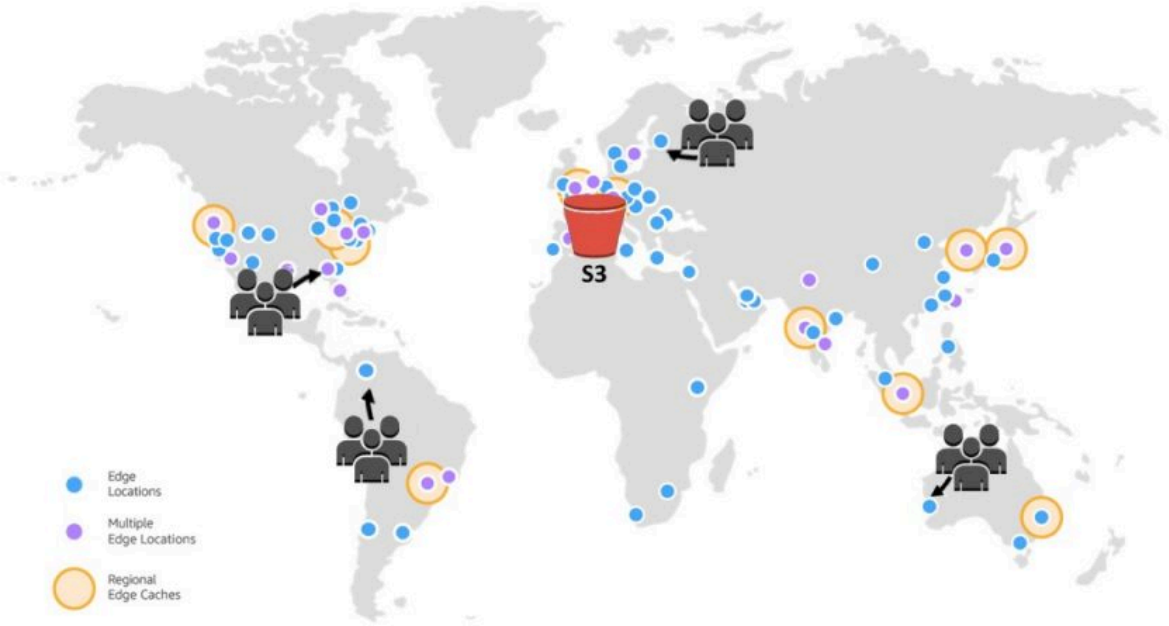
- Konfiguration einer statischen Website — Sie können einen Allzweck-Bucket so konfigurieren, dass er wie eine Website funktioniert. Dieses Tutorial führt Sie durch die Schritte zum Hosten einer Website auf Amazon S3, einschließlich der Erstellung eines Buckets, der Aktivierung des statischen Website-Hostings in der S3-Konsole, der Erstellung eines Indexdokuments und der Erstellung eines Fehlerdokuments. Weitere Informationen finden Sie unter Hosten einer statischen Website mit Amazon S3.
- Konfiguration einer statischen Website mit einer bei Route 53 registrierten benutzerdefinierten Domain — Sie können einen Allzweck-Bucket erstellen und konfigurieren, um eine statische Website zu hosten und Weiterleitungen auf S3 für eine Website mit einem benutzerdefinierten Domainnamen zu erstellen, die bei Amazon Route 53 registriert ist. Sie verwenden Route 53, um Domains zu registrieren und zu definieren, wohin Sie den Internetverkehr für Ihre Domain weiterleiten möchten. Dieses Tutorial zeigt, wie Sie Route 53-Aliaseinträge erstellen, die den Traffic für Ihre Domain und Subdomain an Ihren Allzweck-Bucket weiterleiten, der eine HTML Datei enthält. Weitere Informationen finden Sie unter Verwenden Sie Ihre Domain für eine statische Website in einem Amazon S3 S3-Bucket im Amazon Route 53 53-Entwicklerhandbuch. Nachdem Sie dieses Tutorial abgeschlossen haben, können Sie es optional verwenden CloudFront , um die Leistung Ihrer Website zu verbessern. Weitere Informationen finden Sie unter Beschleunigen Sie Ihre Website mit Amazon CloudFront.
- Bereitstellung einer statischen Website für AWS Amplify Hosting aus einem S3-Allzweck-Bucket — Wir empfehlen, AWS Amplify Hosting zu verwenden, um statische Website-Inhalte zu hosten, die auf S3 gespeichert sind. Amplify Hosting ist ein vollständig verwalteter Service, mit dem Sie Ihre Websites auf einfache Weise in einem weltweit verfügbaren Content Delivery Network (CDN) bereitstellen können CloudFront, das von Amazon betrieben wird und sicheres statisches Website-Hosting ohne umfangreiche Einrichtung ermöglicht. Mit AWS Amplify Hosting können Sie den Speicherort Ihrer Objekte in Ihrem Allzweck-Bucket auswählen, Ihre Inhalte in einem verwalteten Bereich bereitstellen und eine Öffentlichkeit HTTPS URL generierenCDN, damit Ihre Website von überall aus zugänglich ist. Weitere Informationen finden Sie unter Bereitstellen einer statischen Website von S3 mithilfe der Amplify-Konsole im AWS Amplify Hosting-Benutzerhandbuch.

Tutorial: Hosten von On-Demand-Streaming-Videos mit Amazon S3 CloudFront, Amazon und Amazon Route 53

Sie können Amazon S3 mit Amazon verwenden CloudFront , um Videos für die On-Demand-Ansicht auf sichere und skalierbare Weise zu hosten. Video-on-Demand-Streaming (VOD) bedeutet, dass Ihre Videoinhalte auf einem Server gespeichert werden und Zuschauer sie jederzeit ansehen können.

CloudFront ist ein schneller, hochsicherer und programmierbarer Content Delivery Network (CDN) - Dienst. CloudFront kann Ihre Inhalte sicher HTTPS von allen CloudFront Edge-Standorten auf der ganzen Welt aus bereitstellen. Weitere Informationen zu CloudFront finden Sie unter Was ist Amazon CloudFront? im Amazon CloudFront Developer Guide.

CloudFront Caching reduziert die Anzahl der Anfragen, auf die Ihr Ursprungsserver direkt antworten muss. Wenn ein Zuschauer (Endbenutzer) ein Video anfordert, das Sie bereitstellen CloudFront, wird die Anfrage an einen Standort in der Nähe des Randes weitergeleitet, der sich näher am Standort des Betrachters befindet. CloudFront stellt das Video aus seinem Cache bereit und ruft es nur dann aus dem S3-Bucket ab, wenn es noch nicht zwischengespeichert ist. Diese Caching-Verwaltungsfunktion beschleunigt die Bereitstellung Ihres Videos für Viewer weltweit mit geringer Latenz, hohem Durchsatz und hohen Übertragungsgeschwindigkeiten. Weitere Informationen zur CloudFront Cachingverwaltung finden Sie unter Optimieren von Caching und Verfügbarkeit im Amazon CloudFront Developer Guide.



Ziel

In diesem Tutorial konfigurieren Sie einen S3-Bucket CloudFront für das Hosten von On-Demand-Videostreaming mit Amazon Route 53 für Domain Name System (DNS) und benutzerdefinierter Domainverwaltung.

Themen

- Voraussetzungen: Registrieren und Konfigurieren einer benutzerdefinierten Domäne mit Route 53
- Schritt 1: Einen S3-Bucket erstellen
- Schritt 2: Hochladen eines Videos in den S3-Bucket
- Schritt 3: Erstellen Sie eine ursprüngliche CloudFront Zugriffsidentität
- Schritt 4: Erstellen Sie eine CloudFront Distribution
- Schritt 5: Greifen Sie über die CloudFront Distribution auf das Video zu
- Schritt 6: Konfigurieren Sie Ihre CloudFront Distribution so, dass sie Ihren benutzerdefinierten Domainnamen verwendet
- Schritt 7: Greifen Sie über die CloudFront Distribution mit dem benutzerdefinierten Domainnamen auf das S3-Video zu

- (Optional) Schritt 8: Zeigen Sie Daten zu Anfragen an, die bei Ihrer CloudFront Distribution eingegangen sind
- Schritt 9: Bereinigen
- Nächste Schritte

Voraussetzungen: Registrieren und Konfigurieren einer benutzerdefinierten Domäne mit Route 53

Bevor Sie mit diesem Tutorial beginnen, müssen Sie eine benutzerdefinierte Domain (z. B.**`example.com`**) mit Route 53 registrieren und konfigurieren, damit Sie Ihre CloudFront Distribution so konfigurieren können, dass sie später einen benutzerdefinierten Domainnamen verwendet.

Ohne einen benutzerdefinierten Domainnamen ist Ihr S3-Video öffentlich zugänglich und wird über CloudFront einen Server gehostetURL, der wie folgt aussieht:

```
https://CloudFront distribution domain name/Path to an S3 video
```

Beispiel, **`https://d111111abcdef8.cloudfront.net/sample.mp4`**.

Nachdem Sie Ihre CloudFront Distribution für die Verwendung eines mit Route 53 konfigurierten benutzerdefinierten Domainnamens konfiguriert haben, ist Ihr S3-Video öffentlich zugänglich und wird über CloudFront eine URL gehostetURL, die etwa wie folgt aussieht:

```
https://CloudFront distribution alternate domain name/Path to an S3 video
```

Beispiel, **`https://www.example.com/sample.mp4`**. Ein benutzerdefinierter Domänenname ist für die Viewer einfacher und intuitiver zu verwenden.

Informationen zur Anmeldung einer benutzerdefinierten Domäne finden Sie unter Registrieren einer neuen Domäne mit Route 53 im Amazon Route 53-Entwicklerhandbuch.

Wenn Sie einen Domänennamen bei Route 53 registrieren, erstellt Route 53 die gehostete Zone für Sie, die Sie später in diesem Tutorial verwenden werden. In dieser gehosteten Zone speichern Sie Informationen darüber, wie Sie den Traffic für Ihre Domain weiterleiten, z. B. an eine EC2 Amazon-Instance oder eine CloudFront Distribution.

Im Zusammenhang mit der Domainregistrierung, Ihrer Hosting-Zone und DNS Anfragen, die über Ihre Domain eingehen, fallen Gebühren an. Weitere Informationen dazu finden Sie unter Amazon Route 53 – Preise.

> **Note**
>
> Wenn Sie eine Domäne registrieren, kostet diese sofort Geld. Dies kann nicht rückgängig gemacht werden. Sie können wählen, die Domäne nicht automatisch zu erneuern, aber Sie bezahlen im Voraus und besitzen sie für das Jahr. Weitere Informationen finden Sie unter Registrieren einer neuen Domäne im Amazon-Route-53-Entwicklerhandbuch.

Schritt 1: Einen S3-Bucket erstellen

Erstellen Sie einen Bucket, um das Originalvideo zu speichern, das Sie streamen möchten.

So erstellen Sie einen Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie Create Bucket (Bucket erstellen) aus.

   Die Seite Bucket erstellen wird geöffnet.
4. Geben Sie im Feld Bucket Name einen Namen für Ihren Bucket ein (z. B. **`tutorial-bucket`**).

   Weitere Informationen zur Benennung von Buckets in Amazon S3 finden Sie unter Allgemeine Benennungsregeln für Buckets.
5. Wählen Sie unter Region den AWS-Region Ort aus, an dem sich der Bucket befinden soll.

   Wenn möglich, sollten Sie die Region auswählen, die der Mehrheit Ihrer Viewer am nächsten ist. Weitere Informationen zur Bucket-Region finden Sie unter Bucket-Übersicht.
6. Belassen Sie die Einstellungen für den öffentlichen Zugriff für diesen Bucket blockieren bei den Standardeinstellungen (Alle öffentlichen Zugriffe blockieren ist aktiviert).

   Auch wenn „Allen öffentlichen Zugriff blockieren“ aktiviert ist, können Zuschauer weiterhin über das hochgeladene Video auf das hochgeladene Video zugreifen CloudFront. Diese Funktion ist ein großer Vorteil beim Hosten eines in S3 gespeicherten Videos. CloudFront

   Es wird empfohlen, alle Einstellungen aktiviert zu belassen, es sei denn, Sie müssen eine oder mehrere dieser Einstellungen für Ihren Anwendungsfall deaktivieren. Weitere Informationen zum

Blockieren des öffentlichen Zugriffs finden Sie unter Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher.

7. Lassen Sie die restlichen Einstellungen auf die Standardwerte eingestellt.

   (Optional) Informationen zum Konfigurieren zusätzlicher Bucket-Einstellungen für Ihren spezifischen Anwendungsfall finden Sie unter Erstellen eines Buckets.

8. Wählen Sie Bucket erstellen aus.

Schritt 2: Hochladen eines Videos in den S3-Bucket

Im folgenden Verfahren wird beschrieben, wie Sie eine Videodatei mithilfe der Konsole in einen S3-Bucket hochladen. Wenn Sie ein Video in S3 hochladen, können Sie auch Amazon S3 Transfer Acceleration nutzen, um schnelle und sichere Dateiübertragungen zu konfigurieren. Transfer Acceleration kann das Hochladen von Videos zu Ihrem S3-Bucket zur Übertragung größerer Videos über große Entfernungen beschleunigen. Weitere Informationen finden Sie unter Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration.

So laden Sie die Datei in einen Bucket hoch

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich die Option Buckets aus.
3. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie in Schritt 1 (z. B. **`tutorial-bucket`**) erstellt haben, in den Sie die Datei hochladen möchten.
4. Wählen Sie im Tab Objects (Objekte) für Ihren Bucket die Option Upload (Hochladen) aus.
5. Wählen Sie auf der Seite Upload unter Dateien und Ordner die Option Dateien hinzufügen aus.
6. Wählen Sie eine hochzuladende Datei und dann Öffnen aus.

   Sie können beispielsweise eine Videodatei mit dem Namen `sample.mp4` hochladen.

7. Klicken Sie auf Hochladen.

Schritt 3: Erstellen Sie eine ursprüngliche CloudFront Zugriffsidentität

Um den direkten Zugriff auf das Video von Ihrem S3-Bucket aus einzuschränken, erstellen Sie einen speziellen CloudFront Benutzer, eine sogenannte Origin-Zugriffsidentität (OAI). Sie werden das später in diesem Tutorial OAI mit Ihrer Distribution verknüpfen. Durch die Verwendung von stellen

Sie sicherOAI, dass Zuschauer das Video nicht umgehen CloudFront und direkt aus dem S3-Bucket abrufen können. Nur CloudFront OAI sie können auf die Datei im S3-Bucket zugreifen. Weitere Informationen finden Sie unter Beschränken des Zugriffs auf Amazon S3 S3-Inhalte mithilfe von OAI im Amazon CloudFront Developer Guide.

Um eine zu erstellen CloudFront OAI

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im linken Navigationsbereich im Abschnitt Sicherheit Ursprungszugriff aus.
3. Wählen Sie unter der Registerkarte Identitäten Ursprungszugriffsidentität erstellen aus.
4. Geben Sie einen Namen ein (z. B. **`S3-OAI`**) als die neue Ursprungszugriffsidentität ein.
5. Wählen Sie Create (Erstellen) aus.

Schritt 4: Erstellen Sie eine CloudFront Distribution

Um das Video in Ihrem S3-Bucket bereitzustellen und zu verteilen, müssen Sie eine CloudFront Distribution erstellen. CloudFront

Teilschritte

- Erstellen Sie eine CloudFront Distribution
- Überprüfen der Bucket-Richtlinie

Erstellen Sie eine CloudFront Distribution

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im linken Navigationsbereich Verteilungen aus.
3. Wählen Sie Verteilung erstellen aus.
4. Wählen Sie im Abschnitt Urspung für Ursprungsdomäne den Domänennamen Ihres S3-Ursprungs aus, der mit dem Namen des S3-Buckets beginnt, den Sie unter Schritt 1 (z. B. **`tutorial-bucket`**) erstellt haben.
5. Wählen Sie für Ursprungszugriff die Option Legacy-Zugriffsidentitäten aus.
6. Wählen Sie unter Ursprungszugriffsidentität die Ursprungszugriffsidentität aus, die Sie in Schritt 3 (z. B. **`S3-OAI`**) erstellt haben.

7. Wählen Sie unter Bucket-Richtlinie die Option Ja, Bucket-Richtlinie aktualisieren aus.
8. Wählen Sie im Abschnitt Standard-Cache-Verhalten unter Viewer-Protokollrichtlinie die Option Umleiten HTTP zu ausHTTPS.

   Wenn Sie diese Funktion auswählen, werden HTTP Anfragen automatisch weitergeleitet, HTTPS um Ihre Website und die Daten Ihrer Zuschauer zu schützen.
9. Behalten Sie für die anderen Einstellungen im Abschnitt Standard-Cacheverhalten die Standardwerte bei.

   (Optional) Sie können steuern, wie lange Ihre Datei in einem CloudFront Cache CloudFront verbleibt, bevor eine weitere Anfrage an Ihren Ursprung weitergeleitet wird. Eine Reduzierung der Dauer ermöglicht Ihnen, dynamische Inhalte bereitzustellen. Eine Erhöhung der Dauer bedeutet, dass Ihre Viewer eine bessere Leistung erhalten, da es wahrscheinlicher ist, dass Ihre Dateien direkt vom Edge-Cache bereitgestellt werden. Eine längere Dauer verringert darüber hinaus die Last auf Ihrem Ursprung. Weitere Informationen finden Sie unter Verwaltung der Verweildauer von Inhalten im Cache (Ablauf) im Amazon CloudFront Developer Guide.
10. Behalten Sie für die anderen Abschnitte die restlichen Einstellungen auf die Standardwerte eingestellt.

    Weitere Informationen zu den verschiedenen Einstellungsoptionen finden Sie unter Werte, die Sie beim Erstellen oder Aktualisieren einer Distribution angeben im Amazon CloudFront Developer Guide.
11. Klicken Sie unten auf der Seite auf Verteilung erstellen.
12. Auf der Registerkarte Allgemein für Ihre CloudFront Distribution ändert sich unter Details der Wert der Spalte Letzte Änderung für Ihre Distribution von Bereitstellen auf den Zeitstempel, zu dem die Verteilung zuletzt geändert wurde. Dieser Prozess dauert in der Regel einige Minuten.

Überprüfen der Bucket-Richtlinie

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Buckets-Liste den Namen des Buckets aus, den Sie zuvor als Ursprung Ihrer CloudFront Distribution verwendet haben (z. B.**`tutorial-bucket`**).
4. Wählen Sie die Registerkarte Permissions (Berechtigungen).

5. Bestätigen Sie im Abschnitt Bucket-Richtlinie, dass eine Anweisung ähnlich der folgenden im Bucket-Richtlinientext angezeigt wird:

```
{
    "Version": "2008-10-17",
    "Id": "PolicyForCloudFrontPrivateContent",
    "Statement": [
        {
            "Sid": "1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access
 Identity EH1HDMB1FH2TC"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::tutorial-bucket/*"
        }
    ]
}
```

   Dies ist die Aussage, die Ihre CloudFront Distribution zu Ihrer Bucket-Richtlinie hinzugefügt hat, als Sie Ja, die Bucket-Richtlinie früher aktualisieren ausgewählt haben.

   Diese Aktualisierung der Bucket-Richtlinie weist darauf hin, dass Sie die CloudFront Distribution erfolgreich konfiguriert haben, um den Zugriff auf den S3-Bucket einzuschränken. Aufgrund dieser Einschränkung kann auf Objekte im Bucket nur über Ihre CloudFront Distribution zugegriffen werden.

Schritt 5: Greifen Sie über die CloudFront Distribution auf das Video zu

Jetzt CloudFront können Sie das in Ihrem S3-Bucket gespeicherte Video bereitstellen. Um über auf Ihr Video zugreifen zu können CloudFront, müssen Sie Ihren CloudFront Distributionsdomänennamen mit dem Pfad zum Video im S3-Bucket kombinieren.

Um ein Video zum S3 URL zu erstellen, verwenden Sie den Namen der CloudFront Distributionsdomain

1. Melden Sie sich bei an AWS Management Console und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im linken Navigationsbereich Verteilungen aus.

3. Um den Namen der Verteilungsdomäne zu erhalten, gehen Sie wie folgt vor:

   a. Suchen Sie in der Spalte Origins nach der richtigen CloudFront Distribution, indem Sie nach dem Ursprungsnamen suchen, der mit dem S3-Bucket beginnt, den Sie in Schritt 1 erstellt haben (z. B.**`tutorial-bucket`**).

   b. Nachdem Sie die Verteilung in der Liste gefunden haben, erweitern Sie die Spalte Domainname, um den Domainnamenwert für Ihre CloudFront Distribution zu kopieren.

4. Fügen Sie in einer neuen Browser-Registerkarte den kopierten Verteilungsdomänennamen ein.

5. Kehren Sie zum vorherigen Browser-Tab zurück und öffnen Sie die S3-Konsole unter https://console.aws.amazon.com/s3/.

6. Wählen Sie im linken Navigationsbereich Buckets aus.

7. Wählen Sie in der Liste der Buckets den Namen des Buckets, den Sie in Schritt 1 (z. B. **`tutorial-bucket`**) erstellt haben.

8. Wählen Sie in der Liste Objekte den Namen des Videos aus, das Sie in Schritt 2 (z. B. `sample.mp4`) hochgeladen haben.

9. Kopieren Sie auf der Objektdetailseite im Übersicht über das Objekt den Wert des Schlüssel. Dieser Wert ist der Pfad zum hochgeladenen Videoobjekt im S3-Bucket.

10. Kehren Sie zur Browser-Registerkarte zurück, in die Sie zuvor den Namen der Verteilungsdomäne eingefügt haben und geben Sie einen Schrägstrich (**`/`**) nach dem Domainnamen der Distributionsdomäne ein und fügen Sie dann den Pfad zu dem Video, das Sie vorher kopiert haben, ein (zum Beispiel `sample.mp4`).

    Jetzt ist Ihr S3-Video öffentlich zugänglich und wird über CloudFront einen Server gehostetURL, der wie folgt aussieht:

    ```
    https://CloudFront distribution domain name/Path to the S3 video
    ```

    Ersetzen *`CloudFront distribution domain name`* and *`Path to the S3 video`* mit den entsprechenden Werten. Ein Beispiel einer URList **`https://d111111abcdef8.cloudfront.net/sample.mp4`**.

Schritt 6: Konfigurieren Sie Ihre CloudFront Distribution so, dass sie Ihren benutzerdefinierten Domainnamen verwendet

Um Ihren eigenen Domainnamen anstelle des CloudFront Domainnamens zu verwenden, URL um auf das S3-Video zuzugreifen, fügen Sie Ihrer CloudFront Distribution einen alternativen Domainnamen hinzu.

Teilschritte

- Fordern Sie ein SSL Zertifikat an
- Fügen Sie den alternativen Domainnamen zu Ihrer CloudFront Distribution hinzu
- Erstellen Sie einen DNS Eintrag, um den Traffic von Ihrem alternativen Domainnamen zum Domainnamen Ihrer CloudFront Distribution weiterzuleiten
- Prüfen Sie, ob diese Option für Ihren Vertrieb aktiviert IPv6 ist, und erstellen Sie bei Bedarf einen weiteren DNS Datensatz

Fordern Sie ein SSL Zertifikat an

Verwenden HTTPS Sie (), um ein Secure Sockets Layer AWS Certificate Manager (ACMSSL) - Zertifikat anzufordern, damit Ihre Zuschauer Ihren benutzerdefinierten Domainnamen URL für Ihr Videostreaming verwenden können. Das SSL Zertifikat stellt eine verschlüsselte Netzwerkverbindung zur Website her.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die ACM Konsole unter https://console.aws.amazon.com/acm/.
2. Wenn die Einführungsseite angezeigt wird, klicken Sie unter Bereitstellung von Zertifikaten auf Erste Schritte.
3. Wählen Sie auf der Seite Zertifikat anfordern die Option Öffentliches Zertifikat anfordern und dann Zertifikat anfordern aus.
4. Geben Sie auf der Seite Domainnamen hinzufügen den vollqualifizierten Domainnamen (FQDN) der Site ein, die Sie mit einem SSL TLS /-Zertifikat sichern möchten. Sie können ein Sternchen (`*`) verwenden, um ein Platzhalterzertifikat anzufordern, um mehrere Websitenamen in derselben Domäne zu schützen. Geben Sie in diesem Tutorial **`*`** und den benutzerdefinierten Domänennamen ein, den Sie in Voraussetzungen konfiguriert haben. Verwenden Sie zum Beispiel **`*.example.com`** und wählen Sie dann Weiter aus.

Weitere Informationen finden Sie im AWS Certificate Manager Benutzerhandbuch unter So fordern Sie ein ACM öffentliches Zertifikat (Konsole) an.

5. Wählen Sie auf der Seite „Validierungsmethode auswählen“ die Option DNSValidierung aus. Wählen Sie anschließend Weiter.

   Wenn Sie Ihre DNS Konfiguration bearbeiten können, empfehlen wir Ihnen, die DNS Domain-Validierung anstelle der E-Mail-Validierung zu verwenden. DNSDie Validierung hat mehrere Vorteile gegenüber der E-Mail-Validierung. Weitere Informationen finden Sie unter Option 1: DNS Validierung im AWS Certificate Manager Benutzerhandbuch.
6. (Optional) Auf der Seite Tags hinzufügen können Sie Ihr Zertifikat mit Metadaten markieren.
7. Wählen Sie Überprüfen aus.
8. Überprüfen Sie auf der Seite Prüfen, ob die Informationen unter Domänenname und Validierungsmethode korrekt sind. Wählen Sie Bestätigen und anfordern aus.

   Auf der Validierungs-Seite wird angezeigt, dass Ihre Anforderung verarbeitet wird und dass Zertifikatdomänen validiert werden. Das Zertifikat, das auf die Validierung wartet, befindet sich im Status Validierung ausstehend.
9. Wählen Sie auf der Seite Validierung den Abwärtspfeil links neben Ihrem benutzerdefinierten Domainnamen und wählen Sie dann Create record in Route 53 aus, um Ihren Domainbesitz zu validierenDNS.

   Dadurch wird Ihrer DNS Konfiguration ein von AWS Certificate Manager bereitgestellter CNAME Eintrag hinzugefügt.
10. Wählen Sie im Dialogfeld Datensatz in Route 53 erstellen die Option Erstellen.

    Die Seite Validierung sollte unten eine Statusbenachrichtigung von Erfolg anzeigen.
11. Wählen Sie Continue (Fortfahren), um die Listenseite Certificates (Zertifikate) anzuzeigen.

    Der Status für Ihr neues Zertifikat wird sich innerhalb von 30 Minuten von Ausstehende Validierung auf Ausgestellt ändern.

Fügen Sie den alternativen Domainnamen zu Ihrer CloudFront Distribution hinzu

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im linken Navigationsbereich Verteilungen aus.

3. Wählen Sie die ID für die Verteilung aus, die Sie in Schritt 4 erstellt haben.
4. Gehen Sie auf der Registerkarte Allgemeines zum Abschnitt Einstellungen und wählen Sie Bearbeiten aus.
5. Wählen Sie auf der Seite Einstellungen bearbeiten für Alternativer Domainname (CNAME) — optional die Option Element hinzufügen aus, um die benutzerdefinierten Domainnamen hinzuzufügen, die Sie in dem URL von dieser CloudFront Distribution bereitgestellten S3-Video verwenden möchten.

   Wenn Sie in diesem Tutorials beispielsweise Datenverkehr für eine Subdomäne weiterleiten möchten, z. B.`www.example.com` , geben Sie den Subdomänennamen (`www`) mit dem Domänennamen (`example.com`) ein. Geben Sie insbesondere **`www.example.com`** ein.

   > **Note**
   >
   > Der alternative Domainname (CNAME), den Sie hinzufügen, muss durch das SSL Zertifikat abgedeckt sein, das Sie zuvor Ihrer CloudFront Distribution beigefügt haben.

6. Wählen Sie unter Benutzerdefiniertes SSL Zertifikat — optional das SSL Zertifikat aus, das Sie zuvor angefordert haben (z. B.**`*.example.com`**).

   > **Note**
   >
   > Wenn das SSL Zertifikat nicht sofort nach der Anforderung angezeigt wird, warten Sie 30 Minuten, und aktualisieren Sie dann die Liste, bis das SSL Zertifikat für Sie zur Auswahl verfügbar ist.

7. Lassen Sie die restlichen Einstellungen bei den Standardwerten. Wählen Sie Save Changes (Änderungen speichern).
8. Warten Sie auf der Registerkarte Allgemeines für die Verteilung, bis der Wert Zuletzt geändert von Wird bereitgestellt in den Zeitstempel geändert wurde, als die Verteilung zuletzt geändert wurde.

Erstellen Sie einen DNS Eintrag, um den Traffic von Ihrem alternativen Domainnamen zum Domainnamen Ihrer CloudFront Distribution weiterzuleiten

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Route 53-Konsole unter https://console.aws.amazon.com/route53/.

2. Wählen Sie im linken Navigationsbereich Hosted Zones (Bereitgestellte Zonen).
3. Wählen Sie auf der Seite Gehostete Zonen den Namen der gehosteten Zone aus, die Route 53 für Sie in Voraussetzungen erstellt hat (z. B. **`example.com`**).
4. Klicken Sie auf Datensatz erstellen und benutzen Sie dann die Datensatz Quick Create-Methode.
5. Behalten Sie für den Datensatznamen den Wert für den Datensatznamen bei, den CloudFront Sie zuvor hinzugefügt haben.

   Geben Sie in diesem Tutorial den Namen der Subdomäne ohne den Domänennamen ein, um Datenverkehr an eine Subdomäne weiterzuleiten, z. B. `www.example.com`. Geben Sie z. B. nur ein **`www`** im Textfeld vor Ihrem benutzerdefinierten Domänennamen ein.
6. Wählen Sie als Datensatztyp die Option A — Leitet den Datenverkehr an eine IPv4 Adresse und einige AWS Ressourcen weiter.
7. Wählen Sie für Wert den Alias-Toggle, um die Alias-Ressource zu aktivieren.
8. Wählen Sie unter Traffic weiterleiten an die Option Alias to CloudFront Distribution aus der Dropdownliste aus.
9. Wählen Sie im Suchfeld mit der Aufschrift Vertrieb auswählen den Domainnamen der CloudFront Distribution aus, die Sie in Schritt 4 erstellt haben.

   Gehen Sie wie folgt vor, um den Domainnamen Ihrer CloudFront Distribution zu finden:

   a. Melden Sie sich in einem neuen Browser-Tab an AWS Management Console und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v3/home.
   b. Wählen Sie im linken Navigationsbereich Verteilungen aus.
   c. Suchen Sie in der Spalte Origins nach der richtigen CloudFront Distribution, indem Sie nach dem Ursprungsnamen suchen, der mit dem S3-Bucket beginnt, den Sie in Schritt 1 erstellt haben (z. B.**`tutorial-bucket`**).
   d. Nachdem Sie die Verteilung in der Liste gefunden haben, erweitern Sie die Spalte Domainname, um den Domainnamenwert für Ihre CloudFront Distribution zu sehen.
10. Behalten Sie auf der Seite Datensatz erstellen in der Route 53-Konsole die Standardwerte für die Übrigen Einstellungen bei.
11. Wählen Sie Create records (Datensätze erstellen).

Prüfen Sie, ob diese Option für Ihren Vertrieb aktiviert IPv6 ist, und erstellen Sie bei Bedarf einen weiteren DNS Datensatz

Wenn für Ihre Distribution aktiviert IPv6 ist, müssen Sie einen weiteren DNS Datensatz erstellen.

1. Gehen Sie wie folgt vor, um zu überprüfen, ob es für Ihre Distribution aktiviert IPv6 ist:

   a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.

   b. Wählen Sie im linken Navigationsbereich Verteilungen aus.

   c. Wählen Sie die ID der CloudFront Distribution aus, die Sie in Schritt 4 erstellt haben.

   d. Überprüfen Sie auf der Registerkarte Allgemein unter Einstellungen, ob die Option Aktiviert ausgewählt IPv6ist.

      Wenn für Ihre Distribution aktiviert IPv6 ist, müssen Sie einen weiteren DNS Datensatz erstellen.

2. Wenn für Ihre Distribution aktiviert IPv6 ist, gehen Sie wie folgt vor, um einen DNS Datensatz zu erstellen:

   a. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Route 53-Konsole unter https://console.aws.amazon.com/route53/.

   b. Wählen Sie im linken Navigationsbereich Hosted Zones (Bereitgestellte Zonen).

   c. Wählen Sie auf der Seite Gehostete Zonen den Namen der gehosteten Zone aus, die Route 53 für Sie in Voraussetzungen erstellt hat (z. B. **`example.com`**).

   d. Klicken Sie auf Datensatz erstellen und benutzen Sie dann die Datensatz Quick Create-Methode.

   e. Geben Sie für Datensatzname in das Textfeld vor Ihrem benutzerdefinierten Domainnamen denselben Wert ein, den Sie zuvor bei der Erstellung des IPv4 DNS Datensatzes eingegeben haben. Geben Sie in diesem Tutorial beispielsweise nur **`www`** ein, um den Datenverkehr für die Subdomäne `www.example.com` weiterzuleiten.

   f. Wählen Sie als Datensatztyp die Option AAAA- Leitet den Datenverkehr an eine IPv6 Adresse und einige AWS Ressourcen weiter.

   g. Wählen Sie für Wert den Alias-Toggle, um die Alias-Ressource zu aktivieren.

   h. Wählen Sie unter Traffic weiterleiten an die Option Alias to CloudFront Distribution aus der Dropdownliste aus.

i. Wählen Sie im Suchfeld mit der Aufschrift Vertrieb auswählen den Domainnamen der CloudFront Distribution aus, die Sie in Schritt 4 erstellt haben.

j. Lassen Sie die restlichen Einstellungen auf die Standardwerte eingestellt.

k. Wählen Sie Create records (Datensätze erstellen).

Schritt 7: Greifen Sie über die CloudFront Distribution mit dem benutzerdefinierten Domainnamen auf das S3-Video zu

Um mithilfe der benutzerdefinierten Version auf das S3-Video zuzugreifenURL, müssen Sie Ihren alternativen Domainnamen mit dem Pfad zum Video im S3-Bucket kombinieren.

Um eine benutzerdefinierte Option für den URL Zugriff auf das S3-Video über die CloudFront Distribution zu erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im linken Navigationsbereich Verteilungen aus.
3. Gehen Sie wie folgt vor, um den alternativen Domainnamen Ihrer CloudFront Distribution zu erhalten:

   a. Suchen Sie in der Spalte Origins nach der richtigen CloudFront Distribution, indem Sie nach ihrem Ursprungsnamen suchen. Dieser beginnt mit dem S3-Bucket-Namen für den Bucket, den Sie in Schritt 1 erstellt haben (z. B.**`tutorial-bucket`**).

   b. Nachdem Sie die Verteilung in der Liste gefunden haben, erweitern Sie die Spalte Alternative Domainnamen, um den Wert des alternativen Domainnamens Ihrer CloudFront Distribution zu kopieren.
4. Fügen Sie in einem neuen Browser-Tab den alternativen Domainnamen der CloudFront Distribution ein.
5. Kehren Sie zum vorherigen Browser-Tab zurück und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
6. Finden Sie den Pfad zu Ihrem S3-Video, wie im Schritt 5 erklärt wird.
7. Kehren Sie zur Browserregisterkarte zurück, auf der Sie zuvor den alternativen Domänennamen eingefügt haben, geben Sie einen Vorwärts-Schrägstrich (**`/`**) ein und fügen Sie dann den Pfad in Ihr S3-Video ein (z. B. `sample.mp4`).

Jetzt ist Ihr S3-Video öffentlich zugänglich und wird über CloudFront eine benutzerdefinierte Version gehostetURL, die wie folgt aussieht:

```
https://CloudFront distribution alternate domain name/Path to the S3 video
```

Ersetzen *CloudFront distribution alternate domain name* and *Path to the S3 video* mit den entsprechenden Werten. Ein Beispiel einer URList **https://www.example.com/sample.mp4**.

(Optional) Schritt 8: Zeigen Sie Daten zu Anfragen an, die bei Ihrer CloudFront Distribution eingegangen sind

Um Daten zu Anfragen anzuzeigen, die bei Ihrer CloudFront Distribution eingegangen sind

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im linken Navigationsbereich unter Berichte & Analytik die Berichte aus der Konsole aus, die sich über Cache-Statistiken, Beliebte Objekte, Top-Referrer, Verwendung bis zu Viewer erstrecken.

   Sie können jedes Berichts-Dashboard filtern. Weitere Informationen finden Sie unter CloudFront Berichte in der Konsole im Amazon CloudFront Developer Guide.
3. Um Daten zu filtern, wählen Sie die ID der CloudFront Distribution aus, die Sie in Schritt 4 erstellt haben.

Schritt 9: Bereinigen

Wenn Sie ein S3-Streaming-Video mit CloudFront Route 53 nur zu Lernzwecken gehostet haben, löschen Sie die AWS Ressourcen, die Sie zugewiesen haben, damit keine Gebühren mehr anfallen.

> **Note**
>
> Wenn Sie eine Domäne registrieren, kostet diese sofort Geld. Dies kann nicht rückgängig gemacht werden. Sie können wählen, die Domäne nicht automatisch zu erneuern, aber Sie bezahlen im Voraus und besitzen sie für das Jahr. Weitere Informationen finden Sie unter Registrieren einer neuen Domäne im Amazon-Route-53-Entwicklerhandbuch.

Teilschritte

- Löschen Sie die Verteilung CloudFront
- Löschen Sie den Datensatz DNS
- Löschen Sie die öffentlich gehostete Zone für Ihre benutzerdefinierte Domäne
- Löschen des benutzerdefinierten Domänennamens aus Route 53
- Löschen des Originalvideos im S3-Quell-Bucket
- Löschen des S3-Quell-Bucket

Löschen Sie die Verteilung CloudFront

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie im linken Navigationsbereich Verteilungen aus.
3. Suchen Sie in der Spalte Origins nach der richtigen CloudFront Distribution, indem Sie nach ihrem Ursprungsnamen suchen. Dieser beginnt mit dem S3-Bucket-Namen für den Bucket, den Sie in Schritt 1 erstellt haben (z. B.**`tutorial-bucket`**).
4. Um die CloudFront Distribution zu löschen, müssen Sie sie zuerst deaktivieren.

   - Wenn der Wert der Spalte Status Aktiviert ist und der Wert von Letzte Änderung der Zeitstempel ist, in dem die Verteilung zuletzt geändert wurde, deaktivieren Sie die Verteilung weiterhin, bevor Sie sie löschen.
   - Wenn der Wert von Status Aktiviert ist und der Wert von Letzte Änderung Wird bereitgestellt ist, warten Sie, bis sich der Wert von Status in den Zeitstempel ändert, als die Verteilung zuletzt geändert wurde. Fahren Sie dann fort, um die Verteilung vor dem Löschen zu deaktivieren.
5. Gehen Sie wie folgt vor, um die CloudFront Verteilung zu deaktivieren:

   a. Wählen Sie in der Verteilungen-Liste das Kontrollkästchen neben der ID für die Verteilung, die Sie löschen möchten.

   b. Um die Verteilung zu deaktivieren, wählen Sie Deaktivieren, und wählen Sie dann Deaktivieren, um zu bestätigen.

      Wenn Sie eine Verteilung deaktivieren, der ein alternativer Domänenname zugeordnet ist, wird kein Verkehr CloudFront mehr für diesen Domainnamen (z. B.`www.example.com`)

akzeptiert, auch wenn eine andere Distribution einen alternativen Domainnamen mit einem Platzhalter (*) hat, der derselben Domäne entspricht (z. B.`*.example.com`).

c. Der Wert von State (Zustand) wird sofort in Disabled (Deaktiviert) geändert. Warten Sie bis der Wert Zuletzt geändert von Wird bereitgestellt in den Zeitstempel geändert wurde, als die Verteilung zuletzt geändert wurde.

   Da diese Änderung auf alle Edge-Standorte übertragen werden CloudFront muss, kann es einige Minuten dauern, bis die Aktualisierung abgeschlossen ist und Sie mit der Option Löschen die Verteilung löschen können.

6. Gehen Sie wie folgt vor, um die deaktivierte Verteilung zu löschen:

   a. Aktivieren Sie das Kontrollkästchen neben der ID für die Verteilung, die Sie löschen möchten.

   b. Wählen Sie Löschen und Löschen aus, um den Vorgang zu bestätigen.

Löschen Sie den Datensatz DNS

Informationen zum Löschen der öffentlich gehosteten Zone für die Domain (einschließlich des DNS Datensatzes) finden Sie Löschen Sie die öffentlich gehostete Zone für Ihre benutzerdefinierte Domäne im Amazon Route 53 Developer Guide. Wenn Sie nur den in Schritt 6 erstellten DNS Datensatz löschen möchten, gehen Sie wie folgt vor:

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Route 53-Konsole unter https://console.aws.amazon.com/route53/.
2. Wählen Sie im linken Navigationsbereich Hosted Zones (Bereitgestellte Zonen).
3. Wählen Sie auf der Seite Gehostete Zonen den Namen der gehosteten Zone aus, die Route 53 für Sie in Voraussetzungen erstellt hat (z. B. **`example.com`**).
4. Wählen Sie in der Liste der Datensätze das Kontrollkästchen neben den Datensätzen aus, die Sie löschen möchten (die Datensätze, die Sie in Schritt 6 erstellt haben).

   > **Note**
   >
   > Sie können keine Datensätze löschen, deren Type den Wert NS oder hat SOA.

5. Wählen Sie Delete Records (Datensätze löschen).
6. Um die Löschung zu bestätigen, klicken Sie auf Delete (Löschen).

Es dauert einige Zeit, bis Änderungen an Datensätzen auf die Route DNS 53-Server übertragen werden. Derzeit besteht die einzige Möglichkeit, um zu überprüfen, ob Ihre Änderungen übernommen wurden, darin, die GetChange API Aktion zu verwenden. Änderungen werden im Allgemeinen innerhalb von 60 Sekunden an alle Route 53-Name-Server übertragen.

Löschen Sie die öffentlich gehostete Zone für Ihre benutzerdefinierte Domäne

Warning

Wenn Sie Ihre Domänenregistrierung behalten möchten, aber die Weiterleitung von Internetdatenverkehr an Ihre Website oder Webanwendung beenden möchten, empfehlen wir, dass Sie Datensätze in der gehosteten Zone (wie im obigen Abschnitt beschrieben) löschen, statt die gehostete Zone zu löschen.
Wenn Sie eine gehostete Zone löschen, könnte jemand anderes die Domäne nutzen und den Datenverkehr über Ihren Domänennamen an die eigenen Ressourcen weiterleiten.
Wenn Sie eine gehostete Zone löschen, können Sie diese außerdem nicht wiederherstellen. Sie müssen eine neue gehostete Zone erstellen und Sie die Namensserver für Ihre Domain-Registrierung aktualisieren. Es kann bis zu 48 Stunden dauern, bis diese Einstellungen wirksam werden.
Wenn Sie möchten, dass die Domain im Internet nicht verfügbar ist, können Sie Ihren DNS Dienst zunächst auf einen kostenlosen DNS Dienst übertragen und dann die von Route 53 gehostete Zone löschen. Dadurch wird verhindert, dass future DNS Anfragen möglicherweise falsch weitergeleitet werden.

1. Wenn die Domain bei Route 53 registriert ist, finden Sie unter Hinzufügen oder Ändern von Nameservern und Glue-Datensätzen für eine Domain im Amazon Route 53-Entwicklerhandbuch Informationen darüber, wie Sie Route 53-Nameserver durch Nameserver für den neuen DNS Service ersetzen können.
2. Wenn die Domäne mit einer anderen Vergabestellt registriert wurde, verwenden Sie die Methode der Vergabestelle, um Namensserver für die Domäne zu ändern.

> **Note**
>
> Wenn Sie eine gehostete Zone für eine Subdomäne (`www.example.com`) löschen, müssen Sie keine Namensserver für die Domäne (`example.com`) ändern.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Route 53-Konsole unter https://console.aws.amazon.com/route53/.
2. Wählen Sie im linken Navigationsbereich Hosted Zones (Bereitgestellte Zonen).
3. Wählen Sie auf der Seite Hosted Zones den Namen für die gehostete Zone aus, die Sie löschen möchten.
4. Vergewissern Sie sich auf der Registerkarte Datensätze für Ihre gehostete Zone, dass die Hosting-Zone, die Sie löschen möchten, nur einen NS und einen SOADatensatz enthält.

   Enthält sie zusätzliche Datensätze, löschen Sie diese zuerst.

   Wenn Sie NS-Datensätze für Subdomänen in der gehosteten Zone erstellt haben, löschen Sie auch diese Datensätze.
5. Deaktivieren Sie auf der Registerkarte DNSSECSignatur für Ihre gehostete Zone das DNNSSEC Signieren, falls es aktiviert war. Weitere Informationen finden Sie unter Deaktivieren der DNSSEC Signierung im Amazon Route 53-Entwicklerhandbuch.
6. Wählen Sie oben auf der Detailseite der gehosteten Zone Zone löschen aus.
7. Geben Sie **`delete`** ein, um das Löschen zu bestätigen, und wählen Sie dann Löschen.

Löschen des benutzerdefinierten Domänennamens aus Route 53

Bei den meisten Top-Level-Domains (TLDs) können Sie die Registrierung löschen, wenn Sie sie nicht mehr benötigen. Wenn Sie eine Domainnamenregistrierung von Route 53 löschen, bevor die Registrierung abläuft, wird die Registrierungsgebühr AWS nicht zurückerstattet. Weitere Informationen finden Sie unter Löschen einer Domainnamenregistrierung im Amazon-Route 53-Entwicklerhandbuch.

> **Important**
>
> Wenn Sie die Domain zwischen einem anderen Registrar übertragen AWS-Konten oder die Domain an einen anderen Registrar übertragen möchten, löschen Sie die Domain nicht und gehen Sie davon aus, dass Sie sie sofort erneut registrieren. Lesen Sie stattdessen die entsprechende Dokumentation im Entwicklerhandbuch für Amazon Route 53:
>
> - Übertragung einer Domain auf eine andere AWS-Konto
> - Überträgt eine Domain von Amazon Route 53 zu einer anderen Vergabestelle

Löschen des Originalvideos im S3-Quell-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Bucket-Name den Namen des Buckets aus, in den Sie das Video inSchritt 2 (z. B. **`tutorial-bucket`**) hochgeladen haben.
4. Markieren Sie auf der Registerkarte Objekte das Kontrollkästchen neben dem Namen des Objekts, das Sie löschen möchten (z. B. `sample.mp4`).
5. Wählen Sie Löschen.
6. Geben Sie unter Objekte endgültig löschen? den Wert **`permanently delete`** um zu bestätigen, dass Sie dieses Objekt löschen möchten.
7. Wählen Sie Delete objects (Objekte löschen).

Löschen des S3-Quell-Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Liste Buckets das Optionsfeld neben dem Namen des Buckets aus, den Sie in Schritt 1 (z. B. **`tutorial-bucket`**) erstellt haben.
4. Wählen Sie Löschen.

5. Bestätigen Sie auf der Seite Delete bucket (Bucket löschen), dass Sie den Bucket löschen möchten. Geben Sie dazu den Bucket-Namen in das Textfeld ein und wählen Sie Delete bucket (Bucket löschen).

Nächste Schritte

Nachdem Sie dieses Tutorial abgeschlossen haben, können Sie die folgenden verwandten Anwendungsfälle näher untersuchen:

- Transkodieren Sie S3-Videos in Streaming-Formate, die für ein bestimmtes Fernsehgerät oder ein angeschlossenes Gerät benötigt werden, bevor Sie diese Videos bei einer CloudFront Distribution hosten.

  Informationen zur Verwendung von Amazon S3 Batch Operations AWS Lambda und AWS Elemental MediaConvert zur Batch-Transcodierung einer Sammlung von Videos in eine Vielzahl von Ausgabemedienformaten finden Sie unter. Tutorial: Batch-Transcodierung von Videos mit S3 Batch Operations
- Hosten Sie andere in S3 gespeicherte Objekte, wie Bilder, Audio, Animationsgrafiken, Stylesheets,HTML, JavaScript, React-Apps usw., mithilfe von CloudFront und Route 53.

  Ein Beispiel finden Sie unter Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne und Beschleunigen Sie Ihre Website mit Amazon CloudFront.
- Verwenden Sie Amazon S3 Transfer Acceleration, um schnelle und sichere Dateiübertragungen zu konfigurieren. Transfer Acceleration kann das Hochladen von Videos zu Ihrem S3-Bucket zur Übertragung größerer Videos über große Entfernungen beschleunigen. Transfer Acceleration verbessert die Übertragungsleistung, indem der Datenverkehr über die CloudFront weltweit verteilten Edge-Standorte und über die AWS Backbone-Netzwerke geleitet wird. Es verwendet auch Netzwerkprotokoll-Optimierungen. Weitere Informationen finden Sie unter Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration.

Tutorial: Konfigurieren einer statischen Website auf Amazon S3

> **Important**
>
> Amazon S3 wendet jetzt serverseitige Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3) als Basisverschlüsselungsebene für jeden Bucket in Amazon S3

an. Ab dem 5. Januar 2023 werden alle neuen Objekt-Uploads auf Amazon S3 ohne zusätzliche Kosten und ohne Auswirkungen auf die Leistung automatisch verschlüsselt. Der automatische Verschlüsselungsstatus für die Standardverschlüsselungskonfiguration des S3-Buckets und für das Hochladen neuer Objekte ist in den AWS CloudTrail Protokollen, im S3-Inventar, in der S3-Speicherlinse, in der Amazon S3 S3-Konsole und als zusätzlicher Amazon S3 API S3-Antwort-Header im AWS Command Line Interface und AWS SDKs verfügbar. Weitere Informationen finden Sie unter Standardverschlüsselung FAQ.

Sie können einen Amazon S3-Bucket so konfigurieren, dass er sich wie eine Website verhält. In diesem Beispiel sehen Sie die Schritte für das Hosten einer Website auf Amazon S3.

Important

Für das folgende Tutorial ist die Deaktivierung des öffentlichen Zugriffs erforderlich. Wir empfehlen, alle Einstellungen für das Blockieren des öffentlichen Zugriffs aktiviert zu lassen. Wenn Sie alle vier Block Public Access-Einstellungen aktiviert lassen und eine statische Website hosten möchten, können Sie Amazon CloudFront Origin Access Control (OAC) verwenden. Amazon CloudFront bietet die Funktionen, die für die Einrichtung einer sicheren statischen Website erforderlich sind. Statische Amazon S3 S3-Websites unterstützen nur HTTP Endpunkte. Amazon CloudFront verwendet den dauerhaften Speicher von Amazon S3 und bietet gleichzeitig zusätzliche Sicherheitsheader, wie HTTPS z. HTTPSerhöht die Sicherheit, indem es eine normale HTTP Anfrage verschlüsselt und vor gängigen Cyberangriffen schützt. Weitere Informationen finden Sie unter Erste Schritte mit einer sicheren statischen Website im Amazon CloudFront Developer Guide.

Themen

- Schritt 1: Erstellen eines Buckets
- Schritt 2: Aktivieren des statischen Website-Hostings
- Schritt 3: Bearbeiten der Block-Public-Access-Einstellungen
- Schritt 4: Hinzufügen einer Bucket-Richtlinie, die den Inhalt Ihres Buckets öffentlich verfügbar macht
- Schritt 5: Konfigurieren eines Indexdokuments
- Schritt 6: Konfigurieren eines Fehlerdokuments

- Schritt 7: Testen des Website-Endpunkts
- Schritt 8: Bereinigen

Schritt 1: Erstellen eines Buckets

Die folgenden Anweisungen geben einen Überblick darüber, wie Sie Ihre Buckets für das Website-Hosting erstellen. Eine ausführliche step-by-step Anleitung zur Erstellung eines Buckets finden Sie unterErstellen eines Buckets.

So erstellen Sie einen Bucket

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie Create Bucket (Bucket erstellen).
3. Geben Sie den Bucket-Namen ein (z. B. **`example.com`**).
4. Wählen Sie die Region aus, in der Sie Ihren Bucket erstellen möchten.

   Wählen Sie eine Region in Ihrer Nähe aus, um Latenz und Kosten auf einem Minimum zu halten oder behördliche Vorschriften zu erfüllen. Die von Ihnen ausgewählte Region bestimmt Ihren Amazon-S3-Website-Endpunkt. Weitere Informationen finden Sie unter Website-Endpunkte.
5. Um die Standardeinstellungen zu übernehmen und den Bucket zu erstellen, wählen Sie Create (Erstellen).

Schritt 2: Aktivieren des statischen Website-Hostings

Nach der Erstellung eines Buckets können Sie das statische Website-Hosting für Ihren Bucket aktivieren. Sie können einen neuen Bucket erstellen oder einen vorhandenen Bucket verwenden.

So aktivieren Sie das statische Website-Hosting

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie das Hosting statischer Websites aktivieren wollen.
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie unter Static website hosting (Hosting statischer Websites) Edit (Bearbeiten) aus.

5. Wählen Sie Use this bucket to host a website (Diesen Bucket zum Hosten einer Website verwenden).
6. Wählen Sie unter Static website hosting (Hosting statischer Websites) die Option Enable (Aktivieren) aus
7. Geben Sie unter Index document (Index-Dokument) den Dateinamen des Index-Dokuments ein, der typischerweise `index.html` ist.

   Beim Namen des Indexdokuments wird zwischen Groß- und Kleinschreibung unterschieden und er muss exakt mit dem Dateinamen des HTML Indexdokuments übereinstimmen, das Sie in Ihren S3-Bucket hochladen möchten. Wenn Sie Ihren Bucket für das Hosting von Websites konfigurieren, müssen Sie ein Indexdokument angeben. Amazon S3 gibt dieses Indexdokument zurück, wenn Anfragen an die Root-Domäne oder einen der Unterordner gestellt werden. Weitere Informationen finden Sie unter Konfigurieren eines Indexdokuments.
8. Um ein eigenes benutzerdefiniertes Fehlerdokument für Fehler der Klasse 4XX bereitzustellen, geben Sie unter Fehlerdokument den Dateinamen des benutzerdefinierten Fehlerdokuments ein.

   Beim Namen des Fehlerdokuments wird zwischen Groß- und Kleinschreibung unterschieden und er muss genau mit dem Dateinamen des HTML Fehlerdokuments übereinstimmen, das Sie in Ihren S3-Bucket hochladen möchten. Wenn Sie kein benutzerdefiniertes Fehlerdokument angeben und ein Fehler auftritt, gibt Amazon S3 ein HTML Standard-Fehlerdokument zurück. Weitere Informationen finden Sie unter Konfigurieren eines benutzerdefinierten Fehlerdokuments.
9. (Optional) Wenn Sie erweiterte Umleitungsregeln angeben möchten, geben Sie im Feld JSONUmleitungsregeln eine Beschreibung der Regeln ein.

   Beispielsweise können Sie bedingt Anfragen abhängig von bestimmten Objektschlüsselnamen oder Präfixen in der Anfrage weiterleiten. Weitere Informationen finden Sie unter Konfigurieren von Umleitungsregeln für die Verwendung von fortschrittliche bedingten Umleitungen.
10. Wählen Sie Save Changes (Änderungen speichern).

    Amazon S3 ermöglicht statisches Website-Hosting für Ihren Bucket. Unten auf der Seite sehen Sie unter Static website hosting (Hosting statischer Websites) den Website-Endpunkt für Ihren Bucket.
11. Notieren Sie unter Static website hosting (Statisches Website-Hosting) den Wert für Endpoint (Endpunkt).

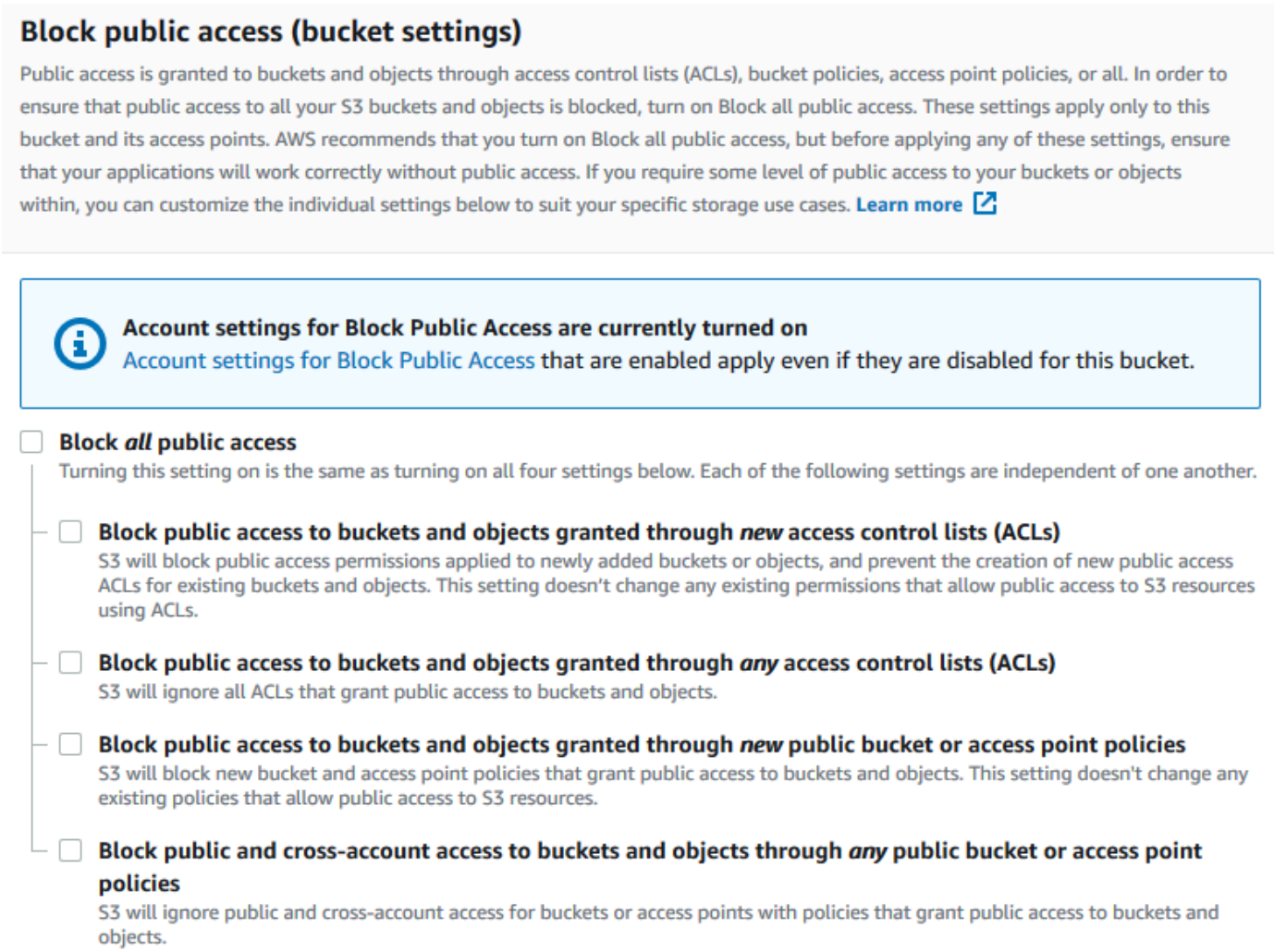
Der Endpoint (Endpunkt) ist der Amazon-S3-Website-Endpunkt für Ihren Bucket. Nachdem Sie den Bucket als statische Website konfiguriert haben, können Sie diesen Endpunkt verwenden, um Ihre Website zu testen.

Schritt 3: Bearbeiten der Block-Public-Access-Einstellungen

Standardmäßig blockiert Amazon S3 den öffentlichen Zugriff auf Ihr Konto und Ihre Buckets. Wenn Sie einen Bucket verwenden möchten, um eine statische Website zu hosten, können Sie diese Schritte verwenden, um Ihre Einstellungen für Block Public Access zu bearbeiten:

> **Warning**
>
> Bevor Sie diese Schritte ausführen, sollten Sie sicherstellen, Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher dass Sie die Risiken verstehen und akzeptieren, die mit der Gewährung eines öffentlichen Zugriffs verbunden sind. Wenn Sie die Einstellungen für Block Public Access deaktivieren, um Ihren Bucket öffentlich zu machen, kann jeder im Internet auf Ihren Bucket zugreifen. Wir empfehlen Ihnen, den gesamten öffentlichen Zugriff auf Ihre Buckets zu blockieren.

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie den Namen des Buckets aus, den Sie als statische Website konfiguriert haben.
3. Wählen Sie Permissions (Berechtigungen).
4. Wählen Sie unter Block public access (bucket settings) (Öffentlichen Zugriff blockieren (Bucket-Einstellungen)), die Option Edit (Bearbeiten).
5. Löschen Sie Block all public access (Gesamten öffentlichen Zugriff blockieren) und wählen Sie Save (Speichern).



Amazon S3 deaktiviert die Einstellungen für den Block Public Access für Ihren Bucket. Um eine öffentliche statische Website zu erstellen, müssen Sie möglicherweise auch die Einstellungen zum Blockieren des öffentlichen Zugriffs für Ihr Konto bearbeiten, bevor Sie eine Bucket-Richtlinie hinzufügen. Wenn die Einstellungen „Öffentlichen Zugriff blockieren“ für Ihr Konto derzeit aktiviert sind, wird unter „Öffentlichen Zugriff blockieren“ (Bucket-Einstellungen) ein Hinweis angezeigt.

Schritt 4: Hinzufügen einer Bucket-Richtlinie, die den Inhalt Ihres Buckets öffentlich verfügbar macht

Nachdem Sie die Einstellungen für S3 Block Public Access bearbeitet haben, können Sie eine Bucket-Richtlinie hinzufügen, um öffentlichen Lesezugriff auf den Bucket zu gewähren. Wenn Sie öffentlichen Lesezugriff gewähren, kann jeder im Internet auf Ihren Bucket zugreifen.

> **Important**
>
> Die zuvor genannte Richtlinie ist nur ein Beispiel und erlaubt Vollzugriff auf die Inhalte Ihres Buckets. Bevor Sie mit diesem Schritt fortfahren, lesen Sie den Abschnitt Wie kann ich die Dateien in meinem Amazon-S3-Bucket sichern?, um sicherzustellen, dass Sie die bewährten Methoden zum Sichern der Dateien in Ihrem S3-Bucket und die Risiken in Zusammenhang mit der Gewährung von öffentlichem Zugriff kennen.

1. Wählen Sie unter Buckets den Namen Ihres Buckets aus.
2. Wählen Sie Permissions (Berechtigungen).
3. Wählen Sie unter Bucket Policy (Bucket-Richtlinie) Edit (Bearbeiten).
4. Um öffentlichen Lesezugriff auf Ihre Website zu gewähren, kopieren Sie die folgende Bucket-Richtlinie und fügen Sie sie in den Bucket policy editor (Bucket-Richtlinieneditor) ein.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::Bucket-Name/*"
            ]
        }
    ]
}
```

5. Aktualisieren Sie den `Resource` zu Ihrem Bucket-Namen.

   Im obigen Beispiel für eine Bucket-Richtlinie *`Bucket-Name`* ist ein Platzhalter für den Bucket-Namen. Um diese Bucket-Richtlinie mit Ihrem eigenen Bucket zu verwenden, müssen Sie diesen Namen so aktualisieren, dass er mit Ihrem Bucket übereinstimmt.
6. Wählen Sie Save Changes (Änderungen speichern).

Es wird eine Meldung angezeigt, die darauf hinweist, dass die Bucket-Richtlinie erfolgreich hinzugefügt wurde.

Wenn die Fehlermeldung `Policy has invalid resource` angezeigt wird, bestätigen Sie, dass der Bucket-Name in der Bucket-Richtlinie mit Ihrem Bucket-Namen übereinstimmt. Informationen zum Hinzufügen einer Bucket-Richtlinie finden Sie unter Wie füge ich eine S3-Bucket-Richtlinie hinzu?

Wenn Sie eine Fehlermeldung erhalten und die Bucket-Richtlinie nicht speichern können, überprüfen Sie Ihr Konto und die Bucket-Einstellungen für Block Public Access, um zu bestätigen, dass Sie den öffentlichen Zugriff auf den Bucket zulassen.

Schritt 5: Konfigurieren eines Indexdokuments

Wenn Sie das statische Website-Hosting für Ihren Bucket aktivieren, geben Sie den Namen des Indexdokuments ein (z. B, **`index.html`**). Nachdem Sie das statische Website-Hosting für den Bucket aktiviert haben, laden Sie eine HTML Datei mit diesem Indexdokumentnamen in Ihren Bucket hoch.

So konfigurieren Sie das Indexdokument

1. Erstellen Sie eine Datei `index.html`.

   Wenn Sie keine `index.html` Datei haben, können Sie wie folgt HTML eine Datei erstellen:

   ```
   <html xmlns="http://www.w3.org/1999/xhtml" >
   <head>
       <title>My Website Home Page</title>
   </head>
   <body>
     <h1>Welcome to my website</h1>
     <p>Now hosted on Amazon S3!</p>
   </body>
   </html>
   ```

2. Speichern Sie die Indexdatei lokal.

   Der Dateiname des Indexdokuments muss genau mit dem Namen des Indexdokuments übereinstimmen, den Sie im Dialogfeld Static website hosting (Statisches Website-Hosting) eingeben. Beim Namen des Indexdokuments wird die Groß- und Kleinschreibung berücksichtigt.

Wenn Sie beispielsweise im Dialogfeld Static website hosting (Statisches Website-Hosting) `index.html` als den Namen des Index document (Indexdokuments) eingeben, muss der Dateiname des Indexdokuments ebenfalls `index.html` und nicht `Index.html` lauten.

3. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
4. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie zum Hosten einer statischen Website verwenden möchten.
5. Aktivieren Sie das Hosting statischer Websites für Ihren Bucket und geben Sie den exakten Namen Ihres Indexdokuments ein (z. B. `index.html`). Weitere Informationen finden Sie unter Aktivieren des Website-Hostings.

   Fahren Sie mit Schritt 6 fort, nachdem Sie das Hosting statischer Websites aktiviert haben.
6. Führen Sie einen der folgenden Schritte aus, um das Indexdokument in Ihren Bucket hochzuladen:

   - Ziehen Sie die Indexdatei per Drag & Drop in das Konsolen-Bucket-Verzeichnis.
   - Wählen Sie Upload (Hochladen) und folgen Sie den Anweisungen zur Auswahl und zum Hochladen der Indexdatei.

   step-by-stepAnweisungen finden Sie unterObjekte hochladen.
7. (Optional) Laden Sie andere Website-Inhalte in Ihren Bucket hoch.

Schritt 6: Konfigurieren eines Fehlerdokuments

Wenn Sie das Hosting statischer Websites für Ihren Bucket aktivieren, geben Sie den Namen des Fehlerdokuments ein (z. B, **`404.html`**). Nachdem Sie das statische Website-Hosting für den Bucket aktiviert haben, laden Sie eine HTML Datei mit diesem Fehlerdokumentnamen in Ihren Bucket hoch.

So konfigurieren Sie ein Fehlerdokument

1. Erstellen Sie ein Fehlerdokument, z. B. `404.html`.
2. Speichern Sie die Fehlerdokumentdatei lokal.

   Der Name des Fehlerdokuments unterscheidet zwischen Groß- und Kleinschreibung und muss genau mit dem Namen übereinstimmen, den Sie beim Aktivieren des statischen Website-Hostings eingeben. Wenn Sie beispielsweise `404.html` im Dialogfeld Hosten einer statischen

Website als Namen des Fehlerdokuments eingeben, muss der Dateiname des Fehlerdokuments ebenfalls `404.html` lauten.

3. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
4. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie zum Hosten einer statischen Website verwenden möchten.
5. Aktivieren Sie das Hosting statischer Websites für Ihren Bucket und geben Sie den exakten Namen Ihres Fehlerdokuments ein (z. B, `404.html`). Weitere Informationen erhalten Sie unter Aktivieren des Website-Hostings und Konfigurieren eines benutzerdefinierten Fehlerdokuments.

   Fahren Sie mit Schritt 6 fort, nachdem Sie das Hosting statischer Websites aktiviert haben.
6. Führen Sie einen der folgenden Schritte aus, um das Fehlerdokument in Ihren Bucket hochzuladen:

   - Ziehen Sie die Fehlerdokumentdatei in das Konsolen-Bucket-Verzeichnis.
   - Wählen Sie Upload (Hochladen) und folgen Sie den Anweisungen zur Auswahl und zum Hochladen der Indexdatei.

   step-by-stepAnweisungen finden Sie unterObjekte hochladen.

Schritt 7: Testen des Website-Endpunkts

Nach der Konfigurierung des statischen Website-Hostings für Ihren Bucket können Sie den Website-Endpunkt testen.

> **Note**
>
> Amazon S3 unterstützt keinen HTTPS Zugriff auf die Website. Wenn Sie verwenden möchtenHTTPS, können Sie Amazon verwenden, CloudFront um eine statische Website bereitzustellen, die auf Amazon S3 gehostet wird.
> Weitere Informationen finden Sie unter Wie verwende ich, CloudFront um eine statische Website bereitzustellen, die auf Amazon S3 gehostet wird? und Erforderlich HTTPS für die Kommunikation zwischen Zuschauern und CloudFront.

1. Wählen Sie unter Buckets den Namen Ihres Buckets aus.

2. Wählen Sie Properties (Eigenschaften).
3. Wählen Sie unten auf der Seite unter Static website hosting (Hosting statischer Websites) Ihren Bucket-Website-Endpunkt.

   Ihr Indexdokument wird in einem separaten Browserfenster geöffnet.

Sie haben jetzt eine auf Amazon S3 gehostete Website. Diese Website steht am Amazon S3-Website-Endpunkt zur Verfügung. Möglicherweise haben Sie jedoch eine Domäne wie `example.com`, die den Inhalt von der von Ihnen erstellten Website bereitstellen soll. Sie könnten auch die Amazon S3-Unterstützung der Root-Domäne nutzen, um Anforderungen für `http://www.example.com` und `http://example.com` zu bedienen. Dafür sind zusätzliche Schritte erforderlich. Ein Beispiel finden Sie unter Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne.

Schritt 8: Bereinigen

Wenn Sie die statische Website nur zur Übung erstellt haben, löschen Sie die AWS -Ressourcen, die Sie zugewiesen haben, damit keine weiteren Kosten für Sie anfallen. Nachdem Sie Ihre AWS Ressourcen gelöscht haben, ist Ihre Website nicht mehr verfügbar. Weitere Informationen finden Sie unter Löschen eines Buckets.

Tutorial: Konfigurieren einer statischen Website mithilfe einer benutzerdefinierten bei Route 53 registrierten Domäne

Angenommen, Sie wollen eine statische Website auf Amazon S3 hosten. Sie haben eine Domäne bei Amazon Route 53 registriert (zum Beispiel `example.com`), und Sie möchten, dass Anforderungen für `http://www.example.com` und `http://example.com` von Ihren Amazon-S3-Inhalten aus bedient werden. Sie können diese Anleitung verwenden, um zu erfahren, wie Sie eine statische Website hosten und Umleitungen auf Amazon S3 für eine Website mit einem benutzerdefinierten Domänennamen, der bei Route 53 registriert ist, anlegen können. Sie können mit einer vorhandenen Website arbeiten, die Sie auf Amazon S3 hosten möchten, oder diesen Walkthrough verwenden, um bei Null anzufangen.

Nachdem Sie diese Anleitung abgeschlossen haben, können Sie optional Amazon verwenden, CloudFront um die Leistung Ihrer Website zu verbessern. Weitere Informationen finden Sie unter Beschleunigen Sie Ihre Website mit Amazon CloudFront.

> **Note**
>
> Amazon S3 S3-Website-Endpunkte unterstützen keine HTTPS Zugriffspunkte. Wenn Sie verwenden möchtenHTTPS, können Sie Amazon verwenden, CloudFront um eine statische Website bereitzustellen, die auf Amazon S3 gehostet wird.
> Ein Tutorial zum sicheren Hosten Ihrer Inhalte mit CloudFront Amazon S3 finden Sie unterTutorial: Hosten von On-Demand-Streaming-Videos mit Amazon S3 CloudFront, Amazon und Amazon Route 53. Weitere Informationen finden Sie unter Wie verwende ich, CloudFront um eine statische Website bereitzustellen, die auf Amazon S3 gehostet wird? und Erforderlich HTTPS für die Kommunikation zwischen Zuschauern und CloudFront.

Automatisierung der statischen Website-Einrichtung mit einer Vorlage AWS CloudFormation

Sie können eine AWS CloudFormation Vorlage verwenden, um die Einrichtung Ihrer statischen Website zu automatisieren. Die AWS CloudFormation Vorlage enthält die Komponenten, die Sie zum Hosten einer sicheren statischen Website benötigen, sodass Sie sich mehr auf den Inhalt Ihrer Website und weniger auf die Konfiguration von Komponenten konzentrieren können.

Die AWS CloudFormation Vorlage umfasst die folgenden Komponenten:

- Amazon S3 – Erstellt einen Amazon-S3-Bucket zum Hosten Ihrer statischen Website.
- CloudFront — Erstellt eine CloudFront Distribution, um Ihre statische Website zu beschleunigen.
- Lambda@Edge – Verwendet Lambda@Edge, um jeder Serverantwort Sicherheits-Header hinzuzufügen. Sicherheits-Header sind eine Gruppe von Headern in der Webserverantwort, die Webbrowser anweisen, zusätzliche Sicherheitsvorkehrungen zu treffen. Weitere Informationen finden Sie im Blogbeitrag Hinzufügen von HTTP Sicherheitsheadern mithilfe von Lambda @Edge und Amazon. CloudFront

Diese AWS CloudFormation Vorlage steht Ihnen zum Herunterladen und Verwenden zur Verfügung. Informationen und Anweisungen finden Sie unter Erste Schritte mit einer sicheren statischen Website im Amazon CloudFront Developer Guide.

Themen

- Bevor Sie beginnen
- Schritt 1: Registrieren einer benutzerdefinierten Domäne bei Route 53
- Schritt 2: Erstellen von zwei Buckets

- Schritt 3: Konfigurieren Ihres Stammdomänen-Buckets für Website-Hosting
- Schritt 4: Konfigurieren Ihres Subdomänen-Buckets für die Website-Umleitung
- Schritt 5: Konfigurieren der Protokollierung für Website-Datenverkehr
- Schritt 6: Hochladen des Index und des Website-Inhalts
- Schritt 7: Hochladen eines Fehlerdokuments
- Schritt 8: Bearbeiten der S3 Block Public Access-Einstellungen
- Schritt 9: Anfügen einer Bucket-Richtlinie
- Schritt 10: Testen Ihres Domänen-Endpunkts
- Schritt 11: Hinzufügen von Aliasdatensätzen für Ihre Domäne und Subdomäne
- Schritt 12: Testen der Website
- Beschleunigen Sie Ihre Website mit Amazon CloudFront
- Bereinigung Ihrer Beispielressourcen

Bevor Sie beginnen

Während Sie den Schritten in diesem Beispiel folgen, arbeiten Sie mit den folgenden Services:

Amazon Route 53 – Sie verwenden Route 53, um Domänen zu registrieren und zu definieren, wohin der Internetdatenverkehr für Ihre Domäne geleitet werden soll. Das Beispiel zeigt, wie Sie Route 53-Aliaseinträge erstellen, die den Verkehr für Ihre Domain (`example.com`) und Subdomain (`www.example.com`) an einen Amazon S3-Bucket weiterleiten, der eine HTML Datei enthält.

Amazon S3 – Mit Amazon S3 erstellen Sie Buckets, laden eine Beispielseite für eine Website hoch, konfigurieren Berechtigungen, sodass jeder den Inhalt sehen kann, und konfigurieren dann die Buckets für das Website-Hosting.

Schritt 1: Registrieren einer benutzerdefinierten Domäne bei Route 53

Wenn Sie noch keinen registrierten Domänennamen haben, wie z. B. `example.com`, registrieren Sie einen mit Route 53. Weitere Informationen finden Sie unter Registrieren einer neuen Domäne im Amazon-Route 53-Entwicklerhandbuch. Nachdem Sie Ihren Domänennamen registriert haben, können Sie Ihre Amazon-S3-Buckets für das Website-Hosting erstellen und konfigurieren.

Schritt 2: Erstellen von zwei Buckets

Um Anforderungen von sowohl der Stammdomäne als auch von der Subdomäne zu unterstützen, erstellen Sie zwei Buckets.

- Domänen-Bucket – `example.com`
- Subdomänen-Bucket – `www.example.com`

Diese Bucket-Namen müssen genau mit Ihrem Domänennamen übereinstimmen. In diesem Beispiel lautet der Domänenname `example.com`. Sie hosten Ihre Inhalte aus dem Stammdomänen-Bucket (`example.com`). Sie erstellen eine Umleitungsanforderung für den Subdomänen-Bucket (`www.example.com`). Falls ein Benutzer `www.example.com` in seinen Browser eingibt, wird er zu `example.com` umgeleitet und sieht den Inhalt, der im Amazon-S3-Bucket mit diesem Namen gehostet wird.

So erstellen Sie Ihre Buckets für Website-Hosting

Die folgenden Anweisungen geben einen Überblick darüber, wie Sie Ihre Buckets für das Website-Hosting erstellen. Eine ausführliche step-by-step Anleitung zur Erstellung eines Buckets finden Sie unterErstellen eines Buckets.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Erstellen Sie den Stammdomänen-Bucket:
   a. Wählen Sie Create Bucket (Bucket erstellen) aus.
   b. Geben Sie den Bucket-Namen ein (z. B. **`example.com`**).
   c. Wählen Sie die Region aus, in der Sie Ihren Bucket erstellen möchten.

      Wählen Sie eine Region in Ihrer Nähe aus, um Latenz und Kosten auf einem Minimum zu halten oder behördliche Vorschriften zu erfüllen. Die von Ihnen ausgewählte Region bestimmt Ihren Amazon-S3-Website-Endpunkt. Weitere Informationen finden Sie unter Website-Endpunkte.
   d. Um die Standardeinstellungen zu übernehmen und den Bucket zu erstellen, wählen Sie Create (Erstellen).
3. Erstellen Sie Ihren Subdomänen-Bucket:
   a. Wählen Sie Create Bucket (Bucket erstellen) aus.
   b. Geben Sie den Bucket-Namen ein (z. B. **`www.example.com`**).
   c. Wählen Sie die Region aus, in der Sie Ihren Bucket erstellen möchten.

Wählen Sie eine Region in Ihrer Nähe aus, um Latenz und Kosten auf einem Minimum zu halten oder behördliche Vorschriften zu erfüllen. Die von Ihnen ausgewählte Region bestimmt Ihren Amazon-S3-Website-Endpunkt. Weitere Informationen finden Sie unter Website-Endpunkte.

d. Um die Standardeinstellungen zu übernehmen und den Bucket zu erstellen, wählen Sie Create (Erstellen).

Im nächsten Schritt konfigurieren Sie `example.com` für das Website-Hosting.

Schritt 3: Konfigurieren Ihres Stammdomänen-Buckets für Website-Hosting

In diesem Schritt konfigurieren Sie Ihren Stammdomänen-Bucket (`example.com`) als Website. Dieser Bucket enthält Ihren Website-Inhalt. Wenn Sie einen Bucket für das Website-Hosting konfigurieren, können Sie über auf die Website zugreife Website-Endpunkte.

So aktivieren Sie das statische Website-Hosting

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie in der Liste Buckets den Namen des Buckets aus, für den Sie das Hosting statischer Websites aktivieren wollen.
3. Wählen Sie Properties (Eigenschaften).
4. Wählen Sie unter Static website hosting (Hosting statischer Websites) Edit (Bearbeiten) aus.
5. Wählen Sie Use this bucket to host a website (Diesen Bucket zum Hosten einer Website verwenden).
6. Wählen Sie unter Static website hosting (Hosting statischer Websites) die Option Enable (Aktivieren) aus
7. Geben Sie unter Index document (Index-Dokument) den Dateinamen des Index-Dokuments ein, der typischerweise `index.html` ist.

   Beim Namen des Indexdokuments wird zwischen Groß- und Kleinschreibung unterschieden und er muss exakt mit dem Dateinamen des HTML Indexdokuments übereinstimmen, das Sie in Ihren S3-Bucket hochladen möchten. Wenn Sie Ihren Bucket für das Hosting von Websites konfigurieren, müssen Sie ein Indexdokument angeben. Amazon S3 gibt dieses Indexdokument zurück, wenn Anfragen an die Root-Domäne oder einen der Unterordner gestellt werden. Weitere Informationen finden Sie unter Konfigurieren eines Indexdokuments.

8. Um ein eigenes benutzerdefiniertes Fehlerdokument für Fehler der Klasse 4XX bereitzustellen, geben Sie unter Fehlerdokument den Dateinamen des benutzerdefinierten Fehlerdokuments ein.

   Beim Namen des Fehlerdokuments wird zwischen Groß- und Kleinschreibung unterschieden und er muss genau mit dem Dateinamen des HTML Fehlerdokuments übereinstimmen, das Sie in Ihren S3-Bucket hochladen möchten. Wenn Sie kein benutzerdefiniertes Fehlerdokument angeben und ein Fehler auftritt, gibt Amazon S3 ein HTML Standard-Fehlerdokument zurück. Weitere Informationen finden Sie unter Konfigurieren eines benutzerdefinierten Fehlerdokuments.
9. (Optional) Wenn Sie erweiterte Umleitungsregeln angeben möchten, geben Sie im Feld JSONUmleitungsregeln eine Beschreibung der Regeln ein.

   Beispielsweise können Sie bedingt Anfragen abhängig von bestimmten Objektschlüsselnamen oder Präfixen in der Anfrage weiterleiten. Weitere Informationen finden Sie unter Konfigurieren von Umleitungsregeln für die Verwendung von fortschrittliche bedingten Umleitungen.
10. Wählen Sie Save Changes (Änderungen speichern).

    Amazon S3 ermöglicht statisches Website-Hosting für Ihren Bucket. Unten auf der Seite sehen Sie unter Static website hosting (Hosting statischer Websites) den Website-Endpunkt für Ihren Bucket.
11. Notieren Sie unter Static website hosting (Statisches Website-Hosting) den Wert für Endpoint (Endpunkt).

    Der Endpoint (Endpunkt) ist der Amazon-S3-Website-Endpunkt für Ihren Bucket. Nachdem Sie den Bucket als statische Website konfiguriert haben, können Sie diesen Endpunkt verwenden, um Ihre Website zu testen.

Nachdem Sie die Blockierungseinstellungen für den öffentlichen Zugriff bearbeitet und eine Bucket-Richtlinie hinzugefügt haben, die öffentlichen Lesezugriff ermöglicht, können Sie den Website-Endpunkt verwenden, um auf Ihre Website zuzugreifen.

Im nächsten Schritt konfigurieren Sie Ihre Subdomäne (`www.example.com`) zur Weiterleitung von Anforderungen an Ihre Domäne (`example.com`).

Schritt 4: Konfigurieren Ihres Subdomänen-Buckets für die Website-Umleitung

Nachdem Sie Ihren Stammdomänen-Bucket für das Website-Hosting konfiguriert haben, können Sie Ihren Subdomänen-Bucket so konfigurieren, dass alle Anforderungen zu der Domäne umgeleitet

werden. In diesem Beispiel werden alle Anforderungen für `www.example.com` an `example.com` umgeleitet.

So konfigurieren Sie eine Umleitungsanforderung

1. Wählen Sie in der Amazon-S3-Konsole in der Liste Buckets Ihren Subdomänen-Bucket aus (in diesem Beispiel `www.example.com`).
2. Wählen Sie Properties (Eigenschaften).
3. Wählen Sie unter Static website hosting (Hosting statischer Websites) Edit (Bearbeiten) aus.
4. Wählen Sie Redirect requests for an object (Anfragen für ein Objekt umleiten).
5. Geben Sie im Feld Target bucket (Ziel-Bucket) Ihre Root-Domäne ein, z. B. **`example.com`**.
6. Wählen Sie für Protocol (Protokoll) die Option http aus.
7. Wählen Sie Save Changes (Änderungen speichern).

Schritt 5: Konfigurieren der Protokollierung für Website-Datenverkehr

Wenn Sie die Anzahl der Besucher nachverfolgen möchten, die auf Ihre Website zugreifen, können Sie optional die Protokollierung für Ihren Stammdomänen-Bucket aktivieren. Weitere Informationen finden Sie unter Protokollieren von Anfragen mit Server-Zugriffsprotokollierung. Wenn Sie Amazon verwenden möchten, CloudFront um Ihre Website zu beschleunigen, können Sie auch die CloudFront Protokollierung verwenden.

So aktivieren Sie die Server-Zugriffsprotokollierung für Ihren Stammdomänen-Bucket

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Erstellen Sie in derselben Region, in der Sie den Bucket erstellt haben, der als statische Website konfiguriert ist, einen Bucket für die Protokollierung, z. B. `logs.example.com`.
3. Erstellen Sie einen Ordner für die Protokolldateien der Server-Zugriffsprotokollierung (z. B, `logs`).
4. (Optional) Wenn Sie die Leistung Ihrer Website verbessern CloudFront möchten, erstellen Sie einen Ordner für die CloudFront Protokolldateien (z. B.`cdn`).

> **Important**
>
> Wenn Sie eine Distribution erstellen oder aktualisieren und die CloudFront Protokollierung aktivieren, CloudFront aktualisiert die Bucket-Zugriffskontrollliste

(ACL), sodass das `awslogsdelivery` Konto `FULL_CONTROL` berechtigt ist, Logs in Ihren Bucket zu schreiben. Weitere Informationen finden Sie unter Erforderliche Berechtigungen für die Konfiguration der Standardprotokollierung und für den Zugriff auf Ihre Protokolldateien im Amazon CloudFront Developer Guide. Wenn der Bucket, der die Logs speichert, die vom Bucket-Besitzer erzwungene Einstellung zur Deaktivierung von S3 Object Ownership verwendetACLs, CloudFront können keine Logs in den Bucket geschrieben werden. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

5. Wählen Sie in der Liste Buckets Ihren Stammdomänen-Bucket aus.
6. Wählen Sie Properties (Eigenschaften).
7. Wählen Sie unter Server access logging (Server-Zugriffsprotokollierung) Edit (Bearbeiten).
8. Wählen Sie Enable aus.
9. Wählen Sie im Ziel-Bucket den Bucket und das Ordnerziel für die Server-Zugriffsprotokolle aus:
   - Navigieren Sie zum Ordner- und Bucket-Speicherort:
     1. Wählen Sie Browse S3 (S3 durchsuchen).
     2. Wählen Sie den Bucket-Namen und dann den Ordner mit den Protokollen aus.
     3. Wählen Sie Choose path (Pfad wählen).
   - Geben Sie den S3-Bucket-Pfad ein, z. B, `s3://logs.example.com/logs/`.
10. Wählen Sie Save Changes (Änderungen speichern).

    In Ihrem Protokoll-Bucket können Sie jetzt auf Ihre Protokolle zugreifen. Amazon S3 schreibt Website-Zugriffsprotokolle alle zwei Stunden in Ihren Bucket zur Protokollierung.

Schritt 6: Hochladen des Index und des Website-Inhalts

Als Nächstes laden Sie das Indexdokument und den optionalen Website-Inhalt zum Stammdomänen-Bucket hoch.

Wenn Sie das statische Website-Hosting für Ihren Bucket aktivieren, geben Sie den Namen des Indexdokuments ein (z. B, **`index.html`**). Nachdem Sie das statische Website-Hosting für den Bucket aktiviert haben, laden Sie eine HTML Datei mit diesem Indexdokumentnamen in Ihren Bucket hoch.

So konfigurieren Sie das Indexdokument

1. Erstellen Sie eine Datei `index.html`.

   Wenn Sie keine `index.html` Datei haben, können Sie wie folgt HTML eine Datei erstellen:

   ```
   <html xmlns="http://www.w3.org/1999/xhtml" >
   <head>
       <title>My Website Home Page</title>
   </head>
   <body>
     <h1>Welcome to my website</h1>
     <p>Now hosted on Amazon S3!</p>
   </body>
   </html>
   ```

2. Speichern Sie die Indexdatei lokal.

   Der Dateiname des Indexdokuments muss genau mit dem Namen des Indexdokuments übereinstimmen, den Sie im Dialogfeld Static website hosting (Statisches Website-Hosting) eingeben. Beim Namen des Indexdokuments wird die Groß- und Kleinschreibung berücksichtigt. Wenn Sie beispielsweise im Dialogfeld Static website hosting (Statisches Website-Hosting) `index.html` als den Namen des Index document (Indexdokuments) eingeben, muss der Dateiname des Indexdokuments ebenfalls `index.html` und nicht `Index.html` lauten.

3. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.

4. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie zum Hosten einer statischen Website verwenden möchten.

5. Aktivieren Sie das Hosting statischer Websites für Ihren Bucket und geben Sie den exakten Namen Ihres Indexdokuments ein (z. B. `index.html`). Weitere Informationen finden Sie unter Aktivieren des Website-Hostings.

   Fahren Sie mit Schritt 6 fort, nachdem Sie das Hosting statischer Websites aktiviert haben.

6. Führen Sie einen der folgenden Schritte aus, um das Indexdokument in Ihren Bucket hochzuladen:

   - Ziehen Sie die Indexdatei per Drag & Drop in das Konsolen-Bucket-Verzeichnis.
   - Wählen Sie Upload (Hochladen) und folgen Sie den Anweisungen zur Auswahl und zum Hochladen der Indexdatei.

step-by-stepAnweisungen finden Sie unterObjekte hochladen.

7. (Optional) Laden Sie andere Website-Inhalte in Ihren Bucket hoch.

Schritt 7: Hochladen eines Fehlerdokuments

Wenn Sie das Hosting statischer Websites für Ihren Bucket aktivieren, geben Sie den Namen des Fehlerdokuments ein (z. B, **`404.html`**). Nachdem Sie das statische Website-Hosting für den Bucket aktiviert haben, laden Sie eine HTML Datei mit diesem Fehlerdokumentnamen in Ihren Bucket hoch.

So konfigurieren Sie ein Fehlerdokument

1. Erstellen Sie ein Fehlerdokument, z. B. `404.html`.
2. Speichern Sie die Fehlerdokumentdatei lokal.

   Der Name des Fehlerdokuments unterscheidet zwischen Groß- und Kleinschreibung und muss genau mit dem Namen übereinstimmen, den Sie beim Aktivieren des statischen Website-Hostings eingeben. Wenn Sie beispielsweise `404.html` im Dialogfeld Hosten einer statischen Website als Namen des Fehlerdokuments eingeben, muss der Dateiname des Fehlerdokuments ebenfalls `404.html` lauten.
3. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
4. Wählen Sie in der Liste Buckets den Namen des Buckets aus, den Sie zum Hosten einer statischen Website verwenden möchten.
5. Aktivieren Sie das Hosting statischer Websites für Ihren Bucket und geben Sie den exakten Namen Ihres Fehlerdokuments ein (z. B, `404.html`). Weitere Informationen erhalten Sie unter Aktivieren des Website-Hostings und Konfigurieren eines benutzerdefinierten Fehlerdokuments.

   Fahren Sie mit Schritt 6 fort, nachdem Sie das Hosting statischer Websites aktiviert haben.
6. Führen Sie einen der folgenden Schritte aus, um das Fehlerdokument in Ihren Bucket hochzuladen:

   - Ziehen Sie die Fehlerdokumentdatei in das Konsolen-Bucket-Verzeichnis.
   - Wählen Sie Upload (Hochladen) und folgen Sie den Anweisungen zur Auswahl und zum Hochladen der Indexdatei.

step-by-stepAnweisungen finden Sie unterObjekte hochladen.

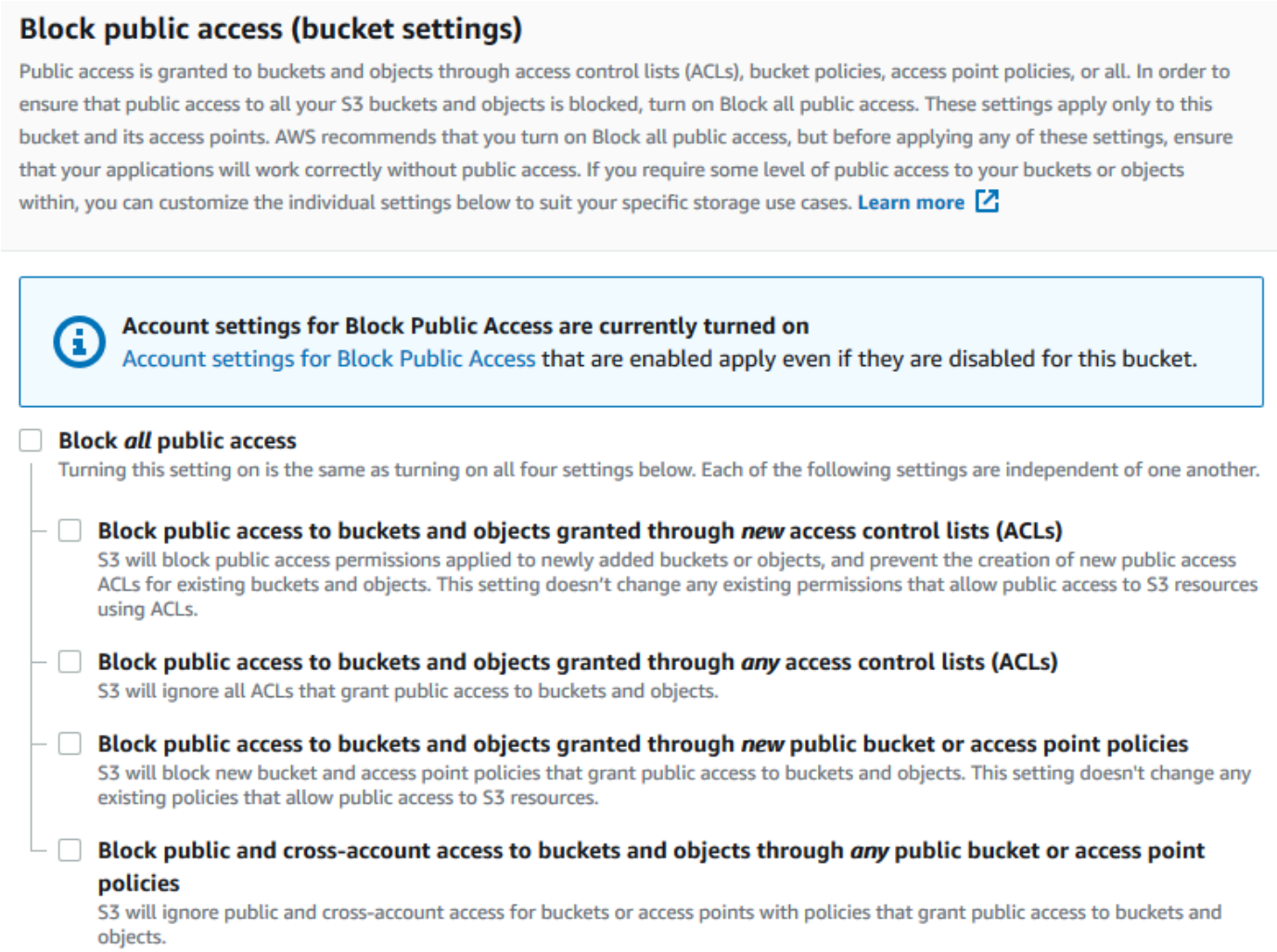
Schritt 8: Bearbeiten der S3 Block Public Access-Einstellungen

In diesem Beispiel bearbeiten Sie die Block-Einstellungen für den öffentlichen Zugriff für den Domänen-Bucket (`example.com`), um den öffentlichen Zugriff zu ermöglichen.

Standardmäßig blockiert Amazon S3 den öffentlichen Zugriff auf Ihr Konto und Ihre Buckets. Wenn Sie einen Bucket verwenden möchten, um eine statische Website zu hosten, können Sie diese Schritte verwenden, um Ihre Einstellungen für Block Public Access zu bearbeiten:

> **Warning**
>
> Bevor Sie diese Schritte ausführen, sollten Sie sicherstellen, Blockieren des öffentlichen Zugriffs auf Ihren Amazon-S3-Speicher dass Sie die Risiken verstehen und akzeptieren, die mit der Gewährung eines öffentlichen Zugangs verbunden sind. Wenn Sie die Einstellungen für Block Public Access deaktivieren, um Ihren Bucket öffentlich zu machen, kann jeder im Internet auf Ihren Bucket zugreifen. Wir empfehlen Ihnen, den gesamten öffentlichen Zugriff auf Ihre Buckets zu blockieren.

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie den Namen des Buckets aus, den Sie als statische Website konfiguriert haben.
3. Wählen Sie Permissions (Berechtigungen).
4. Wählen Sie unter Block public access (bucket settings) (Öffentlichen Zugriff blockieren (Bucket-Einstellungen)), die Option Edit (Bearbeiten).
5. Löschen Sie Block all public access (Gesamten öffentlichen Zugriff blockieren) und wählen Sie Save (Speichern).



Amazon S3 deaktiviert die Einstellungen für den Block Public Access für Ihren Bucket. Um eine öffentliche statische Website zu erstellen, müssen Sie möglicherweise auch die Einstellungen zum Blockieren des öffentlichen Zugriffs für Ihr Konto bearbeiten, bevor Sie eine Bucket-Richtlinie hinzufügen. Wenn die Einstellungen „Öffentlichen Zugriff blockieren“ für Ihr Konto derzeit aktiviert sind, wird unter „Öffentlichen Zugriff blockieren“ (Bucket-Einstellungen) ein Hinweis angezeigt.

Schritt 9: Anfügen einer Bucket-Richtlinie

In diesem Beispiel hängen Sie eine Bucket-Richtlinie an den Domain-Bucket (`example.com`) an, um öffentlichen Lesezugriff zu ermöglichen. Sie ersetzen den *`Bucket-Name`* in der Beispiel-Bucket-Richtlinie zum Beispiel mit dem Namen Ihres Domain-Bucket`example.com`.

Nachdem Sie die Einstellungen für S3 Block Public Access bearbeitet haben, können Sie eine Bucket-Richtlinie hinzufügen, um öffentlichen Lesezugriff auf den Bucket zu gewähren. Wenn Sie öffentlichen Lesezugriff gewähren, kann jeder im Internet auf Ihren Bucket zugreifen.

> **Important**
>
> Die zuvor genannte Richtlinie ist nur ein Beispiel und erlaubt Vollzugriff auf die Inhalte Ihres Buckets. Bevor Sie mit diesem Schritt fortfahren, lesen Sie den Abschnitt Wie kann ich die Dateien in meinem Amazon-S3-Bucket sichern?, um sicherzustellen, dass Sie die bewährten Methoden zum Sichern der Dateien in Ihrem S3-Bucket und die Risiken in Zusammenhang mit der Gewährung von öffentlichem Zugriff kennen.

1. Wählen Sie unter Buckets den Namen Ihres Buckets aus.
2. Wählen Sie Permissions (Berechtigungen).
3. Wählen Sie unter Bucket Policy (Bucket-Richtlinie) Edit (Bearbeiten).
4. Um öffentlichen Lesezugriff auf Ihre Website zu gewähren, kopieren Sie die folgende Bucket-Richtlinie und fügen Sie sie in den Bucket policy editor (Bucket-Richtlinieneditor) ein.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::Bucket-Name/*"
            ]
        }
    ]
}
```

5. Aktualisieren Sie den `Resource` zu Ihrem Bucket-Namen.

Im vorherigen Beispiel für eine Bucket-Richtlinie *Bucket-Name* ist ein Platzhalter für den Bucket-Namen. Um diese Bucket-Richtlinie mit Ihrem eigenen Bucket zu verwenden, müssen Sie diesen Namen so aktualisieren, dass er mit Ihrem Bucket übereinstimmt.

6. Wählen Sie Save Changes (Änderungen speichern).

   Es wird eine Meldung angezeigt, die darauf hinweist, dass die Bucket-Richtlinie erfolgreich hinzugefügt wurde.

   Wenn die Fehlermeldung `Policy has invalid resource` angezeigt wird, bestätigen Sie, dass der Bucket-Name in der Bucket-Richtlinie mit Ihrem Bucket-Namen übereinstimmt. Informationen zum Hinzufügen einer Bucket-Richtlinie finden Sie unter Wie füge ich eine S3-Bucket-Richtlinie hinzu?

   Wenn Sie eine Fehlermeldung erhalten und die Bucket-Richtlinie nicht speichern können, überprüfen Sie Ihr Konto und die Bucket-Einstellungen für Block Public Access, um zu bestätigen, dass Sie den öffentlichen Zugriff auf den Bucket zulassen.

Im nächsten Schritt können Sie die Endpunkte Ihrer Website herausfinden und Ihren Domänen-Endpunkt testen.

Schritt 10: Testen Ihres Domänen-Endpunkts

Nachdem Sie den Stammdomänen-Bucket zum Hosten einer öffentlichen Website konfiguriert haben, können Sie Ihren Endpunkt testen. Weitere Informationen finden Sie unter Website-Endpunkte. Sie können nur den Endpunkt für Ihren Domänen-Bucket testen, da Ihr Subdomänen-Bucket für die Website-Umleitung und nicht für das statische Website-Hosting eingerichtet ist.

> **Note**
>
> Amazon S3 unterstützt keinen HTTPS Zugriff auf die Website. Wenn Sie verwenden möchtenHTTPS, können Sie Amazon verwenden, CloudFront um eine statische Website bereitzustellen, die auf Amazon S3 gehostet wird.
> Weitere Informationen finden Sie unter Wie verwende ich, CloudFront um eine statische Website bereitzustellen, die auf Amazon S3 gehostet wird? und Erforderlich HTTPS für die Kommunikation zwischen Zuschauern und CloudFront.

1. Wählen Sie unter Buckets den Namen Ihres Buckets aus.

2. Wählen Sie Properties (Eigenschaften).
3. Wählen Sie unten auf der Seite unter Static website hosting (Hosting statischer Websites) Ihren Bucket-Website-Endpunkt.

   Ihr Indexdokument wird in einem separaten Browserfenster geöffnet.

Im nächsten Schritt verwenden Sie Amazon Route 53, damit Kunden Ihre beiden benutzerdefinierten Optionen verwenden könnenURLs, um zu Ihrer Website zu navigieren.

Schritt 11: Hinzufügen von Aliasdatensätzen für Ihre Domäne und Subdomäne

In diesem Schritt erstellen Sie die Aliasdatensätze, die Sie der gehosteten Zone für Ihre Domänenzuordnungen `example.com` und `www.example.com` hinzufügen. Statt IP-Adressen verwenden die Alias-Datensätze Amazon S3 Website-Endpunkte. Amazon Route 53 verwaltet ein Mapping zwischen den Alias-Datensätzen und den IP-Adressen, wo sich die Amazon-S3-Buckets befinden. Sie erstellen zwei Alias-Datensätze: einen für Ihre Stamm-Domäne und einen für Ihre Subdomäne.

Hinzufügen eines Aliasdatensatzes für Ihre Stamm-Domäne und Subdomäne

So fügen Sie Ihrer Stamm-Domäne einen Aliasdatensatz hinzu (**`example.com`**)

1. Öffnen Sie die Route 53-Konsole unter https://console.aws.amazon.com/route53/.

   > **Note**
   >
   > Wenn Sie Route 53 noch nicht verwenden, lesen Sie Schritt 1: Registrieren einer Domäne im Amazon-Route 53-Entwicklerhandbuch. Nachdem Sie Ihre Einrichtung abgeschlossen haben, können Sie mit den Anweisungen fortfahren.

2. Wählen Sie Hosted Zones (Gehostete Zonen) aus.
3. Wählen Sie in der Liste der gehosteten Zonen den Namen der gehosteten Zone aus, der dem Domänennamen entspricht.
4. Wählen Sie Create Record Set (Datensatz erstellen).
5. Wählen Sie Switch to wizard (Zu Assistent wechseln) aus.

> **Note**
>
> Wenn Sie Quick Create zum Erstellen Ihrer Aliaseinträge verwenden möchten, lesen Sie Konfigurieren von Route 53 zum Weiterleiten von Datenverkehr an einen S3-Bucket.

6. Wählen Sie Simple Routing (Einfaches Routing), und wählen Sie Next (Weiter).
7. Wählen Sie Define simple record (Einfachen Datensatz definieren).
8. Akzeptieren Sie unter Record name (Datensatzname) den Standardwert, bei dem es sich um den Namen Ihrer gehosteten Zone und Ihrer Domäne handelt.
9. Wählen Sie unter Value/Route traffic to (Wert/Datenverkehr weiterleiten zu) die Option Alias to S3 website endpoint (Alias zu S3-Website-Endpunkt) aus.
10. Wählen Sie die Region aus.
11. Wählen Sie den S3-Bucket.

    Der Bucket-Name sollte mit dem Namen übereinstimmen, der im Feld Name angezeigt wird. In der Liste Choose S3 Bucket (S3-Bucket auswählen) wird der Bucket-Name mit dem Amazon-S3-Website-Endpunkt für die Region angezeigt, in der der Bucket erstellt wurde, zum Beispiel `s3-website-us-west-1.amazonaws.com (example.com)`.

    Choose S3 bucket (S3-Bucket auswählen) listet einen Bucket auf, wenn:

    - Sie den Bucket als statische Website konfiguriert haben.
    - Der Name des Buckets mit dem Namen des Datensatzes übereinstimmt, den Sie anlegen.
    - Der aktuelle AWS-Konto hat den Bucket erstellt.

    Wenn Ihr Bucket nicht in der Auflistung Choose S3 bucket (S3-Bucket auswählen) angezeigt wird, geben Sie den Amazon-S3-Website-Endpunkt für die Region ein, in der der Bucket erstellt wurde, z. B. **`s3-website-us-west-2.amazonaws.com`**. Eine vollständige Liste der Amazon-S3-Website-Endpunkte finden Sie unter Amazon-S3-Website-Endpunkte. Weitere Informationen über das Alias-Target finden Sie unter Wert/Datenverkehr weiterleiten zu im Amazon-Route 53-Entwicklerhandbuch.
12. Wählen Sie unter Datensatztyp die Option A - Leitet Traffic an eine IPv4 Adresse und einige AWS Ressourcen weiter.
13. Wählen Sie unter Evaluate target health (Zielzustand bewerten) die Option No (Nein).
14. Wählen Sie Define simple record (Einfachen Datensatz definieren).

So fügen Sie Ihrer Subdomäne () einen Alias-Datensatz hi (**`www.example.com`**)

1. Wählen Sie unter Configure records (Datensätze konfigurieren) die Option Define simple record (Einfachen Datensatz definieren) aus
2. Geben Sie unter Record name (Datensatzname) für Ihre Subdomäne `www` ein.
3. Wählen Sie unter Value/Route traffic to (Wert/Datenverkehr weiterleiten zu) die Option Alias to S3 website endpoint (Alias zu S3-Website-Endpunkt) aus.
4. Wählen Sie die Region aus.
5. Wählen Sie den S3-Bucket, zum Beispiel, `s3-website-us-west-2.amazonaws.com (www.example.com)`.

   Wenn Ihr Bucket nicht in der Auflistung Choose S3 bucket (S3-Bucket auswählen) angezeigt wird, geben Sie den Amazon-S3-Website-Endpunkt für die Region ein, in der der Bucket erstellt wurde, z. B. **`s3-website-us-west-2.amazonaws.com`**. Eine vollständige Liste der Amazon-S3-Website-Endpunkte finden Sie unter Amazon-S3-Website-Endpunkte. Weitere Informationen über das Alias-Target finden Sie unter Wert/Datenverkehr weiterleiten zu im Amazon-Route 53-Entwicklerhandbuch.
6. Wählen Sie unter Datensatztyp die Option A - Leitet den Verkehr an eine IPv4 Adresse und einige AWS Ressourcen weiter.
7. Wählen Sie unter Evaluate target health (Zielzustand bewerten) die Option No (Nein).
8. Wählen Sie Define simple record (Einfachen Datensatz definieren).
9. Klicken Sie auf der Seite Configure records (Datensätze konfigurieren) auf Create records (Datensätze erstellen).

> **Note**
>
> Änderungen werden im Allgemeinen innerhalb von 60 Sekunden an alle Route 53-Server übertragen. Wenn die Übertragung abgeschlossen ist, können Sie den Datenverkehr an den Amazon-S3-Bucket leiten, indem Sie den Namen des Alias-Datensatzes angeben, den Sie in diesem Verfahren erstellt haben.

Hinzufügen eines Alias-Datensatzes für Ihre Stamm-Domäne und Subdomäne (alte Route 53-Konsole)

So fügen Sie Ihrer Stamm-Domäne einen Aliasdatensatz hinzu (**`example.com`**)

Die Route 53-Konsole wurde überarbeitet. In der Route 53-Konsole können Sie vorübergehend die alte Konsole verwenden. Wenn Sie sich für die Arbeit mit der alten Route 53-Konsole entscheiden, gehen Sie wie folgt vor.

1. Öffnen Sie die Route 53-Konsole unter https://console.aws.amazon.com/route53/.

   > **Note**
   >
   > Wenn Sie Route 53 noch nicht verwenden, lesen Sie Schritt 1: Registrieren einer Domäne im Amazon-Route 53-Entwicklerhandbuch. Nachdem Sie Ihre Einrichtung abgeschlossen haben, können Sie mit den Anweisungen fortfahren.

2. Wählen Sie Hosted Zones (Gehostete Zonen) aus.
3. Wählen Sie in der Liste der gehosteten Zonen den Namen der gehosteten Zone aus, der dem Domänennamen entspricht.
4. Wählen Sie Create Record Set (Datensatz erstellen).
5. Geben Sie die folgenden Werte an:

   Name

   Akzeptieren Sie den Standardwert, bei dem es sich um den Namen Ihrer gehosteten Zone und Ihrer Domäne handelt.

   Für die Stamm-Domäne müssen Sie keine zusätzlichen Informationen in das Feld Name eingeben.

   Typ

   Wählen Sie A — IPv4 Adresse.

   Alias

   Wählen Sie Yes (Ja).

Alias-Ziel

Wählen Sie im Bereich S3 website endpoints (S3-Website-Endpunkte) der Liste Ihren Bucket-Namen aus.

Der Bucket-Name sollte mit dem Namen übereinstimmen, der im Feld Name angezeigt wird. In der Auflistung Alias Target (Alias-Ziel) folgt dem Bucket-Namen beispielsweise der Amazon-S3-Website-Endpunkt für die Region, in der der Bucket erstellt wurde, z. B. `example.com (s3-website-us-west-2.amazonaws.com)`. Alias Target (Alias-Ziel) listet einen Bucket auf, wenn:

- Sie den Bucket als statische Website konfiguriert haben.
- Der Name des Buckets mit dem Namen des Datensatzes übereinstimmt, den Sie anlegen.
- Der aktuelle AWS-Konto hat den Bucket erstellt.

Wenn Ihr Bucket nicht in der Auflistung Alias Target (Alias-Ziel) angezeigt wird, geben Sie den Amazon-S3-Website-Endpunkt für die Region ein, in der der Bucket erstellt wurde, z. B. `s3-website-us-west-2`. Eine vollständige Liste der Amazon-S3-Website-Endpunkte finden Sie unter Amazon-S3-Website-Endpunkte. Weitere Informationen über das Alias-Target finden Sie unter Value/Route Traffic to (Wert/Datenverkehr weiterleiten zu) im Amazon-Route 53-Entwicklerhandbuch.

Routing-Richtlinie

Übernehmen Sie den Standardwert Simple.

Evaluate Target Health

Übernehmen Sie den Standardwert No.

6. Wählen Sie Create (Erstellen) aus.

So fügen Sie Ihrer Subdomäne () einen Alias-Datensatz hi (**`www.example.com`**)

1. Wählen Sie in der gehosteten Zone für Ihre Stamm-Domäne (`example.com`) die Option Create record set (Datensatz erstellen) aus.
2. Geben Sie die folgenden Werte an:

Name

Geben Sie als Subdomäne „www“ in das Feld ein.

Typ

Wählen Sie A — IPv4 Adresse.

Alias

Wählen Sie Yes (Ja).

Alias-Ziel

Wählen Sie im Abschnitt S3 website endpoints (S3-Website-Endpunkte) der Liste denselben Bucket-Namen aus, der im Feld Name angezeigt wird (z. B. `www.example.com (s3-website-us-west-2.amazonaws.com)`).

Routing-Richtlinie

Übernehmen Sie den Standardwert Simple.

Evaluate Target Health

Übernehmen Sie den Standardwert No.

3. Wählen Sie Create aus.

> **Note**
>
> Änderungen werden im Allgemeinen innerhalb von 60 Sekunden an alle Route 53-Server übertragen. Wenn die Übertragung abgeschlossen ist, können Sie den Datenverkehr an den Amazon-S3-Bucket leiten, indem Sie den Namen des Alias-Datensatzes angeben, den Sie in diesem Verfahren erstellt haben.

Schritt 12: Testen der Website

Bestätigen Sie, dass die Website und die Umleitung korrekt funktionieren. Geben Sie in Ihrem Browser Ihre einURLs. In diesem Beispiel können Sie Folgendes versuchenURLs:

- Domain (Domäne) (`http://example.com`) – Zeigt das Indexdokument im `example.com`-Bucket an.
- Subdomain (Subdomäne) (`http://www.example.com`) – Leitet Ihre Anforderung an `http://example.com` weiter. Sie sehen das Indexdokument im `example.com`-Bucket.

Wenn Ihre Website oder Weiterleitungslinks nicht funktionieren, können Sie Folgendes versuchen:

- Clear cache (Cache löschen) – Löschen Sie den Cache Ihres Webbrowsers.
- Check name servers (Nameserver überprüfen) – Wenn Ihre Webseite und Weiterleitungslinks nach dem Löschen des Caches nicht funktionieren, können Sie die Namenserver für Ihre Domäne und die Namenserver für Ihre gehostete Zone vergleichen. Wenn die Namenserver nicht übereinstimmen, müssen Sie möglicherweise Ihre Domänennamenserver so aktualisieren, dass sie mit den unter Ihrer gehosteten Zone aufgelisteten übereinstimmen. Weitere Informationen finden Sie unter Hinzufügen oder Ändern von Namenservern und Kleben von Datensätzen für eine Domäne.

Nachdem Sie Ihre Stammdomain und Subdomain erfolgreich getestet haben, können Sie eine CloudFrontAmazon-Distribution einrichten, um die Leistung Ihrer Website zu verbessern und Protokolle bereitzustellen, anhand derer Sie den Website-Traffic überprüfen können. Weitere Informationen finden Sie unter Beschleunigen Sie Ihre Website mit Amazon CloudFront.

Beschleunigen Sie Ihre Website mit Amazon CloudFront

Sie können Amazon verwenden CloudFront, um die Leistung Ihrer Amazon S3 S3-Website zu verbessern. CloudFront stellt Ihre Website-Dateien (wie HTML Bilder und Videos) in Rechenzentren auf der ganzen Welt (sogenannte Edge-Standorte) zur Verfügung. Wenn ein Besucher eine Datei von Ihrer Website anfordert, leitet er die Anfrage CloudFront automatisch an eine Kopie der Datei am nächstgelegenen Edge-Standort weiter. Das führt zu schnelleren Download-Zeiten, als wenn der Besucher den Inhalt von einem weitere entfernteren Rechenzentrum angefordert hätte.

CloudFront speichert Inhalte an Edge-Standorten für einen von Ihnen angegebenen Zeitraum im Cache. Wenn ein Besucher Inhalte anfordert, die länger als das Ablaufdatum zwischengespeichert wurden, CloudFront überprüft er auf dem Ursprungsserver, ob eine neuere Version des Inhalts verfügbar ist. Wenn eine neuere Version verfügbar ist, wird die neue Version an den Edge-Standort CloudFront kopiert. Änderungen an den ursprünglichen Inhalten werden zu den Edge-Standorten repliziert, indem die Besucher die Inhalte anfordern.

Verwendung CloudFront ohne Route 53

Das Tutorial auf dieser Seite verwendet Route 53, um auf Ihre CloudFront Verteilung hinzuweisen. Wenn Sie jedoch Inhalte bereitstellen möchten, die in einem Amazon S3-Bucket gehostet werden, CloudFront ohne Route 53 zu verwenden, finden Sie weitere Informationen unter CloudFrontAmazon-Tutorials: Einrichtung einer dynamischen Inhaltsverteilung für Amazon S3. Wenn

Sie Inhalte bereitstellen, die in einem Amazon S3 S3-Bucket gehostet werden CloudFront, können Sie einen beliebigen Bucket-Namen verwenden, HTTP und beide HTTPS werden unterstützt.

Automatisieren der Einrichtung mit einer Vorlage AWS CloudFormation

Weitere Informationen zur Verwendung einer AWS CloudFormation Vorlage zur Konfiguration einer sicheren statischen Website, die eine CloudFront Distribution für Ihre Website erstellt, finden Sie unter Erste Schritte mit einer sicheren statischen Website im Amazon CloudFront Developer Guide.

Themen

- Schritt 1: Erstellen Sie eine CloudFront Distribution
- Schritt 2: Aktualisieren der Datensätze für Ihre Domäne und Unterdomäne
- (Optional) Schritt 3: Überprüfen der Protokolldateien

Schritt 1: Erstellen Sie eine CloudFront Distribution

Zunächst erstellen Sie eine CloudFront Distribution. Damit steht Ihre Website in weltweit angesiedelten Rechenzentren zur Verfügung.

So erstellen Sie eine Verteilung mit einem Amazon S3-Ursprung

1. Öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Wählen Sie Create Distribution (Distribution erstellen).
3. Geben Sie auf der Seite Create Distribution (Verteilung erstellen) im Abschnitt Origin Settings (Ursprungseinstellungen) unter Origin Domain Name (Ursprungsdomänenname) den Amazon S3-Website-Endpunkt für Ihren Bucket ein – z. B. **`example.com.s3-website.us-west-1.amazonaws.com`**.

   CloudFront füllt die Origin-ID für dich aus.
4. Behalten Sie für Default Cache Behavior Settings (Standardeinstellungen für das Zwischenspeicherverhalten) die Standardwerte bei.

   Mit den Standardeinstellungen für die Viewer-Protokollrichtlinie kannst du sie HTTPS für deine statische Website verwenden. Weitere Informationen zu diesen Konfigurationsoptionen finden Sie unter Werte, die Sie beim Erstellen oder Aktualisieren einer Web-Distribution angeben im Amazon CloudFront Developer Guide.
5. Gehen Sie unter Distribution Settings (Verteilungseinstellungen) wie folgt vor:

a. Lassen Sie für die Option Price Class (Preisklasse) die Einstellung Use All Edge Locations (Best Performance) (Alle Edge-Standorte verwenden (Beste Leistung)) unverändert.

b. Legen Sie alternative Domainnamen (CNAMEs) für die Root-Domain und die `www` Subdomain fest. In diesem Tutorial sind dies `example.com` und `www.example.com`.

> **Important**
>
> Bevor Sie diesen Schritt ausführen, beachten Sie die Anforderungen für die Verwendung alternativer Domainnamen, insbesondere die Notwendigkeit eines gültigen SSL TLS /-Zertifikats.

c. Wählen Sie für SSLZertifikat die Option Benutzerdefiniertes SSL Zertifikat (example.com) und wählen Sie das benutzerdefinierte Zertifikat aus, das die Domain- und Subdomainnamen abdeckt.

   Weitere Informationen finden Sie unter SSLZertifikat im Amazon CloudFront Developer Guide.

d. Geben Sie in Default Root Object (Standard-Root-Objekt) den Namen Ihres Indexdokuments ein, z. B. `index.html`.

   Wenn die für den Zugriff auf die Distribution URL verwendete Datei keinen Dateinamen enthält, gibt die CloudFront Distribution das Indexdokument zurück. Das Default Root Object (Standard-Root-Objekt) sollte genau mit dem Namen des Indexdokuments für Ihre statische Website übereinstimmen. Weitere Informationen finden Sie unter Konfigurieren eines Indexdokuments.

e. Wählen Sie für Logging (Protokollierung) die Option On (Ein).

> **Important**
>
> Wenn Sie eine Distribution erstellen oder aktualisieren und die CloudFront Protokollierung aktivieren, CloudFront aktualisiert die Bucket-Zugriffskontrollliste (ACL), sodass das `awslogsdelivery` Konto `FULL_CONTROL` berechtigt ist, Logs in Ihren Bucket zu schreiben. Weitere Informationen finden Sie unter Erforderliche Berechtigungen für die Konfiguration der Standardprotokollierung und für den Zugriff auf Ihre Protokolldateien im Amazon CloudFront Developer Guide. Wenn der Bucket, der die Logs speichert, die vom Bucket-Besitzer erzwungene Einstellung zur Deaktivierung von S3 Object Ownership verwendetACLs, CloudFront können keine

Logs in den Bucket geschrieben werden. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.

f. Wählen Sie als Bucket for Logs (Bucket für Protokolle) den Bucket zur Protokollierung aus, den Sie erstellt haben.

   Weitere Informationen zum Konfigurieren eines Protokoll-Buckets finden Sie unter (Optional) Protokollieren des Webdatenverkehrs.

g. Wenn Sie die Protokolle, die durch den Datenverkehr zur CloudFront Distribution generiert werden, in einem Ordner speichern möchten, geben Sie im Feld Protokollpräfix den Ordnernamen ein.

h. Behalten Sie für alle übrigen Einstellungen die Standardwerte bei.

6. Wählen Sie Create Distribution.
7. Um den aktuellen Status der Verteilung anzuzeigen, suchen Sie die Verteilung in der Konsole, und prüfen Sie die Spalte Status.

   Der Status `InProgress` gibt an, dass die Verteilung noch nicht vollständig bereitgestellt ist.

   Nach der Bereitstellung Ihrer Distribution können Sie mit dem neuen CloudFront Domainnamen auf Ihre Inhalte verweisen.
8. Notieren Sie sich den Wert des Domainnamens, der in der CloudFront Konsole angezeigt wird, `dj4p1rv6mvubz.cloudfront.net` z. B.
9. Um zu überprüfen, ob Ihre CloudFront Distribution funktioniert, geben Sie den Domainnamen der Distribution in einen Webbrowser ein.

   Wenn Ihre Website sichtbar ist, funktioniert die CloudFront Verteilung. Wenn auf Ihrer Website eine benutzerdefinierte Domain bei Amazon Route 53 registriert ist, benötigen Sie den CloudFront Domainnamen, um den Datensatz im nächsten Schritt zu aktualisieren.

Schritt 2: Aktualisieren der Datensätze für Ihre Domäne und Unterdomäne

Nachdem Sie nun erfolgreich eine CloudFront Verteilung erstellt haben, aktualisieren Sie den Aliaseintrag in Route 53 so, dass er auf die neue CloudFront Verteilung verweist.

Um den Aliaseintrag so zu aktualisieren, dass er auf eine CloudFront Verteilung verweist

1. Öffnen Sie die Route 53-Konsole unter https://console.aws.amazon.com/route53/.

2. Wählen Sie in der linken Navigation Hosted Zones (Gehostete Zonen).
3. Wählen Sie auf der Seite Hosted Zones (Gehostete Zonen) die gehostete Zone aus, die Sie für Ihre Unterdomäne erstellt haben, beispielsweise `www.example.com`.
4. Wählen Sie unter Datensätzeden A-Eintrag aus, den Sie für Ihre Unterdomäne erstellt haben.
5. Wählen Sie unter Datensatzdetailsden Befehl Datensatz bearbeiten.
6. Wählen Sie unter Traffic weiterleiten an die Option Alias to CloudFront Distribution aus.
7. Wählen Sie unter Verteilung auswählen die CloudFront Verteilung aus.
8. Wählen Sie Save (Speichern) aus.
9. Um den A-Eintrag für die Stammdomäne zur CloudFront Distribution umzuleiten, wiederholen Sie dieses Verfahren für die Stammdomäne, `example.com` z. B.

   Die Aktualisierung der Datensätze wird innerhalb von 2 bis 48 Stunden wirksam.
10. Um zu sehen, ob die neuen A-Einträge wirksam geworden sind, geben Sie in einem Webbrowser Ihre Subdomain einURL, z. B. `http://www.example.com`

    Wenn der Browser Sie nicht mehr zur Stammdomäne umleitet (z. B. `http://example.com`), sind die neuen A-Datensätze vorhanden. Wenn der neue A-Eintrag wirksam geworden ist, wird der Datenverkehr, der vom neuen A-Eintrag an die CloudFront Distribution weitergeleitet wird, nicht an die Stammdomain umgeleitet. Alle Besucher, die über den nächstgelegenen Standort auf die Website verweisen `http://example.com` oder zu CloudFront diesem weitergeleitet `http://www.example.com` werden, profitieren von schnelleren Download-Zeiten.

    > **Tip**
    >
    > Umleitungseinstellungen können von Browsern zwischengespeichert werden. Wenn Sie annehmen, dass die neuen A-Datensatzeinstellungen wirksam sind, aber trotzdem eine Umleitung von `http://www.example.com` nach `http://example.com` feststellen, löschen Sie zum Testen den Verlauf und den Cache des Browsers. Schließen Sie den Browser und öffnen Sie ihn wieder oder verwenden Sie einen anderen Webbrowser, wenn Sie einen weiteren installiert haben.

(Optional) Schritt 3: Überprüfen der Protokolldateien

Die Zugriffsprotokolle teilen Ihnen mit, wie viele Menschen die Website besuchen. Sie enthalten auch wertvolle Geschäftsdaten, die Sie mit anderen Diensten wie Amazon analysieren könnenEMR.

CloudFront Protokolle werden in dem Bucket und Ordner gespeichert, den Sie auswählen, wenn Sie eine CloudFront Distribution erstellen und die Protokollierung aktivieren. CloudFront schreibt innerhalb von 24 Stunden, nachdem die entsprechenden Anfragen gestellt wurden, Logs in Ihren Log-Bucket.

Die Protokolldateien für Ihre Website anzeigen

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie den Protokoll-Bucket für Ihre Website aus.
3. Wählen Sie den CloudFront Logs-Ordner aus.
4. Laden Sie die von geschriebenen `.gzip` Dateien herunter, CloudFront bevor Sie sie öffnen.

   Wenn Sie die Website nur zur Übung erstellt haben, können Sie die von Ihnen zugewiesenen Ressourcen löschen, damit keine weiteren Kosten für Sie anfallen. Lesen Sie dazu den Abschnitt Bereinigung Ihrer Beispielressourcen. Nach der Löschung der AWS -Ressourcen ist die Website nicht mehr verfügbar.

Bereinigung Ihrer Beispielressourcen

Wenn Sie die statische Website zur Übung erstellt haben, sollten Sie die zugewiesenen AWS -Ressourcen löschen, damit keine weiteren Kosten für Sie anfallen. Nach der Löschung der AWS -Ressourcen ist die Website nicht mehr verfügbar.

Aufgaben

- Schritt 1: Löschen Sie die CloudFront Amazon-Distribution
- Schritt 2: Löschen der von Route 53 gehosteten Zone
- Schritt 3: Deaktivieren der Protokollierung und Löschen Ihres S3-Buckets

Schritt 1: Löschen Sie die CloudFront Amazon-Distribution

Bevor Sie eine CloudFront Amazon-Distribution löschen, müssen Sie sie deaktivieren. Eine deaktivierte Verteilung funktioniert nicht mehr und es fallen keine weiteren Kosten für sie an. Sie können eine deaktivierte Verteilung jederzeit wieder aktivieren. Nachdem Sie eine deaktivierte Verteilung gelöscht haben, ist sie nicht länger verfügbar.

Um eine CloudFront Distribution zu deaktivieren und zu löschen

1. Öffnen Sie die CloudFront Konsole unterhttps://console.aws.amazon.com/cloudfront/v4/home.
2. Klicken Sie mit der rechten Maustaste auf die Verteilung, die Sie deaktivieren möchten, und anschließend auf Disable (Deaktivieren).
3. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Yes, Disable (Ja, deaktivieren).
4. Wählen Sie die deaktivierte Verteilung aus, und klicken Sie dann auf Delete (Löschen).
5. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Yes, Delete.

Schritt 2: Löschen der von Route 53 gehosteten Zone

Bevor Sie eine gehostete Zone löschen können, müssen Sie die zuvor erstellten Datensätze löschen. Sie müssen den NS und die SOA Einträge nicht löschen. Diese werden automatisch gelöscht, wenn Sie die gehostete Zone löschen.

Sie löschen die Datensätze wie folgt:

1. Öffnen Sie die Route 53-Konsole unter https://console.aws.amazon.com/route53/.
2. Wählen Sie in der Liste der Domänennamen Ihren Domänennamenaus, und wählen Sie anschließend Go to Record Sets (Zu den Datensätzen).
3. Wählen Sie in der Liste der Datensatzsätze die A-Datensätze aus, die Sie erstellt haben.

   Der einzelne Datensatztyp wird in der Spalte Type (Typ) aufgeführt.
4. Wählen Sie Delete Record Set (Datensatz löschen).
5. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Confirm (Bestätigen).

So löschen Sie eine von Route 53 gehostete Zone:

1. Setzen Sie das vorherige Verfahren fort, indem Sie Back to Hosted Zones (Zurück zu gehosteten Zonen) auswählen.
2. Wählen Sie Ihren Domänennamen und dann Delete Hosted Zone (Gehostete Zone löschen).
3. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Confirm (Bestätigen).

Schritt 3: Deaktivieren der Protokollierung und Löschen Ihres S3-Buckets

Bevor Sie Ihren S3-Bucket löschen, stellen Sie sicher, dass die Protokollierung für den Bucket deaktiviert ist. Andernfalls werden AWS weiterhin Protokolle in Ihren Bucket geschrieben, während Sie ihn löschen.

Deaktivieren Sie die Protokollierung für einen Bucket wie folgt:

1. Öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie unter Buckets (Buckets) den Namen Ihres Buckets aus. Wählen Sie anschließend Properties (Eigenschaften) aus.
3. Wählen Sie unter Properties (Eigenschaften) Logging (Protokollierung) aus.
4. Deaktivieren Sie das Kontrollkästchen Enabled (Aktiviert).
5. Wählen Sie Save (Speichern) aus.

Jetzt können Sie Ihren Bucket löschen. Weitere Informationen finden Sie unter Löschen eines Buckets.

Bereitstellung einer statischen Website für AWS Amplify Hosting aus einem S3-Allzweck-Bucket

Wir empfehlen Ihnen, AWS Amplify Hosting zu verwenden, um statische Website-Inhalte zu hosten, die auf S3 gespeichert sind. Amplify Hosting ist ein vollständig verwalteter Service, mit dem Sie Ihre Websites auf einfache Weise in einem weltweit verfügbaren Content Delivery Network (CDN) bereitstellen können CloudFront, das von Amazon betrieben wird und sicheres statisches Website-Hosting ohne umfangreiche Einrichtung ermöglicht. Mit AWS Amplify Hosting können Sie den Speicherort Ihrer Objekte in Ihrem Allzweck-Bucket auswählen, Ihre Inhalte in einem verwalteten Bereich bereitstellen und eine Öffentlichkeit HTTPS URL generierenCDN, damit Ihre Website von überall aus zugänglich ist. Die Bereitstellung einer statischen Website mit Amplify Hosting bietet Ihnen die folgenden Vorteile und Funktionen:

- Bereitstellung im AWS Content Delivery Network (CDN) von Amazon CloudFront — CloudFront ist ein Webservice, der die Verteilung Ihrer statischen und dynamischen Webinhalte an Ihre Nutzer beschleunigt. CloudFront stellt Ihre Inhalte über ein weltweites Netzwerk von Rechenzentren bereit, die als Edge-Standorte bezeichnet werden. Wenn ein Benutzer Inhalte anfordert, die Sie bereitstellen CloudFront, wird die Anfrage an den Edge-Standort weitergeleitet, der die niedrigste Latenz (Zeitverzögerung) bietet, sodass Inhalte mit der bestmöglichen Leistung, erhöhter

Zuverlässigkeit und Verfügbarkeit bereitgestellt werden. Weitere Informationen finden Sie unter How CloudFront Delivers Content im Amazon CloudFront Developer Guide.

- HTTPSSupport — Ermöglicht eine sichere Kommunikation und Datenübertragung zwischen Ihrer Website und dem Webbrowser eines Benutzers.
- Benutzerdefinierte Domains — Verbinden Sie Ihre Website ganz einfach mit einer benutzerdefinierten Domain, die Sie bei einem Domain-Registrar wie Amazon Route 53 URL gekauft haben.
- Benutzerdefinierte SSL Zertifikate — Wenn Sie Ihre benutzerdefinierte Domain einrichten, können Sie das verwaltete Standardzertifikat verwenden, das Amplify für Sie bereitstellt, oder Sie können Ihr eigenes benutzerdefiniertes Zertifikat verwenden, das Sie bei einer Drittanbieter-Zertifizierungsstelle Ihrer Wahl erworben haben.
- Integrierte Metriken und CloudWatch Überwachung — Überwachen Sie Traffic, Fehler, Datenübertragung und Latenz für Ihre Website.
- Passwortschutz — Beschränken Sie den Zugriff auf Ihre Website, indem Sie in der Amplify-Konsole einen Benutzernamen und ein Passwort einrichten.
- Umleitungen und Umschreibungen — Erstellen Sie Umleitungs- und Umschreibungsregeln in der Amplify-Konsole, damit ein Webserver die Navigation von einem zum anderen umleiten kann. URL

Wenn Sie Ihre Anwendung von einem Amazon S3 S3-Allzweck-Bucket auf Amplify Hosting bereitstellen, basieren die AWS Gebühren auf dem Preismodell von Amplify. Weitere Informationen finden Sie unter AWS Amplify -Preisgestaltung.

> **Important**
>
> Amplify Hosting ist nicht überall dort verfügbar, AWS-Regionen wo Amazon S3 verfügbar ist. Um eine statische Website für Amplify Hosting bereitzustellen, muss sich der Amazon S3 S3-Allzweck-Bucket, der Ihre Website enthält, in einer Region befinden, in der Amplify verfügbar ist. Eine Liste der Regionen, in denen Amplify verfügbar ist, finden Sie unter Amplify-Endpunkte in der. Allgemeine Amazon Web Services-Referenz

Sie können den Bereitstellungsprozess von der Amazon S3 S3-Konsole, der Amplify-Konsole AWS CLI, dem oder dem AWS SDKs aus starten. Sie können Amplify nur von einem Allzweck-Bucket aus bereitstellen, der sich in Ihrem eigenen Konto befindet. Amplify unterstützt keinen kontoübergreifenden Bucket-Zugriff.

Verwenden Sie die folgenden Anweisungen, um ausgehend von der Amazon S3-Konsole eine statische Website aus einem Amazon S3-Allzweck-Bucket auf Amplify Hosting bereitzustellen.

Bereitstellen einer statischen Website für Amplify über die S3-Konsole

So stellen Sie eine statische Website über die Amazon S3 S3-Konsole bereit

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon S3 S3-Konsole unter https://console.aws.amazon.com/s3/.
2. Wählen Sie im linken Navigationsbereich Buckets aus.
3. Wählen Sie in der Bucket-Liste den Allzweck-Bucket aus, der die Website enthält, die Sie für Amplify Hosting bereitstellen möchten.
4. Wählen Sie die Registerkarte Eigenschaften aus.
5. Wählen Sie unter Statisches Website-Hosting die Option Amplify-App erstellen aus. In diesem Schritt wird der Bereitstellungsprozess auf die Amplify-Konsole verschoben.
6. Führen Sie auf der Seite Deploy with S3 die folgenden Schritte aus.
   a. Geben Sie als App-Name den Namen Ihrer App oder Website ein.
   b. Geben Sie unter Branchenname den Namen des Backends Ihrer App ein.
   c. Geben Sie für den S3-Speicherort der zu hostenden Objekte entweder den Verzeichnispfad zu Ihrem Allzweck-Bucket ein oder wählen Sie Browse S3, um ihn zu finden und auszuwählen.
7. Wählen Sie Save and deploy (Speichern und Bereitstellen) aus.

> **Note**
>
> Wenn Sie eines der Objekte für eine statische Website in Ihrem auf Amplify gehosteten Allzweck-Bucket aktualisieren, müssen Sie die Anwendung erneut auf Amplify Hosting bereitstellen, damit die Änderungen wirksam werden. Amplify Hosting erkennt Änderungen an Ihrem Bucket nicht automatisch. Weitere Informationen finden Sie unter Aktualisieren einer statischen Website, die aus einem S3-Bucket auf Amplify bereitgestellt wurde, im AWS Amplify Hosting-Benutzerhandbuch.

Um direkt von der Amplify-Konsole aus zu starten, siehe Bereitstellen einer statischen Website von S3 mithilfe der Amplify-Konsole im AWS Amplify Hosting-Benutzerhandbuch.

Informationen zu den ersten Schritten mit dem AWS SDKs finden Sie unter Erstellen einer Bucket-Richtlinie zur Bereitstellung einer statischen Website aus S3 mithilfe von AWS SDKs im AWS Amplify Hosting-Benutzerhandbuch.

Kontingente

Ihr AWS-Konto hat Standardkontingente, früher als Limits bezeichnet, für jeden AWS Service. Amazon S3 S3-Kontingente umfassen die Anzahl von Allzweck-Buckets, Verzeichnis-Buckets, Access Points und mehr. Sie können für einige Kontingente eine Erhöhung beantragen, aber nicht alle Kontingente können erhöht werden. Diese Erhöhungen werden nicht sofort gewährt, es kann einige Tage dauern, bis Ihre Erhöhung wirksam ist.

Eine Liste der Amazon S3 S3-Kontingente und ihrer Standardwerte finden Sie unter Amazon S3 S3-Kontingente in der Allgemeine AWS-Referenz.

Die Quote wird erhöht

So fordern Sie eine Kontingenterhöhung an

Sie können eine Erhöhung des Kontingents beantragen, indem Sie eine der folgenden Optionen verwenden:

- Von AWS Management Console: Öffnen Sie die Service Quotas Quotas-Konsole. Wählen Sie im Navigationsbereich AWS -Services. Wählen Sie Amazon S3 aus, wählen Sie ein Kontingent aus und folgen Sie den Anweisungen, um eine Kontingenterhöhung zu beantragen. Weitere Informationen finden Sie unter Anfordern einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.
- Von AWS CLI: Verwenden Sie den request-service-quota-increase AWS CLI Befehl. Weitere Informationen finden Sie unter Anfordern einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.

Referenz

Zusätzlich zur Amazon S3-Konsole können Sie mithilfe von Amazon S3,, oder AWS Command Line Interface ()AWS CLI programmgesteuert mit Amazon S3 REST APIs arbeiten. AWS SDKs

- APIReferenz zum Amazon Simple Storage Service
- Codebeispiele in der Amazon Simple Storage Service API Reference
- AWS Befehlszeilenschnittstelle im AWS Command Line Interface Benutzerhandbuch.
- AWS SDKs

SDKDokumentation	Codebeispiele
AWS SDK for C++	AWS SDK for C++ Code-Beispiele
AWS CLI	AWS CLI Codebeispiele.
AWS SDK for Go	AWS SDK for Go Code-Beispiele
AWS SDK for Java	AWS SDK for Java Code-Beispiele
AWS SDK for JavaScript	AWS SDK for JavaScript Code-Beispiele
AWS SDK for Kotlin	AWS SDK for Kotlin Code-Beispiele
AWS SDK for .NET	AWS SDK for .NET Code-Beispiele
AWS SDK for PHP	AWS SDK for PHP Code-Beispiele
AWS Tools for PowerShell	Tools für PowerShell Codebeispiele
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) Code-Beispiele
AWS SDK for Ruby	AWS SDK for Ruby Code-Beispiele
AWS SDK for Rust	AWS SDK for Rust Code-Beispiele
AWS SDK für SAP ABAP	AWS SDK für SAP ABAP Code-Beispiele
AWS SDK for Swift	AWS SDK for Swift Code-Beispiele

Dokumentverlauf

- Aktuelle API Version: 2006-03-01

In der folgenden Tabelle werden die wichtigen Änderungen in den einzelnen Versionen der Amazon Simple Storage Service API Reference und des Amazon S3 S3-Benutzerhandbuchs beschrieben. Um über Aktualisierungen dieser Dokumentation informiert zu werden, können Sie einen RSS Feed abonnieren.

Änderung	Beschreibung	Datum
Amazon Redshift ist jetzt in S3 Access Grants integriert	Amazon Redshift Redshift-Kunden können jetzt S3 Access Grants verwenden, um die Berechtigungen für ihre S3-Daten zu skalieren und zu verwalten. Auf diese Weise können Amazon Redshift Redshift-Kunden S3-Berech tigungen für Unternehm ensidentitäten sowohl für IAM Benutzer AWS IAM Identity Center als auch für Gruppen skalieren. Weitere Informati onen finden Sie unter Amazon Redshift Redshift-Integrati on mit Amazon S3 Access Grants.	15. November 2024
AWS Organizations Mitglieds konten können jetzt wieder auf versehentlich gesperrte Amazon S3 S3-Buckets zugreifen	AWS Organizations Mitglieds konten können jetzt mit einem einfachen Verfahren über AWS Identity and Access Management (IAM) wieder Zugriff auf versehentlich gesperrte Amazon S3 S3-	14. November 2024

	Buckets erhalten. Weitere Informationen finden Sie im AWS Identity and Access Management Benutzerh andbuch unter Ausführen einer privilegierten Aufgabe für ein AWS Organizations Mitglieds konto.	
Resource Control Policies (RCPs), eine neue Art von Autorisierungsrichtlinie, AWS Organizations ist auch für Amazon S3 S3-Buckets verfügbar	Richtlinien zur Ressource nkontrolle (RCPs), eine neue Autorisierungsrichtlinie, die in verwaltet wird, AWS Organizations kann verwendet werden, um die maximal verfügbaren Berechtigungen für Amazon S3 S3-Buckets in Ihrer gesamten Organisation festzulegen. Weitere Informati onen finden Sie unter Richtlini en zur Ressourcenkontrolle (RCPs) im AWS Organizations Benutzerhandbuch.	13. November 2024
Amazon S3 genehmigt jetzt automatisch eine Erhöhung des Bucket-Kontingents auf bis zu 1.000 Buckets	Amazon S3 genehmigt jetzt automatisch eine Erhöhung des Bucket-Kontingents auf bis zu 1.000 Buckets. Um Ihre Bucket-Auslastung einzusehe n oder eine Erhöhung zu beantragen, besuchen Sie die Service Quotas Quotas-Ko nsole.	30. September 2024

Neues Standardverhalten bei der Umstellung auf minimale Objektgrößen für Amazon S3 Lifecycle-Konfigurationen	Amazon S3 wendet jetzt ein Standardverhalten auf S3-Lifecycle-Konfigurationen an, das verhindert, dass Objekte, die kleiner als 128 KB sind, in eine beliebige Speicherklasse übertragen werden. Informationen zum Überschreiben dieses Verhaltens finden Sie unter Zulassen der Übertragung von Objekten, die kleiner als 128 KB sind.	24. September 2024
Directory-Buckets werden jetzt unterstützt, KMS wobei SSE vom Kunden verwaltete Schlüssel verwendet werden.	Directory-Buckets unterstützen jetzt serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) -Schlüsseln (SSE-KMS) unter Verwendung von kundenverwalteten Schlüsseln. Sie haben mehr Optionen zum Verschlüsseln und Verwalten der Sicherheit Ihrer Daten in Verzeichnis-Buckets. Weitere Informationen finden Sie unter Datenschutz und Verschlüsselung in Verzeichnis-Buckets.	17. September 2024

S3 Access Grants unterstützt eine API Aktion, die die Zugriffsberechtigungen des Anrufers auflistet	IAMIdentitäten und IAM Identity Center-Unternehmensverzeichnisidentitäten können jetzt die verwenden , `ListCallerAccessGrants` API um alle Amazon S3 S3-Buckets, -Präfixe und Objekte aufzulisten, auf die sie zugreifen können, wie in ihren S3 Access Grants definiert. Verwenden Sie diese OptionAPI, um alle S3-Daten zu ermitteln, auf die eine IAM oder die Identität des Unternehmensverzeichnisses über S3 Access Grants innerhalb eines bestimmten Bereichs zugreifen kann. AWS-Konto Weitere Informationen finden Sie unter Auflisten der Zugriffsberechtigungen des Anrufers.	5. September 2024

Verbesserte Fehlermeldungen mit verweigertem Zugriff bei Anfragen mit demselben Konto	Amazon S3 enthält jetzt zusätzlichen Kontext in Access Denied (HTTP `403 Forbidden`) -Fehler für Anfragen an Ressourcen innerhalb desselben AWS-Konto. Dieser neue Kontext umfasst die Art der Richtlini e, die den Zugriff verweiger t hat, den Grund für die Ablehnung und Informati onen über den AWS Identity and Access Managemen t (IAM) -Benutzer oder die Rolle, die den Zugriff auf die Ressource angefordert hat. Dieser Kontext hilft Ihnen bei der Behebung von Zugriffsp roblemen, der Identifizierung der Hauptursache von Fehlern bei der Zugriffsverweigeru ng und bei der Behebung falscher Zugriffskontrollen, indem Sie die entsprechenden Richtlinien aktualisieren. Dieser zusätzliche Kontext ist auch in AWS CloudTrai l Protokollen verfügbar . Verbesserte Fehlermel dungen mit dem Status „Zugriff verweigert“ für Anfragen mit demselben Konto sind jetzt in allen Regionen verfügbar AWS-Regionen, auch in den Regionen China AWS GovCloud (US) Regions und	21. August 2024

	China. Weitere Informationen finden Sie unter Beheben von Fehlern mit Zugriffsverweigerung (403 Forbidden) in Amazon S3. .	
Amazon S3 unterstützt die Verwendung von bedingten Schreibvorgängen für `PutObject` und `CompleteMultipartUpload`	Sie können überprüfen, ob ein Objekt in Ihrem Bucket vorhanden ist, bevor Sie es mithilfe von bedingten Schreibvorgängen beim Hochladen erstellen. Dadurch kann verhindert werden, dass vorhandene Daten überschrieben werden. Durch bedingte Schreibvorgänge wird bestätigt, dass sich in Ihrem Bucket noch kein Objekt mit demselben Schlüssel namen befindet. Weitere Informationen finden Sie unter Bedingte Anfragen.	20. August 2024
Amazon S3 berechnet keine Gebühren mehr für mehrere HTTP Fehlercodes	Amazon S3 hat eine Änderung vorgenommen, sodass nicht autorisierte Anfragen, die Kunden nicht initiiert haben, kostenlos sind. Weitere Informationen finden Sie unter Abrechnung für Amazon S3 S3-Fehlerantworten.	19. August 2024

Amazon S3 Select ist für Neukunden nicht mehr verfügbar	Amazon S3 Select ist für Neukunden nicht mehr verfügbar. Bestandskunden von Amazon S3 Select können die Funktion weiterhin wie gewohnt nutzen. Weitere Informationen.	25. Juli 2024
Amazon S3 Inventory unterstützt den InventoryAccessibleOptionalFields Zustandsschlüssel s3:	Amazon S3 Inventory unterstützt den InventoryAccessibleOptionalFields Bedingungsschlüssel s3: zur Steuerung, ob Benutzer optionale Metadatenfelder in ihre Berichte aufnehmen können. Weitere Informationen finden Sie unter Steuern der Erstellung der Konfiguration von S3-Inventarberichten.	20. Februar 2024
IPv6Unterstützung für S3 auf Outposts	Sie können jetzt auf S3 in Outposts-Buckets zugreifen, indem Sie IPv6 über S3 auf Outposts-Dual-Stack-Endpunkten auf Outposts zugreifen. IPv6Die Unterstützung für S3 on Outposts ermöglicht es Ihnen, Ihre S3 on Outposts-Buckets zu verwalten und Ebenenressourcen über Netzwerke zu steuern. IPv6	16. Januar 2024

Neue leistungsstarke Amazon-S3-Speicherklasse mit einer Zone – S3 Express One Zone	Amazon S3 Express One Zone ist eine leistungsstarke Amazon-S3-Speicherklasse mit einer einzelnen Zone, die speziell für den konsisten ten Datenzugriff im einstelli gen Millisekundenbereich für Ihre latenzempfindlichs ten Anwendungen entwickelt wurde. Weitere Informationen finden Sie unter S3 Express One Zone.	28. November 2023
Mountpoint für Amazon S3 fügt Unterstützung für S3 Express One Zone hinzu	Sie können jetzt S3-Express-One-Zone-Verzeichnis-Buck ets mit Mountpoint mounten.	28. November 2023
Lambda-Aufrufschemaversion	Amazon S3 Batch Operation s führt eine neue Lambda-Aufrufschemaversion für Batch-Operations-Aufträge ein, die auf Verzeichnis-Buckets ausgeführt werden. Verwenden von Lambda und Amazon S3 Batch Operations mit Verzeichnis-Buckets.	28. November 2023

Importaktion für Verzeichnis-Buckets	Amazon S3 führt die Importaktion ein. Import ist eine optimierte Methode zur Erstellung von Amazon-S3-Batch-Operations-Aufträgen zum Kopieren von Objekten aus Allzweck-Buckets in Verzeichnis-Buckets. Weitere Informationen finden Sie unter Importieren von Objekten in einen Verzeichnis-Bucket .	28. November 2023
Verwalten des S3-Zugriffs mit S3 Access Grants	Amazon S3 Access Grants ermöglicht es Ihnen, Datenberechtigungen für AWS Identity and Access Management (IAM) Prinzipale zusätzlich zu Verzeichnisidentitäten aus Unternehmensverzeichnissen wie Azure AD. Sie können jetzt S3-Berechtigungen mit den geringsten Rechten durchsetzen und diese Berechtigungen ganz einfach an Ihre Geschäftsanforderungen anpassen. Weitere Informationen finden Sie unter Verwalten des Datenzugriffs mit S3 Access Grants.	26. November 2023
Mountpoint für Amazon S3 fügt Caching-Feature hinzu	Mit Mountpoint können Sie jetzt das Caching für Daten konfigurieren, auf die wiederholt zugegriffen wird.	22. November 2023

Verbesserte Amazon S3 Batch Operations-Manifestgenerierung	Sie können Amazon S3 Batch Operations jetzt anweisen, automatisch ein Manifest auf der Grundlage von Objektfilterkriterien zu generieren, die Sie bei der Erstellung Ihres Auftrags angeben. Diese Option ist für Batch-Replikationsaufträge verfügbar, die Sie in der Amazon S3-Konsole erstellen, oder für jeden Auftragstyp, den Sie mithilfe von AWS CLI AWS SDKs, oder Amazon S3 erstellen RESTAPI. Weitere Informationen finden Sie unter Erstellen eines Amazon-S3-Batch-Operations-Auftrags.	22. November 2023
Bestehende Amazon-S3-Buckets können jetzt Object-Lock-Konfigurationen hinzufügen	Sie können Object Lock jetzt für einen vorhandenen Amazon-S3-Bucket aktivieren. Sie können gesetzliche und andere Aufbewahrungsfristen für neue oder bestehende Buckets festlegen. Weitere Informationen finden Sie unter Verwenden von Object Lock.	20. November 2023
S3-Storage-Lens-Anforderungsmetriken für Präfixe	In S3 Storage Lens werden Anforderungsmetriken für Präfixe in einem Amazon-S3-Bucket eingeführt. Weitere Informationen finden Sie unter Metrikkategorien.	17. November 2023

Amazon-S3-Storage-Lens-Gruppen	In S3 Storage Lens werden Storage-Lens-Gruppen eingeführt. Hierbei handelt es sich um benutzerdefinierte Filter für Objekte auf der Grundlage von Objektmetadaten. Weitere Informationen finden Sie unter Arbeiten mit S3-Storage-Lens-Gruppen.	15. November 2023
Neue IAM Richtlinie	S3 on Outposts führt die serviceverknüpfte Rolle `AWSServiceRoleForS3OnOutposts` ein, die Sie bei der Verwaltung von Netzwerkressourcen unterstützt. Weitere Informationen finden Sie unter Verwenden von serviceverknüpften Rollen für S3 on Outposts.	3. Oktober 2023
Amazon S3 bietet den `Last-Modified` -Zeitpunkt für das Löschen von Markierungen	Amazon S3 gibt die `Last-Modified` Uhrzeit der Löschmarkierungen in den Antwort-Headern von S3 `Head` und `Get` API Vorgängen an. Weitere Informationen finden Sie unter Arbeiten mit Löschmarkierungen.	27. September 2023

Amazon S3 S3-Update für AWS verwaltete Richtlinien	Amazon S3 hat die `s3:Describe*` -Berechtigungen zu `AmazonS3ReadOnlyAccess` hinzugefügt. Weitere Informationen finden Sie unter Von AWS verwaltete Richtlinien für Amazon S3.	11. August 2023
Verbesserte Startzeiten für Standard-Wiederherstellungsanfragen, die über S3 Batch Operations gestellt wurden	Standardabrufe für Wiederherstellungsanfragen, die über S3 Batch Operations gestellt wurden, können jetzt innerhalb von Minuten beginnen. Weitere Informationen finden Sie unter Optionen für den Archivabruf.	9. August 2023
Mountpoint, ein Client mit hohem Durchsatz zum Mounten eines Amazon-S3-Buckets als lokales Dateisystem wurde hinzugefügt.	Mit Mountpoint können Ihre Anwendungen über Dateioperationen auf die in Amazon S3 gespeicherten Objekte zugreifen, sodass Ihre Anwendungen über eine Dateischnittstelle auf den elastischen Speicher und den Durchsatz von Amazon S3 zugreifen können.	9. August 2023

Zweischichtige serverseitige Verschlüsselung mit AWS Key Management Service Schlüsseln (DSSE-) KMS	Die zweischichtige serversei tige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln (DSSE-KMS) wendet zwei Verschlüsselungsebenen auf Objekte an, wenn sie auf Amazon S3 hochgeladen werden. Weitere Informationen finden Sie unter Verwenden der serverseitigen Dual-Layer-Verschlüsselung mit Schlüssel n. AWS KMS	13. Juni 2023
Amazon S3 aktiviert S3 Block Public Access und deaktivie rt S3-Zugriffskontrolllisten (ACLs) für alle neuen Buckets.	Amazon S3 aktiviert jetzt automatisch S3 Block Public Access und deaktiviert S3-Zugriffskontrolllisten (ACLs) für alle neuen S3-Buckets in allen AWS Regionen. Weitere Informationen finden Sie unter Sperren des öffentlichen Zugriffs auf Ihren Amazon S3 S3-Speicher und Steuern des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.	27. April 2023
Metrik zu fehlgeschlagenen S3-Replikationsvorängen	Amazon S3 fügt eine neue Amazon CloudWatch Metrik zur Überwachung von S3-Replikationsfehlern hinzu. Weitere Informationen finden Sie unter Überwachen des Fortschritts mit Replikati onsmetriken.	05. April 2023

Privat DNS	AWS PrivateLink für Amazon S3 unterstützt jetzt PrivateDNS. Weitere Informationen finden Sie unter Privat DNS.	14. März 2023
Support für kontoüber greifende Zugriffspunkte in der Amazon-S3-Konsole	Amazon S3 unterstützt jetzt die Erstellung von kontoüber greifenden Zugriffspunkten mit der Amazon-S3-Konsole. Weitere Informationen finden Sie unter Erstellen von Zugriffspunkten.	14. März 2023
Amazon S3 in Outposts unterstützt S3 Replication in Outposts	Mit der lokalen S3-Replikation können Sie Objekte automatisch in einen einzelnen Ziel-Bucket oder in mehrere Ziel-Buckets von Outposts replizieren. Die Ziel-Buckets können sich in verschiedenen Outposts AWS Outposts oder innerhalb derselben Outposts wie der Quell-Bucket befinden. Weitere Informationen finden Sie unter Replikation von Objekten für S3 in Outposts.	14. März 2023

Access-Point-Alias in Amazon S3 Object Lambda	Wenn Sie einen Object Lambda Access Point erstellen , generiert Amazon S3 automatisch einen eindeutigen Alias für den Object Lambda Access Point. Sie können diesen Alias anstelle eines Amazon S3 S3-Bucket-Namens oder des Object Lambda Access Point Amazon Resource Name (ARN) in einer Anfrage für Access Point-Datenebenenoperationen verwenden. Weitere Informationen finden Sie unter Verwenden eines Alias im Bucket-Stil für Ihren Object Lambda Access Point.	14. März 2023
Kontoübergreifender Support für Multi-Region Access Points in Amazon S3	Amazon S3 unterstützt jetzt die Erstellung von kontoübergreifenden Multi-Region Access Points mit der Amazon-S3-Konsole. Weitere Informationen finden Sie unter Erstellen von Multi-Region Access Points.	14. März 2023

Kontoübergreifende Zugriffspunkte	Amazon S3 unterstützt das Erstellen von kontoüber greifenden Zugriffspunkten. Sie können einen kontoüber greifenden Access Point erstellen, indem Sie die Operation AWS Command Line Interface (AWS CLI) oder die REST API `CreateAccessPoint` Operation verwenden. Weitere Informationen finden Sie unter Erstellen von Zugriffspunkten.	30. November 2022
Amazon S3 unterstützt Failover-Kontrollen für Amazon S3 Multi-Region Access Points	Amazon S3 führt Failover-Kontrollen für Multi-Region Access Points ein. Mit diesen Kontrollen können Sie den S3-Datenzugriffsanforderungsverkehr, der über einen Amazon S3 Multi-Region Access Point weitergeleitet wird, innerhalb weniger Minuten auf eine alternative AWS-Region umstellen, um hochverfügbare Anwendungen zu testen und zu erstellen. Weitere Informationen finden Sie unter Failover-Kontrollen für Amazon S3 Multi-Region Access Points.	28. November 2022

Amazon S3 Storage Lens erhöht die unternehmensweite Sichtbarkeit mit 34 neuen Metriken	S3 Storage Lens führt 34 zusätzliche Metriken ein, um umfassende Möglichkeiten zur Kostenoptimierung aufzudecken, bewährte Methoden für den Datenschutz zu identifizieren und die Leistung von Anwendungsworkflows zu verbessern. Weitere Informationen finden Sie unter Metriken von S3 Storage Lens.	17. November 2022
Amazon S3 unterstützt höhere Geschwindigkeiten für Wiederherstellungsanforderungen für S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive	Amazon S3 unterstützt Wiederherstellungsanfragen mit einer Geschwindigkeit von bis zu 1.000 Transaktionen pro Sekunde AWS-Konto für die Speicherklassen S3 Glacier Flexible Retrieval und S3 Glacier Deep Archive.	15. November 2022

Amazon S3 on Outposts unterstützt zusätzliche S3-Lebenszyklusaktionen und -filter	S3 on Outposts unterstützt zusätzliche S3-Lebenszyklusregeln zur Optimierung des Kapazitätsmanagements. Sie können Objekte ablaufen lassen, wenn sie veralten oder durch neuere Versionen ersetzt werden. Sie können eine Lebenszyklusregel für einen ganzen Bucket oder eine Teilmenge von Objekten in einem Bucket erstellen, indem Sie nach Präfixen, Objekt-Tags oder Objektgröße filtern. Weitere Informationen finden Sie unter Erstellen und Verwalten einer Lebenszyklus-Konfiguration.	2. November 2022
S3-Replikationsunterstützung für SSE -C-Objekte	Sie können Objekte replizieren, die mit der serversei tigen Verschlüsselung unter Verwendung der vom Kunden bereitgestellten Schlüssel erstellt wurden. Weitere Hinweise zum Replizieren verschlüsselter Objekte finden Sie unter Replizieren von Objekten, die mit serversei tiger Verschlüsselung (SSE-C, -S3, SSE -) erstellt wurden. SSE KMS	24. Oktober 2022

Amazon S3 on Outposts unterstützt Zugriffspunkt-Aliasse	Bei S3 on Outposts müssen Sie Zugriffspunkte verwenden, um auf ein Objekt in einem Outposts-Bucket zuzugreifen. Jedes Mal, wenn Sie einen Zugriffspunkt für einen Bucket erstellen, generiert S3 on Outposts automatisch einen Zugriffspunkt-Alias. Sie können diesen Zugriffspunkt-Alias anstelle eines Zugriffspunkts ARN für jeden Vorgang auf der Datenebene verwenden. Weitere Informationen finden Sie unter Verwenden eines Alias im Bucket-Stil für den Zugriffspunkt Ihres S3-on-Outposts-Buckets.	21. Oktober 2022
S3-Objekt-Lambda unterstützt die Operationen `HeadObject`, `ListObjects` und `ListObjectsV2`	Sie können benutzerdefinierten Code verwenden, um die von den S3-Standardanforderungen `GET`, `LIST` oder `HEAD` zurückgegebenen Daten zu ändern, um Zeilen zu filtern, die Größe von Images dynamisch anzupassen, vertrauliche Daten zu unterdrücken und vieles mehr. Weitere Informationen finden Sie unter Transformieren von Objekten mit S3-Objekt Lambda.	4. Oktober 2022

Amazon S3 on Outposts unterstützt die S3-Versionsverwaltung	Wenn diese Option aktiviert ist, speichert die S3-Versionsverwaltung mehrere unterschiedliche Kopien eines Objekts im selben Bucket. Sie können die S3-Versionsverwaltung verwenden, um sämtliche Versionen aller Objekte in Ihren Outposts-Buckets zu speichern, abzurufen oder wiederherzustellen. Mit der S3-Versionsverwaltung können Sie Versionen sowohl nach unbeabsichtigten Benutzeraktionen als auch nach Anwendungsfehlern problemlos wiederherstellen. Weitere Informationen finden Sie unter Verwalten der S3-Versionsverwaltung für Ihren S3-on-Outposts-Bucket.	21. September 2022
AWS Backup für Amazon S3	AWS Backup ist ein vollständig verwalteter, richtlinienbasierter Service, mit dem Sie eine zentrale Backup-Richtlinie zum Schutz Ihrer Amazon S3 S3-Daten definieren können. Weitere Informationen finden Sie unter Verwenden AWS Backup für Amazon S3.	18. Februar 2022

Verwenden der S3-Batchreplikation, um bestehende Objekte zu replizieren	Mit der S3-Batchreplikation können Sie Objekte replizieren, die vor der Einführung einer Replikationskonfiguration existierten. Das Replizieren vorhandener Objekte erfolgt mithilfe eines Auftrags für Batchoperationen. Die S3-Batchreplikation unterscheidet sich von der Live-Replikation, die kontinuierlich und automatisch neue Objekte über Amazon-S3-Buckets hinweg kopiert. Weitere Informationen finden Sie unter Replizieren vorhandener Objekte mit S3-Batch-Replikation.	8. Februar 2022
Umbenennen von S3 Glacier Flexible Retrieval	Die Glacier-Speicherklasse wurde in S3 Glacier Flexible Retrieval umbenannt. Diese Änderung hat keine Auswirkungen auf dieAPI.	30. November 2021

Neue Einstellung für S3-Objekteigentümer zum Deaktivieren ACLs	Sie können die vom Bucket Ownership erzwungene Einstellung für Object Ownership anwenden, um sie ACLs für Ihren Bucket und die darin enthaltenen Objekte zu deaktivieren und die Verantwortung für jedes Objekt in Ihrem Bucket zu übernehmen. Die vom Bucket-Eigentümer erzwungene Einstellung vereinfacht die Zugriffsverwaltung für in Amazon S3 gespeicherte Daten. Weitere Informationen finden Sie unter Kontrolle des Besitzes von Objekten und Deaktivierung ACLs für Ihren Bucket.	30. November 2021
Neue S3-Intelligent-Tiering-Speicherklasse	S3 Intelligent-Tiering Archive Instant Access ist eine zusätzliche Speicherklasse unter S3 Intelligent-Tiering. Weitere Informationen finden Sie unter Wie funktioniert S3 Intelligent-Tiering.	30. November 2021
Neue Speicherklasse S3 Glacier Instant Retrieval	Sie können jetzt Objekte in der Speicherklasse S3 Glacier Instant Retrieval platzieren. Weitere Informationen zu dieser Speicherklasse finden Sie unter Verwenden von Amazon-S3-Speicherklassen.	30. November 2021

AWS Backup für Amazon S3 Preview	AWS Backup ist ein vollständig verwalteter, richtlinienbasierter Service, mit dem Sie eine zentrale Backup-Richtlinie zum Schutz Ihrer Amazon S3 S3-Daten definieren können. Weitere Informationen finden Sie unter Verwenden AWS Backup für Amazon S3.	30. November 2021
AWS Identity and Access Management Access Analyzer für Amazon S3	IAMAccess Analyzer führt Richtlinienprüfungen durch, um Ihre Richtlinie anhand der IAM Richtliniengrammatik und der bewährten Methoden zu validieren. Weitere Informationen zur Überprüfung von Richtlinien mit IAM Access Analyzer finden Sie unter IAMAccess Analyzer-Richtlinienvalidierung im IAMBenutzerhandbuch.	30. November 2021
Neue Ereignistypen	Zu Amazon-S3-Ereignisbenachrichtigungen hinzugefügte Ereignistypen finden Sie unter Amazon-S3-Ereignisbenachrichtigungen.	29. November 2021
Amazon für EventBridge Buckets aktivieren	Sie können EventBridge auf Amazon S3 S3-Buckets das Senden von Ereignissen an Amazon aktivieren EventBridge, siehe Verwenden EventBridge.	29. November 2021

Neue S3-Lebenszyklusfilter	Sie können Lebenszyklusregeln basierend auf der Objektgröße erstellen oder angeben, wie viele nicht aktuelle Objektversionen beibehalten werden sollen. Weitere Informationen finden Sie unter Beispiele der S3-Lebenszyklus-Konfiguration.	23. November 2021
Veröffentlichen Sie Amazon S3 Storage Lens-Metriken auf Amazon CloudWatch	Sie können die Nutzungs- und Aktivitätsmetriken von S3 Storage Lens CloudWatch auf Amazon veröffentlichen, um eine einheitliche Ansicht Ihres Betriebszustands in CloudWatch Dashboards zu erstellen. Sie können auch CloudWatch Funktionen wie Alarme und ausgelöste Aktionen, metrische Berechnungen und Anomalieerkennung verwenden, um die S3 Storage Lens-Metriken zu überwachen und entsprechende Maßnahmen zu ergreifen. Darüber hinaus CloudWatch APIs ermöglichen sie Anwendungen, einschließlich Drittanbietern, den Zugriff auf Ihre S3 Storage Lens-Metriken. Weitere Informationen finden Sie unter Messdaten zur Überwachung von S3 Storage Lens unter CloudWatch.	22. November 2021

Multiregionale Zugriffspunkte	Sie können Multi-Region Access Points verwenden, um einen globalen Endpunkt zu erstellen, mit dem Anwendungen Anforderungen von Amazon-S3-Buckets in mehreren AWS-Regionen ausführen können. Sie können diesen Multi-Region Access Point verwenden, um Daten an einen Bucket mit der niedrigsten Latenz weiterzuleiten. Weitere Informationen zu Multi-Regions-Zugriffspunkte und deren Verwendung finden Sie unter Multi-Regions-Zugriffspunkte in Amazon S3.	2. September 2021

Amazon S3 in Outposts fügt direkten lokalen Zugriff für Anwendungen hinzu	Führen Sie Ihre Anwendungen außerhalb der AWS Outposts Virtual Private Cloud (VPC) aus und greifen Sie auf Ihre S3 on Outposts-Daten zu. Sie können auch direkt aus Ihrem lokalen Netzwerk auf S3 in Outposts zugreifen. Weitere Informationen zur Konfiguration von S3 auf Outposts-Endpunkten mit kundeneigenen IP-Adressen (CoIP) und zum Zugreifen auf Ihre Objekte, indem Sie ein lokales Gateway von Ihrem lokalen Netzwerk aus einrichten, finden Sie unter Zugreifen auf Amazon S3 auf Outposts mit Nur-Zugriffspunkten. VPC	29. Juli 2021
Amazon-S3-Zugriffspunktalias	Wenn Sie einen Zugriffspunkt erstellen, generiert Amazon S3 automatisch einen Alias, den Sie anstelle eines Bucket-Namens für den Datenzugriff verwenden können. Sie können diesen Access Point-Alias anstelle eines Amazon-Ressourcennamens (ARN) für jeden Access Point-Datenebenenvorgang verwenden. Weitere Informationen finden Sie unter Verwenden eines Alias im Bucket-Stil für Ihren Zugriffspunkt.	26. Juli 2021

Amazon S3 Inventory und S3 Batch Operations unterstützen den Status des S3-Bucket-Schlüssels	Amazon-S3-Bestand und Batch-Vorgänge unterstützen das Identifizieren und Kopieren vorhandener Objekte mit S3-Bucket-Schlüssel. S3-Bucket-Schlüssel beschleunigen die Preissenkung der serverseitigen Verschlüsselung für vorhandene Objekte. Weitere Informationen finden Sie unter Amazon S3 Inventory und Batch Operations Copy object (Batch-Vorgänge Objekt kopieren).	3. Juni 2021
Amazon S3 Storage-Lens Metriken Konto-Snapshot	Der S3-Storage-Lens-Konto-Snapshot zeigt Ihren Gesamtspeicher, die Objektzahl und die durchschnittliche Objektgröße auf der Startseite der S3-Konsole (Buckets) an, indem Metriken aus Ihrem Standard-Dashboard zusammengefasst werden. Weitere Informationen finden Sie unter S3 Storage Lens Metrics Konto-Snapshot.	5. Mai 2021
Erhöhte Unterstützung von Amazon-S3-in-Outposts-Endpunkten	S3 in Outposts unterstützt jetzt bis zu 100 Endpunkte pro Outpost. Weitere Informationen finden Sie unter S3-in-Outposts-Netzeinschränkungen.	29. April 2021

Amazon S3 on Outposts — Ereignisbenachrichtigungen in Amazon CloudWatch Events	Sie können CloudWatch Ereignisse verwenden, um eine Regel zu erstellen, mit der alle S3 on API Outposts-Ereignisse erfasst und über alle unterstützten CloudWatch Ziele benachrichtigt werden. Weitere Informationen finden Sie unter Empfangen von S3 on Outposts-Ereignisbenachrichtigungen mithilfe von CloudWatch Ereignissen.	19. April 2021
S3 Objekt Lambda	Mit S3 Object Lambda können Sie GET-Anforderungen von Amazon S3 eigenen Code hinzufügen, um Daten zu ändern und zu verarbeiten, wenn sie an eine Anwendung zurückgegeben werden. Sie können mit benutzerdefiniertem Code die von standardmäßigen S3-GET-Anforderungen zurückgegebenen Daten ändern, um u. a. Zeilen zu filtern, Bilder dynamisch in der Größe zu ändern und vertrauliche Daten zu redigieren und mehr. Weitere Informationen finden Sie unter Transformieren von Objekten.	18. März 2021

AWS PrivateLink	Mit AWS PrivateLink Amazon S3 können Sie eine direkte Verbindung zu S3 herstelle n, indem Sie einen Schnittst ellenendpunkt in Ihrer virtuelle n privaten Cloud (VPC) verwenden, anstatt eine Verbindung über das Internet herzustellen. Schnittstellen-Endpunkte sind direkt von Anwendungen aus zugänglic h, die sich vor Ort oder in einer anderen AWS-Region befinden. Weitere Informati onen finden Sie unter AWS PrivateLink für Amazon S3.	2. Februar 2021
Verwaltung der Kapazität von Amazon S3 on Outposts mit AWS CloudTrail	Verwaltungsereignisse von S3 on Outposts sind in CloudTrail Protokollen verfügbar. Weitere Informationen finden Sie unter Verwaltung der Kapazität von S3 on Outposts mit CloudTrail.	21. Dezember 2020

Starke Konsistenz	Amazon S3 bietet insgesamt eine hohe read-after-write Konsistenz für Objekte PUT und DELETE Anfragen von Objekten in Ihrem S3-Bucket AWS-Regionen. Darüber hinaus sind Lesevorgänge in Amazon S3 Select, Amazon-S3-Zugriffskontrolllisten, Amazon-S3-Objekt-Tags und Objekt-Metadaten (z. B. HEAD-Objekt) durchweg konsisten t. Weitere Informationen finden Sie unter Amazon-S3-Datenkonsistenzmodell.	1. Dezember 2020
Synchronisierung der Amazon-S3-Replikatänderung	Die Amazon S3 S3-Replik atmodifikationssynchronisie rung sorgt dafür, dass Objektmetadaten wie Tags und Object Lock-Einstellungen zwischen Quellobjekten und Replikaten synchron bleiben. ACLs Wenn diese Funktion aktiviert ist, repliziert Amazon S3 Metadatenänderungen, die entweder am Quellobje kt oder an den Replikatk opien vorgenommen wurden. Weitere Informationen finden Sie unter Replizieren von Metadatenänderungen mit der Synchronisierung von Replikatänderungen.	1. Dezember 2020

Amazon-S3-Bucket-Schlüssel	Amazon S3 Bucket Keys reduzieren die Kosten der serverseitigen Amazon S3 S3-Verschlüsselung mit AWS Key Management Service (SSE-KMS). Dieser neue Bucket-Level-Schlüssel für serversei tige Verschlüsselung kann die Kosten für AWS KMS -Anfr agen um bis zu 99 Prozent senken, indem der Anforderu ngsdatenverkehr von Amazon S3 zu AWS KMS verringert wird. Weitere Informationen finden Sie unter Senkung der Kosten von SSE — mithilfe KMS von S3 Bucket Keys.	1. Dezember 2020

Amazon-S3-Storage-Lens	S3 Storage Lens aggregiert Ihre Metriken und zeigt die Informationen im Abschnitt Account snapshot (Konto-Snapshot) der Seite Buckets der Amazon-S3-Konsole an. S3 Storage Lens bietet außerdem ein interaktives Dashboard, mit dem Sie Erkenntnisse und Trends visualisieren, Ausreißer kennzeichnen und Empfehlungen zur Optimierung der Speicherkosten erhalten und bewährte Datenschutzmethoden anwenden können. Das Dashboard verfügt über Drilldown-Optionen, um Erkenntnisse auf Organisations-, Konto-, AWS-Region-, Speicherklassen-, Bucket-, Präfix- oder Objekt- oder Storage-Lens-Gruppenebene zu generieren. Sie können auch einen täglichen Metrikexport in CSV oder senden Parquet in einen S3-Bucket formatieren. Weitere Informationen finden Sie unter Bewertung Ihrer Speicheraktivität und -nutzung mit S3 Storage Lens.	18. November 2020

Verfolgen von S3-Anfragen mit AWS X-Ray	Amazon S3 lässt sich in X-Ray integrieren, um den Trace-Kontext zu verbreiten und Ihnen eine Anforderungskette mit Upstream- und Downstream-Knoten bereitzustellen. Weitere Informationen finden Sie unter Verfolgen von Anforderungen mit X-Ray.	16. November 2020
S3-Replikationsmetriken	S3-Replikationsmetriken enthalten detaillierte Metriken für die Replikationsregeln in Ihrer Replikationskonfiguration. Weitere Informationen finden Sie unter Replikationsmetriken und Amazon-S3-Ereignis-Benachrichtigungen.	9. November 2020
S3-Intelligent-Tiering-Archivzugriff und tiefer Archivzugriff	S3-Intelligent-Tiering-Archivzugriff und tiefer Archivzugriff sind zusätzliche Speicherstufen unter S3-Intelligent-Tiering. Weitere Informationen finden Sie unter Speicherklasse zum automatischen Optimieren häufig und selten aufgerufener Objekte.	9. November 2020

Löschmarkierungsreplikation	Mit der Replikation von Löschmarkierungen können Sie sicherstellen, dass Löschmarkierungen für Ihre Replikationsregeln in Ihre Ziel-Buckets kopiert werden. Weitere Informationen finden Sie unter Verwenden der Löschmarkierungs-Replikation.	9. November 2020
S3 Object Ownership	Object Ownership ist eine S3-Bucket-Einstellung, mit der Sie die Eigentümerschaft an neuen Objekten, die in Ihre Buckets hochgeladen werden, steuern können. Weitere Informationen finden Sie unter Verwenden von S3-Objekt-Eigentümerschaft.	2. Oktober 2020

Amazon S3 in Outposts	Mit Amazon S3 on Outposts können Sie S3-Buckets auf Ihren AWS Outposts Ressourcen erstellen und Objekte für Anwendungen, die lokalen Datenzugriff, lokale Datenverarbeitung und Datenresidenz erfordern , einfach lokal speichern und abrufen. Sie können S3 auf Outposts über das AWS Management Console, AWS CLI AWS SDKs, oder REST API verwenden. Weitere Informationen finden Sie unter Verwenden von Amazon S3 in Outposts.	30. September 2020
Bedingung „Bucket-Eigentümer“	Sie können die Bedingung Amazon S3 S3-Bucket-Besitzer verwenden, um sicherzustellen, dass die Buckets, die Sie in Ihren S3-Vorgängen verwenden AWS-Konten , zu den von Ihnen erwarteten gehören. Weitere Informationen finden Sie unter Bucket-Eigentümer-Bedingung .	11. September 2020

Unterstützung von S3-Batch-Vorgang für die Aufbewahrung von Objektsperren	Sie können jetzt Batch-Vorgänge mit der S3-Objektsperre verwenden, um Aufbewahrungs-Einstellungen auf viele Amazon-S3-Objekte gleichzeitig anzuwenden. Weitere Informationen finden Sie unter Festlegen von Aufbewahrungsfristen für die S3-Objektsperre mit S3-Batchvorgängen.	4. Mai 2020
Unterstützung von S3-Batchoperationen für die gesetzliche Aufbewahrungspflicht von Objektsperren	Sie können jetzt Batchvorgänge mit der S3-Objektsperre verwenden, um vielen Amazon-S3-Objekten gleichzeitig eine gesetzliche Aufbewahrungsfrist zuzuweisen. Weitere Informationen finden Sie unter Verwenden von S3-Batch-Vorgänge zum Festlegen von gesetzlichlichen Aufbewahrungsfristen für die S3-Objektsperre.	4. Mai 2020
Aufgaben-Tags für S3-Batchvorgängen	Sie können Ihren S3-BatchVorgängeaufträgen Markierungen hinzufügen, um diese Aufträge zu steuern und zu kennzeichnen. Weitere Informationen finden Sie unter Tags für S3-Batchoperationeaufträge.	16. März 2020

Amazon-S3-Zugriffspunkte	Amazon-S3-Zugriffspunkte vereinfachen die skalierbare Verwaltung des Datenzugriffs auf freigegebene Datensätze in S3. Zugriffspunkte sind benannte Netzwerkendpunkte, die Buckets zugeordnet sind, mit denen Sie S3-Objekt-Vorgänge ausführen können. Weitere Informationen finden Sie unter Verwalten des Datenzugriffs mit Amazon-S3-Zugriffspunkten.	02. Dezember 2019
Zugriffs-Analyzer für Amazon S3	Access Analyzer for Amazon S3 warnt Sie vor S3-Buckets, die so konfiguriert sind, dass sie allen Personen im Internet oder auf anderen Websites Zugriff gewähren AWS-Konten, einschließlich Konten außerhalb Ihrer Organisation. Weitere Informationen finden Sie unter Verwenden des Zugriffs-Analyzer für Amazon S3.	02. Dezember 2019

S3-Replikationszeitsteuerung (S3RTC)	S3 Replication Time Control (S3RTC) repliziert die meisten Objekte, die Sie auf Amazon S3 hochladen, innerhalb von Sekunden und 99,99 Prozent dieser Objekte innerhalb von 15 Minuten. Weitere Informationen finden Sie unter Objekte mit S3 Replication Time Control (S3) replizieren. RTC	20. November 2019
Same-Region Replication (SRR, Replikation in derselben Region)	Sie können Same-Region Replication (SRR) verwenden, um Objekte zwischen Amazon S3 S3-Buckets in demselben zu kopieren. AWS-RegionInformationen sowohl zur regionsübergreifenden Replikation (CRR) als auch zur Replikation in derselben Region finden Sie unter Replikation.	18. September 2019
Unterstützung der regionsübergreifenden Replikation für die S3-Objektsperre	Die regionsübergreifende Replikation unterstützt jetzt Objektsperren. Weitere Informationen finden Sie unter Was repliziert Amazon S3?.	28. Mai 2019

S3-Batch-Vorgänge	Mit S3-Batchvorgängen können Sie groß angelegte Batchvorgänge für Amazon-S3-Objekte durchführen. S3-Batch-Vorgänge kann eine einzelne Operation für Listen von Objekten ausführen, die Sie angeben. Ein einziger Auftrag kann die festgeleg te Operation auf Milliarden von Objekten mit mehreren Exabytes an Daten durchführ en. Weitere Informationen finden Sie unter Durchführen von S3-Batch-Vorgänge.	30. April 2019
Region Asien-Pazifik (Hongkong)	Amazon S3 ist jetzt in der Region Asien-Pazifik (Hongkong) verfügbar. Weitere Informationen zu Regionen und Endpunkten von Amazon S3 finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Refer enz.	24. April 2019
Hinzufügung eines neuen Feldes zu den Serverzug riffsprotokollen	Amazon S3 hat den Serverzug riffsprotokollen das folgende neue Feld hinzugefügt: Transport Layer Security (TLS) -Version. Weitere Informati onen finden Sie unter Server-Zugriffsprotokollformat.	28. März 2019

Neue Archivspeicherklasse	Amazon S3 bietet eine neue Archivspeicherklasse, S3 Glacier Deep Archive (`DEEP_ARCHIVE`), zum Speichern selten benötigte r Objekte. Weitere Informati onen finden Sie unter Speicherklassen.	27. März 2019
Hinzufügung neuer Felder zu den Serverzugriffsprotokollen	In Amazon S3 wurden die folgenden neuen Felder zu den Serverzugriffs-Pro tokollen hinzugefügt: Host-ID, Signature Version (Signatur e Version), Cipher Suite, Authentication Type (Authentif izierungstyp) und Host Header. Weitere Informationen finden Sie unter Server-Zu griffsprotokollformat.	5. März 2019
Unterstützung für Amazon-S3-Inventory-Dateien im Parquet-Format	Amazon S3 unterstützt jetzt das Apache Parquet-Format (Parquet) zusätzlic h zu den von Apache optimierten Dateiformaten für Zeilen, Spalten (ORC) und kommagetrennte Werte (CSV) für Inventarausgabedateien. Weitere Informationen finden Sie unter Bestand.	4. Dezember 2018

S3-Objektsperre	Amazon S3 bietet jetzt Object Lock-Funktionen, die Write Once Read Many (WORM) -Schutz für Amazon S3 S3-Objekte bieten. Weitere Informationen finden Sie unter Sperren von Objekten.	26. November 2018
Wiederherstellungsgeschwindigkeit-Upgrade	Mit dem Amazon-S3-Wiederherstellungs-Geschwindigkeits-Upgrade können Sie während der Ausführung einer Wiederherstellung aus der S3-Speicherklasse Glacier Flexible Retrieval die Geschwindigkeit der Wiederherstellung erhöhen. Weitere Informationen finden Sie unter Wiederherstellen archivierter Objekte.	26. November 2018
Wiederherstellen von Ereignisbenachrichtigungen	Amazon-S3-Ereignisbenachrichtigungen unterstützen jetzt Initiierungs- und Abschlussereignisse, wenn Objekte aus der S3-Speicherklasse Glacier Flexible Retrieval wiederhergestellt werden. Weitere Informationen finden Sie unter Ereignis-Benachrichtigungen.	26. November 2018

PUT direkt zur Speicherklasse S3 Glacier Flexible Retrieval	Die PUT-Operation von Amazon S3 unterstützt jetzt beim Erstellen von Objekten die Angabe von S3 Glacier Flexible Retrieval als Speicherklasse. Zuvor mussten Sie Objekte aus anderen Amazon-S3-Speicherklassen in die S3-Speicherklasse Glacier Flexible Retrieval übertragen. Wenn Sie S3 Cross-Region Replication (CRR) verwenden, können Sie jetzt auch S3 Glacier Flexible Retrieval als Speicherklasse für replizierte Objekte angeben. Weitere Informationen zur Speicherklasse S3 Glacier Flexible Retrieval finden Sie unter Speicherklassen. Weitere Informationen zum Angeben der Speicherklasse für replizierte Objekte finden Sie unter Übersicht über die Replikations-Konfiguration. Weitere Informationen zu den REST API Änderungen, die direkt PUT an S3 Glacier Flexible Retrieval vorgenommen wurden, finden Sie unter Dokumentenverlauf: PUT direkt zu S3 Glacier Flexible Retrieval.	26. November 2018

Neue Speicherklasse	Amazon S3 stellt jetzt eine neue Speicherklasse mit dem Namen S3 Intelligent-Tierung (`INTELLIGENT_TIERING`) bereit, die für Langzeitdaten mit veränderlichen oder unbekannten Zugriffsmustern entwickelt wurde. Weitere Informationen finden Sie unter Speicherklassen.	26. November 2018
Amazon S3 Block Public Access	Amazon S3 bietet jetzt die Möglichkeit, den öffentlichen Zugriff auf Buckets und Objekte auf Bucket- oder Kontobasis zu blockieren. Weitere Informationen finden Sie unter Verwenden von Amazon S3 Block Public Access.	15. November 2018

Verbesserungen beim Filtern in Regeln für die regionsübergreifende Replikation () CRR	In einer CRR Regelkonfiguration können Sie einen Objektfilter angeben, um eine Teilmenge von Objekten auszuwählen, auf die die Regel angewendet werden soll. Zuvor konnten Sie ausschließlich nach Objektschlüsselpräfixen filtern. In dieser Version können Sie nach Objektschlüsselpräfixen, einem oder mehreren Objekt-Markierungen oder beidem filtern. Weitere Informationen finden Sie unter CRRSetup: Überblick über die Replikationskonfiguration.	19. September 2018
Neue Amazon S3 Select-Funktionen	Amazon S3 Select unterstützt jetzt Apache Parquet-Eingaben, Abfragen zu verschachtelten JSON Objekten und zwei neue CloudWatch Amazon-Überwachungsmetriken (`SelectScannedBytes` und `SelectReturnedBytes`).	5. September 2018
Updates sind jetzt verfügbar über RSS	Sie können jetzt einen RSS Feed abonnieren, um Benachrichtigungen über Aktualisierungen des Amazon S3 S3-Benutzerhandbuchs zu erhalten.	19. Juni 2018

Frühere Updates

In der folgenden Tabelle sind die wichtigen Änderungen in jeder Version des Amazon-S3-Benutzerhandbuchs vor dem 19. Juni 2018 beschrieben.

Änderungen	Beschreibung	Datum
Aktualisierung von Codebeispielen	Aktualisierte Codebeispiele: • C# – Alle Beispiele wurden für die Verwendung des aufgabenbasierten asynchronen Musters aktualisiert. Weitere Informationen finden Sie unter Amazon Web Services Asynchronous APIs für. NETim AWS SDK for .NET Entwicklerhandbuch. Codebeispiele sind nun mit Version 3 des AWS SDK for .NET konform. • Java – Alle Beispiele wurden für die Verwendung des Client Builder-Modells aktualisiert. Weitere Informationen zum Client Builder-Modell finden Sie unter Erstellen von Service-Clients. • PHP— Alle Beispiele wurden aktualisiert, um AWS SDK for PHP Version 3.0 zu verwenden. Weitere Informati onen zur AWS SDK for PHP Version 3.0 finden Sie unter AWS SDK for PHP. • Ruby — Der Beispielcode wurde aktualisiert, sodass die Beispiele mit Version 3 funktionieren. AWS SDK for Ruby	30. April 2018
Amazon S3 meldet jetzt die flexiblen Abruf- und `ONEZONE_IA` Speicherklassen von S3 Glacier an die Speichermetriken von	Diese Speichermetriken melden nicht nur tatsächliche Byte-Zahlen, sondern beinhalten auch Overhead-Bytes pro Objekt für entsprechende Speicherklassen (`ONEZONE_IA` , `STANDARD_IA` und S3 Glacier Flexible Retrieval): • Für Objekte der Speicherklassen `ONEZONE_IA` und `STANDARD_IA` meldet Amazon S3 Objekte kleiner als 128 KB als 128 KB. Weitere Informationen finden Sie	30. April 2018

Änderungen	Beschreibung	Datum
Amazon CloudWatch Logs	unter Amazon S3 S3-Speicherklassen verstehen und verwalten. • Für Objekte der Speicherklasse S3 Glacier Flexible Retrieval melden die Speichermetriken die folgenden Overheads: • Ein Overhead von 32 KB pro Objekt zu dem Preis, der für Speicherklasse S3 Glacier Flexible Retrieval berechnet wird • Ein Overhead von 8 KB pro Objekt zu dem Preis, der für Speicherklasse `STANDARD` berechnet wird Weitere Informationen finden Sie unter Übergang von Objekten mit Amazon-S3-Lebenszyklus. Weitere Informationen zu Speichermetriken finden Sie unter Metriken mit Amazon überwachen CloudWatch.	
Neue Speicherklasse	Amazon S3 bietet jetzt eine neue Speicherklasse, `STANDARD_IA` (IA für „Infrequent Access“ (seltener Zugriff)), um Objekte zu speichern. Diese Speicherklasse ist für langlebige Daten und Daten mit weniger häufigem Zugriff optimiert. Weitere Informationen finden Sie unter Amazon S3 S3-Speicherklassen verstehen und verwalten.	4. April 2018
Amazon S3 Select	a>Amazon S3 unterstützt jetzt das Abrufen von Objektinh alten auf der Grundlage eines Ausdrucks. SQL Weitere Informationen finden Sie unter Abfragen vorhandener Daten mit Amazon S3 Select.	4. April 2018

Änderungen	Beschreibung	Datum
Region Asien-Pazifik (Osaka – regional)	Amazon S3 ist jetzt in der Region Asien-Pazifik (Osaka – regional) verfügbar. Weitere Informationen zu Regionen und Endpunkten von Amazon S3 finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Refer enz. **Important** Sie können die Region Asien-Pazifik (Osaka – regional) nur in Verbindung mit der Region Asien-Pazifik (Tokio) verwenden. Um Zugang zur Region Asien-Pazifik (Osaka – regional) anzufordern, wenden Sie sich an Ihren Vertriebsmitarbeiter	12. Februar 2018
Erstellungszeitste mpel für Amazon S3 Inventory	Amazon S3 Inventory beinhaltet jetzt einen Zeitstempel mit dem Datum und der Uhrzeit der Erstellung des Amazon-S3-Bestandsberichts. Mithilfe des Zeitstempels können Sie Änderungen in Ihrem Amazon-S3-Speicher ab dem Startzeitpunkt, zu dem der Bestandsbericht erstellt wurde, ermitteln.	16. Januar 2018
Region Europa (Paris)	Amazon S3 ist jetzt in der Region Europa (Paris) verfügbar . Weitere Informationen zu Regionen und Endpunkten von Amazon S3 finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Referenz.	18. Dezember 2017
Region „China (Ningxia)“	Amazon S3 ist jetzt in der Region China (Ningxia) verfügbar Weitere Informationen zu Regionen und Endpunkten von Amazon S3 finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Referenz.	29. November 2017

Änderungen	Beschreibung	Datum
Support für Amazon S3 ORC S3-Lagerbestandsdateien im Format	Amazon S3 unterstützt jetzt das für Apache optimierte Format row columnar (ORC) zusätzlich zum Dateiformat mit kommagetrennten Werten (CSV) für Inventarausgabedateien. Außerdem können Sie jetzt Amazon S3 S3-Inventar standardmäßig abfragen, SQL indem Sie Amazon Athena, Amazon Redshift Spectrum und andere Tools wie Presto, Apache Hive und Apache Spark verwenden. Weitere Informationen finden Sie unter Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory.	17. November 2017
Standard-Verschlüsselung für S3-Buckets	Die Amazon-S3-Standard-Verschlüsselung bietet eine Methode zum Festlegen des Verhaltens der Standard-Verschlüsselung für einen S3-Bucket. Sie können die Standard-Verschlüsselung in einem Bucket festlegen, sodass alle Objekte beim Speichern im Bucket verschlüsselt werden. Die Objekte werden serverseitig mit entweder verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) oder AWS verwalteten Schlüsseln (SSE-KMS) verschlüsselt. Weitere Informationen finden Sie unter Einstellen des Verhaltens der serverseitigen Verschlüsselung für Amazon S3-Buckets.	06. November 2017
Verschlüsselungsstatus im Amazon S3 Inventory	Amazon S3 unterstützt jetzt das Einsetzen eines Verschlüsselungsstatus in Amazon S3 Inventory, sodass Sie sehen können, wie Ihre Objekte im Ruhezustand für die Überwachung der Compliance von Vorschriften oder andere Zwecke verschlüsselt werden. Sie können Amazon S3 S3-Inventar auch so konfigurieren, dass es mit serverseitiger Verschlüsselung (SSE) oder SSE — verschlüsselt wird, KMS sodass alle Lagerbestandsdateien entsprechend verschlüsselt werden. Weitere Informationen finden Sie unter Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory.	06. November 2017

Änderungen	Beschreibung	Datum
Verbesserungen bei der regionsübergreifenden Replikation () CRR	Die regionsübergreifende Replikation unterstützt jetzt Folgendes: • In einem kontoübergreifenden Szenario können Sie eine CRR Konfiguration hinzufügen, um den Besitz des Replikats auf das Replikat zu ändern AWS-Konto , dem der Ziel-Bucket gehört. Weitere Informationen finden Sie unter Ändern des Replikat-Eigentümers. • Standardmäßig repliziert Amazon S3 keine Objekte in Ihrem Quell-Bucket, die mithilfe serverseitiger Verschlüsselung mit Schlüsseln erstellt wurden, die in gespeichert sind. AWS KMS In Ihrer CRR Konfiguration können Sie Amazon S3 jetzt anweisen, diese Objekte zu replizieren. Weitere Informationen finden Sie unter Replikation verschlüsselter Objekte (SSE-S3, SSE -, DSSE - KMSKMS, SSE -C).	06. November 2017
Region Europa (London)	Amazon S3 ist jetzt in der Region Europa (London) verfügbar. Weitere Informationen zu Regionen und Endpunkten von Amazon S3 finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Referenz.	13. Dezember 2016
Region Kanada (Zentral)	Amazon S3 jetzt in der Region Kanada (Zentral) verfügbar Weitere Informationen zu Regionen und Endpunkten von Amazon S3 finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Referenz.	8. Dezember 2016

Änderungen	Beschreibung	Datum
Markieren von Objekten	Amazon S3 unterstützt jetzt das Markieren von Objekten. Das Markieren von Objekten ermöglicht Ihnen, Speicher zu kategorisieren. Präfixe für Objektschlüsselnamen ermöglichen Ihnen außerdem, die Speicherung zu kategorisieren. Die Markierung von Objekten fügt ihr eine weitere Dimension hinzu. Die Markierung bietet weitere Vorteile. Dazu zählen: • Objekt-Tags ermöglichen eine detaillierte Zugriffskontrolle von Berechtigungen (Sie könnten einem IAM Benutzer beispielsweise Berechtigungen für schreibgeschützte Objekte mit bestimmten Tags gewähren). • Differenzierte Kontrolle bei der Spezifizierung einer Lebenszyklus-Konfiguration. Sie können Markierungen spezifizieren, um eine Untermenge von Objekten auszuwählen, auf welche die Lebenszyklus-Regel zutrifft. • Wenn Sie Cross-Region Replication (CRR) konfiguriert haben, kann Amazon S3 die Tags replizieren. Sie müssen der IAM Rolle, die für Amazon S3 erstellt wurde, die erforderlichen Berechtigungen erteilen, damit sie annehmen kann, Objekte in Ihrem Namen zu replizieren. • Sie können CloudWatch Metriken und CloudTrail Ereignisse auch so anpassen, dass Informationen anhand bestimmter Tag-Filter angezeigt werden. Weitere Informationen finden Sie unter Kategorisieren des Speichers mithilfe von Markierungen.	29. November 2016

Änderungen	Beschreibung	Datum
Amazon S3 Lifecycle unterstützt jetzt tagbasierte Filter	Amazon S3 unterstützt jetzt auf Tags basierende Filter bei der Konfiguration des Lebenszyklus. Sie können jetzt Lebenszyklusregeln angeben, in denen Sie ein Schlüssel präfix, ein oder mehrere Objekt-Markierungen oder eine Kombination aus beidem angeben, um eine Untermeng e der Objekte auszuwählen, auf welche die Lebenszyk lusregel zutrifft. Weitere Informationen finden Sie unter Verwaltung des Lebenszyklus von Objekten.	29. November 2016
CloudWatch Metriken für Buckets anfordern	Amazon S3 unterstützt jetzt CloudWatch Metriken für Anfragen an Buckets. Wenn Sie diese Metriken für einen Bucket aktivieren, erstellen diese Metriken Berichte in 1-Minuten-Intervallen. Sie können auch konfigurieren, welche Objekte in einem Bucket diese Anforderungsmetriken melden. Weitere Informationen finden Sie unter Metriken mit Amazon überwachen CloudWatch.	29. November 2016
Amazon S3 Inventory	Amazon S3 unterstützt jetzt den Speicherbestand. Amazon S3 Inventory stellt eine Flatfile-Dateiausgabe mit durch Kommata getrennten Werten (.csv) Ihrer Objekte und der zugehörigen Metadaten auf täglicher oder wöchentlicher Basis für einen S3-Bucket oder ein gemeinsames Präfix (d. h. Objekte, deren Namen mit derselben Zeichenfolge beginnen) bereit. Weitere Informationen finden Sie unter Katalogisieren und Analysieren Ihrer Daten mit S3 Inventory.	29. November 2016

Änderungen	Beschreibung	Datum
Amazon S3 Analytics – Speicherklassen-Analyse	Die neue analytische Funktion von Amazon S3, die Speicherklassen-Analyse, beobachtet Datenzugriffsmuster, anhand derer Sie entscheiden können, wann Sie `STANDARD`-Speicher mit weniger häufigem Zugriff in die Speicherklasse `STANDARD_IA` (IA für „infrequent access“ (seltener Zugriff)) überführen sollen. Wenn die Speicherklassen-Analyse die seltenen Zugriffsmuster für eine gefilterte Datenmenge im Laufe der Zeit beobachtet, können Sie Ihre Lebenszykluskonfigurationen unter Verwendung der Analyseergebnisse verbessern. Diese Funktion beinhaltet auch eine detaillierte tägliche Analyse Ihrer Speichernutzung auf der angegebenen Bucket-, Präfix- oder Tag-Ebene, die Sie in einen S3-Bucket exportieren können.	29. November 2016
Neuer Expedited- und Bulk-Datenabrufe bei der Wiederherstellung archivierter Objekte aus S3 Glacier	Amazon S3 unterstützt jetzt den Expedited- und Bulk-Datenabruf zusätzlich zum Standardabruf bei der Wiederherstellung archivierter Objekte in S3 Glacier. Weitere Informationen finden Sie unter Wiederherstellen eines archivierten Objekts.	21. November 2016
CloudTrail Protokollierung von Objekten	CloudTrail unterstützt die Protokollierung von Amazon S3 API S3-Vorgängen auf Objektebene wie `GetObject` `PutObject` , und`DeleteObject` . Sie können Ihre Event-Selektoren so konfigurieren, dass sie API Operationen auf Objektebene protokollieren. Weitere Informationen finden Sie unter Protokollieren Amazon S3 API S3-Anrufen mit AWS CloudTrail.	21. November 2016
Region USA Ost (Ohio)	Amazon S3 ist jetzt in der Region USA Ost (Ohio) verfügbar. Weitere Informationen zu Regionen und Endpunkten von Amazon S3 finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Referenz.	17. Oktober 2016

Änderungen	Beschreibung	Datum
IPv6Unterstützung für Amazon S3 Transfer Acceleration	Amazon S3 unterstützt jetzt Internet Protocol Version 6 (IPv6) für Amazon S3 Transfer Acceleration. Sie können eine Verbindung zu Amazon S3 herstellen, IPv6 indem Sie den neuen Dual-Stack-Endpunkt für Transfer Acceleration verwenden. Weitere Informationen finden Sie unter Erste Schritte mit Amazon S3 Transfer Acceleration.	6. Oktober 2016
IPv6Unterstützung	Amazon S3 unterstützt jetzt Internet Protocol Version 6 (IPv6). Sie können über IPv6 Dual-Stack-Endpunkte auf Amazon S3 zugreifen. Weitere Informationen finden Sie unter Anfragen an Amazon S3 stellen IPv6 in der Amazon S3 API S3-Referenz.	11. August 2016
Region Asien-Pazifik (Mumbai)	Amazon S3 ist jetzt in der Region Asien-Pazifik (Mumbai) verfügbar. Weitere Informationen zu Regionen und Endpunkten von Amazon S3 finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Referenz.	27. Juni 2016
Amazon S3 Transfer Acceleration	Amazon S3 Transfer Acceleration ermöglicht die schnelle, einfache und sichere Übertragung von Dateien über große Entfernungen zwischen Ihrem Client und einem S3-Bucket . Transfer Acceleration nutzt die Vorteile der CloudFront weltweit verteilten Edge-Standorte von Amazon. Weitere Informationen finden Sie unter Konfigurieren schneller, sicherer Dateiübertragungen mit Amazon S3 Transfer Acceleration.	19. April 2016
Lebenszyklusunterstützung, um abgelaufene Löschmarkierungen für Objekte zu entfernen	Die `Expiration` -Aktion der Lebenszykluskonfiguration erlaubt jetzt, Amazon S3 anzuweisen, abgelaufene Löschmarkierungen für Objekte in einem versionsgesteuerten Bucket zu entfernen. Weitere Informationen finden Sie unter Elemente, die Lebenszyklus-Aktionen beschreiben.	16. März 2016

Änderungen	Beschreibung	Datum
Die Bucket-Lebenszyklus-Konfiguration unterstützt jetzt eine Aktion, um unvollständige mehrteilige Uploads abzubrechen	Die Bucket-Lebenszykluskonfiguration unterstützt jetzt die `AbortIncompleteMultipartUpload` -Aktion. Mit dieser Aktion können Sie Amazon S3 zum Abbrechen mehrteiliger Uploads anweisen, die nicht innerhalb einer bestimmten Anzahl von Tagen nach der Initiierung abgeschlossen werden. Wenn ein mehrteiliger Upload für eine Abbruchoperation in Frage kommt, löscht Amazon S3 alle hochgeladenen Teile und bricht den mehrteiligen Upload ab. Konzeptuelle Informationen finden Sie in folgenden Themen im Amazon-S3-Benutzerhandbuch: • Abbrechen eines mehrteiligen Uploads • Elemente, die Lebenszyklus-Aktionen beschreiben Die folgenden API Operationen wurden aktualisiert, um die neue Aktion zu unterstützen: • PUTBucket-Lebenszyklus — Mit der XML Konfiguration können Sie jetzt die `AbortIncompleteMultipartUpload` Aktion in einer Lebenszyklus-Konfigurationsregel angeben. • Teile auflisten und mehrteiligen Upload initiieren — Beide API Operationen geben jetzt zwei zusätzliche Antwortheader (und`x-amz-abort-rule-id`) zurück`x-amz-abort-date` , wenn der Bucket über eine Lebenszyklusregel verfügt, die die `AbortIncompleteMultipartUpload` Aktion spezifiziert. Diese Header in der Antwort geben an, wann der initiierte mehrteilige Upload für eine Abbruchoperation in Frage kommt, und welche Lebenszyklusregel gilt.	16. März 2016

Änderungen	Beschreibung	Datum
Region Asien-Pazifik (Seoul)	Amazon S3 ist jetzt in der Region Asien-Pazifik (Seoul) verfügbar. Weitere Informationen zu Regionen und Endpunkten von Amazon S3 finden Sie unter Regionen und Endpunkte in der Allgemeine AWS-Referenz.	6. Januar 2016
Neuer Bedingungsschlüssel und eine Änderung des mehrteiligen Uploads	IAMRichtlinien unterstützen jetzt einen Amazon S3 `s3:x-amz-storage-class` S3-Bedingungsschlüssel. Weitere Informationen finden Sie unter Beispiele für Bucket-Richtlinien mit Bedingungsschlüsseln. Sie müssen nicht mehr der Initiator eines mehrteiligen Uploads sein, um Teile hochzuladen und den Upload abzuschließen. Weitere Informationen finden Sie unter Mehrteiliger Upload API und Berechtigungen.	14. Dezember 2015
Umbenennung der US-Standardregion	Die Regionsnamen-Zeichenfolge hat sich von "USA Standard" in "USA Ost (Nord-Virginia)" geändert. Dies ist nur eine Aktualisierung des Regionsnamens, die Funktionalität bleibt unverändert.	11. Dezember 2015

Änderungen	Beschreibung	Datum
Neue Speicherklasse	Amazon S3 bietet jetzt eine neue Speicherklasse, STANDARD _IA (IA, für seltenen Zugriff), zum Speichern von Objekten. Diese Speicherklasse ist für langlebige Daten und Daten mit weniger häufigem Zugriff optimiert . Weitere Informationen finden Sie unter Amazon S3 S3-Speicherklassen verstehen und verwalten. Mit Updates der Lifecycle-Konfiguration können Sie jetzt Objekte in die Speicherklasse STANDARD _IA überführe n. Weitere Informationen finden Sie unter Verwaltung des Lebenszyklus von Objekten. Zuvor hat die Funktion der regionsübergreifenden Replikati on die Speicherklasse des Quellobjekts für Objektreplikate verwendet. Wenn Sie jetzt eine regionsübergreifende Replikation konfigurieren, können Sie eine Speicherklasse für das im Ziel-Bucket erstellte Objektreplikat spezifizieren. Weitere Informationen finden Sie unter Objekte innerhalb und zwischen Regionen replizieren.	16. September 2015
AWS CloudTrail Integration	Mit der neuen AWS CloudTrail Integration können Sie Amazon S3 API S3-Aktivitäten in Ihrem S3-Bucket aufzeichnen. Sie können CloudTrail damit die Erstellung oder Löschung von S3-Buckets, Änderungen der Zugriffsk ontrolle oder Änderungen der Lebenszykluskonfiguration verfolgen. Weitere Informationen finden Sie unter Protokoll ieren Amazon S3 API S3-Anrufen mit AWS CloudTrail.	1. September 2015

Änderungen	Beschreibung	Datum
Erhöhung des Bucket-Limits	Amazon S3 unterstützt jetzt die Erhöhung des Bucket-Limits. Standardmäßig können Kunden bis zu 100 Buckets in ihren eigenen Buckets erstellen. AWS-Konto Kunden, die weitere Buckets benötigen, können dieses Limit erhöhen, indem sie eine Service-Limit-Erhöhung senden. Weitere Informationen zum Erhöhen des Bucket-Kontingents finden Sie unter AWS-Service -Service-Quotas in der allgemeinen AWS -Referenz. Weitere Informationen erhalten Sie unter Mit dem AWS SDKs und Bucket-Kontingente, Beschränkungen und Einschränkungen.	4. August 2015
Aktualisierung des Datenkonsistenzmodells	Amazon S3 unterstützt jetzt read-after-write Konsistenz für neue Objekte, die zu Amazon S3 in der Region USA Ost (Nord-Virginia) hinzugefügt wurden. Vor diesem Update unterstützten alle Regionen außer der Region USA Ost (Nord-Virginia) read-after-write Konsistenz für neue Objekte, die auf Amazon S3 hochgeladen wurden. Mit dieser Erweiterung unterstützt Amazon S3 jetzt read-after-write Konsistenz in allen Regionen für neue Objekte, die zu Amazon S3 hinzugefügt wurden. Read-after-writeKonsistenz ermöglicht es Ihnen, Objekte unmittelbar nach der Erstellung in Amazon S3 abzurufen. Weitere Informationen finden Sie unter Regionen.	4. August 2015
Ereignis-Benachrichtigungen	Amazon-S3-Ereignisbenachrichtigungen wurden aktualisiert, um Benachrichtigungen hinzuzufügen, wenn Objekte gelöscht wurden, und das Filtern für Objektnamen mit Präfix- und Suffix-Vergleich hinzuzufügen. Weitere Informationen finden Sie unter Amazon-S3-Ereignis-Benachrichtigungen.	28. Juli 2015

Änderungen	Beschreibung	Datum
CloudWatch Amazon-Integration	Die neue CloudWatch Amazon-Integration ermöglicht es Ihnen, Ihre Amazon S3-Nutzung anhand von CloudWatch Kennzahlen für Amazon S3 zu überwachen und Alarme einzustellen. Unterstützte Metriken sind unter anderem die Gesamtzahl der Bytes für den Standardspeicher, die Gesamtzahl der Bytes für den Speicher mit reduzierter Redundanz sowie die Gesamtzahl der Objekte für einen bestimmten S3-Bucket. Weitere Informationen finden Sie unter Metriken mit Amazon überwachen CloudWatch.	28. Juli 2015
Unterstützung für das Löschen und Leeren nicht leerer Buckets	Amazon S3 unterstützt jetzt das Löschen und Leeren nicht leerer Buckets. Weitere Informationen finden Sie unter Leeren eines Buckets.	16. Juli 2015
Bucket-Richtlinien für VPC Amazon-Endgeräte	Amazon S3 hat Unterstützung für Bucket-Richtlinien für Virtual Private Cloud (VPC) (VPC) -Endpunkte hinzugefügt. Sie können S3-Bucket-Richtlinien verwenden, um den Zugriff auf Buckets von bestimmten oder bestimmten VPC Endpunkten aus zu steuern. VPCs VPCEndgeräte sind einfach zu konfigurieren, äußerst zuverlässig und bieten eine sichere Verbindung zu Amazon S3, ohne dass ein Gateway oder eine NAT Instance erforderlich ist. Weitere Informationen finden Sie unter Steuern des Zugriffs von VPC Endpunkten aus mit Bucket-Richtlinien.	29. April 2015
Ereignis-Benachrichtigungen	Amazon S3 S3-Ereignisbenachrichtigungen wurden aktualisiert, um die Umstellung auf ressourcenbasierte Berechtigungen für AWS Lambda Funktionen zu unterstützen. Weitere Informationen finden Sie unter Amazon-S3-Ereignis-Benachrichtigungen.	9. April 2015

Änderungen	Beschreibung	Datum
Regionsübergreifende Replikation	Neu: Amazon S3 unterstützt jetzt regionsübergreifende Replikation Regionsübergreifende Replikation ist das automatische, asynchrone Kopieren von Objekten über Buckets in verschiedenen Buckets. AWS-Regionen Weitere Informationen finden Sie unter Objekte innerhalb und zwischen Regionen replizieren.	24. März 2015
Ereignis-Benachrichtigungen	Amazon S3 unterstützt jetzt neue Ereignistypen und Ziele in einer Bucket-Benachrichtigungskonfiguration. Vor dieser Version unterstützte Amazon S3 nur den ReducedRedundancyLostObject Ereignistyp s3: und ein SNS Amazon-Thema als Ziel. Weitere Informationen zu den neuen Ereignistypen finden Sie unter Amazon-S3-Ereignis-Benachrichtigungen.	13. November 2014
Serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln	Serverseitige Verschlüsselung mit AWS Key Management Service (AWS KMS) Schlüsseln (SSE-KMS) Amazon S3 unterstützt jetzt serverseitige Verschlüsselung mit AWS KMS. Mit dieser Funktion können Sie den Envelope-Schlüssel verwalten und Amazon S3 S3-Aufrufe aufrufen AWS KMS, AWS KMS um innerhalb der von Ihnen festgelegten Berechtigungen auf den Envelope-Schlüssel zuzugreifen. Weitere Informationen zur serverseitigen Verschlüsselung mit AWS KMS finden Sie unter Schutz von Daten mithilfe serverseitiger Verschlüsselung mit. AWS Key Management Service	12. November 2014
Region Europa (Frankfurt)	Amazon S3 ist jetzt in der Region Europa (Frankfurt) verfügbar.	23. Oktober 2014

Änderungen	Beschreibung	Datum
Serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln	Amazon S3 unterstützt jetzt serverseitige Verschlüsselung mit vom Kunden bereitgestellten Verschlüsselungsschlüsseln (SSE-C). Die serverseitige Verschlüsselung ermöglicht Ihnen, Amazon S3 aufzufordern, Ihre ruhenden Daten zu verschlüsseln. Wenn Sie SSE -C verwenden, verschlüsselt Amazon S3 Ihre Objekte mit den von Ihnen bereitgestellten benutzerdefinierten Verschlüsselungsschlüsseln. Amazon S3 übernimmt die Verschlüsselung für Sie, deshalb können Sie von der Verwendung Ihrer eigenen Verschlüsselungsschlüssel profitieren, ohne die Kosten für das Schreiben oder Ausführen Ihres eigenen Verschlüsselungscodes tragen zu müssen. Weitere Informationen zu SSE -C finden Sie unter Serverseitige Verschlüsselung (Verwendung von vom Kunden bereitgestellten Verschlüsselungsschlüsseln).	12. Juni 2014
Lebenszyklus-Unterstützung für das Versioning	Vor diesem Release wurde die Lebenszyklus-Konfiguration nur für nicht versionsfähige Buckets unterstützt. Jetzt können Sie den Lebenszyklus sowohl für nicht versioning-fähige, als auch für versioning-fähige Buckets konfigurieren. Weitere Informationen finden Sie unter Verwaltung des Lebenszyklus von Objekten.	20. Mai 2014
Überarbeitung der Themen zur Zugriffskontrolle	Überarbeitung der Dokumentation zur Amazon-S3-Zugriffskontrolle. Weitere Informationen finden Sie unter Identity and Access Management für Amazon S3.	15. April 2014
Überarbeitung des Themas zur Protokollierung des Server-Zugriffs	Überarbeitung der Dokumentation zur Protokollierung des Server-Zugriffs. Weitere Informationen finden Sie unter Protokollieren von Anfragen mit Server-Zugriffsprotokollierung.	26. November 2013
. NETSDKBeispiele wurden auf Version 2.0 aktualisiert	. NETSDKDie Beispiele in diesem Handbuch entsprechen jetzt Version 2.0.	26. November 2013

Änderungen	Beschreibung	Datum
SOAPSupport Over HTTP veraltet	SOAPSupport Over HTTP ist veraltet, aber es ist immer noch verfügbar. HTTPS Neue Amazon S3 S3-Funktionen werden nicht unterstütztSOAP. Wir empfehlen Ihnen, entweder den REST API oder den zu verwenden AWS SDKs.	20. September 2013
Support für IAM-Richt linienvariablen	IAMDie Richtliniensprache unterstützt jetzt Variablen . Wenn eine Richtlinie ausgewertet wird, werden alle Richtlinienvariablen durch Werte ersetzt, die von Kontext-basierten Informationen aus der Sitzung des authentif izierten Benutzers bereitgestellt werden. Sie können Richtlinienvariablen zum Definieren von allgemeinen Richtlinien nutzen, ohne explizit alle Komponenten der Richtlinien aufzulisten. Weitere Informationen zu Richtlini envariablen finden Sie unter Übersicht über IAM Richtlini envariablen im IAMBenutzerhandbuch. Beispiele für Richtlinienvariablen in Amazon S3 finden Sie unter Beispiele für identitätsbasierte Richtlinien für Amazon S3.	3. April 2013
Konsolensupport für die Zahlung durch den Anforderer	Sie können jetzt Ihren Bucket für Requester Pays unter Verwendung der Amazon-S3-Konsole konfigurieren. Weitere Informationen finden Sie unter Nutzen von Buckets mit Zahlung durch den Anforderer für Speicherübertragun gen und Nutzung.	31. Dezember 2012

Änderungen	Beschreibung	Datum
Root-Domänen-Unterstützung für das Website-Hosting	Amazon S3 unterstützt jetzt das Hosting statischer Websites in der Root-Domäne. Besucher Ihrer Website können jetzt von ihrem Browser aus auf Ihre Site zugreifen , ohne www in der Webadresse angeben zu müssen (sie können z. B. example.com statt www.example.com verwenden). Viele Kunden hosten bereits statische Websites auf Amazon S3, die von einer www-Subdomän e aus zugänglich sind (z. B. www.example.com). Zuvor mussten Sie für eine Unterstützung des Root-Domänen-Zugriffs Ihren eigenen Webserver betreiben, um die Root-Domänen-Anfragen über einen Proxy von den Browsern auf Ihre Website auf Amazon S3 weiterzugeben. Der Betrieb eines Web-Servers, um Anforderungen über einen Proxy weiterzugeben, führt zusätzliche Kosten, Betriebsa ufwand und einen weitere potenziellen Ausfallpunkt ein. Jetzt können Sie den Vorteil der hohen Verfügbarkeit und Robustheit von Amazon S3 für www- und Root-Domänen-Adressen nutzen. Weitere Informationen finden Sie unter Hosten einer statischen Website mit Amazon S3.	27. Dezember 2012
Überarbeitung der Konsole	Die Amazon-S3-Konsole wurde aktualisiert. Die Dokumentationsthemen, die auf die Konsole verweisen, wurden entsprechend überarbeitet.	14. Dezember 2012

Änderungen	Beschreibung	Datum
Unterstützung für die Archivierung von Daten in S3 Glacier	Amazon S3 unterstützt jetzt eine Speicheroption, die Ihnen gestattet, den kostengünstigen Speicherservice von S3 Glacier für die Datenarchivierung zu nutzen. Für die Archivierung von Objekten definieren Sie Archivier ungsregeln, die Objekte identifizieren, sowie einen Zeitrahmen, der festlegt, wann Amazon S3 diese Objekte in S3 Glacier archivieren soll. Sie können die Regeln für einen Bucket ganz einfach mit der Amazon S3-Konsol e oder programmgesteuert mit Amazon S3 API oder festlegen. AWS SDKs Weitere Informationen finden Sie unter Verwaltung des Lebenszyklus von Objekten.	13. November 2012
Unterstützung für Seitenumleitungen von Websites	Für einen Bucket, der als Website konfiguriert ist, unterstüt zt Amazon S3 jetzt die Umleitung einer Anfrage für ein Objekt an ein anderes Objekt im selben Bucket oder an ein externes Objekt. URL Weitere Informationen finden Sie unter (Optional) Konfigurieren einer Webseitenumleitung. Weitere Informationen zum Hosting von Websites finden Sie unter Hosten einer statischen Website mit Amazon S3.	4. Oktober 2012
Support für ursprungs übergreifendes Ressourcen-Sharing () CORS	Amazon S3 unterstützt jetzt Cross-Origin Resource Sharing (CORS). CORSdefiniert eine Art und Weise, wie Client-Webanwendungen, die in einer Domain geladen sind, mit Ressourcen in einer anderen Domain interagieren oder auf diese zugreifen können. Mit der CORS Unterstüt zung in Amazon S3 können Sie umfangreiche clientseitige Webanwendungen auf Amazon S3 aufbauen und selektiv domänenübergreifenden Zugriff auf Ihre Amazon S3 S3-Ressourcen gewähren. Weitere Informationen finden Sie unter Verwendung von ursprungsübergreifendem Ressourcenaustausch () CORS.	31. August 2012

Änderungen	Beschreibung	Datum
Unterstützung von Kostenzuordnungs-Markierungen	Amazon S3 unterstützt jetzt Kostenzuordnungs-Markierungen, um S3-Buckets zu markieren, sodass Sie die Kosten für Projekte oder andere Kriterien einfacher nachverfolgen können. Weitere Informationen zur Verwendung der Markierung von Buckets finden Sie unter Verwenden von Kostenzuordnungs-Markierungen für S3-Buckets.	21. August 2012
Support für MFA -geschützten API Zugriff in Bucket-Richtlinien	Amazon S3 unterstützt jetzt den MFA -geschützten API Zugriff, eine Funktion, mit der die AWS Multi-Faktor-Authentifizierung für ein zusätzliches Maß an Sicherheit beim Zugriff auf Ihre Amazon S3 S3-Ressourcen erzwungen werden kann. Es handelt sich um eine Sicherheitsfunktion, bei der Benutzer den physischen Besitz eines MFA Geräts nachweisen müssen, indem sie einen gültigen MFA Code angeben. Weitere Informationen finden Sie unter AWS -Multi-Faktor-Authentifizierung. Sie können jetzt für alle Anfragen zum Zugriff auf Ihre Amazon S3 S3-Ressourcen eine MFA Authentifizierung verlangen. Um die MFA Authentifizierung zu erzwingen, unterstützt Amazon S3 jetzt die `aws:MultiFactorAuthAge` Key-in-a-Bucket-Richtlinie. Ein Beispiel für eine Bucket-Richtlinie finden Sie in Beispiel Erforderlich MFA.	10. Juli 2012
Unterstützung des Ablaufens von Objekten	Sie können das Ablaufen von Objekten nutzen, um das automatische Entfernen von Daten nach einem konfigurierten Zeitraum einzuplanen. Sie legen die Ablaufzeit eines Objekts fest, indem Sie einem Bucket eine Lebenszyklus-Konfiguration hinzufügen.	27. Dezember 2011
Neue unterstützte Region	Amazon S3 unterstützt jetzt die Region Südamerika (São Paulo). Weitere Informationen finden Sie unter Zugreifen auf einen Amazon S3 S3-Bucket.	14. Dezember 2011

Änderungen	Beschreibung	Datum
Multi-Object Delete	Amazon S3 unterstützt jetzt Multi-Object DeleteAPI, sodass Sie mehrere Objekte in einer einzigen Anfrage löschen können. Mit dieser Funktion können Sie eine große Anzahl von Objekten schneller aus Amazon S3 entfernen, als wenn Sie mehrere einzelne DELETE Anfragen verwenden würden. Weitere Informationen finden Sie unter Löschen von Amazon-S3-Objekten.	7. Dezember 2011
Neue unterstützte Region	Amazon S3 unterstützt jetzt die Region USA West (Oregon). Weitere Informationen finden Sie unter Buckets und Regionen.	8. November 2011
Aktualisierung der Dokumentation	Korrektur von Fehlern in der Dokumentation	8. November 2011
Aktualisierung der Dokumentation	Neben den Korrekturen von Fehlern in der Dokumentation enthält diese Version auch die folgenden Erweiterungen: • Neue serverseitige Verschlüsselungsabschnitte mit dem AWS SDK for PHP und dem AWS SDK for Ruby (sieheServerseitige Verschlüsselung mit verwalteten Amazon S3 S3-Schlüsseln (SSE-S3) angeben).	17. Oktober 2011

Änderungen	Beschreibung	Datum
Unterstützung der serverseitigen Verschlüsselung	Amazon S3 unterstützt jetzt die serverseitige Verschlüsselung Diese ermöglicht Ihnen, Amazon S3 aufzufordern, Ihre ruhenden Daten zu verschlüsseln, d. h. Ihre Objektdaten zu verschlüsseln, wenn Amazon S3 Ihre Daten in seinen Rechenzentren auf Datenträger schreibt. Zusätzlich zu den REST API Updates sind das AWS SDK for Java und. NETstellen die erforderlichen Funktionen bereit, um serverseitige Verschlüsselung anzufordern. Sie können auch eine serverseitige Verschlüsselung anfordern, wenn Sie Objekte mit der AWS Management Console hochladen. Weitere Informationen zur Datenverschlüsselung finden Sie unter Verwenden der Datenverschlüsselung.	4. Oktober 2011
Aktualisierung der Dokumentation	Neben den Korrekturen von Fehlern in der Dokumentation enthält diese Version auch die folgenden Erweiterungen: • Ruby und PHP Beispiele wurden dem Abschnitt Anfragen stellen in der Amazon S3 API S3-Referenz hinzugefügt. • Es wurden Abschnitte hinzugefügt, in denen beschrieben wird, wie Presigned URLs generiert und verwendet wird. Weitere Informationen erhalten Sie unter Objekte mit vorsignierten teilen URLs und Objekte mit vorsignierten teilen URLs. • Ein vorhandener Abschnitt wurde aktualisiert, um AWS Explorers für Eclipse und Visual Studio vorzustellen. Weitere Informationen finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS SDKs in der Amazon S3 API S3-Referenz.	22. September 2011

Änderungen	Beschreibung	Datum
Unterstützung von Anforderungen unter Verwendung temporärer Sicherheitsanmeldeinformationen	Sie können nicht nur Ihre Sicherheitsdaten AWS-Konto und IAM Benutzerdaten verwenden, um authentifizierte Anfragen an Amazon S3 zu senden, sondern auch Anfragen mit temporären Sicherheitsanmeldedaten senden, die Sie von AWS Identity and Access Management (IAM) erhalten. Sie können die AWS Security Token Service API oder die AWS SDK Wrapper-Bibliotheken verwenden, um diese temporären Anmeldeinformationen anzufordern. IAM Sie können diese temporären Anmeldeinformationen für Ihre eigene Verwendung erhalten, oder sie verbundenen Benutzern und Anwendungen erteilen. Diese Funktion ermöglicht es Ihnen, Ihre Benutzer außerhalb zu verwalten AWS und ihnen temporäre Sicherheitsanmeldedaten für den Zugriff auf Ihre AWS Ressourcen zur Verfügung zu stellen. Weitere Informationen finden Sie unter Anfragen stellen in der Amazon S3 API S3-Referenz. Weitere Informationen zur IAM Unterstützung temporärer Sicherheitsanmeldedaten finden Sie unter Temporäre Sicherheitsanmeldedaten im IAMBenutzerhandbuch.	3. August 2011

Änderungen	Beschreibung	Datum
Der mehrteilige Upload API wurde um das Kopieren von Objekten mit bis zu 5 TB erweitert	Vor dieser Version API unterstützte Amazon S3 das Kopieren von Objekten mit einer Größe von bis zu 5 GB. Um das Kopieren von Objekten zu ermöglichen, die größer als 5 GB sind, erweitert Amazon S3 jetzt den API mehrteiligen Upload um einen neuen Vorgang. `Upload Part (Copy)` Sie können diese Operation für einen mehrteiligen Upload verwenden, um Objekte mit einer Größe von 5 TB zu kopieren. Weitere Informationen finden Sie unter Objekte kopieren, verschieben und umbenennen. Konzeptionelle Informationen zum mehrteiligen Upload finden Sie API unter. Hochladen und Kopieren von Objekten mit mehrteiligen Uploads	21. Juni 2011
SOAPAPIAnrufe über HTTP sind deaktiviert	Um die Sicherheit zu erhöhen, SOAP API HTTP sind Call-Over deaktiviert. Authentifizierte und anonyme SOAP Anfragen müssen mit SSL an Amazon S3 gesendet werden.	6. Juni 2011
IAM ermöglicht eine kontoübergreifende Delegierung	Bisher benötigte ein IAM Benutzer für den Zugriff auf eine Amazon S3 S3-Ressource Berechtigungen sowohl vom Elternteil AWS-Konto als auch vom Eigentümer der Amazon S3 S3-Ressource. Beim kontoübergreifende n Zugriff benötigt der IAM Benutzer jetzt nur noch die Erlaubnis des Eigentümerkontos. Das heißt, wenn ein Ressourcenbesitzer Zugriff auf eine gewährt AWS-Konto, AWS-Konto kann er seinen IAM Benutzern jetzt Zugriff auf diese Ressourcen gewähren. Weitere Informationen finden Sie im Benutzerhandbuch unter Erstellen einer Rolle zur Delegierung von Berechtigungen an einen IAM IAM Benutzer. Weitere Informationen über die Angabe von Prinzipalen in einer Bucket-Richtlinie finden Sie unter Prinzipien für Bucket-Richtlinien.	6. Juni 2011

Änderungen	Beschreibung	Datum
Neuer Link	Die Endpunktinformationen dieses Service befinden sich jetzt in der allgemeinen AWS -Referenz. Weitere Informationen finden Sie unter Regionen und Endpunkte in der allgemeinen AWS -Referenz.	1. März 2011
Unterstützung des Hostings statischer Websites in Amazon S3	Amazon S3 führt die erweiterte Unterstützung des Hostings statischer Websites ein. Dies beinhaltet die Unterstützung von Indexdokumenten und eines benutzerdefinierten Fehlerdokuments. Wenn Sie diese Funktionen verwenden, geben Anforderungen an das Stammverzeichnis Ihres Buckets oder eines Unterordners (z. B. `http://mywebsite.com/subfolder`) Ihr Indexdokument statt der Liste der Objekte in Ihrem Bucket zurück. Wenn ein Fehler festgestellt wird, gibt Amazon S3 Ihre benutzerdefinierte Fehlermeldung statt einer Amazon-S3-Fehlermeldung zurück. Weitere Informationen finden Sie unter Hosten einer statischen Website mit Amazon S3.	6. Juni 2011
Die Endpunktinformationen dieses Service befinden sich jetzt in der allgemeinen AWS -Referenz. Weitere Informationen finden Sie unter Regionen und Endpunkte in der allgemeinen AWS -Referenz.	1. März 2011	

Änderungen	Beschreibung	Datum
Unterstützung des Hostings statischer Websites in Amazon S3	Amazon S3 führt die erweiterte Unterstützung des Hostings statischer Websites ein. Dies beinhaltet die Unterstützung von Indexdokumenten und eines benutzerdefinierten Fehlerdokuments. Wenn Sie diese Funktionen verwenden , geben Anforderungen an das Stammverzeichnis Ihres Buckets oder eines Unterordners (z. B. `http://mywebsite.com/subfolder`) Ihr Indexdokument statt der Liste der Objekte in Ihrem Bucket zurück. Wenn ein Fehler festgestellt wird, gibt Amazon S3 Ihre benutzerdefinierte Fehlermeldung statt einer Amazon-S3-Fehlermeldung zurück. Weitere Informationen finden Sie unter Hosten einer statischen Website mit Amazon S3.	17. Februar 2011
APISupport für Response-Header	Das GET Objekt ermöglicht es Ihnen REST API jetzt, die Antwortheader der REST GET Objektanforderung für jede Anfrage zu ändern. Das bedeutet, Sie können Objekt-Metadaten in der Antwort ändern, ohne das eigentliche Objekt zu ändern. Weitere Informationen finden Sie unter Herunterladen von Objekten.	14. Januar 2011
Unterstützung für große Objekte	In Amazon S3 wurde die maximale Größe für ein Objekt, das Sie in einem S3-Bucket speichern können, von 5 GB auf 5 TB erhöht. Wenn Sie das verwenden RESTAPI, können Sie Objekte mit bis zu 5 GB in einem einzigen PUT Vorgang hochladen. Bei größeren Objekten müssen Sie den Multipart Upload verwenden RESTAPI, um Objekte in Teilen hochzuladen. Weitere Informationen finden Sie unter Hochladen und Kopieren von Objekten mit mehrteiligen Uploads.	9. Dezember 2010
Mehrteiliger Upload	Der mehrteilige Upload unterstützt schnellere, flexiblere Uploads in Amazon S3. Mithilfe dieser Funktion können Sie ein einzelnes Objekt in mehreren Teilen hochladen . Weitere Informationen finden Sie unter Hochladen und Kopieren von Objekten mit mehrteiligen Uploads.	10. November 2010

Änderungen	Beschreibung	Datum
Unterstützung kanonischer IDs in Bucket-Richtlinien	Sie können jetzt kanonische Richtlinien IDs in Bucket-Richtlinien angeben. Weitere Informationen finden Sie unter Prinzipien für Bucket-Richtlinien	17. September 2010
Amazon S3 funktioniert mit IAM	Dieser Service ist jetzt in AWS Identity and Access Management (IAM) integriert. Weitere Informationen finden Sie IAM im IAMBenutzerhandbuch unter „Dieses Werk mit“.AWS-Services	2. September 2010
Benachrichtigungen	Mit der Amazon S3-Benachrichtigungsfunktion können Sie einen Bucket so konfigurieren, dass Amazon S3 eine Nachricht zu einem Amazon Simple Notification Service (AmazonSNS) -Thema veröffentlicht, wenn Amazon S3 ein wichtiges Ereignis in einem Bucket erkennt. Weitere Informationen finden Sie unter Einrichten von Benachrichtigungen für Bucket-Ereignisse.	14. Juli 2010
Bucket-Richtlinien	Bucket-Richtlinien sind ein Zugriffsmanagementsystem, mit dem Sie Zugriffsberechtigungen für Buckets, Objekte und Objektmengen festlegen. Diese Funktionalität ergänzt Zugriffsrichtlinien und ersetzt sie in vielen Fällen. Weitere Informationen finden Sie unter Bucket-Richtlinien für Amazon S3.	6. Juli 2010
Path-Style-Syntax in allen Regionen verfügbar	Amazon S3 unterstützt jetzt die Pfad-Syntax für alle Buckets in der Region US Classic, oder wenn sich der Bucket innerhalb derselben Region befindet wie der Endpunkt der Anfrage. Weitere Informationen finden Sie unter Virtuelles Hosting.	9. Juni 2010
Neuer Endpunkt für Europa (Irland)	Amazon S3 bietet jetzt einen Endpunkt für Europa (Irland): `http://s3-eu-west-1.amazonaws.com` .	9. Juni 2010

Änderungen	Beschreibung	Datum
Konsole	Sie können Amazon S3 jetzt über die AWS Management Console verwenden. Sie können sich über alle Funktionen von Amazon S3 in der Konsole im Benutzerhandbuch des Amazon Simple Storage Service informieren.	9. Juni 2010
Reduzierte Redundanz	Amazon S3 ermöglicht Ihnen jetzt, Ihre Speicherkosten zu reduzieren, indem Objekte in Amazon S3 mit reduzierter Redundanz gespeichert werden. Weitere Informationen finden Sie unter Reduced Redundancy Storage.	12. Mai 2010
Neue unterstützte Region	Amazon S3 unterstützt jetzt die Region Asien-Pazifik (Singapur). Weitere Informationen finden Sie unter Buckets und Regionen.	28. April 2010
Objekt-Versioning	In dieser Version wird ein Objekt-Versioning eingeführt. Alle Objekte können jetzt einen Schlüssel und eine Version haben. Wenn Sie das Versioning für einen Bucket aktivieren, gibt Amazon S3 allen Objekten, die einem Bucket hinzugefügt werden, eine eindeutige Versions-ID. Diese Funktion ermöglicht Ihnen eine Wiederherstellung nach einem unbeabsichtigtem Überschreiben oder Löschen. Weitere Informationen finden Sie unter Versioning und Verwenden von Versioning.	8. Februar 2010
Neue unterstützte Region	Amazon S3 unterstützt jetzt die Region USA West (Nordkalifornien). Der neue Endpunkt für Anfragen an diese Region ist `s3-us-west-1.amazonaws.com` . Weitere Informationen finden Sie unter Buckets und Regionen.	2. Dezember 2009

Änderungen	Beschreibung	Datum
AWS SDK for .NET	AWS bietet jetzt Bibliotheken, Beispielcode, Tutorials und andere Ressourcen für Softwareentwickler, die es vorziehen, Anwendungen mithilfe von. NETsprachspezifische API Operationen anstelle von oder. REST SOAP Diese Bibliotheken bieten grundlegende Funktionen (die nicht im REST Oder enthalten sind SOAPAPIs) wie Anforderungsauthentifizierung, erneute Anforderungsversuche und Fehlerbehandlung, sodass der Einstieg erleichtert wird. Weitere Informationen zu sprachspezifischen Bibliotheken und Ressourcen finden Sie unter Entwickeln mit Amazon S3 unter Verwendung von AWS SDKs in der Amazon S3 API S3-Referenz.	11. November 2009

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.