



User Guide

AWS IoT SiteWise



AWS IoT SiteWise: User Guide

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

Die Handelsmarken und die Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist AWS IoT SiteWise?	1
Wie AWS IoT SiteWise funktioniert	1
Investieren Sie Industriedaten	2
Modellieren Sie Ressourcen, um die gesammelten Daten zu kontextualisieren	3
Analysieren Sie mithilfe von Abfragen, Alarmen und Prognosen	3
Visualisieren Sie den Betrieb	4
Daten speichern	4
Integration in andere Services	5
Anwendungsfälle für AWS IoT SiteWise	5
Fertigung	5
Nahrungsmittel und Getränke	5
Energie und Versorgung	6
Arbeiten mit AWS SDKs	6
Konzepte	7
Erste Schritte	14
Voraussetzungen	14
Richten Sie ein AWS Konto	15
Melde dich an für ein AWS-Konto	15
Verwenden Sie die Schnellstart-Demo	16
Erstellen Sie die Demo AWS IoT SiteWise	16
Löschen Sie die AWS IoT SiteWise Demo	18
Lernprogramme	20
OEE berechnen	20
Voraussetzungen	20
Berechnen der OEE	21
Daten aufnehmen	24
Voraussetzungen	25
Schritt 1: Erstellen Sie eine AWS IoT Richtlinie	25
Schritt 2: Erstelle ein AWS IoT Ding	29
Schritt 3: Erstellen Sie ein Geräte-Asset-Modell	34
Schritt 4: Erstellen Sie ein Geräteflotten-Asset-Modell	35
Schritt 5: Erstellen und konfigurieren Sie ein Geräte-Asset	37
Schritt 6: Erstellen und konfigurieren Sie ein Geräteflotten-Asset	38

Schritt 7: Erstellen Sie in AWS IoT Core eine Regel, um Daten an Geräteressourcen zu senden	39
Schritt 8: Führen Sie das Geräteclient-Skript aus	42
Schritt 9: Ressourcen nach dem Tutorial bereinigen	50
Integrieren Sie Daten mit Edge SiteWise	51
Voraussetzungen	53
Schritt 1: Erstellen Sie eine AWS IoT Richtlinie	54
Schritt 2: Erstellen und konfigurieren Sie ein AWS IoT Ding	55
Schritt 3: Konfigurieren Sie Ihr SiteWise Edge-MQTT-fähiges V3-Gateway	56
Schritt 4: Installieren Sie die SiteWise Edge-Gateway-Software	58
Schritt 5: Konfigurieren Sie den EMQX-Broker für die Verbindung zu externen Anwendungen	59
Schritt 6: Veröffentlichen Sie Daten mit Mosquitto	62
Schritt 7: Ziele angeben	64
Schritt 8: Pfadfilter angeben	66
Schritt 9: Konfigurieren Sie Ihre AWS IoT-Ressourcen	70
Schritt 10: Visualisieren Sie Ihre Daten	70
Schritt 11: Ressourcen nach dem Tutorial bereinigen	72
Weitere Ressourcen	75
Daten in Grafana visualisieren und teilen	75
Voraussetzungen	76
Schritt 1: Konfigurieren Sie Ihren Amazon Managed Grafana-Arbeitsbereich	77
Schritt 2: Als Datenquelle hinzufügen AWS IoT SiteWise	79
Schritt 3: Erstellen Sie ein Dashboard, um Ihre Daten zu untersuchen und zu visualisieren ...	80
(optional) Schritt 4: Richten Sie Warnmeldungen ein, um die Leistung zu überwachen	83
Schritt 5: Ressourcen nach dem Tutorial bereinigen	84
Weitere Ressourcen	75
Visualisieren und teilen Sie Daten in SiteWise Monitor	85
Voraussetzungen	86
Schritt 1: Erstellen Sie ein Portal in Monitor SiteWise	87
Schritt 2: Melden Sie sich bei einem Portal an	91
Schritt 3: Erstellen Sie ein Windparkprojekt	93
Schritt 4: Erstellen Sie ein Dashboard zur Visualisierung von Windparkdaten	97
Schritt 5: Erkunden Sie das Portal	104
Schritt 6: Bereinigen Sie die Ressourcen nach dem Tutorial	105
In Amazon DynamoDB veröffentlichen	108

Voraussetzungen	108
Schritt 1: Konfigurieren Sie die Konfiguration AWS IoT SiteWise , um Aktualisierungen von Eigenschaftswerten zu veröffentlichen	109
Schritt 2: Erstellen Sie eine Regel in Core AWS IoT	110
Schritt 3: Konfiguration der DynamoDB-Regelaktion	112
Schritt 4: Erkunden Sie Daten in DynamoDB	112
Bereinigen von Ressourcen	113
Weitere Ressourcen	114
Daten aufnehmen in AWS IoT SiteWise	116
Datenströme verwalten	116
Konfigurieren Sie Berechtigungen und Einstellungen	118
Ordnen Sie einer Anlageneigenschaft einen Datenstrom zu	119
Trennen Sie die Zuordnung eines Datenstroms zu einer Anlageneigenschaft	120
Löschen Sie einen Datenstrom	121
Aktualisieren Sie einen Alias für eine Vermögenseigenschaft	123
Gängige Szenarien	124
Daten aufnehmen mit AWS IoT SiteWise APIs	126
BatchPutAssetPropertyValue API	126
CreateBulkImportJob API	129
Verwenden Sie Regeln AWS IoT Core	138
Gewähren Sie den erforderlichen Zugriff	138
Konfigurieren Sie die Regelaktion	140
Senken Sie die Kosten mit Basic Ingest	149
Verwenden Sie AWS IoT Greengrass den Stream-Manager	150
Verwenden Sie SiteWise Edge-Gateways	151
Die wichtigsten Konzepte von Gateways	151
Vorteile der Implementierung von SiteWise Edge	152
Self-host ein Gateway	153
Voraussetzungen	154
Erstellen Sie ein Gateway	159
Installieren Sie die Gateway-Software	162
MQTT-enabled, V3-Gateways	165
Klassische Streams, V2-Gateways	235
Fügen Sie Datenquellen hinzu	252
Komponenten für SiteWise Edge	301
Ressourcen filtern	302

Proxyunterstützung und Trust Stores	304
Verwenden von APIs	311
Hosten Sie ein Gateway auf Siemens Industrial Edge	329
Sicherheit	330
Siemens Secure Storage und die AWS IoT SiteWise Edge-Anwendung	330
Ziele für Siemens Industrial Edge Geräte	331
Migrieren Sie von der Vorschau-Anwendung	333
Fehlerbehebung	334
AWS IoT SiteWise Changelog der Edge-Anwendung	334
Voraussetzungen	336
Erstellen Sie ein Gateway	336
Erstellen eines Siemens Databus user	338
Greifen Sie auf die Anwendung zu	339
Installieren Sie die Anwendung	339
Aktualisieren Sie eine installierte Anwendungskonfiguration	342
Daten, die durch die Nutzung dieses Dienstes generiert wurden	343
Ziele und Pfadfilter	344
Verstehen Sie Ziele	344
Informationen zu Pfadfiltern	348
Fügen Sie ein Echtzeit-Ziel hinzu	352
Fügen Sie ein gepuffertes Ziel mit Amazon S3 hinzu	361
Hinzufügen von Pfadfiltern	368
Ziele verwalten	371
Gateways verwalten	378
Verwalten Sie Ihr SiteWise Edge-Gateway mit der AWS IoT SiteWise Konsole	379
Verwalten Sie SiteWise Edge-Gateways mit AWS OpsHub AWS IoT SiteWise	379
Greifen Sie mit lokalen Betriebssystemanmeldedaten auf Ihr SiteWise Edge-Gateway zu	381
Das SiteWise Edge-Gateway-Zertifikat verwalten	384
Ändern Sie die Version der SiteWise Edge Gateway-Komponentenpakete	385
SiteWise Edge-Gateways auflisten	385
Beschreiben Sie ein Edge-Gateway SiteWise	386
Erstellen Sie ein Edge-Gateway SiteWise	387
Aktualisieren Sie ein Edge-Gateway SiteWise	388
Aktualisieren Sie die Gateway-Funktionskonfiguration	389
Tag-Gateway-Ressourcen	390
Listet die Tags für ein Gateway auf	391

Tags von einem Gateway entfernen	392
Aktualisieren Sie die Version einer Komponente AWS IoT SiteWise	393
Löschen Sie ein SiteWise Edge-Gateway	393
Gateways sichern und wiederherstellen	395
Tägliche Backups von metrischen Daten	395
Stellen Sie ein SiteWise Edge-Gateway wieder her	396
AWS IoT SiteWise Daten wiederherstellen	397
Bestätigen Sie erfolgreiche Backups und Wiederherstellungen	398
Ältere Gateways ()AWS IoT Greengrass Version 1	399
Modellieren Sie Industrieanlagen	401
Überblick über die Vermögenswerte	401
Eigenschaftsalias identifizieren Gerätedatenströme	401
Anlagenhierarchien stellen Gerätebeziehungen dar	401
Anlagenmodelle standardisieren die Darstellung von Geräten	402
Modellierungsoptionen für Industrieanlagen	402
Anlagen erstellen und verwalten	402
Verwaltung komplexer Anlagenmodelle	403
Komponenten- und Modellzustände	403
Überprüfen Sie den Status eines Assets	404
Überprüfen Sie den Status eines Asset- oder Komponentenmodells	406
Versionen von Asset-Modellen	408
Rufen Sie die aktive Version eines Asset- oder Komponentenmodells (Konsole) ab	409
Rufen Sie die aktive Version eines Asset- oder Komponentenmodells ab (AWS CLI)	410
Benutzerdefinierte zusammengesetzte Modelle (Komponenten)	411
Integrierte benutzerdefinierte Verbundmodelle	412
Component-model-based benutzerdefinierte Verbundmodelle	414
Verwenden Sie Pfade, um auf benutzerdefinierte Eigenschaften von Verbundmodellen zu verweisen	415
Schnittstellen zu Anlagenmodellen	417
Anwendungsfall zur Standardisierung von Anlagenmodellen	418
Struktur und Komponenten	420
Überlegungen	420
Verstehen Sie die Beziehung zwischen Schnittstelle und Asset-Modell	421
Erstellen Sie eine Schnittstelle	426
Wenden Sie eine Schnittstelle auf ein Asset-Modell an	428
Schnittstellen verwalten	429

Weitere Beispiele für Benutzeroberflächen	433
Richten Sie Objekt-IDs ein	437
Arbeiten Sie mit Objekt-UUIDs	437
Verwenden Sie externe IDs	438
Modelle erstellen	440
Erstellen Sie Asset-Modelle inAWS IoT SiteWise	441
Komponentenmodelle erstellen	457
Definieren Sie Dateneigenschaften	461
Erstellen Sie benutzerdefinierte Verbundmodelle (Komponenten)	545
Anlagen erstellen	550
Erstellen Sie eine Anlage (Konsole)	550
Erstellen Sie ein Asset (AWS CLI)	551
Konfigurieren Sie ein neues Asset	553
Suchen Sie nach Vermögenswerten	553
Voraussetzungen	554
Erweiterte Suche auf AWS-IoT-SiteWise-Konsole	554
Attributwerte aktualisieren	557
Vermögenswerte zuordnen und trennen	560
Elemente zuordnen und deren Zuordnung aufheben (Konsole)	560
Vermögenswerte zuordnen und deren Zuordnung aufheben (AWS CLI)	562
Aktualisieren Sie Ressourcen und Modelle	564
Aktualisieren Sie die Anlagen inAWS IoT SiteWise	564
Aktualisieren Sie Objektmodelle, Komponentenmodelle und Schnittstellen	566
Aktualisieren Sie benutzerdefinierte Verbundmodelle (Komponenten)	572
Optimistisches Sperren für Asset-Modell-Schreibvorgänge	576
Löschen Sie Objekte und Modelle inAWS IoT SiteWise	580
Vermögenswerte löschen	580
Löschen Sie Modelle und Schnittstellen	583
Massenoperationen mit Anlagen und Modellen	585
Wichtige Konzepte und Terminologie	586
Unterstützte Funktionen	587
Voraussetzungen für den Massenvorgang	588
Führen Sie einen Massenimportauftrag aus	591
Führen Sie einen Massenexportauftrag aus	593
Verfolgung des Auftragsfortschritts und Fehlerbehandlung	596
Beispiele für den Import von Metadaten	603

Beispiele für den Export von Metadaten	619
Auftragsschema für die Übertragung von Metadaten	621
Überwachen Sie Daten mit Alarmen	641
Arten von Alarmen	641
Alarmzustände	643
Eigenschaften des Alarmstatus	643
Definieren Sie Alarme für Anlagenmodelle	646
Anforderungen für Alarmmeldungen	650
Definieren Sie AWS IoT Events Alarme	650
Definieren Sie externe Alarme	687
Konfigurieren Sie Alarme für Anlagen	689
Konfigurieren Sie einen Schwellenwert (Konsole)	689
Konfigurieren Sie einen Schwellenwert (AWS CLI)	690
Konfigurieren Sie die Benachrichtigungseinstellungen	692
Reagieren Sie auf Alarme	695
Reagieren Sie auf einen Alarm (Konsole)	695
Reagieren Sie auf einen Alarm (API)	699
Erfassen Sie einen externen Alarmstatus	699
Ordnen Sie externe Alarmstatus-Streams zu	700
Daten zum Alarmstatus aufnehmen	702
AWS IoT SiteWise Assistentin	704
Konfigurieren Sie den AWS IoT SiteWise Assistenten	704
Erstellen eines Datensatzes	706
Bearbeiten Sie einen Datensatz	711
Löschen Sie einen Datensatz	713
AWS IoT SiteWise Fragen des Assistenten	714
Überwachen Sie Daten mit AWS IoT SiteWise Monitor	715
SiteWise Rollen überwachen	716
SAML-Verbund	718
SiteWise Konzepte überwachen	719
Fangen Sie mit AWS IoT SiteWise Monitor (Classic) an	721
Erstellen Sie ein Portal	723
Konfigurieren Sie Ihr Portal	724
Laden Sie Administratoren ein	728
Fügen Sie Portalbenutzer hinzu	731
Dashboards erstellen (CLI)	736

Schalten Sie Alarmer für Ihre Portale ein	742
Aktivieren Sie Ihr Portal am Edge	746
Verwalte deine Portale	746
Erste Schritte mit AWS IoT SiteWise Monitor (AI-aware)	758
Erstellen Sie ein Portal	760
Konfigurieren Sie Ihr Portal	761
Verwalte deine Portale	764
Löschen Sie ein Portal	769
Erstellen Sie Dashboards mit AWS CLI	770
Anmeldung zum Portal	775
Erstellen eines Projekts	776
Projekt aktualisieren	777
Projekt löschen	777
Erstellen eines Dashboards	778
Aktualisieren eines Dashboards	780
Löschen eines Dashboards	781
Dashboard konfigurieren	782
Daten abfragen von AWS IoT SiteWise	806
Aktuelle Anlagenwerte abfragen	807
Fragen Sie den aktuellen Wert einer Asset-Eigenschaft ab (Konsole)	807
Fragen Sie den aktuellen Wert einer Anlageeigenschaft ab (AWS CLI)	808
Fragen Sie historische Werte von Vermögenswerten ab	809
Aggregate von Vermögenswerten abfragen	811
Aggregate für eine Anlageneigenschaft (API)	812
Aggregate für eine Anlageimmobilie () AWS CLI	813
AWS IoT SiteWiseSprache abfragen	814
Sprachreferenz abfragen	815
Optimierung von Abfragen	841
Metadatenfilter	841
Rohdatenfilter	842
JOIN-Optimierung	843
Große Abfragen	845
ODBC	845
Syntax der Verbindungszeichenfolge	846
Beispiele für Verbindungszeichenfolgen	851
Fehlerbehebung	852

Interagiere mit anderen Diensten	854
Machen Sie sich mit den Eigenschaften von Assets in MQTT-Themen vertraut	855
Arbeiten Sie mit Benachrichtigungen	855
Aktivieren Sie Benachrichtigungen über Vermögenseigenschaften (Konsole)	856
Aktivieren Sie Benachrichtigungen über Vermögenseigenschaften (AWS CLI)	857
Benachrichtigungen abfragen	859
Daten nach Amazon S3 exportieren	862
Integrieren Sie Grafana	862
Integrieren Sie mit AWS IoT TwinMaker	864
Aktivierung der Integration	864
Integrieren und AWS IoT SiteWise AWS IoT TwinMaker	865
Erkennen Sie Geräteanomalien	866
Fügen Sie eine Vorhersagedefinition hinzu (Konsole)	867
Trainieren Sie eine Vorhersage (Konsole)	871
Starten oder beenden Sie die Inferenz für eine Vorhersage (Konsole)	872
Eine Vorhersagedefinition hinzufügen (CLI)	873
Trainieren Sie eine Vorhersage und starten Sie die Inferenz (CLI)	876
Eine Vorhersage trainieren (CLI)	878
Inferenz auf eine Vorhersage starten oder beenden (CLI)	879
Native Erkennung von Anomalien	882
Systemeigene Funktionen zur Erkennung von Anomalien	883
Voraussetzungen	884
Einrichtung AWS CLI für das Berechnungsmodell APIs	884
Anforderungen an die Immobilie	884
Voraussetzungen für die Kennzeichnung	885
Voraussetzungen für die Modellevaluierung	885
Voraussetzungen für Modellmetriken	886
Aktivieren Sie die Erkennung von Anomalien auf den Sensoren einer Anlage	887
Erstellen Sie ein Berechnungsmodell (AWS CLI)	887
ExecuteAction Vorbereitung der API-Nutzlast	888
Trainiere das Modell (AWS CLI)	889
Startet und beendet das erneute Trainieren des Modells (AWS CLI)	892
Inferenz starten und beenden (AWS CLI)	895
Finden Sie Datenbindungen	898
Ermöglichen Sie die Erkennung von Anomalien an Sensoren in allen Anlagen	899
Erstellen Sie ein Berechnungsmodell (AWS CLI)	899

ExecuteAction Vorbereitung der API-Nutzlast	901
Trainiere das Modell (AWS CLI)	902
Startet und beendet das erneute Trainieren des Modells (AWS CLI)	904
Inferenz starten und beenden (AWS CLI)	905
Erweiterte Trainingskonfigurationen	908
Konfiguration der Samplerate	908
Kennzeichnen Sie Ihre Daten	910
Bewerten Ihres Modells	912
Generieren Sie Modellmetriken	913
Erweiterte Inferenzkonfigurationen	918
Hochfrequente Inferenzierung (5 Minuten — 1 Stunde)	918
Niederfrequenz-Inferenzierung (2 Stunden — 1 Tag)	919
Flexible Planung	920
Aktivierung der Modellversion	922
Überprüfung der Modellversionen	923
Überprüfen Sie die Inferenzergebnisse	924
Inferenzergebnisse abrufen	924
Verstehen Sie die Inferenzergebnisse	925
Löst bei anomalem Verhalten benutzerdefinierte Aktionen aus (AWS-Managementkonsole)	926
Best Practices	927
Machen Sie sich mit dem Mindestzeitraum vertraut	927
Stichprobenerhebung für hochfrequente Daten und Konsistenz zwischen Training und Inferenz	928
Empfehlungen zur Kennzeichnung	928
Verwalten des Datenspeichers	930
Konfigurieren Sie die Speichereinstellungen	931
Auswirkungen auf die Datenspeicherung	932
Konfigurieren Sie für Warm Tier (Konsole)	932
Für warme Stufe (AWS CLI) konfigurieren (.....	934
Konfigurieren Sie für Cold-Tier (Konsole)	937
Für Cold Tier (AWS CLI) konfigurieren	940
Beheben Sie Fehler bei den Speichereinstellungen	946
Fehler: Bucket ist nicht vorhanden	946
Fehler: Zugriff auf den Amazon S3-Pfad verweigert	946
Fehler: Rollen-ARN kann nicht übernommen werden	947

Fehler: Auf den regionsübergreifenden Amazon S3 S3-Bucket konnte nicht zugegriffen werden	947
Dateipfade und Schemas von Daten, die auf der kalten Ebene gespeichert wurden	947
Gerätedaten (Messungen)	948
Metriken, Transformationen und Aggregationen	952
Asset-Metadaten	957
Metadaten der Asset-Hierarchie	962
Speicherdaten, Indexdateien	964
Codebeispiele	965
Grundlagen	966
Hallo AWS IoT SiteWise	966
Kennenlernen der Grundlagen	971
Aktionen	1032
Sicherheit	1100
Datenschutz	1101
Richtlinie für den Datenverkehr zwischen Netzwerken	1102
AWS IoT SiteWiseAssistentin bei der Verbesserung des Business Services	1102
Datenverschlüsselung	1102
Verschlüsselung im Ruhezustand	1103
Verschlüsselung während der Übertragung	1106
Schlüsselverwaltung	1108
Identity and Access Management	1110
Zielgruppe	1111
Authentifizieren mit Identitäten	1111
WieAWS IoT SiteWisefunktioniert mit IAM	1112
Verwaltete Richtlinien	1132
Service-linked Rollen	1140
Richten Sie Berechtigungen für Alarmer ein	1162
Cross-service verwirrter Stellvertreter, Prävention inAWS IoT SiteWise	1169
Fehlerbehebung für -Identität und -Zugriff	1170
Compliance-Validierung	1172
Ausfallsicherheit	1174
Sicherheit der Infrastruktur	1174
Konfigurations- und Schwachstellenanalyse	1175
VPC-Endpunkte	1176
Unterstützte -API-Operationen	1176

Erstellen eines Schnittstellen-VPC-Endpunkts	1179
ZugriffAWS IoT SiteWiseüber einen VPC-Endpunkt mit Schnittstelle	1179
Erstellen einer VPC-Endpunktrichtlinie	1181
Bewährte Methoden für die Gewährleistung der Sicherheit	1182
Verwenden Sie Authentifizierungsdaten auf Ihren OPC UA-Servern	1182
Verwenden Sie verschlüsselte Kommunikationsmodi für Ihre OPC UA-Server	1183
Halten Sie Ihre Komponenten auf dem neuesten Stand	1183
Verschlüsseln Sie das SiteWise Dateisystem Ihres Edge-Gateways	1183
Sicherer Zugriff auf Ihre Edge-Konfiguration	1184
Daten sichern auf Industrielles Edge-Management von Siemens	1184
Gewähren SiteWise Sie Monitor-Benutzern die geringstmöglichen Berechtigungen	1184
Legen Sie vertrauliche Informationen nicht offen	1185
FollowAWS IoT GreengrassBewährte Methoden für die Sicherheit	1185
Weitere Informationen finden Sie auch unter	1185
Protokollieren und Überwachen	1186
Überwachen Sie Serviceprotokolle	1187
Die Anmeldung verwalten AWS IoT SiteWise	1188
Beispiel: Einträge in der AWS IoT SiteWise Protokolldatei	1190
SiteWise Edge-Gateway-Protokolle überwachen	1190
Verwenden Sie Amazon CloudWatch Logs	1191
Verwenden Sie Serviceprotokolle	1192
Überwachen Sie mit CloudWatch Amazon-Metriken	1194
AWS IoT Greengrass Version 2 Gateway-Metriken	1195
API-Aufrufe protokollieren mit AWS CloudTrail	1204
AWS IoT SiteWise Informationen in CloudTrail	1205
AWS IoT SiteWise Datenereignisse in CloudTrail	1206
AWS IoT SiteWise Verwaltungsereignisse in CloudTrail	1209
Beispiel: Einträge in AWS IoT SiteWise Protokolldateien	1209
Markieren Ihrer -Ressourcen mit Tags (Markierungen)	1211
Verwenden Sie Tags in AWS IoT SiteWise	1211
Tag mit dem AWS-Managementkonsole	1211
Taggen Sie mit der API AWS IoT SiteWise	1212
Verwenden Sie Tags mit IAM-Richtlinien	1213
Fehlerbehebung	1215
Fehlerbehebung für ein Gateway	1215
SiteWise Edge-Gateway-Protokolle konfigurieren und darauf zugreifen	1215

Behebung von Problemen mit dem SiteWise Edge-Gateway	1216
Fehlerbehebung bei der AWS IoT SiteWise Edge-Anwendung auf Siemens Industrial Edge	1223
Fehlerbehebung bei Open-Source-Integrationen am Edge	1224
Probleme beheben AWS IoT Greengrass	1226
Problembehandlung bei einem Portal	1227
Benutzer und Administratoren können nicht auf das AWS IoT SiteWise Portal zugreifen ...	1227
Problembehandlung bei einer AWS IoT SiteWise Regelaktion	1228
AWS IoT Core Protokolle konfigurieren	1229
Konfigurieren Sie eine Aktion zum erneuten Veröffentlichen von Fehlern	1230
Beheben Sie Regelprobleme	1232
Problembehandlung bei einer Regel ()AWS IoT SiteWise	1234
Problembehandlung bei einer Regel (DynamoDB)	1236
Problembehandlung beim Massenimport und -export	1239
Endpunkte und Kontingente	1241
Endpunkte	1241
Kontingente	1241
Kontingente für AWS IoT SiteWise Vermögenswerte und Anlagemodelle	1241
Schnittstellenkontingente	1247
Kontingente für Daten zu AWS IoT SiteWise Vermögenswerten	1248
Kontingente für SiteWise Edge-Gateways	1256
Kontingente für AWS IoT SiteWise Monitor	1257
Kontingente für den AWS IoT SiteWise Massenimport und -export von Metadaten	1259
Kontingente für den AWS IoT SiteWise Massenimport von Daten	1260
AWS IoT SiteWise Grenzwerte für die Drosselung der Assistant-API	1262
Kontingente für die Erkennung von Anomalien	1263
Kontingente für SQL	1264
Dokumentverlauf	1266
.....	mccxcix

Was ist AWS IoT SiteWise?

AWS IoT SiteWise ist ein verwalteter Service, mit dem Sie Daten von Industrieanlagen in großem Umfang sammeln, speichern, organisieren und überwachen können, damit Sie bessere, datengestützte Entscheidungen treffen können. Sie können ihn verwenden, AWS IoT SiteWise um Betriebsabläufe in allen Anlagen zu überwachen, schnell allgemeine industrielle Leistungskennzahlen zu berechnen und Anwendungen zu entwickeln, die Daten zu Industrieanlagen analysieren, um kostspielige Geräteprobleme zu vermeiden und Produktionslücken zu schließen.

Mit können Ihre operativen Benutzer Webanwendungen erstellen AWS IoT SiteWise Monitor, mit denen Sie Ihre Industriedaten in Echtzeit anzeigen und analysieren können. Sie erhalten Einblicke in Ihre industriellen Operationen, indem Sie Metriken wie z. B. die mittlere störungsfreie Zeit und die Gesamtanlageneffektivität (Overall Equipment Effectiveness, OEE) konfigurieren und überwachen.

AWS IoT SiteWise Edge ist eine Komponente AWS IoT SiteWise, die die Erfassung, Speicherung und Verarbeitung von Daten auf lokalen Geräten ermöglicht. Dies ist nützlich, wenn Sie nur eingeschränkten Zugang zum Internet haben oder Ihre Daten privat halten müssen.

Themen

- [Wie AWS IoT SiteWise funktioniert](#)
- [Anwendungsfälle für AWS IoT SiteWise](#)
- [Verwenden Sie diesen Service mit einem AWS SDK](#)
- [AWS IoT SiteWise Konzepte](#)

Wie AWS IoT SiteWise funktioniert

AWS IoT SiteWise bietet ein Framework zur Ressourcenmodellierung, mit dem Sie Darstellungen Ihrer industriellen Geräte, Prozesse und Anlagen erstellen können. Die Darstellungen Ihrer Geräte und Prozesse werden in als Anlagenmodelle bezeichnet AWS IoT SiteWise. Mit Anlagenmodellen definieren Sie, welche Rohdaten verwendet werden sollen, und wie diese zu nützlichen Kennzahlen verarbeitet werden. Erstellen und visualisieren Sie Anlagen und Modelle für Ihren industriellen Betrieb in der [AWS IoT SiteWise Konsole](#). Sie können Anlagenmodelle auch so konfigurieren, dass sie Daten am Netzwerkrand oder in der AWS Cloud sammeln und verarbeiten.

Themen

- [Investieren Sie Industriedaten](#)

- [Modellieren Sie Ressourcen, um die gesammelten Daten zu kontextualisieren](#)
- [Analysieren Sie mithilfe von Abfragen, Alarmen und Prognosen](#)
- [Visualisieren Sie den Betrieb](#)
- [Daten speichern](#)
- [Integration in andere Services](#)

Investieren Sie Industriedaten

Beginnen Sie mit der Nutzung, AWS IoT SiteWise indem Sie Industriedaten aufnehmen. Die Erfassung Ihrer Daten erfolgt auf eine von mehreren Arten:

- Direkte Erfassung von Servern vor Ort: Verwenden Sie Protokolle wie OPC UA, um Daten direkt von Geräten vor Ort zu lesen. Stellen Sie die SiteWise Edge-Gateway-Software, kompatibel mit AWS IoT Greengrass V2, auf einer Vielzahl von Plattformen wie gängigen industriellen Gateways oder virtuellen Servern bereit. Sie können bis zu 100 OPC UA-Server mit einem einzigen AWS IoT SiteWise Gateway verbinden. Weitere Informationen finden Sie unter [AWS IoT SiteWise Anforderungen an das selbst gehostete Edge-Gateway](#).

Beachten Sie, dass Protokolle wie Modbus TCP und Ethernet/IP (EIP) durch unsere Partnerschaft mit Domatica im Rahmen von unterstützt werden. AWS IoT Greengrass V2

- Edge-Datenverarbeitung mit Paketen: Erweitern Sie Ihr SiteWise Edge-Gateway, indem Sie Pakete hinzufügen, um umfassende Edge-Funktionen zu ermöglichen. Mit SiteWise Edge, verfügbar auf AWS IoT Greengrass V2, wird die Datenverarbeitung direkt vor Ort ausgeführt, bevor sie mithilfe eines AWS IoT Greengrass Streams sicher in die AWS Cloud übertragen wird. Weitere Informationen finden Sie unter [Richten Sie eine OPC UA-Quelle in SiteWise Edge ein](#).
- Adaptive Erfassung über Amazon S3 mit Massenoperationen: Wenn Sie mit einer großen Anzahl von Assets oder Asset-Modellen arbeiten, verwenden Sie Massenoperationen, um Ressourcen massenweise aus Amazon S3 S3-Buckets zu importieren und zu exportieren. Weitere Informationen finden Sie unter [Massenoperationen mit Anlagen und Modellen](#).
- MQTT-Nachrichten mit AWS IoT Kernregeln: Verwenden Sie für Geräte, die mit AWS IoT Core verbunden sind und MQTT-Nachrichten senden, die AWS IoT Core-Regel-Engine, um diese Nachrichten weiterzuleiten. AWS IoT SiteWise Wenn Sie mit Core verbundene Geräte haben, die [MQTT-Nachrichten](#) senden, verwenden Sie die AWS IoT Core-Regel-Engine, um diese Nachrichten weiterzuleiten. AWS IoT SiteWise Weitere Informationen finden Sie unter [Daten AWS IoT SiteWise mithilfe AWS IoT Core von Regeln aufnehmen](#).

- AWS IoT SiteWiseAPI: Ihre Anwendungen am Edge oder in der Cloud können Daten direkt an senden. AWS IoT SiteWise Weitere Informationen finden Sie unter [Daten aufnehmen mit AWS IoT SiteWise APIs](#).

Modellieren Sie Ressourcen, um die gesammelten Daten zu kontextualisieren

Nach der Datenaufnahme können Sie anhand der Daten virtuelle Repräsentationen Ihrer Anlagen, Prozesse und Einrichtungen erstellen, indem Sie Modelle Ihrer physischen Abläufe erstellen. Ein Asset, das ein Gerät oder einen Prozess darstellt, überträgt Datenströme in die AWS Cloud. Vermögenswerte können auch logische Gerätegruppierungen bedeuten. Hierarchien werden durch die Zuordnung von Ressourcen gebildet, um komplexe Abläufe widerzuspiegeln. Diese Hierarchien ermöglichen es Anlagen, auf Daten aus zugehörigen untergeordneten Anlagen zuzugreifen. Vermögenswerte werden anhand von Anlagenmodellen erstellt. Asset-Modelle sind deklarative Strukturen, die Asset-Formate standardisieren. Verwenden Sie Komponenten von Assets für die Organisation und Wartbarkeit Ihrer Modelle wieder. Weitere Informationen finden Sie unter [Modellieren Sie Industrieanlagen](#).

Mit können Sie Ihre Ressourcen so konfigurierenAWS IoT SiteWise, dass die eingehenden Daten in kontextbezogene Metriken und Transformationen umgewandelt werden.

- Transformiert die Arbeit beim Empfang von Gerätedaten.
- Metriken werden in von Ihnen definierten Intervallen berechnet.

Metriken und Transformationen gelten sowohl für einzelne Anlagen als auch für mehrere Anlagen. AWS IoT SiteWiseberechnet automatisch häufig verwendete statistische Aggregate wie Durchschnitt, Summe und Anzahl über verschiedene Zeiträume, die für Ihre Gerätedaten, Kennzahlen und Transformationen relevant sind.

Anlagen können synchronisiert werden mit. AWS IoT TwinMaker Weitere Informationen finden Sie unter [Integrieren und AWS IoT SiteWise AWS IoT TwinMaker](#).

Analysieren Sie mithilfe von Abfragen, Alarmen und Prognosen

Analysieren Sie das gesammelte Datum, AWS IoT SiteWise indem Sie Abfragen ausführen und Alarme einrichten. Sie können Amazon Lookout auch verwenden, um Anomalien innerhalb von Kennzahlen automatisch zu erkennen und deren Ursachen zu identifizieren.

- Richten Sie spezifische Alarmer ein, um Ihr Team zu benachrichtigen, wenn Geräte oder Prozesse von der optimalen Leistung abweichen, und sorgen Sie so für eine schnelle Identifizierung und Lösung von Problemen. Weitere Informationen finden Sie unter [Überwachen Sie Daten mit AlarmerAWS IoT SiteWise](#).
- Verwenden Sie die AWS IoT SiteWise API-Operationen, um die aktuellen Werte, historischen Werte und Aggregate Ihrer Anlageneigenschaften über bestimmte Zeitintervalle abzufragen. Weitere Informationen finden Sie unter [Daten abfragen vonAWS IoT SiteWise](#).
- Verwenden Sie die Anomalieerkennung mit Amazon Lookout for Equipment, um Änderungen an Geräten oder Betriebsbedingungen zu identifizieren und zu visualisieren. Mit der Erkennung von Anomalien können Sie vorbeugende Wartungsmaßnahmen für Ihren Betrieb festlegen. Diese Integration ermöglicht es Kunden, Daten zwischen Amazon Lookout for Equipment AWS IoT SiteWise und Amazon Lookout for Equipment zu synchronisieren. Weitere Informationen finden Sie unter [Erkennen Sie Anomalien mit Lookout for Equipment](#).

Visualisieren Sie den Betrieb

Richten Sie SiteWise Monitor ein, um Webanwendungen für Ihre operativen Mitarbeiter zu erstellen. Die Webanwendungen helfen den Mitarbeitern, Ihre Abläufe zu visualisieren. Verwalten Sie verschiedene Zugriffsebenen für Ihre Mitarbeiter mithilfe von IAM Identity Center oder IAM. Konfigurieren Sie individuelle Logins und Berechtigungen für jeden Mitarbeiter, um bestimmte Teilbereiche eines gesamten Industriebetriebs einzusehen. AWS IoT SiteWise stellt diesen Mitarbeitern einen [Anwendungsleitfaden](#) zur Verfügung, in dem sie lernen, wie sie Monitor verwenden können SiteWise .

Weitere Informationen zur Visualisierung Ihrer Abläufe finden Sie unter [Überwachen Sie Daten mit AWS IoT SiteWise Monitor](#).

Daten speichern

Sie können Zeitreihenspeicher in Ihren industriellen Data Lake integrieren. AWS IoT SiteWise verfügt über drei Speicherebenen für industrielle Daten:

- Eine Hot-Storage-Tier, die für Echtzeitanwendungen optimiert ist.
- Eine warme Speicherebene, die für analytische Workloads optimiert ist.
- Ein vom Kunden verwaltetes Kühltier, das Amazon S3 für Betriebsdaten anwendungen mit hoher Latenztoleranz verwendet.

AWS IoT SiteWise hilft Ihnen bei der Verwaltung der Speicherkosten, indem aktuelle Daten in der Hot-Storage-Tier aufbewahrt werden. Anschließend definieren Sie Richtlinien zur Datenspeicherung, um historische Daten in Speicher mit warmer oder kalter Speicherebene zu verschieben. Weitere Informationen finden Sie unter [Datenspeicher verwalten in AWS IoT SiteWise](#).

Sie können auch Asset-Metadaten importieren und exportieren. Weitere Informationen finden Sie unter [Asset-Metadaten](#).

Integration in andere Services

AWS IoT SiteWise lässt sich in mehrere AWS Dienste integrieren, um eine AWS IoT Komplettlösung in der AWS Cloud zu entwickeln. Weitere Informationen finden Sie unter [Interagiere mit anderen AWS Diensten](#).

Anwendungsfälle für AWS IoT SiteWise

AWS IoT SiteWise wird in einer Vielzahl von Branchen für viele industrielle Datenerfassungs- und Analyseanwendungen eingesetzt.

Sammeln Sie konsistent Daten aus all Ihren Quellen, um Probleme schnell zu lösen. AWS IoT SiteWise bietet Fernüberwachung, um die Daten direkt vor Ort oder aus mehreren Quellen in vielen Einrichtungen zu sammeln. AWS IoT SiteWise bietet die notwendige Flexibilität für industrielle IoT-Datenlösungen.

Fertigung

AWS IoT SiteWise kann den Prozess der Erfassung und Nutzung von Daten aus Ihren Geräten vereinfachen, um Ineffizienzen zu lokalisieren und zu minimieren und so den industriellen Betrieb zu verbessern. AWS IoT SiteWise hilft Ihnen bei der Erfassung von Daten aus Fertigungslinien und Anlagen. Mit AWS IoT SiteWise können Sie die Daten in die AWS Cloud übertragen und Leistungskennzahlen für Ihre spezifischen Geräte und Prozesse erstellen. Sie können die erstellten Kennzahlen verwenden, um die Gesamteffektivität Ihres Betriebs zu verstehen und Innovations- und Verbesserungsmöglichkeiten zu identifizieren. Sie können sich auch Ihren Herstellungsprozess ansehen und Geräte- und Prozessmängel, Produktionslücken oder Produktfehler identifizieren.

Nahrungsmittel und Getränke

Anlagen in der Nahrungsmittel- und Getränkeindustrie verarbeiten eine große Bandbreite von Lebensmitteln. So mahlen sie zum Beispiel Getreide zu Mehl, schneiden und verpacken Fleisch

und erstellen, kochen und gefrieren für die Erwärmung in Mikrowellen geeignete Mahlzeiten. Lebensmittelverarbeitungsanlagen erstrecken sich häufig über mehrere Standorte, wobei sich die Anlagen- und Gerätebediener an einem zentralen Standort befinden, um Prozesse und Ausrüstung zu überwachen. Kühlaggregate bewerten beispielsweise die Handhabung und das Verfallsdatum der Zutaten. Sie überwachen das Abfallaufkommen in allen Anlagen, um die betriebliche Effizienz sicherzustellen. Mit AWS IoT SiteWise können Sie Sensordatenströme von mehreren Standorten nach Produktionslinie und Anlage gruppieren, sodass Ihre Verfahrenstechniker die Anlagen besser verstehen und Verbesserungen vornehmen können.

Energie und Versorgung

Mit AWS IoT SiteWise können Sie Geräteprobleme einfacher und effizienter lösen. Sie können die Leistung Ihrer Anlagen aus der Ferne und in Echtzeit überwachen. Greifen Sie von überall auf historische Gerätedaten zu, um potenzielle Probleme zu lokalisieren, präzise Ressourcen bereitzustellen und Probleme schneller zu verhindern und zu beheben.

Verwenden Sie diesen Service mit einem AWS SDK

AWS Software Development Kits (SDKs) sind für viele beliebte Programmiersprachen verfügbar. Jedes SDK bietet eine API, Codebeispiele und Dokumentation, die es Entwicklern erleichtern, Anwendungen in ihrer bevorzugten Sprache zu erstellen.

SDK-Dokumentation	Codebeispiele
AWS SDK für C++	AWS SDK für C++ Code-Beispiele
AWS CLI	AWS CLI Codebeispiele
AWS SDK für Go	AWS SDK für Go Codebeispiele
AWS SDK für Java	AWS SDK für Java Codebeispiele
AWS SDK für JavaScript	AWS SDK für JavaScript Codebeispiele
AWS SDK für Kotlin	AWS SDK für Kotlin Codebeispiele
AWS SDK für .NET	AWS SDK für .NET Codebeispiele
AWS SDK für PHP	AWS SDK für PHP Codebeispiele

SDK-Dokumentation	Codebeispiele
AWS -Tools für PowerShell	AWS -Tools für PowerShellCodebeispiele
AWS SDK für Python (Boto3)	AWS SDK für Python (Boto3)Codebeispiele
AWS SDK für Ruby	AWS SDK für RubyCodebeispiele
AWS SDK für Rust	AWS SDK für RustCodebeispiele
AWS SDK für SAP ABAP	AWS SDK für SAP ABAPCodebeispiele
AWS SDK für Swift	AWS SDK für SwiftCodebeispiele

Beispiel für die Verfügbarkeit

Sie können nicht finden, was Sie brauchen? Fordern Sie ein Codebeispiel an, indem Sie unten den Link Provide feedback (Feedback geben) auswählen.

AWS IoT SiteWise Konzepte

Im Folgenden sind die Kernkonzepte von aufgeführt AWS IoT SiteWise:

Aggregate

Aggregate sind grundlegende Metriken oder Messungen, die AWS IoT SiteWise automatisch für alle Zeitreihendaten berechnet werden. Weitere Informationen finden Sie unter [Abfragen von Asset-Eigenschaftenaggregaten inAWS IoT SiteWise](#).

Komponente

Wenn Sie Daten AWS IoT SiteWise aus Ihren Industrieanlagen eingeben oder aufnehmen, werden Ihre Geräte, Anlagen und Prozesse jeweils als Anlagen angezeigt. Jedem Asset sind Daten zugeordnet. Beispielsweise kann ein Gerät eine Seriennummer, einen Standort, eine Marke und ein Modell sowie ein Installationsdatum haben. Es kann auch Zeitreihenwerte für Verfügbarkeit, Leistung, Qualität, Temperatur, Druck und mehr enthalten. Gruppieren Sie Ressourcen in Hierarchien, sodass sie auf Daten zugreifen können, die in ihren

untergeordneten Anlagen gespeichert sind. Weitere Informationen finden Sie unter [Modellieren Sie Industrieanlagen](#).

Komponentenhierarchie

Richten Sie Anlagenhierarchien ein, um logische Darstellungen Ihrer industriellen Abläufe zu erstellen. Definieren Sie dazu eine Hierarchie in einem Anlagenmodell und ordnen Sie anhand dieses Modells erstellte Anlagen der angegebenen Hierarchie zu. Bei Kennzahlen in übergeordneten Anlagen können Daten aus den Eigenschaften untergeordneter Anlagen kombiniert werden. Auf diese Weise können Sie Kennzahlen berechnen, die Einblicke in Ihren gesamten Betrieb oder einen bestimmten Teil davon bieten. Weitere Informationen finden Sie unter [Definieren Sie die Hierarchien der Anlagenmodelle](#).

Komponentenmodell

Jede Anlage wird anhand eines Anlagenmodells erstellt. Vermögensmodelle sind Strukturen, die das Format Ihrer Vermögenswerte definieren und standardisieren. Sie sorgen für konsistente Informationen über mehrere Anlagen desselben Typs hinweg, sodass Sie Daten in Anlagen verwalten können, die Gerätegruppen repräsentieren. In jedem Komponentenmodell können Sie [Attribute](#), Zeitreiheneingaben ([Messungen](#)), Zeitreihentransformationen ([Transformationen](#)), Zeitreihenaggregationen ([Metriken](#)) und [Komponentenhierarchien](#) definieren. Weitere Informationen finden Sie unter [Modellieren Sie Industrieanlagen](#).

Entscheiden Sie, wo die Eigenschaften Ihres Asset-Modells verarbeitet werden, indem Sie Ihr Asset-Modell für den Edge-Bereich konfigurieren. Verwenden Sie diese Funktion, um Asset-Daten auf Ihren lokalen Geräten zu verwalten und zu überwachen.

Komponenteneigenschaft

Bei Anlageneigenschaften handelt es sich um die Strukturen innerhalb jeder Anlage, die industrielle Daten enthalten. Jede Immobilie hat einen Datentyp und kann auch eine Einheit haben. Eine Eigenschaft kann ein [Attribut](#), eine [Messung](#), eine [Transformation](#) oder eine [Metrik](#) sein. Weitere Informationen finden Sie unter [Definieren Sie Dateneigenschaften](#).

Konfigurieren Sie die Asset-Eigenschaften für die Berechnung am Edge. Weitere Informationen zur Verarbeitung von Daten am Netzwerkrand finden Sie unter [Richten Sie eine OPC UA-Quelle in SiteWise Edge ein](#).

Attribut

Bei Attributen handelt es sich um Eigenschaften eines Assets, die in der Regel konstant bleiben, z. B. der Gerätehersteller oder der Gerätestandort. Attribute können voreingestellte Werte haben.

Jedes aus einem Asset-Modell erstellte Asset enthält die Standardwerte der in diesem Modell definierten Attribute. Weitere Informationen finden Sie unter [Definieren Sie statische Daten \(Attribute\)](#).

Berechnungsmodell

A `ComputationModel` ist eine Abstraktion für bestimmte Arten von Berechnungen, die auf Ihre Daten angewendet werden können. Sie definiert den Entwurf für eine Reihe von Berechnungen, in denen Benutzer Eingabe, Ausgabe und Konfiguration für eine bestimmte Rechenengine beschreiben können. `ComputationModel` ist eine neue Ressource mit ARN und ist statusbehaftet und versioniert. Weitere Informationen finden Sie unter [Erstellen Sie ein Berechnungsmodell \(AWS CLI\)](#).

Dashboard

Jedes Projekt enthält eine Reihe von Dashboards. Dashboards stellen eine Reihe von Visualisierungen für die Werte einer Gruppe von Komponenten bereit. Projekteigentümer erstellen die Dashboards und die darin enthaltenen Visualisierungen. Wenn ein Projekteigentümer bereit ist, die Gruppe von Dashboards freizugeben, kann der Eigentümer Betrachter zu dem Projekt einladen, wodurch diese Zugriff auf alle Dashboards in dem Projekt erhalten. Wenn Sie eine andere Gruppe von Betrachtern für verschiedene Dashboards wünschen, müssen Sie die Dashboards auf Projekte aufteilen. Wenn sich Zuschauer Dashboards ansehen, können sie den Zeitraum so anpassen, dass sie sich bestimmte Daten ansehen.

Datensatz

Datensätze sind Datensammlungen, die Zeitreihendaten, Daten und non-time-series Daten, die nicht zur Ausrüstung gehören, wie Schichtpläne, Wartungsaufzeichnungen und Mitarbeiterdatenbanken, darstellen. Sie unterstützen externe Daten und nutzen AWS IoT SiteWise Analysefunktionen. Es umfasst Datensatzquellen, Datensatzschema und Datensatzparameter. Der AWS IoT SiteWise Assistent verwendet Datensätze, die Amazon Kendra Kendra-Indizes verwenden.

Datenstrom

Geben Sie Industriedaten ein oder nehmen Sie sie auf, AWS IoT SiteWise noch bevor Sie Anlagenmodelle und Anlagen erstellen. AWS IoT SiteWise generiert automatisch Datenströme, um Rohdatenströme von Ihren Geräten zu sammeln.

Alias für Datenströme

Datenstream-Aliase helfen Ihnen dabei, einen Datenstrom einfach zu identifizieren. Der Alias `server1-windfarm/3/turbine/7/temperature` gibt beispielsweise Temperaturwerte an,

die von Turbine #7 im Windpark #3 stammen. Der Begriff `server1` ist der Name der Datenquelle, anhand dessen der OPC UA-Server identifiziert werden kann. Er `server1-` ist ein Präfix, das allen von diesem OPC UA-Server gemeldeten Datenströmen zugewiesen wird.

Zuordnung von Datenströmen

Nachdem Sie Asset-Modelle und Assets erstellt haben, verknüpfen Sie Datenstreams mit den in Ihren Assets definierten Asset-Eigenschaften, um Ihre Daten zu strukturieren. AWS IoT SiteWise kann dann Asset-Modelle und Assets verwenden, um eingehende Daten aus Ihren Datenströmen zu verarbeiten. Sie können Datenströme auch von Asset-Eigenschaften trennen. Weitere Informationen finden Sie unter [Datenströme verwalten für AWS IoT SiteWise](#).

Ziele

Ziele in SiteWise Edge stellen die Endpunkte dar, an die Sie Ihre Telemetrie oder verarbeiteten Daten senden möchten. SiteWise Edge unterstützt AWS IoT SiteWise Hot-Tier, Buffered Ingestion oder einen Amazon S3 S3-Bucket als Ziele. Sie können Ziele so konfigurieren, dass sie bestimmte MQTT-Themen mithilfe von Pfadfiltern abonnieren. Weitere Informationen finden Sie unter [Verstehen Sie AWS IoT SiteWise Edge-Ziele](#).

Formel

Jede [Transformations](#) - und [Metrikeigenschaft](#) enthält eine Formel, die beschreibt, wie die Eigenschaft Daten transformiert oder aggregiert. Diese Formeln beinhalten Eigenschaftseingaben, Operatoren und Funktionen, die von angeboten werden. AWS IoT SiteWise Weitere Informationen finden Sie unter [Verwenden Sie Formelausdrücke](#).

Schnittstelle

Eine Schnittstelle ist ein Modelltyp, der eine Standardstruktur definiert, die auf verschiedene Asset-Modelle angewendet werden kann. Weitere Informationen finden Sie unter [Schnittstellen für das Asset-Modell](#).

Messung

Messungen sind Eigenschaften einer Anlage, die die rohen Sensorzeitreihendatenströme von einem Gerät oder einer Ausrüstung darstellen. Weitere Informationen finden Sie unter [Definieren Sie Datenströme von Geräten \(Messungen\)](#).

Metrik

Metriken sind Eigenschaften eines Assets, die aggregierte Zeitreihendaten darstellen. Jede Metrik wird von einem mathematischen Ausdruck ([Formel](#)) begleitet, der beschreibt, wie Datenpunkte

aggregiert werden, und ein Zeitintervall für die Berechnung dieser Aggregation. Metriken generieren einen einzelnen Datenpunkt für jedes angegebene Zeitintervall. Weitere Informationen finden Sie unter [Aggregieren Sie Daten aus Immobilien und anderen Vermögenswerten \(Metriken\)](#).

MQTT

MQTT (Message Queuing Telemetry Transport) ist ein einfaches Messaging-Protokoll für Sensoren und Geräte.

Pakete

SiteWise Edge-Gateways verwenden Pakete, um zu bestimmen, wie Daten gesammelt, verarbeitet und weitergeleitet werden. Weitere Informationen zu den verfügbaren Paketen für Ihr SiteWise Edge-Gateway finden Sie unter [the section called “Verwenden von Paketen”](#).

Datenerfassungspaket

Verwenden Sie das Datenerfassungspaket, damit Ihr SiteWise Edge-Gateway Ihre Industriedaten sammeln und an das AWS Ziel Ihrer Wahl weiterleiten kann.

Datenverarbeitungspaket

Verwenden Sie das Datenverarbeitungspaket, um Ihre Daten bis zu 30 Tage lang am Netzwerkrand zu verarbeiten, zu speichern und abzurufen. Tauschen Sie Edge-verarbeitete Daten über SiteWise Edge zu und von lokalen Anwendungen aus. APIs

OPC UA

OPC UA (Open Platform Communications Unified Architecture) ist ein Kommunikationsprotokoll für die industrielle Automatisierung.

Pfadfilter

Verwenden Sie Pfadfilter innerhalb eines Gateways, um MQTT-Themen zu abonnieren und an AWS IoT SiteWise unterstützten Zielen zu veröffentlichen. MQTT-basierte Quellen, Datenverarbeitungspipelines und Ziele tauschen alle Daten mithilfe von MQTT-Themen auf einem selbst gehosteten, MQTT-fähigen V3-Gateway aus. Sie können Themenfilter definieren, um die Daten zu spezifizieren, die Sie aufnehmen oder an verschiedene Ziele weiterleiten möchten.

Portal

Ein AWS IoT SiteWise Monitor Portal ist eine Webanwendung, mit der Sie Ihre AWS IoT SiteWise Daten visualisieren und teilen können. Ein Portal verfügt über einen oder mehrere Administratoren und enthält keine oder mehrere Projekte.

Portaladministrator

Jedes SiteWise Monitor-Portal hat einen oder mehrere Portaladministratoren.

Portaladministratoren verwenden das Portal, um Projekte zu erstellen, die Sammlungen von Komponenten und Dashboards enthalten. Der Portaladministrator weist dann jedem Projekt Komponenten und Eigentümer zu. Durch die Steuerung des Zugriffs auf das Projekt legen Portaladministratoren fest, welche Komponenten von Projekteigentümern und -betrachtern angezeigt werden können.

Projekt

Jedes SiteWise Monitor-Portal enthält eine Reihe von Projekten. Jedem Projekt ist eine Teilmenge Ihrer AWS IoT SiteWise -Komponenten zugeordnet. Projekteigentümer erstellen ein oder mehrere Dashboards, um eine konsistente Möglichkeit zum Anzeigen der mit diesen Komponenten verknüpften Daten bereitzustellen. Projekteigentümer können Betrachter zu dem Projekt einladen, damit diese die Komponenten und Dashboards in dem Projekt anzeigen können. Das Projekt ist die grundlegende Einheit für die gemeinsame Nutzung innerhalb von SiteWise Monitor. Projekteigentümer können Benutzer einladen, denen der AWS Administrator Zugriff auf das Portal gewährt hat. Ein Benutzer muss Zugriff auf ein Portal haben, bevor ein Projekt in diesem Portal für diesen Benutzer freigegeben werden kann.

Projekteigentümer

Jedes SiteWise Monitor-Projekt hat Besitzer. Projekteigentümer erstellen Visualisierungen in Form von Dashboards, um Betriebsdaten konsistent darzustellen. Wenn Dashboards zur Freigabe bereit sind, kann der Projekteigentümer Betrachter zu dem Projekt einladen. Projekteigentümer können dem Projekt auch andere Eigentümer zuweisen. Projekteigentümer können Schwellenwerte und Benachrichtigungseinstellungen für Alarmer konfigurieren.

Projektbetrachter

Jedes SiteWise Monitor-Projekt hat Zuschauer. Projektbetrachter können eine Verbindung mit dem Portal herstellen, um die Dashboards anzuzeigen, die Projekteigentümer erstellt haben. In jedem Dashboard können Projektbetrachter den Zeitraum anpassen, um die Betriebsdaten besser zu verstehen. Projektbetrachter können nur Dashboards in den Projekten anzeigen, auf die sie Zugriff haben. Projektbeobachter können Alarmer bestätigen und die Schlummerfunktion aktivieren.

Eigenschaftsalias

Sie haben die Möglichkeit, Aliase für Asset-Eigenschaften zu erstellen, wie z. B. einen OPC UA-Server-Datenstream-Pfad (z. B. /company/windfarm/3/turbine/7/temperature), um die

Identifizierung einer Anlageneigenschaft beim Erfassen oder Abrufen von Asset-Daten zu vereinfachen. Wenn Sie ein [SiteWise Edge-Gateway](#) verwenden, um Daten von Servern aufzunehmen, müssen Ihre Eigenschaftsalias mit den Pfaden Ihrer Rohdatenströme übereinstimmen. Weitere Informationen finden Sie unter [Datenströme verwalten für AWS IoT SiteWise](#).

Eigenschaftsbenachrichtigung

Wenn Sie Eigenschaftsbenachrichtigungen für eine Vermögenseigenschaft aktivieren, AWS IoT SiteWise veröffentlicht AWS IoT Core jedes Mal, wenn diese Eigenschaft einen neuen Wert erhält, eine MQTT-Nachricht. Die Nutzdaten der Nachricht enthalten Details zur Aktualisierung dieses Eigenschaftswerts. Verwenden Sie Benachrichtigungen über Immobilienwerte, um Lösungen zu entwickeln, die Ihre Industriedaten AWS IoT SiteWise mit anderen AWS Diensten verbinden. Weitere Informationen finden Sie unter [Interagiere mit anderen AWS Diensten](#).

SiteWise Edge-Gateway

Ein SiteWise Edge-Gateway wird beim Kunden installiert, um Daten zu sammeln, zu verarbeiten und weiterzuleiten. Ein SiteWise Edge-Gateway stellt über verschiedene Protokolle eine Verbindung zu Ihren industriellen Datenquellen her, um Daten zu sammeln, zu verarbeiten und an die AWS Cloud zu senden. SiteWise Edge-Gateways können sich auch mit Partnerdatenquellen verbinden. Weitere Informationen finden Sie unter [Verwenden Sie AWS IoT SiteWise Edge-Gateways](#).

Transformation

Transformationen sind Eigenschaften eines Assets, die transformierte Zeitreihendaten darstellen. Jede Transformation wird von einem mathematischen Ausdruck ([Formel](#)) begleitet, der angibt, wie Datenpunkte von einer Form in eine andere konvertiert werden. Die transformierten Datenpunkte stehen in einer one-to-one Beziehung zu den Eingabedatenpunkten. Weitere Informationen finden Sie unter [Daten transformieren \(transformiert\)](#).

Visualisierung

In jedem Dashboard entscheiden die Projekteigentümer, wie die Eigenschaften und Alarmer mit dem Projekt verknüpften Objekte angezeigt werden sollen. Die Verfügbarkeit kann als Liniendiagramm dargestellt werden, während andere Werte als Balkendiagramme oder wichtige Leistungsindikatoren (KPIs) dargestellt werden können. Alarmer lassen sich am besten als Statusraster und Statuszeitleisten anzeigen. Projekteigentümer passen jede Visualisierung an, um die Daten für diese Komponente optimal darzustellen.

Fangen Sie an mitAWS IoT SiteWise

Mit AWS IoT SiteWise können Sie Ihre Daten sammeln, organisieren, analysieren und visualisieren.

AWS IoT SiteWisebietet eine Demo, mit der Sie den Service erkunden können, ohne eine echte Datenquelle konfigurieren zu müssen. Weitere Informationen finden Sie unter [Benutze die AWS IoT SiteWise Demo](#).

Sie können die folgenden Tutorials absolvieren, um sich mit bestimmten Funktionen von vertraut zu machenAWS IoT SiteWise:

- [Daten aufnehmen, um Dinge AWS IoT SiteWise zu erstellen AWS IoT](#)
- [Visualisieren und teilen Sie Windparkdaten in SiteWise Monitor](#)
- [Aktualisierungen von Immobilienwerten in Amazon DynamoDB veröffentlichen](#)

In den folgenden Themen erfahren Sie mehr überAWS IoT SiteWise:

- [Daten aufnehmen inAWS IoT SiteWise](#)
- [Modellieren Sie Industrieanlagen](#)
- [Konfigurieren Sie Edge-Funktionen aufAWS IoT SiteWiseEdge](#)
- [Überwachen Sie Daten mit AWS IoT SiteWise Monitor](#)
- [Daten abfragen vonAWS IoT SiteWise](#)
- [Interagiere mit anderen AWS Diensten](#)

Themen

- [Voraussetzungen](#)
- [Richten Sie eine einAWSKonto](#)

Voraussetzungen

Sie müssen über ein AWS Konto verfügen, um damit beginnen zu könnenAWS IoT SiteWise. Falls Sie noch keines haben, finden Sie im folgenden Abschnitt Anweisungen zur Einrichtung eines Kontos.

Verwenden Sie eine Region, in AWS IoT SiteWise der verfügbar ist. Weitere Informationen finden Sie unter [AWS IoT SiteWise-Endpunkte und -Kontingente](#). Sie können die Regionsauswahl in der AWS-Managementkonsole, um zu einer dieser Regionen zu wechseln.

Richten Sie ein AWS-Konto

Melde dich an für ein AWS-Konto

Um loszulegen AWS, benötigen Sie ein AWS-Konto. Informationen zum Erstellen eines AWS-Kontos finden Sie unter [Erste Schritte mit einem AWS-Konto](#) im AWS-KontenverwaltungsReferenzhandbuch.

Benutze die AWS IoT SiteWise Demo

Mit der AWS IoT SiteWise Demo können Sie ganz einfach die Umgebung erkunden AWS IoT SiteWise . AWS IoT SiteWise stellt die Demo als CloudFormation Vorlage bereit, die Sie einsetzen können, um Asset-Modelle, Assets und ein SiteWise Monitor-Portal zu erstellen und Beispieldaten für bis zu einer Woche zu generieren.

Important

Sobald Sie die Demo erstellt haben, werden Ihnen die Ressourcen in Rechnung gestellt, die durch diese Demo erstellt und verbraucht werden.

Themen

- [Erstellen Sie die Demo AWS IoT SiteWise](#)
- [Löschen Sie die AWS IoT SiteWise Demo](#)

Erstellen Sie die Demo AWS IoT SiteWise

Sie können die AWS IoT SiteWise Demo von der AWS IoT SiteWise Konsole aus erstellen.

Note

Die Demo erstellt Lambda-Funktionen, eine CloudWatch Event-Regel und die für die Demo erforderlichen AWS Identity and Access Management (IAM-) Rollen. Möglicherweise sehen Sie diese Ressourcen in Ihrem AWS Konto. Wir empfehlen Ihnen, diese Ressourcen beizubehalten, bis Sie mit der Demo fertig sind. Wenn Sie die Ressourcen löschen, funktioniert die Demo möglicherweise nicht mehr richtig.

Um die Demo in der AWS IoT SiteWise Konsole zu erstellen

1. Navigieren Sie zur [AWS IoT SiteWise Konsole](#) und suchen Sie die SiteWise Demo in der oberen rechten Ecke der Seite.

2. (Optional) Ändern Sie unter SiteWise Demo das Feld Tage, bis die Demo-Assets aufbewahrt werden sollen, um anzugeben, wie viele Tage die Demo aufbewahrt werden soll, bevor sie gelöscht wird.
3. (Optional) Gehen Sie wie folgt vor, um ein SiteWise Monitor-Portal zur Überwachung von Beispieldaten zu erstellen.

 Note

Die SiteWise Monitor-Ressourcen, die in dieser Demo erstellt und verbraucht werden, werden Ihnen in Rechnung gestellt. Weitere Informationen finden Sie in der AWS IoT SiteWise Preisübersicht unter [SiteWise Monitor](#).

- a. Wählen Sie „Ressourcen überwachen“.
- b. Wählen Sie „Erlaubnis“.
- c. Wählen Sie eine bestehende IAM-Rolle aus, die Ihren föderierten IAM-Benutzern Zugriff auf das Portal gewährt.

 Important

Ihre IAM-Rolle muss über die folgenden Berechtigungen verfügen.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:Describe*",
        "iotsitewise:List*",
        "iotsitewise:Get*",
        "cloudformation:DescribeStacks",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedRolePolicies",
        "sso:DescribeRegisteredRegions",
        "organizations:DescribeOrganization"
      ]
    }
  ]
}
```

```
    ],  
    "Resource": "*"    
  }  
]  
}
```

Weitere Informationen zur Arbeit mit SiteWise Monitor finden Sie unter [Was ist AWS IoT SiteWise Monitor?](#) im AWS IoT SiteWise Monitor Anwendungshandbuch.

4. Wählen Sie Create demo (Demo erstellen) aus.

Das Erstellen der Demo dauert ca. 3 Minuten. Wenn die Demo nicht erstellt werden kann, verfügt Ihr Konto möglicherweise nicht über ausreichende Berechtigungen. Wechseln Sie zu einem Konto mit Administratorberechtigungen oder führen Sie die folgenden Schritte aus, um die Demo zu löschen, und versuchen Sie es erneut:

a. Wählen Sie Demo löschen aus.

Das Löschen der Demo dauert etwa 15 Minuten.

b. Wenn die Demo nicht gelöscht wird, öffnen Sie die [CloudFormation Konsole](#), wählen Sie den Stack mit dem Namen Io TSite WiseDemoAssets aus und wählen Sie in der oberen rechten Ecke Löschen aus.

c. Wenn die Demo erneut nicht gelöscht werden kann, folgen Sie den Schritten in der CloudFormation Konsole, um die Ressourcen zu überspringen, die nicht gelöscht werden konnten, und versuchen Sie es erneut.

5. Nachdem die Demo erfolgreich erstellt wurde, können Sie die Demo-Assets und -Daten in der [AWS IoT SiteWise Konsole](#) erkunden.

Löschen Sie die AWS IoT SiteWise Demo

Die AWS IoT SiteWise Demo löscht sich nach einer Woche oder nach der Anzahl der Tage, die Sie ausgewählt haben, wenn Sie den Demo-Stack von der CloudFormation Konsole aus erstellt haben. Sie können die Demo vorher löschen, wenn Sie mit den Demoressourcen fertig sind. Sie können die Demo auch löschen, wenn die Demo nicht erstellt werden kann. Gehen Sie wie folgt vor, um die Demo manuell zu löschen.

Um die Demo zu löschen AWS IoT SiteWise

1. Navigieren Sie zur [CloudFormation -Konsole](#).
2. Wählen Sie IoTSiteWiseDemoAssets aus der Liste der Stacks aus.
3. Wählen Sie Löschen aus.

Wenn Sie den Stack löschen, werden alle für die Demo erstellten Ressourcen gelöscht.

4. Wählen Sie im Bestätigungsdiaologfeld Stack löschen aus.

Das Löschen des Stacks dauert etwa 15 Minuten. Wenn die Demo nicht gelöscht werden kann, wählen Sie oben rechts erneut Löschen aus. Wenn die Demo erneut nicht gelöscht werden kann, folgen Sie den Schritten in der CloudFormation Konsole, um die Ressourcen zu überspringen, die nicht gelöscht werden konnten, und versuchen Sie es erneut.

AWS IoT SiteWise Tutorials

Willkommen auf der AWS IoT SiteWise Tutorial-Seite. Diese wachsende Sammlung von Tutorials vermittelt Ihnen das Wissen und die Fähigkeiten, die Sie benötigen, um sich mit den Feinheiten von vertraut zu machen. AWS IoT SiteWise Diese Tutorials bieten eine Vielzahl grundlegender Themen, die auf Ihre Bedürfnisse zugeschnitten sind. Während Sie sich mit den Tutorials befassen, erhalten Sie wertvolle Einblicke in verschiedene Aspekte von. AWS IoT SiteWise

Jedes Tutorial verwendet ein bestimmtes Ausrüstungsbeispiel. Diese Tutorials sind für Testumgebungen vorgesehen und verwenden fiktive Firmennamen, Modelle, Vermögenswerte, Eigenschaften usw. In den Tutorials finden Sie allgemeine Anleitungen. Die Tutorials sind nicht für den direkten Einsatz in einer Produktionsumgebung vorgesehen, es sei denn, sie werden sorgfältig geprüft und an die individuellen Anforderungen Ihres Unternehmens angepasst.

Themen

- [Berechnung der Gesamtanlageneffektivität in AWS IoT SiteWise](#)
- [Daten aufnehmen, um Dinge AWS IoT SiteWise zu erstellen AWS IoT](#)
- [Integrieren Sie Daten mithilfe eines MQTT-fähigen V3-Gateways in SiteWise Edge](#)
- [Daten in Grafana visualisieren und teilen](#)
- [Visualisieren und teilen Sie Windparkdaten in SiteWise Monitor](#)
- [Aktualisierungen von Immobilienwerten in Amazon DynamoDB veröffentlichen](#)

Berechnung der Gesamtanlageneffektivität in AWS IoT SiteWise

Dieses Tutorial enthält ein Beispiel dafür, wie die Gesamtanlageneffektivität (OEE) für einen Herstellungsprozess berechnet wird. Folglich können Ihre OEE-Berechnungen oder -Formeln von den hier dargestellten abweichen. Im Allgemeinen ist OEE definiert als $\text{Availability} * \text{Quality} * \text{Performance}$. Weitere Informationen über die Berechnung der OEE finden Sie unter [Overall equipment effectiveness](#) auf Wikipedia.

Voraussetzungen

Um dieses Tutorial abzuschließen, müssen Sie die Datenerfassung für ein Gerät mit den folgenden drei Daten-Streams konfigurieren:

- **Equipment_State**— Ein numerischer Code, der den Zustand der Maschine darstellt, z. B. Leerlauf, Störung, geplanter Stopp oder Normalbetrieb.
- **Good_Count**— Ein Datenstrom, bei dem jeder Datenpunkt die Anzahl der erfolgreichen Operationen seit dem letzten Datenpunkt enthält.
- **Bad_Count**— Ein Datenstrom, bei dem jeder Datenpunkt die Anzahl der erfolglosen Operationen seit dem letzten Datenpunkt enthält.

Informationen zum Konfigurieren der Datenerfassung finden Sie im Abschnitt [Daten aufnehmen in AWS IoT SiteWise](#). Wenn keine industriellen Operationen verfügbar sind, können Sie ein Skript schreiben, das Beispieldaten über die AWS IoT SiteWise -API generiert und hochlädt.

Berechnen der OEE

In diesem Tutorial erstellen Sie ein Komponentenmodell, das die OEE aus drei Dateneingabe-Streams berechnet: **Equipment_State**, **Good_Count**, und **Bad_Count**. Stellen Sie sich in diesem Beispiel eine allgemeine Verpackungsmaschine vor, beispielsweise eine Maschine, die zum Verpacken von Zucker, Kartoffelchips oder Farbe verwendet wird. Erstellen Sie in der [AWS IoT SiteWise Konsole](#) ein AWS IoT SiteWise Asset-Modell mit den folgenden Messungen, Transformationen und Metriken. Anschließend können Sie ein Asset erstellen, das die Verpackungsmaschine darstellt, und beobachten, wie die Gesamtanlageneffektivität AWS IoT SiteWise berechnet wird.

Definieren Sie die folgenden [Messungen](#), um die Rohdaten-Streams von der Verpackungsmaschine darzustellen.

Messungen

- **Equipment_State**— Ein Datenstrom (oder eine Messung), der den aktuellen Zustand der Verpackungsmaschine in numerischen Codes wiedergibt:
 - **1024**— Die Maschine befindet sich im Leerlauf.
 - **1020**— Ein Fehler, z. B. ein Fehler oder eine Verzögerung.
 - **1000**— Ein geplanter Stopp.
 - **1111**— Ein normaler Betrieb.
- **Good_Count**— Ein Datenstrom, bei dem jeder Datenpunkt die Anzahl der erfolgreichen Operationen seit dem letzten Datenpunkt enthält.

- **Bad_Count**— Ein Datenstrom, bei dem jeder Datenpunkt die Anzahl der erfolglosen Operationen seit dem letzten Datenpunkt enthält.

Legen Sie mithilfe des **Equipment_State**-Messdaten-Streams und der darin enthaltenen Codes die folgenden [Transformationen](#) (oder abgeleiteten Messungen) fest. Transformationen haben eine one-to-one Beziehung zu Rohmessungen.

Transformationen

- **Idle** = `eq(Equipment_State, 1024)`— Ein transformierter Datenstrom, der den Ruhezustand der Maschine enthält.
- **Fault** = `eq(Equipment_State, 1020)`— Ein transformierter Datenstrom, der den Fehlerstatus der Maschine enthält.
- **Stop** = `eq(Equipment_State, 1000)`— Ein transformierter Datenstrom, der den geplanten Stoppstatus der Maschine enthält.
- **Running** = `eq(Equipment_State, 1111)`— Ein transformierter Datenstrom, der den normalen Betriebszustand der Maschine enthält.

Definieren Sie anhand der Rohmessungen und der transformierten Messungen die folgenden [Metriken](#), die Maschinendaten über bestimmte Zeitintervalle aggregieren. Wählen Sie für jede Metrik dasselbe Zeitintervall aus, wenn Sie die Metriken in diesem Abschnitt definieren.

Metriken

- **Successes** = `sum(Good_Count)`— Die Anzahl der erfolgreich befüllten Pakete im angegebenen Zeitintervall.
- **Failures** = `sum(Bad_Count)`— Die Anzahl der Pakete, die im angegebenen Zeitintervall nicht erfolgreich gefüllt wurden.
- **Idle_Time** = `statetime(Idle)`— Die gesamte Leerlaufzeit der Maschine (in Sekunden) pro festgelegtem Zeitintervall.
- **Fault_Time** = `statetime(Fault)`— Die Gesamtfehlerzeit der Maschine (in Sekunden) pro festgelegtem Zeitintervall.
- **Stop_Time** = `statetime(Stop)`— Die gesamte geplante Stoppzeit der Maschine (in Sekunden) pro festgelegtem Zeitintervall.
- **Run_Time** = `statetime(Running)`— Die Gesamtbetriebszeit (in Sekunden) der Maschine ohne Probleme pro festgelegtem Zeitintervall.

- $\text{Down\_Time} = \text{Idle\_Time} + \text{Fault\_Time} + \text{Stop\_Time}$ — Die gesamte Ausfallzeit der Maschine (in Sekunden) über das angegebene Zeitintervall, berechnet als Summe der Maschinenzustände außer `Run_Time`.
- $\text{Availability} = \text{Run\_Time} / (\text{Run\_Time} + \text{Down\_Time})$ — Die Betriebszeit der Maschine oder der Prozentsatz der geplanten Zeit, während der die Maschine während des angegebenen Zeitintervalls betriebsbereit ist.
- $\text{Quality} = \text{Successes} / (\text{Successes} + \text{Failures})$ — Der Prozentsatz der erfolgreich abgefüllten Pakete der Maschine in den angegebenen Zeitintervallen.
- $\text{Performance} = ((\text{Successes} + \text{Failures}) / \text{Run\_Time}) / \text{Ideal\_Run\_Rate}$ — Die Leistung der Maschine im angegebenen Zeitintervall als Prozentsatz der für Ihren Prozess idealen Durchlaufgeschwindigkeit (in Sekunden).

Ihre `Ideal_Run_Rate` beläuft sich beispielsweise auf 60 Pakete pro Minute (1 Paket pro Sekunde). Wenn Ihr Wert pro Minute oder pro Stunde angegeben `Ideal_Run_Rate` wird, müssen Sie ihn durch den entsprechenden Umrechnungsfaktor für Einheiten dividieren, da er in Sekunden angegeben `Run_Time` ist.

- $\text{OEE} = \text{Availability} * \text{Quality} * \text{Performance}$ — Die Gesamtanlageneffektivität der Maschine über das angegebene Zeitintervall. Diese Formel berechnet OEE als Bruchteil von 1.

Note

Wenn OEE als Transformation definiert ist, werden die Ausgabewerte für jeden der Eingabewerte berechnet. Es besteht die Möglichkeit, dass unerwartete Werte generiert werden, da bei der Transformationsauswertung die neuesten verfügbaren Werte für alle beitragenden Eigenschaften in der Formel berücksichtigt werden. Bei Eigenschaftenaktualisierungen mit demselben Zeitstempel können Ausgabewerte durch Aktualisierungen anderer eingehender Eigenschaften überschrieben werden. Wenn beispielsweise Verfügbarkeit, Qualität und Leistung berechnet werden, wird die Gesamtanlageneffektivität anhand der letzten verfügbaren Datenpunkte für die anderen beiden Eigenschaften berechnet. Diese beitragenden Werte haben dieselben Zeitstempel und führen zu falschen OEE-Ausgabewerten. Die Reihenfolge ist für die Berechnung von Transformationen nicht garantiert.

Daten aufnehmen, um Dinge AWS IoT SiteWise zu erstellen AWS IoT

In diesem Tutorial erfahren Sie, wie Sie mithilfe AWS IoT SiteWise von AWS IoT Geräteschatten Daten aus einer Flotte von Geräten aufnehmen. Geräteschatten sind JSON-Objekte, die aktuelle Statusinformationen für ein AWS IoT Gerät speichern. Weitere Informationen finden Sie unter [Device Shadow Service](#) im AWS IoT Entwicklerhandbuch.

Nachdem Sie dieses Tutorial abgeschlossen haben, können Sie einen Vorgang einrichten, der auf AWS IoT Dingen AWS IoT SiteWise basiert. Indem Sie AWS IoT Dinge verwenden, können Sie Ihren Betrieb in andere nützliche Funktionen von integrieren AWS IoT. Sie können beispielsweise AWS IoT Funktionen für die folgenden Aufgaben konfigurieren:

- Konfigurieren Sie zusätzliche Regeln für das Streamen von Daten zu [AWS IoT Events](#) [Amazon DynamoDB](#) und anderen AWS Services. Weitere Informationen finden Sie unter [Regeln](#) im AWS IoT Entwicklerhandbuch.
- Indexieren, durchsuchen und aggregieren Sie Ihre Gerätedaten mit dem AWS IoT Fleet Indexing Service. Weitere Informationen finden Sie unter [Fleet Indexing Service](#) im AWS IoT Entwicklerhandbuch.
- Prüfen und sichern Sie Ihre Geräte mit AWS IoT Device Defender. Weitere Informationen finden Sie unter [AWS IoT Device Defender](#) im AWS IoT -Entwicklerhandbuch.

In diesem Tutorial erfahren Sie, wie Sie Daten aus den Geräteschatten AWS IoT von Dingen in Assets aufnehmen. AWS IoT SiteWise Dazu erstellen Sie ein oder mehrere AWS IoT Dinge und führen ein Skript aus, das den Geräteshadow jedes Dings mit Daten zur CPU- und Speichernutzung aktualisiert. In diesem Tutorial verwenden Sie CPU- und Speichernutzungsdaten, um realistische Sensordaten zu imitieren. Anschließend erstellen Sie eine Regel mit einer AWS IoT SiteWise Aktion, die diese Daten bei AWS IoT SiteWise jeder Aktualisierung des Geräteshadows an ein Asset sendet. Weitere Informationen finden Sie unter [Daten AWS IoT SiteWise mithilfe AWS IoT Core von Regeln aufnehmen](#).

Themen

- [Voraussetzungen](#)
- [Schritt 1: Erstellen Sie eine AWS IoT Richtlinie](#)
- [Schritt 2: Erstellen und konfigurieren Sie ein AWS IoT Ding](#)
- [Schritt 3: Erstellen Sie ein Geräte-Asset-Modell](#)

- [Schritt 4: Erstellen Sie ein Geräteflotten-Asset-Modell](#)
- [Schritt 5: Erstellen und konfigurieren Sie ein Geräte-Asset](#)
- [Schritt 6: Erstellen und konfigurieren Sie ein Geräteflotten-Asset](#)
- [Schritt 7: Erstellen Sie in AWS IoT Core eine Regel, um Daten an Geräteressourcen zu senden](#)
- [Schritt 8: Führen Sie das Geräteclient-Skript aus](#)
- [Schritt 9: Ressourcen nach dem Tutorial bereinigen](#)

Voraussetzungen

Zum Durcharbeiten dieses Tutorials ist Folgendes erforderlich:

- Ein AWS Konto. Falls Sie noch keines haben, beachten Sie die Informationen unter [Richten Sie eine einAWSKonto](#).
- Ein Entwicklungscomputer, auf dem Windows, macOS Linux, oder ausgeführt wird, Unix um auf den zuzugreifen AWS-Managementkonsole. Weitere Informationen finden Sie unter [Erste Schritte mit AWS-Managementkonsole](#).
- Ein AWS Identity and Access Management (IAM-) Benutzer mit Administratorrechten.
- Python3 ist auf Ihrem Entwicklungscomputer oder auf dem Gerät installiert, das Sie als Objekt registrieren möchten AWS IoT .

Schritt 1: Erstellen Sie eine AWS IoT Richtlinie

Erstellen Sie in diesem Verfahren eine AWS IoT Richtlinie, die es Ihren AWS IoT Dingen ermöglicht, auf die in diesem Tutorial verwendeten Ressourcen zuzugreifen.

Console

Gehen Sie wie folgt vor, um mithilfe der AWS IoT Core Konsole eine AWS IoT Richtlinie zu erstellen:

Um eine AWS IoT Richtlinie zu erstellen

1. Melden Sie sich an der [AWS-Managementkonsole](#) an.
2. Sehen Sie sich die [AWS Regionen](#) an, in denen AWS IoT SiteWise es unterstützt wird. Wechseln Sie ggf. zu einer dieser unterstützten Regionen.

3. Navigieren Sie zur [AWS IoT -Konsole](#). Wenn die Schaltfläche Gerät Connect angezeigt wird, wählen Sie sie aus.
4. Wählen Sie im linken Navigationsbereich Sicherheit und dann Richtlinien aus.
5. Wählen Sie Erstellen aus.
6. Geben Sie einen Namen für die AWS IoT Richtlinie ein (z. **B.SiteWiseTutorialDevicePolicy**).
7. Wählen Sie unter Richtliniendokument die Option JSON aus, um die folgende Richtlinie im JSON-Format einzugeben. Ersetzen Sie *region* und *account-id* durch Ihre Region und Konto-ID, z. B. **us-east-1** und **123456789012**.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource": "arn:aws:iot:us-east-1:123456789012:client/SiteWiseTutorialDevice*"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Publish",
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/${iot:Connection.Thing.ThingName}/shadow/update",
        "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/${iot:Connection.Thing.ThingName}/shadow/delete",
        "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/${iot:Connection.Thing.ThingName}/shadow/get"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/${iot:Connection.Thing.ThingName}/shadow/update/accepted",

```

```

        "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/
        ${iot:Connection.Thing.ThingName}/shadow/delete/accepted",
        "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/
        ${iot:Connection.Thing.ThingName}/shadow/get/accepted",
        "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/
        ${iot:Connection.Thing.ThingName}/shadow/update/rejected",
        "arn:aws:iot:us-east-1:123456789012:topic/$aws/things/
        ${iot:Connection.Thing.ThingName}/shadow/delete/rejected"
    ]
  },
  {
    "Effect": "Allow",
    "Action": "iot:Subscribe",
    "Resource": [
      "arn:aws:iot:us-east-1:123456789012:topicfilter/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/update/accepted",
      "arn:aws:iot:us-east-1:123456789012:topicfilter/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/delete/accepted",
      "arn:aws:iot:us-east-1:123456789012:topicfilter/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/get/accepted",
      "arn:aws:iot:us-east-1:123456789012:topicfilter/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/update/rejected",
      "arn:aws:iot:us-east-1:123456789012:topicfilter/$aws/things/
      ${iot:Connection.Thing.ThingName}/shadow/delete/rejected"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "iot:GetThingShadow",
      "iot:UpdateThingShadow",
      "iot:DeleteThingShadow"
    ],
    "Resource": "arn:aws:iot:us-
    east-1:123456789012:thing/SiteWiseTutorialDevice*"
  }
]
}

```

8. Wählen Sie Erstellen aus.

AWS CLI

 Important

Diese Richtlinie verwendet Platzhalter, um die AWS IoT SiteWise CLI-Größenbeschränkungen einzuhalten. Für restriktivere Berechtigungen mit expliziten Themenpfaden erstellen Sie die Richtlinie stattdessen über die AWS IoT SiteWise Konsole. Weitere Informationen finden Sie im Beispiel für eine IoT-Richtlinie auf der Registerkarte.

Verwenden Sie den folgenden AWS CLI Befehl, um eine IoT-Richtlinie zu erstellen:

```
aws iot create-policy \  
  --policy-name "SiteWiseTutorialDevicePolicy" \  
  --policy-document '{  
    "Version": "2012-10-17",  
    "Statement": [  
      {  
        "Effect": "Allow",  
        "Action": "iot:Connect",  
        "Resource": "arn:aws:iot:region:account-id:client/SiteWiseTutorialDevice*"  
      },  
      {  
        "Effect": "Allow",  
        "Action": ["iot:Publish", "iot:Receive"],  
        "Resource": [  
          "arn:aws:iot:region:account-id:topic/$aws/things/  
${iot:Connection.Thing.ThingName}/shadow/*"  
        ]  
      },  
      {  
        "Effect": "Allow",  
        "Action": "iot:Subscribe",  
        "Resource": [  
          "arn:aws:iot:region:account-id:topicfilter/$aws/things/  
${iot:Connection.Thing.ThingName}/shadow/*"  
        ]  
      },  
      {  
        "Effect": "Allow",  
        "Action": [  

```

```
        "iot:GetThingShadow",
        "iot:UpdateThingShadow",
        "iot:DeleteThingShadow"
    ],
    "Resource": "arn:aws:iot:region:account-id:thing/SiteWiseTutorialDevice*"
}
]
```

Verwenden Sie den folgenden Befehl, um zu überprüfen, ob Ihre Richtlinie erfolgreich erstellt wurde:

```
aws iot get-policy --policy-name "SiteWiseTutorialDevicePolicy"
```

Diese Richtlinie ermöglicht es Ihren AWS IoT Geräten, mithilfe von MQTT-Nachrichten Verbindungen herzustellen und mit Geräteschatten zu kommunizieren. Weitere Informationen zu MQTT-Nachrichten finden Sie unter [Was ist MQTT?](#) . Um mit Geräteschatten zu interagieren, veröffentlichen und empfangen Ihre AWS IoT Dinge MQTT-Nachrichten zu Themen, die mit `$aws/things/thing-name/shadow/` beginnen. Diese Richtlinie enthält eine Ding-RichtlinienvARIABLE, die als `${iot:Connection.Thing.ThingName}`. Diese Variable ersetzt in jedem Thema den Namen der verbundenen Sache. Die `iot:Connect` Anweisung beschränkt, welche Geräte Verbindungen herstellen können, und stellt so sicher, dass die RichtlinienvARIABLE `thing` nur Namen ersetzen kann, die mit `SiteWiseTutorialDevice` beginnen.

Weitere Informationen finden Sie unter [Ding-RichtlinienvARIABLEN](#) im AWS IoT Entwicklerhandbuch.

Note

Diese Richtlinie gilt für Objekte, deren Namen mit `SiteWiseTutorialDevice` beginnen. Um einen anderen Namen für Ihre Objekte zu verwenden, müssen Sie die Richtlinie entsprechend aktualisieren.

Schritt 2: Erstellen und konfigurieren Sie ein AWS IoT Ding

In diesem Verfahren erstellen und konfigurieren Sie ein AWS IoT Ding. Sie können Ihren Entwicklungscomputer als ein AWS IoT Ding bezeichnen. Denken Sie im weiteren Verlauf daran, dass die Prinzipien, die Sie hier lernen, auch auf konkrete Projekte angewendet werden können. Sie

haben die Flexibilität, AWS IoT Dinge auf jedem Gerät zu erstellen und einzurichten, auf dem ein AWS IoT SDK ausgeführt werden kann, einschließlich AWS IoT Greengrass FreeRTOS. Weitere Informationen finden Sie unter [AWS IoT SDKs](#) im AWS IoT -Entwicklerhandbuch.

Console

Um etwas zu erstellen und zu konfigurieren AWS IoT

1. Öffnen Sie eine Befehlszeile, und führen Sie den folgenden Befehl aus, um ein Verzeichnis für dieses Lernprogramm zu erstellen.

```
mkdir iot-sitewise-rule-tutorial  
cd iot-sitewise-rule-tutorial
```

2. Führen Sie den folgenden Befehl aus, um ein Verzeichnis für die Zertifikate Ihres Objekts zu erstellen.

```
mkdir device1
```

Wenn Sie zusätzliche Objekte erstellen, erhöhen Sie die Nummer im Verzeichnisnamen entsprechend, um zu verfolgen, welche Zertifikate zu welchem Objekt gehören.

3. Navigieren Sie zur [AWS IoT -Konsole](#).
4. Wählen Sie im linken Navigationsbereich im Abschnitt Verwalten die Option Alle Geräte aus. Wählen Sie dann Things (Objekte) aus.
5. Wenn das Dialogfeld You don't have any things yet (Sie haben noch keine Objekte) angezeigt wird, wählen Sie Create a thing (Objekt erstellen) aus. Wählen Sie andernfalls Dinge erstellen aus.
6. Wählen Sie auf der Seite „Dinge erstellen“ die Option „Ein einzelnes Ding erstellen“ und dann „Weiter“ aus.
7. Geben Sie auf der Seite „Dingeigenschaften angeben“ einen Namen für Ihr AWS IoT Ding ein (z. B. **SiteWiseTutorialDevice1**) und wählen Sie dann Weiter. Wenn Sie zusätzliche Objekte erstellen, erhöhen Sie die Nummer im Namen des Objekts entsprechend.

 Important

Der Name des Dings muss mit dem Namen übereinstimmen, der in der Richtlinie verwendet wurde, die Sie in Schritt 1: AWS IoT Richtlinie erstellen erstellt haben. Andernfalls kann Ihr Gerät keine Verbindung zu herstellen AWS IoT.

8. Wählen Sie auf der Seite Gerätezertifikat konfigurieren — optional die Option Neues Zertifikat automatisch generieren (empfohlen) und anschließend Weiter aus. Mithilfe von Zertifikaten können AWS IoT Sie Ihre Geräte sicher identifizieren.
9. Wählen Sie auf der Seite Richtlinien an Zertifikat anhängen — optional die Richtlinie aus, die Sie in Schritt 1: AWS IoT Richtlinie erstellen erstellt haben, und wählen Sie Ding erstellen aus.
10. Gehen Sie im Dialogfeld Zertifikate und Schlüssel herunterladen wie folgt vor:
 - a. Wählen Sie die Download-Links, um das Zertifikat, den öffentlichen Schlüssel und den privaten Schlüssel Ihres Objekts herunterzuladen. Speichern Sie alle drei Dateien in dem Verzeichnis, das Sie für die Zertifikate Ihres Objekts erstellt haben (zum Beispiel **iot-sitewise-rule-tutorial/device1**).

 Important

Dies ist das einzige Mal, dass Sie das Zertifikat und die Schlüssel Ihres Objekts herunterladen können, die Sie benötigen, damit Ihr Gerät erfolgreich eine Verbindung mit AWS IoT herstellen kann.

- b. Wählen Sie den Link Herunterladen, um ein Root-CA-Zertifikat herunterzuladen. Speichern Sie das CA-Stammzertifikat der Zertifizierungsstelle in **iot-sitewise-rule-tutorial**. Wir empfehlen Ihnen, Amazon Root CA 1 herunterzuladen.
11. Wählen Sie Fertig aus.

AWS CLI

Gehen Sie wie folgt vor, um ein Objekt zu erstellen und zu konfigurieren AWS IoT , indem Sie AWS CLI:

1. Öffnen Sie eine Befehlszeile und führen Sie den folgenden Befehl aus, um ein Verzeichnis für dieses Tutorial zu erstellen:

```
mkdir iot-sitewise-rule-tutorial
```

2. Navigieren Sie zum Verzeichnis für das Tutorial:

```
cd iot-sitewise-rule-tutorial
```

3. Führen Sie den folgenden Befehl aus, um ein Verzeichnis für die Zertifikate Ihres Dings zu erstellen:

```
mkdir device1
```

Wenn Sie zusätzliche Objekte erstellen, erhöhen Sie die Nummer im Verzeichnisnamen entsprechend, um zu verfolgen, welche Zertifikate zu welchem Objekt gehören.

4. Erstelle ein AWS IoT Ding:

```
aws iot create-thing --thing-name "SiteWiseTutorialDevice1"
```

 Important

Der Name des Dings muss dem Namensmuster entsprechen, das in der Richtlinie verwendet wurde, die Sie in Schritt 1 erstellt haben. Andernfalls kann Ihr Gerät keine Verbindung zu herstellen AWS IoT.

5. Erstellen Sie ein Zertifikat und speichern Sie die Dateien. Notieren Sie sich den Zertifikat-ARN aus der Ausgabe — Sie benötigen ihn in den nächsten Schritten:

```
aws iot create-keys-and-certificate \  
  --set-as-active \  
  --certificate-pem-outfile "device1/device.pem.crt" \  
  --public-key-outfile "device1/public.pem.key" \  
  --private-key-outfile "device1/private.pem.key"
```

6. Hängen Sie die Richtlinie, die Sie in Schritt 1 erstellt haben, an das Zertifikat an:

```
aws iot attach-policy \  
  --policy-name "SiteWiseTutorialDevicePolicy" \  
  --target "certificate-arn"
```

7. Hängen Sie das Zertifikat an das Ding an:

```
aws iot attach-thing-principal \  
  --thing-name "SiteWiseTutorialDevice1" \  
  --principal "certificate-arn"
```

8. Laden Sie das Amazon-Root-CA-Zertifikat herunter:

```
curl https://www.amazontrust.com/repository/AmazonRootCA1.pem >  
AmazonRootCA1.pem
```

Dieses Zertifikat ist erforderlich, damit Ihr Gerät erfolgreich eine Verbindung herstellen kann AWS IoT.

Important

Bewahren Sie Ihre Zertifikate und Schlüssel sicher auf. Sie können diese Anmeldeinformationen nicht erneut herunterladen, nachdem Sie sie erstellt haben.

Sie haben jetzt AWS IoT etwas auf Ihrem Computer registriert. Führen Sie einen der folgenden nächsten Schritte aus:

- Fahren Sie mit Schritt 3 fort: Erstellen eines Geräte-Asset-Modells, ohne zusätzliche AWS IoT Dinge zu erstellen. Sie können dieses Lernprogramm mit nur einem Objekt abschließen.
- Wiederholen Sie die Schritte in diesem Abschnitt auf einem anderen Computer oder Gerät, um weitere AWS IoT -Objekte zu erstellen. Für dieses Tutorial empfehlen wir, diese Option zu befolgen, damit Sie eindeutige CPU- und Speicherauslastungsdaten von mehreren Geräten erfassen können.
- Wiederholen Sie die Schritte in diesem Abschnitt auf demselben Gerät (Ihrem Computer), um weitere AWS IoT -Objekte zu erstellen. Jedes Gerät AWS IoT empfängt ähnliche CPU- und Speichernutzungsdaten von Ihrem Computer. Verwenden Sie daher diesen Ansatz, um zu demonstrieren, dass nicht eindeutige Daten von mehreren Geräten aufgenommen werden.

Schritt 3: Erstellen Sie ein Geräte-Asset-Modell

In diesem Verfahren erstellen Sie ein Asset-Modell, das Ihre Geräte repräsentiert, die CPU- und Speichernutzungsdaten streamen. AWS IoT SiteWise Um Daten in Anlagen zu verarbeiten, die Gerätegruppen repräsentieren, setzen Asset-Modelle konsistente Informationen für mehrere Anlagen desselben Typs voraus. Weitere Informationen finden Sie unter [Modellieren Sie Industrieanlagen](#).

So erstellen Sie ein Komponentenmodell, das ein Gerät darstellt

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Models (Modelle) aus.
3. Wählen Sie Asset-Modell erstellen aus.
4. Geben Sie unter Modelldetails einen Namen für Ihr Modell ein. Beispiel, **SiteWise Tutorial Device Model**.
5. Führen Sie unter Measurement definitions (Messungsdefinitionen) die folgenden Schritte aus:
 - a. Geben Sie unter Name **CPU Usage** ein.
 - b. Geben Sie unter Unit (Einheit) % ein.
 - c. Lassen Sie den Data type (Datentyp) bei Double (Doppelt).

Messungseigenschaften stellen die Rohdatenströme eines Geräts dar. Weitere Informationen finden Sie unter [Definieren Sie Datenströme von Geräten \(Messungen\)](#).

6. Wählen Sie Neue Messung hinzufügen, um eine zweite Messeigenschaft hinzuzufügen.
7. Führen Sie in der zweiten Zeile unter Measurement definitions (Messungsdefinitionen) die folgenden Schritte aus:
 - a. Geben Sie unter Name **Memory Usage** ein.
 - b. Geben Sie unter Unit (Einheit) % ein.
 - c. Lassen Sie den Data type (Datentyp) bei Double (Doppelt).
8. Führen Sie unter Metric definitions (Metrikdefinitionen) die folgenden Schritte aus:
 - a. Geben Sie unter Name **Average CPU Usage** ein.
 - b. Geben Sie unter Formula (Formel) **avg(CPU Usage)** ein. Wählen Sie CPU Usage aus der Autovervollständigungsliste aus, wenn sie angezeigt wird.
 - c. Geben Sie unter Time interval (Zeitintervall) **5 minutes** ein.

Metrikeigenschaften definieren Aggregationsberechnungen, die alle Eingabedatenpunkte über ein Intervall verarbeiten und einen einzelnen Datenpunkt pro Intervall ausgeben. Diese Metrikeigenschaft berechnet alle 5 Minuten die durchschnittliche CPU-Auslastung jedes Geräts. Weitere Informationen finden Sie unter [Aggregieren Sie Daten aus Immobilien und anderen Vermögenswerten \(Metriken\)](#).

9. Wählen Sie Neue Metrik hinzufügen, um eine zweite Metrikeigenschaft hinzuzufügen.
10. Führen Sie in der zweiten Zeile unter Metric definitions (Metrikdefinitionen) die folgenden Schritte aus:
 - a. Geben Sie unter Name **Average Memory Usage** ein.
 - b. Geben Sie unter Formula (Formel) **avg(Memory Usage)** ein. Wählen Sie Memory Usage aus der Autovervollständigungsliste aus, wenn sie angezeigt wird.
 - c. Geben Sie unter Time interval (Zeitintervall) **5 minutes** ein.

Diese Metrikeigenschaft berechnet alle 5 Minuten die durchschnittliche Speicherbelegung jedes Geräts.

11. (Optional) Fügen Sie weitere zusätzliche Metriken hinzu, die Sie pro Gerät berechnen möchten. Einige interessante Funktionen sind `min` und `max`. Weitere Informationen finden Sie unter [Verwenden Sie Formelausdrücke](#). In Schritt 4: Erstellen eines Geräteflotten-Asset-Modells erstellen Sie ein übergeordnetes Asset, das anhand von Daten aus Ihrer gesamten Geräteflotte Kennzahlen berechnen kann.
12. Wählen Sie Modell erstellen aus.

Schritt 4: Erstellen Sie ein Geräteflotten-Asset-Modell

In diesem Verfahren erstellen Sie ein Asset-Modell, AWS IoT SiteWise das Ihre Sammlung von Geräten symbolisiert. Innerhalb dieses Asset-Modells legen Sie eine Struktur fest, die es Ihnen ermöglicht, zahlreiche Geräte-Assets zu einem übergeordneten Flotten-Asset zu verknüpfen. Anschließend skizzieren Sie Kennzahlen im Flotten-Asset-Modell, um Daten aus allen verbundenen Gerätebeständen zu konsolidieren. Dieser Ansatz bietet Ihnen umfassende Einblicke in die Gesamtleistung Ihrer gesamten Flotte.

So erstellen Sie ein Komponentenmodell, das eine Geräteflotte darstellt:

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Models (Modelle) aus.
3. Wählen Sie Asset-Modell erstellen aus.
4. Geben Sie unter Modelldetails einen Namen für Ihr Modell ein. Beispiel, **SiteWise Tutorial Device Fleet Model**.
5. Führen Sie unter Hierarchy definitions (Hierarchiedefinitionen) die folgenden Schritte aus:
 - a. Geben Sie unter Hierarchy name (Hierarchiename) **Device** ein.
 - b. Wählen Sie unter Hierarchy model (Hierarchiemodell) das Gerätekomponentenmodell (**SiteWise Tutorial Device Model**).

Eine Hierarchie definiert eine Beziehung zwischen einem übergeordneten (Flotten-) Komponentenmodell und einem untergeordneten (Geräte-) Komponentenmodell. Übergeordnete Komponenten können auf die Eigenschaftendaten von untergeordneten Komponenten zugreifen. Wenn Sie Komponenten später erstellen, müssen Sie untergeordnete Komponenten gemäß einer Hierarchiedefinition im übergeordneten Komponentenmodell den übergeordneten Komponenten zuordnen. Weitere Informationen finden Sie unter [Anlagenhierarchien stellen Gerätebeziehungen dar](#).

6. Führen Sie unter Metric definitions (Metrikdefinitionen) die folgenden Schritte aus:
 - a. Geben Sie unter Name **Average CPU Usage** ein.
 - b. Geben Sie unter Formula (Formel) **avg(Device | Average CPU Usage)** ein. Wenn die Autovervollständigungsliste angezeigt wird, wählen Sie Device, um eine Hierarchie auszuwählen, und dann Average CPU Usage, um die Metrik aus der zuvor erstellten Gerätekomponente auszuwählen.
 - c. Geben Sie unter Time interval (Zeitintervall) **5 minutes** ein.

Diese Metrikeigenschaft berechnet die durchschnittliche CPU-Auslastung aller Gerätekomponenten, die einer Flottenkomponente über die **Device**-Hierarchie zugeordnet sind.

7. Wählen Sie Neue Metrik hinzufügen, um eine zweite Metrikeigenschaft hinzuzufügen.
8. Führen Sie in der zweiten Zeile unter Metric definitions (Metrikdefinitionen) die folgenden Schritte aus:

- a. Geben Sie unter Name **Average Memory Usage** ein.
- b. Geben Sie unter Formula (Formel) **avg(Device | Average Memory Usage)** ein. Wenn die Autovervollständigungsliste angezeigt wird, wählen Sie Device, um eine Hierarchie auszuwählen, und dann Average Memory Usage, um die Metrik aus der zuvor erstellten Gerätekomponente auszuwählen.
- c. Geben Sie unter Time interval (Zeitintervall) **5 minutes** ein.

Diese Metrikeigenschaft berechnet die durchschnittliche Speicherbelegung aller Gerätekomponenten, die einer Flottenkomponente über die **Device**-Hierarchie zugeordnet sind.

9. (Optional) Fügen Sie weitere zusätzliche Metriken hinzu, die Sie für Ihre gesamte Geräteflotte berechnen möchten.
10. Wählen Sie Modell erstellen aus.

Schritt 5: Erstellen und konfigurieren Sie ein Geräte-Asset

In diesem Verfahren generieren Sie ein Geräte-Asset, das auf Ihrem Geräte-Asset-Modell basiert. Anschließend definieren Sie Eigenschaftsalise für jede Messungseigenschaft. Ein Eigenschaftsalias ist eine eindeutige Zeichenfolge, die eine Asset-Eigenschaft identifiziert. Später können Sie eine Eigenschaft für den Datenupload identifizieren, indem Sie die Aliase anstelle der Asset-ID und der Eigenschafts-ID verwenden. Weitere Informationen finden Sie unter [Datenströme verwalten für AWS IoT SiteWise](#).

So erstellen Sie eine Gerätekomponente und definieren Eigenschaftsalise

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Klicken Sie im linken Navigationsbereich auf Assets (Komponenten).
3. Wählen Sie dann Create asset (Komponente erstellen) aus.
4. Wählen Sie unter Modellinformationen das Asset-Modell Ihres Geräts aus. **SiteWise Tutorial Device Model**
5. Geben Sie unter Inventarinformationen einen Namen für Ihr Asset ein. Beispiel, **SiteWise Tutorial Device 1**.
6. Wählen Sie dann Create asset (Komponente erstellen) aus.
7. Wählen Sie für Ihre neue Gerätekomponente Edit (Bearbeiten).
8. Unter Messungen:

- a. Geben Sie unter CPU Usage **/tutorial/device/SiteWiseTutorialDevice1/cpu** als Eigenschaftensalias ein. Sie nehmen den Namen der AWS IoT Sache in den Eigenschaftensalias auf, sodass Sie mithilfe einer einzigen AWS IoT Regel Daten von all Ihren Geräten aufnehmen können.
 - b. Geben Sie unter Memory Usage **/tutorial/device/SiteWiseTutorialDevice1/memory** als Eigenschaftensalias ein.
9. Wählen Sie Speichern.

Wenn Sie zuvor mehrere AWS IoT Dinge erstellt haben, wiederholen Sie die Schritte 3 bis 10 für jedes Gerät und erhöhen Sie die Zahl im Assetnamen und den Eigenschaftensaliasnamen entsprechend. Beispielsweise sollte der Name der zweiten Gerätekomponente **SiteWise Tutorial Device 2** sein und die Eigenschaftensaliase sollten **/tutorial/device/SiteWiseTutorialDevice2/cpu** und **/tutorial/device/SiteWiseTutorialDevice2/memory** sein.

Schritt 6: Erstellen und konfigurieren Sie ein Geräteflotten-Asset

In diesem Verfahren erstellen Sie ein Geräteflotten-Asset, das von Ihrem Geräteflotten-Assetmodell abgeleitet ist. Anschließend verknüpfen Sie Ihre individuellen Geräte-Assets mit dem Flotten-Asset. Diese Zuordnung ermöglicht es, anhand der metrischen Eigenschaften der Flottenanlage Daten von mehreren Geräten zusammenzustellen und zu analysieren. Diese Daten bieten Ihnen einen konsolidierten Überblick über die Gesamtleistung der gesamten Flotte.

So erstellen Sie eine Geräteflottenkomponente und ordnen ihr Gerätekomponenten zu:

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Klicken Sie im linken Navigationsbereich auf Assets (Komponenten).
3. Wählen Sie dann Create asset (Komponente erstellen) aus.
4. Wählen Sie unter Modellinformationen das Asset-Modell Ihrer Geräteflotte aus **SiteWise Tutorial Device Fleet Model**.
5. Geben Sie unter Inventarinformationen einen Namen für Ihr Asset ein. Beispiel, **SiteWise Tutorial Device Fleet 1**.
6. Wählen Sie dann Create asset (Komponente erstellen) aus.
7. Wählen Sie für Ihre neue Geräteflottenkomponente Edit (Bearbeiten).

8. Wählen Sie unter Diesem Asset zugeordnete Assets die Option Verbundenes Asset hinzufügen aus und gehen Sie wie folgt vor:
 - a. Wählen Sie unter Hierarchy (Hierarchie) die Option Device aus. Diese Hierarchie identifiziert die hierarchische Beziehung zwischen Geräte- und Geräteflottenkomponenten. Sie haben diese Hierarchie im Geräteflottenkomponentenmodell früher in diesem Tutorial definiert.
 - b. Wählen Sie unter Asset (Komponente) Ihre Gerätekomponente, SiteWise Tutorial Device 1, aus.
9. (Optional) Wenn Sie zuvor mehrere Geräte-Assets erstellt haben, wiederholen Sie die Schritte 8 bis 10 für jedes Geräte-Asset, das Sie erstellt haben.
10. Wählen Sie Speichern.

Sie sollten nun Ihre Gerätekomponenten als Hierarchie sehen.

Schritt 7: Erstellen Sie in AWS IoT Core eine Regel, um Daten an Geräteressourcen zu senden

In diesem Verfahren richten Sie eine Regel in ein AWS IoT Core. Die Regel dient dazu, Benachrichtigungen aus Geräteschatten zu interpretieren und die Daten an Ihre Geräteressourcen in zu übertragen AWS IoT SiteWise. Jedes Mal, wenn der Shadow Ihres Geräts aktualisiert wird, wird eine MQTT-Nachricht AWS IoT gesendet. Sie können eine Regel erstellen, die aktiv wird, wenn sich Geräteschatten basierend auf der MQTT-Nachricht ändern. In diesem Fall besteht das Ziel darin, die Aktualisierungsnachricht zu verarbeiten, die Eigenschaftswerte zu extrahieren und sie an Ihre Geräteressourcen in AWS IoT SiteWise zu übertragen.

Um eine Regel mit einer AWS IoT SiteWise Aktion zu erstellen

1. Navigieren Sie zur [AWS IoT -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Nachrichtenweiterleitung und dann Regeln aus.
3. Wählen Sie Regel erstellen aus.
4. Geben Sie einen Namen und eine Beschreibung für Ihre Regel ein und wählen Sie dann Weiter.
5. Geben Sie die folgende SQL-Anweisung ein und wählen Sie dann Weiter.

```
SELECT
  *
FROM
  '$aws/things/+/shadow/update/accepted'
```

```
WHERE  
  startsWith(topic(3), "SiteWiseTutorialDevice")
```

Diese Regelabfrageanweisung funktioniert, weil der Geräte-Schattenservice Schattenaktualisierungen in `$aws/things/thingName/shadow/update/accepted` veröffentlicht. Weitere Informationen zu Device Shadows finden Sie unter [Device Shadow Service](#) im AWS IoT Entwicklerhandbuch.

In der WHERE-Klausel verwendet diese Regelabfrageanweisung die `topic(3)`-Funktion, um den Namen des Objekts aus dem dritten Segment des Themas abzurufen. Anschließend filtert die Anweisung Geräte heraus, die Namen haben, die nicht mit denen der Geräte für das Tutorial übereinstimmen. Weitere Informationen zu AWS IoT SQL finden Sie in der [AWS IoT SQL-Referenz](#) im AWS IoT Entwicklerhandbuch.

6. Wählen Sie unter Regelaktionen die Option Nachrichtendaten an Asset-Eigenschaften senden in aus AWS IoT SiteWise und gehen Sie wie folgt vor:
 - a. Wählen Sie By property alias (Nach Eigenschaftensalias).
 - b. Geben Sie unter Property alias (Eigenschaftensalias) **`/tutorial/device/${topic(3)}/cpu`** ein.

Die `${...}` Syntax ist eine Ersatzvorlage. AWS IoT wertet den Inhalt in den geschweiften Klammern aus. Diese Substitutionsvorlage ruft den Namen des Objekts aus dem Thema ab, um einen Alias zu erstellen, der für jedes Thema eindeutig ist. Weitere Informationen finden Sie im [Entwicklerhandbuch unter AWS IoT Substitutionsvorlagen](#).

 Note

Da ein Ausdruck in einer Substitutionsvorlage getrennt von der SELECT-Anweisung ausgewertet wird, können Sie keine Substitutionsvorlage verwenden, um auf einen Alias zu verweisen, der mit einer AS-Klausel erstellt wurde. Zusätzlich zu den unterstützten Funktionen und Operatoren können Sie nur in der ursprünglichen Nutzlast vorhandene Informationen referenzieren.

- c. Geben **`${concat(topic(3), "-cpu-", floor(state.reported.timestamp))}`** Sie im Feld Eintrags-ID — optional den Wert ein.

Der Eintrag identifiziert jeden Versuch, einen Wert einzugeben, IDs eindeutig. Wenn ein Eintrag einen Fehler zurückgibt, finden Sie die Eintrags-ID in der Fehlerausgabe, um

das Problem zu beheben. Die Substitutionsvorlage in dieser Eintrags-ID kombiniert den Namen des Objekts und den gemeldeten Zeitstempel des Geräts. Beispielsweise könnte die resultierende Eintrags-ID wie `SiteWiseTutorialDevice1-cpu-1579808494` aussehen.

- d. Geben Sie unter Time in seconds (Zeit in Sekunden) **`${floor(state.reported.timestamp)}`** ein.

Diese Substitutionsvorlage berechnet die Zeit in Sekunden aus dem gemeldeten Zeitstempel des Geräts. In diesem Tutorial melden Geräte Zeitstempel in Sekunden nach Unix-Epoche als Gleitkommazahl.

- e. Geben **`${floor((state.reported.timestamp % 1) * 1E9)}`** Sie im Feld Offset in Nanos — optional den Wert ein.

Diese Substitutionsvorlage berechnet die Verschiebung in Nanosekunden aus der Zeit in Sekunden, indem der Dezimalteil des gemeldeten Zeitstempels des Geräts konvertiert wird.

 Note

AWS IoT SiteWise setzt voraus, dass Ihre Daten einen aktuellen Zeitstempel in der Unix-Epochenzeit haben. Wenn Ihre Geräte die Zeit nicht genau melden, können Sie die aktuelle Zeit von der AWS IoT -Regelengine mit [timestamp\(\)](#) abrufen. Diese Funktion meldet die Zeit in Millisekunden. Daher müssen Sie die Zeitparameter Ihrer Regelaktion auf die folgenden Werte aktualisieren:

- Geben Sie unter Time in seconds (Zeit in Sekunden) **`${floor(timestamp() / 1E3)}`** ein.
- Geben Sie unter Offset in Nanos (Verschiebung in Nanosekunden) **`${(timestamp() % 1E3) * 1E6}`** ein.

- f. Wählen Sie unter Data type (Datentyp) die Option Double (Doppelt).

Dieser Datentyp muss mit dem Datentyp der Komponenteneigenschaft übereinstimmen, die Sie im Komponentenmodell definiert haben.

- g. Geben Sie unter Value (Wert) **`${state.reported.cpu}`** ein. In Substitutionsvorlagen verwenden Sie den `.`-Operator, um einen Wert aus einer JSON-Struktur abzurufen.
- h. Wählen Sie Add entry (Eintrag hinzufügen), um einen neuen Eintrag für die Speicherbelegungseigenschaft hinzuzufügen, und führen Sie die folgenden Schritte für diese Eigenschaft erneut aus:

- i. Wählen Sie By property alias (Nach Eigenschaftentalias).
 - ii. Geben Sie unter Property alias (Eigenschaftentalias) **/tutorial/device/\${topic(3)}/memory** ein.
 - iii. Geben Sie im Feld Eintrags-ID — optional den Wert ein. **\${concat(topic(3), "-memory-", floor(state.reported.timestamp))}**
 - iv. Geben Sie unter Time in seconds (Zeit in Sekunden) **\${floor(state.reported.timestamp)}** ein.
 - v. Geben **\${floor((state.reported.timestamp % 1) * 1E9)}** Sie im Feld Offset in Nanos — optional den Wert ein.
 - vi. Wählen Sie unter Data type (Datentyp) die Option Double (Doppelt).
 - vii. Geben Sie unter Value (Wert) **\${state.reported.memory}** ein.
 - i. Wählen Sie unter IAM-Rolle die Option Neue Rolle erstellen aus, um eine IAM-Rolle für diese Regelaktion zu erstellen. Diese Rolle ermöglicht AWS IoT die Übertragung von Daten auf Eigenschaften in Ihrer Geräteflotte und deren Asset-Hierarchie.
 - j. Geben Sie einen Rollennamen ein und wählen Sie Create.
7. (Optional) Konfigurieren Sie eine Fehleraktion, die Sie zur Problembehandlung Ihrer Regel verwenden können. Weitere Informationen finden Sie unter [Problembehandlung bei einer Regel \(AWS IoT SiteWise\)](#).
 8. Wählen Sie Weiter aus.
 9. Überprüfen Sie die Einstellungen und wählen Sie Erstellen, um die Regel zu erstellen.

Schritt 8: Führen Sie das Geräteclient-Skript aus

In diesem Tutorial verwenden Sie kein echtes Gerät, um Daten zu melden. Stattdessen führen Sie ein Skript aus, um den Geräteschatten AWS IoT Ihres Geräts mit der CPU- und Speicherauslastung zu aktualisieren, um echte Sensordaten nachzuahmen. Um das Skript auszuführen, müssen Sie zuerst die erforderlichen Python Pakete installieren. In diesem Verfahren installieren Sie die erforderlichen Python Pakete und führen dann das Geräteclientskript aus.

So konfigurieren und führen Sie das Geräte-Clientskript aus

1. Navigieren Sie zur [AWS IoT -Konsole](#).
2. Wählen Sie unten im linken Navigationsbereich Settings (Einstellungen) aus.

- Speichern Sie Ihren benutzerdefinierten Endpunkt zur Verwendung mit dem Geräte-Clientskript. Sie verwenden diesen Endpunkt, um mit den Schatten Ihres Objekts zu interagieren. Dieser Endpunkt ist eindeutig für Ihr Konto in der aktuellen Region.

Ihr benutzerdefinierter Endpunkt sollte wie im folgenden Beispiel aussehen.

```
identifizier.iot.region.amazonaws.com
```

- Öffnen Sie eine Befehlszeile, und führen Sie den folgenden Befehl aus, um zu dem zuvor erstellten Tutorialverzeichnis zu navigieren.

```
cd iot-sitewise-rule-tutorial
```

- Führen Sie den folgenden Befehl aus, um das AWS IoT-Geräte-SDK for Python zu installieren:

```
pip3 install AWSIoTPythonSDK
```

Weitere Informationen finden Sie [AWS IoT-Geräte-SDK for Python](#) im AWS IoT Entwicklerhandbuch

- Führen Sie den folgenden Befehl aus, um psutil, eine plattformübergreifende Prozess- und Systemdienstprogramm-bibliothek, zu installieren.

```
pip3 install psutil
```

Weitere Informationen finden Sie unter [psutil](#) im Python-Paketindex.

- Erstellen Sie eine Datei mit dem Namen `thing_performance.py` im Verzeichnis `iot-sitewise-rule-tutorial`, und kopieren Sie dann den folgenden Python-Code in diese Datei.

```
import AWSIoTPythonSDK.MQTTLib as AWSIoTPyMQTT

import json
import psutil
import argparse
import logging
import time

# Configures the argument parser for this program.
```

```
def configureParser():
    parser = argparse.ArgumentParser()
    parser.add_argument(
        "-e",
        "--endpoint",
        action="store",
        required=True,
        dest="host",
        help="Your AWS IoT custom endpoint",
    )
    parser.add_argument(
        "-r",
        "--rootCA",
        action="store",
        required=True,
        dest="rootCAPath",
        help="Root CA file path",
    )
    parser.add_argument(
        "-c",
        "--cert",
        action="store",
        required=True,
        dest="certificatePath",
        help="Certificate file path",
    )
    parser.add_argument(
        "-k",
        "--key",
        action="store",
        required=True,
        dest="privateKeyPath",
        help="Private key file path",
    )
    parser.add_argument(
        "-p",
        "--port",
        action="store",
        dest="port",
        type=int,
        default=8883,
        help="Port number override",
    )
    parser.add_argument(
```

```

        "-n",
        "--thingName",
        action="store",
        required=True,
        dest="thingName",
        help="Targeted thing name",
    )
    parser.add_argument(
        "-d",
        "--requestDelay",
        action="store",
        dest="requestDelay",
        type=float,
        default=1,
        help="Time between requests (in seconds)",
    )
    parser.add_argument(
        "-v",
        "--enableLogging",
        action="store_true",
        dest="enableLogging",
        help="Enable logging for the AWS IoT Device SDK for Python",
    )
    return parser

# An MQTT shadow client that uploads device performance data to AWS IoT at a
# regular interval.
class PerformanceShadowClient:
    def __init__(
        self,
        thingName,
        host,
        port,
        rootCAPath,
        privateKeyPath,
        certificatePath,
        requestDelay,
    ):
        self.thingName = thingName
        self.host = host
        self.port = port
        self.rootCAPath = rootCAPath
        self.privateKeyPath = privateKeyPath

```



```
self.certificatePath = certificatePath
self.requestDelay = requestDelay

# Updates this thing's shadow with system performance data at a regular
interval.
def run(self):
    print("Connecting MQTT client for {}".format(self.thingName))
    mqttClient = self.configureMQTTClient()
    mqttClient.connect()
    print("MQTT client for {} connected".format(self.thingName))
    deviceShadowHandler = mqttClient.createShadowHandlerWithName(
        self.thingName, True
    )

    print("Running performance shadow client for {}...
\n".format(self.thingName))
    while True:
        performance = self.readPerformance()
        print("{}".format(self.thingName))
        print("CPU:\t{}".format(performance["cpu"]))
        print("Memory:\t{}\n".format(performance["memory"]))
        payload = {"state": {"reported": performance}}
        deviceShadowHandler.shadowUpdate(
            json.dumps(payload), self.shadowUpdateCallback, 5
        )
        time.sleep(args.requestDelay)

# Configures the MQTT shadow client for this thing.
def configureMQTTClient(self):
    mqttClient = AWSIoTPyMQTT.AWSIoTMQTTShadowClient(self.thingName)
    mqttClient.configureEndpoint(self.host, self.port)
    mqttClient.configureCredentials(
        self.rootCAPath, self.privateKeyPath, self.certificatePath
    )
    mqttClient.configureAutoReconnectBackoffTime(1, 32, 20)
    mqttClient.configureConnectDisconnectTimeout(10)
    mqttClient.configureMQTTOperationTimeout(5)
    return mqttClient

# Returns the local device's CPU usage, memory usage, and timestamp.
def readPerformance(self):
    cpu = psutil.cpu_percent()
    memory = psutil.virtual_memory().percent
    timestamp = time.time()
```

```

        return {"cpu": cpu, "memory": memory, "timestamp": timestamp}

# Prints the result of a shadow update call.
def shadowUpdateCallback(self, payload, responseStatus, token):
    print("{} {}".format(self.thingName))
    print("Update request {} {} \n".format(token, responseStatus))

# Configures debug logging for the AWS IoT Device SDK for Python.
def configureLogging():
    logger = logging.getLogger("AWSIoTPythonSDK.core")
    logger.setLevel(logging.DEBUG)
    streamHandler = logging.StreamHandler()
    formatter = logging.Formatter(
        "%(asctime)s - %(name)s - %(levelname)s - %(message)s"
    )
    streamHandler.setFormatter(formatter)
    logger.addHandler(streamHandler)

# Runs the performance shadow client with user arguments.
if __name__ == "__main__":
    parser = configureParser()
    args = parser.parse_args()
    if args.enableLogging:
        configureLogging()
    thingClient = PerformanceShadowClient(
        args.thingName,
        args.host,
        args.port,
        args.rootCAPath,
        args.privateKeyPath,
        args.certificatePath,
        args.requestDelay,
    )
    thingClient.run()

```

8. Führen Sie **thing_performance.py** über die Befehlszeile mit den folgenden Parametern aus:

- **-n, --thingName** — Der Name Ihres Dings, z. **SiteWiseTutorialDevice1 B**.
- **-e, --endpoint** — Ihr benutzerdefinierter AWS IoT Endpunkt, den Sie zuvor in diesem Verfahren gespeichert haben.
- **-r, --rootCA** — Der Pfad zu Ihrem AWS IoT Root-CA-Zertifikat.

- `-c, --cert` — Der Pfad zu deinem AWS IoT Ding-Zertifikat.
- `-k, --key` — Der Pfad zu Ihrem privaten Schlüssel AWS IoT für Ihr Ding-Zertifikat.
- `-d, --requestDelay` — (Optional) Die Wartezeit in Sekunden zwischen den einzelnen Device-Shadow-Updates. Standard ist 1 Sekunde.
- `-v, --enableLogging` — (Optional) Wenn dieser Parameter vorhanden ist, druckt das Skript Debug-Meldungen von AWS IoT-Geräte-SDK for Python

Ihr Befehl sollte ähnlich wie im folgenden Beispiel aussehen.

```
python3 thing_performance.py \  
  --thingName SiteWiseTutorialDevice1 \  
  --endpoint identifier.iot.region.amazonaws.com \  
  --rootCA AmazonRootCA1.pem \  
  --cert device1/thing-id-certificate.pem.crt \  
  --key device1/thing-id-private.pem.key
```

Wenn Sie das Skript für zusätzliche AWS IoT Dinge ausführen, aktualisieren Sie den Namen und das Zertifikatsverzeichnis entsprechend.

9. Versuchen Sie, Programme auf Ihrem Gerät zu öffnen und zu schließen, um zu sehen, wie sich die CPU- und Speichernutzung ändern. Das Skript druckt jede Ablesung von CPU- und Speichernutzung. Wenn das Skript Daten erfolgreich zum Geräteschattenservice hochlädt, sollte die Ausgabe des Skripts wie im folgenden Beispiel aussehen.

```
[SiteWiseTutorialDevice1]  
CPU:    24.6%  
Memory: 85.2%  
  
[SiteWiseTutorialDevice1]  
Update request e6686e44-fca0-44db-aa48-3ca81726f3e3 accepted
```

10. Gehen Sie folgendermaßen vor, um zu überprüfen, ob das Skript den Geräteschatten aktualisiert:
 - a. Navigieren Sie zur [AWS IoT -Konsole](#).
 - b. Wählen Sie im linken Navigationsbereich Alle Geräte und dann Dinge aus.
 - c. Wähle dein Ding, SiteWiseTutorialDevice.

- d. Wählen Sie die Registerkarte Geräteschatten, wählen Sie Classic Shadow und vergewissern Sie sich, dass der Shadow-Status wie im folgenden Beispiel aussieht.

```
{
  "reported": {
    "cpu": 24.6,
    "memory": 85.2,
    "timestamp": 1579567542.2835066
  }
}
```

Wenn der Schattenstatus Ihres Dings leer ist oder nicht wie im vorherigen Beispiel aussieht, überprüfen Sie, ob das Skript ausgeführt wird und ob die Verbindung erfolgreich hergestellt wurde AWS IoT. Wenn das Skript beim Herstellen einer Verbindung zu weiterhin zu einem Timeout kommt AWS IoT, überprüfen Sie, ob Ihre [Ding-Richtlinie](#) gemäß dieser Anleitung konfiguriert ist.

11. Gehen Sie folgendermaßen vor, um zu überprüfen, ob die Regelaktion Daten an AWS IoT SiteWise sendet:
- a. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
 - b. Klicken Sie im linken Navigationsbereich auf Assets (Komponenten).
 - c. Klicken Sie auf den Pfeil neben Ihrem Geräteflotten-Asset (SiteWise Tutorial Device Fleet 1), um dessen Asset-Hierarchie zu erweitern, und wählen Sie dann Ihr Geräte-Asset (SiteWise Tutorial Device1) aus.
 - d. Wählen Sie Measurements (Messungen).
 - e. Stellen Sie sicher, dass die Zellen Latest value (Letzter Wert) Werte für die Eigenschaften CPU Usage und Memory Usage aufweisen.

Measurements				
Name	Alias	Notification status	Notification topic	Latest value
CPU Usage	/tutorial/device/SiteWiseTutorialDevice1/cpu	⊖ Disabled	–	24.6
Memory Usage	/tutorial/device/SiteWiseTutorialDevice1/memory	⊖ Disabled	–	85.2

- f. Wenn die Eigenschaften CPU Usage und Memory Usage nicht über die neuesten Werte verfügen, aktualisieren Sie die Seite. Wenn nach einigen Minuten keine Werte angezeigt werden, finden Sie weitere Informationen unter [Problembehandlung bei einer Regel \(\)AWS IoT SiteWise](#).

Sie haben dieses Tutorial abgeschlossen. Wenn Sie Live-Visualisierungen Ihrer Daten untersuchen möchten, können Sie ein Portal in AWS IoT SiteWise Monitor konfigurieren. Weitere Informationen finden Sie unter [Überwachen Sie Daten mit AWS IoT SiteWise Monitor](#). Andernfalls können Sie STRG+C in der Eingabeaufforderung betätigen, um das Geräte-Clientskript zu stoppen. Es ist unwahrscheinlich, dass das Python-Programm so viele Nachrichten sendet, dass Kosten anfallen, aber es hat sich bewährt, das Programm zu beenden, wenn Sie fertig sind.

Schritt 9: Ressourcen nach dem Tutorial bereinigen

Note

Die in diesem Tutorial erstellten Ressourcen sind für das Tutorial [Daten in SiteWise Edge integrieren](#) erforderlich. Bereinigen Sie die Ressourcen in diesem Schritt nicht, wenn Sie ihn abschließen möchten.

Nachdem Sie das Tutorial zur Erfassung von Daten aus AWS IoT Dingen abgeschlossen haben, sollten Sie Ihre Ressourcen bereinigen, um zusätzliche Kosten zu vermeiden.

So löschen Sie hierarchische Objekte in AWS IoT SiteWise

1. [Navigieren Sie zur Konsole AWS IoT SiteWise](#)
2. Klicken Sie im linken Navigationsbereich auf Assets (Komponenten).
3. Wenn Sie Elemente in löschen AWS IoT SiteWise, müssen Sie sie zunächst trennen.

Führen Sie die folgenden Schritte aus, um die Zuordnung Ihrer Gerätekomponenten zu Ihrer Geräteflottenkomponente aufzuheben:

- a. Wählen Sie Ihr Geräteflotten-Asset (SiteWise Tutorial Device Fleet 1) aus.
- b. Wählen Sie Bearbeiten aus.
- c. Wählen Sie unter Assets associated to this asset (Dieser Komponente zugeordnete Komponenten) die Option Disassociate (Zuordnung aufheben) für jede Gerätekomponente, die dieser Geräteflottenkomponente zugeordnet ist.
- d. Wählen Sie Speichern.

Sie sollten nun Ihre Gerätekomponenten nicht mehr als Hierarchie organisiert sehen.

4. Wählen Sie Ihre Gerätekomponente (SiteWise Tutorial Device 1).

5. Wählen Sie Löschen aus.
6. Geben Sie in das Bestätigungsfeld **Delete** ein, und wählen Sie dann Delete (Löschen).
7. Wiederholen Sie die Schritte 4 bis 6 für jedes Geräte-Asset und das Geräteflotten-Asset (SiteWise Tutorial Device Fleet 1).

Um hierarchische Asset-Modelle zu löschen in AWS IoT SiteWise

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wenn Sie dies noch nicht getan haben, löschen Sie Ihre Geräte- und Geräteflottenkomponenten. Weitere Informationen finden Sie im [vorhergehenden Verfahren](#). Sie können ein Modell nicht löschen, wenn Sie Komponenten haben, die aus diesem Modell erstellt wurden.
3. Wählen Sie im linken Navigationsbereich Models (Modelle) aus.
4. Wählen Sie Ihr Geräteflottenkomponentenmodell (SiteWise Tutorial Device Fleet Model).

Wenn Sie hierarchische Anlagenmodelle löschen, löschen Sie zunächst das übergeordnete Anlagenmodell.

5. Wählen Sie Löschen aus.
6. Geben Sie in das Bestätigungsfeld **Delete** ein, und wählen Sie dann Delete (Löschen).
7. Wiederholen Sie die Schritte 4 bis 6 für Ihr Gerätekomponentenmodell (SiteWise Tutorial Device Model).

Um eine Regel zu deaktivieren oder zu löschen in AWS IoT Core

1. Navigieren Sie zur [AWS IoT -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Nachrichtenweiterleitung und dann Regeln aus.
3. Wählen Sie Ihre Regel aus und klicken Sie auf Löschen.
4. Geben Sie im Bestätigungsdialogfeld den Namen der Regel ein und wählen Sie dann Löschen.

Integrieren Sie Daten mithilfe eines MQTT-fähigen V3-Gateways in SiteWise Edge

Dieses Tutorial führt Sie durch die Integration von Geräten und Sensoren von Drittanbietern, die das MQTT-Messaging-Protokoll verwenden, mit dem AWS IoT SiteWise MQTT-fähigen V3-Gateway. Sie erfahren, wie Sie ein AWS IoT SiteWise Edge-Gateway einrichten, um Daten von Ihren

MQTT-fähigen Geräten zu sammeln und zu überwachen. AWS IoT SiteWise ermöglicht es Ihnen, Daten zu Industrieanlagen zu sammeln, zu verarbeiten und zu überwachen. Nutzen Sie SiteWise Edge-Funktionen, um den industriellen IoT-Betrieb zu optimieren und Rohdaten in umsetzbare Erkenntnisse umzuwandeln.

In diesem Tutorial verwenden wir Daten aus einer Windpark-Demonstration, um wichtige Konzepte zu veranschaulichen. Nachdem Sie sich mit dem Prozess vertraut gemacht haben, können Sie das Tutorial mit Ihren eigenen Daten wiederholen.

Nachdem Sie dieses Tutorial abgeschlossen haben, können Sie die folgenden Aufgaben ausführen:

- Richten Sie ein MQTT-fähiges V3-Gateway ein und konfigurieren Sie es für den Empfang von Daten von Industriegeräten
- Verarbeiten und validieren Sie eingehende MQTT-Nachrichten von Ihren Geräten am Netzwerkrand
- Zeigen Sie Gerätedaten AWS IoT SiteWise mithilfe einer Visualisierungsplattform eines Drittanbieters an
- Senden Sie verarbeitete Daten von Ihrem Edge-Gateway an das AWS Cloud , um eine zentrale Speicherung und weitere Analysen zu ermöglichen

Darüber hinaus können Sie die Funktionen Ihres Edge-Gateways nutzen, indem Sie eine Verbindung zu anderen AWS IoT Diensten herstellen, um die folgenden Aufgaben auszuführen:

- Konfigurieren Sie AWS IoT Regeln, um Daten an Dienste wie [Amazon S3](#), [Amazon Timestream](#) und [AWS Lambda](#) weiterzuleiten.
- Verwenden Sie diese [AWS IoT Device Defender](#) Option, um Ihre Gateway-Konfigurationen aus der Ferne zu verwalten und zu aktualisieren.
- Implementieren Sie mithilfe von AWS IoT Sicherheitsfunktionen eine sichere Geräteauthentifizierung und -autorisierung. Weitere Informationen finden Sie unter [AWS IoT Sicherheit](#) im AWS IoT Entwicklerhandbuch.
- Erstellen Sie automatisierte Warnmeldungen und Benachrichtigungen auf der Grundlage von Gerätedaten. Weitere Informationen finden Sie unter [Regeln für AWS IoT](#) im AWS IoT Entwicklerhandbuch.

 Note

Dieses Tutorial bezieht sich auf Dienste, Tools und Dokumentation von Drittanbietern. AWS ist kein Anbieter oder Lieferant für Produkte oder Dienstleistungen von Drittanbietern und kann die Richtigkeit der Informationen externer Anbieter nicht garantieren. Evaluieren und validieren Sie alle Tools von Drittanbietern vor der Bereitstellung.

Themen

- [Voraussetzungen](#)
- [Schritt 1: Erstellen Sie eine AWS IoT Richtlinie](#)
- [Schritt 2: Erstellen und konfigurieren Sie ein AWS IoT Ding](#)
- [Schritt 3: Konfigurieren Sie Ihr SiteWise Edge-MQTT-fähiges V3-Gateway](#)
- [Schritt 4: Installieren Sie die SiteWise Edge-Gateway-Software](#)
- [Schritt 5: Konfigurieren Sie den EMQX-Broker für die Verbindung zu externen Anwendungen](#)
- [Schritt 6: Veröffentlichen Sie Daten mit Mosquitto](#)
- [Schritt 7: Ziele angeben](#)
- [Schritt 8: Pfadfilter angeben](#)
- [Schritt 9: Konfigurieren Sie Ihre AWS IoT-Ressourcen](#)
- [Schritt 10: Visualisieren Sie Ihre Daten](#)
- [Schritt 11: Ressourcen nach dem Tutorial bereinigen](#)
- [Weitere Ressourcen](#)

Voraussetzungen

Zum Durcharbeiten dieses Tutorials ist Folgendes erforderlich:

- Ein AWS-Konto. Falls Sie noch keines haben, beachten Sie die Informationen unter [Richten Sie eine einAWSKonto](#).
- Ein AWS Identity and Access Management (IAM-) Benutzer mit Administratorrechten. Weitere Informationen finden Sie unter [Identitäts- und Zugriffsmanagement fürAWS IoT SiteWise](#).
- Die neueste Version von ist auf Ihrem Gerät Python installiert.

 Important

Für dieses Tutorial müssen Ressourcen verwendet werden, die im Tutorial [Daten aufnehmen](#) erstellt wurden. Sie müssen es abschließen, bevor Sie mit diesem Tutorial fortfahren können.

Schritt 1: Erstellen Sie eine AWS IoT Richtlinie

In diesem Tutorial wird die AWS IoT Richtlinie verwendet, die Sie im Tutorial [Daten aufnehmen](#) erstellt haben. Diese Richtlinie legt die Sicherheitsregeln für Ihre Geräte fest und erstellt eine digitale Darstellung Ihrer externen Geräte und Sensoren in AWS IoT. Die Richtlinie ermöglicht es Ihren Geräten von Drittanbietern, Daten AWS IoT Core mithilfe von MQTT (Message Queuing Telemetry Transport) zu senden. [Weitere Informationen zu MQTT-Nachrichten finden Sie unter Was ist MQTT?](#)

Console

Stellen Sie sicher, dass eine AWS IoT Richtlinie abgeschlossen ist. Eine ausführliche Anleitung finden Sie in [Schritt 1](#) im Tutorial [Daten aufnehmen](#).

Um zu überprüfen, ob Sie über eine aktive AWS IoT Richtlinie verfügen

1. Navigieren Sie zur [AWS IoT -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Securities und dann Policies aus.
3. Wählen Sie die Richtlinie aus, die Sie erstellt haben. Beispiel, **SiteWiseTutorialDevicePolicy**.
4. Vergewissern Sie sich, dass der Status der Richtlinie als Aktiv aufgeführt ist.

AWS CLI

Stellen Sie sicher, dass eine AWS IoT Richtlinie abgeschlossen ist. Eine ausführliche Anleitung finden Sie in [Schritt 1](#) im Tutorial [Daten aufnehmen](#).

Verwenden Sie den folgenden Befehl AWS CLI [get-policy](#) in der AWS CLI Befehlsreferenz, um zu überprüfen, ob Sie über eine aktive AWS IoT Richtlinie verfügen:

```
aws iot get-policy --policy-name "SiteWiseTutorialDevicePolicy"
```

Diese Richtlinie ermöglicht es Ihren AWS IoT Geräten, mithilfe von MQTT-Nachrichten Verbindungen herzustellen und mit Geräteschatten zu kommunizieren. Um mit Geräteschatten zu interagieren, veröffentlichen und empfangen Ihre AWS IoT Geräte MQTT-Nachrichten zu Themen, die mit beginnen. `$aws/things/thing-name/shadow/` Diese Richtlinie enthält eine Ding-RichtlinienvARIABLE, die als `${iot:Connection.Thing.ThingName}`. Diese Variable ersetzt in jedem Thema den Namen der verbundenen Sache. Die `iot:Connect` Anweisung beschränkt, welche Geräte Verbindungen herstellen können, und stellt so sicher, dass die RichtlinienvARIABLE `thing` nur Namen ersetzen kann, die mit `SiteWiseTutorialDevice` beginnen.

Weitere Informationen finden Sie unter [Ding-RichtlinienvARIABLEN](#) im AWS IoT Entwicklerhandbuch.

Note

Diese Richtlinie gilt für Objekte, deren Namen mit `SiteWiseTutorialDevice` beginnen. Um einen anderen Namen für Ihre Objekte zu verwenden, müssen Sie die Richtlinie entsprechend aktualisieren.

Schritt 2: Erstellen und konfigurieren Sie ein AWS IoT Ding

Registrieren Sie in diesem Schritt Ihr Edge-Gerät als AWS IoT Ding und generieren Sie die Zertifikate und Schlüssel Ihres Dings, die für die sichere Kommunikation mit AWS IoT SiteWise Edge benötigt werden. Dieser Prozess bildet die Grundlage dafür, dass Ihr Gerät Daten von Drittanbietern über Ihr MQTT-fähiges V3-Gateway senden kann.

Console

Stellen Sie sicher, dass die Erstellungs- und Konfigurationsschritte für eine Sache abgeschlossen sind. AWS IoT Eine ausführliche Anleitung finden Sie in [Schritt 2](#) im Tutorial [Daten aufnehmen](#).

Um zu überprüfen, ob Sie ein aktives AWS IoT Objekt haben

1. Navigieren Sie zur [AWS IoT -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Alle Geräte und dann Dinge aus.
3. Wählen Sie das Ding aus, das Sie erstellt haben. Beispiel, **SiteWiseTutorialDevice1**.
4. Vergewissern Sie sich, dass der Status unter Zertifikate als aktiv aufgeführt ist.

AWS CLI

Stellen Sie sicher, dass die Erstellungs- und Konfigurationsschritte für ein Objekt AWS IoT abgeschlossen sind. Eine ausführliche Anleitung finden Sie in [Schritt 2](#) im Tutorial [Daten aufnehmen](#).

Verwenden Sie den folgenden AWS CLI Befehl, um zu überprüfen, ob Sie über eine aktive AWS IoT Richtlinie verfügen:

```
aws iot describe-thing --thing-name "SiteWiseTutorialDevice1"
```

Nachdem Sie diese Schritte abgeschlossen haben, können Sie Ihr Gerät sicher mit AWS IoT SiteWise Edge verbinden. Sie haben ein lokales Verzeichnis zum Speichern Ihrer Zertifikate und Schlüssel erstellt, die Sie für die MQTT-Authentifizierung generiert haben. Ihr Gerät ist als Objekt in AWS IoT der [AWS IoT Konsole](#) registriert und Ihr Gerät ist bereit, Daten in SiteWise Edge zu integrieren. Sie können Ihre Industrieanlagen oder andere Geräte an die AWS IoT Plattform anschließen und mit der Datenaufnahme in SiteWise Edge beginnen.

Schritt 3: Konfigurieren Sie Ihr SiteWise Edge-MQTT-fähiges V3-Gateway

Erstellen Sie in diesem Schritt Ihr AWS IoT SiteWise Edge-MQTT-fähiges V3-Gateway und konfigurieren Sie es für den Empfang von Daten vom EMQX-Broker. Das Gateway fungiert als Brücke zwischen Ihren Geräten und AWS IoT. Auf diese Weise können Sie Daten lokal am Edge verarbeiten, bevor Sie sie an den senden AWS Cloud. Diese Konfiguration reduziert die Bandbreite und verringert die Verzögerungen bei der Cloud-Verarbeitung.

Console

Um Ihr AWS IoT SiteWise MQTT-fähiges V3-Gateway zu erstellen

1. Melden Sie sich bei der [AWS-Managementkonsole](#) an, und öffnen Sie die [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Edge-Gateways und dann Create Gateway aus.
3. Wählen Sie unter Bereitstellungsziel die Option Selbst gehostetes Gateway aus.
4. Wählen Sie unter Optionen für selbst gehostetes Gateway die Option MQTT-fähig, V3-Gateway — empfohlen aus.
5. Unter Gateway-Konfiguration:

- a. Geben Sie im Feld Gateway-Name einen Namen für Ihr Gateway ein. Beispiel, **SiteWise Tutorial Device Gateway**.
 - b. Wählen Sie unter Greengrass Device OS die entsprechende Option für Ihr Gerät aus.
6. Unter Erweiterte Konfiguration:
- a. Wählen Sie Standard-Setup.
 - b. Geben Sie einen Namen für das Greengrass-Core-Gerät ein oder verwenden Sie den von AWS IoT SiteWise generierten Namen.
7. Wählen Sie Create gateway (Gateway erstellen).
8. Wählen Sie im Bestätigungsdialogfeld Generieren und herunterladen, um ein Installationsprogramm für Ihr SiteWise Edge-Gateway zu generieren. Weitere Informationen finden Sie unter [Erstellen eines selbst gehosteten SiteWise Edge-Gateways](#).

 Warning

Speichern Sie die Installationsdatei an einem sicheren Ort. Diese Datei kann nicht neu generiert werden und wird benötigt, um das Gateway-Setup in späteren Schritten abzuschließen.

AWS CLI

Wird verwendet AWS CLI , um ein selbst gehostetes Gateway zu erstellen. Sie müssen einen Namen für das Gateway angeben und die Plattform und die Gateway-Version angeben. Weitere Informationen finden Sie unter [CreateGateway](#) in der AWS IoT SiteWise -API-Referenz.

Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie die Platzhalter für Benutzereingaben durch Ihre Informationen.

```
aws iotsitewise create-gateway \  
  --gateway-name SiteWise Tutorial Device Gateway \  
  --gateway-platform greengrassV2={coreDeviceThingName=your-core-device-thing-name, coreDeviceOperatingSystem=LINUX_AMD64} \  
  --gateway-version 3 \  
  [--cli-input-json your-configuration]
```

- `gateway-name`— Ein eindeutiger Name für das Gateway, zum Beispiel `SiteWise Tutorial Device Gateway`.
- `gateway-platform`— Geben Sie `greengrassV2`. Weitere Informationen finden Sie unter [CreateGateway](#) in der AWS IoT SiteWise -API-Referenz.
 - `coreDeviceThingName`— Der Name der AWS IoT Sache für Ihr AWS IoT Greengrass V2 Kerngerät. Beispiel, `SiteWiseTutorialDevice1`.
 - `coreDeviceOperatingSystem`— Das Betriebssystem des Kerngeräts in AWS IoT Greengrass V2. Für Gateway-Version 3 ist die Angabe des Betriebssystems erforderlich. Zu den Optionen gehören: `LINUX_AARCH64`, und `LINUX_AMD64`. `WINDOWS_AMD64`
- `gateway-version`— Die Version des Gateways.
 - Wird 3 für die Gateway-Version verwendet, um ein MQTT-fähiges V3-Gateway zu erstellen.
- `cli-input-json`— Eine JSON-Datei mit Anforderungsparametern.

Verwenden Sie den folgenden AWS CLI Befehl, um zu überprüfen, ob Ihr Gateway erfolgreich erstellt wurde:

```
aws iotsitewise describe-gateway --gateway-id your-gateway-id
```

Schritt 4: Installieren Sie die SiteWise Edge-Gateway-Software

Verwenden Sie zur Installation der Gateway-Software das Installationspaket, das Sie im vorherigen Schritt heruntergeladen haben. Der Installationsvorgang konfiguriert die erforderlichen Komponenten, startet den Greengrass-Core-Service und registriert Ihr Gerät bei. AWS IoT Greengrass Stellen Sie nach Abschluss der Installation sicher, dass Ihr Gateway in der [AWS IoT SiteWise Konsole](#) unter Edge-Gateways angezeigt wird und dass der Greengrass-Dienst auf Ihrem Gerät ordnungsgemäß ausgeführt wird.

Detaillierte Anweisungen finden Sie unter [Installieren Sie das AWS IoT SiteWise Edge-Gateway-Software auf Ihrem lokalen Gerät](#).

Schritt 5: Konfigurieren Sie den EMQX-Broker für die Verbindung zu externen Anwendungen

Note

Sie müssen Ihr SiteWise Edge-MQTT-fähiges V3-Gateway bereitgestellt haben, bevor Sie fortfahren können. Das Gateway bietet die erforderlichen Infrastruktur- und Sicherheitseinstellungen, die für die Konfiguration des EMQX-Brokers erforderlich sind. Die Broker-Konfiguration schlägt ohne eine aktive Gateway-Bereitstellung fehl.

Konfigurieren Sie den EMQX-Broker, um eine sichere Kommunikation zwischen Ihren IoT-Geräten und externen Anwendungen zu ermöglichen. Der EMQX-Broker fungiert als zentraler Messaging-Hub, der Daten zwischen Ihren IoT-Geräten, Gateways und Anwendungen weiterleitet. Der EMQX-Broker gewährleistet eine zuverlässige Nachrichtenzustellung auf Ihrem Gateway und den verbundenen Anwendungen am Edge. Weitere Informationen finden Sie unter [Connect externe Anwendungen mit dem EMQX-Broker](#).

Um den EMQX-Broker zu konfigurieren

1. Richten Sie den EMQX-Broker ein. Ausführliche Anweisungen zur Konfiguration finden Sie in den Schritten 1—14 unter [Aktualisieren der EMQX-Bereitstellungskonfiguration für die Authentifizierung](#).
2. Richten Sie MQTT-Themen für die Windparküberwachung ein. Weitere Informationen zu den MQTT-Anforderungen finden Sie unter [Anforderungen an das MQTT-Thema](#)
 - a. CPU-Auslastung: SiteWiseTutorialDevice/cpu
 - b. Speichernutzung: SiteWiseTutorialDevice/memory
 - c. Zeitstempel: SiteWiseTutorialDevice/timestamp
3. Überprüfen Sie Ihre Konfiguration und schließen Sie die Bereitstellung ab.
 - a. Wählen Sie Bestätigen, um Ihre Einstellungen zu speichern.
 - b. Wählen Sie Weiter, bis Sie zum Schritt Überprüfen gelangen.
 - c. Wählen Sie auf der Seite Review (Prüfen) die Option Deploy (Bereitstellen) aus.
 - d. Warten Sie, bis die Bereitstellung erfolgreich abgeschlossen wurde, bevor Sie fortfahren.

4. Bereiten Sie Nachrichten im Payload-Format vor, um sie an den EMQX-Broker zu senden. Weitere Informationen zur Strukturierung von Payloads finden Sie unter [Aktualisieren der EMQX-Bereitstellungskonfiguration für die Authentifizierung](#).
5. Implementieren Sie die folgenden Sicherheitsmaßnahmen:
 - a. Verwenden Sie die TLS-Verschlüsselung (Transport Layer Security) (Port 8833), um Daten bei der Übertragung zu schützen. Weitere Informationen finden Sie unter [Konfigurieren Sie TLS für sichere Verbindungen zum EMQX-Broker auf AWS IoT SiteWise Edge](#).
 - b. Richten Sie die Authentifizierung mit Benutzername und Passwort ein, um die Geräteidentitäten zu überprüfen. Diese Sicherheitsmaßnahme trägt zum Schutz Ihrer Daten bei und stellt sicher, dass nur autorisierte Geräte eine Verbindung zu Ihrem System herstellen können. Weitere Informationen finden Sie unter [Aktivieren Sie die Authentifizierung mit Benutzername und Passwort](#).

Mit EMQX können Sie Autorisierungsregeln erstellen, die auf Kennungen wie Benutzername, IP-Adresse oder Client-ID basieren. Dies ist nützlich, um den Zugriff auf Ihre Daten zu kontrollieren. Weitere Informationen finden Sie unter [Richten Sie Autorisierungsregeln ein für AWS IoT SiteWise Edge in EMQX](#).

Nach erfolgreicher Bereitstellung ist Ihr EMQX-Broker konfiguriert und bereit, eine sichere Verbindung mit externen Anwendungen herzustellen.

 Note

Das Payload-Format muss einer bestimmten Struktur folgen, damit AWS IoT SiteWise Edge Ihre Daten ordnungsgemäß verarbeiten und aufnehmen kann. Weitere Informationen zur erforderlichen Struktur finden Sie unter [Struktur der JSON-Nutzlast](#)

Beispiel: Fügen Sie JSON-Payloads für CPU, Arbeitsspeicher und Zeitstempel hinzu

CPU-JSON-Nutzlast

```
{
  "propertyAlias": "SiteWiseTutorialDevice/cpu",
  "propertyValues": [
    {
      "quality": "GOOD",
      "timestamp": {
```

```
        "offsetInNanos": 0,
        "timeInSeconds": 1753206441
    },
    "value": {
        "integerValue": 45.2
    }
}
]
```

JSON-Nutzlast des Speichers

```
{
  "propertyAlias": "SiteWiseTutorialDevice/memory",
  "propertyValues": [
    {
      "quality": "GOOD",
      "timestamp": {
        "offsetInNanos": 0,
        "timeInSeconds": 1753206441
      },
      "value": {
        "integerValue": 67.8
      }
    }
  ]
}
```

JSON-Nutzlast mit Zeitstempel

```
{
  "propertyAlias": "SiteWiseTutorialDevice/timestamp",
  "propertyValues": [
    {
      "quality": "GOOD",
      "timestamp": {
        "offsetInNanos": 0,
        "timeInSeconds": 1753206441
      },
      "value": {
        "integerValue": 23.5
      }
    }
  ]
}
```



```
]
}
```

Note

Jede JSON-Nutzlast muss separat als einzelne Nachricht veröffentlicht werden. Kombinieren Sie nicht mehrere Eigenschaftswerte in einer einzigen Nachricht. Senden Sie jede CPU-, Speicher- und Zeitstempel-Payload als eigene MQTT-Publikation.

Die Payload definiert die erforderliche JSON-Struktur, die Ihre IoT-Geräte verwenden müssen, um Gerätedaten über den EMQX-Broker an Edge zu senden. SiteWise Dieses Format stellt sicher, dass Ihre Geräte identifiziert und die Sensorwerte verarbeitet werden AWS IoT SiteWise können. Nachdem Sie diese Konfigurationen und Nutzlaststrukturen implementiert haben, ist Ihr Windpark-Überwachungssystem bereit, Daten zu sammeln und zu verarbeiten.

Schritt 6: Veröffentlichen Sie Daten mit Mosquitto

Nachdem Sie Ihr MQTT-fähiges V3-Gateway erstellt haben, konfigurieren Sie Eclipse Mosquitto so, dass Testdaten an Edge gesendet werden. SiteWise Mosquitto ist ein Open-Source-MQTT-Nachrichtenbroker, der das MQTT-Protokoll für einfache Nachrichtenübermittlung zwischen Geräten verwendet. Mit dem Mosquitto-Client können Sie Nachrichten zu MQTT-Themen veröffentlichen und dabei Daten von Windparksensoren simulieren. Simulieren Sie mit Mosquitto Gerätedaten, ohne dass Dienste oder zusätzliche Geräte von Drittanbietern erforderlich sind. Weitere Informationen finden Sie in der [Dokumentation](#) auf der offiziellen Eclipse Mosquitto-Website. In diesem Tutorial werden lokale Daten aus dem Tutorial [Daten aufnehmen und fiktive Daten](#) zu Demonstrationszwecken verwendet.

Verwenden Sie den Mosquitto CLI-Client, um den SiteWise Edge EMQX-Broker zu testen

1. Installieren Sie Mosquitto auf Ihrem lokalen Gerät. Eine ausführliche Anleitung finden [Sie unter Mosquitto herunterladen auf der offiziellen Eclipse Mosquitto-Website](#).
2. Weitere Informationen zum Anschließen externer Anwendungen zur Übertragung von Industriedaten finden Sie unter [Connect externe Anwendungen mit dem EMQX-Broker](#)

Important

Stellen Sie sicher, dass die MQTT-Verbindungseinstellungen, die Sie hier konfigurieren, mit den Einstellungen übereinstimmen, die im Befehl Mosquitto publish verwendet wurden. Der

Host muss die IP-Adresse oder der Hostname Ihres Edge-Gateways sein. SiteWise Der Port ist normalerweise 1883 (oder 8883 bei Verwendung von SSL/TLS).

Verwenden Sie Mosquitto, um Testdaten zu veröffentlichen. Öffnen Sie eine Befehlszeile und führen Sie die folgenden Befehle aus:

Beispiel: CPU-Eigenschaft

```
mosquitto_pub -h localhost -p 1883 -t "SiteWiseTutorialDevice/cpu" -m '{
  "propertyAlias": "SiteWiseTutorialDevice/cpu",
  "propertyValues": [
    {
      "quality": "GOOD",
      "timestamp": {
        "timeInSeconds": 1753206441,
        "offsetInNanos": 0
      },
      "value": {
        "integerValue": 45.2
      }
    }
  ]
}'
```

Beispiel: Speichereigenschaft

```
mosquitto_pub -h localhost -p 1883 -t "SiteWiseTutorialDevice/memory" -m '{
  "propertyAlias": "SiteWiseTutorialDevice/memory",
  "propertyValues": [
    {
      "quality": "GOOD",
      "timestamp": {
        "timeInSeconds": 1753206441,
        "offsetInNanos": 0
      },
      "value": {
        "integerValue": 72.1
      }
    }
  ]
}'
```

Beispiel: Eigenschaft Timestamp

```
mosquitto_pub -h localhost -p 1883 -t "SiteWiseTutorialDevice/timestamp" -m '{
  "propertyAlias": "SiteWiseTutorialDevice/timestamp",
  "propertyValues": [
    {
      "quality": "GOOD",
      "timestamp": {
        "timeInSeconds": 1753206441,
        "offsetInNanos": 0
      },
      "value": {
        "integerValue": 1683000000
      }
    }
  ]
}'
```

Note

Die Verwendung von localhost als EMQX-Brokeradresse dient nur zu Demonstrationszwecken. In Produktionsumgebungen oder beim Herstellen einer Verbindung von externen Geräten müssen Sie die entsprechende EMQX-Broker-Adresse für Ihre spezifische Bereitstellungskonfiguration verwenden. Ausführliche Verbindungsanweisungen finden Sie unter [Connect eine Anwendung mit dem EMQX-Broker auf AWS IoT SiteWise Edge](#)

Schritt 7: Ziele angeben

Geben Sie in diesem Schritt Ziele an, um zu bestimmen, wohin Ihre Quelldaten geleitet werden sollen. Verwenden Sie AWS IoT SiteWise Amazon S3 S3-Pufferung als Ziel. Diese Option bietet eine skalierbare Möglichkeit, Ihre IoT-Daten zu speichern und zu verarbeiten.

Console

Um Ziele hinzuzufügen

1. Navigieren Sie zur [AWS IoT SiteWise Konsole](#) und wählen Sie Edge-Gateways aus.
2. Wählen Sie unter SiteWise Tutorial Device Gateway die Option Ziele hinzufügen aus.

3. Wählen Sie unter Zieldetails die Option Mit Amazon S3 AWS IoT SiteWise gepuffert aus. Weitere Informationen zu Zieltypen finden Sie unter [AWS IoT SiteWise Gateway-Ziele](#).
4. Geben Sie unter Zielname einen Namen für Ihr Ziel ein, SiteWise Tutorial S3 Destination z. B.
5. Geben Sie unter S3-Upload-Einstellungen Ihren S3-Bucket-Standort ein. Beispiel, `s3://sitewise-tutorial-mqtt-data-[your-account-id]`. Weitere Informationen zu Amazon S3 finden Sie unter [Erstellen, Konfigurieren und Arbeiten mit Amazon S3 S3-Buckets](#) im Amazon Simple Storage Service-Benutzerhandbuch.
6. Geben Sie unter Häufigkeit des Daten-Uploads einen Wert zwischen 1 Minute und 30 Tagen ein. Beispiel, `1 minute`.
7. Unter Datenspeichereinstellungen:
 - Deaktivieren Sie die Option Daten in Speicher kopieren. Diese Einstellung wird zwar für Produktionsumgebungen empfohlen, für dieses Tutorial benötigen Sie sie jedoch nicht. Wenn Sie diese Option deaktivieren, wird die Option Daten aus S3 löschen automatisch deaktiviert.
8. Wählen Sie Add destination (Ziel hinzufügen).

 Note

In diesem Tutorial wird ein Intervall von 1 Minute zum Testen verwendet. Nachdem Sie das Tutorial abgeschlossen haben, können Sie dieses Intervall an Ihre Produktionsanforderungen anpassen oder es löschen, um zusätzliche Kosten zu vermeiden.

AWS CLI

Beispiel: Erstellen Sie ein neues AWS IoT SiteWise Ziel, das mit Amazon S3 gepuffert wurde

Verwenden Sie den [update-gateway-capability-configuration](#) in der AWS CLI Befehlsreferenz angegebenen, um den Herausgeber zu konfigurieren. Stellen Sie den Parameter `capabilityNamespace` auf `iotsitewise:publisher:3` ein.

```
{  
  "sources": [  

```

```

{
  "type": "MQTT"
},
"destinations": [
  {
    "type": "SITEWISE_BUFFERED",
    "name": "your-s3-destination-name",
    "config": {
      "targetBucketArn": "arn:aws:s3:::amzn-s3-demo-bucket/Optional/SomeFolder",
      "publishPolicy": {
        "publishFrequency": "1m",
        "localSizeLimitGB": 10
      },
      "siteWiseImportPolicy": {
        "enableSiteWiseStorageImport": true,
        "enableDeleteAfterImport": true,
        "bulkImportJobRoleArn": "arn:aws:iam::123456789012:role/your-role-name"
      }
    },
    "filters": [
      {
        "type": "PATH",
        "config": {
          "paths": [
            "#"
          ]
        }
      }
    ]
  }
]
}

```

Weitere Informationen zu Zielen finden Sie unter [Fügen Sie ein AWS IoT SiteWise gepuffertes Ziel mit Amazon S3 hinzu](#).

Schritt 8: Pfadfilter angeben

In diesem Schritt konfigurieren Sie Pfadfilter, um anzugeben, welche MQTT-Themen für Ihre Windpark-Gerätedaten überwacht werden sollen.

Pfadfilter folgen der Platzhalterspezifikation für MQTT-Themen, die zwei Sonderzeichen unterstützt:

- **+—** Dieses Symbol steht für einen Platzhalter mit einer Ebene, der einer beliebigen Zeichenfolge auf einer einzelnen Ebene entspricht.
- **#—** Dieses Symbol steht für einen Platzhalter mit mehreren Ebenen, der einer beliebigen Anzahl von Ebenen in der Themenhierarchie entspricht.

 Note

Weitere Informationen zu anderen Pfadfiltern finden Sie unter [Sonderzeichen in Pfadfilternamen](#)

Console

So konfigurieren Sie Ihre Pfadfilter

Unter Pfadfilter:

1. Navigieren Sie zur [AWS IoT SiteWise Konsole](#) und wählen Sie Edge-Gateways aus.
2. Wählen Sie unter SiteWise Tutorial Device Gateway die Option Ziele hinzufügen aus.
3. Wählen Sie Pfadfilter hinzufügen, um die folgenden Pfadfilter manuell einzugeben:
 - SiteWiseTutorialDevice/#
 - windfarm/+/turbine/+/performance/#
 - cpu/+/idle-time
 - cpu/+/interruption-count/+
 - +/memory/consumption
 - timestamp/+/measurement
 - device/+/status/+
 - system/+/performance-log
4. Wählen Sie Add destination (Ziel hinzufügen).

Weitere Informationen zu bewährten Methoden für Pfadfilter finden Sie unter [Bewährte Methoden für Pfadfilter](#).

AWS CLI

Verwenden Sie die folgenden AWS CLI Befehle, um Ihre Pfadfilter zu konfigurieren:

Beispiel 1: Gerätedaten mit Platzhaltern

```
{
  "destinations": [
    {
      "name": "All Device Data Destination"
    }
  ],
  "filters": [
    {
      "type": "PATH",
      "config": {
        "paths": [
          "SiteWiseTutorialDevice/#",
          "windfarm/*/turbine/*/performance/#"
        ]
      }
    }
  ]
}
```

Diese Pfadfilterkonfiguration verwendet Platzhalter mit mehreren Ebenen (#), um alle Daten aus der Turbine SiteWiseTutorialDevice und alle Leistungsdaten von beliebigen Turbinen im Windpark zu erfassen.

Beispiel 2: CPU- und Speicherleistung

```
{
  "destinations": [
    {
      "name": "Performance Metrics Destination"
    }
  ],
  "filters": [
    {
      "type": "PATH",
      "config": {
        "paths": [
```

```
        "cpu/+/idle-time",
        "+/memory/consumption",
        "cpu/+/interruption-count/+"
    ]
}
]
```

In diesem Beispiel werden verschiedene CPU-Metriken (Leerlaufzeit und Anzahl der Unterbrechungen) und Daten zum Speicherverbrauch geräteübergreifend erfasst.

Beispiel 3: Gerätediagnose

```
{
  "destinations": [
    {
      "name": "Device Diagnostics Destination"
    }
  ],
  "filters": [
    {
      "type": "PATH",
      "config": {
        "paths": [
          "device/+/status/+",
          "system/+/performance-log"
        ]
      }
    }
  ]
}
```

Diese Konfiguration verwendet den + Platzhalter, um Diagnosedaten von mehreren Geräten zu erfassen, insbesondere Systemleistungsprotokolle und Gerätestatusaktualisierungen.

Diese drei Pfadfilter entsprechen den MQTT-Themen, die Sie verwenden, um Testdaten mit Mosquitto zu veröffentlichen. Die Filter stellen sicher, dass Ihr SiteWise Edge-Gateway die relevanten MQTT-Nachrichten erfasst und verarbeitet. Weitere Informationen zum Hinzufügen von Pfadfiltern finden Sie unter [Fügen Sie Pfadfilter zu AWS IoT SiteWise Edge-Zielen hinzu](#).

Schritt 9: Konfigurieren Sie Ihre AWS IoT-Ressourcen

In diesem Schritt erstellen Sie die erforderlichen AWS IoT SiteWise Anlagenmodelle und Ressourcen, um Ihre simulierten Geräte von Drittanbietern darzustellen und die Datenaufnahme über Ihr Edge-Gateway zu ermöglichen.

Bevor Sie mit diesem Schritt beginnen, sollten Sie die Schritte 3 bis 8 des Lernprogramms Daten aufnehmen abgeschlossen haben. In diesen Schritten werden die grundlegenden Komponenten für die Integration Ihrer Drittanbieterdaten über das MQTT-fähige V3-Gateway festgelegt. Außerdem richten Sie Regeln ein, die definieren, wie Ihre Sensordaten in das System einfließen AWS IoT SiteWise, und führen ein Geräteclient-Skript aus, das industrielle Windparkdaten simuliert.

Um Ihre AWS IoT-Ressourcenkonfiguration zu validieren

1. Verwenden Sie den folgenden AWS CLI Befehl, um zu überprüfen, ob Sie Ihr Tutorial-Gerätemodell und Ihr SiteWise Tutorial-Geräteflottenmodell erstellt und SiteWise ordnungsgemäß konfiguriert haben:

```
aws iotsitewise describe-asset-model --asset-model-id your-device-model-id
```

Verwenden Sie den folgenden AWS CLI Befehl, um die ID Ihrer Objektmodelle abzurufen:

```
aws iotsitewise list-asset-models
```

2. Verwenden Sie den folgenden AWS CLI Befehl, um zu überprüfen, ob Sie Ihr Tutorial Device 1-Asset und Ihr SiteWise SiteWise Tutorial Device Fleet 1-Asset erstellt und ordnungsgemäß konfiguriert haben:

```
aws iotsitewise describe-asset --asset-id your-asset-id
```

Verwenden Sie den folgenden AWS CLI Befehl, um die ID Ihrer Assets abzurufen:

```
aws iotsitewise list-assets
```

Schritt 10: Visualisieren Sie Ihre Daten

Richten Sie die Open-Source-Version von Grafana ein, um die Gerätedaten Ihres Windparks zu visualisieren. Grafana ist eine Visualisierungsplattform, die Ihre Betriebsdaten in Echtzeit

anzeigt. Mithilfe dieser Dashboards können Sie die betriebliche Effizienz verfolgen und den Wartungsbedarf in Ihrer gesamten Infrastruktur ermitteln. Weitere Informationen zur Integration finden Sie unter [Integrieren Sie AWS IoT SiteWise mit Grafana](#).

Um Grafana einzurichten

1. Anweisungen zum Herunterladen und Installieren der neuesten Version von Grafana finden [Sie unter Grafana installieren auf der offiziellen Grafana-Website](#).
2. Detaillierte Konfigurationsanweisungen für Ihr Betriebssystem finden Sie unter [Grafana konfigurieren](#) auf der offiziellen Grafana-Website.
3. Konfigurieren Sie die AWS IoT SiteWise Datenquelle. Dadurch können Sie das AWS IoT SiteWise Plugin auf Ihrem Grafana einrichten. Eine ausführliche Anleitung zur Verwendung des Plug-ins finden Sie unter [Connect zu einer AWS IoT SiteWise Datenquelle](#) herstellen im Amazon Managed Grafana-Benutzerhandbuch.

 Important

Stellen Sie sicher, dass Sie über die neueste Version von Grafana verfügen, um die Kompatibilität mit der AWS IoT SiteWise Datenquelle zu gewährleisten.

Nachdem Sie diese Schritte abgeschlossen haben, können Sie Grafana-Dashboards erstellen und anpassen, um die Betriebskennzahlen Ihres Windparks anzuzeigen. Auf diese Weise können Sie die Leistung Ihres Windparks an der Peripherie in Echtzeit verfolgen und analysieren.

 Note

Dieses Tutorial verwendet zwar die Open-Source-Version von Grafana, bietet AWS aber auch Amazon Managed Grafana für Produktionsumgebungen. Amazon Managed Grafana ist ein vollständig verwalteter Service, der die Einrichtung, Konfiguration und Wartung Ihrer eigenen Grafana-Server überflüssig macht.

Erwägen Sie ein Upgrade auf Amazon Managed Grafana, wenn Sie bereit sind, Ihre Lösung zu skalieren. Detaillierte Anweisungen zum Verbinden Ihrer SiteWise Daten mit Grafana finden Sie im Tutorial [Daten in Grafana visualisieren und teilen](#).

Sie haben das Tutorial abgeschlossen. In diesem Verfahren haben Sie AWS IoT SiteWise Edge so konfiguriert, dass Gerätedaten von Drittanbietern mithilfe eines MQTT-fähigen V3-Gateways integriert werden. Dieses Setup ermöglicht es Ihnen, Daten zu Industrieanlagen am Netzwerkrand zu sammeln, zu verarbeiten und zu visualisieren, wodurch Latenz und Betriebskosten reduziert werden. Mithilfe der Windpark-Demo haben Sie Betriebsmetriken wie CPU- und Speicherauslastungsdaten über Ihr MQTT-fähiges V3-Gateway gesammelt und verarbeitet.

Um Ihre IoT-Lösung zu verbessern, sollten Sie erwägen, erweiterte Funktionen wie die Erkennung von Anomalien durch Nutzung [Erkennen Sie Anomalien mit Lookout for Equipment](#) oder Integration mit anderen AWS Diensten wie [Amazon Quick im Amazon Quick](#) User Guide für erweiterte Analysen zu erkunden.

Schritt 11: Ressourcen nach dem Tutorial bereinigen

Nachdem Sie dieses Tutorial zur Integration von Daten in AWS IoT SiteWise Edge abgeschlossen haben, sollten Sie Ihre Ressourcen bereinigen, um zusätzliche Kosten zu vermeiden.

So löschen Sie hierarchische Assets in AWS IoT SiteWise

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Klicken Sie im linken Navigationsbereich auf Assets (Komponenten).
3. Wenn Sie Elemente in löschen AWS IoT SiteWise, müssen Sie sie zunächst trennen.

Führen Sie die folgenden Schritte aus, um die Zuordnung Ihrer Gerätekomponenten zu Ihrer Geräteflottenkomponente aufzuheben:

- a. Wählen Sie Ihr Geräteflotten-Asset (SiteWise Tutorial Device Fleet 1) aus.
- b. Wählen Sie Bearbeiten aus.
- c. Wählen Sie unter Assets associated to this asset (Dieser Komponente zugeordnete Komponenten) die Option Disassociate (Zuordnung aufheben) für jede Gerätekomponente, die dieser Geräteflottenkomponente zugeordnet ist.
- d. Wählen Sie Speichern.

Note

Die Geräte-Assets sind jetzt nicht mehr hierarchisch organisiert.

4. Wählen Sie Ihre Gerätekomponente (SiteWise Tutorial Device 1).

5. Wählen Sie Löschen aus.
6. Geben Sie im Bestätigungsdialogfeld die Eingabe ein **Delete**, und wählen Sie dann Löschen.
7. Wiederholen Sie die Schritte 4 bis 6 für jedes Geräte-Asset und das Geräteflotten-Asset (SiteWise Tutorial Device Fleet 1).

Um hierarchische Objektmodelle zu löschen AWS IoT SiteWise

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Löschen Sie Ihr Gerät und Ihre Geräteflotten-Assets.
3. Wählen Sie im linken Navigationsbereich Models (Modelle) aus.
4. Wählen Sie Ihr Geräteflottenkomponentenmodell (SiteWise Tutorial Device Fleet Model). Sie können ein Modell nicht löschen, wenn Sie Komponenten haben, die aus diesem Modell erstellt wurden.

Wenn Sie hierarchische Asset-Modelle löschen, löschen Sie zunächst das übergeordnete Asset-Modell.

5. Wählen Sie Löschen aus.
6. Geben Sie im Bestätigungsdialogfeld die Eingabe ein **Delete**, und wählen Sie dann Löschen.
7. Wiederholen Sie die Schritte 4 bis 6 für Ihr Gerätekomponentenmodell (SiteWise Tutorial Device Model).

Um eine Regel zu deaktivieren oder zu löschen in AWS IoT Core

1. Navigieren Sie zur [AWS IoT -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Nachrichtenweiterleitung und dann Regeln aus.
3. Wählen Sie Ihre Regel aus und klicken Sie auf Löschen.
4. Geben Sie im Bestätigungsdialogfeld den Namen der Regel ein und wählen Sie dann Löschen.

Um einen Amazon S3 S3-Bucket zu löschen

1. Navigieren Sie zur [Amazon S3 S3-Konsole](#).
2. Wählen Sie im linken Navigationsbereich die Option Allzweck-Bucket aus.
3. Wählen Sie in der Bucket-Liste das Optionsfeld neben dem Bucket aus, den Sie erstellt haben, und wählen Sie dann oben auf der Seite Leer aus.

4. Bestätigen Sie im Bestätigungsdialogfeld den Löschvorgang und wählen Sie dann Leeren aus.
5. Wenn der Bucket leer ist, wählen Sie Löschen, um den Bucket zu löschen.
6. Geben Sie im Bestätigungsdialogfeld den Namen Ihres Buckets ein, um das Löschen zu bestätigen.
7. Wählen Sie Bucket löschen aus.

Um ein SiteWise Edge-Gateway zu löschen

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Edge-Gateways aus.
3. Wählen Sie unter Gateways das Gateway aus, das Sie für dieses Tutorial erstellt haben. Beispiel, SiteWise Tutorial Device Gateway.
4. Wählen Sie Löschen aus.
5. Um zu bestätigen, dass Sie das Gateway löschen möchten, geben Sie den Text **Delete** in das Bestätigungsdialogfeld ein und wählen Sie dann im angezeigten Fenster Löschen aus.

Um dein IoT-Ding zu löschen

1. Navigieren Sie zur [AWS IoT -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Verwalten und dann Dinge aus.
3. Wählen Sie das IoT-Ding aus, das Sie für dieses Tutorial erstellt haben. Beispiel, SiteWiseTutorialDevice1.
4. Wählen Sie Löschen aus.
5. Geben Sie im Bestätigungsdialogfeld den Namen des Dings ein und wählen Sie dann Löschen.

Um AWS IoT Greengrass Core zu deinstallieren

Deinstallieren Sie die AWS IoT Greengrass Core-Software von Ihrem lokalen Gerät. Ausführliche Anweisungen finden [Sie unter Deinstallieren der AWS IoT Greengrass Core-Software](#) im AWS IoT Greengrass Entwicklerhandbuch, Version 2.

Important

Durch die Deinstallation von Greengrass werden alle lokalen Konfigurationen und Daten entfernt. Stellen Sie sicher, dass Sie alle wichtigen Informationen gesichert haben, bevor Sie fortfahren.

(Optional) Um Ressourcen von Drittanbietern zu löschen

Erwägen Sie nach Abschluss dieses Tutorials, alle externen Ressourcen, die Sie erstellt haben, herunterzufahren. Auf diese Weise können Sie verhindern, dass Gebühren von Drittanbietern anfallen.

Weitere Ressourcen

Weitere Informationen finden Sie in den folgenden Ressourcen:

- [Interagiere mit anderen AWS Diensten](#)
- [Verwenden Sie AWS IoT SiteWise Edge-Gateways](#)
- [Fehlerbehebung bei einem SiteWise Edge-Gateway](#)
- [Bewährte Sicherheitsmethoden für AWS IoT SiteWise](#)
- [AWS IoT-Preisgestaltung](#)
- [Daten aufnehmen in AWS IoT SiteWise](#)
- [Verwenden Sie Tags in AWS IoT SiteWise](#)

Daten in Grafana visualisieren und teilen

Dieses Tutorial führt Sie durch die Konfiguration des AWS IoT SiteWise Datenquellen-Plugins mit Grafana, einer Datenvisualisierungsplattform. Mit Grafana können Sie Dashboards erstellen, die Ihre Industriedaten visualisieren und überwachen. In diesem Tutorial wird ein Beispieldatensatz aus einer Windpark-Demonstration verwendet, um wichtige Konzepte zu veranschaulichen. Nachdem Sie sich mit dem Prozess vertraut gemacht haben, können Sie das Tutorial mit Ihren eigenen Daten wiederholen.

Nachdem Sie dieses Tutorial abgeschlossen haben, können Sie Folgendes tun:

- Sammeln, Abfragen und Analysieren von Daten aus Industrieanlagen
- Erstellen Sie interaktive Grafana-Dashboards zur Visualisierung von Kennzahlen zur Anlagenleistung
- Überwachen Sie Betriebsdaten über eine einheitliche Oberfläche
- Teilen Sie mithilfe der Kollaborationsfunktionen von Grafana Erkenntnisse mit Ihrem Team
- Kombinieren Sie AWS IoT SiteWise Daten mit anderen AWS Datenquellen wie [Amazon CloudWatch](#) oder [Amazon Timestream](#)

Themen

- [Voraussetzungen](#)
- [Schritt 1: Konfigurieren Sie Ihren Amazon Managed Grafana-Arbeitsbereich](#)
- [Schritt 2: Als Datenquelle hinzufügen AWS IoT SiteWise](#)
- [Schritt 3: Erstellen Sie ein Dashboard, um Ihre Daten zu untersuchen und zu visualisieren](#)
- [\(optional\) Schritt 4: Richten Sie Warnmeldungen ein, um die Leistung zu überwachen](#)
- [Schritt 5: Ressourcen nach dem Tutorial bereinigen](#)
- [Weitere Ressourcen](#)

Voraussetzungen

Zum Durcharbeiten dieses Tutorials ist Folgendes erforderlich:

- Ein AWS-Konto. Falls Sie noch keines haben, beachten Sie die Informationen unter [Richten Sie eine einAWSKonto](#).
- Ein AWS Identity and Access Management (IAM-) Benutzer mit Administratorrechten. Detaillierte Anweisungen finden Sie unter [the section called “WieAWS IoT SiteWisefunktioniert mit IAM”](#).
- Eine laufende AWS IoT SiteWise Demo.

Note

Dieses Tutorial erfordert die Verwendung von Ressourcen, die in Use [the AWS IoT SiteWise demo](#) erstellt wurden. Sie müssen es abschließen, bevor Sie mit diesem Tutorial fortfahren können.

Die Erstellung der Demo dauert in der Regel etwa 3 Minuten. Wenn die Demo nicht erstellt werden kann, deutet dies möglicherweise auf unzureichende Berechtigungen in Ihrem AWS Konto hin. Wechseln Sie in diesem Fall zu einem Konto mit Administratorzugriff. Weitere Informationen zu erforderlichen Berechtigungen finden Sie unter [WieAWS IoT SiteWise funktioniert mit IAM](#).

 Important

Behalten Sie alle Demo-Ressourcen, bis Sie dieses Tutorial abgeschlossen haben. Das Löschen von Komponenten kann die Funktionalität der Demo beeinträchtigen und Ihre Fähigkeit beeinträchtigen, Daten in Grafana anzuzeigen.

Schritt 1: Konfigurieren Sie Ihren Amazon Managed Grafana-Arbeitsbereich

In diesem Verfahren erstellen und konfigurieren Sie einen Amazon Managed Grafana-Arbeitsbereich zur Visualisierung Ihrer Windparkdaten.

1. Melden Sie sich bei der [Amazon Managed Grafana-Konsole](#) an.
2. Wählen Sie Workspace erstellen.
3. Geben Sie unter Workspace-Details einen Namen für Ihren Workspace ein, z. B.
SiteWiseTutorialDemo
4. Wählen Sie unter Grafana-Version die neueste Version aus. Wählen Sie diese Version für die meisten up-to-date Funktionen und Fähigkeiten. Weitere Informationen zu verschiedenen Versionssätzen finden Sie unter [Unterschiede zwischen Grafana-Versionen](#) im Amazon Managed Grafana-Benutzerhandbuch.
5. Wählen Sie Weiter aus.
6. Wählen Sie unter Authentifizierungszugriff die Option AWS IAM Identity Center aus.
 - Wenn AWS IAM Identity Center in Ihrem Konto nicht aktiviert ist, werden Sie aufgefordert, es zuerst einzurichten. Eine ausführliche Anleitung zur Einrichtung des Benutzerzugriffs finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Managed Grafana](#) im Amazon Managed Grafana-Benutzerhandbuch.
7. Wählen Sie unter Berechtigungstyp die Option Service managed aus. Amazon Managed Grafana erstellt und konfiguriert automatisch die erforderlichen IAM-Rollen und -Berechtigungen für alle AWS Datenquellen, die Sie in diesem Workspace verwenden möchten. Für Mitgliedskonten von

Organisationen ist die Option Service managed nur verfügbar, wenn das Konto als delegierter Administrator benannt ist. Informationen zur Einrichtung delegierter Administratorkonten finden Sie unter [Registrieren eines delegierten Administrator-Mitgliedskontos](#) im CloudFormation Benutzerhandbuch.

8. Führen Sie unter Workspace-Konfigurationsoptionen die folgenden Aktionen aus:
 - a. Wählen Sie Grafana-Warnung einschalten aus. Mit dieser Einstellung können Sie Benachrichtigungen über eine zentrale Oberfläche in Ihrem Workspace erstellen und verwalten. Weitere Informationen finden Sie unter [Arbeiten mit Grafana-Benachrichtigungen im Amazon](#) Managed Grafana-Benutzerhandbuch.
 - b. Wählen Sie Plugin-Verwaltung einschalten aus. Auf diese Weise können Sie Plugins in Ihrem Workspace installieren, aktualisieren und deinstallieren. Weitere Informationen finden Sie unter [Erweitern Sie Ihren Workspace mit Plugins](#) im Amazon Managed Grafana-Benutzerhandbuch.

 Important

Achten Sie darauf, die Plugin-Verwaltung zu aktivieren. Wenn Sie diese Option nicht auswählen, können Sie im folgenden Schritt keine Datenquelle hinzufügen AWS IoT SiteWise .

9. Wählen Sie unter Netzwerkzugriffskontrolle die Option Offener Zugriff aus. In diesem Tutorial verwenden Sie Demo-Daten, damit Sie den Workspace öffentlich zugänglich machen können.
 - Offener Zugang — Ermöglicht den öffentlichen Zugriff auf deinen Workspace.
 - Eingeschränkter Zugriff — Beschränkt den Zugriff auf bestimmte IP-Bereiche oder VPC-Endpunkte. Weitere Informationen finden Sie unter [So funktioniert VPC-Konnektivität](#) im Amazon Managed Grafana-Benutzerhandbuch.
10. Wählen Sie Weiter aus.
11. Wählen Sie auf der Seite Einstellungen für verwaltete Serviceberechtigungen die Option Girokonto aus, damit Amazon Managed Grafana automatisch Richtlinien und Berechtigungen für den Zugriff auf AWS Daten in Ihrem Konto erstellt.
12. Wählen Sie unter Datenquellen die Option aus AWS IoT SiteWise. Weitere Informationen finden Sie unter [Connect zu einer AWS IoT SiteWise Datenquelle](#) herstellen im Amazon Managed Grafana-Benutzerhandbuch.

13. (Optional) Wählen Sie unter Benachrichtigungskanäle Amazon SNS aus, damit Grafana-Benachrichtigungen über Amazon SNS gesendet werden können. Dadurch wird eine IAM-Richtlinie erstellt, die das Veröffentlichen von Themen in Amazon SNS ermöglicht, deren Namen mit Grafana beginnen. Sie müssen die Einrichtung des Benachrichtigungskanals später in Ihrer Grafana-Konsole im Workspace abschließen.
14. Bestätigen Sie die Workspace-Details und wählen Sie Workspace erstellen. Dieser Vorgang dauert ein paar Minuten.
15. Weisen Sie auf der Registerkarte Authentifizierung unter AWS IAM Identity Center Ihrem Workspace Benutzer oder Gruppen zu, indem Sie wie folgt vorgehen:
 - Um dem Benutzer zuzuweisen, der die AWS IoT SiteWise Daten verwalten soll, wählen Sie Neuen Benutzer oder neue Gruppe zuweisen. Wählen Sie dann in der Dropdownliste Aktionen die Option Zum Administrator ernennen aus, um dem Benutzer Administratorrechte zu gewähren.

 Important

Um den Grafana-Workspace zu verwalten, müssen Sie die Admin Rolle mindestens einem Benutzer zuweisen. Dieser Benutzer hat vollen Zugriff auf die Grafana-Workspace-Konsole.

Sie haben jetzt Ihren Grafana-Workspace eingerichtet und konfiguriert. Im nächsten Schritt können Sie eine Datenquelle hinzufügen AWS IoT SiteWise und mit der Erstellung von Visualisierungen für Ihre Windparkdaten beginnen. Von Ihrem Workspace aus können Sie Ihre Industriedaten in Echtzeit abfragen, visualisieren und analysieren. Weitere Informationen zu Amazon Managed Grafana-Workspaces finden Sie unter [Verwenden Sie Ihren Grafana-Workspace](#) im Amazon Managed Grafana-Benutzerhandbuch.

Schritt 2: Als Datenquelle hinzufügen AWS IoT SiteWise

Um Ihnen bei der Visualisierung Ihrer Daten zu helfen, enthält Amazon Managed Grafana das AWS Data Sources-Plugin, das den Verbindungsprozess vereinfacht. AWS-Services Dieses Plugin ist in Ihrem Workspace vorinstalliert und bietet eine einheitliche Oberfläche für die Erkennung und Konfiguration von AWS Ressourcen als Datenquellen. Für die Visualisierung Ihres Windparks verwenden Sie dieses Plugin, um AWS IoT SiteWise eine Verbindung herzustellen. Weitere

Informationen finden Sie unter [Connect zu einer AWS IoT SiteWise Datenquelle](#) herstellen im Amazon Managed Grafana-Benutzerhandbuch.

Bevor Sie mit der Abfrage Ihrer Windparkdaten beginnen können, benötigt das AWS Data Sources-Plugin die entsprechenden Berechtigungen für den Zugriff auf Ihre AWS IoT SiteWise Ressourcen. Diese Berechtigungen wurden automatisch konfiguriert, AWS IoT SiteWise als Sie im vorherigen Schritt eine Datenquelle ausgewählt haben. Weitere Informationen zu Plugin-Berechtigungen finden Sie unter [Erforderliche Berechtigungen](#) im Amazon Managed Grafana-Benutzerhandbuch.

Um eine Verbindung AWS IoT SiteWise zu Ihrem Grafana-Workspace herzustellen

1. Öffnen Sie die [Amazon Managed Grafana-Konsole](#). Wählen Sie auf Ihrer Workspace-Detailseite die URL aus, die unter Grafana-Workspace-URL angezeigt wird. Die Workspace-URL öffnet die Anmeldeseite der Grafana Workspace Console.
2. Wählen Sie Mit AWS IAM Identity Center anmelden und geben Sie Ihre Anmeldeinformationen ein. Diese Anmeldeinformationen funktionieren nur, wenn Sie auf die E-Mail von Amazon Managed Grafana geantwortet haben, in der Sie aufgefordert wurden, ein Passwort für IAM Identity Center zu erstellen.
3. Wählen Sie im linken Navigationsbereich Apps, dann AWS Datenquellen und dann die AWS-ServicesRegisterkarte aus.
4. Wählen Sie unter Jetzt installieren aus AWS IoT SiteWise, um die neueste Version des AWS IoT SiteWise Plug-ins zu installieren.
5. Navigieren Sie zur Registerkarte Datenquellen und wählen Sie IoT SiteWise als Dienst aus.
6. Wählen Sie unter Standardregion die Region aus, aus der Sie Daten abrufen möchten, z. B. USA Ost (Nord-Virginia).
7. Nachdem Sie die Parameter für das Plugin angegeben haben, wählen Sie Datenquelle hinzufügen aus.
8. Wählen Sie Gehe zu Einstellungen.
9. Wählen Sie unter Verbindungsdetails die Option Speichern und testen aus, um sicherzustellen, dass der Dienst funktioniert.

Schritt 3: Erstellen Sie ein Dashboard, um Ihre Daten zu untersuchen und zu visualisieren

Erstellen Sie in diesem Schritt ein Grafana-Dashboard, um die zuvor erstellten Demo-Windparkdaten zu visualisieren. Dashboards helfen Ihnen bei der Überwachung Ihrer Daten, indem sie mehrere

Visualisierungen in einer einzigen Ansicht anzeigen. Mithilfe von Dashboards können Sie Kennzahlen verfolgen, Muster analysieren und Erkenntnisse aus Ihren Industriedaten gewinnen. Weitere Informationen finden Sie unter [Erstellen Sie Ihr erstes Dashboard](#) im Amazon Managed Grafana-Benutzerhandbuch.

Um dein erstes Dashboard in Grafana zu erstellen

1. Wählen Sie im linken Navigationsbereich Dashboards und dann Dashboard erstellen aus, um mit der Erstellung Ihres ersten Dashboards zu beginnen.
2. Wählen Sie Visualisierung hinzufügen aus. Dadurch wird der Panel-Editor geöffnet, in dem Sie Datenquellen, Abfragen und Visualisierungseinstellungen konfigurieren können.
3. Wählen Sie auf der Registerkarte Abfrage die AWS IoT SiteWise Datenquelle aus dem Dropdownmenü aus.
4. Wählen Sie unter Abfragetyp im Dropdownmenü die Option Eigenschaftswertaggregate abrufen aus, um aggregierte Werte für Asset-Eigenschaften im Zeitverlauf abzurufen.
5. Wählen Sie Erkunden aus, um die verfügbaren Vermögenswerte in Ihrer Hierarchie anzuzeigen. Wählen Sie auf der Registerkarte Hierarchie die Option Demo Wind Farm Asset und dann Demo Turbine Asset 1 aus.
6. Wählen Sie unter Eigenschaft aus den verfügbaren Eigenschaften die Option Durchschnittliche Leistung aus. Wählen Sie Abfragen ausführen aus, um die Abfrage auszuführen, sodass Sie eine Vorschau der Ausgabe anzeigen können. Die Visualisierung wird aktualisiert und zeigt nun die durchschnittlichen Leistungsdaten für Demo Turbine Asset 1.
7. Geben Sie dem neuen Bereich im rechten Navigationsbereich einen Titel, z. **Turbine Demo 1 (Average Power)** B. Wählen Sie Apply (Übernehmen), um die Änderungen zu speichern.

 Warning

Wenn Sie Änderungen am Dashboard vornehmen, speichern Sie das Dashboard, bevor Sie die Seite aktualisieren oder die Seite verlassen. Andernfalls verlieren Sie Ihren Fortschritt.

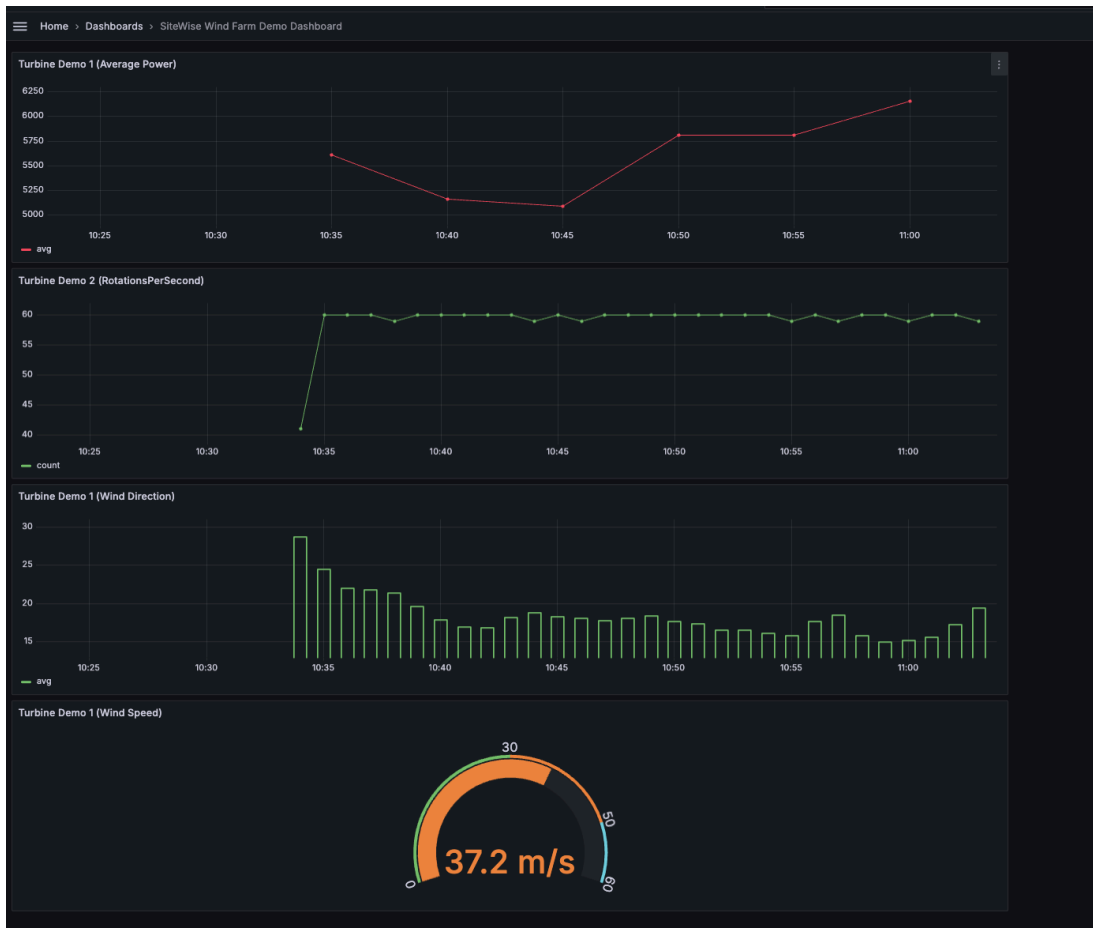
8. Wählen Sie in der oberen rechten Ecke Dashboard speichern aus. Sie werden aufgefordert, einen Namen für Ihr Dashboard einzugeben, zum Beispiel. **SiteWise Wind Farm Demo Dashboard**
9. Wählen Sie Speichern.

Informationen zum Teilen von Dashboards finden Sie unter [Teilen von Dashboards und Panels](#) im Amazon Managed Grafana-Benutzerhandbuch.

Um ein weiteres Fenster zur Visualisierung der Windgeschwindigkeit hinzuzufügen

1. Wählen Sie Visualisierung hinzufügen aus, um ein leeres Fenster zu öffnen.
2. Wählen Sie auf der Registerkarte Abfrage die AWS IoT SiteWise Datenquelle aus dem Dropdownmenü aus.
3. Wählen Sie unter Abfragetyp im Dropdownmenü die Option Eigenschaftswert abrufen aus und wählen Sie Demo Wind Farm Asset dann unter Asset die Option aus. Demo Turbine Asset 1
4. Wählen Sie unter Eigenschaft aus den verfügbaren Eigenschaften die Option Windgeschwindigkeit aus. Wählen Sie Abfragen ausführen aus, um die Änderungen zu aktualisieren.
5. Wählen Sie unter Visualisierung die Option Messgerät aus. Messgeräte eignen sich am besten zur Anzeige einzelner Echtzeitmetriken wie der Windgeschwindigkeit.
6. Geben Sie dem neuen Bereich im rechten Navigationsbereich einen Titel, z. B. **Turbine Demo 1 (Wind Speed)**
7. Wählen Sie in den Bedienfeldoptionen unter Standardoptionen die Option Einheit aus. Wählen Sie „Geschwindigkeit“ und anschließend „meters/second (m/s)“.
8. Wählen Sie Apply (Übernehmen), um die Änderungen zu speichern.

Die folgende Abbildung zeigt, wie Ihre Grafana-Dashboards aussehen könnten, wenn Sie diesen Schritt abschließen.



(optional) Schritt 4: Richten Sie Warnmeldungen ein, um die Leistung zu überwachen

Warnmeldungen weisen auf Statusänderungen hin, nachdem sie aufgetreten sind, um Leistungsprobleme mit Ihren Industrieanlagen zu identifizieren. Weitere Informationen finden Sie unter [Amazon Managed Grafana-Warnmeldungen](#) im Amazon Managed Grafana-Benutzerhandbuch.

Um Benachrichtigungen in Grafana einzurichten

1. Wählen Sie auf der Registerkarte Regel die Turbine Demo 1 (Average Power) Optionen Alle bis auswerten 5m und Für bis 15m aus. Diese Konfiguration bewertet den durchschnittlichen Stromverbrauch alle 5 Minuten und löst eine Warnung aus, wenn der Zustand länger als 15 Minuten andauert.
2. Wählen Sie unter Bedingungen die Option LIEGT UNTER UND geben Sie Watt ein **7,020**. Mit dieser Einstellung werden Sie benachrichtigt, wenn die durchschnittlichen Turbinenbedingungen länger als 5 Minuten unter 7.020 Watt fallen. Weitere Informationen zum Erstellen von

Benachrichtigungen finden Sie unter [Felder für Warnmeldungsregeln](#) im Amazon Managed Grafana-Benutzerhandbuch.

Sie haben das Tutorial abgeschlossen. In diesem Verfahren haben Sie einen Grafana-Workspace erstellt und ihn für die Visualisierung von AWS IoT SiteWise Windparkdaten konfiguriert. Sie haben ein interaktives Dashboard mit mehreren Widget-Typen erstellt, darunter ein Zeitreihendiagramm für die durchschnittliche Leistung und ein Messgerät für die Windgeschwindigkeit. Außerdem richten Sie Warnmeldungen ein, um die Turbinenleistung zu überwachen, sodass Sie potenzielle Probleme erkennen können, bevor sie die Produktion stören. Sie können Ihr Dashboard weiter verbessern, indem Sie weitere Visualisierungen hinzufügen, zusätzliche Warnmeldungen erstellen oder andere AWS Datenquellen verbinden, um tiefere Einblicke in Ihre industriellen Abläufe zu erhalten.

Schritt 5: Ressourcen nach dem Tutorial bereinigen

Nachdem Sie dieses Tutorial zur Datenvisualisierung mit Grafana abgeschlossen haben, sollten Sie Ihre Ressourcen bereinigen, um zusätzliche Kosten zu vermeiden.

Um die Demo zu löschen AWS IoT SiteWise

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie in der oberen rechten Ecke der Seite die Option Demo löschen aus.
3. Geben Sie in das Bestätigungsfeld **DELETE** ein, und wählen Sie dann Delete (Löschen).

Weitere Informationen finden Sie unter [Löschen Sie die AWS IoT SiteWise Demo](#).

Wenn Sie ein Amazon Managed Grafana löschen, werden auch alle Konfigurationsdaten für diesen Workspace gelöscht. Dazu gehören Dashboards, Datenquellenkonfiguration, Benachrichtigungen und Snapshots.

Um einen Amazon Managed Grafana-Workspace zu löschen

1. Öffnen Sie die [Amazon Managed Grafana-Konsole](#).
2. Wählen Sie im linken Navigationsbereich das Menüsymbol.
3. Wählen Sie Alle Workspaces.
4. Wählen Sie den Namen des Workspace, den Sie löschen möchten.
5. Wählen Sie Löschen aus.
6. Um den Löschvorgang zu bestätigen, gib den Namen des Workspace ein und wähle Löschen.

 Note

Mit diesem Verfahren wird ein Workspace gelöscht. Andere Ressourcen dürfen nicht gelöscht werden. Beispielsweise werden IAM-Rollen, die vom Workspace verwendet wurden, nicht gelöscht (sie können jedoch entsperrt werden, wenn sie nicht mehr verwendet werden).

Weitere Informationen finden Sie unter [Löschen eines Amazon Managed Grafana-Workspace](#) im Amazon Managed Grafana-Benutzerhandbuch.

Weitere Ressourcen

Weitere Informationen zur Visualisierung von Daten finden Sie in den folgenden Ressourcen:

- [Fehlerbehebung bei Amazon Managed Grafana-Identität und Zugriff](#) im Amazon Managed Grafana-Benutzerhandbuch
- [Bewährte Sicherheitsmethoden](#) im Amazon Managed Grafana-Benutzerhandbuch
- [Integrieren Sie AWS IoT SiteWise mit Grafana](#)
- [Verarbeiten und visualisieren Sie Daten mit SiteWise Edge- und Open-Source-Tools](#)
- [Benutzer, Teams und Berechtigungen](#) im Amazon Managed Grafana-Benutzerhandbuch
- [Amazon Managed Grafana-Berechtigungen und Richtlinien für AWS Datenquellen](#) im Amazon Managed Grafana-Benutzerhandbuch

Visualisieren und teilen Sie Windparkdaten in SiteWise Monitor

In diesem Tutorial wird erklärt, wie Industriedaten über verwaltete Webanwendungen, sogenannte Portale, visualisiert und gemeinsam genutzt AWS IoT SiteWise Monitor werden können. Jedes Portal umfasst Projekte, sodass Sie flexibel wählen können, auf welche Daten innerhalb jedes Projekts zugegriffen werden kann. Geben Sie anschließend Personen in Ihrer Organisation an, die auf jedes Portal zugreifen können. Ihre Benutzer melden sich mit AWS IAM Identity Center Konten bei Portalen an, sodass Sie Ihren vorhandenen Identitätsspeicher oder einen von verwalteten Speicher verwenden können AWS.

Sie und Ihre Benutzer mit ausreichenden Berechtigungen können in jedem Projekt Dashboards erstellen, um Ihre industriellen Daten sinnvoll zu visualisieren. Anschließend können Ihre Benutzer

diese Dashboards anzeigen, um schnell Einblicke in Ihre Daten zu erhalten und Ihren Betrieb zu überwachen. Sie können administrative oder schreibgeschützte Berechtigungen für jedes Projekt für jeden Benutzer in Ihrem Unternehmen konfigurieren. Weitere Informationen finden Sie unter [Überwachen Sie Daten mit AWS IoT SiteWise Monitor](#).

Im Laufe des Tutorials erweitern Sie die AWS IoT SiteWise Demo, indem Sie einen Beispieldatensatz für einen Windpark bereitstellen. Sie konfigurieren ein Portal in SiteWise Monitor, erstellen ein Projekt und Dashboards zur Visualisierung der Windparkdaten. Das Tutorial behandelt auch die Erstellung zusätzlicher Benutzer sowie die Zuweisung von Berechtigungen, um das Projekt und die zugehörigen Dashboards zu besitzen oder anzusehen.

Note

Wenn Sie SiteWise Monitor verwenden, wird Ihnen pro Benutzer, der sich bei einem Portal anmeldet, eine Gebühr berechnet (pro Monat). In diesem Tutorial erstellen Sie drei Benutzer, aber Sie müssen sich nur mit einem Benutzer anmelden. Nachdem Sie dieses Tutorial abgeschlossen haben, fallen Gebühren für einen Benutzer an. Weitere Informationen finden Sie unter [AWS IoT SiteWise – Preise](#).

Themen

- [Voraussetzungen](#)
- [Schritt 1: Erstellen Sie ein Portal in Monitor SiteWise](#)
- [Schritt 2: Melden Sie sich bei einem Portal an](#)
- [Schritt 3: Erstellen Sie ein Windparkprojekt](#)
- [Schritt 4: Erstellen Sie ein Dashboard zur Visualisierung von Windparkdaten](#)
- [Schritt 5: Erkunden Sie das Portal](#)
- [Schritt 6: Bereinigen Sie die Ressourcen nach dem Tutorial](#)

Voraussetzungen

Zum Durcharbeiten dieses Tutorials ist Folgendes erforderlich:

- Ein AWS-Konto. Falls Sie noch keines haben, beachten Sie die Informationen unter [Richten Sie eine einAWSKonto](#).

- Ein Entwicklungscomputer, auf dem Windows, macOS Linux, oder ausgeführt wird, Unix um auf den zuzugreifen AWS-Managementkonsole. Weitere Informationen finden Sie unter [Was ist AWS-Managementkonsole?](#)
- Ein AWS Identity and Access Management (IAM-) Benutzer mit Administratorrechten.
- Eine laufende AWS IoT SiteWise Windpark-Demo. Wenn Sie die Demo einrichten, definiert sie Modelle und Anlagen AWS IoT SiteWise und streamt Daten an sie, um einen Windpark darzustellen. Weitere Informationen finden Sie unter [Benutze die AWS IoT SiteWise Demo](#).
- Wenn Sie IAM Identity Center in Ihrem Konto aktiviert haben, melden Sie sich bei Ihrem AWS Organizations Verwaltungskonto an. Weitere Informationen zu finden Sie unter [Terminologie und Konzepte für AWS Organizations](#). Wenn Sie IAM Identity Center nicht aktiviert haben, werden Sie es in diesem Tutorial aktivieren und Ihr Konto als Verwaltungskonto einrichten.

Wenn Sie sich nicht bei Ihrem AWS Organizations Verwaltungskonto anmelden können, können Sie das Tutorial teilweise abschließen, sofern Sie einen IAM Identity Center-Benutzer in Ihrer Organisation haben. In diesem Fall können Sie das Portal und die Dashboards erstellen, aber Sie können keine neuen IAM Identity Center-Benutzer erstellen, um sie Projekten zuzuweisen.

Schritt 1: Erstellen Sie ein Portal in Monitor SiteWise

In diesem Verfahren erstellen Sie ein Portal in AWS IoT SiteWise Monitor. Jedes Portal ist eine verwaltete Webanwendung, bei der Sie und Ihre Benutzer sich mit AWS IAM Identity Center Konten anmelden können. Mit IAM Identity Center können Sie den vorhandenen Identitätsspeicher Ihres Unternehmens verwenden oder einen eigenen erstellen, der von verwaltet wird AWS. Die Mitarbeiter Ihres Unternehmens können sich anmelden, ohne einen separaten AWS-Konten Account erstellen zu müssen.

So erstellen Sie ein Portal

1. Melden Sie sich bei der [AWS IoT SiteWise -Konsole](#) an.
2. Prüfen Sie, [AWS IoT SiteWise welche Endgeräte und Kontingente](#) unterstützt werden, und wechseln Sie bei AWS IoT SiteWise Bedarf zwischen den Regionen. Sie müssen die AWS IoT SiteWise Demo in derselben Region ausführen.
3. Wählen Sie im linken Navigationsbereich die Option Portale aus.
4. Wählen Sie Create Portal (Portal erstellen) aus.
5. Wenn Sie IAM Identity Center bereits aktiviert haben, fahren Sie mit Schritt 6 fort. Gehen Sie andernfalls wie folgt vor, um IAM Identity Center zu aktivieren:

- a. Geben Sie auf der Seite Aktivieren AWS IAM Identity Center (SSO) Ihre E-Mail-Adresse, Ihren Vornamen und Nachnamen ein, um einen IAM Identity Center-Benutzer für Sie als Portaladministrator zu erstellen. Verwenden Sie eine E-Mail-Adresse, auf die Sie zugreifen können, damit Sie eine E-Mail erhalten, mit der Sie ein Passwort für Ihren neuen IAM Identity Center-Benutzer festlegen können.

In einem Portal erstellt der Portaladministrator Projekte und weist Benutzer Projekten zu. Sie können später weitere Benutzer erstellen.

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Enable SSO

Step 2
Portal configuration

Step 3
Invite administrators

Step 4
Assign users

Enable AWS Single Sign-On (SSO)

AWS IoT SiteWise Monitor requires SSO to create a portal and invite users. Create your first user below to enable AWS Single-Sign On. Later in this process, you'll have the opportunity to create other users by using the AWS SSO console. [Learn more](#)

Create a user

Email address
john.doe@example.com

First name
John

Last name
Doe

Upon creation this application will enable AWS Organizations and Single Sign-On. [Learn more](#)

Cancel **Create user**

- b. Wählen Sie Create user (Benutzer erstellen) aus.
6. Führen Sie auf der Seite Portalkonfiguration die folgenden Schritte aus:
- a. Geben Sie einen Namen für Ihr Portal ein, z. B. **WindFarmPortal**.
 - b. (Optional) Geben Sie eine Beschreibung für Ihr Portal ein. Wenn Sie über mehrere Portale verfügen, verwenden Sie aussagekräftige Beschreibungen, um den Überblick über die Inhalte der einzelnen Portale zu behalten.
 - c. (Optional) Laden Sie ein Bild hoch, das im Portal angezeigt werden soll.
 - d. Geben Sie eine E-Mail-Adresse ein, an die sich Portalbenutzer wenden können, wenn sie ein Problem mit dem Portal haben und Hilfe vom AWS Administrator Ihres Unternehmens benötigen, um das Problem zu lösen.
 - e. Wählen Sie Create Portal (Portal erstellen) aus.

7. Auf der Seite Administratoren einladen können Sie dem Portal IAM Identity Center-Benutzer als Administratoren zuweisen. Portaladministratoren verwalten Berechtigungen und Projekte innerhalb eines Portals. Gehen Sie auf dieser Seite wie folgt vor:
 - a. Wählen Sie einen Benutzer als Portaladministrator aus. Wenn Sie IAM Identity Center zu einem früheren Zeitpunkt in diesem Tutorial aktiviert haben, wählen Sie den Benutzer aus, den Sie erstellt haben.

The screenshot shows the 'Invite administrators' page in the AWS IoT SiteWise Monitor console. The breadcrumb trail at the top is 'AWS IoT SiteWise > Monitor > Portals > Create portal'. On the left, there are three steps: 'Step 1 Portal configuration', 'Step 2 Invite administrators' (which is the current step), and 'Step 3 Assign users'. The main heading is 'Invite administrators'. Below the heading, there is a description: 'Select the users that you want to be portal administrators. When invited, portal administrators control users' access to the operational data of your Sitewise assets. [Learn more](#)'. There is a button 'Send invite to selected users' in the top right. Below this, there is a section 'Users (1)' with a search bar 'Find resources'. A table lists users with columns 'Display name' and 'Email'. The first user, 'John Doe' with email 'john.doe@example.com', is selected, indicated by a blue checkmark in a circle. Below the table, there is a section 'Selected users (1)'. At the bottom right, there are 'Cancel' and 'Next' buttons, with the 'Next' button highlighted by a red circle.

- b. (Optional) Wählen Sie Send invite to selected users (Einladung an ausgewählte Benutzer senden) aus. Ihr E-Mail-Client wird geöffnet und eine Einladung wird im Nachrichtentext angezeigt. Sie können die E-Mail anpassen, bevor Sie sie an die Portaladministratoren senden. Sie können die E-Mail-Nachricht auch später an Ihre Portaladministratoren senden. Wenn Sie SiteWise Monitor zum ersten Mal ausprobieren und der Portaladministrator sein werden, müssen Sie sich keine E-Mail senden.
 - c. Wählen Sie Weiter aus.
8. Auf der Seite „Benutzer zuweisen“ können Sie dem Portal IAM Identity Center-Benutzer zuweisen. Portaladministratoren können diese Benutzer später als Projekteigentümer oder -betrachter zuweisen. Projekteigentümer können Dashboards in Projekten erstellen. Projektbetrachter haben nur Lesezugriff auf die ihnen zugewiesenen Projekte. Auf dieser Seite können Sie IAM Identity Center-Benutzer erstellen, die Sie dem Portal hinzufügen möchten.

Note

Wenn Sie nicht mit Ihrem AWS Organizations Verwaltungskonto angemeldet sind, können Sie keine IAM Identity Center-Benutzer erstellen. Wählen Sie Benutzer zuweisen aus, um das Portal ohne Portalbenutzer zu erstellen, und überspringen Sie dann diesen Schritt.

Gehen Sie auf dieser Seite wie folgt vor:

- a. Führen Sie die folgenden Schritte zweimal aus, um zwei IAM Identity Center-Benutzer zu erstellen:
 - i. Wählen Sie Benutzer erstellen, um ein Dialogfeld zu öffnen, in dem Sie Details für den neuen Benutzer eingeben können.
 - ii. Geben Sie eine E-Mail-Adresse, einen Vornamen und einen Nachnamen für den neuen Benutzer ein. IAM Identity Center sendet dem Benutzer eine E-Mail, damit er sein Passwort einrichten kann. Wenn Sie sich als diese Benutzer beim Portal anmelden möchten, wählen Sie eine E-Mail-Adresse aus, auf die Sie zugreifen können. Jede E-Mail-Adresse muss eindeutig sein. Ihre Benutzer melden sich mit ihrer E-Mail-Adresse als Benutzernamen beim Portal an.

Create user [X]

Create a new AWS user. You can assign this user access to AWS applications and services

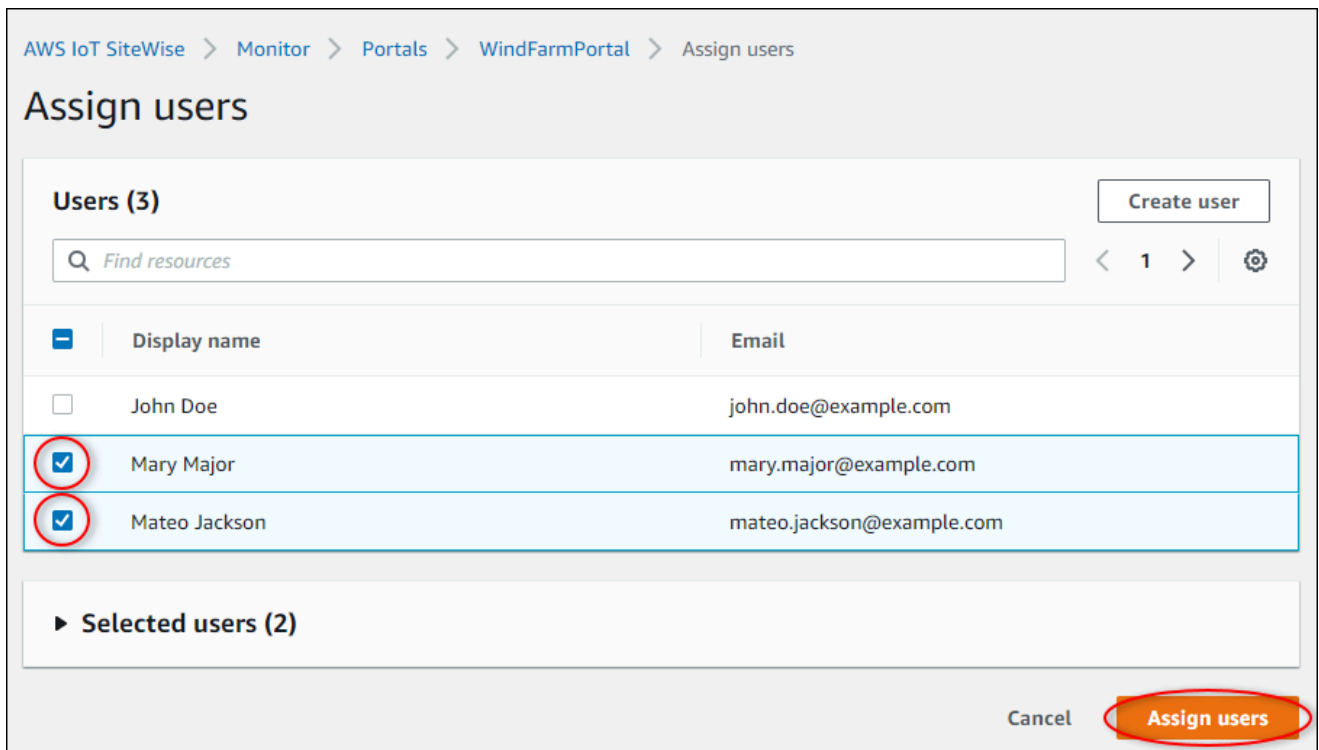
Email address
mary.major@example.com

First name
Mary

Last name
Major

Cancel Create user

- iii. Wählen Sie Create user (Benutzer erstellen) aus.
- b. Wählen Sie die beiden IAM Identity Center-Benutzer aus, die Sie im vorherigen Schritt erstellt haben.



- c. Wählen Sie Benutzer zuweisen, um diese Benutzer zum Portal hinzuzufügen.

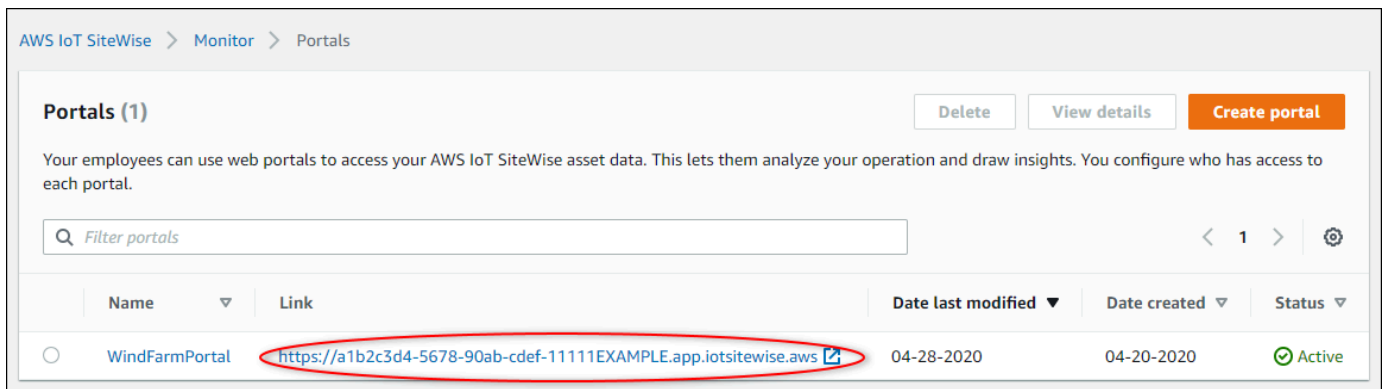
Die Seite „Portale“ wird geöffnet, wobei das neue Portal aufgelistet ist.

Schritt 2: Melden Sie sich bei einem Portal an

In diesem Verfahren melden Sie sich mit dem AWS IAM Identity Center Benutzer, den Sie dem Portal hinzugefügt haben, bei Ihrem neuen Portal an.

So melden Sie sich bei einem Portal an

1. Wählen Sie auf der Seite Portale den Link Ihres neuen Portals aus, um das Portal in einer neuen Registerkarte zu öffnen.



2. Wenn Sie zu Beginn des Tutorials Ihren ersten IAM Identity Center-Benutzer erstellt haben, gehen Sie wie folgt vor, um ein Passwort für Ihren Benutzer zu erstellen:
 - a. Überprüfen Sie Ihre E-Mail nach der Betreffzeile Invitation to join AWS IAM Identity Center.
 - b. Öffnen Sie diese Einladungs-E-Mail und wählen Sie Accept invitation aus.
 - c. Legen Sie im neuen Fenster ein Passwort für Ihren IAM Identity Center-Benutzer fest.

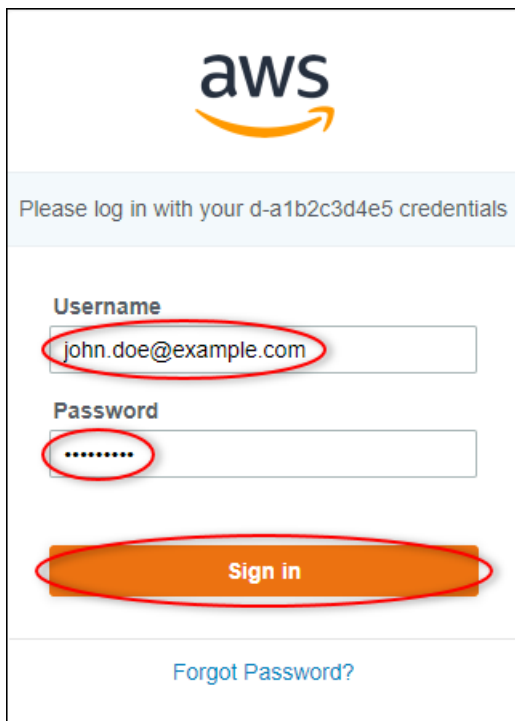
Wenn Sie sich später als zweiter und dritter IAM Identity Center-Benutzer, den Sie zuvor erstellt haben, am Portal anmelden möchten, können Sie auch diese Schritte ausführen, um Passwörter für diese Benutzer festzulegen.

Note

Wenn Sie keine E-Mail erhalten haben, können Sie in der IAM Identity Center-Konsole ein Passwort für Ihren Benutzer generieren. Weitere Informationen finden Sie im Benutzerhandbuch unter [Zurücksetzen des IAM Identity Center-Benutzerkennworts für einen Endbenutzer](#). AWS IAM Identity Center

3. Geben Sie Ihr IAM Identity Center ein und. UsernamePassword Wenn Sie Ihren IAM Identity Center-Benutzer zu einem früheren Zeitpunkt in diesem Tutorial erstellt haben, Username ist dies die E-Mail-Adresse des Portal-Administratorbenutzers, den Sie erstellt haben.

Alle Portalbenutzer, einschließlich des Portaladministrators, müssen sich mit ihren IAM Identity Center-Benutzeranmeldedaten anmelden. Diese Anmeldeinformationen sind in der Regel nicht mit den Anmeldeinformationen identisch, mit denen Sie sich bei der AWS-Managementkonsole anmelden.



aws

Please log in with your d-a1b2c3d4e5 credentials

Username
john.doe@example.com

Password
.....

Sign in

[Forgot Password?](#)

4. Wählen Sie Sign in.

Ihr Portal wird geöffnet.

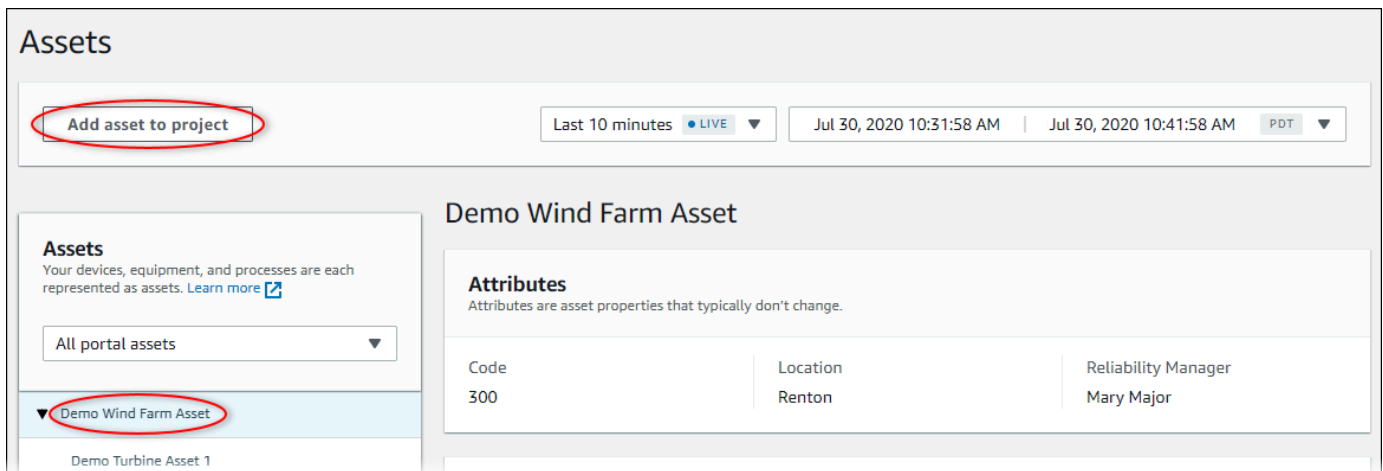
Schritt 3: Erstellen Sie ein Windparkprojekt

In diesem Verfahren erstellen Sie ein Projekt in Ihrem Portal. Projekte sind Ressourcen, die eine Reihe von Berechtigungen, Ressourcen und Dashboards definieren, die Sie konfigurieren können, um Asset-Daten in diesem Projekt zu visualisieren. Mit Projekten definieren Sie, wer Zugriff auf welche Teilmengen Ihrer Operation hat und wie die Daten dieser Teilmengen visualisiert werden. Sie können Portalbenutzer als Eigentümer oder Betrachter für jedes Projekt zuweisen. Projekteigentümer können Dashboards erstellen, um Daten zu visualisieren und das Projekt mit anderen Benutzern zu teilen. Projektbetrachter können Dashboards anzeigen, sie aber nicht bearbeiten. Weitere Informationen zu Rollen in SiteWise Monitor finden Sie unter [SiteWise Rollen überwachen](#).

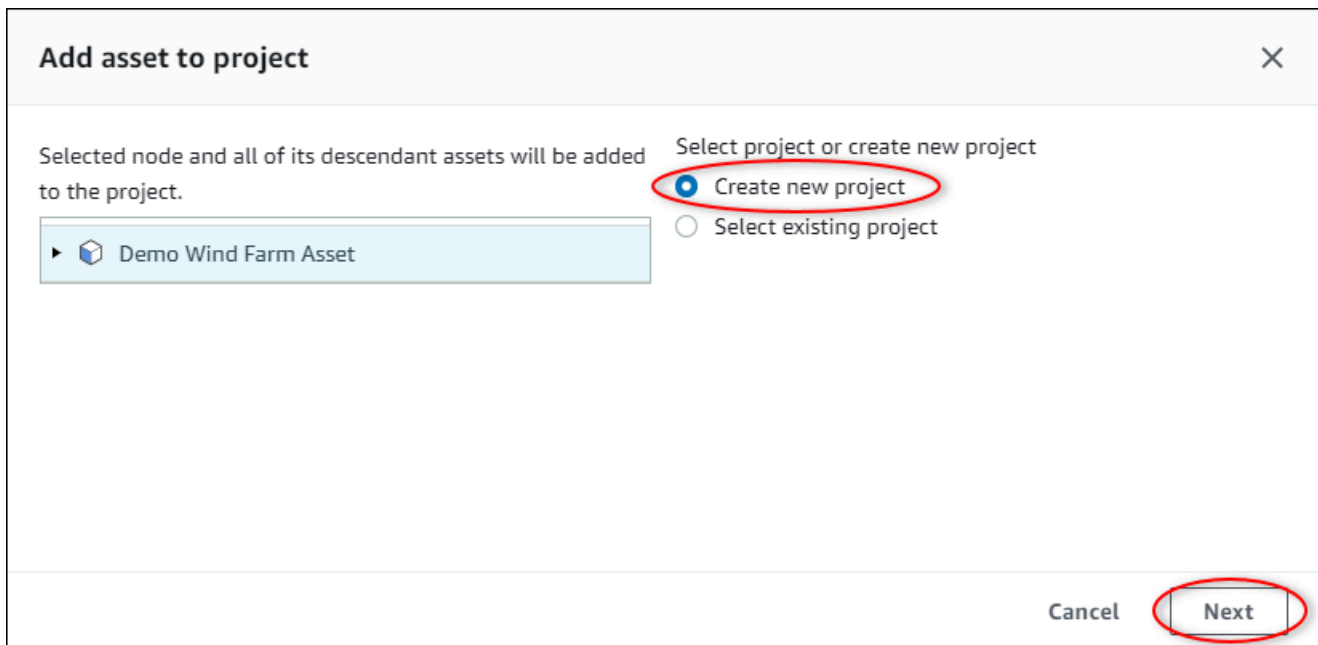
So erstellen Sie ein Windparkprojekt

1. Wählen Sie im linken Navigationsbereich Ihres Portals die Registerkarte Assets aus. Auf der Seite Assets können Sie alle im Portal verfügbaren Assets erkunden und Assets zu Projekten hinzufügen.

2. Wählen Sie im Komponentenbrowser die Option Demo Wind Farm Asset aus. Wenn Sie ein Asset auswählen, können Sie die aktuellen und historischen Daten dieses Assets untersuchen. Sie können auch drücken Shift, um mehrere Vermögenswerte auszuwählen und deren Daten zu vergleichen side-by-side.
3. Wählen Sie oben links „Asset zum Projekt hinzufügen“. Projekte enthalten Dashboards, die Ihre Portalbenutzer anzeigen können, um Ihre Daten zu erkunden. Jedes Projekt hat Zugriff auf eine Teilmenge Ihrer Ressourcen in AWS IoT SiteWise. Wenn Sie einem Projekt eine Komponente hinzufügen, können alle Benutzer mit Zugriff auf dieses Projekt auch auf Daten für diese Komponente und ihre untergeordneten Elemente zugreifen.



4. Wählen Sie im Dialogfeld „Objekt zum Projekt hinzufügen“ die Option „Neues Projekt erstellen“ und anschließend „Weiter“.



5. Geben Sie im Dialogfeld Neues Projekt erstellen einen Projektnamen und eine Projektbeschreibung für Ihr Projekt ein und wählen Sie dann Asset zum Projekt hinzufügen.

Create new project [X]

Project name
Wind Farm 1
The project name can have up to 256 characters.

Project description
A project that contains dashboards for wind farm #1.
The project description can have up to 2048 characters.

Cancel Previous **Add asset to project**

Die Seite Ihres neuen Projekts wird geöffnet.

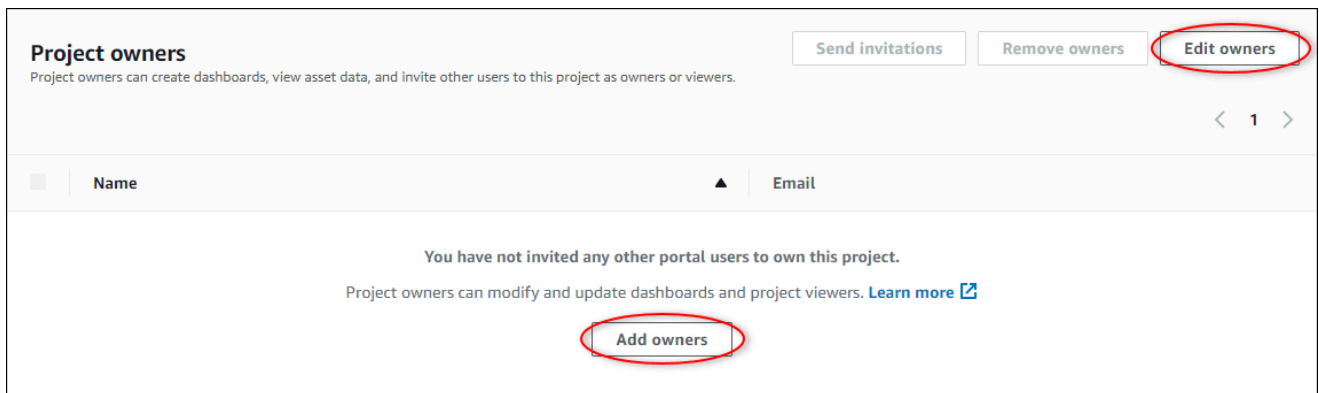
6. Auf der Projektseite können Sie Portalbenutzer als Eigentümer oder Betrachter dieses Projekts hinzufügen.

Note

Wenn Sie nicht bei Ihrem AWS Organizations Verwaltungskonto angemeldet sind, haben Sie möglicherweise keine Portalbenutzer, die Sie diesem Projekt zuweisen können. Sie können diesen Schritt also überspringen.

Gehen Sie auf dieser Seite wie folgt vor:

- a. Wählen Sie unter Projektinhaber die Option Eigentümer hinzufügen oder Benutzer bearbeiten aus.



Project owners

Project owners can create dashboards, view asset data, and invite other users to this project as owners or viewers.

[Send invitations](#) [Remove owners](#) [Edit owners](#)

< 1 >

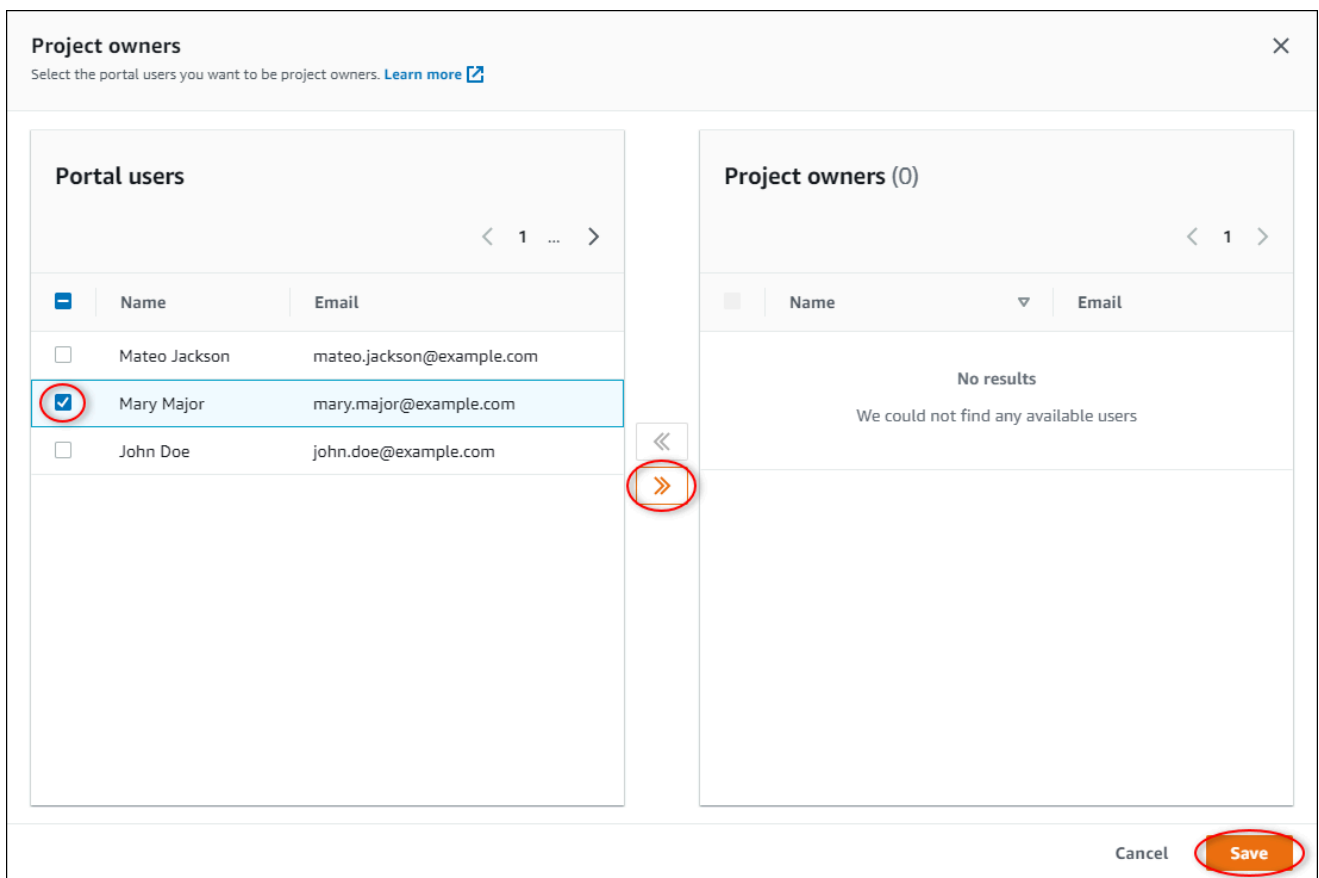
Name **Email**

You have not invited any other portal users to own this project.

Project owners can modify and update dashboards and project viewers. [Learn more](#)

[Add owners](#)

- b. Wählen Sie den Benutzer, der als Projekteigentümer hinzugefügt werden soll (z. B. Mary Major), und dann das Symbol >> aus.



Project owners

Select the portal users you want to be project owners. [Learn more](#)

Portal users

< 1 ... >

☐	Name	Email
☐	Mateo Jackson	mateo.jackson@example.com
☒	Mary Major	mary.major@example.com
☐	John Doe	john.doe@example.com

<< >>

Project owners (0)

< 1 >

Name **Email**

No results

We could not find any available users

Cancel [Save](#)

- c. Wählen Sie Speichern.

Ihr IAM Identity Center-Benutzer Mary Major kann sich bei diesem Portal anmelden, um die Dashboards in diesem Projekt zu bearbeiten und dieses Projekt mit anderen Benutzern in diesem Portal zu teilen.

- d. Wählen Sie unter Projekt-Viewer die Option Zuschauer hinzufügen oder Benutzer bearbeiten aus.

- e. Wählen Sie den Benutzer aus, der als Projektbetrachter hinzugefügt werden soll (z. B. Mateo Jackson), und wählen Sie dann das Symbol >>.
- f. Wählen Sie Speichern.

Ihr IAM Identity Center-Benutzer Mateo Jackson kann sich bei diesem Portal anmelden, um die Dashboards im Windparkprojekt anzusehen, aber nicht zu bearbeiten.

Schritt 4: Erstellen Sie ein Dashboard zur Visualisierung von Windparkdaten

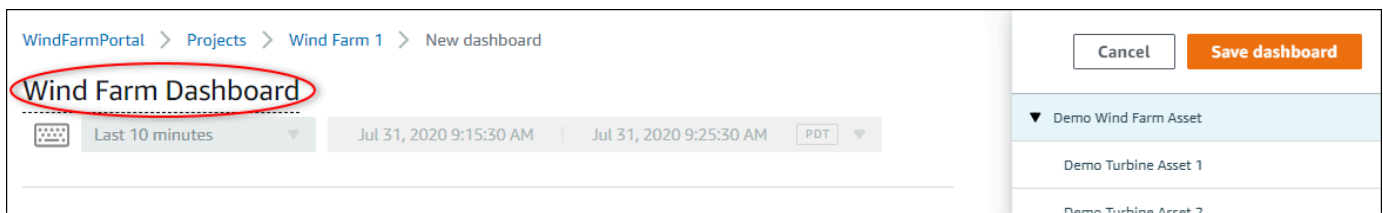
In diesem Verfahren erstellen Sie Dashboards, um die Demo-Windparkdaten zu visualisieren. Dashboards enthalten anpassbare Visualisierungen der Komponentendaten Ihres Projekts. Jede Visualisierung kann einen anderen Typ haben, z. B. ein Liniendiagramm, ein Balkendiagramm oder eine KPI-Anzeige (Key Performance Indicator). Sie können den Visualisierungstyp auswählen, der für Ihre Daten am besten geeignet ist. Projekteigentümer können Dashboards bearbeiten, wohingegen Projektbetrachter nur Dashboards anzeigen können, um Einblicke zu gewinnen.

So erstellen Sie ein Dashboard mit Visualisierungen

1. Wählen Sie auf der Seite Ihres neuen Projekts die Option Dashboard erstellen aus, um ein Dashboard zu erstellen und dessen Bearbeitungsseite zu öffnen.

Auf der Bearbeitungsseite eines Dashboards können Sie Komponenteneigenschaften aus der Komponentenhierarchie in das Dashboard ziehen, um Visualisierungen zu erstellen. Anschließend können Sie Titel, Legendentitel, Typ, Größe und Position jeder Visualisierung im Dashboard bearbeiten.

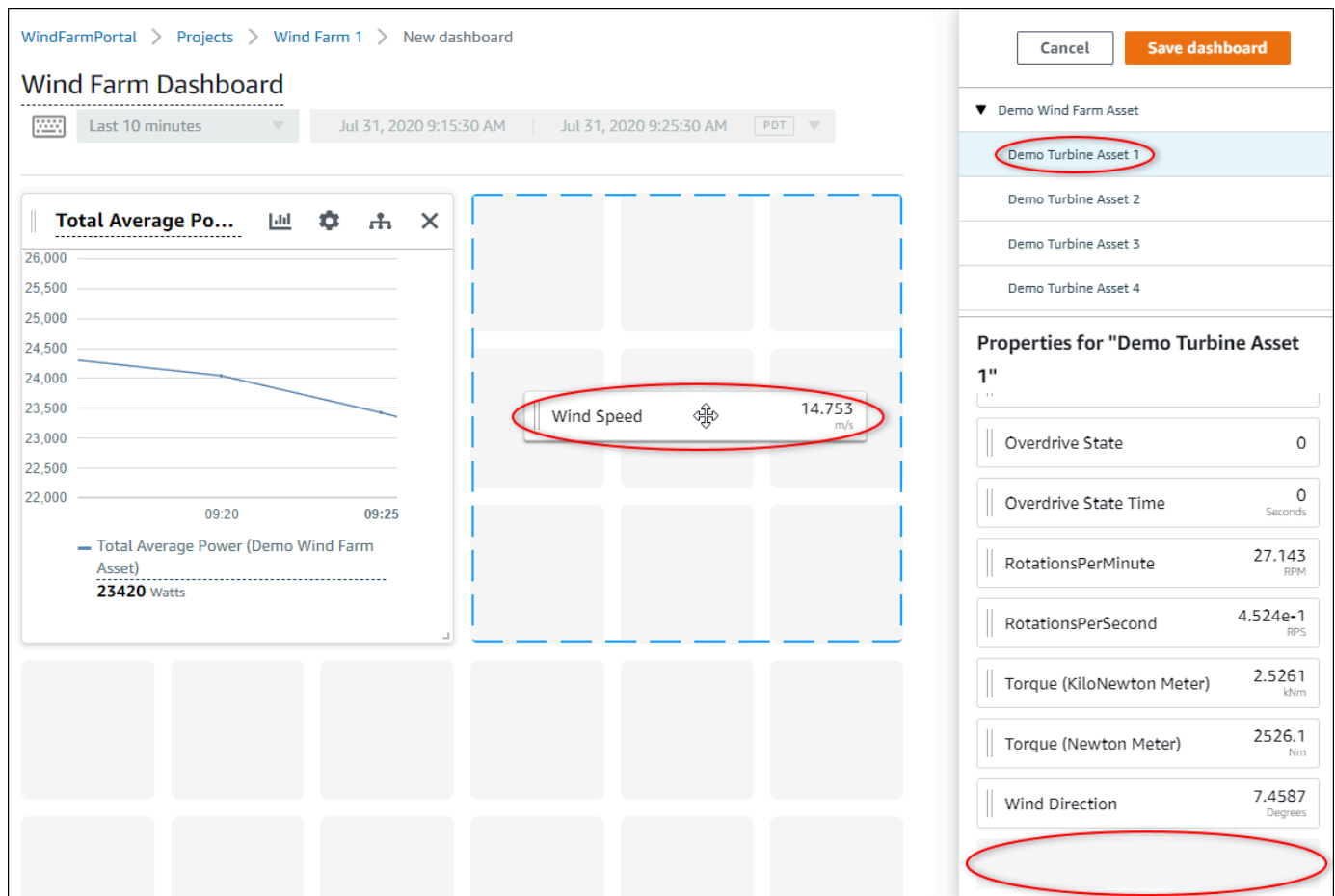
2. Geben Sie einen Namen für Ihr Dashboard ein.



3. Ziehen Sie Total Average Power von der Demo Wind Farm Asset in das Dashboard, um eine Visualisierung zu erstellen.

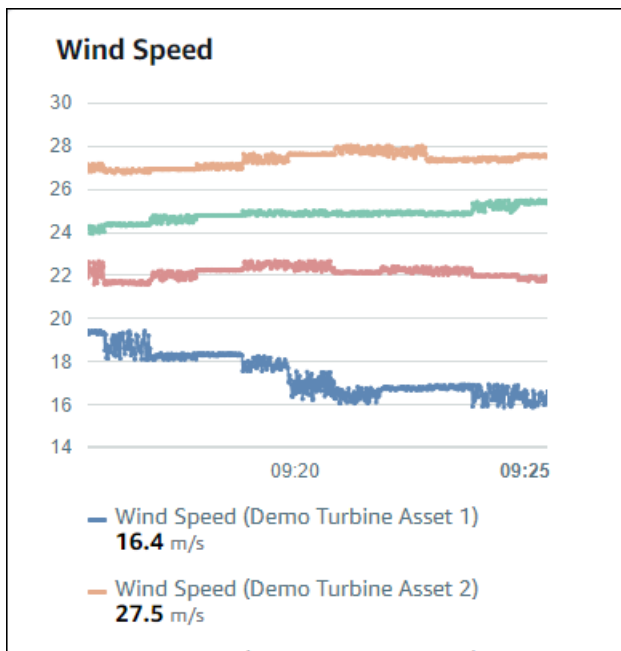
The screenshot displays the 'Wind Farm Dashboard' in the AWS IoT SiteWise console. The breadcrumb navigation at the top reads 'WindFarmPortal > Projects > Wind Farm 1 > New dashboard'. The dashboard title is 'Wind Farm Dashboard'. Below the title, there are filters for 'Last 10 minutes', a date range from 'Jul 31, 2020 9:15:30 AM' to 'Jul 31, 2020 9:25:30 AM', and a time zone dropdown set to 'PDT'. The main area is a grid of 12 placeholder widgets. One widget in the second row, first column, is highlighted with a red dashed border and contains the text 'Total Average Power' with a small icon, followed by the value '24038 Watts'. To the right of the dashboard grid is a sidebar. At the top of the sidebar are 'Cancel' and 'Save dashboard' buttons. Below them is a section titled 'Demo Wind Farm Asset' with a dropdown arrow. Under this section, there is a list of four assets: 'Demo Turbine Asset 1', 'Demo Turbine Asset 2', 'Demo Turbine Asset 3', and 'Demo Turbine Asset 4'. Below the list is a section titled 'Properties for "Demo Wind Farm Asset"'. This section contains two property cards: 'Code' with a value of '300' and 'Total Overdrive State Time' with a value of '0 seconds'. The 'Total Overdrive State Time' card is highlighted with a red oval.

4. Wählen Demo Turbine Asset 1 aus, ob die Eigenschaften für diese Anlage angezeigt werden sollen, und ziehen Sie sie dann Wind Speed auf das Dashboard, um eine Visualisierung für die Windgeschwindigkeit zu erstellen.

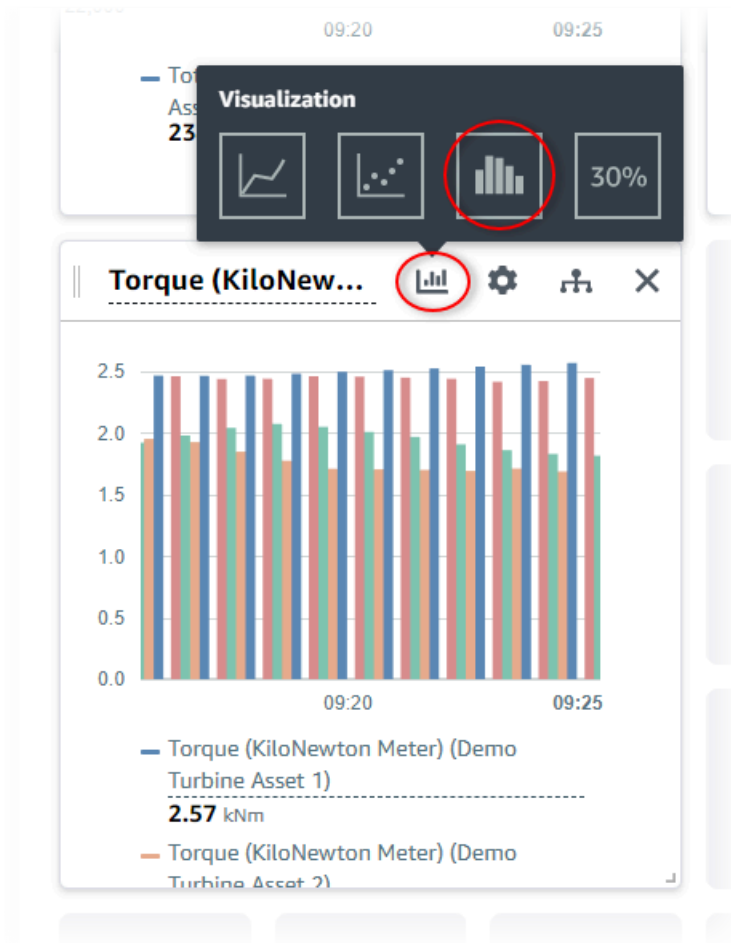


5. Fügen Sie Wind Speed der neuen Visualisierung der Windgeschwindigkeit für jede Demo Turbine Asset 2, 3 und 4 (in dieser Reihenfolge) hinzu.

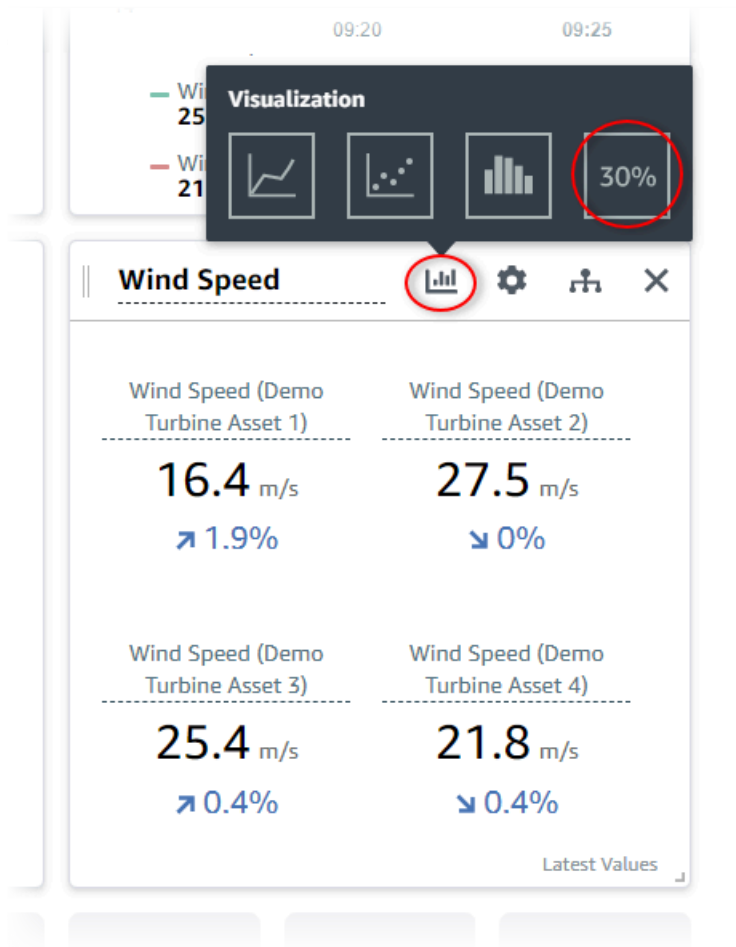
Ihre Wind Speed-Visualisierung sollte dem folgenden Screenshot ähnlich aussehen.



6. Wiederholen Sie die Schritte 4 und 5 für die Torque (KiloNewton Meter)Eigenschaften der Windturbinen, um eine Visualisierung für das Drehmoment der Windturbine zu erstellen.
7. Wählen Sie das Symbol für den Visualisierungstyp für die Torque (KiloNewton Meter)-Visualisierung und dann das Balkendiagrammsymbol aus.

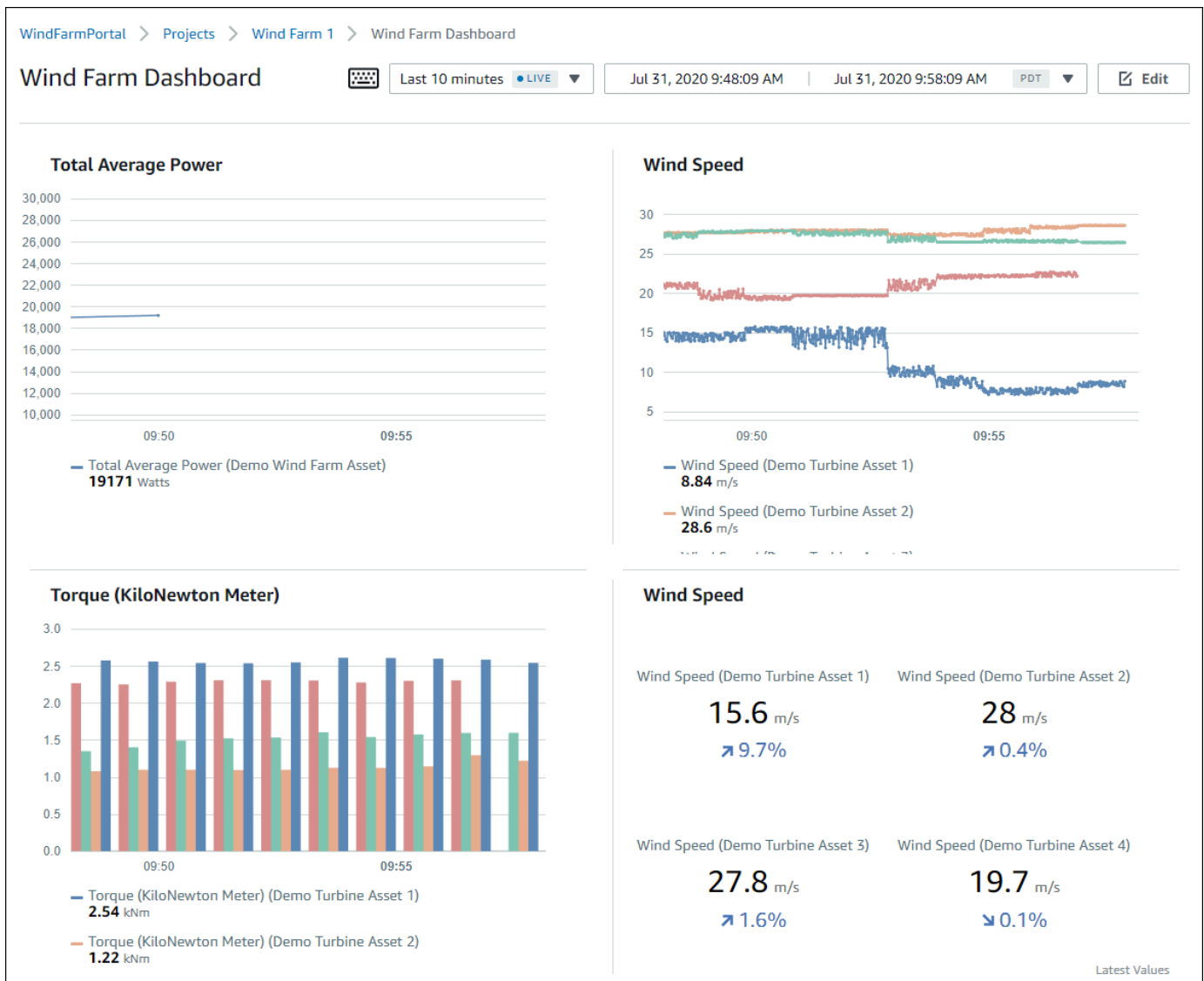


8. Wiederholen Sie die Schritte 4 und 5 für die Wind DirectionEigenschaften der Windturbinen, um eine Visualisierung der Windrichtung zu erstellen.
9. Wählen Sie das Symbol für den Visualisierungstyp für die Wind Direction-Visualisierung und dann das KPI-Diagrammsymbol (30%) aus.



10. (Optional) Nehmen Sie nach Bedarf weitere Änderungen an Titel, Legendentitel, Typ, Größe und Position der Visualisierung vor.
11. Wählen Sie oben rechts Dashboard speichern, um Ihr Dashboard zu speichern.

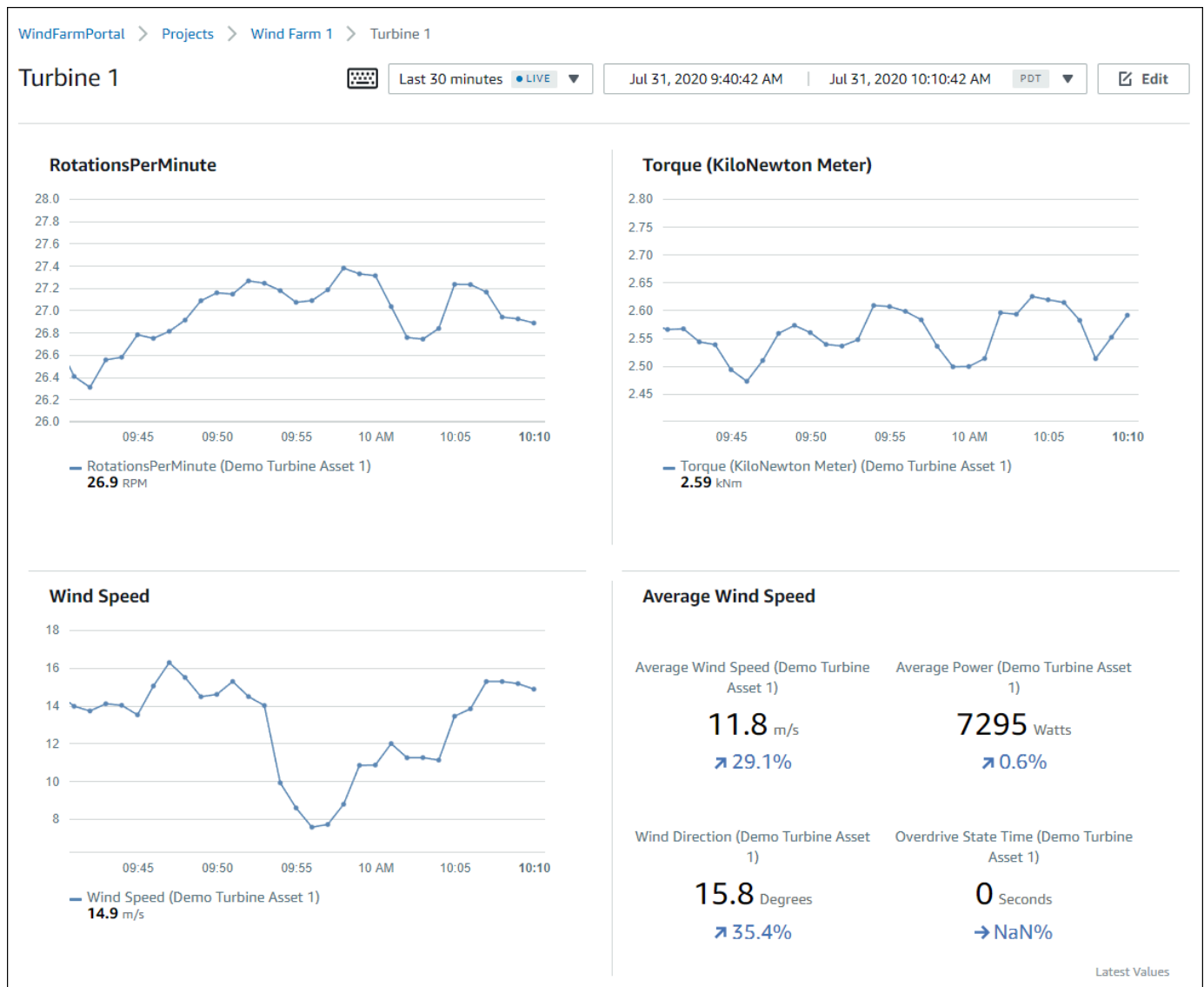
Ihr Dashboard sollte dem folgenden Screenshot ähnlich aussehen.



12. (Optional) Erstellen Sie für jede Windkraftanlagen-Komponente ein zusätzliches Dashboard.

Als bewährte Methode empfehlen wir, für jede Komponente ein Dashboard zu erstellen, damit Ihre Projektbetrachter alle Probleme mit den einzelnen Komponenten untersuchen können. Sie können jeder Visualisierung nur bis zu 5 Komponenten hinzufügen. Daher müssen Sie in vielen Szenarien mehrere Dashboards für Ihre hierarchischen Komponenten erstellen.

Ein Dashboard für eine Demo-Windkraftanlage könnte ähnlich dem folgenden Screenshot aussehen.



13. (Optional) Ändern Sie die Zeitachse oder wählen Sie Datenpunkte in einer Visualisierung aus, um die Daten im Dashboard zu erkunden. Weitere Informationen finden Sie im AWS IoT SiteWise Monitor Anwendungsleitfaden unter [Dashboards anzeigen](#).

Schritt 5: Erkunden Sie das Portal

In diesem Verfahren können Sie das Portal als Benutzer mit weniger Berechtigungen als ein AWS IoT SiteWise Portaladministrator erkunden.

Um das Portal zu erkunden und das Tutorial zu beenden

- (Optional) Wenn Sie dem Projekt weitere Benutzer als Eigentümer oder Betrachter hinzugefügt haben, können Sie sich als diese Benutzer beim Portal anmelden. Auf diese Weise können Sie das Portal als Benutzer mit weniger Berechtigungen als ein Portaladministrator erkunden.

Important

Ihnen wird für jeden Benutzer, der sich bei einem Portal anmeldet, eine Gebühr berechnet. Weitere Informationen finden Sie unter [AWS IoT SiteWise – Preise](#).

Gehen Sie wie folgt vor, um das Portal als andere Benutzer zu erkunden:

- a. Wählen Sie unten links im Portal Abmelden aus, um die Webanwendung zu beenden.
- b. Wählen Sie oben rechts im IAM Identity Center-Anwendungsportal Abmelden, um sich von Ihrem IAM Identity Center-Benutzer abzumelden.
- c. Melden Sie sich beim Portal als der IAM Identity Center-Benutzer an, den Sie als Projektinhaber oder Projektbetrachter zugewiesen haben. Weitere Informationen finden Sie unter [Schritt 2: Melden Sie sich bei einem Portal an](#).

Sie haben das Tutorial abgeschlossen. Wenn Sie mit der Erkundung Ihres Demo-Windparks in SiteWise Monitor fertig sind, folgen Sie dem nächsten Verfahren, um Ihre Ressourcen zu bereinigen.

Schritt 6: Bereinigen Sie die Ressourcen nach dem Tutorial

Nachdem Sie das Tutorial abgeschlossen haben, können Sie Ihre Ressourcen bereinigen. Es fallen keine Gebühren für AWS IoT SiteWise an, wenn sich Benutzer nicht bei Ihrem Portal anmelden, aber Sie können Ihr Portal und Ihre AWS-IAM-Identity-Center-Verzeichnis -Benutzer löschen. Ihre Demo-Windparkkomponenten werden am Ende der Dauer gelöscht, die Sie beim Erstellen der Demo gewählt haben, oder Sie können die Demo manuell löschen. Weitere Informationen finden Sie unter [Löschen Sie die AWS IoT SiteWise Demo](#).

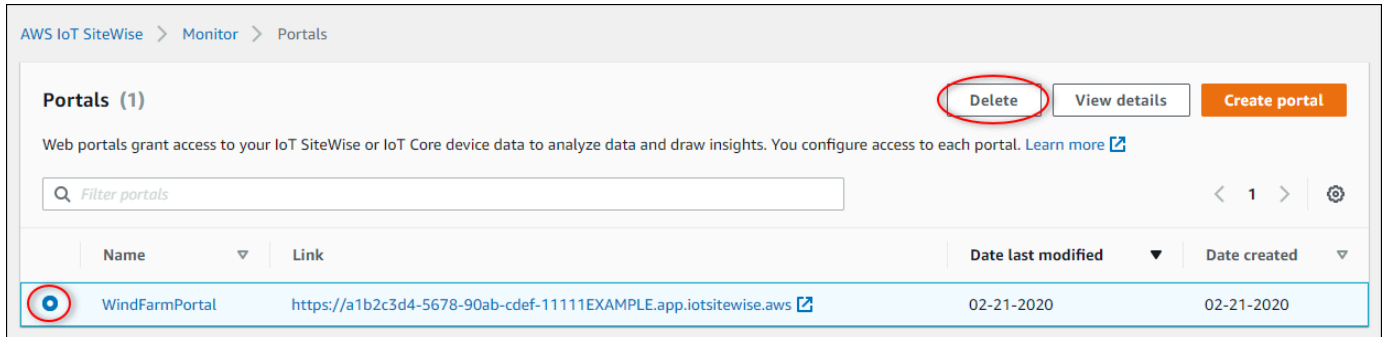
Gehen Sie wie folgt vor, um Ihre Portal- und IAM Identity Center-Benutzer zu löschen.

So löschen Sie ein Portal

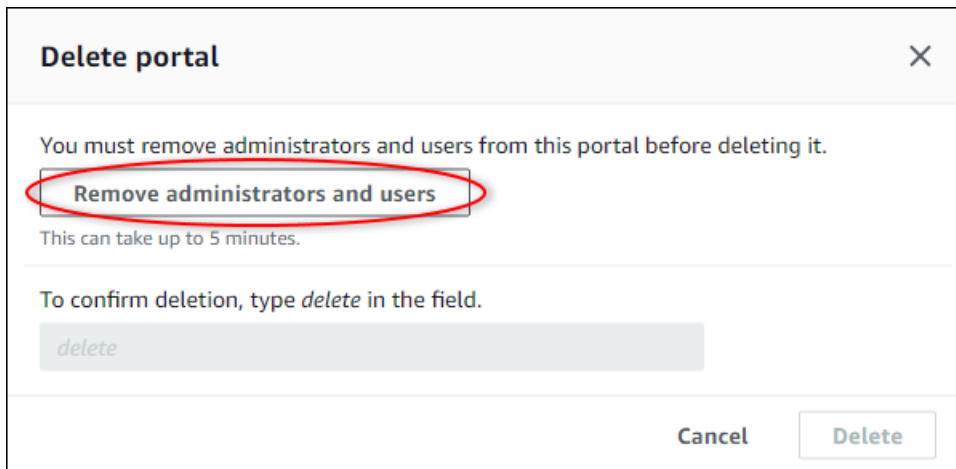
1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).

2. Wählen Sie im linken Navigationsbereich die Option Portale aus.
3. Wählen Sie Ihr Portal WindFarmPortal und anschließend Löschen aus.

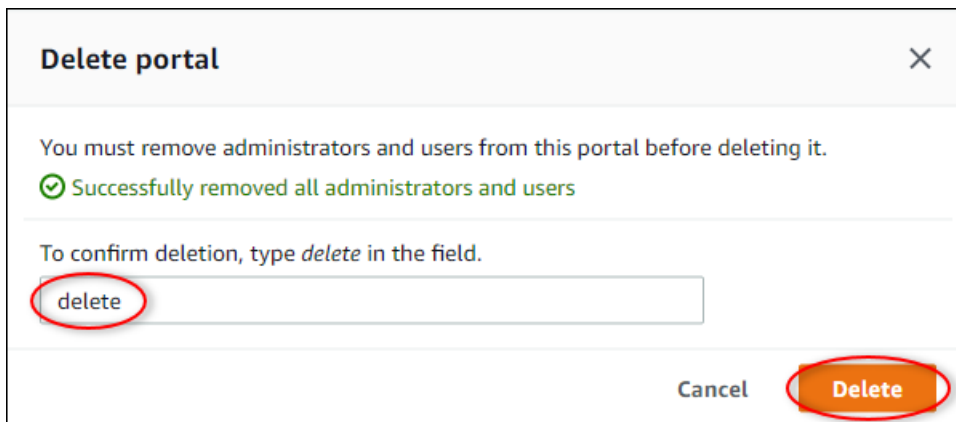
Wenn Sie ein Portal oder ein Projekt löschen, sind die Komponenten, die gelöschten Projekten zugeordnet sind, nicht betroffen.



4. Wählen Sie im Dialogfeld Portal löschen die Option Administratoren und Benutzer entfernen aus.

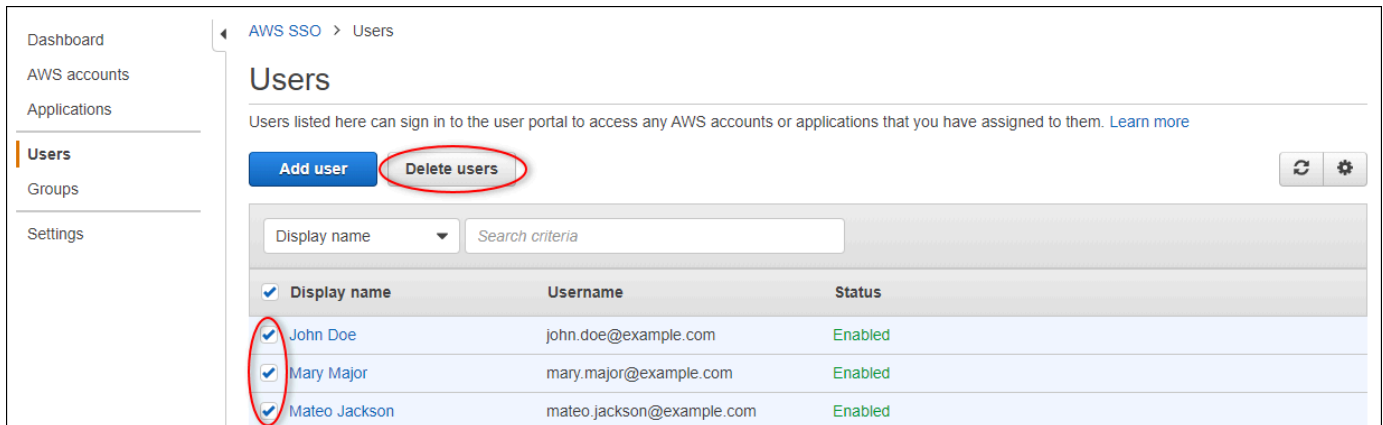


5. Geben Sie **delete** ein, um das Löschen zu bestätigen, und wählen Sie dann Löschen.

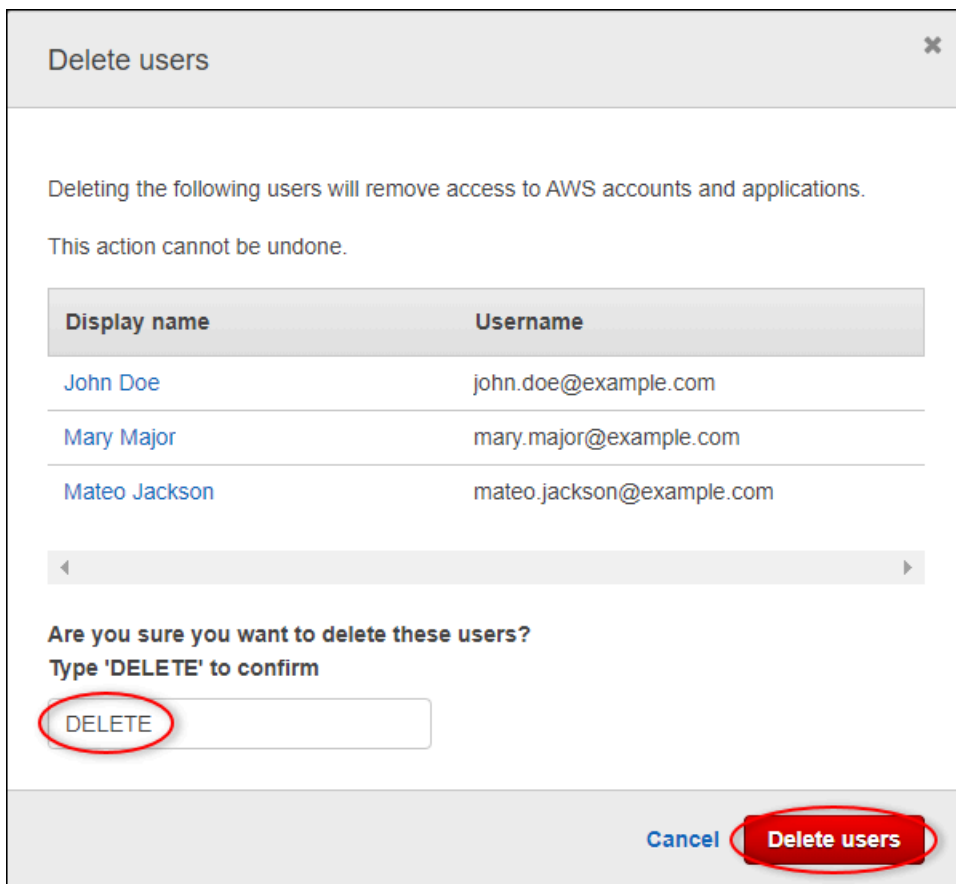


Um IAM Identity Center-Benutzer zu löschen

1. Navigieren Sie zur [IAM Identity Center-Konsole](#).
2. Wählen Sie im linken Navigationsbereich Benutzer aus.
3. Aktivieren Sie das Kontrollkästchen für jeden zu löschenden Benutzer und wählen Sie Benutzer löschen aus.



4. Geben **DELETE** im Dialogfeld „Benutzer löschen“ den Text ein und wählen Sie dann Benutzer löschen aus.



Aktualisierungen von Immobilienwerten in Amazon DynamoDB veröffentlichen

In diesem Tutorial wird eine bequeme Methode zum Speichern Ihrer Daten mithilfe von [Amazon DynamoDB](#) vorgestellt, sodass Sie einfacher auf historische Asset-Daten zugreifen können, ohne die API wiederholt abfragen zu müssen. AWS IoT SiteWise Nachdem Sie dieses Tutorial abgeschlossen haben, können Sie benutzerdefinierte Software erstellen, die Ihre Anlagendaten nutzt, z. B. eine Live-Karte der Windgeschwindigkeit und -richtung in einem gesamten Windpark. Wenn Sie Ihre Daten überwachen und visualisieren möchten, ohne eine benutzerdefinierte Softwarelösung zu implementieren, finden Sie weitere Informationen unter [Überwachen Sie Daten mit AWS IoT SiteWise Monitor](#).

In diesem Tutorial bauen Sie auf der AWS IoT SiteWise Demo auf, die einen Beispieldatensatz für einen Windpark enthält. Sie konfigurieren Eigenschaftswertaktualisierungen aus der Windpark-Demo, um Daten über AWS IoT Kernregeln an eine von Ihnen DynamoDB DynamoDB-Tabelle zu senden. Wenn Sie Eigenschaftswertaktualisierungen aktivieren, AWS IoT SiteWise sendet Ihre Daten AWS IoT Core in MQTT-Nachrichten an. Definieren Sie dann AWS IoT Core-Regeln, die je nach Inhalt dieser Nachrichten Aktionen ausführen, z. B. die DynamoDB-Aktion. Weitere Informationen finden Sie unter [Interagiere mit anderen AWS Diensten](#).

Themen

- [Voraussetzungen](#)
- [Schritt 1: Konfigurieren Sie die Konfiguration AWS IoT SiteWise , um Aktualisierungen von Eigenschaftswerten zu veröffentlichen](#)
- [Schritt 2: Erstellen Sie eine Regel in Core AWS IoT](#)
- [Schritt 3: Konfiguration der DynamoDB-Regelaktion](#)
- [Schritt 4: Erkunden Sie Daten in DynamoDB](#)
- [Bereinigen von Ressourcen](#)
- [Weitere Ressourcen](#)

Voraussetzungen

Zum Durcharbeiten dieses Tutorials ist Folgendes erforderlich:

- Ein AWS-Konto Falls Sie noch keines haben, beachten Sie die Informationen unter [Richten Sie eine einAWSKonto](#).
- Ein Entwicklungscomputer, auf dem Windows, macOS, Linux oder Unix ausgeführt wird, um auf die zuzugreifen AWS-Managementkonsole. Weitere Informationen finden Sie unter [Was ist der AWS-Managementkonsole?](#)
- Ein AWS Identity and Access Management (IAM-) Benutzer mit Administratorrechten. Detaillierte Anweisungen finden Sie unter [the section called “WieAWS IoT SiteWisefunktioniert mit IAM”](#).
- Eine laufende AWS IoT SiteWise Demo. Weitere Informationen finden Sie unter [Benutze die AWS IoT SiteWise Demo](#).

 Note

Dieses Tutorial erfordert die Verwendung von Ressourcen, die in der erstellt wurden [Benutze die AWS IoT SiteWise Demo](#). Sie müssen es abschließen, bevor Sie mit diesem Tutorial fortfahren können.

 Important

Behalten Sie alle Demo-Ressourcen, bis Sie dieses Tutorial abgeschlossen haben. Das Löschen von Komponenten kann die Funktionalität der Demo beeinträchtigen und Ihre Fähigkeit, das Tutorial abzuschließen, beeinträchtigen.

Schritt 1: Konfigurieren Sie die Konfiguration AWS IoT SiteWise , um Aktualisierungen von Eigenschaftswerten zu veröffentlichen

In diesem Verfahren aktivieren Sie Benachrichtigungen über Eigenschaftswerte für die Eigenschaften Wind Speed Ihrer Demo-Turbinenkomponenten. Nachdem Sie Benachrichtigungen über Eigenschaftswerte aktiviert haben, AWS IoT SiteWise veröffentlicht Sie jede Wertaktualisierung in einer MQTT-Nachricht in AWS IoT Core.

So aktivieren Sie Benachrichtigungen über Eigenschaftswerte für Komponenteneigenschaften:

1. Melden Sie sich bei der [AWS IoT SiteWise -Konsole](#) an.

2. Überprüfen Sie die [AWS IoT SiteWise Endpunkte und Kontingente](#), auf denen dies unterstützt AWS IoT SiteWise wird, und wechseln Sie AWS gegebenenfalls zwischen den Regionen. Wechseln Sie zu einer Region, in der Sie die AWS IoT SiteWise Demo ausführen.
3. Klicken Sie im linken Navigationsbereich auf Assets (Komponenten).
4. Wählen Sie den Pfeil neben Demo Wind Farm Asset aus, um die Hierarchie der Windparkkomponente zu erweitern.
5. Wählen Sie eine Demoturbine und Edit (Bearbeiten) aus.
6. Wählen Sie Measurements (Messungen).
7. Aktualisieren Sie den MQTT-Benachrichtigungsstatus der Wind SpeedImmobilie auf AKTIV.
8. Wählen Sie unten auf der Seite die Option Save aus.
9. Wiederholen Sie die Schritte 5 bis 7 für jede Demo-Turbinenkomponente.
10. Wählen Sie eine Demoturbine aus (z. B. Demo Turbine Asset 1).
11. Wählen Sie Measurements (Messungen).
12. Wählen Sie das Kopiersymbol neben der Eigenschaft Wind Speed aus, um das Benachrichtigungsthema in die Zwischenablage zu kopieren. Speichern Sie das Benachrichtigungsthema zur späteren zu verwendende Verwendung in diesem Tutorial. Sie müssen nur das Benachrichtigungsthema einer Turbine aufzeichnen.

Das Benachrichtigungsthema sollte wie im folgenden Beispiel aussehen.

```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE
```

Schritt 2: Erstellen Sie eine Regel in Core AWS IoT

In diesem Schritt erstellen Sie eine Regel in AWS IoT Core, die die Benachrichtigungen über den Eigenschaftswert analysiert und Daten in eine Amazon DynamoDB-Tabelle einfügt. AWS IoT Kernregeln analysieren MQTT-Nachrichten und führen Aktionen aus, die auf dem Inhalt und dem Thema jeder Nachricht basieren. Anschließend erstellen Sie eine Regel mit einer DynamoDB-Aktion, um Daten in eine DynamoDB-Tabelle einzufügen, die Sie im Rahmen dieses Tutorials erstellen.

So erstellen Sie eine Regel mit einer DynamoDB-Aktion

1. Navigieren Sie zur [AWS IoT -Konsole](#).

2. Wählen Sie im linken Navigationsbereich Message Routing und dann Rules aus.
3. Wählen Sie Regel erstellen aus.
4. Geben Sie unter Regeleigenschaften einen Namen und eine Beschreibung für die Regel ein.
5. Suchen Sie das Benachrichtigungsthema, das Sie zuvor in diesem Tutorial gespeichert haben.

```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE
```

Ersetzen Sie die Asset-ID (die ID danach `assets/`) im Thema durch eine `+`. Dadurch wird die Windgeschwindigkeit-Eigenschaft für alle Demo-Windturbinenanlagen ausgewählt. Der `+`-Themenfilter akzeptiert alle Knoten einer einzelnen Ebene in einem Thema. Ihr Thema sollte wie das folgende Beispiel aussehen.

```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/+/
properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

6. Geben Sie die folgende Regelabfrageanweisung ein. Ersetzen Sie das Thema im FROM-Abschnitt durch Ihr Benachrichtigungsthema.

```
SELECT
  payload.assetId AS asset,
  (SELECT VALUE (value.doubleValue) FROM payload.values) AS windspeed,
  timestamp() AS timestamp
FROM
  '$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/+/
properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE'
WHERE
  type = 'PropertyValueUpdate'
```

7. Navigieren Sie unter Regelaktionen zu Aktion 1.
8. Wählen Sie auf der Seite Aktion auswählen die Option Dynamo DBv2 aus. Dadurch wird die Nachricht in mehrere Spalten einer DynamoDB-Tabelle aufgeteilt
9. Wählen Sie unter Tabellename die Option Neue Tabelle erstellen aus. Sie erstellen eine Amazon DynamoDB-Tabelle, um Windgeschwindigkeitsdaten aus der Regelaktion zu erhalten.
10. Geben Sie in der [DynamoDB-Konsole](#) unter Tabellename einen Namen für Ihre Tabelle ein.
11. Gehen Sie für den Partitionsschlüssel wie folgt vor:

- a. Geben Sie „**timestamp**“ als Partitionsschlüssel ein.
 - b. Wählen Sie den Typ Number (Nummer) aus.
 - c. Aktivieren Sie das Kontrollkästchen Add sort key (Sortierschlüssel hinzufügen).
 - d. Geben Sie **asset** als Sortierschlüssel ein, und belassen Sie den Standardsortierschlüsseltyp auf String (Zeichenfolge).
12. Wählen Sie Create table (Tabelle erstellen) aus.
 13. Kehren Sie mit der Seite Configure action (Aktion konfigurieren) zur Registerkarte zurück.
 14. Aktualisieren Sie auf der Aktionsseite „Regel anhängen“ die Liste mit den Tabellennamen und wählen Sie Ihre neue DynamoDB-Tabelle aus, die Sie im vorherigen Schritt erstellt haben.

Schritt 3: Konfiguration der DynamoDB-Regelaktion

In diesem Schritt konfigurieren Sie die Amazon DynamoDB DynamoDB-Regelaktion, um Daten aus Eigenschaftswertaktualisierungen in Ihre neue DynamoDB-Tabelle einzufügen.

So konfigurieren Sie die DynamoDB-Regelaktion

1. Wählen Sie Rolle erstellen aus, um eine IAM-Rolle zu erstellen, die AWS IoT Core-Zugriff zur Ausführung der Regelaktion gewährt.
2. Geben Sie einen Rollennamen ein, zum Beispiel. WindSpeedDataRole Wählen Sie Rolle erstellen aus.
3. Wählen Sie Weiter aus.
4. Wählen Sie unten auf der Seite Erstellen aus, um die Erstellung der Regel abzuschließen.

Ihre Demo-Asset-Daten sollten nun in Ihrer DynamoDB-Tabelle erscheinen.

Schritt 4: Erkunden Sie Daten in DynamoDB

Erkunden Sie in diesem Schritt die Windgeschwindigkeitsdaten der Demo-Assets in Ihrer neuen Amazon DynamoDB-Tabelle.

Um Asset-Daten in DynamoDB zu untersuchen

1. Kehren Sie zu der Registerkarte mit der geöffneten DynamoDB-Tabelle zurück.

2. Wählen Sie in der Tabelle, die Sie zuvor erstellt haben, die Registerkarte Tabellenelemente durchsuchen, um die Daten in der Tabelle anzuzeigen. Aktualisieren Sie die Seite, wenn keine Zeilen in der Tabelle angezeigt werden. Wenn nach einigen Minuten keine Zeilen angezeigt werden, finden Sie weitere Informationen unter [Problembehandlung bei einer Regel \(DynamoDB\)](#).
3. Wählen Sie in einer Zeile in der Tabelle das Bearbeitungssymbol aus, um die Daten zu erweitern.
4. Wählen Sie den Pfeil neben der Struktur windspeed aus, um die Liste der Datenpunkte für die Windgeschwindigkeit zu erweitern. Jede Liste enthält eine Reihe von Datenpunkten zur AWS IoT SiteWise Windgeschwindigkeit, an die die Windpark-Demo gesendet hat. Möglicherweise benötigen Sie ein anderes Datenformat, wenn Sie eine Regelaktion für Ihre eigene Verwendung einrichten. Weitere Informationen finden Sie unter [Benachrichtigungen über Vermögenseigenschaften abfragen in AWS IoT SiteWise](#).

Nachdem Sie das Tutorial abgeschlossen haben, können Sie die Regel deaktivieren oder löschen und Ihre DynamoDB-Tabelle löschen, um zusätzliche Gebühren zu vermeiden. Informationen zum Bereinigen Ihrer Ressourcen finden Sie unter [Bereinigen von Ressourcen](#).

Sie können auch lernen, wie Sie benutzerdefinierte Anwendungen erstellen, um diese Daten zu nutzen und zu visualisieren. Eine Anleitung zur Visualisierung von AWS IoT SiteWise Daten finden Sie unter [Daten in Grafana visualisieren und teilen](#).

Bereinigen von Ressourcen

Nachdem Sie das Tutorial abgeschlossen haben, bereinigen Sie Ihre Ressourcen, um zusätzliche Kosten zu vermeiden.

Um die Demo zu löschen AWS IoT SiteWise

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie in der oberen rechten Ecke der Seite die Option Demo löschen aus.
3. Geben Sie in das Bestätigungsfeld **DELETE** ein, und wählen Sie dann Delete (Löschen).

Weitere Informationen finden Sie unter [Löschen Sie die AWS IoT SiteWise Demo](#).

Gehen Sie wie folgt vor, um Benachrichtigungen zur Aktualisierung von Eigenschaftswerten zu deaktivieren (falls Sie die Demo nicht gelöscht haben), Ihre AWS IoT Regel zu deaktivieren oder zu löschen und Ihre DynamoDB-Tabelle zu löschen.

So deaktivieren Sie Aktualisierungsbenachrichtigungen für Komponenteneigenschaften:

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Klicken Sie im linken Navigationsbereich auf Assets (Komponenten).
3. Wählen Sie den Pfeil neben Demo Wind Farm Asset aus, um die Hierarchie der Windparkkomponente zu erweitern.
4. Wählen Sie eine Demoturbine und Edit (Bearbeiten) aus.
5. Ändern Sie den Benachrichtigungsstatus der Wind SpeedImmobilie auf INAKTIV.
6. Wählen Sie unten auf der Seite die Option Save asset (Komponente speichern) aus.
7. Wiederholen Sie die Schritte 4 bis 6 für jede Demo-Turbinenkomponente.

Um eine Regel zu deaktivieren oder zu löschen AWS IoT Core

1. Navigieren Sie zur [AWS IoT -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Nachrichtenweiterleitung und dann Regeln aus.
3. Wählen Sie Ihre Regel aus und klicken Sie auf Löschen.
4. Geben Sie im Bestätigungsdialogfeld den Namen der Regel ein und wählen Sie dann Löschen.

So löschen Sie eine DynamoDB-Tabelle

1. Navigieren Sie zur [DynamoDB-Konsole](#).
2. Wählen Sie im linken Navigationsbereich Tables (Tabellen) aus.
3. Wählen Sie die Tabelle aus, die Sie zuvor erstellt haben, z. B. WindSpeedData
4. Wählen Sie Löschen aus.
5. Geben Sie im Bestätigungsdialogfeld ein, **confirm** um die Tabelle zu löschen.

Weitere Ressourcen

Weitere Informationen zur Arbeit mit DynamoDB und zur Überwachung Ihrer Daten finden Sie in den folgenden Ressourcen:

- [Inklusive Monitoring-Metriken CloudWatch im](#) DynamoDB Developer Guide
- [Bewährte Methoden für das Entwerfen und effektive Verwenden von Partitionsschlüsseln finden Sie im](#) DynamoDB-Entwicklerhandbuch
- [Regeln für AWS IoT](#) im Entwicklerhandbuch AWS IoT
- [Daten in Grafana visualisieren und teilen](#)

Daten aufnehmen in AWS IoT SiteWise

AWS IoT SiteWise wurde entwickelt, um Industriedaten effizient zu sammeln und mit entsprechenden Ressourcen zu korrelieren, die verschiedene Aspekte industrieller Abläufe repräsentieren. Diese Dokumentation konzentriert sich auf die praktischen Aspekte der Datenerfassung und bietet mehrere Methoden AWS IoT SiteWise, die auf unterschiedliche industrielle Anwendungsfälle zugeschnitten sind. Anweisungen zum Aufbau Ihrer virtuellen industriellen Operationen finden Sie unter [Modellieren Sie Industrieanlagen](#).

Sie können Industriedaten AWS IoT SiteWise mit einer der folgenden Optionen an senden:

- **AWS IoT SiteWise Edge** — Verwenden Sie [das SiteWise Edge-Gateway](#) als Vermittler zwischen AWS IoT SiteWise und Ihren Datenservern. AWS IoT SiteWise stellt AWS IoT Greengrass Komponenten bereit, die Sie auf jeder Plattform bereitstellen können, die AWS IoT Greengrass zur Einrichtung eines SiteWise Edge-Gateways ausgeführt werden kann. Diese Option unterstützt die Verknüpfung mit dem [OPC UA-Serverprotokoll](#).
- **AWS IoT SiteWise API** — Verwenden Sie die [AWS IoT SiteWise API](#), um Daten aus einer anderen Quelle hochzuladen. Verwenden Sie unsere [BatchPutAssetPropertyValue](#) Streaming-API für die Aufnahme innerhalb von Sekunden oder die stapelorientierte [CreateBulkImportJob](#) API, um eine kostengünstige Aufnahme in größeren Chargen zu ermöglichen.
- **AWS IoT Kernregeln** — Verwenden Sie [AWS IoT Kernregeln](#), um Daten aus MQTT-Nachrichten hochzuladen, die von einer Sache oder einem anderen Dienst veröffentlicht wurden. AWS IoT AWS
- **AWS IoT Greengrass Stream-Manager** — Verwenden Sie den [AWS IoT Greengrass Stream-Manager](#), um Daten aus lokalen Datenquellen mit einem Edge-Gerät hochzuladen. Diese Option eignet sich für Situationen, in denen Daten von lokalen oder Edge-Standorten stammen.

Diese Methoden bieten eine Reihe von Lösungen für die Verwaltung von Daten aus verschiedenen Quellen. Machen Sie sich mit den Einzelheiten der einzelnen Optionen vertraut, um sich ein umfassendes Bild von den Möglichkeiten der Datenaufnahme zu machen. AWS IoT SiteWise

Datenströme verwalten für AWS IoT SiteWise

Ein Datenstrom ist die Ressource, die historische Zeitreihendaten enthält. Jeder Datenstrom wird durch einen eindeutigen Alias identifiziert, was es einfacher macht, den Ursprung der einzelnen Daten zu verfolgen. Datenströme werden automatisch erstellt AWS IoT SiteWise, wenn die ersten

Zeitreihendaten empfangen werden. Wenn die ersten Zeitreihendaten mit einem Alias identifiziert werden, wird ein neuer Datenstrom mit diesem Alias AWS IoT SiteWise erstellt, sofern diesem Alias noch keine Asset-Eigenschaften zugewiesen wurden. Wenn die ersten Zeitreihendaten mit einer Objekt-ID und einer Eigenschafts-ID identifiziert wurden, AWS IoT SiteWise wird alternativ ein neuer Datenstrom erstellt und dieser Datenstrom der Anlageneigenschaft zugeordnet.

Es gibt zwei Möglichkeiten, einer Anlageneigenschaft einen Alias zuzuweisen. Die verwendete Methode hängt davon ab, ob Daten AWS IoT SiteWise zuerst gesendet oder zuerst ein Asset erstellt wird.

- Wenn Daten an AWS IoT SiteWise first gesendet werden, wird dadurch automatisch ein Datenstream mit dem zugewiesenen Alias erstellt. Wenn das Asset später erstellt wird, verwenden Sie die [AssociateTimeSeriesToAssetProperty](#)API, um den Datenstream und seinen Alias der Asset-Eigenschaft zuzuordnen.
- Wenn ein Asset zuerst erstellt wird, verwenden Sie die [UpdateAssetProperty](#)API, um einer Asset-Eigenschaft einen Alias zuzuweisen. Wenn später Daten an gesendet werden AWS IoT SiteWise, wird der Datenstrom automatisch erstellt und der Asset-Eigenschaft zugeordnet.

Derzeit können Sie Datenströme nur Messungen zuordnen. Bei Messungen handelt es sich um eine Art von Anlageneigenschaft, die die rohen Sensordatenströme von Geräten darstellt, z. B. Temperaturwerte mit Zeitstempel oder Werte für Umdrehungen pro Minute (U/min) mit Zeitstempel.

Wenn diese Messungen Metriken oder Transformationen definieren, lösen die eingehenden Daten spezifische Berechnungen aus. Es ist wichtig zu beachten, dass eine Anlageneigenschaft jeweils nur mit einem Datenstrom verknüpft werden kann.

AWS IoT SiteWise verwendet TimeSeries die Ressource Amazon Resource Name (ARN), um Ihre Lagergebühren zu ermitteln. Weitere Informationen finden Sie unter [AWS IoT SiteWise – Preise](#).

In den folgenden Abschnitten erfahren Sie, wie Sie die AWS IoT SiteWise Konsole oder API zur Verwaltung von Datenströmen verwenden.

Themen

- [Konfigurieren Sie Berechtigungen und Einstellungen](#)
- [Ordnen Sie einen Datenstrom einer Anlageneigenschaft zu](#)
- [Trennen Sie die Zuordnung eines Datenstroms zu einer Anlageneigenschaft](#)
- [Löschen Sie einen Datenstrom](#)

- [Aktualisieren Sie den Alias einer Asset-Eigenschaft](#)
- [Gängige Szenarien](#)

Konfigurieren Sie Berechtigungen und Einstellungen

Datenströme werden automatisch erstellt AWS IoT SiteWise, wenn die ersten Zeitreihendaten empfangen werden. Wenn die aufgenommenen Daten keiner Anlageneigenschaft zugeordnet sind, wird ein neuer getrennter Datenstrom AWS IoT SiteWise erstellt, der so konfiguriert werden kann, dass er einer Anlageneigenschaft zugeordnet werden kann. Konfigurieren Sie die Zugriffskontrolle des Gateways, an das Daten gesendet werden AWS IoT SiteWise, und geben Sie mithilfe von IAM-Richtlinien den Typ der Daten an, die aufgenommen werden sollen.

Die folgende IAM-Richtlinie deaktiviert die getrennte Datenaufnahme vom Gateway, ermöglicht aber weiterhin die Datenaufnahme in Datenstreams, die mit einer Asset-Eigenschaft verknüpft sind:

Example IAM-Benutzerrichtlinie, die die getrennte Datenaufnahme vom Gateway deaktiviert

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutAssetPropertyValuesUsingAssetIdAndPropertyId",
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "arn:aws:iotsitewise:*:*:asset/*"
    },
    {
      "Sid": "AllowPutAssetPropertyValuesUsingAliasWithAssociatedAssetProperty",
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "arn:aws:iotsitewise:*:*:time-series/*",
      "Condition": {
        "StringLikeIfExists": {
          "iotsitewise:isAssociatedWithAssetProperty": "true"
        }
      }
    }
  ],
  {
```

```

    "Sid": "DenyPutAssetPropertyValuesUsingAliasWithNoAssociatedAssetProperty",
    "Effect": "Deny",
    "Action": "iotsitewise:BatchPutAssetPropertyValue",
    "Resource": "arn:aws:iotsitewise:*:*:time-series/*",
    "Condition": {
      "StringLikeIfExists": {
        "iotsitewise:isAssociatedWithAssetProperty": "false"
      }
    }
  ]
}

```

Example IAM-Benutzerrichtlinie, die die gesamte Datenaufnahme über das Gateway deaktiviert

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyPutAssetPropertyValues",
      "Effect": "Deny",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": [
        "arn:aws:iotsitewise:*:*:asset/*",
        "arn:aws:iotsitewise:*:*:time-series/*"
      ]
    }
  ]
}

```

Ordnen Sie einen Datenstrom einer Anlageneigenschaft zu

Verwalten Sie Ihre Datenströme mit dem AWS-IoT-SiteWise-Konsole oder AWS CLI.

Console

Verwenden Sie die AWS IoT SiteWise Konsole, um Ihre Datenströme zu verwalten.

Um Datenströme zu verwalten (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Datenströme aus.
3. Wählen Sie einen Datenstrom aus, indem Sie entweder nach einem Datenstream-Alias filtern oder im Dropdownmenü Filter die Option Nicht zugeordnete Datenströme auswählen.
4. Wählen Sie den zu aktualisierenden Datenstrom aus. Sie können mehrere Datenströme auswählen. Klicken Sie oben rechts auf Datenströme verwalten.
5. Wählen Sie unter Datenstream-Zuordnungen aktualisieren den Datenstrom aus, der verknüpft werden soll, und klicken Sie auf die Schaltfläche Messung auswählen.
6. Suchen Sie im Bereich „Messung auswählen“ nach der entsprechenden Eigenschaft zur Messung von Vermögenswerten. Wählen Sie die Kennzahl aus und klicken Sie dann auf Auswählen.
7. Führen Sie die Schritte 4 und 5 für andere in Schritt 3 ausgewählte Datenströme aus. Weisen Sie allen Datenströmen Asset-Eigenschaften zu.
8. Wählen Sie „Aktualisieren“, um die Änderungen zu übernehmen. Zur Bestätigung der Aktualisierung wird ein Bestätigungsbanner angezeigt.

AWS CLI

Um einen Datenstrom (identifiziert durch seinen Alias) einer Asset-Eigenschaft (identifiziert durch seinen IDs) zuzuordnen, führen Sie den folgenden Befehl aus:

```
aws iotsitewise associate-time-series-to-asset-property \  
  --alias <data-stream-alias> \  
  --assetId <asset-ID> \  
  --propertyId <property-ID>
```

Trennen Sie die Zuordnung eines Datenstroms zu einer Anlageneigenschaft

Console

Verwenden Sie die AWS IoT SiteWise Konsole, um Ihren Datenstrom von einer Objekteigenschaft zu trennen.

So trennen Sie Datenstreams von einer Asset-Eigenschaft (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich die Option Datenströme aus.
3. Wählen Sie einen Datenstrom aus, indem Sie entweder nach einem Datenstream-Alias filtern oder im Dropdownmenü Filter die Option Zugeordnete Datenströme auswählen.
4. Wählen Sie den Datenstrom aus, dessen Verknüpfung aufgehoben werden soll. Die Aliasspalte für den Datenstrom muss einen Alias enthalten. Die Spalten Assetname und Asset-Eigenschaftsname müssen die Werte der Asset-Eigenschaft enthalten, mit der der Datenstream verknüpft ist. Sie können mehrere Datenströme auswählen.
5. Klicken Sie oben rechts auf Datenströme verwalten.
6. Klicken Sie im Abschnitt Datenstromzuordnungen aktualisieren in der Spalte Messname auf X. In der Spalte submitted Status sollte ein Status angezeigt werden.
7. Wählen Sie „Aktualisieren“, um die Änderungen zu übernehmen. Der Datenstrom ist jetzt von der Asset-Eigenschaft getrennt, und der Alias wird nun zur Identifizierung des Datenstroms verwendet.

AWS CLI

Führen Sie den folgenden Befehl aus, um einen Datenstrom von einer Asset-Eigenschaft (identifiziert durch sein ID s und seinen Alias) zu trennen:

```
aws iotsitewise disassociate-time-series-from-asset-property \  
  --alias <asset-property-alias> \  
  --assetId <asset-ID> \  
  --propertyId <property-ID>
```

Der Datenstrom ist jetzt von der Asset-Eigenschaft getrennt, und der Alias wird verwendet, um den Datenstrom zu identifizieren. Der Alias ist nicht mehr mit der Asset-Eigenschaft verknüpft, da er jetzt dem Datenstrom zugeordnet ist.

Löschen Sie einen Datenstrom

Wenn eine Eigenschaft aus einem Asset-Modell entfernt wird, werden die Eigenschaften und ihre Datenströme aus allen Assets AWS IoT SiteWise gelöscht, die vom Asset-Modell verwaltet werden.

Außerdem werden alle Eigenschaften und deren Datenströme eines Assets gelöscht, wenn das Asset gelöscht wird. Wenn Daten eines Datenstroms beibehalten werden müssen, müssen sie vor dem Löschen von der Objekteigenschaft getrennt werden.

Warning

Wenn eine Eigenschaft aus einem Asset gelöscht wird, wird auch der zugehörige Datenstrom gelöscht. Um den Datenstrom beizubehalten, trennen Sie ihn zunächst von der Asset-Eigenschaft, bevor Sie die Eigenschaft aus dem Asset-Modell löschen oder das Asset löschen.

Console

Verwenden Sie die AWS IoT SiteWise Konsole, um Ihren Datenstream von einer Anlageneigenschaft zu trennen.

Um einen Datenstream zu löschen (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Datenströme aus.
3. Wählen Sie einen Datenstrom aus, indem Sie nach dem Datenstream-Alias filtern.
4. Wählen Sie den zu löschenden Datenstrom aus. Sie können mehrere Datenströme auswählen.
5. Wählen Sie die Schaltfläche Löschen, um den Datenstrom zu löschen.

AWS CLI

Verwenden Sie die [DeleteTimeSeries](#) API, um einen bestimmten Datenstrom anhand seines Alias zu löschen.

```
aws iotsitewise delete-time-series \  
  --alias <data-stream-alias>
```

Aktualisieren Sie den Alias einer Asset-Eigenschaft

Aliase müssen innerhalb einer AWS Region eindeutig sein. Dazu gehören Aliase sowohl für Asset-Eigenschaften als auch für Datenströme. Weisen Sie einer Asset-Eigenschaft keinen Alias zu, wenn eine andere Eigenschaft oder ein anderer Datenstrom diesen Alias verwendet.

Console

Verwenden Sie die AWS IoT SiteWise Konsole, um einen Alias für eine Asset-Eigenschaft zu aktualisieren.

Um einen Alias für eine Asset-Eigenschaft zu aktualisieren (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Komponenten aus.
3. Wählen Sie das Asset aus der Tabelle aus.
4. Klicken Sie auf die Schaltfläche Edit (Bearbeiten).
5. Wählen Sie den Eigenschaftstyp in der Tabelle Eigenschaften aus.
6. Suchen Sie die Eigenschaft und geben Sie den neuen Alias in das Textfeld Eigenschaftsalias ein.
7. Klicken Sie auf die Schaltfläche Speichern, um die Änderungen zu speichern.

AWS CLI

Führen Sie den folgenden Befehl aus, um einen Alias für eine Asset-Eigenschaft zu aktualisieren:

```
aws iotsitewise update-asset-property \  
  --asset-id <asset-ID> \  
  --property-id <property-ID> \  
  --property-alias <asset-property-alias> \  
  --property-notification-state <ENABLED|DISABLED>
```

Note

Wenn Eigenschaftsbenachrichtigungen derzeit aktiviert sind, müssen sie erneut bereitgestellt werden, um sicherzustellen, dass sie weiterhin aktiviert sind.

Gängige Szenarien

Einen Datenstrom verschieben

Um die Zuordnung eines Datenstroms zu einer anderen Objekteigenschaft zu ändern, trennen Sie zunächst die Zuordnung des Datenstroms zur aktuellen Objekteigenschaft. Wenn Sie die Zuordnung eines Datenstroms zu einer Anlageneigenschaft aufheben, muss dieser Anlageneigenschaft ein Alias zugewiesen sein.

```
aws iotsitewise disassociate-time-series-from-asset-property \  
  --alias <asset-property-alias> \  
  --assetId <asset-ID> \  
  --propertyId <property-ID>
```

Ordnen Sie den Datenstrom nun der neuen Anlageneigenschaft erneut zu.

```
aws iotsitewise associate-time-series-from-asset-property \  
  --alias <data-stream-alias> \  
  --assetId <new-asset-ID> \  
  --propertyId <new-property-ID>
```

Fehler beim Zuweisen eines Alias zu einer Asset-Eigenschaft

Wenn Sie die `UpdateAssetProperty` API verwenden, um einer Eigenschaft einen Alias zuzuweisen, wird möglicherweise die folgende Fehlermeldung angezeigt:

```
Given alias <data-stream-alias> for property <property-name> with ID <property-ID>  
already in use by another property or data stream
```

Diese Fehlermeldung weist darauf hin, dass der Alias der Eigenschaft nicht zugewiesen ist, da er derzeit von einer anderen Eigenschaft oder einem Datenstrom verwendet wird.

Dies passiert, wenn Daten AWS IoT SiteWise mit einem Alias aufgenommen werden. Wenn Daten mit einem Alias gesendet werden, der nicht von einem anderen Datenstrom oder einer anderen Asset-Eigenschaft verwendet wird, wird ein neuer Datenstrom mit diesem Alias erstellt. Die beiden folgenden Optionen lösen das Problem.

- Verwenden Sie die `AssociateTimeSeriesToAssetProperty` API, um den Datenstrom mit seinem Alias der Asset-Eigenschaft zuzuordnen.
- Stoppen Sie vorübergehend die Datenaufnahme und löschen Sie den Datenstrom. Verwenden Sie die `UpdateAssetProperty` API, um der Asset-Eigenschaft den Alias zuzuweisen, und aktivieren Sie dann die Datenaufnahme wieder.

Fehler beim Zuordnen eines Datenstroms zu einer Asset-Eigenschaft

Beim Zuordnen eines Datenstroms zu einer Asset-Eigenschaft wird die folgende Fehlermeldung angezeigt.

```
assetProperty <property-name> with assetId <asset-ID> propertyId <property-ID> contains data
```

Diese Fehlermeldung weist darauf hin, dass die Asset-Eigenschaft bereits mit einem Datenstrom verknüpft ist, der Daten enthält. Dieser Datenstrom muss getrennt oder gelöscht werden, bevor dieser Asset-Eigenschaft ein anderer Datenstrom zugeordnet werden kann.

Note

Wenn die Zuordnung eines Datenstroms zu einer Anlageneigenschaft aufgehoben wird, wird der der Eigenschaft zugewiesene Alias dem Datenstrom zugewiesen. Damit dieser Alias der Eigenschaft weiterhin zugewiesen bleibt, weisen Sie dieser Eigenschaft einen neuen Alias zu, bevor Sie die Zuordnung zum Datenstrom aufheben.

Gehen Sie wie folgt vor, um die in der Asset-Eigenschaft gespeicherten Daten beizubehalten:

- Stellen Sie sicher, dass keine Daten in die Objekteigenschaft aufgenommen werden, um zu verhindern, dass ein neuer Datenstrom entsteht.
- Verwenden Sie die `UpdateAssetProperty` API, um einen neuen Alias festzulegen, der dem aktuell zugewiesenen Datenstrom zugewiesen wird.
- Verwenden Sie die `DisassociateTimeSeriesFromAssetProperty` API, um den aktuellen Datenstrom von der Asset-Eigenschaft zu trennen.
- Verwenden Sie die `AssociateTimeSeriesToAssetProperty` API, um den gewünschten Datenstrom der Asset-Eigenschaft zuzuordnen.

Wenn die in der Asset-Eigenschaft gespeicherten Daten gelöscht werden müssen, gehen Sie wie folgt vor:

- Stellen Sie sicher, dass keine Daten in die Objekteigenschaft aufgenommen werden, um zu verhindern, dass ein neuer Datenstrom entsteht.
- Verwenden Sie die `DeleteTimeSeries` API, um den aktuell zugewiesenen Datenstrom zu löschen.
- Verwenden Sie die `AssociateTimeSeriesToAssetProperty` API, um den gewünschten Datenstrom der Asset-Eigenschaft zuzuordnen.

Daten aufnehmen mit AWS IoT SiteWise APIs

Verwenden Sie diese AWS IoT SiteWise APIs Option, um Industriedaten mit Zeitstempel an die Attribut- und Messeigenschaften Ihrer Anlagen zu senden. Der APIs akzeptiert Strukturen, die Nutzdaten enthalten timestamp-quality-value (TQV).

BatchPutAssetPropertyValue API

Verwenden Sie die [BatchPutAssetPropertyValue](#)-Operation, um Ihre Daten hochzuladen. Mit diesem Vorgang können Sie mehrere Dateneinträge gleichzeitig hochladen, um Daten von mehreren Geräten zu sammeln und alles in einer einzigen Anfrage zu senden.

Important

Der [BatchPutAssetPropertyValue](#) Vorgang unterliegt den folgenden Kontingenten:

- Bis zu 10 [Einträge](#) pro Anfrage.
- Bis zu 10 [Eigenschaftswerte](#) (TQV-Datenpunkte) pro Eintrag.
- AWS IoT SiteWise lehnt alle Daten ab, deren Zeitstempel mehr als 7 Tage in der Vergangenheit oder mehr als 10 Minuten in der future liegt.

Weitere Informationen zu diesen Kontingenten finden Sie unter [BatchPutAssetPropertyValue](#) in der AWS IoT SiteWise -API-Referenz.

Um eine Vermögenseigenschaft zu identifizieren, geben Sie eine der folgenden Optionen an:

- Das `assetId` Ende `propertyId` der Anlageneigenschaft, an die Daten gesendet werden.
- The `propertyAlias`, bei dem es sich um einen Datenstream-Alias handelt (z. B. `/company/windfarm/3/turbine/7/temperature`). Um diese Option verwenden zu können, müssen Sie zuerst den Alias der Komponenteneigenschaft festlegen. Informationen zum Festlegen von Eigenschaftsaliasnamen finden Sie unter [Datenströme verwalten für AWS IoT SiteWise](#).

Im folgenden Beispiel wird veranschaulicht, wie die Messwerte einer Windkraftanlage für die Temperatur und die Umdrehungen pro Minute (U/min) aus Nutzlasten, die in einer JSON-Datei gespeichert sind, gesendet werden.

```
aws iotsitewise batch-put-asset-property-value --cli-input-json file://batch-put-payload.json
```

Die Beispiel-Payload in `batch-put-payload.json` hat den folgenden Inhalt.

```
{
  "enablePartialEntryProcessing": true,
  "entries": [
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/temperature",
      "propertyValues": [
        {
          "value": {
            "integerValue": 38
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    },
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/rpm",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 15.09
          },
          "timestamp": {
```

```

        "timeInSeconds": 1575691200
      },
      "quality": "GOOD"
    }
  ],
  },
  {
    "entryId": "unique entry ID",
    "propertyAlias": "/company/windfarm/3/turbine/7/rpm",
    "propertyValues": [
      {
        "value": {
          "nullValue":{"valueType": "D"}
        },
        "timestamp": {
          "timeInSeconds": 1575691200
        },
        "quality": "BAD"
      }
    ]
  }
]
}

```

Wenn Sie `enablePartialEntryProcessing` als `true` angeben, können alle Werte aufgenommen werden, die nicht zu einem Fehler führen. Das Standardverhalten ist `false`. Wenn ein Wert ungültig ist, schlägt die Aufnahme des gesamten Eintrags fehl.

Jeder Eintrag in der Nutzlast enthält eine `entryId`, die Sie als eindeutige Zeichenfolge definieren können. Bei fehlgeschlagenen Anforderungseinträgen enthält jeder Fehler die `entryId` der entsprechenden Anforderung, woran Sie erkennen können, welche Anforderungen zu wiederholen sind.

Jede Struktur in der Liste von `propertyValues` ist eine timestamp-quality-value (TQV-) Struktur, die `value`, `a` und optional `a timestamp` enthält. `quality`

- `value`— Eine Struktur, die je nach Typ der festzulegenden Eigenschaft eines der folgenden Felder enthält:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`

- `stringValue`
- `nullValue`
- `nullValue`— Eine Struktur mit dem folgenden Feld, das den Typ des Eigenschaftswerts mit dem Wert Null und der BAD Qualität oder bezeichnet. UNCERTAIN
 - `valueType`— Aufzählung von {"B", "D", "S", "I"}
- `timestamp`— Eine Struktur, die die aktuelle Unix-Epochezeit in Sekunden enthält, `timeInSeconds` Sie können den `offsetInNanos` Schlüssel auch in der `timestamp` Struktur angeben, wenn Sie über zeitlich genaue Daten verfügen. AWS IoT SiteWise lehnt alle Datenpunkte ab, deren Zeitstempel älter als 7 Tage in der Vergangenheit oder neuer als 10 Minuten in der future sind.
- `quality`— (Optional) Eine der folgenden Qualitätszeichenfolgen:
 - GOOD— (Standard) Die Daten sind von keinen Problemen betroffen.
 - BAD— Die Daten sind von einem Problem wie einem Sensorausfall betroffen.
 - UNCERTAIN— Die Daten sind von einem Problem wie einer Sensorungenauigkeit betroffen.

Weitere Informationen zum AWS IoT SiteWise Umgang mit Datenqualität bei Berechnungen finden Sie unter [Datenqualität in Formelausdrücken](#).

CreateBulkImportJob API

Verwenden Sie die `CreateBulkImportJob` API, um große Datenmengen aus Amazon S3 zu importieren. Ihre Daten müssen im CSV-Format in Amazon S3 gespeichert werden. Datendateien können die folgenden Spalten haben.

Note

Daten, die älter als der 1. Januar 1970 00:00:00 UTC sind, werden nicht unterstützt. Um eine Vermögenseigenschaft zu identifizieren, geben Sie eine der folgenden Optionen an.

- Das `ASSET_ID` Ende `PROPERTY_ID` der Anlageneigenschaft, an die Sie Daten senden.
- `TheALIAS`, bei dem es sich um einen Datenstream-Alias handelt (z. B. `/company/windfarm/3/turbine/7/temperature`). Um diese Option verwenden zu können, müssen Sie zuerst den Alias der Komponenteneigenschaft festlegen. Weitere Informationen zur Festlegung von Eigenschaftsaliasnamen finden Sie unter [the section called "Datenströme verwalten"](#).

- **ALIAS**— Der Alias, der die Eigenschaft identifiziert, z. B. ein Datenstream-Pfad eines OPC UA-Servers (zum Beispiel `/company/windfarm/3/turbine/7/temperature`). Weitere Informationen finden Sie unter [Datenströme verwalten für AWS IoT SiteWise](#).
- **ASSET_ID**— Die ID des Assets.
- **PROPERTY_ID**— Die ID der Anlageeigenschaft.
- **DATA_TYPE**— Der Datentyp der Eigenschaft kann einer der folgenden sein.
 - **STRING**— Eine Zeichenfolge mit bis zu 1024 Byte.
 - **INTEGER**— Eine 32-Bit-Ganzzahl mit Vorzeichen im Bereich $[-2.147.483.648, 2.147.483.647]$.
 - **DOUBLE**— Eine Fließkommazahl mit einem Bereich $[-10^{100}, 10^{100}]$ und einer doppelten IEEE-754-Genauigkeit.
 - **BOOLEAN**— `true` oder `false`
- **TIMESTAMP_SECONDS**— Der Zeitstempel des Datenpunkts in der Unix-Epochezeit.
- **TIMESTAMP_NANO_OFFSET**— Der Nanosekunden-Offset, aus dem konvertiert wurde. **TIMESTAMP_SECONDS**
- **QUALITY**— (Fakultativ) Die Qualität des Vermögenswerts. Der Wert kann einer der folgenden sein.
 - **GOOD**— (Standard) Die Daten sind von keinen Problemen betroffen.
 - **BAD**— Die Daten sind von einem Problem wie einem Sensorausfall betroffen.
 - **UNCERTAIN**— Die Daten sind von einem Problem wie einer Sensorungenauigkeit betroffen.

Weitere Informationen zum AWS IoT SiteWise Umgang mit Datenqualität bei Berechnungen finden Sie unter [Datenqualität in Formelausdrücken](#).

- **VALUE**— Der Wert der Vermögenseigenschaft.

Example Datendatei (en) im CSV-Format

```
asset_id,property_id,DOUBLE,1635201373,0,GOOD,1.0
asset_id,property_id,DOUBLE,1635201374,0,GOOD,2.0
asset_id,property_id,DOUBLE,1635201375,0,GOOD,3.0
```

```
unmodeled_alias1,DOUBLE,1635201373,0,GOOD,1.0
unmodeled_alias1,DOUBLE,1635201374,0,GOOD,2.0
unmodeled_alias1,DOUBLE,1635201375,0,GOOD,3.0
unmodeled_alias1,DOUBLE,1635201376,0,GOOD,4.0
unmodeled_alias1,DOUBLE,1635201377,0,GOOD,5.0
unmodeled_alias1,DOUBLE,1635201378,0,GOOD,6.0
```

```
unmodeled_alias1,DOUBLE,1635201379,0,GOOD,7.0
unmodeled_alias1,DOUBLE,1635201380,0,GOOD,8.0
unmodeled_alias1,DOUBLE,1635201381,0,GOOD,9.0
unmodeled_alias1,DOUBLE,1635201382,0,GOOD,10.0
```

AWS IoT SiteWise stellt die folgenden API-Operationen bereit, um einen Massenimportauftrag zu erstellen und Informationen über einen vorhandenen Auftrag abzurufen.

- [CreateBulkImportJob](#)— Erstellt einen neuen Massenimportauftrag.
- [DescribeBulkImportJob](#)— Ruft Informationen über einen Massenimportjob ab.
- [ListBulkImportJob](#)— Ruft eine paginierte Liste mit Zusammenfassungen aller Massenimportaufträge ab.

Erstellen Sie einen AWS IoT SiteWise Massenimportauftrag ()AWS CLI

Verwenden Sie den [CreateBulkImportJob](#) API-Vorgang, um Daten von Amazon S3 zu zu übertragen AWS IoT SiteWise. Die [CreateBulkImportJob](#) API ermöglicht die Aufnahme großer Mengen historischer Daten und die gepufferte Aufnahme analytischer Datenströme in kleinen Batches. Sie bietet ein kostengünstiges Primitiv für die Datenaufnahme. Das folgende Beispiel verwendet die AWS CLI.

Important

Bevor Sie einen Massenimportauftrag erstellen, müssen Sie AWS IoT SiteWise Warm Tier oder AWS IoT SiteWise Cold Tier aktivieren. Weitere Informationen finden Sie unter [Konfigurieren Sie die Speichereinstellungen in AWS IoT SiteWise](#).

Die [CreateBulkImportJob](#) API unterstützt die Aufnahme von historischen Daten AWS IoT SiteWise mit der Option, den Parameter festzulegen. `adaptive-ingestion-flag`

- Wenn diese Option auf gesetzt ist `false`, nimmt die API historische Daten auf, ohne Berechnungen oder Benachrichtigungen auszulösen.
- Wenn diese Option aktiviert ist `true`, nimmt die API neue Daten auf, berechnet Metriken und transformiert die Daten, um die laufenden Analysen und Benachrichtigungen innerhalb von sieben Tagen zu optimieren.

Führen Sie den folgenden Befehl aus. *file-name* Ersetzen Sie es durch den Namen der Datei, die die Konfiguration des Massenimportauftrags enthält.

```
aws iotsitewise create-bulk-import-job --cli-input-json file://file-name.json
```

Example Konfiguration für den Massenimport-Job

Im Folgenden finden Sie Beispiele für Konfigurationseinstellungen:

- Ersetzen Sie *adaptive-ingestion-flag* durch `true` oder `false`.
 - Wenn diese Option auf `false` gesetzt ist, nimmt der Massenimportjob historische Daten in AWS IoT SiteWise auf.
 - Wenn diese Option auf `true` gesetzt ist, führt der Massenimportjob Folgendes aus:
 - Nimmt neue Daten auf in AWS IoT SiteWise.
 - Berechnet Kennzahlen und Transformationen und unterstützt Benachrichtigungen für Daten mit einem Zeitstempel innerhalb von sieben Tagen.
- *delete-files-after-import-flag* Ersetzen Sie durch `true`, um die Daten aus dem Amazon S3 S3-Daten-Bucket zu löschen, nachdem sie in den AWS IoT SiteWise Warm-Tier-Speicher aufgenommen wurden.
- Ersetzen Sie `amzn-s3-demo-bucket` *-for-errors* durch den Namen des Amazon S3 S3-Buckets, an den Fehler im Zusammenhang mit diesem Massenimportauftrag gesendet werden.
- Ersetzen Sie `amzn-s3-demo-bucket` *-for-errors-prefix* durch das Präfix des Amazon S3 S3-Buckets, an den Fehler im Zusammenhang mit diesem Massenimportauftrag gesendet werden.

Amazon S3 verwendet das Präfix als Ordnernamen, um Daten im Bucket zu organisieren. Jedes Amazon S3 S3-Objekt hat einen Schlüssel, der seine eindeutige Kennung im Bucket ist. Jedes Objekt in einem Bucket besitzt genau einen Schlüssel. Das Präfix muss mit einem Schrägstrich enden (/). Weitere Informationen finden Sie unter [Objekte mithilfe von Präfixen organisieren](#) im Amazon Simple Storage Service-Benutzerhandbuch.

- Ersetzen Sie `amzn-s3-demo-bucket` *-data* durch den Namen des Amazon S3 S3-Buckets, aus dem Daten importiert werden.
- *data-bucket-key* Ersetzen Sie es durch den Schlüssel des Amazon S3 S3-Objekts, das Ihre Daten enthält. Jedes Objekt hat einen Schlüssel, der eine eindeutige Kennung ist. Jedes Objekt hat genau einen Schlüssel.
- *data-bucket-version-id* Ersetzen Sie es durch die Versions-ID, um eine bestimmte Version des Amazon S3 S3-Objekts zu identifizieren, das Ihre Daten enthält. Dieser Parameter ist optional.
- *column-name* Ersetzen Sie durch den in der CSV-Datei angegebenen Spaltennamen.

- **job-name** Ersetzen Sie ihn durch einen eindeutigen Namen, der den Massenimportauftrag identifiziert.
- **job-role-arn** Ersetzen Sie es durch die IAM-Rolle, die das Lesen von Amazon S3 S3-Daten ermöglicht AWS IoT SiteWise .

Note

Stellen Sie sicher, dass Ihre Rolle über die im folgenden Beispiel gezeigten Berechtigungen verfügt. Ersetzen Sie `amzn-s3-demo-bucket-data` durch den Namen des Amazon S3 S3-Buckets, der Ihre Daten enthält. `amzn-s3-demo-bucket-for-errors` Ersetzen Sie es außerdem durch den Namen des Amazon S3 S3-Buckets, an den Fehler im Zusammenhang mit diesem Massenimportauftrag gesendet werden.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket-data",
        "arn:aws:s3:::amzn-s3-demo-bucket-data/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket-for-errors",
        "arn:aws:s3:::amzn-s3-demo-bucket-for-errors/*"
      ],
      "Effect": "Allow"
    }
  ]
}
```



```
    }
  ]
}
```

```
{
  "adaptiveIngestion": adaptive-ingestion-flag,
  "deleteFilesAfterImport": delete-files-after-import-flag,
  "errorReportLocation": {
    "bucket": "amzn-s3-demo-bucket-for-errors",
    "prefix": "amzn-s3-demo-bucket-for-errors-prefix"
  },
  "files": [
    {
      "bucket": "amzn-s3-demo-bucket-data",
      "key": "data-bucket-key",
      "versionId": "data-bucket-version-id"
    }
  ],
  "jobConfiguration": {
    "fileFormat": {
      "csv": {
        "columnNames": [ "column-name" ]
      }
    }
  },
  "jobName": "job-name",
  "jobRoleArn": "job-role-arn"
}
```

Example response

```
{
  "jobId": "f8c031d0-01d1-4b94-90b1-afe8bb93b7e5",
  "jobStatus": "PENDING",
  "jobName": "myBulkImportJob"
}
```

Beschreiben Sie einen AWS IoT SiteWise Massenimportauftrag (AWS CLI)

Verwenden Sie den [DescribeBulkImportJob](#) API-Vorgang, um Informationen zu einem bestimmten Massenimportauftrag in abzurufen AWS IoT SiteWise. Dieser Vorgang gibt Details wie den Status des Jobs, die Erstellungszeit und Fehlerinformationen zurück, falls der Job fehlgeschlagen ist. Sie können diesen Vorgang verwenden, um den Auftragsfortschritt zu überwachen und Probleme zu beheben. Zur Verwendung `DescribeBulkImportJob` benötigen Sie die Auftrags-ID des `CreateBulkImportJob` Vorgangs. Die API gibt folgende Informationen zurück:

- Liste der importierten Dateien, einschließlich ihrer Amazon S3 S3-Bucket-Speicherorte und Schlüssel
- Speicherort des Fehlerberichts (falls zutreffend)
- Details zur Jobkonfiguration, wie Dateiformat und CSV-Spaltennamen
- Zeitstempel für die Erstellung von Job und die letzte Aktualisierung
- Aktueller Auftragsstatus (z. B. ob der Job in Bearbeitung, abgeschlossen oder fehlgeschlagen ist)
- Für den Importjob verwendeter ARN der IAM-Rolle

Überprüfen Sie bei abgeschlossenen Aufträgen die Ergebnisse, um die erfolgreiche Datenintegration zu bestätigen. Wenn ein Job fehlschlägt, überprüfen Sie die Fehlerdetails, um Probleme zu diagnostizieren und zu lösen.

job-ID Ersetzen Sie es durch die ID des Massenimportauftrags, den Sie abrufen möchten.

```
aws iotsitewise describe-bulk-import-job --job-id job-ID
```

Example response

```
{
  "files": [
    {
      "bucket": "amzn-s3-demo-bucket1",
      "key": "100Tags12Hours.csv"
    },
    {
      "bucket": "amzn-s3-demo-bucket2",
      "key": "BulkImportData1MB.csv"
    },
    {
```

```

        "bucket": " amzn-s3-demo-bucket3",
        "key": "UnmodeledBulkImportData1MB.csv"
    },
    "errorReportLocation": {
        "prefix": "errors/",
        "bucket": "amzn-s3-demo-bucket-for-errors"
    },
    "jobConfiguration": {
        "fileFormat": {
            "csv": {
                "columnNames": [
                    "ALIAS",
                    "DATA_TYPE",
                    "TIMESTAMP_SECONDS",
                    "TIMESTAMP_NANO_OFFSET",
                    "QUALITY",
                    "VALUE"
                ]
            }
        }
    },
    "jobCreationDate": 1645745176.498,
    "jobStatus": "COMPLETED",
    "jobName": "myBulkImportJob",
    "jobLastUpdateDate": 1645745279.968,
    "jobRoleArn": "arn:aws:iam::123456789012:role/DemoRole",
    "jobId": "f8c031d0-01d1-4b94-90b1-afe8bb93b7e5"
}

```

Auflisten von AWS IoT SiteWise Massenimportaufträgen (AWS CLI)

Verwenden Sie den [ListBulkImportJobs](#) API-Vorgang, um eine Liste mit Zusammenfassungen für Massenimportaufträge in AWS IoT SiteWise abzurufen. Dieser Vorgang bietet eine effiziente Möglichkeit, Ihre Datenimportprozesse zu überwachen und zu verwalten. Es gibt die folgenden Schlüsselinformationen für jeden Job zurück:

- Job-ID. Eine eindeutige Kennung für jeden Massenimportauftrag
- Name des Job. Der Name, den Sie dem Job bei der Erstellung zugewiesen haben
- Aktueller Status. Der aktuelle Status des Jobs (z. B. ABGESCHLOSSEN, LÄUFT, FEHLGESCHLAGEN)

ListBulkImportJobs ist besonders nützlich, um sich einen umfassenden Überblick über all Ihre Massenimportaufträge zu verschaffen. Dies kann Ihnen helfen, mehrere Datenimporte nachzuverfolgen, alle Jobs zu identifizieren, die Aufmerksamkeit erfordern, und einen organisierten Arbeitsablauf aufrechtzuerhalten. Der Vorgang unterstützt die Seitennummerierung, sodass Sie eine große Anzahl von Auftragszusammenfassungen effizient abrufen können. Sie können den von diesem Vorgang IDs zurückgegebenen Auftrag zusammen mit dem [DescribeBulkImportJob](#) Vorgang verwenden, um detailliertere Informationen zu bestimmten Aufträgen abzurufen. Dieser zweistufige Prozess ermöglicht es Ihnen, sich zunächst einen Überblick über alle Jobs zu verschaffen und anschließend die Details der Jobs, die für Sie von Interesse sind, aufzuschlüsseln. Bei der Verwendung können Sie Filter anwenden ListBulkImportJobs, um die Ergebnisse einzugrenzen. Sie können beispielsweise Jobs nach ihrem Status filtern, um nur abgeschlossene Jobs oder nur laufende Jobs abzurufen. Mit dieser Funktion können Sie sich auf die relevantesten Informationen für Ihre aktuelle Aufgabe konzentrieren. Der Vorgang gibt auch a zurücknextToken, wenn mehr Ergebnisse verfügbar sind. Sie können dieses Token in nachfolgenden Aufrufen verwenden, um den nächsten Satz von Auftragszusammenfassungen abzurufen, sodass Sie alle Ihre Massenimportaufträge durchlaufen können, auch wenn Sie über eine große Anzahl von Aufträgen verfügen. Das folgende Beispiel zeigt, wie Sie ListBulkImportJobs mit dem eine Liste AWS CLI abgeschlossener Jobs abrufen können.

```
aws iotsitewise list-bulk-import-jobs --filter COMPLETED
```

Example Filter „Antwort auf abgeschlossene Jobs“

```
{
  "jobSummaries":[
    {
      "id":"bdbbfa52-d775-4952-b816-13ba1c7cb9da",
      "name":"myBulkImportJob",
      "status":"COMPLETED"
    },
    {
      "id":"15ffc641-dbd8-40c6-9983-5cb3b0bc3e6b",
      "name":"myBulkImportJob2",
      "status":"COMPLETED"
    }
  ]
}
```

Mit diesem Befehl wird veranschaulicht, wie `ListBulkImportJobs` Sie eine Liste von Aufträgen abrufen können, die fehlerhaft abgeschlossen wurden. Das Maximum ist auf 50 Ergebnisse festgelegt und wir verwenden ein Next-Token für paginierte Ergebnisse.

```
aws iotsitewise list-bulk-import-jobs --filter COMPLETED_WITH_FAILURES --max-results 50  
--next-token "string"
```

Daten AWS IoT SiteWise mithilfe AWS IoT Core von Regeln aufnehmen

Senden Sie Daten AWS IoT SiteWise an AWS IoT Dinge und andere AWS Dienste mithilfe von Regeln in AWS IoT Core. Regeln transformieren MQTT-Nachrichten und führen Aktionen aus, um mit AWS Diensten zu interagieren. Die AWS IoT SiteWise Regelaktion leitet Nachrichtendaten von der API an den [BatchPutAssetPropertyValue](#) Vorgang weiter. AWS IoT SiteWise Weitere Informationen finden Sie unter [Regeln](#) und [AWS IoT SiteWise Maßnahmen](#) im AWS IoT Entwicklerhandbuch.

Ein Tutorial, in dem die Schritte beschrieben werden, die zum Einrichten einer Regel erforderlich sind, die Daten über Geräteschatten aufnimmt, finden Sie unter [Daten aufnehmen, um Dinge AWS IoT SiteWise zu erstellen AWS IoT](#).

Sie können Daten auch AWS IoT SiteWise an andere AWS Dienste senden. Weitere Informationen finden Sie unter [Interagiere mit anderen AWS Diensten](#).

Themen

- [Gewähren AWS IoT Sie den erforderlichen Zugriff](#)
- [Konfigurieren Sie die AWS IoT SiteWise Regelaktion](#)
- [Senken Sie die Kosten mit Basic Ingest in AWS IoT SiteWise](#)

Gewähren AWS IoT Sie den erforderlichen Zugriff

Sie verwenden IAM-Rollen, um die AWS Ressourcen zu steuern, auf die jede Regel Zugriff hat. Bevor Sie eine Regel erstellen, müssen Sie eine IAM-Rolle mit einer Richtlinie erstellen, die es der Regel ermöglicht, Aktionen für die erforderliche AWS Ressource auszuführen. AWS IoT nimmt diese Rolle bei der Ausführung einer Regel an.

Wenn Sie die Regelaktion in der AWS IoT Konsole erstellen, können Sie ein Root-Asset auswählen, um eine Rolle zu erstellen, die Zugriff auf eine ausgewählte Asset-Hierarchie hat.

Weitere Informationen zum manuellen Definieren einer Rolle für eine Regel finden Sie im AWS IoT Entwicklerhandbuch unter [Gewährung AWS IoT der erforderlichen Zugriffs](#) - und [Pass-Rollenberechtigungen](#).

Für die AWS IoT SiteWise Regelaktion müssen Sie eine Rolle definieren, die den `iotsitewise:BatchPutAssetPropertyValue` Zugriff auf die Asset-Eigenschaften ermöglicht, an die die Regel Daten sendet. Um die Sicherheit zu erhöhen, können Sie in der Condition Eigenschaft einen Pfad zur AWS IoT SiteWise Asset-Hierarchie angeben.

Die folgende Beispielvertrauensrichtlinie ermöglicht den Zugriff auf eine bestimmte Komponente und ihre untergeordneten Elemente.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iotsitewise:assetHierarchyPath": [
            "/root node asset ID",
            "/root node asset ID/*"
          ]
        }
      }
    }
  ]
}
```

Entfernen Sie das Condition aus der Richtlinie, um Zugriff auf all Ihre Ressourcen zu gewähren. Die folgende Beispielvertrauensrichtlinie ermöglicht den Zugriff auf alle Ihre Komponenten in der aktuellen Region.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "*"
    }
  ]
}
```

Konfigurieren Sie die AWS IoT SiteWise Regelaktion

Die AWS IoT SiteWise Regelaktion sendet Daten aus der MQTT-Nachricht, die die Regel initiiert hat, an die Asset-Eigenschaften in AWS IoT SiteWise. Sie können mehrere Dateneinträge gleichzeitig in verschiedene Asset-Eigenschaften hochladen, um Updates für alle Sensoren eines Geräts in einer Nachricht zu senden. Sie können für jede Dateneingabe auch mehrere Datenpunkte gleichzeitig hochladen.

Note

Wenn Sie AWS IoT SiteWise mit der Regelaktion Daten an senden, müssen Ihre Daten alle Anforderungen des BatchPutAssetPropertyValue Vorgangs erfüllen. Beispielsweise darf der Zeitstempel Ihrer Daten nicht früher als 7 Tage vor der aktuellen Unix-Epoche liegen. Weitere Informationen finden Sie unter [Erfassen von Daten mit der AWS IoT SiteWise -API](#).

Für jede Dateneingabe in der Regelaktion identifizieren Sie eine Komponenteneigenschaft und geben den Zeitstempel, die Qualität und den Wert jedes Datenpunkts für diese Komponenteneigenschaft an. Die Regelaktion erwartet Zeichenfolgen für alle Parameter.

Zur korrekten Identifizierung einer Komponenteneigenschaft in einer Eingabe können Sie eine der folgenden Angaben machen:

- Die Asset ID (Komponenten-ID) (`assetId`) und die Property ID (Eigenschaften-ID) (`propertyId`) der Komponenteneigenschaft, an die Sie Daten senden. Sie können die Asset-ID und die Property-

ID mithilfe der finden. AWS-IoT-SiteWise-Konsole Wenn Sie die Asset-ID kennen, können Sie den AWS CLI to call verwenden, [DescribeAsset](#)um die Immobilien-ID zu ermitteln.

- Das Property alias (Eigenschaftsalias) (propertyAlias), bei dem es sich um ein Datenstrom-Alias handelt (z. B. /company/windfarm/3/turbine/7/temperature). Um diese Option verwenden zu können, müssen Sie zuerst den Alias der Komponenteneigenschaft festlegen. Weitere Informationen zur Festlegung von Eigenschaftsaliasnamen finden Sie unter [Datenströme verwalten für AWS IoT SiteWise](#).

Verwenden Sie für den Zeitstempel in jedem Eintrag den von Ihrem Gerät gemeldeten Zeitstempel oder den von bereitgestellten Zeitstempel. AWS IoT Core Der Zeitstempel hat zwei Parameter:

- Zeit in Sekunden (timeInSeconds) — Die Unix-Epoche in Sekunden, zu der der Sensor oder das Gerät die Daten gemeldet hat.
- Offset in Nanos (offsetInNanos) — (Optional) Der Abstand zwischen Nanosekunden und der Zeit in Sekunden.

Important

Wenn Ihr Zeitstempel eine Zeichenfolge ist, einen Dezimalteil hat oder nicht in Sekunden angegeben ist, wird die Anfrage AWS IoT SiteWise zurückgewiesen. Sie müssen den Zeitstempel in Sekunden und Nanosekunden-Offset konvertieren. Verwenden Sie Funktionen der AWS IoT Regel-Engine, um den Zeitstempel zu konvertieren. Weitere Informationen finden Sie hier:

- [Abrufen von Zeitstempeln für Geräte, die keine genaue Uhrzeit melden](#)
- [Konvertierung von Zeitstempeln im Zeichenkettenformat](#)

Sie können Ersatzvorlagen für mehrere Parameter in der Aktion verwenden, um Berechnungen durchzuführen, Funktionen aufzurufen und Werte aus der Nachrichtennutzlast abzurufen. Weitere Informationen finden Sie im Entwicklerhandbuch unter [Substitutionsvorlagen](#).AWS IoT

Note

Da ein Ausdruck in einer Substitutionsvorlage getrennt von der SELECT-Anweisung ausgewertet wird, können Sie keine Substitutionsvorlage verwenden, um auf einen Alias

zu verweisen, der mit einer AS-Klausel erstellt wurde. Zusätzlich zu den unterstützten Funktionen und Operatoren können Sie nur in der ursprünglichen Nutzlast vorhandene Informationen referenzieren.

Themen

- [Abrufen von Zeitstempeln für Geräte, die keine genaue Uhrzeit melden](#)
- [Konvertierung von Zeitstempeln im Zeichenkettenformat](#)
- [Konvertierung von Zeitstempelzeichenfolgen mit einer Genauigkeit von Nanosekunden](#)
- [Beispiele für Regelkonfigurationen](#)
- [Problembehandlung bei der -Regelaktion](#)

Abrufen von Zeitstempeln für Geräte, die keine genaue Uhrzeit melden

Wenn Ihr Sensor oder Ihre Ausrüstung keine genauen Zeitdaten meldet, rufen Sie mit [timestamp\(\)](#) die aktuelle Unix-Epochenzeit von der AWS IoT Regel-Engine ab. Diese Funktion gibt die Zeit in Millisekunden aus. Sie müssen den Wert also in Zeit in Sekunden und den Offset in Nanosekunden umrechnen. Verwenden Sie dazu die folgenden Konvertierungen:

- Verwenden Sie für Time in seconds (Zeit in Sekunden) (`timeInSeconds`)
 $\text{\$}\{\text{floor}(\text{timestamp}() / 1\text{E}3)\}$, um die Zeit von Millisekunden in Sekunden zu konvertieren.
- Verwenden Sie für Offset in nanos (Verschiebung in Nanosekunden) (`offsetInNanos`)
 $\text{\$}\{(\text{timestamp}() \% 1\text{E}3) * 1\text{E}6\}$, um den Nanosekunden-Versatz des Zeitstempels zu berechnen.

Konvertierung von Zeitstempeln im Zeichenkettenformat

Wenn Ihr Sensor oder Ihre Ausrüstung Zeitdaten im Zeichenkettenformat meldet (z. B. `2020-03-03T14:57:14.699Z`), verwenden Sie [time_to_epoch](#) (String, String). Diese Funktion gibt den Zeitstempel und das Formatmuster als Parameter ein und gibt die Zeit in Millisekunden aus. Dann müssen Sie die Zeit in Zeit in Sekunden und den Offset in Nanosekunden umrechnen. Verwenden Sie dazu die folgenden Konvertierungen:

- Verwenden Sie für Time in seconds (`timeInSeconds`),
 $\text{\$}\{\text{floor}(\text{time\_to\_epoch}(\text{"2020-03-03T14:57:14.699Z"}, \text{"yyyy-MM-}$

`dd'T'HH:mm:ss'Z'") / 1E3))` um die Zeitstempelzeichenfolge in Millisekunden und dann in Sekunden zu konvertieren.

- Verwenden Sie für Offset in nanos (`offsetInNanos`), um den Nanosekunden-Offset der Zeitstempelzeichenfolge `${(time_to_epoch("2020-03-03T14:57:14.699Z", "yyyy-MM-dd'T'HH:mm:ss'Z'") % 1E3) * 1E6}` zu berechnen.

Note

Die `time_to_epoch` Funktion unterstützt Zeitstempelzeichenfolgen mit einer Genauigkeit von bis zu Millisekunden. Um Zeichenketten mit Mikro- oder Nanosekundengenauigkeit zu konvertieren, konfigurieren Sie eine AWS Lambda Funktion, die Ihre Regel aufruft, um den Zeitstempel in numerische Werte umzuwandeln. Weitere Informationen finden Sie unter [Konvertierung von Zeitstempelzeichenfolgen mit einer Genauigkeit von Nanosekunden](#).

Konvertierung von Zeitstempelzeichenfolgen mit einer Genauigkeit von Nanosekunden

Wenn Ihr Gerät Zeitstempelinformationen im Zeichenkettenformat mit einer Genauigkeit im Nanosekundenbereich sendet (z. B. `2020-03-03T14:57:14.699728491Z`), gehen Sie wie folgt vor, um Ihre Regelaktion zu konfigurieren. Sie können eine AWS Lambda Funktion erstellen, die den Zeitstempel aus einer Zeichenfolge in Zeit in Sekunden (`timeInSeconds`) und Offset in Nanos (`offsetInNanos`) umwandelt. Verwenden Sie dann [aws_lambda \(FunctionArn, inputJson\)](#) in Ihren Regelaktionsparametern, um diese Lambda-Funktion aufzurufen und die Ausgabe in Ihrer Regel zu verwenden.

Note

Dieser Abschnitt enthält erweiterte Anweisungen, die davon ausgehen, dass Sie mit dem Erstellen der folgenden Ressourcen vertraut sind:

- Lambda-Funktionen. Weitere Informationen finden Sie unter [Erstellen Ihrer ersten Lambda-Funktion](#) im Entwicklerhandbuch für AWS Lambda .
- AWS IoT Regeln mit der AWS IoT SiteWise Regelaktion. Weitere Informationen finden Sie unter [Daten AWS IoT SiteWise mithilfe AWS IoT Core von Regeln aufnehmen](#).

Um eine AWS IoT SiteWise Regelaktion zu erstellen, die Zeitstempelzeichenfolgen analysiert

1. Erstellen Sie eine Lambda-Funktion mit den folgenden Eigenschaften:

- Funktionsname — Verwenden Sie einen beschreibenden Funktionsnamen (z. **B.ConvertNanosecondTimestampFromString**).
- Runtime — Verwenden Sie eine Python-3-Runtime wie Python 3.11 (python3.11).
- Berechtigungen — Erstellen Sie eine Rolle mit grundlegenden Lambda-Berechtigungen (AWS LambdaBasicExecutionRole).
- Ebenen — Fügen Sie die Schicht AWS SDKPandas-Python311 hinzu, die von der Lambda-Funktion verwendet werden soll. numpy
- Funktionscode — Verwenden Sie den folgenden Funktionscode, der ein Zeichenkettenargument mit dem Namen verwendet timestamp und Werte für diesen Zeitstempel ausgibttimeInSeconds. offsetInNanos

```
import json
import math
import numpy

# Converts a timestamp string into timeInSeconds and offsetInNanos in Unix epoch
time.
# The input timestamp string can have up to nanosecond precision.
def lambda_handler(event, context):
    timestamp_str = event['timestamp']
    # Parse the timestamp string as nanoseconds since Unix epoch.
    nanoseconds = numpy.datetime64(timestamp_str, 'ns').item()
    time_in_seconds = math.floor(nanoseconds / 1E9)
    # Slice to avoid precision issues.
    offset_in_nanos = int(str(nanoseconds)[-9:])
    return {
        'timeInSeconds': time_in_seconds,
        'offsetInNanos': offset_in_nanos
    }
```

[Diese Lambda-Funktion gibt Zeitstempelzeichenfolgen im Format ISO 8601 unter Verwendung von datetime64 von ein. NumPy](#)

 Note

Wenn Ihre Zeitstempelzeichenfolgen nicht im ISO 8601-Format vorliegen, können Sie eine Lösung implementieren, die das Zeitstempelformat definiert. pandas [Weitere Informationen finden Sie unter pandas.to_datetime.](#)

2. Wenn Sie die AWS IoT SiteWise Aktion für Ihre Regel konfigurieren, verwenden Sie die folgenden Ersatzvorlagen für Zeit in Sekunden (**timeInSeconds**) und Offset in Nanos (**offsetInNanos**). Diese Ersetzungsvorlagen gehen davon aus, dass Ihre Nachrichtennutzlast die Zeitstempelzeichenfolge in `timestamp` enthält. Die `aws_lambda`-Funktion verwendet eine JSON-Struktur für ihren zweiten Parameter, so dass Sie die folgenden Ersetzungsvorlagen bei Bedarf ändern können.

- Verwenden Sie für Zeit in Sekunden (`timeInSeconds`) die folgende Ersetzungsvorlage.

```
${aws_lambda('arn:aws:lambda:region:account-id:function:ConvertNanosecondTimestampFromString', {'timestamp': timestamp}).timeInSeconds}
```

- Verwenden Sie für Verschiebung in Nanosekunden (`offsetInNanos`) die folgende Ersetzungsvorlage.

```
${aws_lambda('arn:aws:lambda:region:account-id:function:ConvertNanosecondTimestampFromString', {'timestamp': timestamp}).offsetInNanos}
```

Ersetzen Sie für jeden Parameter `region` und `account-id` durch Ihre Region und AWS Konto-ID. Wenn Sie einen anderen Namen für Ihre Lambda-Funktion verwendet haben, ändern Sie diesen ebenfalls.

3. Erteilen Sie mit der AWS IoT Erlaubnis Berechtigungen zum Aufrufen Ihrer Funktion. `lambda:InvokeFunction` Weitere Informationen finden Sie unter [aws_lambda\(functionArn, inputJson\)](#).
4. Testen Sie Ihre Regel (verwenden Sie beispielsweise den AWS IoT MQTT-Testclient) und stellen Sie sicher, dass die von Ihnen AWS IoT SiteWise gesendeten Daten empfangen werden.

Wenn Ihre Regel nicht wie erwartet funktioniert, finden Sie weitere Informationen unter [Problembehandlung bei einer AWS IoT SiteWise Regelaktion](#).

 Note

Diese Lösung ruft die Lambda-Funktion zweimal für jede Zeitstempelzeichenfolge auf. Sie können eine weitere Regel erstellen, um die Anzahl der Lambda-Funktionsaufrufen zu reduzieren, wenn Ihre Regel mehrere Datenpunkte verarbeitet, die in jeder Nutzlast denselben Zeitstempel haben.

Erstellen Sie dazu eine Regel mit einer Aktion zum erneuten Veröffentlichen, die Lambda aufruft und die ursprüngliche Nutzlast mit der in und konvertierten Zeitstempelzeichenfolge veröffentlicht. `timeInSeconds offsetInNanos` Erstellen Sie dann eine Regel mit einer AWS IoT SiteWise Regelaktion, um die konvertierte Payload zu verwenden. Mit diesem Ansatz reduzieren Sie die Häufigkeit, mit der die Regel Lambda aufruft, erhöhen jedoch die Anzahl der ausgeführten AWS IoT Regelaktionen. Berücksichtigen Sie die Preise für jeden Service, wenn Sie diese Lösung auf Ihren Anwendungsfall anwenden.

Beispiele für Regelkonfigurationen

Dieser Abschnitt enthält Beispielregelkonfigurationen zum Erstellen einer Regel mit einer AWS IoT SiteWise Aktion.

Example Beispielregelaktion, die Eigenschaftsaliasse als Nachrichtenthemen verwendet

Im folgenden Beispiel wird eine Regel mit einer AWS IoT SiteWise Aktion erstellt, die das Thema (über [topic \(\)](#)) als Eigenschaftsalias zur Identifizierung von Asset-Eigenschaften verwendet. Verwenden Sie dieses Beispiel, um eine Regel für die Aufnahme von Daten vom Typ Doppeltyp für alle Windturbinen in allen Windparks zu definieren. Für dieses Beispiel müssen Sie Eigenschaftsaliasnamen für die Eigenschaften aller Turbinenanlagen definieren. Sie müssten eine zweite, ähnliche Regel definieren, um Daten vom Typ Ganzzahl aufzunehmen.

```
aws iot create-topic-rule \  
  --rule-name SiteWiseWindFarmRule \  
  --topic-rule-payload file://sitewise-rule-payload.json
```

Die Beispielnutzlast in `sitewise-rule-payload.json` hat folgenden Inhalt.

```
{  
  "sql": "SELECT * FROM '/company/windfarm/+/turbine/+/+' WHERE type = 'double'",  
  "description": "Sends data to the wind turbine asset property with the same alias as  
the topic",
```

```

"ruleDisabled": false,
"awsIotSqlVersion": "2016-03-23",
"actions": [
  {
    "iotSiteWise": {
      "putAssetPropertyValueEntries": [
        {
          "propertyAlias": "${topic()}",
          "propertyValues": [
            {
              "timestamp": {
                "timeInSeconds": "${timeInSeconds}"
              },
              "value": {
                "doubleValue": "${value}"
              }
            }
          ]
        }
      ],
      "roleArn": "arn:aws:iam::account-id:role/MySiteWiseActionRole"
    }
  }
]
}

```

Senden Sie mit dieser Regelaktion die folgende Nachricht als Thema für die Datenaufnahme an einen Alias für eine Windenergieanlage (z. B. /company/windfarm/3/turbine/7/temperature).

```

{
  "type": "double",
  "value": "38.3",
  "timeInSeconds": "1581368533"
}

```

Example Beispielregelaktion, die timestamp() verwendet, um die Zeit zu bestimmen

Im folgenden Beispiel wird eine Regel mit einer AWS IoT SiteWise Aktion erstellt, die eine Anlageneigenschaft anhand von timestamp () identifiziert IDs und mithilfe von [timestamp \(\)](#) die aktuelle Uhrzeit bestimmt.

```
aws iot create-topic-rule \
  --rule-name SiteWiseAssetPropertyRule \
  --topic-rule-payload file:///sitewise-rule-payload.json
```

Die Beispielnutzlast in `sitewise-rule-payload.json` hat folgenden Inhalt.

```
{
  "sql": "SELECT * FROM 'my/asset/property/topic'",
  "description": "Sends device data to an asset property",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [
    {
      "iotSiteWise": {
        "putAssetPropertyValueEntries": [
          {
            "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
            "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
            "propertyValues": [
              {
                "timestamp": {
                  "timeInSeconds": "${floor(timestamp() / 1E3)}",
                  "offsetInNanos": "${(timestamp() % 1E3) * 1E6}"
                },
                "value": {
                  "doubleValue": "${value}"
                }
              }
            ]
          }
        ],
        "roleArn": "arn:aws:iam::account-id:role/MySiteWiseActionRole"
      }
    }
  ]
}
```

Senden Sie mit dieser Regelaktion die folgende Nachricht an den, `my/asset/property/topic` um Daten aufzunehmen.

```
{
  "type": "double",
```

```
"value": "38.3"
}
```

Problembehandlung bei der -Regelaktion

Um Ihre AWS IoT SiteWise Regelaktion zu beheben AWS IoT Core, konfigurieren Sie CloudWatch Protokolle oder konfigurieren Sie eine Aktion zum erneuten Veröffentlichen von Fehlern für Ihre Regel. Weitere Informationen finden Sie unter [Problembehandlung bei einer AWS IoT SiteWise Regelaktion](#).

Senken Sie die Kosten mit Basic Ingest in AWS IoT SiteWise

AWS IoT Core [bietet eine Funktion namens Basic Ingest, mit der Sie Daten senden können, AWS IoT Core ohne dass Messaging-Kosten anfallen.](#) AWS IoT Basic Ingest optimiert den Datenfluss für umfangreiche Datenaufnahme-Workloads, indem der Message Broker aus dem Aufnahmepfad entfernt wird. publish/subscribe Sie können Basic Ingest verwenden, wenn Sie wissen, an welche Regeln Ihre Nachrichten weitergeleitet werden sollen.

Um Basic Ingest zu verwenden, senden Sie Nachrichten direkt an eine bestimmte Regel mit einem speziellen Thema, `$aws/rules/rule-name`. Um beispielsweise eine Nachricht an eine Regel mit dem Namen `SiteWiseWindFarmRule` zu senden, senden Sie eine Nachricht an das Thema `$aws/rules/SiteWiseWindFarmRule`.

Wenn Ihre Regelaktion Substitutionsvorlagen verwendet, die [topic\(Decimal\) \(Thema \(Dezimal\)\)](#) enthalten, können Sie das ursprüngliche Thema am Ende des speziellen Basic Ingest-Themas übergeben, z. B. `$aws/rules/rule-name/original-topic`. Wenn Sie beispielsweise Basic Ingest mit dem Aliasbeispiel für Windparks aus dem vorherigen Abschnitt verwenden möchten, können Sie Nachrichten an das folgende Thema senden.

```
$aws/rules/SiteWiseWindFarmRule//company/windfarm/3/turbine/7/temperature
```

Note

Das obige Beispiel enthält einen zweiten Schrägstrich (`//`), weil das Basic Ingest-Präfix (`$aws/rules/rule-name/`) aus dem Thema AWS IoT entfernt wird, das für die Regelaktion sichtbar ist. In diesem Beispiel erhält die Regel das Thema `/company/windfarm/3/turbine/7/temperature`.

Weitere Informationen finden Sie im [Entwicklerhandbuch unter Senkung der Messaging-Kosten durch einfaches Ingest](#). AWS IoT

Verwenden Sie AWS IoT Greengrass den Stream-Manager in AWS IoT SiteWise

AWS IoT Greengrass Stream Manager ist eine Integrationsfunktion, die die Übertragung von Datenströmen aus lokalen Quellen in die AWS Cloud erleichtert. Er fungiert als Zwischenschicht, die Datenflüsse verwaltet und es Geräten, die am Netzwerkrand betrieben werden, ermöglicht, Daten zu sammeln und zu speichern, bevor sie an sie gesendet werden AWS IoT SiteWise, um sie weiter zu analysieren und zu verarbeiten.

Fügen Sie ein Datenziel hinzu, indem Sie eine lokale Quelle auf der AWS IoT SiteWise Konsole konfigurieren. Sie können den Stream Manager auch in Ihrer benutzerdefinierten AWS IoT Greengrass Lösung verwenden, um AWS IoT SiteWise Daten aufzunehmen.

Note

Um Daten aus OPC UA-Quellen aufzunehmen, konfigurieren Sie ein AWS IoT SiteWise Edge-Gateway, das auf läuft. AWS IoT Greengrass Weitere Informationen finden Sie unter [Verwenden Sie AWS IoT SiteWise Edge-Gateways](#).

Weitere Informationen zur Konfiguration eines Ziels für lokale Quelldaten finden Sie unter. [Verstehen Sie AWS IoT SiteWise Edge-Ziele](#)

Weitere Informationen zum Ingestieren von Daten mithilfe des Stream-Managers in einer benutzerdefinierten AWS IoT Greengrass Lösung finden Sie in den folgenden Themen im AWS IoT Greengrass Version 2 Entwicklerhandbuch:

- [Was ist AWS IoT Greengrass?](#)
- [Datenströme auf dem AWS IoT Greengrass Core verwalten](#)
- [Daten in AWS IoT SiteWise Asset-Eigenschaften exportieren](#)

Verwenden Sie AWS IoT SiteWise Edge-Gateways

AWS IoT SiteWise Edge erweitert die Cloud-Funktionen auf industrielle Edge-Umgebungen und ermöglicht so die lokale Datenverarbeitung, Analyse und Entscheidungsfindung. SiteWise Edge lässt sich in AWS IoT SiteWise und andere AWS Dienste integrieren, um umfassende industrielle IoT-Lösungen bereitzustellen. Gateways dienen als Vermittler zwischen Ihren Industrieanlagen und AWS IoT SiteWise.

SiteWise Edge-Gateways werden auf zwei verschiedenen Bereitstellungszielen ausgeführt:

- AWS IoT Greengrass V2
- Siemens Industrial Edge

Sie können ein SiteWise Edge-Gateway verwenden, um Daten am Edge zu sammeln und in der Cloud zu veröffentlichen. Bei Gateways, die darauf laufen AWS IoT Greengrass, können Sie Daten auch am Edge mithilfe von Asset-Modellen und Assets verarbeiten.

Die AWS IoT SiteWise Edge-Anwendung on Siemens Industrial Edge unterstützt die Integration zwischen Industrieanlagen, AWS IoT SiteWise sodass Sie Rohmaschinendaten aggregieren und verarbeiten und Analysen lokal ausführen können, bevor Sie verfeinerte Daten in die AWS Cloud senden.

Die wichtigsten Konzepte von SiteWise Edge-Gateways

SiteWise Edge bietet mehrere nützliche Funktionen für Edge-Computing in industriellen Umgebungen.

Lokale Datenerfassung und -verarbeitung

Unterstützt die Datenerfassung von Industrieanlagen mithilfe von Protokollen wie OPC-UA und MQTT. Gateways laufen auf Core-Geräten oder AWS IoT Greengrass Siemens Industrial Edge

Offline-Betrieb

Setzt die Erfassung und Verarbeitung von Daten bei Internetausfällen fort und synchronisiert sie mit der Cloud, wenn die Konnektivität wiederhergestellt ist.

Edge-Computing mit Komponenten AWS IoT Greengrass

Nutzt IoT SiteWise Publisher, um Daten für lokale Transformationen und Berechnungen an die Cloud und den AWS IoT SiteWise Prozessor weiterzuleiten. Sowohl der Herausgeber als auch der Prozessor sind AWS IoT Greengrass V2 Komponenten. Weitere Informationen zu AWS IoT Greengrass Komponenten finden Sie unter [AWS-provided components](#).

Integration mit AWS IoT SiteWise zur Erweiterung der Cloud-Funktionen

Funktioniert mit den AWS IoT SiteWise Cloud-Funktionen und dehnt Asset-Modelle, Analysen und Dashboards auf den Edge-Bereich aus.

Bei Gateways mit aktiviertem Datenverarbeitungspaket können Sie AWS OpsHub for verwenden, um Ihre SiteWise Edge-Gateways zentral AWS IoT SiteWise zu verwalten. AWS OpsHub bietet Funktionen zur Fernverwaltung und -überwachung. Weitere Informationen finden Sie unter [Verwalten Sie SiteWise Edge-Gateways mit AWS OpsHub AWS IoT SiteWise](#).

Integration von Partnerdatenquellen

Connect eine Partnerdatenquelle mit Ihrem Gateway und empfangen Sie Daten vom Partner in Ihrem SiteWise Edge-Gateway und der AWS Cloud. Weitere Informationen finden Sie unter [Partnerdatenquellen auf SiteWise Edge-Gateways](#).

Lokale Visualisierung am Edge

Bietet benutzerdefinierte Dashboards für Einblicke in Echtzeit am Netzwerkrand.

Überwachen Sie Daten lokal in Ihrer Einrichtung mithilfe von SiteWise Monitor-Portalen auf Ihren lokalen Geräten. Weitere Informationen finden Sie unter [Aktivierung Ihres AWS IoT SiteWise Portals am Edge](#).

Vorteile der Implementierung von SiteWise Edge

SiteWise Edge bietet zahlreiche Vorteile, die industrielle Abläufe und Entscheidungsprozesse erheblich verbessern können.

- Betriebserkenntnisse in Echtzeit ohne Verzögerungen bei der Cloud-Verarbeitung
- Betriebskontinuität in unverbundenen Umgebungen
- Geringere Bandbreiten- und Speicherkosten durch Edge-Vorverarbeitung
- Höhere Zuverlässigkeit durch die Möglichkeit, lokale, datengestützte Entscheidungen zu treffen

Self-host ein AWS IoT SiteWise Edge-Gateway mit AWS IoT Greengrass V2

Richten Sie AWS IoT SiteWise Edge so ein, dass Daten von Industrieanlagen lokal erfasst, verarbeitet und visualisiert werden, bevor sie an die Cloud gesendet werden. Self-host verwenden AWS IoT Greengrass Version 2.

Ein AWS IoT SiteWise Edge-Gateway fungiert als Vermittler zwischen Ihren Industrieanlagen und AWS IoT SiteWise. Das SiteWise Edge-Gateway läuft weiter AWS IoT Greengrass Version 2 und unterstützt die Datenerfassung und -verarbeitung vor Ort.

Es gibt zwei Arten von selbst gehosteten Gateways:

MQTT-enabled, V3-Gateway

Die MQTT-enabled V3-Gateway-Architektur bietet verbesserte Funktionen zur Datenaufnahme. Sie nutzt das MQTT-Protokoll für eine effiziente Datenkommunikation und bietet konfigurierbare Datenziele. Dazu gehören Optionen für die gepufferte Datenaufnahme mit Amazon S3 sowie für die Datenaufnahme in Echtzeit. Sie können Pfadfilter implementieren, um bestimmte MQTT-Themen zu abonnieren und so eine gezielte Datenerfassung zu ermöglichen. Beachten Sie MQTT-enabled, dass das V3-Gateway die Data Processing Pack-Funktion nicht unterstützt. Weitere Informationen finden Sie unter [MQTT-enabled, V3-Gateways für AWS IoT SiteWise Edge](#).

Classic Streams, V2-Gateway

Das Classic Streams, V2-Gateway stellt die traditionelle AWS IoT SiteWise Edge-Gateway-Architektur dar. Es eignet sich gut für bestehende SiteWise Edge-Bereitstellungen und für Benutzer, die an den etablierten Workflow gewöhnt sind. Während das Classic-Streaming, unterstützt das V2-Gateway das Datenverarbeitungspaket. Beachten Sie jedoch, dass die vom Datenverarbeitungspaket generierten Daten nicht über Amazon S3 aufgenommen werden können. Verwenden Sie das Classic Streams, V2-Gateway, wenn Sie die Kompatibilität mit bestehenden Bereitstellungen aufrechterhalten möchten oder wenn Sie die Funktionalität des Datenverarbeitungspakets benötigen. Weitere Informationen finden Sie unter [Klassische Streams, V2-Gateways für AWS IoT SiteWise Edge](#).

Topics

- [AWS IoT SiteWise Anforderungen an das selbst gehostete Edge-Gateway](#)
- [Erstellen Sie ein selbst gehostetes SiteWise Edge-Gateway](#)

- [Installieren Sie das AWS IoT SiteWise Edge-Gateway-Software auf Ihrem lokalen Gerät](#)
- [MQTT-enabled, V3-Gateways für AWS IoT SiteWise Edge](#)
- [Klassische Streams, V2-Gateways für AWS IoT SiteWise Edge](#)
- [Fügen Sie Datenquellen zu Ihrem hinzu AWS IoT SiteWise Edge-Gateway](#)
- [AWS IoT Greengrass Komponenten für AWS IoT SiteWise Edge](#)
- [Filtern Sie Ressourcen auf einem SiteWise Edge-Gateway](#)
- [Konfigurieren Sie die Proxyunterstützung und verwalten Sie Trust Stores für AWS IoT SiteWise Edge](#)
- [Verwenden Sie AWS IoT SiteWise APIs am Edge](#)

AWS IoT SiteWise Anforderungen an das selbst gehostete Edge-Gateway

AWS IoT SiteWise Edge-Gateways werden AWS IoT Greengrass V2 als eine Reihe von AWS IoT Greengrass Komponenten ausgeführt, die die Datenerfassung, -verarbeitung und -veröffentlichung vor Ort unterstützen. Um ein SiteWise Edge-Gateway zu konfigurieren, das auf läuft AWS IoT Greengrass V2, erstellen Sie ein Gateway in der AWS Cloud und führen Sie die SiteWise Edge-Gateway-Software aus, um Ihr lokales Gerät einzurichten. Wenn Sie die verwenden AWS-Managementkonsole, um das SiteWise Edge-Gateway zu erstellen, wird ein Installationsskript bereitgestellt. Führen Sie dieses Skript auf Ihrem Ziel-Gateway-Gerät aus, um die erforderliche Software und Abhängigkeiten einzurichten.

Lokale Geräteanforderungen

Lokale Geräte müssen die folgenden Anforderungen erfüllen, um die SiteWise Edge-Gateway-Software installieren und ausführen zu können.

- Unterstützt die AWS IoT Greengrass V2 Core-Softwareversion [v2.3.0 oder neuer](#). Weitere Informationen finden Sie unter [Anforderungen](#) im AWS IoT Greengrass Version 2 Entwicklerhandbuch.
- Eine der folgenden unterstützten Plattformen:
 - Betriebssystem: Ubuntu 20.04 oder höher
Architektur: x86_64 (AMD64) oder ARMv8 (Aarch64)
 - Betriebssystem: Red Hat Enterprise Linux (RHEL) 8
Architektur: x86_64 (AMD64) oder ARMv8 (Aarch64)

- Betriebssystem: Amazon Linux 2

Architektur: x86_64 (AMD64) oder ARMv8 (Aarch64)

- Betriebssystem: Debian 11

Architektur: x86_64 (AMD64) oder ARMv8 (Aarch64)

- Betriebssystem: Windows Server 2019 und später

Architektur: x86_64 (AMD64)

 Note

ARM-Plattformen unterstützen SiteWise Edge-Gateways nur mit Data Collection Pack. Das Datenverarbeitungspaket wird nicht unterstützt.

- Mindestens 4 GB RAM.
- Für die SiteWise Edge-Gateway-Software stehen mindestens 10 GB Festplattenspeicher zur Verfügung.
- Konfigurieren Sie Ihr lokales Gerät, um sicherzustellen, dass auf die richtigen Anschlüsse zugegriffen werden kann. Eine vollständige Liste der erforderlichen ausgehenden Dienstendpunkte finden Sie unter [Erforderliche Dienstendpunkte für AWS IoT SiteWise Edge-Gateways](#).
- Java Runtime Environment (JRE) Version 11 oder höher. Java muss in der PATH Umgebungsvariablen auf dem Gerät verfügbar sein. Um Java zur Entwicklung benutzerdefinierter Komponenten zu verwenden, müssen Sie ein Java Development Kit (JDK) installieren. [Wir empfehlen die Verwendung von Amazon Corretto oder OpenJDK.](#)

Amazon S3 S3-Buckets zur Zulassungsliste für lokale Geräte

Konfigurieren Sie Ihr lokales Gerät so, dass es Firewall-Zugriff auf den folgenden Amazon S3 S3-Bucket gewährt. Konfigurieren Sie den Zugriff auf der Grundlage der jeweiligen Regionen für Ihre Geräte.

Region	Endpoint
Asien-Pazifik (Tokio)	<code>https://iot-sitewise-gateway-ap-northeast-1-785558802005.s3.ap-northeast-1.amazonaws.com</code>

Region	Endpoint
Asien-Pazifik (Seoul)	https://iot-sitewise-gateway-ap-northeast-2-310055672453.s3.ap-northeast-2.amazonaws.com
Asien-Pazifik (Mumbai)	https://iot-sitewise-gateway-ap-south-1-677656657204.s3.ap-south-1.amazonaws.com
Asien-Pazifik (Singapur)	https://iot-sitewise-gateway-ap-southeast-1-475191558554.s3.ap-southeast-1.amazonaws.com
Asien-Pazifik (Sydney)	https://iot-sitewise-gateway-ap-southeast-2-396319432685.s3.ap-southeast-2.amazonaws.com
Kanada (Zentral)	https://iot-sitewise-gateway-ca-central-1-842060018567.s3.ca-central-1.amazonaws.com
China (Peking)	https://iot-sitewise-gateway-cn-north-1-237124890262.s3.cn-north-1.amazonaws.com.cn
Europe (Frankfurt)	https://iot-sitewise-gateway-eu-central-1-748875242063.s3.eu-central-1.amazonaws.com
Europa (Irland)	https://iot-sitewise-gateway-eu-west-1-383414315062.s3.eu-west-1.amazonaws.com
USA Ost (Nord-Virginia)	https://iot-sitewise-gateway-us-east-1-223558168232.s3.us-east-1.amazonaws.com und https://iot-sitewise-gateway-us-east-1-223558168232.s3.amazonaws.com/
USA Ost (Ohio)	https://iot-sitewise-gateway-us-east-2-005072661813.s3.us-east-2.amazonaws.com
AWS GovCloud (US-West)	https://iot-sitewise-gateway-us-gov-west-1-599984565679.s3.us-gov-west-1.amazonaws.com/
USA West (Oregon)	https://iot-sitewise-gateway-us-west-2-502577205460.s3.us-west-2.amazonaws.com

Anforderungen an das Datenverarbeitungspaket

Note

Das Feature Data Processing Pack (DPP) ist für Neukunden nicht mehr verfügbar. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter [Data processing pack availability change](#).

- Wenn Sie das Datenverarbeitungspaket am Edge mit verwenden möchten AWS IoT SiteWise, muss Ihr lokales Gerät außerdem die folgenden Anforderungen erfüllen:
 - Hat einen x86-64-Bit-Quad-Core-Prozessor.
 - Hat mindestens 16 GB RAM.
 - Hat mindestens 32 GB RAM, wenn Sie es verwenden Microsoft Windows.
 - Hatte mindestens 256 GB freien Festplattenspeicher.
 - Das lokale Gerät muss eingehenden Netzwerkverkehr auf Port 443 zulassen.
 - Die folgenden Ports sind für die Verwendung durch reserviert AWS IoT SiteWise: 80, 443, 3001, 4569, 4572, 8000, 8081, 8082, 8084, 8085, 8445, 8086, 9000, 9500, 11080 und 50010. Die Verwendung eines reservierten Ports für den Datenverkehr kann zu einem Verbindungsabbruch führen.

Note

Die AWS IoT Greengrass V2 Stream Manager-Komponente hat ihre eigenen Anforderungen. Weitere Informationen finden Sie unter [Konfiguration](#) im AWS IoT Greengrass Version 2 Entwicklerhandbuch.

- Die Mindestanforderungen an Festplattenspeicher und Rechenkapazität hängen von einer Vielzahl von Faktoren ab, die für Ihre Implementierung und Ihren Anwendungsfall spezifisch sind.
- Der für das Caching von Daten zur zeitweiligen Internetkonnektivität benötigte Festplattenspeicher ist von folgenden Faktoren abhängig:
 - Zahl der hochgeladenen Daten-Streams
 - Datenpunkte pro Daten-Stream pro Sekunde
 - Größe jedes Datenpunkts

- Erwartete Netzwerkausfallzeit
- Die zum Abfragen und Hochladen von Daten benötigte Rechenkapazität ist von folgenden Faktoren abhängig:
 - Zahl der hochgeladenen Daten-Streams
 - Datenpunkte pro Daten-Stream pro Sekunde

Konfigurieren Sie Berechtigungen für die Verwendung von SiteWise Edge-Gateways

Sie benötigen die folgenden Berechtigungen, um SiteWise Edge-Gateways verwenden zu können:

Note

Wenn Sie die AWS IoT SiteWise Konsole verwenden, um Ihr SiteWise Edge-Gateway zu erstellen, werden diese Berechtigungen für Sie hinzugefügt.

- Die IAM-Rolle für Ihr SiteWise Edge-Gateway muss es Ihnen ermöglichen, ein SiteWise Edge-Gateway auf einem AWS IoT Greengrass V2 Gerät zu verwenden, um Asset-Modelldaten und Asset-Daten zu verarbeiten.

Die Rolle ermöglicht es dem folgenden Dienst, die Rolle zu übernehmen:`credentials.iot.amazonaws.com`.

Details zu Berechtigungen

Die Rolle muss über die folgenden Berechtigungen verfügen:

- `iotsitewise`— Ermöglicht Prinzipalen das Abrufen von Asset-Modelldaten und Asset-Daten am Edge.
- `iot`— Ermöglicht Ihren AWS IoT Greengrass V2 Geräten die Interaktion mit AWS IoT.
- `logs`— Ermöglicht Ihren AWS IoT Greengrass V2 Geräten, Protokolle an Amazon CloudWatch Logs zu senden.
- `s3`— Ermöglicht Ihren AWS IoT Greengrass V2 Geräten, benutzerdefinierte Komponentenartefakte von Amazon S3 herunterzuladen.

JSON

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "iotsitewise:BatchPutAssetPropertyValue",
      "iotsitewise:List*",
      "iotsitewise:Describe*",
      "iotsitewise:Get*"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iot:DescribeCertificate",
      "logs:CreateLogGroup",
      "logs:CreateLogStream",
      "logs:PutLogEvents",
      "logs:DescribeLogStreams",
      "s3:GetBucketLocation",
      "s3:GetObject",
      "iot:Connect",
      "iot:Publish",
      "iot:Subscribe",
      "iot:Receive",
      "iot:DescribeEndpoint"
    ],
    "Resource": "*"
  }
]
}

```

Erstellen Sie ein selbst gehostetes SiteWise Edge-Gateway

Verwenden Sie die AWS IoT SiteWise Konsole oder AWS CLI um ein selbst gehostetes SiteWise Edge-Gateway zu erstellen. In diesem Verfahren wird beschrieben, wie Sie ein selbst gehostetes SiteWise Edge-Gateway erstellen, das Sie auf Ihrer eigenen Hardware installieren. Informationen zum Erstellen eines SiteWise Edge-Gateways, das auf Siemens Industrial Edge läuft, finden Sie unter [Hosten Sie ein SiteWise Edge-Gateway auf Siemens Industrial Edge](#).

Erstellen Sie ein SiteWise Edge-Gateway

Console

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie Create gateway (Gateway erstellen).
4. Wählen Sie für Bereitstellungsziel auswählen die Option Self-Hosted Gateway aus.
5. Wählen Sie entweder MQTT-enabledV3-Gateway oder Classic Streams, V2-Gateway aus. Weitere Informationen zu den einzelnen Optionen finden Sie unter [Self-host einAWS IoT SiteWiseEdge-Gateway mitAWS IoT Greengrass V2](#). Das MQTT-enabled V3-Gateway wird aufgrund seiner zukunftsfähigen Funktionen empfohlen.
6. Geben Sie im Abschnitt Gateway-Konfiguration einen Namen für Ihr SiteWise Edge-Gateway ein, oder verwenden Sie den von AWS IoT SiteWise generierten Namen.
7. Wählen Sie unter Greengrass-Gerätebetriebssystem das Betriebssystem des Geräts aus, auf dem Sie dieses SiteWise Edge-Gateway installieren möchten.

Note

Das Datenverarbeitungspaket ist nur auf x86-Plattformen verfügbar. Es ist nur auf den Classic-Streams, V2-Gateway verfügbar

8. (Optional) Um Daten am Edge zu verarbeiten und zu organisieren, wählen Sie unter Edge-Funktionen die Option Data Processing Pack aus.

Note

Informationen dazu, wie Sie Benutzergruppen in Ihrem Unternehmensverzeichnis Zugriff auf dieses SiteWise Edge-Gateway gewähren, finden Sie unter [Richten Sie die Edge-Funktion in SiteWise Edge ein](#)

9. (Optional) Gehen Sie unter „Erweiterte Konfiguration“ wie folgt vor:
 - Wählen Sie für das Greengrass-Core-Gerät eine der folgenden Optionen:
 - Standard-Setup — verwendet AWS automatisch die Standardeinstellungen, um ein Greengrass-Core-Gerät in AWS IoT Greengrass V2 zu erstellen.

1. Geben Sie einen Namen für das Greengrass-Core-Gerät ein oder verwenden Sie den von AWS IoT SiteWise generierten Namen.
 - Erweiterte Einrichtung — Wählen Sie diese Option, wenn Sie ein vorhandenes Greengrass-Core-Gerät verwenden oder eines manuell erstellen möchten.
 1. Wählen Sie ein Greengrass-Core-Gerät oder wählen Sie Create Greengrass Core-Gerät, um eines in der AWS IoT Greengrass V2 Konsole zu erstellen. Weitere Informationen finden Sie unter [Einrichten von AWS IoT Greengrass V2-Core-Geräten](#) im AWS IoT Greengrass Version 2-Entwicklerhandbuch.
10. Wählen Sie Create gateway (Gateway erstellen).
 11. Wählen Sie im Dialogfeld „SiteWise Edge-Gateway-Installationsprogramm generieren“ die Option Generieren und herunterladen aus. AWS IoT SiteWise generiert automatisch ein Installationsprogramm, mit dem Sie Ihr lokales Gerät konfigurieren können.

Important

Sie können diese Datei nicht regenerieren. Stellen Sie sicher, dass Sie die Installationsdatei an einem sicheren Ort speichern, da Sie sie später verwenden werden.

AWS CLI

Um ein selbst gehostetes Gateway mithilfe von zu erstellen AWS CLI, geben Sie einen Namen für das Gateway ein, geben Sie die Plattform und die Gateway-Version an. Es gibt viele andere Optionen, die Sie bei der Erstellung eines Gateways angeben können. Weitere Informationen finden Sie unter [create-gateway](#) in der AWS CLIBefehlsreferenz für AWS IoT SiteWise

Wenn Sie dieses Beispiel verwenden möchten, ersetzen Sie die Platzhalter für Benutzereingaben durch Ihre Informationen.

```
aws iotsitewise create-gateway \
  --gateway-name your-gateway-name \
  --gateway-platform greengrassV2={coreDeviceThingName=your-core-device-thing-name, coreDeviceOperatingSystem=LINUX_AMD64} \
  --gateway-version 3 \
  [--cli-input-json your-configuration]
```

- `gateway-name`— Ein eindeutiger Name für das Gateway.
- `gateway-platform`— Gibt die Gateway-Plattformkonfiguration an. Geben Sie für selbst gehostete Gateways ein. `greengrassV2` Weitere Informationen finden Sie unter [Optionen](#) im Abschnitt `Create-Gateway` der Befehlsreferenz für AWS CLI. AWS IoT SiteWise
- `coreDeviceThingName`— Der Name des AWS IoT für Ihr AWS IoT Greengrass V2 Kerngerät.
- `coreDeviceOperatingSystem`— Das Betriebssystem des Kerngeräts in AWS IoT Greengrass V2. Die Angabe des Betriebssystems ist für erforderlich `gateway-version 3` und gilt nicht für `gateway-version 2`. Zu den Optionen gehören: `LINUX_AARCH64`, `LINUX_AMD64`, und `WINDOWS_AMD64`
- `gateway-version`— Die Version des Gateways.
 - Um ein MQTT-enabled V3-Gateway zu erstellen, verwenden Sie 3 für die Gateway-Version.
 - Um ein Classic-Streams-V2-Gateway zu erstellen, verwenden Sie 2 für die Gateway-Version.
- `cli-input-json`— Eine JSON-Datei mit Anforderungsparametern.

Nachdem Sie das SiteWise Edge-Gateway erstellt haben, [Installieren Sie das AWS IoT SiteWise Edge-Gateway-Software auf Ihrem lokalen Gerät](#).

Installieren Sie das AWS IoT SiteWise Edge-Gateway-Software auf Ihrem lokalen Gerät

Nachdem Sie ein AWS IoT SiteWise Edge-Gateway erstellt haben, installieren Sie die SiteWise Edge-Gateway-Software auf Ihrem lokalen Gerät. Die SiteWise Edge-Gateway-Software kann auf lokalen Geräten installiert werden, auf denen Linux- oder Microsoft Windows Serverbetriebssysteme installiert sind.

Important

Stellen Sie sicher, dass Ihr lokales Gerät eine Verbindung zum Internet herstellt.

Linux

Das folgende Verfahren verwendet SSH, um eine Verbindung zu Ihrem lokalen Gerät herzustellen. Alternativ können Sie ein USB-Flash-Laufwerk oder andere Tools verwenden,

um die Installationsdatei auf Ihr lokales Gerät zu übertragen. Wenn Sie SSH nicht verwenden möchten, fahren Sie mit Schritt 2: Installieren Sie die SiteWise Edge-Gateway-Software unten fort.

SSH-Voraussetzungen

Bevor Sie über SSH eine Verbindung zu Ihrem Gerät herstellen, müssen Sie die folgenden Voraussetzungen erfüllen.

- Linux und macOS — Laden Sie OpenSSH herunter und installieren Sie es. Weitere Informationen finden Sie unter <https://www.openssh.com>.

Schritt 1: Kopieren Sie das Installationsprogramm auf Ihr SiteWise Edge-Gateway-Gerät

In den folgenden Anweisungen wird erklärt, wie Sie mithilfe eines SSH-Clients eine Verbindung zu Ihrem lokalen Gerät herstellen.

1. Um eine Verbindung zu Ihrem Gerät herzustellen, führen Sie den folgenden Befehl in einem Terminalfenster auf Ihrem Computer aus *username* und ersetzen Sie ihn durch einen Benutzernamen *IP* mit erhöhten Rechten und einer IP-Adresse.

```
ssh username@IP
```

2. Führen Sie den folgenden Befehl aus, um die AWS IoT SiteWise generierte Installationsdatei auf Ihr SiteWise Edge-Gateway-Gerät zu übertragen.

Note

- *path-to-saved-installer* Ersetzen Sie es durch den Pfad auf Ihrem Computer, den Sie zum Speichern der Installationsdatei verwendet haben, und durch den Namen der Installationsdatei.
- *IP-address* Ersetzen Sie es durch die IP-Adresse Ihres lokalen Geräts.
- *directory-to-receive-installer* Ersetzen Sie es durch den Pfad auf Ihrem lokalen Gerät, den Sie für den Empfang der Installationsdatei verwenden.

```
scp path-to-saved-installer.sh user-name@IP-address:directory-to-receive-installer
```

Schritt 2: Installieren Sie die SiteWise Edge-Gateway-Software

Führen Sie in den folgenden Verfahren die Befehle in einem Terminalfenster auf Ihrem SiteWise Edge-Gateway-Gerät aus.

1. Erteilen Sie der Installationsdatei die Ausführungsberechtigung.

```
chmod +x path-to-installer.sh
```

2. Führen Sie das Installationsprogramm aus.

```
sudo ./path-to-installer.sh
```

Windows Server

Voraussetzungen

Sie müssen die folgenden Voraussetzungen erfüllen, um die SiteWise Edge-Gateway-Software zu installieren:

- Microsoft WindowsServer 2019 oder höher installiert
- Administratorrechte
- PowerShell Version 5.1 oder höher installiert
- SiteWise Das Edge-Gateway-Installationsprogramm wurde auf den Windows-Server heruntergeladen, wo es bereitgestellt wird

Schritt 1: PowerShell Als Administrator ausführen

1. Melden Sie sich auf dem Windows-Server, auf dem Sie das SiteWise Edge-Gateway installieren möchten, als Administrator an.
2. Geben Sie PowerShell in die Windows-Suchleiste ein.
3. Öffnen Sie in den Suchergebnissen das Kontextmenü (Rechtsklick) in der PowerShell Windows-App. Wählen Sie Als Administrator ausführen aus.

Schritt 2: Installieren Sie die SiteWise Edge-Gateway-Software

Führen Sie die folgenden Befehle in einem Terminalfenster auf Ihrem SiteWise Edge Gateway-Gerät aus.

1. Entsperren Sie das SiteWise Edge-Gateway-Installationsprogramm.

```
unblock-file path-to-installer.ps1
```

2. Führen Sie das Installationsprogramm aus.

```
./path-to-installer.ps1
```

 Note

Wenn die Skriptausführung auf dem System deaktiviert ist, ändern Sie die Richtlinie zur Skriptausführung in `RemoteSigned`.

```
Set-ExecutionPolicy RemoteSigned
```

Der nächste Schritt hängt von der Art des selbst gehosteten Gateways ab, das Sie benötigen. Fahren Sie fort mit [MQTT-enabled, V3-Gateways fürAWS IoT SiteWiseEdge](#) oder [Klassische Streams, V2-Gateways fürAWS IoT SiteWiseEdge](#).

MQTT-enabled, V3-Gateways fürAWS IoT SiteWiseEdge

AWS IoT SiteWise kann V3-Gateways verwenden MQTT-enabled, was eine bedeutende Weiterentwicklung der SiteWise Edge-Gateway-Architektur darstellt. Dieser Gateway-Typ nutzt das MQTT-Protokoll (Message Queuing Telemetry Transport) für die Datenkommunikation und bietet so mehr Flexibilität und Effizienz in industriellen IoT-Implementierungen.

Das MQTT-enabled V3-Gateway verwendet MQTT für die Datenübertragung und ermöglicht so ein schlankes Publish-Subscribe-Netzwerkprotokoll, das Nachrichten effizient zwischen Geräten und der Cloud transportiert. Sie können verschiedene Datenziele einrichten, darunter die direkte Datenaufnahme in Echtzeit AWS IoT SiteWise und die gepufferte Datenaufnahme mit Amazon S3. Um eine präzise Datenerfassung zu ermöglichen, können Sie Pfadfilter implementieren, um bestimmte MQTT-Themen zu abonnieren.

MQTT-enabled, V3-Gateways verfügen über ein vorkonfiguriertes Echtzeitziel mit Filtern, die auf „#“ eingestellt sind (alle Themen), die Sie nach Bedarf anpassen oder entfernen können. Um die Datenverwaltung zu optimieren, kann in jedem Gateway nur ein Echtzeitziel vorhanden sein.

Die MQTT-enabled Architektur unterscheidet sich erheblich vom Classic Streams, V2-Gateway. Während V2 einen stream-basierten Ansatz verwendet, verwendet V3 MQTT und bietet mehr konfigurierbare Datenziele und Filteroptionen. Beachten Sie jedoch, dass V3 das Datenverarbeitungspaket, das in V2 verfügbar ist, nicht unterstützt.

Das MQTT-enabled V3-Gateway bietet mehrere Vorteile:

- Verbesserte Skalierbarkeit aufgrund der geringen Größe von MQTT, die eine bessere Handhabung zahlreicher Geräte und eine hochfrequente Datenübertragung ermöglicht.
- Verbesserte Datenkontrolle durch Pfadfilter, die eine detaillierte Verwaltung der Datenerfassung ermöglichen und unnötige Datenübertragungen und -verarbeitung reduzieren.
- Flexible Datenverarbeitung, die je nach Bedarf eine Konfiguration zwischen Echtzeitverarbeitung und gepuffertem Speicher ermöglicht.
- Anpassung an moderne IoT-Kommunikationsstandards, wodurch die Voraussetzungen für future Verbesserungen und Integrationen geschaffen werden.

Erwägen Sie die Einführung des MQTT-enabled V3-Gateways für neue Implementierungen, insbesondere wenn Sie flexible Datenaufnahmeoptionen und eine präzise Kontrolle über die Datenerfassung benötigen.

Note

Für bestehende Bereitstellungen oder Szenarien, die das Datenverarbeitungspaket erfordern, bleibt das Classic Streams, V2-Gateway eine praktikable Option.

Durch das Angebot beider Gateway-Typen AWS IoT SiteWise wird sichergestellt, dass Sie die Lösung wählen können, die Ihren spezifischen industriellen IoT-Anforderungen am besten entspricht, unabhängig davon, ob Sie erweiterte MQTT-Funktionen oder Kompatibilität mit bestehenden Systemen bevorzugen.

Ziele und Pfadfilter

In den folgenden Themen erfahren Sie mehr über Ziele und Pfadfilter in MQTT-enabled Gateways:

- [Verstehen Sie AWS IoT SiteWise Edge-Ziele](#)
- [Fügen Sie ein AWS IoT SiteWise Edge-Echtzeitziel hinzu](#)

- [Fügen Sie ein AWS IoT SiteWise gepuffertes Ziel mit Amazon S3 hinzu](#)
- [Machen Sie sich mit Pfadfiltern für AWS IoT SiteWise Edge-Ziele vertraut](#)
- [Fügen Sie Pfadfilter zu AWS IoT SiteWise Edge-Zielen hinzu](#)
- [AWS IoT SiteWise Edge-Ziele verwalten](#)

Connect externe Anwendungen mit dem EMQX-Broker

In diesem Handbuch wird erläutert, wie Sie externe Anwendungen über einen EMQX-Broker auf Ihrem bereitgestellten MQTT-enabled V3-Gateway mit Ihrem AWS IoT SiteWise Edge-Gateway verbinden. Zu den externen Anwendungen können benutzerdefinierte Überwachungstools, Visualisierungssoftware von Drittanbietern oder ältere Systeme gehören, die mit Ihren Industriedaten am Netzwerkrand interagieren müssen.

Wir behandeln die Konfigurationsschritte sowohl für Linux als auch für Microsoft Windows Umgebungen, einschließlich der Konfiguration der EMQX-Bereitstellung, der TLS-Einrichtung für sichere Verbindungen und Autorisierungsregeln zur Zugriffskontrolle für bestimmte Themen.

Note

EMQX ist kein Anbieter oder Lieferant für Edge. AWS IoT SiteWise

Important

Um Verbindungen zu Ihrem Gateway zu sichern, empfehlen wir dringend, die zertifikatsbasierte Authentifizierung über die Authentifizierungsfunktion für AWS IoT Greengrass Client-Geräte zu verwenden. Diese Methode bietet zuverlässige Sicherheit durch gegenseitige TLS-Authentifizierung (mTLS). Weitere Informationen finden Sie im AWS IoT Greengrass Version 2 Entwicklerhandbuch unter [Connect von Client-Geräten mit Kerngeräten](#).

Wenn Sie die zertifikatsbasierte Authentifizierung nicht verwenden können, folgen Sie dieser Anleitung, um die Authentifizierung mit Benutzernamen und Passwörtern einzurichten.

Voraussetzungen

- Ein SiteWise Edge MQTT-enabled, V3-Gateway, das bereitgestellt wurde und online ist

- Zugriff auf den Gateway-Host
- Zugriff auf die AWS IoT Greengrass Konsolen AWS IoT SiteWise und

Themen

- [Nachrichten-Payload-Format für den EMQX-Broker auf AWS IoT SiteWise Edge](#)
- [Konfigurieren Sie den EMQX-Broker](#)
- [Connect eine Anwendung mit dem EMQX-Broker auf AWS IoT SiteWise Edge](#)
- [Richten Sie Autorisierungsregeln ein für AWS IoT SiteWise Edge in EMQX](#)

Nachrichten-Payload-Format für den EMQX-Broker auf AWS IoT SiteWise Edge

Damit die SiteWise IoT-Publisher-Komponente Daten aus Ihrer externen Anwendung verarbeiten und in der AWS IoT SiteWise Cloud veröffentlichen kann, muss die an den Broker gesendete Nutzlast bestimmte Anforderungen erfüllen.

Das Verständnis des Payload-Formats ist der Schlüssel zu einer erfolgreichen MQTT-Kommunikation mit Edge. AWS IoT SiteWise Der Verbindungsaufbau wird zwar in späteren Abschnitten behandelt, wir stellen Ihnen jedoch zunächst die Payload-Anforderungen vor, um Ihnen bei der Planung Ihrer Implementierung zu helfen.

Anforderungen an das MQTT-Thema

Es gibt keine Einschränkungen hinsichtlich der Struktur von MQTT-Themen, einschließlich der Anzahl der verwendeten Ebenen oder Zeichen. Wir empfehlen jedoch, dass das Thema mit dem `propertyAlias` Feld in der Payload übereinstimmt.

Example Beispiel für einen Eigenschaftsalias

Wenn das MQTT-Thema lautet `site1/line1/compressor1/temperature`, stellen Sie sicher, dass die `propertyAlias` Treffer übereinstimmen.

```
{
  "assetId": "compressor_asset_01",
  "propertyAlias": "site1/line1/compressor1/temperature",
  "propertyId": "temperature_sensor_01",
  "propertyValues": [
    {
      "quality": "GOOD",
      "timestamp": {
```

```
        "offsetInNanos": 0,
        "timeInSeconds": 1683000000
    },
    "value": {
        "doubleValue": 23.5
    }
}
]
```

Struktur der JSON-Nutzlast

Die MQTT-Nachrichtennutzdaten sind in JSON geschrieben und folgen dem in der [PutAssetPropertyValueEntry API-Referenz](#) definierten Nachrichtenformat. [AWS IoT SiteWise](#)

```
{
  "assetId": "string",
  "propertyAlias": "string",
  "propertyId": "string",
  "propertyValues": [
    {
      "quality": "string",
      "timestamp": {
        "offsetInNanos": number,
        "timeInSeconds": number
      },
      "value": {
        "booleanValue": boolean,
        "doubleValue": number,
        "integerValue": number,
        "stringValue": "string"
      }
    }
  ]
}
```

Note

Damit eine Nachricht als gültig angesehen werden kann, kann nur eine der folgenden Bedingungen erfüllt sein:

- Das `propertyAlias` ist gesetzt, oder

- `assetId` sowohl als auch `propertyId` sind gesetzt

Das `PutAssetPropertyValueEntry` hat ein `entryId` Feld, das in diesem Zusammenhang nicht erforderlich ist.

Konfigurieren Sie den EMQX-Broker

In diesem Abschnitt wird beschrieben, wie Sie Benutzernamen und Passwörter hinzufügen. Außerdem wird beschrieben, wie Sie mithilfe des hinzugefügten Benutzernamens und Kennworts eine TLS-Verbindung von einer externen Quelle aus herstellen. Sie können den EMQX-Broker unter Linux oder konfigurieren. Microsoft Windows

Note

Um den Broker zu konfigurieren, benötigen Sie ein Core-Gerät, das mit der EMQX-Standardkonfiguration in Ihrem MQTT-enabled V3-Gateway eingerichtet ist.

Important

Nach Abschluss dieses Verfahrens empfehlen wir dringend, Autorisierungsregeln zu konfigurieren. Weitere Informationen finden Sie unter [Richten Sie Autorisierungsregeln ein für AWS IoT SiteWise Edge in EMQX](#). Autorisierungsregeln für hinzugefügte Benutzer erhöhen die Sicherheit.

Aktualisieren Sie die EMQX-Bereitstellungskonfiguration für die Authentifizierung

Um die EMQX-Bereitstellungskonfiguration für die Authentifizierung zu aktualisieren

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie in der linken Navigationsleiste im Abschnitt Edge die Option Edge-Gateways aus.
3. Wählen Sie das zu konfigurierende Gateway aus.
4. Kopieren Sie im Abschnitt Edge-Gateway-Konfiguration Ihren Greengrass-Core-Gerätewert. Speichern Sie ihn für die spätere Verwendung.
5. Öffnen Sie die [AWS IoT-Konsole](#).

6. Wählen Sie in der linken Navigationsleiste im Bereich Verwalten die Option Greengrass-Geräte und dann Bereitstellungen aus.
7. Suchen Sie den Core-Gerätewert, den Sie zuvor gespeichert haben, und wählen Sie diesen Link, um die Bereitstellung zu öffnen.
8. Wählen Sie die Dropdownschaltfläche „Aktionen“ und anschließend „Überarbeiten“.
9. Lesen Sie die angezeigte Meldung und wählen Sie dann Bereitstellung überarbeiten aus. Die Seite „Ziel angeben“ wird angezeigt.
10. Wählen Sie Weiter, bis Sie zum Schritt Komponenten konfigurieren gelangen.
11. Wählen Sie das Optionsfeld `aws.greengrass.clientdevices.mqtt.EMQX`.
12. Wählen Sie die Schaltfläche Komponente konfigurieren. Eine Konfigurationsseite für die Komponente wird angezeigt.
13. Wählen Sie unter Konfigurationsupdate die Option Auf Standardkonfiguration zurücksetzen für die Komponentenversion: 2.* aus. *.
14. Geben Sie je nach Betriebssystem im Abschnitt Konfiguration zum Zusammenführen die folgende Konfiguration ein.

Linux

```
{
  "emqxConfig": {
    "authorization": {
      "no_match": "allow"
    },
    "listeners": {
      "tcp": {
        "default": {
          "enabled": true,
          "enable_authn": false
        }
      },
      "ssl": {
        "default": {
          "enabled": true,
          "enable_authn": true,
          "ssl_options": {
            "verify": "verify_none",
            "fail_if_no_peer_cert": false
          }
        }
      }
    }
  }
}
```

```

    }
  },
  "authentication": {
    "enable": true,
    "backend": "built_in_database",
    "mechanism": "password_based",
    "password_hash_algorithm": {
      "iterations": 210000,
      "mac_fun": "sha512",
      "name": "pbkdf2"
    },
    "user_id_type": "username"
  },
  "dashboard": {
    "listeners": {
      "http": {
        "bind": 18083
      }
    }
  },
  "authMode": "bypass",
  "dockerOptions": "-p 8883:8883 -p 127.0.0.1:1883:1883\n-p 127.0.0.1:18083:18083 -v emqx-data:/opt/emqx/data -e\nEMQX_NODE__NAME=emqx@local",
  "requiresPrivilege": "true"
}

```

Windows

```

{
  "emqxConfig": {
    "authorization": {
      "no_match": "allow"
    },
    "listeners": {
      "tcp": {
        "default": {
          "enabled": true,
          "enable_authn": false
        }
      },
      "ssl": {

```

```

        "default": {
            "enabled": true,
            "enable_authn": true,
            "ssl_options": {
                "verify": "verify_none",
                "fail_if_no_peer_cert": false
            }
        },
        "authentication": {
            "enable": true,
            "backend": "built_in_database",
            "mechanism": "password_based",
            "password_hash_algorithm": {
                "iterations": 210000,
                "mac_fun": "sha512",
                "name": "pbkdf2"
            },
            "user_id_type": "username"
        },
        "dashboard": {
            "listeners": {
                "http": {
                    "bind": 18083
                }
            }
        },
        "authMode": "bypass",
        "requiresPrivilege": "true"
    }
}

```

Das `dockerOptions` Feld ist nur für Linux-Gateways bestimmt.

15. Wählen Sie Bestätigen aus.
16. Wählen Sie Weiter, bis Sie zum Schritt Überprüfen gelangen.
17. Wählen Sie Bereitstellen.
18. Wenn die Bereitstellung erfolgreich war, fahren Sie mit dem nächsten Schritt fort.

Aktivieren Sie die Authentifizierung mit Benutzername und Passwort

In diesem Abschnitt erfahren Sie, wie Sie Benutzernamen und Passwörter über die EMQX-Dashboard-GUI hinzufügen.

Note

Die bereitgestellten EMQX-related Anweisungen dienen nur als Referenz. Da sich die Dokumentation und Funktionen von EMQX im Laufe der Zeit ändern können und wir ihre Dokumentation nicht pflegen, empfehlen wir, die [offizielle Dokumentation von EMQX](#) für die aktuellsten Informationen zu konsultieren.

EMQX Dashboard

Um die Authentifizierung mit Benutzername und Passwort über das EMQX-Dashboard zu aktivieren

1. Stellen Sie sicher, dass Sie sich innerhalb des Gateway-Hosts befinden.
2. Öffnen Sie ein Browserfenster und besuchen Sie <http://localhost:18083/>.
3. Geben Sie den Standardbenutzernamen von **admin** und das Standardkennwort von **public**. Weitere Informationen finden Sie unter [EMQX Dashboard](#) in den EMQX-Dokumenten.
4. Nach der Anmeldung werden Sie aufgefordert, Ihr Passwort zu ändern. Aktualisieren Sie Ihr Passwort, um zum EMQX Dashboard zu gelangen.
5. Wählen Sie in der linken Navigationsleiste das Schildsymbol und dann Authentifizierung.
6. Wählen Sie in der Built-in Datenbankzeile die Schaltfläche Benutzer aus.
7. Wählen Sie die Schaltfläche mit dem Pluszeichen, um Benutzer hinzuzufügen. Ein Bildschirm „Hinzufügen“ wird angezeigt.
8. Geben Sie einen Benutzernamen und ein Passwort für den Benutzer der externen Anwendung ein.
9. Wählen Sie Speichern. Der von Ihnen gewählte Benutzername wird in der Tabelle der Authentifizierungsseite angezeigt.

 Note

Für den neuen Benutzer gelten bestehende oder standardmäßige Autorisierungsregeln. Es wird empfohlen, sie zu überprüfen und an die Anforderungen Ihrer externen Anwendung anzupassen.

EMQX Management with Linux

Verwenden Sie das AWS IoT SiteWise EMQX CLI-Tool unter. `/greengrass/v2/bin/swe-emqx-cli`

Um die Authentifizierung mit Benutzername und Passwort über EMQX Management unter Linux zu aktivieren

1. Ändern Sie das Admin-Passwort, indem Sie den folgenden Befehl ausführen:

```
/greengrass/v2/bin/swe-emqx-cli admin change-pwd
```

2. Gehen Sie wie folgt vor, wenn Sie dazu aufgefordert werden:
 1. Geben Sie Ihren aktuellen Administratorbenutzer (Standard ist `admin`) und Ihr Passwort (Standard ist `public`) ein.
 2. Geben Sie Ihr neues Passwort ein und bestätigen Sie es.

Bei Erfolg wird die folgende Meldung angezeigt:

```
admin password changed successfully
```

3. Fügen Sie Benutzer für externe Anwendungen hinzu, indem Sie den folgenden Befehl ausführen:

```
/greengrass/v2/bin/swe-emqx-cli users add
```

4. Gehen Sie wie folgt vor, wenn Sie dazu aufgefordert werden:
 1. Geben Sie den Benutzernamen für den neuen Benutzer ein.
 2. Geben Sie das Passwort für den neuen Benutzer ein und bestätigen Sie es.

Bei Erfolg wird die folgende Meldung angezeigt:

```
User '[username]' created successfully
```

5. Überprüfen Sie die Benutzerkonfiguration, indem Sie den folgenden Befehl ausführen:

```
/greengrass/v2/bin/swe-emqx-cli users list
```

Die Ausgabe zeigt alle konfigurierten Benutzer:

```
Users:
- [your-added-username]

Total users: 1
```

EMQX Management with Windows

Verwenden Sie das AWS IoT SiteWise EMQX CLI Tool an einem der folgenden Orte:

- PowerShell: C:\greengrass\v2\bin\swe-emqx-cli.ps1
- Eingabeaufforderung: C:\greengrass\v2\bin\swe-emqx-cli.bat

Um die Authentifizierung mit Benutzername und Passwort über EMQX Management unter Windows zu aktivieren

1. Ändern Sie das Admin-Passwort, indem Sie den folgenden Befehl ausführen:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 admin change-pwd
```

2. Gehen Sie wie folgt vor, wenn Sie dazu aufgefordert werden:
 1. Geben Sie Ihren aktuellen Administratorbenutzer (Standard ist `admin`) und Ihr Passwort (Standard ist `public`) ein.
 2. Geben Sie Ihr neues Passwort ein und bestätigen Sie es.

Bei Erfolg wird die folgende Meldung angezeigt:

```
admin password changed successfully
```

3. Fügen Sie Benutzer für externe Anwendungen hinzu, indem Sie den folgenden Befehl ausführen:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 users add
```

4. Gehen Sie wie folgt vor, wenn Sie dazu aufgefordert werden:
 1. Geben Sie den Benutzernamen für den neuen Benutzer ein.
 2. Geben Sie das Passwort für den neuen Benutzer ein und bestätigen Sie es.

Bei Erfolg wird die folgende Meldung angezeigt:

```
User '[username]' created successfully
```

5. Überprüfen Sie die Benutzerkonfiguration, indem Sie den folgenden Befehl ausführen:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 users list
```

Die Ausgabe zeigt alle konfigurierten Benutzer:

```
Users:
- [your-added-username]

Total users: 1
```

Connect eine Anwendung mit dem EMQX-Broker auf AWS IoT SiteWise Edge

Der EMQX-Broker verwendet Transport Layer Security (TLS) auf Port 8883, um die gesamte Kommunikation zu verschlüsseln und so sicherzustellen, dass Ihre Daten während der Übertragung geschützt bleiben. In diesem Abschnitt werden Sie durch die Schritte zum Herstellen von Verbindungen zwischen Ihren Anwendungen und dem EMQX-Broker geführt. Die Einhaltung dieser Schritte trägt dazu bei, die Integrität und Vertraulichkeit Ihrer Industriedaten zu wahren. Der Verbindungsprozess umfasst zwei Hauptansätze: die automatische IP-Erkennung mithilfe von Komponenten oder die manuelle Konfiguration von DNS-Namen und IP-Adressen als Subject Alternative Names (SANs) in Ihren TLS-Zertifikaten. Jede Methode hat ihre eigenen Vorteile,

abhängig von Ihrer Netzwerkkonfiguration und Ihren Sicherheitsanforderungen. Diese Dokumentation führt Sie durch beide Optionen.

Themen

- [Konfigurieren Sie TLS für sichere Verbindungen zum EMQX-Broker auf AWS IoT SiteWise Edge](#)
- [Testen Sie die EMQX-Brokerverbindung auf AWS IoT SiteWise Edge](#)
- [Verwenden Sie Ihre eigene CA](#)
- [Öffnen Sie Port 8883 für externe Firewallverbindungen](#)

Konfigurieren Sie TLS für sichere Verbindungen zum EMQX-Broker auf AWS IoT SiteWise Edge

AWS IoT Greengrass generiert standardmäßig ein TLS-Serverzertifikat für den EMQX-Broker, das von der Core Device Certificate Authority (CA) signiert ist. Weitere Informationen finden Sie unter [Client-Geräte über einen MQTT-Broker mit einem AWS IoT Greengrass Core-Gerät verbinden](#).

Rufen Sie das TLS-Zertifikat ab

Um das CA-Zertifikat zu erhalten, führen Sie den folgenden Befehl auf dem Gateway-Host aus:

Linux

Führen Sie den folgenden Befehl in einer Shell-Sitzung auf dem Gateway-Host aus:

```
/greengrass/v2/bin/swe-emqx-cli cert
```

Mit diesem Befehl wird der Speicherort des Zertifikats angezeigt und der Inhalt des Zertifikats gedruckt.

Sie können das Zertifikat alternativ mit diesem Befehl in einer Datei speichern:

```
/greengrass/v2/bin/swe-emqx-cli cert --output /path/to/certificate.pem
```

Windows

Führen Sie den folgenden Befehl in einer PowerShell Sitzung auf dem Gateway-Host aus:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 cert
```

Mit diesem Befehl wird der Speicherort des Zertifikats angezeigt und der Inhalt des Zertifikats gedruckt.

Sie können das Zertifikat alternativ mit diesem Befehl in einer Datei speichern:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 cert --output C:\path\to\certificate.pem
```

Die CLI findet das Zertifikat automatisch, unabhängig vom genauen Pfad auf Ihrem System.

Kopieren Sie den Inhalt der Datei `ca.pem` in die externe Anwendung, die Sie mit dem Broker verbinden. Speichern Sie diese unter `BrokerCoreDeviceCA.pem`.

Fügen Sie dem TLS-Serverzertifikat benutzerdefinierte names/IP DNS-Adressen hinzu

Der alternative Name des Antragstellers (SAN) auf dem von AWS IoT Greengrass `localhost` generierten Zertifikat. Beim Herstellen einer TLS-Verbindung von außerhalb des Gateway-Hosts schlägt der TLS-Überprüfungsschritt fehl, da der Hostname des Brokers nicht mit dem Hostnamen `localhost` auf dem Serverzertifikat übereinstimmt.

Um das Problem mit nicht übereinstimmenden Hostnamen zu beheben, AWS IoT Greengrass bietet diese Option zwei Möglichkeiten zur Verwaltung von Endpunkten auf Kerngeräten. In diesem Abschnitt werden beide Optionen behandelt. Ausführlichere Informationen finden Sie unter [Manage core device endpoints](#) im AWS IoT Greengrass Version 2 Developer Guide.

- Verwenden Sie den Abschnitt Automatisierte IP-Erkennung, um über die IP-Adresse des Kerngeräts eine Verbindung zum EMQX-Broker herzustellen.
- Um mithilfe eines DNS-Namens statt einer IP-Adresse eine Verbindung zum EMQX-Broker herzustellen, verwenden Sie den Abschnitt Manuelle Verwaltung.

Automated IP discovery

Diese Option ermöglicht es Ihrem Core-Gerät, seine IP-Adresse automatisch zu ermitteln und sie dem Broker-Zertifikat als Subject Alternative Name (SAN) hinzuzufügen.

1. Fügen Sie die `aws.greengrass.clientdevices.IPDetector` Komponente zur Bereitstellung Ihres Kerngeräts hinzu.
2. Stellen Sie die Änderungen auf Ihrem Gerät bereit
3. Warten Sie, bis die Bereitstellung abgeschlossen ist.

Nach Abschluss der Bereitstellung können Sie mithilfe der IP-Adresse des Brokers eine sichere TLS-Verbindung herstellen.

Die IP-Adresse wird dem Brokerzertifikat automatisch als SAN hinzugefügt.

Manual DNS and IP Configuration

Sie können Ihrem TLS-Zertifikat manuell DNS-Namen und IP-Adressen als Subject Alternative Names (SANs) hinzufügen. Diese Methode ist nützlich, wenn Sie einen DNS-Namen für Ihren Gateway-Host konfiguriert haben.

Important

Wenn Sie die IPDetector-Komponente verwenden, entfernen Sie sie aus Ihrer Bereitstellung, bevor Sie fortfahren. Die IPDetector-Komponente überschreibt manuelle Endpunktconfigurationen.

Um Endpunkte manuell zu konfigurieren

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie in der linken Navigationsleiste im Abschnitt Edge die Option Edge-Gateways aus.
3. Wählen Sie das zu konfigurierende Gateway aus.
4. Wählen Sie im Abschnitt Edge-Gateway-Konfiguration die URL Ihres Greengrass-Core-Geräts aus. Die Seite des Core-Geräts wird angezeigt.
5. Wählen Sie die Registerkarte Client-Geräte.
6. Wählen Sie Endpunkte verwalten aus.
7. Geben Sie im Dialogfeld „Endpunkte verwalten“ die DNS-Namen und alle IP-Adressen ein, die Sie als SANs hinzufügen möchten. Verwenden Sie Port 8883.
8. Wählen Sie Aktualisieren aus.

Das TLS-Serverzertifikat des Brokers wird automatisch aktualisiert und schließt Ihre neuen Endpunkte mit ein.

Um das Update des TLS-Serverzertifikats unter Linux zu überprüfen

1. Starten Sie eine Shell-Sitzung auf Ihrem Gateway-Host.

```
docker exec emqx openssl x509 -in ./data/cert.pem -text -noout | grep -A1
"Subject Alternative Name"
```

2. Der Befehl gibt eine Ausgabe zurück, die der folgenden ähnelt:

```
X509v3 Subject Alternative Name:
DNS: endpoint_you_added, DNS:localhost
```

3. Stellen Sie sicher, dass Ihr Endpunkt in der Liste der SANs aufgeführt ist.

Um das Update des TLS-Serverzertifikats unter Windows zu überprüfen

1. Starten Sie eine Shell-Sitzung auf Ihrem Gateway-Host.

```
(Get-PfxCertificate -FilePath "C:\greengrass\v2\work
\aws.greengrass.clientdevices.mqtt.EMQX\v2\data\cert.pem").Extensions | Where-
Object { $_.Oid.FriendlyName -eq "Subject Alternative Name" } | ForEach-Object
{ "Subject Alternative Name:", ($_.Format($true) -split "`n")[0..1] }
```

2. Der Befehl gibt eine Ausgabe zurück, die der folgenden ähnelt:

```
Subject Alternative Name:
DNS Name=your-endpoint
DNS Name=localhost
```

3. Stellen Sie sicher, dass der von Ihnen hinzugefügte Endpunkt in der Liste der SANs enthalten ist.

Testen Sie die EMQX-Brokerverbindung auf AWS IoT SiteWise Edge

Nachdem Sie Ihren EMQX-Broker mit TLS-Zertifikaten und Authentifizierungsdaten konfiguriert haben, müssen Sie unbedingt überprüfen, ob Ihr Setup ordnungsgemäß funktioniert. Durch das Testen der Verbindung können Sie sicherstellen, dass Ihre Sicherheitskonfigurationen ordnungsgemäß implementiert sind und dass Clients erfolgreich verschlüsselte Verbindungen zum Broker herstellen können. In diesem Abschnitt wird gezeigt, wie Sie Ihre Broker-Verbindung mit dem Mosquitto Command Line Interface (CLI) -Client testen, einem weit verbreiteten MQTT-Client-Tool, das TLS-Verschlüsselung und -Authentifizierung unterstützt.

Verwenden Sie den Mosquitto CLI-Client, um die EMQX-Broker-Verbindung zu testen

In diesem Schritt verwenden wir den Mosquitto CLI-Client, um unser Setup zu testen und sicherzustellen, dass wir mit dem zuvor erstellten Benutzernamen und Passwort erfolgreich eine Verbindung zum Broker herstellen können. Die `BrokerCoreDeviceCA.pem` folgenden Schritte finden Sie unter Schritt 3: TLS einrichten.

```
mosquitto_sub -h hostname/ip address \  
  -p 8883 \  
  -t "#" \  
  -q 1 \  
  -u username -P password \  
  --cafile BrokerCoreDeviceCA.pem
```

Note

Möglicherweise wird eine `SSL:verify` Fehlermeldung angezeigt, wenn die `hostname/IP` Adresse, mit der Sie eine Verbindung herstellen, nicht mit dem Subject Alternative Name (SAN) übereinstimmt, der auf dem CA-Zertifikat steht, das Sie an den Client weitergeben. Informationen dazu, wie Sie ein Zertifikat mit dem richtigen SAN erhalten, finden Sie unter „Hinzufügen von benutzerdefinierten names/IP DNS-Adressen zum TLS-Serverzertifikat“ unter Schritt 3: TLS einrichten.

Zu diesem Zeitpunkt können alle Benutzer alle Themen auf dem Broker veröffentlichen und abonnieren. Fahren Sie mit [Richten Sie Autorisierungsregeln ein für AWS IoT SiteWise Edge in EMQX](#) fort.

Verwenden Sie Ihre eigene CA

AWS IoT Greengrass beschreibt, wie Sie Ihre eigene Authentifizierungskomponente für Client-Geräte so konfigurieren, dass sie Ihre eigene Zertifizierungsstelle (CA) verwendet. Die Authentifizierungskomponente für Clientgeräte (`aws.greengrass.clientdevices.Auth`) authentifiziert Client-Geräte und autorisiert Aktionen auf Client-Geräten. Weitere Informationen finden Sie unter [Verwenden Ihrer eigenen Zertifizierungsstelle im Entwicklerhandbuch](#). AWS IoT Greengrass Version 2

Um Ihre eigene Zertifizierungsstelle zu verwenden, fügen Sie die `aws.greengrass.clientdevices.Auth` Komponente zu Ihrer Bereitstellung hinzu, sodass Sie eine benutzerdefinierte Konfiguration angeben können.

Öffnen Sie Port 8883 für externe Firewallverbindungen

Linux

Fügen Sie in Ihrer Linux-Host-Firewallregel eine eingehende Regel für Port 8883 hinzu, um eingehende Verbindungen von außerhalb des Gateway-Hosts zuzulassen. Wenn Firewalls vorhanden sind, stellen Sie sicher, dass eingehende TLS-Verbindungen auf Port 8883 zulässig sind.

Windows

Fügen Sie in Ihrer Microsoft Windows Host-Firewallregel eine Regel für eingehenden Datenverkehr für Port 8883 hinzu, um eingehende Verbindungen von außerhalb des Gateway-Hosts zuzulassen. Stellen Sie sicher, dass es sich bei der Regel um eine Zulassungsregel vom Typ Port handelt, wobei Port 8883 angegeben wird. Sie können dies entsprechend Ihrer Netzwerkkonfiguration konfigurieren, um Verbindungen von Ihren externen Anwendungen zum Broker zuzulassen.

Richten Sie Autorisierungsregeln ein für AWS IoT SiteWise Edge in EMQX

EMQX unterstützt das Hinzufügen von Autorisierungsregeln auf der Grundlage von Kennungen wie Benutzername, IP-Adresse oder Client-ID. Dies ist nützlich, wenn Sie die Anzahl der externen Anwendungen einschränken möchten, die eine Verbindung zu verschiedenen Vorgängen oder Themen herstellen.

Themen

- [Konfigurieren Sie die Autorisierung mithilfe der integrierten Datenbank unter Linux](#)
- [Konfigurieren Sie die Autorisierung mithilfe der integrierten Datenbank in Windows](#)
- [Aktualisieren Sie die EMQX-Bereitstellungskonfiguration für die Autorisierung](#)
- [Fügen Sie Autorisierungsregeln über das EMQX-Dashboard für Benutzer hinzu](#)

Konfigurieren Sie die Autorisierung mithilfe der integrierten Datenbank unter Linux

Wenn Sie Autorisierungsregeln konfigurieren, gibt es zwei Konfigurationsoptionen, die von Ihrer Bereitstellungseinrichtung abhängen.

- **Docker**— Wenn Sie eine Docker Standardinstallation ohne ausführen Litmus Edge, verwenden Sie die Docker Bridge Gateway-Konfiguration. Dies ist normalerweise der Fall, wenn Sie nur AWS IoT SiteWise Komponenten bereitgestellt haben.

- Litmus Edge— Wenn Sie auf Ihrem Gateway Litmus Edge installiert haben, verwenden Sie die Litmus EdgeNetzwerk-Subnetzkonfiguration.

Note

Wenn Sie das Docker Bridge-Gateway zunächst konfigurieren und später installieren Litmus Edge, konfigurieren Sie die Autorisierungsregeln mithilfe der Litmus Edge Netzwerk-Subnetzoption neu, um eine ordnungsgemäße Kommunikation zwischen allen Komponenten sicherzustellen.

Um grundlegende Autorisierungsregeln hinzuzufügen

1. Stellen Sie sicher, dass der EMQX-Broker bereitgestellt ist und ausgeführt wird.
2. Starten Sie eine Shell-Sitzung auf Ihrem Gateway-Host.

Docker without Litmus Edge

Führen Sie für eine Docker Standardinstallation ohne folgenden Litmus Edge Befehl aus:

```
/greengrass/v2/bin/swe-emqx-cli acl init
```

Litmus Edge network subnet

Wenn du verwendest Litmus Edge, ermittle die Litmus Edge-Netzwerksubnetz-IP:

```
docker network inspect LitmusNetwork | grep IPAM -A9
```

Notiere dir den Subnetzwert aus der Ausgabe und führe den folgenden Befehl aus.

`litmus_subnet_ip` Ersetzen Sie ihn durch den Subnetzwert aus dem vorherigen Schritt.

```
/greengrass/v2/bin/swe-emqx-cli acl init litmus_subnet_ip
```

Das Tool erstellt automatisch Autorisierungsregeln und wendet sie an, um Verbindungen von der angegebenen IP-Adresse zum Broker zuzulassen. Es ermöglicht den Zugriff auf alle Themen. Dazu gehören der IoT SiteWise OPC UA Collector und der IoT SiteWise Publisher.

3. Fahren Sie mit [Aktualisieren Sie die EMQX-Bereitstellungskonfiguration für die Autorisierung](#) fort.

Konfigurieren Sie die Autorisierung mithilfe der integrierten Datenbank in Windows

In diesem Abschnitt wird die Konfiguration von Autorisierungsregeln mithilfe der integrierten Datenbank für Windows-Bereitstellungen behandelt.

Um grundlegende Autorisierungsregeln hinzuzufügen

1. Stellen Sie sicher, dass der EMQX-Broker bereitgestellt ist und ausgeführt wird.
2. Führen Sie das AWS IoT SiteWise EMQX CLI-Tool aus:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 acl init
```

Das Tool erstellt und wendet automatisch ACL-Regeln an, die Verbindungen von localhost (127.0.0.1) zum Broker zulassen. Es ermöglicht den Zugriff auf alle Themen. Dazu gehören der IoT SiteWise OPC UA Collector und der IoT SiteWise Publisher.

3. Fahren Sie mit [Aktualisieren Sie die EMQX-Bereitstellungskonfiguration für die Autorisierung](#) fort.

Aktualisieren Sie die EMQX-Bereitstellungskonfiguration für die Autorisierung

Um die EMQX-Bereitstellungskonfiguration für die Autorisierung zu aktualisieren

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie in der linken Navigationsleiste im Abschnitt Edge die Option Edge-Gateways aus.
3. Wählen Sie das zu konfigurierende Gateway aus.
4. Kopieren Sie im Abschnitt Edge-Gateway-Konfiguration Ihren Greengrass-Core-Gerätewert. Speichern Sie ihn für die spätere Verwendung.
5. Öffnen Sie die [AWS IoT-Konsole](#).
6. Wählen Sie in der linken Navigationsleiste im Bereich Verwalten die Option Greengrass-Geräte und dann Bereitstellungen aus.
7. Suchen Sie den Core-Gerätewert, den Sie zuvor gespeichert haben, und wählen Sie diesen Link, um die Bereitstellung zu öffnen.
8. Wählen Sie die Dropdownschaltfläche „Aktionen“ und anschließend „Überarbeiten“.
9. Lesen Sie die angezeigte Meldung und wählen Sie dann Bereitstellung überarbeiten aus. Die Seite „Ziel angeben“ wird angezeigt.
10. Wählen Sie Weiter, bis Sie zum Schritt Komponenten konfigurieren gelangen.

11. Wählen Sie das Optionsfeld `aws.greengrass.clientdevices.mqtt.EMQX`.
12. Wählen Sie die Schaltfläche Komponente konfigurieren. Eine Konfigurationsseite für die Komponente wird angezeigt.
13. Wählen Sie unter Konfigurationsupdate die Option Auf Standardkonfiguration zurücksetzen für die Komponentenversion: 2.* aus. *.
14. Fügen Sie je nach Betriebssystem den folgenden Inhalt in den Abschnitt Konfiguration zum Zusammenführen ein.

Linux

```
{
  "emqxConfig": {
    "authorization": {
      "no_match": "deny",
      "sources": [
        {
          "type": "built_in_database"
        },
        {
          "type": "file",
          "path": "data/authz/acl.conf"
        }
      ]
    },
    "listeners": {
      "tcp": {
        "default": {
          "enabled": true,
          "enable_authn": false
        }
      },
      "ssl": {
        "default": {
          "enabled": true,
          "enable_authn": true,
          "ssl_options": {
            "verify": "verify_none",
            "fail_if_no_peer_cert": false
          }
        }
      }
    }
  },
}
```

```

    "authentication": {
      "enable": true,
      "backend": "built_in_database",
      "mechanism": "password_based",
      "password_hash_algorithm": {
        "iterations": 210000,
        "mac_fun": "sha512",
        "name": "pbkdf2"
      },
      "user_id_type": "username"
    },
    "dashboard": {
      "listeners": {
        "http": {
          "bind": 18083
        }
      }
    }
  },
  "authMode": "bypass",
  "dockerOptions": "-p 8883:8883 -p 127.0.0.1:1883:1883\n-p 127.0.0.1:18083:18083 -v emqx-data:/opt/emqx/data -e\nEMQX_NODE__NAME=emqx@local",
  "requiresPrivilege": "true"
}

```

Windows

```

{
  "emqxConfig": {
    "authorization": {
      "no_match": "deny",
      "sources": [
        {
          "type": "built_in_database"
        },
        {
          "type": "file",
          "path": "C:\\greengrass\\v2\\work\\aws.greengrass.clientdevices.mqtt.EMQX\\v2\\data\\authz\\acl.conf"
        }
      ]
    }
  },

```

```
"listeners": {
  "tcp": {
    "default": {
      "enabled": true,
      "enable_authn": false
    }
  },
  "ssl": {
    "default": {
      "enabled": true,
      "enable_authn": true,
      "ssl_options": {
        "verify": "verify_none",
        "fail_if_no_peer_cert": false
      }
    }
  }
},
"authentication": {
  "enable": true,
  "backend": "built_in_database",
  "mechanism": "password_based",
  "password_hash_algorithm": {
    "iterations": 210000,
    "mac_fun": "sha512",
    "name": "pbkdf2"
  },
  "user_id_type": "username"
},
"dashboard": {
  "listeners": {
    "http": {
      "bind": 18083
    }
  }
},
"authMode": "bypass",
"requiresPrivilege": "true"
}
```

15. Wählen Sie Bestätigen aus.

16. Wählen Sie Weiter, bis Sie zum Schritt Überprüfen gelangen.

17. Wählen Sie Bereitstellen.

Note

Ab diesem Zeitpunkt können Sie die ACL-Datei nicht mehr bearbeiten, um die Autorisierungsregeln zu aktualisieren. Alternativ können Sie [Fügen Sie Autorisierungsregeln über das EMQX-Dashboard für Benutzer hinzu](#) nach einer erfolgreichen Bereitstellung mit fortfahren.

Fügen Sie Autorisierungsregeln über das EMQX-Dashboard für Benutzer hinzu

Sie können Autorisierungsregeln mit dem EMQX-Dashboard oder dem EMQX-CLI-Tool hinzufügen oder aktualisieren. AWS IoT SiteWise Das AWS IoT SiteWise EMQX CLI-Tool verwaltet die Autorisierung mithilfe der integrierten Datenbank von EMQX.

Note

Das Hinzufügen von Autorisierungsregeln ist ein fortgeschrittener Konfigurationsschritt, der Kenntnisse der MQTT-Themenmuster und der Zugriffskontrolle erfordert. Weitere Informationen zum Erstellen von Autorisierungsregeln mithilfe der integrierten Datenbank von EMQX finden Sie unter Use [Built-in Database](#) in den EMQX-Dokumenten.

Note

Die bereitgestellten EMQX-related Anweisungen dienen nur als Referenz. Da sich die Dokumentation und Funktionen von EMQX im Laufe der Zeit ändern können und wir ihre Dokumentation nicht pflegen, empfehlen wir, die [offizielle Dokumentation von EMQX](#) für die aktuellsten Informationen zu konsultieren.

EMQX dashboard

Dieses Verfahren zeigt, wie Sie Autorisierungsregeln im EMQX-Dashboard hinzufügen können.

Auf das EMQX-Dashboard kann nur vom Gateway-Host aus zugegriffen werden. Wenn Sie versuchen, von außerhalb des Gateway-Hosts eine Verbindung herzustellen, können Sie nicht auf das Dashboard zugreifen.

Um Autorisierungsregeln mithilfe des EMQX-Dashboards hinzuzufügen

1. Stellen Sie sicher, dass Sie sich innerhalb des Gateway-Hosts befinden.
2. Öffnen Sie ein Browserfenster und besuchen Sie <http://localhost:18083/>.
3. Melden Sie sich im EMQX-Dashboard an. Bei diesem Verfahren wird davon ausgegangen, dass Sie Ihre Standardanmeldedaten in etwas Ihrer Wahl geändert haben. Weitere Informationen zur Ersteinrichtung finden Sie unter [Aktivieren Sie die Authentifizierung mit Benutzername und Passwort](#).
4. Wählen Sie im Dropdownmenü das Schildsymbol und anschließend Autorisierung aus.
5. Wählen Sie in der Built-in Datenbankzeile die Schaltfläche „Berechtigungen“.
6. Fügen Sie im Abschnitt Built-in Datenbankautorisierung die Benutzerautorisierungsregeln für Ihre Geschäftsanforderungen hinzu oder aktualisieren Sie sie. Weitere Anleitungen zur Erstellung von Regeln finden Sie im Abschnitt [Built-in Datenbank verwenden](#) in den EMQX-Dokumenten.

AWS IoT SiteWiseCLI tool using Linux

Um Autorisierungsregeln mit dem zu verwalten AWS IoT SiteWise EMQX CLI-Tool unter Linux:

- Fügen Sie Autorisierungsregeln für einen Benutzer im folgenden Format hinzu:

```
/greengrass/v2/bin/swe-emqx-cli auth add your-username your-action your-permission your-topic [your-action-permission-topic]
```

Example Fügen Sie Autorisierungsregeln für einen Benutzer hinzu

Dieses Beispiel zeigt, wie Regeln für einen Benutzer mit dem Namen hinzugefügt werden `system1`:

```
/greengrass/v2/bin/swe-emqx-cli auth add system1 \  
  publish allow "sensors/#" \  
  subscribe allow "control/#" \  
  all deny "#"
```

Example: Autorisierungsregeln für einen Benutzer anzeigen

Führen Sie den folgenden Befehl aus, um die Autorisierungsregeln für die `system1` Benutzer anzuzeigen:

```
/greengrass/v2/bin/swe-emqx-cli auth list system1
```

Example: Alle vorhandenen Autorisierungsregeln anzeigen

Führen Sie den folgenden Befehl aus, um alle Autorisierungsregeln anzuzeigen, die Sie derzeit haben:

```
/greengrass/v2/bin/swe-emqx-cli auth list
```

Example: Löscht alle Autorisierungsregeln für einen Benutzer

Führen Sie den folgenden Befehl aus, um alle Autorisierungsregeln zu löschen, die für einen bestimmten Benutzer gelten:

```
/greengrass/v2/bin/swe-emqx-cli auth delete system1
```

Sie werden aufgefordert, den Löschvorgang zu bestätigen.

AWS IoT SiteWiseCLI tool using Windows

Um Autorisierungsregeln mit dem zu verwalten AWS IoT SiteWise EMQX CLI-Tool in Windows PowerShell:

- Fügen Sie Autorisierungsregeln für einen Benutzer im folgenden Format hinzu:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth add your-username your-action your-permission your-topic [your-action-permission-topic]
```

Example: Autorisierungsregeln für einen Benutzer hinzufügen

Dieses Beispiel zeigt, wie Regeln für einen Benutzer mit dem Namen hinzugefügt werden `system1`:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth add system1 `
  publish allow "sensors/#" `
  subscribe allow "control/#" `
```

```
all deny "#"
```

Example: Autorisierungsregeln für einen Benutzer anzeigen

Führen Sie den folgenden Befehl aus, um die Autorisierungsregeln für die system1 Benutzer anzuzeigen:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth list system1
```

Example: Alle vorhandenen Autorisierungsregeln anzeigen

Führen Sie den folgenden Befehl aus, um alle Autorisierungsregeln anzuzeigen, die Sie derzeit haben:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth list
```

Example: Löscht alle Autorisierungsregeln für einen Benutzer

Führen Sie den folgenden Befehl aus, um alle Autorisierungsregeln zu löschen, die für einen bestimmten Benutzer gelten:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth delete system1
```

Sie werden aufgefordert, den Löschvorgang zu bestätigen.

Verarbeiten und visualisieren Sie Daten mit SiteWise Edge- und Open-Source-Tools

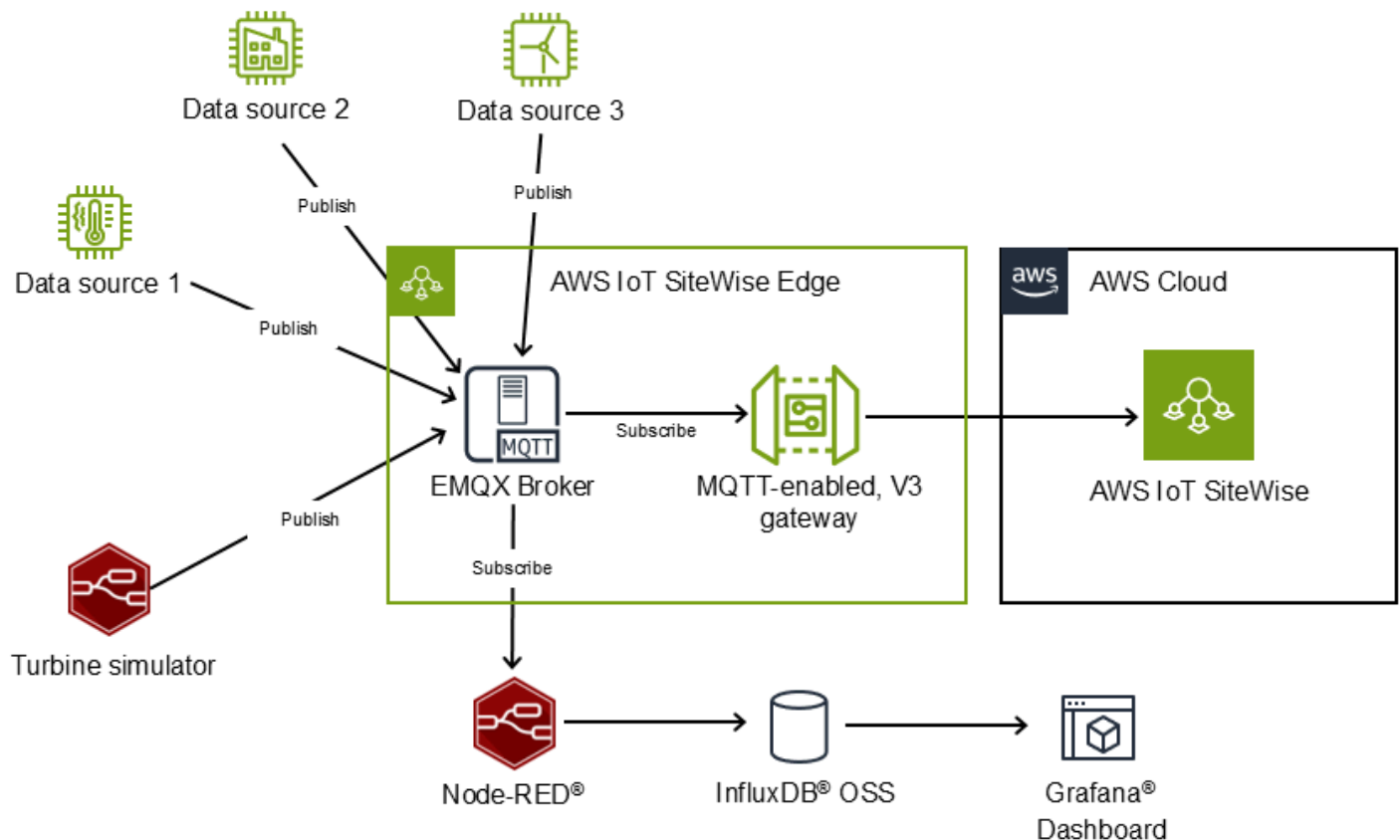
Konfigurieren Sie AWS IoT SiteWise Edge-MQTT-fähige Gateways mit Open-Source-Tools für die lokale Verarbeitung und Visualisierung, um Ihre industriellen Datenverwaltungsfunktionen zu verbessern.

Mit SiteWise Edge können Sie mithilfe externer Open-Source-Tools eine lokale Datenverarbeitungspipeline erstellen. [Verwenden Sie Node-RED®, um Zeitreihendaten mit InfluxDB® zu speichern und den Betrieb über Grafana® -Dashboards zu überwachen.](#)

Node-RED verarbeitet und transformiert Ihre Datenflüsse, während InfluxDB die Speicherung von Zeitreihendaten ermöglicht. Grafana zeigt Ihre Betriebsdaten in Echtzeit an. Verwenden Sie diese Tools mit SiteWise Edge, um Daten zwischen Ihrer lokalen Umgebung und der zu synchronisieren AWS Cloud, sodass Sie sowohl sofortige lokale Einblicke als auch langfristige Cloud-Analysefunktionen erhalten.

Note

Node-RED®, InfluxDB® und Grafana® sind keine Anbieter oder Lieferanten für Edge SiteWise

**Note**

In diesem Handbuch verwenden wir die Open-Source-Version von [Grafana](#) for SiteWise Edge im Gegensatz zum [Amazon Managed Grafana-Service](#).

Optionen für die Bereitstellung

Sie können diese Lösung mit einem von zwei Ansätzen bereitstellen. Mit einer Microsoft Windows manuellen Einrichtung kontrollieren Sie die Konfiguration und Integration der Komponenten in Ihre Infrastruktur. Mit können Sie Docker verwendenLinux, um vorkonfigurierte Komponenten in Containern bereitzustellen.

Wählen Sie die Methode, die Ihren betrieblichen Anforderungen entspricht.

- [Richten Sie Open-Source-Integrationen manuell ein \(Windows\)](#)— Für benutzerdefinierte Konfigurationen oder bestehende Infrastruktur
- [Richten Sie Open-Source-Integrationen mit Docker \(Linux\) ein](#)— Für eine schnelle Bereitstellung mit vorkonfigurierten Komponenten

Übersicht über das Beispiel eines Windparks

In diesem Leitfaden wird anhand eines Beispiels für einen Windpark veranschaulicht, wie Sie die Windgeschwindigkeit einer Turbine in einem Windpark überwachen können. Dieses praktische Szenario veranschaulicht allgemeine Anforderungen an die industrielle Überwachung, bei denen sowohl lokale als auch cloudbasierte Transparenz für die betriebliche Effizienz von Nutzen sind.

Mit dieser Integration können Sie:

- Erfassen Sie Daten von Industrieanlagen mithilfe eines AWS IoT SiteWise Edge-Gateways
- Verarbeiten Sie Daten lokal mit Node-RED, InfluxDB und Grafana
- Speichern Sie Daten lokal mit InfluxDB
- Überwachen Sie Daten in Echtzeit mithilfe von Grafana-Dashboards

In diesem Handbuch verwenden wir das Beispiel eines Windparks. Wir verwenden Node-RED, um eine Turbine zu simulieren, die Windgeschwindigkeitdaten generiert. Node-RED übersetzt die Datennutzdaten, veröffentlicht die Daten auf dem SiteWise Edge-MQTT-Broker, abonniert den Empfang von Daten vom Broker und speichert die Daten lokal in InfluxDB. Dieser Ansatz stellt sicher, dass alle Betriebsdaten sowohl lokal für den sofortigen Zugriff als auch in der Cloud für weitere Analysen verfügbar sind. Durch die Implementierung dieses Musters erhöhen Sie die Widerstandsfähigkeit gegenüber Netzwerkunterbrechungen und behalten gleichzeitig die Möglichkeit, fortschrittliche Analysen in der AWS Cloud durchzuführen. Grafana stellt für die lokale Überwachung eine Verbindung zu InfluxDB her und bietet Betreibern so Echtzeiteinblicke in Metriken ohne Cloud-Abhängigkeiten. Ein SiteWise Edge-MQTT-fähiges Gateway stellt eine Verbindung zu demselben MQTT-Broker her, an den Daten gesendet werden sollen AWS IoT SiteWise, und schafft so eine Brücke zwischen Ihren Edge-Operationen und Cloud-basierten Diensten.

Sie können Ihre eigenen Daten und Konfigurationen verwenden, um einen ähnlichen Workflow zu erstellen, der auf Ihre spezifischen industriellen Anforderungen zugeschnitten ist, unabhängig davon, ob Sie Produktionsanlagen, Versorgungsinfrastrukturen oder andere Industrieanlagen überwachen.

Anforderungen für Open-Source-Integrationen

Stellen Sie vor der Implementierung von Open-Source-Integrationen mit SiteWise Edge sicher, dass Ihre Umgebung die erforderlichen Anforderungen erfüllt.

- Hardwareanforderungen — Ihre Gateway-Hardware muss die Anforderungen für SiteWise Edge-Gateways erfüllen. Weitere Informationen finden Sie unter [AWS IoT SiteWise Anforderungen an das selbst gehostete Edge-Gateway](#) für MQTT-fähige V3-Gateways und [Anforderungen für die AWS IoT SiteWise Edge-Anwendung](#)

Important

Stellen Sie bei der Bereitstellung zusätzlicher Open-Source-Komponenten sicher, dass Ihre Hardware die Anforderungen für InfluxDB, Node-RED und Grafana erfüllt.

- Ihre Netzwerkkonfiguration muss sowohl die lokale Kommunikation zwischen Komponenten als auch Cloud-Konnektivität für Edge unterstützen. SiteWise
- Alle Dienste müssen auf demselben Host ausgeführt werden.

Sicherheitsüberlegungen

Es wird empfohlen, die gesamte Kommunikation zwischen Komponenten zu verschlüsseln, insbesondere beim Zugriff auf Schnittstellen von nicht lokalen Netzwerken aus. Implementieren Sie die richtigen Zugriffskontrollen für jede Komponente und befolgen Sie die AWS bewährten Methoden für die Konfiguration AWS IoT SiteWise des Edge-Gateways und die AWS Kontosicherheit.

Entwicklungsumgebung

Dieses Handbuch zeigt, wie Node-RED, InfluxDB und Grafana lokal auf einem Gateway-Host ausgeführt und darauf zugegriffen wird. Implementieren Sie für Produktionsbereitstellungen, die externen Zugriff erfordern, Sicherheitsmaßnahmen wie TLS-Verschlüsselung, Authentifizierung und Autorisierung. Halten Sie sich an die bewährten Sicherheitsmethoden der einzelnen Anwendungen.

Software von Drittanbietern

Diese Lösung verwendet Software von Drittanbietern, die nicht von InfluxDB AWS, Node-RED, Grafana und dem Plugin verwaltet wird. `node-red-contrib-influxdb` Stellen Sie vor der Bereitstellung sicher, dass diese Komponenten den Sicherheitsanforderungen, Compliance-Standards und Governance-Richtlinien Ihres Unternehmens entsprechen.

Important

In diesem Handbuch wird auf Software von Drittanbietern verwiesen und verwendet, die nicht Eigentum von Drittanbietern ist oder von dieser nicht verwaltet wird AWS. Stellen Sie vor der Implementierung sicher, dass alle Komponenten Ihren Sicherheits-, Compliance- und Governance-Anforderungen entsprechen. Halten Sie die gesamte Software mit den neuesten Sicherheitspatches auf dem neuesten Stand und befolgen Sie die bewährten Methoden zur Sicherung Ihrer Edge-Bereitstellung.

InfluxDB, Node-RED, Grafana sind keine Anbieter oder Lieferanten für Edge. SiteWise

Weitere Überlegungen

Berücksichtigen Sie diese zusätzlichen Faktoren bei der Implementierung von Open-Source-Integrationen mit Edge. SiteWise

- Verwenden Sie die neuesten Versionen aller Dienste, Tools und Komponenten.
- Filtern und aggregieren Sie Daten vor der Cloud-Übertragung lokal, um die Kosten für die AWS IoT SiteWise Datenaufnahme zu senken. Konfigurieren Sie geeignete Datenaufbewahrungszeiträume in InfluxDB und dimensionieren Sie Ihre Gateway-Hardware richtig. Weitere Informationen finden Sie unter [AWS IoT SiteWise Preise](#).
- Implementieren Sie regelmäßige Backup-Verfahren für alle Daten.
- Überwachen Sie die Ressourcennutzung auf Ihrem Gateway und konfigurieren Sie entsprechende Ressourcenlimits für jede Komponente. Implementieren Sie Datenaufbewahrungsrichtlinien in InfluxDB, um die Festplattennutzung zu verwalten.

Richten Sie Open-Source-Integrationen manuell ein (Windows)

Verwenden Sie diese Anleitung, um manuell einen Zeitreihen-Bucket für Windgeschwindigkeitsdaten zu erstellen, der eine Verbindung zu Grafana® und Node-RED® herstellt.

Installieren und konfigurieren Sie Node-RED, InfluxDB® und Grafana manuell, um Ihre Bereitstellungskonfiguration Microsoft Windows zu kontrollieren. Mit InfluxDB können Sie Zeitreihendaten von Ihren Geräten speichern und verwalten.

Voraussetzungen für die manuelle Einrichtung

Bevor Sie beginnen, müssen Sie die folgenden Anforderungen erfüllen:

 Note

Führen Sie alle Dienste (SiteWise Edge, InfluxDB, Node-RED und Grafana) auf demselben Host aus.

- Installieren Sie ein MQTT-fähiges V3-Gateway. Weitere Informationen finden Sie unter [MQTT-enabled, V3-Gateways fürAWS IoT SiteWiseEdge](#).
- Installieren Sie diese Dienste lokal und führen Sie sie aus:
 - InfluxDB OSS v2. Installationsschritte finden Sie unter InfluxDB [installieren](#).
 - Node-RED. Installationsschritte finden Sie unter [Node-RED lokal installieren](#).
 - Grafana. Installationsschritte finden [Sie unter Grafana installieren](#).

Richten Sie lokalen Speicher mit InfluxDB ein

Mit InfluxDB® können Sie Zeitreihendaten von Ihren Geräten lokal speichern. Der Zweck der lokalen Speicherfähigkeit besteht darin, die betriebliche Transparenz bei Netzwerkunterbrechungen aufrechtzuerhalten und die Latenz für zeitkritische Anwendungen zu reduzieren. Sie können Analysen und Visualisierungen am Netzwerkrand durchführen und haben dennoch die Möglichkeit, Daten selektiv an die Cloud weiterzuleiten.

In diesem Abschnitt erstellen Sie einen Zeitreihen-Bucket für Windgeschwindigkeitdaten von Turbinen und generieren ein API-Token für Grafana® - und Node-RED® -Konnektivität. Der InfluxDB-Bucket dient als dedizierter Speichercontainer für Ihre Zeitreihendaten, ähnlich einer Datenbank in herkömmlichen Systemen. Das API-Token ermöglicht einen sicheren programmatischen Zugriff auf Ihre Daten.

Um InfluxDB einzurichten

1. [Nachdem Sie die erforderlichen Schritte abgeschlossen und sichergestellt haben, dass alle Tools auf demselben Host ausgeführt werden, öffnen Sie Ihren Webbrowser und gehen Sie zu http://127.0.0.1:8086.](#)
2. (Optional) Aktivieren Sie die TLS-Verschlüsselung für mehr Sicherheit. Weitere Informationen finden Sie unter [Aktivieren der TLS-Verschlüsselung](#) in der InfluxData Documentation.
3. Erstellen Sie einen InfluxDB-Bucket mit Zeitreihen, um Daten von Node-RED zu speichern. Der Bucket dient als dedizierter Container für Ihre Windparkdaten, sodass Sie die für diesen

Datensatz spezifischen Aufbewahrungsrichtlinien organisieren und verwalten können. Weitere Informationen finden Sie unter [Buckets verwalten](#) in der InfluxData Documentation.

4. (Optional) Konfigurieren Sie den Datenaufbewahrungszeitraum für Ihren Edge-Standort. Durch die Festlegung geeigneter Aufbewahrungsfristen können Speicherressourcen effizient verwaltet werden, da ältere Daten, die für den lokalen Betrieb nicht mehr benötigt werden, automatisch entfernt werden.

Informationen zur Datenspeicherung finden Sie unter [Datenspeicherung in InfluxDB](#) im InfluxData Documentation

5. Generieren Sie ein API-Token für den Bucket. Dieses Token ermöglicht eine sichere Kommunikation zwischen InfluxDB und anderen Komponenten wie Node-RED und Grafana. Auf diese Weise können nur autorisierte Dienste aus Ihrem Datenspeicher lesen oder in ihn schreiben. Weitere Informationen finden Sie unter [Erstellen eines Tokens](#) in der InfluxData Documentation.

Nachdem Sie diese Schritte abgeschlossen haben, können Sie Zeitreihendaten in Ihrer InfluxDB-Instance speichern und so eine Grundlage für die lokale Datenpersistenz und -analyse in Ihrer Edge-Umgebung schaffen.

Konfigurieren Sie Node-RED-Flows für die Datenintegration AWS IoT SiteWise

Mit Node-RED® können Sie zwei Abläufe implementieren, um Daten zwischen Ihren Geräten und zu verwalten. AWS IoT SiteWise Diese Datenflüsse bilden zusammen eine umfassende Datenverwaltungslösung, die sowohl den lokalen als auch den Cloud-Datenfluss berücksichtigt.

- Ablauf der Datenveröffentlichung — Publiziert in der Cloud. Der Datenveröffentlichungsfluss sendet Daten an AWS IoT SiteWise. Dieser Flow simuliert ein Turbinengerät, indem Sensordaten generiert, in ein AWS IoT SiteWise Format übersetzt und im SiteWise Edge-MQTT-Broker veröffentlicht werden. Auf diese Weise können Sie die Cloud-Funktionen für Speicherung, Analyse und Integration mit anderen Diensten nutzen AWS IoT SiteWise. AWS

Weitere Informationen finden Sie unter [Konfigurieren Sie den Ablauf der Datenveröffentlichung](#).

- Ablauf der Datenspeicherung — Speichert Daten am Netzwerkrand. Der Datenaufbewahrungsablauf abonniert den SiteWise Edge-MQTT-Broker, um Daten zu empfangen, sie in das InfluxDB® -Format zu übersetzen und sie zur Überwachung lokal zu speichern. Dieser lokale Speicher bietet sofortigen Zugriff auf Betriebsdaten, reduziert die Latenz für zeitkritische Anwendungen und gewährleistet die Kontinuität bei Netzwerkunterbrechungen.

Weitere Informationen finden Sie unter [Konfigurieren Sie den Datenaufbewahrungsablauf](#).

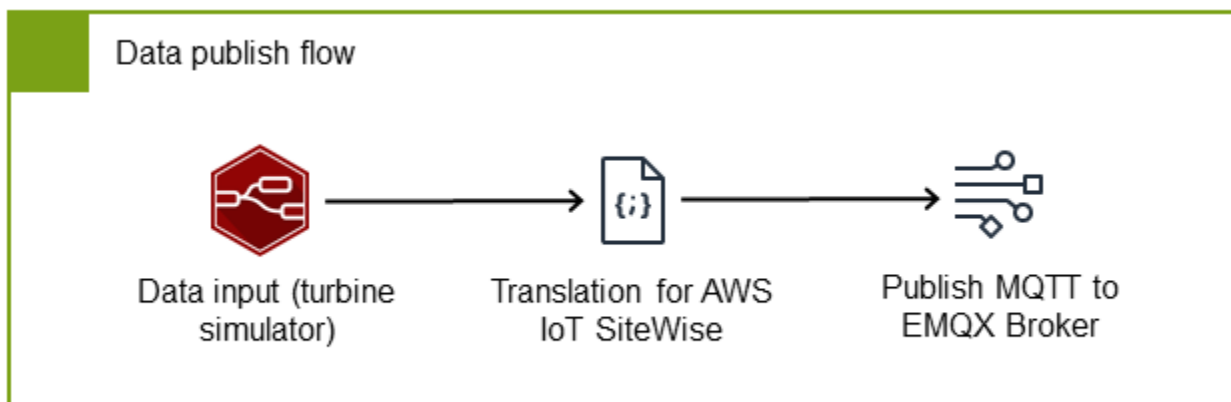
Diese beiden Datenflüsse arbeiten zusammen, um sicherzustellen, dass Daten sowohl an diese gesendet als auch lokal gespeichert werden, AWS IoT SiteWise sodass sie sofort darauf zugreifen können.

[Um auf Ihre Node-RED-Konsole zuzugreifen, gehen Sie zu `http://127.0.0.1:1880`](#). Informationen zur Aktivierung von TLS finden Sie unter [TLS-Verschlüsselung aktivieren](#).

Konfigurieren Sie den Ablauf der Datenveröffentlichung

Der Datenveröffentlichungsablauf verwendet drei Knoten, um eine Pipeline zu erstellen, die Ihre Industriedaten an die Cloud sendet. Dieser Ablauf ist unerlässlich, um cloudbasierte Analysen, Langzeitspeicherung und Integration mit anderen AWS Diensten zu ermöglichen. Zunächst werden simulierte Gerätedaten an den SiteWise Edge MQTT-Broker gesendet. Das Gateway empfängt die Daten vom Broker und ermöglicht so die Übertragung in die AWS IoT SiteWise Cloud, wo Sie leistungsstarke Analyse- und Visualisierungsfunktionen nutzen können.

- Dateneingabe — Empfängt Gerätedaten von Ihren Industrieanlagen oder Simulatoren
- Datenübersetzer für AWS IoT SiteWise — Übersetzt Daten in AWS IoT SiteWise ein Format, um die Kompatibilität mit dem SiteWise Edge-Gateway sicherzustellen
- MQTT-Publisher — Veröffentlicht Daten auf dem SiteWise Edge-MQTT-Broker und stellt sie sowohl lokalen als auch Cloud-Nutzern zur Verfügung



Konfigurieren Sie den Dateneingabeknoten

In diesem Beispiel verwendet der Dateneingabeknoten ein simuliertes Windturbinengerät, das Windgeschwindigkeitsdaten generiert. Dieser Knoten dient als Einstiegspunkt für Ihre Industriedaten, unabhängig davon, ob sie aus simulierten Quellen (wie in unserem Beispiel) oder aus tatsächlichen Industrieanlagen in Produktionsumgebungen stammen.

Wir verwenden ein benutzerdefiniertes JSON-Format für die Datennutzdaten, um eine standardisierte Struktur bereitzustellen, die sowohl mit lokalen Verarbeitungstools als auch mit dem AWS IoT SiteWise Cloud-Dienst effizient funktioniert. Dieses Format enthält neben den tatsächlichen Messwerten auch wichtige Metadaten wie Zeitstempel und Qualitätsindikatoren und ermöglicht so ein umfassendes Datenmanagement und eine Qualitätsverfolgung in Ihrer gesamten Pipeline. Importieren Sie den Inject-Node, um simulierte Daten in diesem standardisierten JSON-Format mit Zeitstempeln, Qualitätsindikatoren und Werten zu erhalten.

[Weitere Informationen zum Node-RED-Injektionsknoten finden Sie im Abschnitt Inject in der Dokumentation. Node-RED](#)

Der Turbinensimulator generiert jede Sekunde Windgeschwindigkeitsdaten in diesem standardisierten JSON-Format:

Example: Nutzlast der Turbinendaten

```
{
  name: string,          // Property name/identifier
  timestamp: number,     // Epoch time in nanoseconds
  quality: "GOOD" | "UNCERTAIN" | "BAD",
  value: number | string | boolean
}
```

Dieses Format bietet mehrere Vorteile:

- Das `name` Feld identifiziert die spezifische Eigenschaft oder Messung, sodass Sie mehrere Datenpunkte von demselben Gerät aus verfolgen können
- Die Angabe `timestamp` in Nanosekunden gewährleistet eine präzise Zeiterfassung für eine genaue historische Analyse
- Der `quality` Indikator hilft Ihnen, Daten auf der Grundlage ihrer Zuverlässigkeit zu filtern und zu verwalten
- Das flexible `value` Feld unterstützt verschiedene Datentypen, um verschiedene Sensorausgänge zu ermöglichen

Example: Inject-Knoten eines Turbinensimulators

```
[
  {
    "id": "string",
    "type": "inject",
    "z": "string",
    "name": "Turbine Simulator",
    "props": [
      {
        "p": "payload.timestamp",
        "v": "",
        "vt": "date"
      },
      {
        "p": "payload.quality",
        "v": "GOOD",
        "vt": "str"
      },
      {
        "p": "payload.value",
        "v": "$random()",
        "vt": "jsonata"
      },
      {
        "p": "payload.name",
        "v": "/Renton/WindFarm/Turbine/WindSpeed",
        "vt": "str"
      }
    ],
    "repeat": "1",
    "crontab": "",
    "once": false,
    "onceDelay": "",
    "topic": "",
    "x": 270,
    "y": 200,
    "wires": [
      [
        "string"
      ]
    ]
  }
]
```

]

Konfigurieren Sie einen Knoten für die Datenübersetzung

Das SiteWise Edge-Gateway benötigt Daten in einem bestimmten Format, um die Kompatibilität mit der AWS IoT SiteWise Cloud sicherzustellen. Der Translator-Knoten ist eine wichtige Komponente, die Ihre Eingabedaten in das erforderliche AWS IoT SiteWise Payload-Format konvertiert. Dieser Übersetzungsschritt stellt sicher, dass Ihre Industriedaten ordnungsgemäß verarbeitet, gespeichert und später in der AWS IoT SiteWise Cloud-Umgebung analysiert werden können.

Durch die Standardisierung des Datenformats in dieser Phase ermöglichen Sie die Integration zwischen Ihren Edge-Geräten und dem Cloud-Dienst, sodass Sie Funktionen zur Anlagenmodellierung, Analyse und Visualisierung nutzen können. Verwenden Sie diese Struktur:

Example: Payload-Struktur für das Analysieren von SiteWise Edge-Daten

```
{
  "propertyAlias": "string",
  "propertyValues": [
    {
      "value": {
        "booleanValue": boolean,
        "doubleValue": number,
        "integerValue": number,
        "stringValue": "string"
      },
      "timestamp": {
        "timeInSeconds": number,
        "offsetInNanos": number
      },
      "quality": "GOOD" | "UNCERTAIN" | "BAD",
    }
  ]
}
```

Note

Ordnen Sie `propertyAlias` die Ihrer MQTT-Themenhierarchie zu (zum Beispiel). / Renton/WindFarm/Turbine/WindSpeed Dadurch wird sichergestellt, dass Ihre Daten ordnungsgemäß mit der richtigen Asset-Eigenschaft in AWS IoT SiteWise verknüpft sind.

Weitere Informationen zum Konzept „Datenstream-Alias“ finden Sie unter [AWS IoT SiteWise Konzepte](#).

1. Importieren Sie den Beispielfunktionsknoten für die AWS IoT SiteWise Payload-Übersetzung. Diese Funktion übernimmt die Konvertierung von Ihrem standardisierten Eingabeformat in ein AWS IoT SiteWise-kompatibles Format und gewährleistet so die korrekte Formatierung von Zeitstempeln, Qualitätsindikatoren und Werteingabe.

```
[
  {
    "id": "string",
    "type": "function",
    "z": "string",
    "name": "Translate to SiteWise payload",
    "func": "let input = msg.payload;\nlet output = {};\n\noutput[\"propertyAlias\"] = input.name;\n\nlet propertyVal = {};\n\nlet timeInSeconds = Math.floor(input.timestamp / 1000);\nlet offsetInNanos = (input.timestamp % 1000) * 1000000;\n\npropertyVal[\"timestamp\"] = {\n  \"timeInSeconds\": timeInSeconds,\n  \"offsetInNanos\": offsetInNanos,\n};\n\npropertyVal[\"quality\"] = input.quality;\n\nlet typeNameConverter = {\n  \"number\": (x) => Number.isInteger(x) ? \"integerValue\" : \"doubleValue\",\n  \"boolean\": (x) => \"booleanValue\",\n  \"string\": (x) => \"stringValue\",\n};\n\nlet typeName = typeNameConverter[typeof input.value]\n(propertyVal[\"value\"] = {};\npropertyVal[\"value\"][typeName] = input.value;\n\noutput[\"propertyValues\"] = [propertyVal];\n\nreturn {\n  payload: JSON.stringify(output);\n};",
    "outputs": 1,
    "timeout": "",
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 530,
    "y": 200,
    "wires": [
      [
        "string"
      ]
    ]
  }
]
```

2. Stellen Sie sicher, dass der JavaScript Code die Windgeschwindigkeitdaten korrekt übersetzt. Die Funktion erfüllt mehrere wichtige Aufgaben:
 - Extrahiert den Eigenschaftsnamen aus der Eingabe und legt ihn als `propertyAlias` fest
 - Konvertiert den Zeitstempel von Millisekunden in das erforderliche Sekunden- und Nanosekundenformat
 - Behält den Qualitätsindikator für die Überwachung der Datenzuverlässigkeit bei
 - Erkennt automatisch den Wertetyp und formatiert ihn entsprechend den Anforderungen AWS IoT SiteWise
3. Connect den Knoten mit Ihrem Flow und verknüpfen Sie ihn zwischen dem Dateneingabeknoten und dem MQTT-Publisher.

Eine Anleitung zum Schreiben einer Funktion, die auf Ihre Geschäftsanforderungen zugeschnitten ist, finden Sie in der Node-RED Dokumentation unter [Funktionen schreiben](#)

Konfigurieren Sie den MQTT-Publisher

Nach der Übersetzung sind die Daten bereit für die Veröffentlichung auf dem SiteWise Edge MQTT-Broker.

Konfigurieren Sie den MQTT-Publisher mit diesen Einstellungen, um Daten an den SiteWise Edge MQTT-Broker zu senden:

Um den MQTT-Ausgangsknoten zu importieren

1. Importieren Sie einen ausgehenden MQTT-Konfigurationsknoten mit. `"type": "mqtt out"`
Mit ausgehenden MQTT-Knoten können Sie die Konfiguration eines Brokers gemeinsam nutzen.
2. Geben Sie Schlüssel-Wert-Paare für Informationen ein, die für die MQTT-Broker-Verbindung und das Nachrichten-Routing relevant sind.

Importieren Sie den Beispielknoten. `mqtt out`

Example

```
[
  {
    "id": "string",
    "type": "mqtt out",
    "z": "string",
```

```

    "name": "Publish to MQTT broker",
    "topic": "/Renton/WindFarm/Turbine/WindSpeed",
    "qos": "1",
    "retain": "",
    "respTopic": "",
    "contentType": "",
    "userProps": "",
    "correl": "",
    "expiry": "",
    "broker": "string",
    "x": 830,
    "y": 200,
    "wires": []
  },
  {
    "id": "string",
    "type": "mqtt-broker",
    "name": "emqx",
    "broker": "127.0.0.1",
    "port": "1883",
    "clientId": "",
    "autoConnect": true,
    "usetls": false,
    "protocolVersion": "5",
    "keepalive": 15,
    "cleansession": true,
    "autoUnsubscribe": true,
    "birthTopic": "",
    "birthQos": "0",
    "birthPayload": "",
    "birthMsg": {},
    "closeTopic": "",
    "closePayload": "",
    "closeMsg": {},
    "willTopic": "",
    "willQos": "0",
    "willPayload": "",
    "willMsg": {},
    "userProps": "",
    "sessionExpiry": ""
  }
]

```


Der Beispiel-MQTT-Ausgangsknoten erstellt die MQTT-Verbindung mit den folgenden Informationen:

- Server: 127.0.0.1
- Anschluss: 1883
- Protokoll: MQTT V5

Anschließend konfiguriert der MQTT-Ausgangsknoten das Nachrichten-Routing mit den folgenden Informationen:

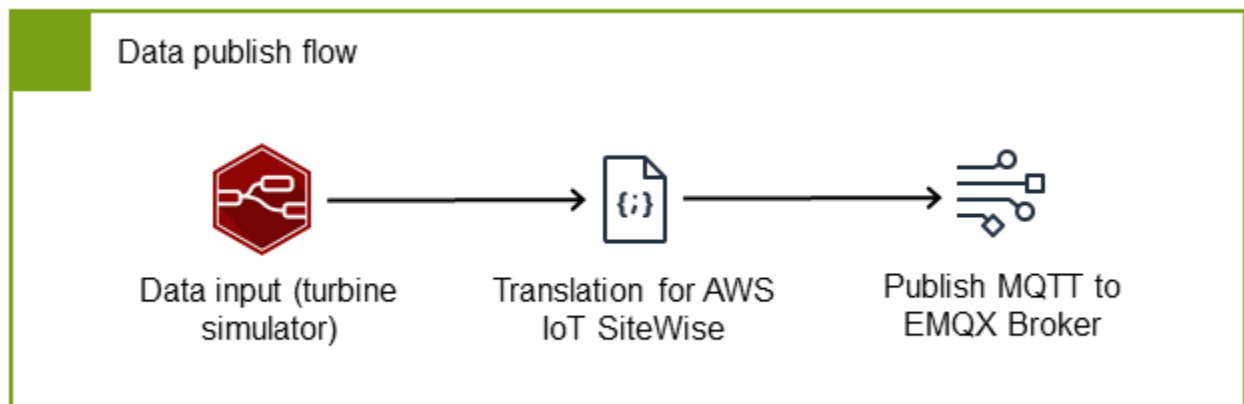
- Topic: /Renton/WindFarm/Turbine/WindSpeed
- QoS: 1

Stellen Sie die Knoten bereit und überprüfen Sie sie

Gehen Sie nach der Konfiguration der drei Knoten für den Datenveröffentlichungsfluss wie folgt vor, um den Flow bereitzustellen und zu überprüfen, ob die Daten korrekt übertragen werden AWS IoT SiteWise

So stellen Sie Verbindungen bereit und überprüfen sie

1. Connect die drei Knoten, wie im Datenveröffentlichungsablauf gezeigt.



2. Wählen Sie Deploy, um alle Änderungen an der Knotenverbindung zu übernehmen.
3. Navigieren Sie zur [AWS IoT SiteWise Konsole](#) und wählen Sie Datenstreams aus.
4. Stellen Sie sicher, dass das Alias-Präfix im Drop-down-Menü ausgewählt ist. Suchen Sie dann nach dem /Renton/WindFarm/Turbine/WindSpeed Alias.

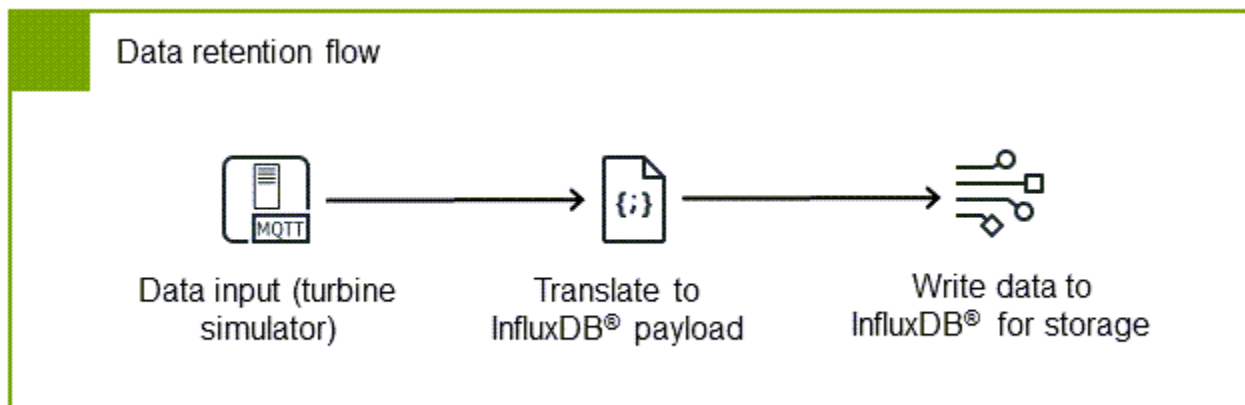
Wenn Sie bei Ihrer Suche den richtigen Alias sehen, haben Sie den Flow bereitgestellt und die Datenübertragung verifiziert.

Konfigurieren Sie den Datenaufbewahrungsablauf

Der Datenaufbewahrungsablauf kann verwendet werden, um die betriebliche Transparenz am Netzwerkrand aufrechtzuerhalten. Dies ist nützlich bei Netzwerkunterbrechungen oder wenn Sie sofortigen Zugriff auf Ihre Daten benötigen. Dieser Flow abonniert den MQTT-Broker für den Empfang von Gerätedaten, konvertiert sie in das InfluxDB®-Format und speichert sie lokal. Durch die Implementierung dieses Flows erstellen Sie einen belastbaren lokalen Datenspeicher, auf den Betreiber ohne Cloud-Abhängigkeiten zugreifen können, was eine Überwachung und Entscheidungsfindung in Echtzeit am Netzwerkrand ermöglicht.

Der Datenfluss besteht aus drei wichtigen Komponenten, die zusammenarbeiten, um sicherzustellen, dass Ihre Daten ordnungsgemäß erfasst und gespeichert werden:

- MQTT-Abonnement-Client — Empfängt Daten vom Broker und stellt so sicher, dass Sie alle relevanten Industriedaten erfassen
- InfluxDB-Übersetzer — Konvertiert AWS IoT SiteWise Nutzdaten in das InfluxDB-Format und bereitet die Daten für eine effiziente Speicherung von Zeitreihen vor
- InfluxDB-Writer — Verwaltet den lokalen Speicher und stellt so die Datenpersistenz und Verfügbarkeit für lokale Anwendungen sicher



Richten Sie den MQTT-Abonnement-Client ein

- Konfigurieren Sie den MQTT-Abonnement-Client in Node-RED so, dass er Daten vom MQTT EMQX-Broker empfängt, indem Sie das folgende Beispiel importieren. AWS IoT SiteWise

Example: MQTT im Knoten

```
[
  {
    "id": "string",
    "type": "mqtt in",
    "z": "string",
    "name": "Subscribe to MQTT broker",
    "topic": "/Renton/WindFarm/Turbine/WindSpeed",
    "qos": "1",
    "datatype": "auto-detect",
    "broker": "string",
    "nl": false,
    "rap": true,
    "rh": 0,
    "inputs": 0,
    "x": 290,
    "y": 340,
    "wires": [
      [
        "string"
      ]
    ]
  },
  {
    "id": "string",
    "type": "mqtt-broker",
    "name": "emqx",
    "broker": "127.0.0.1",
    "port": "1883",
    "clientid": "",
    "autoConnect": true,
    "usetls": false,
    "protocolVersion": "5",
    "keepalive": 15,
    "cleansession": true,
    "autoUnsubscribe": true,
    "birthTopic": "",
    "birthQos": "0",
    "birthPayload": "",
    "birthMsg": {},
    "closeTopic": "",
    "closePayload": "",
  }
]
```

```

    "closeMsg": {},
    "willTopic": "",
    "willQos": "0",
    "willPayload": "",
    "willMsg": {},
    "userProps": "",
    "sessionExpiry": ""
  }
]

```

Dieses Abonnement stellt sicher, dass alle relevanten Daten, die auf dem Broker veröffentlicht werden, für die lokale Speicherung erfasst werden, sodass Sie eine vollständige Aufzeichnung Ihrer industriellen Abläufe erhalten. Der Knoten verwendet dieselben MQTT-Verbindungsparameter wie der [Konfigurieren Sie den MQTT-Publisher](#) Abschnitt mit den folgenden Abbonnementeinstellungen:

- Thema — /Renton/WindFarm/Turbine/WindSpeed
- QoS — 1

Weitere Informationen finden Sie in der Node-RED Dokumentation unter [Connect zu einem MQTT-Broker](#) herstellen.

Konfigurieren Sie den InfluxDB-Übersetzer

[InfluxDB organisiert Daten mithilfe von Tags für die Indizierung und Feldern für Werte.](#) Diese Organisation optimiert die Abfrageleistung und Speichereffizienz für Zeitreihendaten. Importieren Sie den Beispielfunktionsknoten, der JavaScript Code zur Konvertierung der AWS IoT SiteWise Nutzlast in das InfluxDB-Format enthält. Der Übersetzer teilt die Eigenschaften in zwei Gruppen auf:

- Tags — Qualitäts- und Namenseigenschaften für eine effiziente Indexierung
- Felder — Zeitstempel (in Millisekunden seit der Epoche) und Wert

Example: Funktionsknoten für die Übersetzung in eine InfluxDB-Nutzlast

```

[
  {
    "id": "string",
    "type": "function",
    "z": "string",
    "name": "Translate to InfluxDB payload",

```

```

    "func": "let data = msg.payload;\n\nlet timeInSeconds =
data.propertyValues[0].timestamp.timeInSeconds;\nlet offsetInNanos =
data.propertyValues[0].timestamp.offsetInNanos;\nlet timestampInMilliseconds =
(timeInSeconds * 1000) + (offsetInNanos / 1000000);\n\nmsg.payload = [\n    {\n
        \"timestamp(milliseconds_since_epoch)\": timestampInMilliseconds,\n
        \"value\": data.propertyValues[0].value.doubleValue\n    },\n    {\n
        \"name\": data.propertyAlias,\n
        \"quality\": data.propertyValues[0].quality\n    }\n]\n\nreturn msg",
    "outputs": 1,
    "timeout": "",
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 560,
    "y": 340,
    "wires": [
        [
            "string"
        ]
    ]
}
]

```

Zusätzliche Konfigurationsoptionen finden Sie [node-red-contrib-influxdb](#) im Node-RED-Repository. GitHub

Richten Sie den InfluxDB-Writer ein

Der InfluxDB-Writer-Knoten ist die letzte Komponente in Ihrem Datenaufbewahrungsablauf, der für die Speicherung Ihrer Industriedaten in der lokalen InfluxDB-Datenbank verantwortlich ist. Dieser lokale Speicher ist wichtig, um die betriebliche Transparenz bei Netzwerkunterbrechungen aufrechtzuerhalten und den sofortigen Zugriff auf Daten für zeitkritische Anwendungen zu ermöglichen.

1. Installieren Sie das `node-red-contrib-influxdb` Paket über die Option Palette verwalten. Dieses Paket enthält die notwendigen Knoten für die Verbindung von Node-RED mit InfluxDB.
2. Fügen Sie Ihrem Flow einen InfluxDB-Ausgangsknoten hinzu. Dieser Knoten übernimmt das eigentliche Schreiben von Daten in Ihre InfluxDB-Datenbank.
3. Konfigurieren Sie die Servereigenschaften, um eine sichere Verbindung zu Ihrer InfluxDB-Instanz herzustellen:

- a. Version auf 2.0 setzen — Dies gibt an, dass Sie eine Verbindung zu InfluxDB v2.x herstellen, die eine andere API als frühere Versionen verwendet
 - b. URL setzen auf `http://127.0.0.1:8086` — Dies verweist auf Ihre lokale InfluxDB-Instanz
 - c. Geben Sie Ihr InfluxDB-Authentifizierungstoken ein. Dieses sichere Token autorisiert die Verbindung zu Ihrer Datenbank. Sie haben das Token während des [Richten Sie lokalen Speicher mit InfluxDB ein](#) Verfahrens generiert.
4. Geben Sie die Speicherortparameter an, um zu definieren, wo und wie Ihre Daten gespeichert werden:
- a. Geben Sie den Namen Ihrer InfluxDB-Organisation ein — Die Organisation ist ein Arbeitsbereich für eine Gruppe von Benutzern, zu dem Ihre Buckets und Dashboards gehören. Weitere Informationen finden Sie unter Organisationen [verwalten](#) in der. InfluxData Documentation
 - b. Geben Sie den InfluxDB-Bucket an (zum Beispiel `WindFarmData`) — Der Bucket entspricht einer Datenbank in herkömmlichen Systemen und dient als Container für Ihre Zeitreihendaten
 - c. Legen Sie die InfluxDB-Messung fest (zum Beispiel `TurbineData`) — Die Messung ähnelt einer Tabelle in relationalen Datenbanken, in der verwandte Datenpunkte organisiert werden

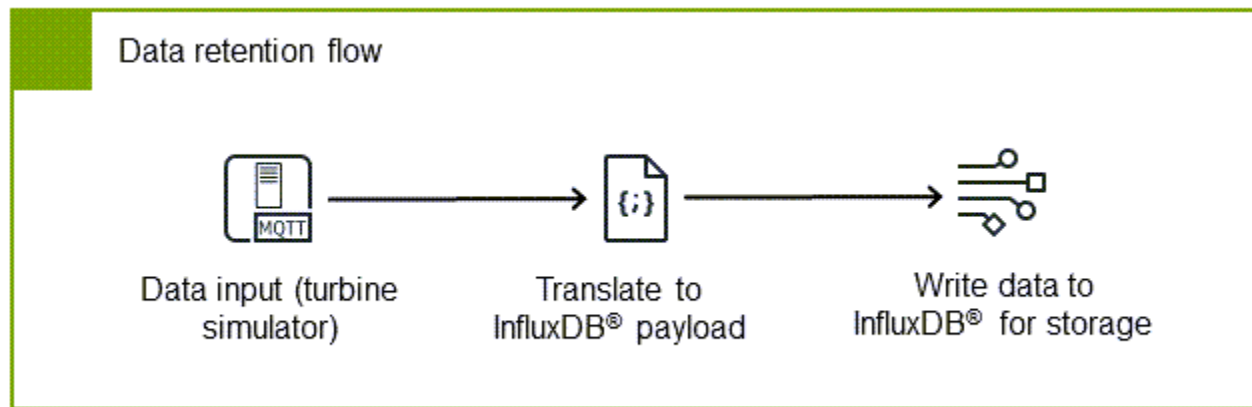
 Note

Suchen Sie den Namen Ihrer Organisation in der linken Seitenleiste der InfluxDB-Instanz. Die Organisations-, Bucket- und Messkonzepte sind grundlegend für das Datenorganisationsmodell von InfluxDB. Weitere Informationen finden Sie in der [InfluxDB-Dokumentation](#).

Stellen Sie den Aufbewahrungsablauf bereit und überprüfen Sie ihn

Nachdem Sie alle Komponenten des Datenaufbewahrungsflusses konfiguriert haben, müssen Sie das System bereitstellen und überprüfen, ob es ordnungsgemäß funktioniert. Diese Überprüfung stellt sicher, dass Ihre Industriedaten ordnungsgemäß lokal gespeichert werden, sodass Sie sofort darauf zugreifen und sie analysieren können.

1. Connect die drei Knoten wie im Flussdiagramm zur Datenspeicherung gezeigt. Dadurch entsteht eine vollständige Pipeline vom Datenabonnement bis zum lokalen Speicher.



2. Wählen Sie Deploy, um Ihre Änderungen zu übernehmen und den Ablauf zu aktivieren. Dadurch wird der Datenerfassungs- und Speicherprozess gestartet.
3. Verwenden Sie den InfluxDB Data Explorer, um Ihre Daten abzufragen und zu visualisieren. Mit diesem Tool können Sie überprüfen, ob die Daten ordnungsgemäß gespeichert werden, und erste Visualisierungen Ihrer Zeitreihendaten erstellen.

Im Datenexplorer sollten Sie sehen können, wie Ihre Windgeschwindigkeitsmessungen im Laufe der Zeit aufgezeichnet werden, sodass bestätigt werden kann, dass die gesamte Pipeline von der Datengenerierung bis zur lokalen Speicherung ordnungsgemäß funktioniert.

Weitere Informationen finden Sie unter [Abfrage im Daten-Explorer](#) im InfluxData Documentation.

Da sowohl der Datenveröffentlichungsfluss als auch der Datenaufbewahrungsfluss implementiert sind, verfügen Sie jetzt über ein vollständiges System, das Daten an die AWS IoT SiteWise Cloud sendet und gleichzeitig eine lokale Kopie beibehält, um sofortigen Zugriff und Stabilität zu gewährleisten. Dieser Dual-Path-Ansatz stellt sicher, dass Sie die Vorteile cloudbasierter Analysen und Speicher nutzen und gleichzeitig die betriebliche Transparenz am Netzwerkrand wahren.

Richten Sie Grafana für Edge ein SiteWise

Mit Grafana® können Sie lokale Echtzeit-Überwachungs-Dashboards für Ihre Industriedaten erstellen. Durch die Visualisierung der in InfluxDB® gespeicherten Daten können Sie den Bedienern sofortige Einblicke in die Geräteleistung, Prozesseffizienz und potenzielle Probleme geben. Diese Transparenz am Netzwerkrand ist wichtig für zeitkritische Abläufe und die Aufrechterhaltung der Kontinuität bei Netzwerkunterbrechungen.

Konfigurieren Sie die Datenquelle

Durch die Verbindung von Grafana mit Ihrer InfluxDB-Datenbank entsteht eine leistungsstarke Visualisierungsebene für Ihre Industriedaten. Diese Verbindung ermöglicht Dashboards zur Echtzeitüberwachung, mit denen Betreiber fundierte Entscheidungen ohne Cloud-Abhängigkeiten treffen können.

1. Greifen Sie lokal auf Ihre Grafana-Instanz zu, indem Sie in Ihrem [Browser zu `http://127.0.0.1:3000`](#) navigieren. Wenn die Aktivierung von TLS erforderlich ist, finden Sie in der Grafana LabsDokumentation Informationen unter [Grafana HTTPS für sicheren Webverkehr einrichten](#).
2. Fügen Sie eine InfluxDB-Datenquelle hinzu, die auf den InfluxDB-Zeitreihen-Bucket verweist, in den Node-RED Daten schreibt. Beispiel, WindFarmData. Diese Verbindung stellt die Verbindung zwischen Ihren gespeicherten Daten und der Visualisierungsplattform her.
3. Detaillierte Anweisungen finden [Sie unter Konfiguration der InfluxDB-Datenquelle](#) in der Grafana Labs Dokumentation.

Erstellen Sie ein Grafana-Dashboard für SiteWise Edge-Daten

Die Erstellung eines Dashboards ist der letzte Schritt beim Aufbau Ihrer lokalen Überwachungslösung. Dashboards bieten visuelle Darstellungen Ihrer Industriedaten, sodass Trends, Anomalien und potenzielle Probleme auf einen Blick leichter erkannt werden können.

- Folgen Sie der Anleitung, um ein Dashboard zu erstellen. Weitere Informationen finden Sie in der Grafana LabsDokumentation unter [Erstellen Sie Ihr erstes Dashboard](#). Bei dieser Vorlage wird davon ausgegangen, dass Ihr Bucket benannt ist WindFarmData und die MessungTurbineData.

Sie können auch die Schnellstartanleitung verwenden, indem Sie die mitgelieferte Beispiel-Dashboard-Vorlage importieren, um schnell ein Dashboard mit einem Zeitreihendiagramm für die Daten zu erstellen, die Node-RED im vorherigen Abschnitt generiert hat. Diese Vorlage bietet einen Ausgangspunkt, den Sie an Ihre spezifischen Überwachungsanforderungen anpassen können.

```
{
  "__inputs": [
    {
      "name": "DS_WINDFARM-DEMO",
```



```
    "label": "windfarm-demo",
    "description": "",
    "type": "datasource",
    "pluginId": "influxdb",
    "pluginName": "InfluxDB"
  }
],
"__elements": {},
"__requires": [
  {
    "type": "grafana",
    "id": "grafana",
    "name": "Grafana",
    "version": "11.6.0-pre"
  },
  {
    "type": "datasource",
    "id": "influxdb",
    "name": "InfluxDB",
    "version": "1.0.0"
  },
  {
    "type": "panel",
    "id": "timeseries",
    "name": "Time series",
    "version": ""
  }
],
"annotations": {
  "list": [
    {
      "builtIn": 1,
      "datasource": {
        "type": "grafana",
        "uid": "-- Grafana --"
      },
      "enable": true,
      "hide": true,
      "iconColor": "rgba(0, 211, 255, 1)",
      "name": "Annotations & Alerts",
      "type": "dashboard"
    }
  ]
},
```

```
"editable": true,
"fiscalYearStartMonth": 0,
"graphTooltip": 0,
"id": null,
"links": [],
"panels": [
  {
    "datasource": {
      "type": "influxdb",
      "uid": "${DS_WINDFARM-DEMO}"
    },
    "fieldConfig": {
      "defaults": {
        "color": {
          "mode": "palette-classic"
        },
        "custom": {
          "axisBorderShow": false,
          "axisCenteredZero": false,
          "axisColorMode": "text",
          "axisLabel": "",
          "axisPlacement": "auto",
          "barAlignment": 0,
          "barWidthFactor": 0.6,
          "drawStyle": "line",
          "fillOpacity": 0,
          "gradientMode": "none",
          "hideFrom": {
            "legend": false,
            "tooltip": false,
            "viz": false
          },
          "insertNulls": false,
          "lineInterpolation": "linear",
          "lineWidth": 1,
          "pointSize": 5,
          "scaleDistribution": {
            "type": "linear"
          },
          "showPoints": "auto",
          "spanNulls": false,
          "stacking": {
            "group": "A",
            "mode": "none"
          }
        }
      }
    }
  }
]
```

```
    },
    "thresholdsStyle": {
      "mode": "off"
    }
  },
  "mappings": [],
  "thresholds": {
    "mode": "absolute",
    "steps": [
      {
        "color": "green"
      },
      {
        "color": "red",
        "value": 80
      }
    ]
  }
},
"overrides": []
},
"gridPos": {
  "h": 8,
  "w": 12,
  "x": 0,
  "y": 0
},
"id": 1,
"options": {
  "legend": {
    "calcs": [],
    "displayMode": "list",
    "placement": "bottom",
    "showLegend": true
  },
  "tooltip": {
    "hideZeros": false,
    "mode": "single",
    "sort": "none"
  }
},
"pluginVersion": "11.6.0-pre",
"targets": [
  {
```

```

      "datasource": {
        "type": "influxdb",
        "uid": "${DS_WINDFARM-DEMO}"
      },
      "query": "from(bucket: \"WindFarmData\")\n  |> range(start:
v.timeRangeStart, stop: v.timeRangeStop)\n  |> filter(fn: (r) => r[\"_measurement
\"] == \"TurbineData\")\n  |> filter(fn: (r) => r[\"_field\"] == \"value\")\n
  |> filter(fn: (r) => r[\"name\"] == \"/Renton/WindFarm/Turbine/WindSpeed\")\n
  |> filter(fn: (r) => r[\"quality\"] == \"GOOD\")\n  |> aggregateWindow(every:
v.windowPeriod, fn: mean, createEmpty: false)\n  |> yield(name: \"mean\")",
      "refId": "A"
    }
  ],
  "title": "Panel Title",
  "type": "timeseries"
}
],
"schemaVersion": 41,
"tags": [],
"templating": {
  "list": []
},
"time": {
  "from": "now-6h",
  "to": "now"
},
"timepicker": {},
"timezone": "browser",
"title": "demo dashboard",
"uid": "fejc0t08o6d4wb",
"version": 1,
"weekStart": ""
}

```

Richten Sie Open-Source-Integrationen mit Docker (Linux) ein

Für einen optimierten Bereitstellungsprozess können Sie Docker verwenden, um Node-RED®, InfluxDB® und Grafana® in einer Linux-Umgebung einzurichten. Diese Methode verwendet vorkonfigurierte Container, was eine schnelle Bereitstellung und eine einfachere Verwaltung der Komponenten ermöglicht.

Voraussetzungen für das Docker-Setup

Bevor Sie beginnen, stellen Sie sicher, dass die folgenden Voraussetzungen erfüllt sind:

- Ein MQTT-fähiges V3-Gateway. Weitere Informationen finden Sie unter [MQTT-enabled, V3-Gateways für AWS IoT SiteWise Edge](#).
- Das Docker Compose-Plugin. Installationsschritte finden Sie unter [Installieren des Docker Compose Plug-ins in der Dokumentation](#) zu den Docker-Handbüchern.

Stellen Sie die Dienste bereit

Bei dieser Bereitstellung werden SiteWise Edge, InfluxDB, Node-RED und Grafana auf demselben Host ausgeführt.

Richten Sie die Umgebung ein

1. Verschaffen Sie sich Root-Zugriff:

```
sudo -i
```

2. Erstellen Sie eine .env-Datei oder exportieren Sie diese Umgebungsvariablen:

```
export INFLUXDB_PASSWORD=your-secure-influxdb-password  
export INFLUXDB_TOKEN=your-secure-influxdb-token  
export GRAFANA_PASSWORD=your-secure-grafana-password
```

Konfigurieren Sie das Docker-Netzwerk

- Erstellen Sie ein Bridge-Netzwerk mit dem Namen SiteWiseEdgeNodeRedDemoNetwork.

```
docker network create --driver=bridge SiteWiseEdgeNodeRedDemoNetwork
```

Bereiten Sie die Docker Compose Datei vor

Kopieren Sie den Inhalt der folgenden YAML-Datei auf Ihr SiteWise Edge-Gateway-Gerät.

Erweitern Sie, um das Beispiel für die Docker Compose YAML-Datei anzuzeigen

```
services:
```

```
influxdb:
  image: influxdb:latest
  container_name: influxdb
  ports:
    - "127.0.0.1:8086:8086"
  volumes:
    - influxdb-storage:/influxdbv2
  environment:
    - DOCKER_INFLUXDB_INIT_MODE=setup
    - DOCKER_INFLUXDB_INIT_USERNAME=admin
    - DOCKER_INFLUXDB_INIT_PASSWORD=${INFLUXDB_PASSWORD}
    - DOCKER_INFLUXDB_INIT_ORG=iot-sitewise-edge
    - DOCKER_INFLUXDB_INIT_BUCKET=WindFarmData
    - DOCKER_INFLUXDB_INIT_RETENTION=0
    - DOCKER_INFLUXDB_INIT_ADMIN_TOKEN=${INFLUXDB_TOKEN}
  networks:
    - SiteWiseEdgeNodeRedDemoNetwork
  restart: unless-stopped

grafana:
  image: grafana/grafana:latest
  container_name: grafana
  ports:
    - "127.0.0.1:3000:3000"
  volumes:
    - grafana-storage:/var/lib/grafana
    - ./grafana/provisioning:/etc/grafana/provisioning
  environment:
    - GF_SECURITY_ADMIN_USER=admin
    - GF_SECURITY_ADMIN_PASSWORD=${GRAFANA_PASSWORD}
    - GF_INSTALL_PLUGINS=grafana-clock-panel,grafana-simple-json-datasource
    - GF_PATHS_PROVISIONING=/etc/grafana/provisioning
    - GF_PATHS_CONFIG=/etc/grafana/grafana.ini
    - GF_LOG_LEVEL=info
  configs:
    - source: grafana_datasource
      target: /etc/grafana/provisioning/datasources/influxdb.yaml
    - source: grafana_preload_dashboard_config
      target: /etc/grafana/provisioning/dashboards/dashboard.yaml
    - source: grafana_preload_dashboard
      target: /etc/grafana/provisioning/dashboards/demo_dashboard.json
  depends_on:
    - influxdb
  networks:
```

```
- SiteWiseEdgeNodeRedDemoNetwork
restart: unless-stopped

nodered:
  build:
    context: .
    dockerfile_inline: |
      FROM nodered/node-red:latest
      RUN npm install node-red-contrib-influxdb
  container_name: nodered
  ports:
    - "127.0.0.1:1880:1880"
  volumes:
    - node_red_data:/data
  environment:
    - NODE_RED_ENABLE_SAFE_MODE=false
    - NODE_RED_ENABLE_PALETTE_EDIT=true
    - NODE_RED_AUTO_INSTALL_MODULES=true
  configs:
    - source: nodered_flows
      target: /data/flows.json
    - source: nodered_settings
      target: /data/settings.js
    - source: nodered_flows_cred
      target: /data/flows_cred.json
  depends_on:
    - influxdb
  networks:
    - SiteWiseEdgeNodeRedDemoNetwork
restart: unless-stopped

volumes:
  influxdb-storage:
  grafana-storage:
  node_red_data:

networks:
  SiteWiseEdgeNodeRedDemoNetwork:
    external: true

configs:
  grafana_datasource:
    content: |
      apiVersion: 1
```

```
datasources:
  - name: windfarm-demo
    type: influxdb
    access: proxy
    url: http://influxdb:8086
    jsonData:
      version: Flux
      organization: iot-sitewise-edge
      defaultBucket: WindFarmData
      tlsSkipVerify: true
    secureJsonData:
      token: ${INFLUXDB_TOKEN}
    editable: false

grafana_preload_dashboard_config:
  content: |
    apiVersion: 1
    providers:
      - name: "Dashboard provider"
        orgId: 1
        type: file
        options:
          path: /etc/grafana/provisioning/dashboards

grafana_preload_dashboard:
  content: |
    {
      "annotations": {
        "list": [
          {
            "builtIn": 1,
            "datasource": {
              "type": "grafana",
              "uid": "-- Grafana --"
            },
            "enable": true,
            "hide": true,
            "iconColor": "rgba(0, 211, 255, 1)",
            "name": "Annotations & Alerts",
            "type": "dashboard"
          }
        ]
      },
      "editable": true,
```



```
"fiscalYearStartMonth": 0,
"graphTooltip": 0,
"id": 1,
"links": [],
"panels": [
  {
    "datasource": {
      "type": "influxdb",
      "uid": "PEB0DCBF338B3CEB2"
    },
    "fieldConfig": {
      "defaults": {
        "color": {
          "mode": "palette-classic"
        },
        "custom": {
          "axisBorderShow": false,
          "axisCenteredZero": false,
          "axisColorMode": "text",
          "axisLabel": "",
          "axisPlacement": "auto",
          "barAlignment": 0,
          "barWidthFactor": 0.6,
          "drawStyle": "line",
          "fillOpacity": 0,
          "gradientMode": "none",
          "hideFrom": {
            "legend": false,
            "tooltip": false,
            "viz": false
          },
          "insertNulls": false,
          "lineInterpolation": "linear",
          "lineWidth": 1,
          "pointSize": 5,
          "scaleDistribution": {
            "type": "linear"
          },
          "showPoints": "auto",
          "spanNulls": false,
          "stacking": {
            "group": "A",
            "mode": "none"
          }
        }
      }
    }
  },

```

```
        "thresholdsStyle": {
            "mode": "off"
        },
    },
    "mappings": [],
    "thresholds": {
        "mode": "absolute",
        "steps": [
            {
                "color": "green"
            },
            {
                "color": "red",
                "value": 80
            }
        ]
    },
    "overrides": []
},
"gridPos": {
    "h": 8,
    "w": 12,
    "x": 0,
    "y": 0
},
"id": 1,
"options": {
    "legend": {
        "calcs": [],
        "displayMode": "list",
        "placement": "bottom",
        "showLegend": true
    },
    "tooltip": {
        "hideZeros": false,
        "mode": "single",
        "sort": "none"
    }
},
"pluginVersion": "11.6.0",
"targets": [
    {
        "datasource": {
```

```

        "type": "influxdb",
        "uid": "PEB0DCBF338B3CEB2"
    },
    "query": "from(bucket: \"WindFarmData\")\n  |> range(start:
v.timeRangeStart, stop: v.timeRangeStop)\n  |> filter(fn: (r) => r[\"_measurement
\"] == \"TurbineData\")\n  |> filter(fn: (r) => r[\"_field\"] == \"value\")\n
  |> filter(fn: (r) => r[\"name\"] == \"/Renton/WindFarm/Turbine/WindSpeed\")\n
  |> filter(fn: (r) => r[\"quality\"] == \"GOOD\")\n  |> aggregateWindow(every:
v.windowPeriod, fn: mean, createEmpty: false)\n  |> yield(name: \"mean\")",
    "refId": "A"
  }
],
"title": "Wind Speed",
"type": "timeseries"
}
],
"preload": false,
"schemaVersion": 41,
"tags": [],
"templating": {
  "list": []
},
"time": {
  "from": "now-6h",
  "to": "now"
},
"timepicker": {},
"timezone": "browser",
"title": "Demo Dashboard",
"uid": "eejtureqjo9a8c",
"version": 2
}

nodered_flows:
content: |
[
  {
    "id": "95fce448fdd43b47",
    "type": "tab",
    "label": "Demo Flow",
    "disabled": false,
    "info": ""
  },
  {

```

```

    "id": "5f63740b66af3386",
    "type": "mqtt out",
    "z": "95fce448fdd43b47",
    "name": "Publish to MQTT broker",
    "topic": "/Renton/WindFarm/Turbine/WindSpeed",
    "qos": "1",
    "retain": "",
    "respTopic": "",
    "contentType": "",
    "userProps": "",
    "correl": "",
    "expiry": "",
    "broker": "5744207557fa19be",
    "x": 830,
    "y": 200,
    "wires": []
  },
  {
    "id": "8f2eb590d596679b",
    "type": "function",
    "z": "95fce448fdd43b47",
    "name": "Translate to SiteWise payload",
    "func": "let input = msg.payload;\nlet output = {};\n\noutput[\"propertyAlias\n\"] = input.name;\n\nlet propertyVal = {}\n\nlet timeInSeconds =
  Math.floor(input.timestamp / 1000);\nlet offsetInNanos = (input.timestamp % 1000) *
  1000000;\n\npropertyVal[\"timestamp\"] = {\n  \"timeInSeconds\": timeInSeconds,\n  \"offsetInNanos\": offsetInNanos,\n};\n\npropertyVal[\"quality\"] = input.quality\n\nlet typeNameConverter = {\n  \"number\": (x) => Number.isInteger(x) ?
  \"integerValue\" : \"doubleValue\", \n  \"boolean\": (x) => \"booleanValue\", \n
  \"string\": (x) => \"stringValue\", \n}\n\nlet typeName = typeNameConverter[typeof
  input.value](input.value)\npropertyVal[\"value\"] = {}\npropertyVal[\"value\"]
[typeName] = input.value;\n\noutput[\"propertyValues\"] = [propertyVal]\n\nreturn {\n
  payload: JSON.stringify(output)\n};",
    "outputs": 1,
    "timeout": "",
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 530,
    "y": 200,
    "wires": [
      [
        "5f63740b66af3386"
      ]
    ]
  }
]

```

```

    ]
  ]
},
{
  "id": "4b78cbdea5e3258c",
  "type": "inject",
  "z": "95fce448fdd43b47",
  "name": "Turbine Simulator",
  "props": [
    {
      "p": "payload.timestamp",
      "v": "",
      "vt": "date"
    },
    {
      "p": "payload.quality",
      "v": "GOOD",
      "vt": "str"
    },
    {
      "p": "payload.value",
      "v": "$$random()",
      "vt": "jsonata"
    },
    {
      "p": "payload.name",
      "v": "/Renton/WindFarm/Turbine/WindSpeed",
      "vt": "str"
    }
  ],
  "repeat": "1",
  "crontab": "",
  "once": false,
  "onceDelay": "",
  "topic": "",
  "x": 270,
  "y": 200,
  "wires": [
    [
      "8f2eb590d596679b"
    ]
  ]
},
{

```

```

    "id": "b658bf337ea2e316",
    "type": "influxdb out",
    "z": "95fce448fdd43b47",
    "influxdb": "2f1c38495035d2e4",
    "name": "Store data in InfluxDB",
    "measurement": "TurbineData",
    "precision": "",
    "retentionPolicy": "",
    "database": "",
    "retentionPolicyV18Flux": "",
    "org": "iot-sitewise-edge",
    "bucket": "WindFarmData",
    "x": 840,
    "y": 340,
    "wires": []
  },
  {
    "id": "9432d39af35b202f",
    "type": "function",
    "z": "95fce448fdd43b47",
    "name": "Translate to InfluxDB payload",
    "func": "let data = msg.payload;\n\nlet timeInSeconds =
data.propertyValues[0].timestamp.timeInSeconds;\nlet offsetInNanos =
data.propertyValues[0].timestamp.offsetInNanos;\nlet timestampInMilliseconds =
(timeInSeconds * 1000) + (offsetInNanos / 1000000);\n\nmsg.payload = [\n  {\n    \"timestamp(milliseconds_since_epoch)\": timestampInMilliseconds,\n    \"value\": data.propertyValues[0].value.doubleValue\n  },\n  {\n    \"name\": data.propertyAlias,\n    \"quality\": data.propertyValues[0].quality\n  }\n]\n\nreturn msg",
    "outputs": 1,
    "timeout": "",
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 560,
    "y": 340,
    "wires": [
      [
        "b658bf337ea2e316"
      ]
    ]
  },
  {

```

```

      "id": "b689403d2c80816b",
      "type": "mqtt in",
      "z": "95fce448fdd43b47",
      "name": "Subscribe to MQTT broker",
      "topic": "/Renton/WindFarm/Turbine/WindSpeed",
      "qos": "1",
      "datatype": "auto-detect",
      "broker": "5744207557fa19be",
      "nl": false,
      "rap": true,
      "rh": 0,
      "inputs": 0,
      "x": 290,
      "y": 340,
      "wires": [
        [
          "9432d39af35b202f"
        ]
      ]
    },
    {
      "id": "4f59bed8e829fc35",
      "type": "comment",
      "z": "95fce448fdd43b47",
      "name": "Data Publish Flow",
      "info": "dfgh",
      "x": 270,
      "y": 160,
      "wires": []
    },
    {
      "id": "b218c7fc58c8b6e7",
      "type": "comment",
      "z": "95fce448fdd43b47",
      "name": "Data Retention flow",
      "info": "",
      "x": 270,
      "y": 300,
      "wires": []
    },
    {
      "id": "5744207557fa19be",
      "type": "mqtt-broker",
      "name": "emqx",

```

```

    "broker": "emqx",
    "port": "1883",
    "clientid": "",
    "autoConnect": true,
    "usetls": false,
    "protocolVersion": "5",
    "keepalive": 15,
    "cleansession": true,
    "autoUnsubscribe": true,
    "birthTopic": "",
    "birthQos": "0",
    "birthPayload": "",
    "birthMsg": {},
    "closeTopic": "",
    "closePayload": "",
    "closeMsg": {},
    "willTopic": "",
    "willQos": "0",
    "willPayload": "",
    "willMsg": {},
    "userProps": "",
    "sessionExpiry": ""
  },
  {
    "id": "2f1c38495035d2e4",
    "type": "influxdb",
    "hostname": "influxdb",
    "port": 8086,
    "protocol": "http",
    "database": "",
    "name": "InfluxDB",
    "usetls": false,
    "tls": "",
    "influxdbVersion": "2.0",
    "url": "http://influxdb:8086",
    "timeout": "",
    "rejectUnauthorized": false
  }
]

```

nodered_flows_cred:

```

content: |
  {
    "2f1c38495035d2e4": {

```



```

        "token": "${INFLUXDB_TOKEN}"
    }
}

nodered_settings:
  content: |
    module.exports = {
      flowFile: 'flows.json',
      credentialSecret: false,
      adminAuth: null,
      editorTheme: {
        projects: {
          enabled: false
        }
      }
    }
  }
}

```

Aktualisieren Sie die SiteWise Edge-Bereitstellung

1. Navigieren Sie zur [AWS IoT Konsole](#)
2. Wählen Sie im linken Navigationsmenü im Bereich Verwalten die Option Greengrass-Geräte und dann Core-Geräte aus.
3. Wählen Sie das mit Ihrem SiteWise Edge-Gateway verbundene Kerngerät aus.
4. Wählen Sie die Registerkarte Bereitstellungen und dann den Wert für die Bereitstellungs-ID aus.
5. Wählen Sie „Aktionen“ und anschließend „Überarbeiten“.
6. Lesen Sie die Popup-Meldung und wählen Sie dann „Bereitstellung überarbeiten“.
7. Wählen Sie in Schritt 2 — Komponenten auswählen die folgenden Komponenten aus und klicken Sie dann auf Weiter.
 - `aws.greengrass.clientdevices.mqtt.EMQX`
 - `aws.iot.SiteWiseEdgePublisher`
8. Wählen Sie in Schritt 3 — Komponenten konfigurieren den `aws.greengrass.clientdevices.mqtt.EMQX` Komponentenwert aus und fügen Sie die folgende Netzwerkkonfiguration hinzu:

```

{
  "emqxConfig": {
    "authorization": {

```

```

        "no_match": "allow"
      },
      "listeners": {
        "tcp": {
          "default": {
            "enabled": true,
            "enable_authn": false
          }
        }
      }
    },
    "authMode": "bypass",
    "dockerOptions": "-p 127.0.0.1:1883:1883 --
network=SiteWiseEdgeNodeRedDemoNetwork",
    "requiresPrivilege": "true"
  }
}

```

9. Wählen Sie Weiter aus.
10. Wählen Sie in Schritt 4 — Erweiterte Einstellungen konfigurieren die Option Weiter.
11. Wählen Sie Bereitstellen

Starten Sie die Dienste

1. Starten Sie die Dienste mit der Docker Compose-Datei. Führen Sie den folgenden Befehl in dem Verzeichnis aus, das die `compose.yaml` Datei enthält.

```
docker compose up -d
```

2. Erstellen Sie einen SSH-Tunnel für den Zugriff auf die Dienste:

```
ssh -i path_to_your_ssh_key -L 1880:127.0.0.1:1880 -L 3000:127.0.0.1:3000 -L
8086:127.0.0.1:8086 username@gateway_ip_address
```

Diese Bereitstellung erstellt die folgenden Dienste in: `SiteWiseEdgeNodeRedDemoNetwork` network

InfluxDB v2 (Port 8086)

Beinhaltet vorkonfigurierte Organisations- (iot-sitewise-edge), WindFarmData InfluxDB-Bucket und Administratoranmeldedaten

Node-RED (Port 1880)

Beinhaltet InfluxDB-Knoten und vorkonfigurierte Flows für die Integration AWS IoT SiteWise

Grafana (Hafen 3000)

Beinhaltet einen Admin-Benutzer, eine InfluxDB-Datenquelle und ein Überwachungs-Dashboard

Greifen Sie auf die Dienste zu

Greifen Sie nach der Bereitstellung mit den folgenden URLs Anmeldeinformationen auf die Dienste zu:

Note

Sie können von Ihrem Host oder dem Gateway-Computer aus auf jeden Dienst zugreifen.

Details zum Zugriff auf den Dienst

Service	URL	Anmeldeinformationen
Node-RED	http://127.0.0.1:1880	Keine Anmeldeinformationen erforderlich
InfluxDB	http://127.0.0.1:8086	Nutzername: admin Passwort: \$INFLUXDB_PASSWORD
Grafana	http://127.0.0.1:3000	Nutzername: admin Passwort: \$GRAFANA_PASSWORD

Überprüfen der Bereitstellung

Um sicherzustellen, dass Ihre Bereitstellung erfolgreich ist, führen Sie die folgenden Prüfungen durch:

1. Stellen Sie für Node-RED sicher, dass zwei vorinstallierte Flows vorhanden sind:

- Ablauf der Datenveröffentlichung
 - Ablauf der Datenspeicherung
2. Bestätigen Sie in der AWS IoT SiteWise Konsole das Vorhandensein eines Datenstroms mit dem Alias/Renton/WindFarm/Turbine/WindSpeed. AWS IoT SiteWise
 3. Verwenden Sie für InfluxDB den Data Explorer, um die Datenspeicherung in der TurbineData Messung innerhalb des WindFarmData Buckets zu überprüfen.
 4. Sehen Sie sich für Grafana das Dashboard an, um die Anzeige der von Node-RED generierten Zeitreihendaten zu bestätigen.

Prozessdaten für Open-Source-Integrationen

Die Daten können in verschiedenen Phasen mithilfe verschiedener Tools verarbeitet werden (z. B. Transformation oder Aggregation), die jeweils unterschiedliche Überwachungsanforderungen erfüllen.

Verarbeiten Sie Daten mit Node-RED-Knoten

Transformieren Sie Ihre Daten in Echtzeit mithilfe der integrierten Node-RED® -Verarbeitungsknoten. Konfigurieren Sie diese Knoten über die Node-RED-Konsole, um Ihre Datenpipeline zu erstellen.

Knoten für die Datentransformation

Transformieren Sie einzelne Datenpunkte, ähnlich wie Transforms in AWS IoT SiteWise, mithilfe dieser Knoten:

- Knoten ändern — Führt einfache Wertänderungen an Ihren Daten durch.
- Funktionsknoten — Ermöglicht benutzerdefinierte JavaScript Transformationen für die komplexe Datenverarbeitung.

Knoten zur Berechnung von Metriken

Kombinieren Sie mehrere Datenpunkte zu einer einzigen Ausgabe, ähnlich wie bei Metrics in AWS IoT SiteWise, und verwenden Sie dabei die folgenden Knoten:

- Batch-Knoten — Gruppiert mehrere Nachrichten für die Batch-Verarbeitung.
- Join Node — Kombiniert mehrere Datenströme zu einer einzigen Ausgabe.
- Aggregatorknoten — Berechnet aggregierte Metriken aus mehreren Datenpunkten.

Weitere Knotenoptionen finden Sie in der [Node-RED-Bibliothek](#).

Erstellen Sie InfluxDB-Aufgaben

Während Node-RED sich durch grundlegende Datenverarbeitung mit schneller Einrichtung auszeichnet, können komplexe metrische Berechnungen bei der flussbasierten Programmierung zu einer Herausforderung werden. InfluxDB® Tasks bieten eine Alternative in Form von geplanten Flux-Skripten für fortgeschrittene Verarbeitungsanforderungen.

Verwenden Sie InfluxDB-Aufgaben für:

- Statistische Aggregationen großer Datensätze
- Mathematische Operationen an mehreren Eigenschaften
- Abgeleitete Messungen aus mehreren Quellen

Funktionen der Aufgabe

- Geplante Ausführung — Führen Sie Aufgaben auf der Grundlage von Cron-Ausdrücken aus
- Stapelverarbeitung — Optimieren Sie die Abläufe für Zeitreihendaten
- Fehlerbehebung — Fehlgeschlagene Operationen werden automatisch wiederholt
- Überwachung — Verfolgen Sie die Ausführung anhand detaillierter Protokolle

Verwalten Sie Aufgaben über die InfluxDB-Benutzeroberfläche, API oder CLI. Weitere Informationen finden Sie unter [Daten mit InfluxDB-Aufgaben verarbeiten](#).

Verwenden Sie Grafana-Transformationen

Transformieren Sie die Datenvisualisierung in Grafana®, ohne die Quelldaten in InfluxDB zu ändern. Grafana-Transformationen gelten nur für die Visualisierungsebene.

- Visual Builder - Erstellen Sie Transformationen, ohne Code zu schreiben
- Live-Vorschau - Transformationsergebnisse in Echtzeit anzeigen
- Multisource — Verarbeiten Sie Daten aus mehreren Datenbankquellen
- Speichereffizient — Transformieren Sie Daten zum Zeitpunkt der Visualisierung, ohne Zwischenergebnisse zu speichern

Weitere Informationen finden Sie unter [Daten transformieren](#).

Fehlerbehebung bei Open-Source-Integrationen

Weitere Informationen zur Problembehandlung im Zusammenhang mit Open-Source-Integrationen für SiteWise Edge-Gateways finden Sie unter [Fehlerbehebung bei Open-Source-Integrationen am Edge](#)

Klassische Streams, V2-Gateways fürAWS IoT SiteWiseEdge

Machen Sie sich mit den Funktionen und Einschränkungen von Classic Streams, V2-Gateways für AWS IoT SiteWise Edge vertraut.

Das Classic Streams, V2-Gateway behält die traditionelle Funktionalität bei, die aus früheren AWS IoT SiteWise Implementierungen vor der Einführung von MQTT-enabled V3-Gateways bekannt war. Diese SiteWise Edge-Gateways werden als klassische Streams und V2-Gateways betrachtet. Sie behalten die Abwärtskompatibilität bei und sind mit dem Datenverarbeitungspaket funktionsfähig. Das Classic-Streaming-Gateway bietet zwar zuverlässige Leistung für bestehende Setups, weist jedoch im Vergleich zu neueren Gateway-Optionen Einschränkungen auf. Insbesondere ist dieser Gateway-Typ nicht vollständig mit den erweiterten Funktionen kompatibel MQTT-enabled, die im V3-Gateway-Ziel verfügbar sind. Um das MQTT-Messaging-Protokoll zu verwenden, können Sie ein neues MQTT-enabled V3-Gateway erstellen. Weitere Informationen finden Sie unter [MQTT-enabled, V3-Gateways fürAWS IoT SiteWiseEdge](#).

Themen

- [Verwenden Sie Pakete, um Daten in SiteWise Edge zu sammeln und zu verarbeiten](#)
- [Konfigurieren Sie dieAWS IoT SiteWisePublisher-Komponente](#)
- [Ziele undAWS IoT GreengrassStream-Manager](#)
- [Konfigurieren Sie Edge-Funktionen aufAWS IoT SiteWiseEdge](#)
- [Konfigurieren Sie die Edge-Datenverarbeitung fürAWS IoT SiteWiseModelle und Anlagen](#)

Verwenden Sie Pakete, um Daten in SiteWise Edge zu sammeln und zu verarbeiten

Note

Das Feature Data Processing Pack (DPP) ist für Neukunden nicht mehr verfügbar. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter [Data processing pack availability change](#).

AWS IoT SiteWise Edge-Gateways verwenden unterschiedliche Pakete, um zu bestimmen, wie Ihre Daten erfasst und verarbeitet werden sollen.

Derzeit sind die folgenden Pakete verfügbar:

- **Datenerfassungspaket** — Verwenden Sie dieses Paket, um Ihre Industriedaten zu sammeln und an AWS Cloud-Ziele weiterzuleiten. Standardmäßig ist dieses Paket automatisch für Ihr SiteWise Edge-Gateway aktiviert.
- **Datenverarbeitungspaket** — Verwenden Sie dieses Paket, um die SiteWise Edge-Gateway-Kommunikation mit Edge-konfigurierten Anlagenmodellen und Anlagen zu aktivieren. Sie können mit der Edge-Konfiguration steuern, welche Komponentendaten vor Ort berechnet und verarbeitet werden sollen. Anschließend können Sie Ihre Daten an AWS IoT SiteWise oder andere AWS Dienste senden. Weitere Hinweise zum Datenverarbeitungspaket finden Sie unter [the section called “Konfigurieren Sie die Edge-Datenverarbeitung”](#).

Aktualisieren von Paketen

Important

Ein Upgrade von Datenverarbeitungspaket-Versionen von Versionen vor (und einschließlich) 2.0.x auf Version 2.1.x führt zum Datenverlust lokal gespeicherter Messungen.

SiteWise Edge-Gateways verwenden unterschiedliche Pakete, um zu bestimmen, wie Ihre Daten erfasst und verarbeitet werden sollen. Sie können die AWS IoT SiteWise Konsole verwenden, um Pakete zu aktualisieren.

Um Pakete zu aktualisieren (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie in der linken Navigationsleiste im Abschnitt Edge die Option Edge-Gateways aus.
3. Wählen Sie in der Gateways-Liste das SiteWise Edge-Gateway mit den Paketen aus, die Sie aktualisieren möchten.
4. Wählen Sie im Abschnitt Gateway-Konfiguration die Option Verfügbare Softwareupdates aus.
5. Wählen Sie auf der Seite Softwareversionen bearbeiten die Option Updates aus.

 Note

Sie können nur Pakete aktualisieren, die aktiviert sind. Um die Liste der Pakete zu finden, die für dieses SiteWise Edge-Gateway aktiviert sind, wählen Sie Übersicht und dann den Abschnitt Edge-Funktionen.

6. Gehen Sie auf der Seite Softwareversionen bearbeiten im Abschnitt Gateway-Komponenten-Updates wie folgt vor:
 - Um den OPC UA Collector zu aktualisieren, wählen Sie eine Version und dann Deploy.
 - Um den Publisher zu aktualisieren, wählen Sie eine Version und dann Deploy.
 - Um das Data Processing Pack zu aktualisieren, wählen Sie eine Version und dann Bereitstellen aus.
7. Wenn Sie mit der Bereitstellung neuer Versionen fertig sind, wählen Sie Fertig.

Falls Sie Probleme beim Upgrade der Packs haben, finden Sie weitere Informationen unter [Pakete können nicht für SiteWise Edge-Gateways bereitgestellt werden](#).

Änderung der Verfügbarkeit von Datenverarbeitungspaketen

 Note

Das Feature Data Processing Pack (DPP) ist für Neukunden nicht mehr verfügbar. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter [Data processing pack availability change](#).

Funktionen, die der Funktion des AWS IoT-Datenverarbeitungspakets ähneln, finden Sie entweder in [Open-Source-Alternativen](#) oder in unseren [Partnerintegrationen](#). Das AWS IoT SiteWise Data Processing Pack ist eine Funktion AWS IoT SiteWise, die Datentransformationen, Metriken, Filterung, lokale Speicherung und Visualisierung am Netzwerkrand ermöglicht.

 Note

AWS IoT SiteWise und das AWS IoT SiteWise Edge-Datenerfassungspaket ist weiterhin verfügbar, aber das Datenverarbeitungspaket wechselt in den Wartungsmodus.

Optionen für die Migration

Erkunden Sie diese Migrationsoptionen, um die Funktionalität des Data Processing Pack zu ersetzen.

Open-source Alternativen

Erstellen Sie lokale Datenverarbeitungspipelines Node-RED für die Datentransformation, InfluxDB für die Speicherung von Zeitreihen und Grafana für die Visualisierung. Diese Tools arbeiten mit V3-Gateways über MQTT zusammen MQTT-enabled, um Edge-Verarbeitung und lokale Einblicke zu ermöglichen und gleichzeitig Daten mit der Cloud zu synchronisieren. AWS

Weitere Informationen finden Sie unter [Verarbeiten und Visualisieren von Daten mit SiteWise Edge- und Open-Source-Tools](#).

Partnerintegrationen

Connect Industrieanlagen und Sensoren über Datenquellen von Drittanbietern wie CloudRail EasyEdge, und Litmus Edge. Diese Greengrass-Komponenten wurden in Zusammenarbeit mit AWS über 200 Industrieprotokollen für eine umfassende Datenerfassung und -verarbeitung entwickelt und unterstützen diese.

Weitere Informationen finden [Sie unter Hinzufügen einer Datenquelle](#).

MQTT-enabled, V3-Gateways

MQTT-enabled, V3-Gateways verwenden das einfache MQTT-Protokoll für eine effiziente Datenkommunikation und bieten flexible Datenziele, einschließlich Echtzeitaufnahme und gepufferte Amazon S3 S3-Erfassung. Sie können Pfadfilter für eine präzise Datenerfassung implementieren und von verbesserter Skalierbarkeit und Ausrichtung der IoT-Standards profitieren. MQTT-enabled, V3-Gateways bieten Cloud-basierte Datenverarbeitung über AWS IoT SiteWise zentrale Dienste wie Anlagenmodelle, berechnete Eigenschaften, Alarme und historische Datenabfragen.

Weitere Informationen finden Sie unter [MQTT-enabled, V3-Gateways fürAWS IoT SiteWiseEdge](#).

Häufig gestellte Fragen

Kann ich schrittweise migrieren?

Ja, Sie können schrittweise mit einer beliebigen Kombination der Migrationsoptionen migrieren. Sie können V3-Gateways MQTT-enabled, Open-Source-Alternativen oder Partnerintegrationen neben

vorhandenen Classic-Streams und V2-Gateways mit dem Datenverarbeitungspaket bereitstellen. Alle Optionen können Daten an dieselbe Umgebung senden. AWS IoT SiteWise

Wie lange kann ich das Datenverarbeitungspaket weiter verwenden?

Das Datenverarbeitungspaket bleibt für Bestandskunden im Wartungsmodus verfügbar. Sie erhalten eine Vorankündigung, falls Änderungen der Verfügbarkeit geplant sind. Überwachen Sie die AWS Serviceankündigungen und die Benachrichtigungen Ihres Kontos im Hinblick auf Aktualisierungen.

Konfigurieren Sie die AWS IoT SiteWise Publisher-Komponente

Nachdem Sie ein AWS IoT SiteWise Edge-Gateway erstellt und die Software installiert haben, können Sie die Publisher-Komponente so einrichten, dass Ihr SiteWise Edge-Gateway Daten in die AWS Cloud exportieren kann. Verwenden Sie die Publisher-Komponente, um zusätzliche Funktionen zu aktivieren oder Standardeinstellungen zu konfigurieren. Weitere Informationen finden Sie unter [AWS IoT SiteWise Publisher](#) im AWS IoT Greengrass Version 2 Entwicklerhandbuch.

Note

Die Publisher-Konfiguration unterscheidet sich je nach verwendetem Gateway-Typ. Verwenden Sie für klassische Stream- und V2-Gateways den `iotsitewise:publisher:2` Namespace. Verwenden Sie für MQTT-enabled V3-Gateways den Namespace `iotsitewise:publisher:3`.

Console

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie das SiteWise Edge-Gateway aus, für das Sie den Publisher konfigurieren möchten.
4. Wählen Sie im Abschnitt Publisher-Konfiguration die Option Bearbeiten
5. Wählen Sie für die Reihenfolge der Veröffentlichung eine der folgenden Optionen aus:
 - Älteste Daten zuerst veröffentlichen — Das SiteWise Edge-Gateway veröffentlicht die ältesten Daten standardmäßig zuerst in der Cloud.
 - Neueste Daten zuerst veröffentlichen — Das SiteWise Edge-Gateway veröffentlicht die neuesten Daten zuerst in der Cloud.

6. (Optional) Wenn Sie nicht möchten, dass das SiteWise Edge-Gateway Ihre Daten komprimiert, deaktivieren Sie die Option Komprimierung beim Hochladen von Daten aktivieren.
7. (Optional) Wenn Sie keine alten Daten veröffentlichen möchten, wählen Sie „Abgelaufene Daten ausschließen“ und gehen Sie wie folgt vor:
 - Geben Sie für den Stichtag einen Wert ein und wählen Sie eine Einheit aus. Die Sperrfrist muss zwischen fünf Minuten und sieben Tagen liegen. Wenn die Sperrfrist beispielsweise drei Tage beträgt, werden Daten, die älter als drei Tage sind, nicht in der Cloud veröffentlicht.
8. (Optional) Um benutzerdefinierte Einstellungen für den Umgang mit Daten auf Ihrem lokalen Gerät festzulegen, wählen Sie Lokale Speichereinstellungen und gehen Sie wie folgt vor:
 - a. Geben Sie für den Aufbewahrungszeitraum eine Zahl ein und wählen Sie eine Einheit aus. Der Aufbewahrungszeitraum muss zwischen einer Minute und 30 Tagen liegen und mindestens dem Rotationszeitraum entsprechen. Wenn die Aufbewahrungsfrist beispielsweise 14 Tage beträgt, löscht das SiteWise Edge-Gateway alle Daten am Edge, die älter als die angegebene Sperrfrist sind, nachdem sie 14 Tage lang gespeichert wurden.
 - b. Geben Sie für den Rotationszeitraum eine Zahl ein und wählen Sie eine Einheit aus. Der Rotationszeitraum muss länger als eine Minute und gleich oder kürzer als der Aufbewahrungszeitraum sein. Angenommen, der Rotationszeitraum beträgt zwei Tage. Das SiteWise Edge-Gateway sammelt Daten, die älter als die Sperrfrist sind, und speichert sie in einer einzigen Datei. Bei selbst gehosteten Gateways überträgt das SiteWise Edge-Gateway alle zwei Tage einen Datenstapel in das folgende lokale Verzeichnis: `AWS IoT Greengrass V2 /greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports`
 - c. Geben Sie für Speicherkapazität einen Wert ein, der größer oder gleich 1 ist. Wenn die Speicherkapazität 2 GB beträgt, beginnt das SiteWise Edge-Gateway mit dem Löschen von Daten, wenn mehr als 2 GB an Daten lokal gespeichert sind.
9. Wählen Sie Speichern.

AWS CLI

Verwenden Sie die [UpdateGatewayCapabilityConfiguration](#) API, um den Herausgeber zu konfigurieren.

Stellen Sie den Parameter `capabilityNamespace` auf `iotsitewise:publisher:2` ein.

Example: Publisher-Konfiguration für Classic Stream, V2-Gateways

Der Herausgeber-Namespace: `iotsitewise:publisher:2`

```
{
  "SiteWisePublisherConfiguration": {
    "publishingOrder": "TIME_ORDER",
    "enableCompression": true,
    "dropPolicy": {
      "cutoffAge": "7d",
      "exportPolicy": {
        "retentionPeriod": "7d",
        "rotationPeriod": "6h",
        "exportSizeLimitGB": 10
      }
    }
  },
  "SiteWiseS3PublisherConfiguration": {
    "accessRoleArn": "arn:aws:iam:123456789012:role/roleName",
    "streamToS3ConfigMapping": [
      {
        "streamName": "S3_OPC-UA_Data_Collector",
        "targetBucketArn": "arn:aws:s3:::amzn-s3-demo-bucket/dataCollector",
        "publishPolicy": {
          "publishFrequency": "10m",
          "localSizeLimitGB": 10
        },
        "siteWiseImportPolicy": {
          "enableSiteWiseStorageImport": true,
          "enableDeleteAfterImport": true
        }
      }
    ]
  }
}
```

Der Herausgeber stellt die folgenden Konfigurationsparameter bereit, die Sie anpassen können:

SiteWisePublisherConfiguration

publishingOrder

Die Reihenfolge, in der Daten in der Cloud veröffentlicht werden. Der Wert dieses Parameters kann einer der folgenden sein:

- `TIME_ORDER`(Älteste Daten zuerst veröffentlichen) — Die frühesten Daten werden standardmäßig zuerst in der Cloud veröffentlicht.
- `RECENT_DATA`(Neueste Daten zuerst veröffentlichen) — Die neuesten Daten werden zuerst in der Cloud veröffentlicht.

enableCompression

Stellen Sie diese Option ein `true`, um Daten vor der Veröffentlichung zu komprimieren. Durch Datenkomprimierung kann die Bandbreitennutzung reduziert werden.

dropPolicy

(Optional) Eine Richtlinie, die steuert, welche Daten in der Cloud veröffentlicht werden.

cutoffAge

Das maximale Alter der zu veröffentlichenden Daten, angegeben in Tagen, Stunden und Minuten. Zum Beispiel `7d` oder `1d7h16m`. Daten, die älter sind als die von Ihnen angegebenen, werden nicht gesendet AWS IoT SiteWise.

Daten, die vor dem Stichtag liegen, werden nicht in der Cloud veröffentlicht. Das Mindestalter muss zwischen fünf Minuten und sieben Tagen liegen.

Sie können, und verwenden `mh`, `d` wenn Sie ein Mindestalter angeben. Hinweis, `m` der Minuten, `h` Stunden und Tage `d` darstellt.

exportPolicy

(Optional) Eine Richtlinie, die die Datenspeicherung am Netzwerkrand verwaltet. Diese Richtlinie gilt für Daten, die vor dem Stichtag liegen.

retentionPeriod

Ihr SiteWise Edge-Gateway löscht alle Daten am Edge, die vor dem Sperrzeitraum liegen, aus dem lokalen Speicher, nachdem sie für den angegebenen Aufbewahrungszeitraum gespeichert wurden. Die Aufbewahrungsdauer muss zwischen einer Minute und 30 Tagen liegen und mindestens dem Rotationszeitraum entsprechen.

Sie können `m`, und verwenden `h`, `d` wenn Sie einen Aufbewahrungszeitraum angeben. Hinweis, `m` der Minuten, `h` Stunden und Tage `d` darstellt.

`rotationPeriod`

Das Zeitintervall, über das Daten, die vor dem Stichtag liegen, gebündelt und in einer einzigen Datei gespeichert werden sollen. Das SiteWise Edge-Gateway überträgt am Ende jeder Rotationsperiode einen Datenstapel in das folgende lokale Verzeichnis: `/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports`. Der Rotationszeitraum muss länger als eine Minute und gleich oder kürzer als der Aufbewahrungszeitraum sein.

Sie können `m`, und verwenden `h`, `d` wenn Sie einen Rotationszeitraum angeben. Hinweis, `m` der Minuten, `h` Stunden und Tage `d` darstellt.

`exportSizeLimitGB`

Die maximal zulässige Größe der lokal gespeicherten Daten in GB. Wenn dieses Kontingent überschritten wird, beginnt das SiteWise Edge-Gateway mit dem Löschen der frühesten Daten, bis die Größe der lokal gespeicherten Daten dem Kontingent entspricht oder darunter liegt. Der Wert dieses Parameters muss größer oder gleich 1 sein.

`SiteWiseS3PublisherConfiguration`

`accessRoleArn`

Die Zugriffsrolle, die die AWS IoT SiteWise Erlaubnis erteilt, den Amazon S3 S3-Bucket zu verwalten, in dem Sie veröffentlichen.

`streamToS3ConfigMapping`

Eine Reihe von Konfigurationen, die einen Stream einer Amazon S3 S3-Konfiguration zuordnen.

`streamName`

Der Stream, aus dem gelesen und in der Amazon S3 S3-Konfiguration veröffentlicht werden soll.

`targetBucketArn`

Der BucketARN, in dem veröffentlicht werden soll.

`publishPolicy`

`publishFrequency`

Die Häufigkeit, mit der das SiteWise Edge-Gateway im Amazon S3 S3-Bucket veröffentlicht.

`localSizeLimitGB`

Die maximale Größe der auf die lokale Festplatte geschriebenen Dateien. Wenn dieser Schwellenwert überschritten wird, veröffentlicht der Herausgeber alle gepufferten Daten an seinem Ziel.

`siteWiseImportPolicy`

`enableSiteWiseStorageImport`

Stellen Sie dies auf `true`, um Daten aus einem Amazon S3 S3-Bucket in den AWS IoT SiteWise Speicher zu importieren.

`enableDeleteAfterImport`

Stellen Sie diese Option auf `true`, um die Datei im Amazon S3 S3-Bucket nach der Aufnahme in den AWS IoT SiteWise Speicher zu löschen.

Ziele und AWS IoT Greengrass Stream-Manager

AWS IoT Greengrass Mit Stream Manager können Sie Daten an die folgenden AWS Cloud Ziele senden: Kanäle in AWS IoT Analytics, Streams in Amazon Kinesis Data Streams, Asset-Eigenschaften in AWS IoT SiteWise oder Objekte in Amazon Simple Storage Service (Amazon S3). Weitere Informationen finden Sie im AWS IoT Greengrass Version 2 Entwicklerhandbuch unter [Datenstreams auf dem AWS IoT Greengrass Core verwalten](#).

Example: Nachrichtenstruktur des Datenstroms

Das folgende Beispiel zeigt die erforderliche Datenstrom-Nachrichtenstruktur, die vom AWS IoT Greengrass Stream-Manager übertragen wird.

```
{  
  "assetId": "string",
```

```
"propertyAlias": "string",
"propertyId": "string",
"propertyValues": [
  {
    "quality": "string",
    "timestamp": {
      "offsetInNanos": number,
      "timeInSeconds": number
    },
    "value": {
      "booleanValue": boolean,
      "doubleValue": number,
      "integerValue": number,
      "stringValue": "string"
    }
  }
]
```

Note

Die Struktur der Datenstromnachricht muss entweder (`assetId` und `propertyId`) oder `propertyAlias` enthalten.

`assetId`

(Optional) Die ID des zu aktualisierenden Assets.

`propertyAlias`

(Optional) Der Alias, der die Eigenschaft identifiziert, z. B. ein Datenstream-Pfad eines OPC UA-Servers. Beispiel:

```
/company/windfarm/3/turbine/7/temperature
```

Weitere Informationen finden Sie im AWS IoT SiteWise Benutzerhandbuch unter [Datenströme verwalten](#).

`propertyId`

(Optional) Die ID der Asset-Eigenschaft für diesen Eintrag.

propertyValues

(Erforderlich) Die Liste der hochzuladenden Eigenschaftswerte. Sie können bis zu 10 `propertyValues` Array-Elemente angeben.

quality

(Optional) Die Qualität des Immobilienwerts.

timestamp

(Erforderlich) Der Zeitstempel des Immobilienwerts.

offsetInNanos

(Optional) Der Nanosekunden-Offset von. `timeInSeconds`

timeInSeconds

(Erforderlich) Das Zeitstempeldatum in Sekunden im Unix-Epochenformat. Daten in Bruchteilen von Nanosekunden werden bereitgestellt von. `offsetInNanos`

value

(Erforderlich) Der Wert der Vermögenseigenschaft.

Note

In dem `value` Feld kann nur einer der folgenden Werte vorhanden sein.

booleanValue

(Optional) Objekteigenschaftsdaten vom Typ Boolean (`true` oder `false`).

doubleValue

(Optional) Anlageneigenschaftsdaten vom Typ Double (Fließkommazahl).

integerValue

(Optional) Daten zu Vermögenswerten vom Typ Ganzzahl (ganze Zahl).

stringValue

(Optional) Daten zu Vermögenswerten vom Typ Zeichenfolge (Zeichenfolge).

Konfigurieren Sie Edge-Funktionen auf AWS IoT SiteWise Edge

Note

Das Feature Data Processing Pack (DPP) ist für Neukunden nicht mehr verfügbar. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter [Data processing pack availability change](#).

Sie können AWS IoT SiteWise Edge verwenden, um Daten zu sammeln und vorübergehend zu speichern, sodass Sie Gerätedaten lokal organisieren und verarbeiten können. Durch die Aktivierung der Edge-Verarbeitung können Sie festlegen, dass nur aggregierte Daten an die Cloud gesendet werden, um Ihre Bandbreitennutzung und Ihre Cloud-Speicherkosten in der AWS Cloud zu optimieren. Durch die Verwendung von AWS IoT SiteWise Komponenten können Sie Daten am Edge sammeln und verarbeiten. AWS IoT Greengrass, bevor Sie sie an das Netzwerk in der AWS Cloud senden, oder Sie mithilfe von SiteWise Edge-APIs vor Ort verwalten.

Die Datenerfassung erfolgt über Datenpakete und AWS IoT SiteWise Komponenten, die darauf ausgelegt sind, in der AWS IoT Greengrass ausgeführt zu werden.

Note

- AWS IoT SiteWise speichert Ihre Edge-Daten bis zu 30 Tage auf Ihren SiteWise Edge-Gateways. Die Aufbewahrungsdauer Ihrer Daten hängt vom verfügbaren Speicherplatz Ihres Geräts ab.
- Wenn Ihr SiteWise Edge-Gateway 30 Tage lang nicht mit dem AWS Cloud verbunden war, wird das [Data Processing Pack](#) automatisch deaktiviert.

Themen

- [Richten Sie die Edge-Funktion in SiteWise Edge ein](#)

Richten Sie die Edge-Funktion in SiteWise Edge ein

AWS IoT SiteWise stellt die folgenden Pakete bereit, anhand derer Ihr Edge-Gateway bestimmen kann, wie Ihre Daten erfasst und verarbeitet werden sollen. Wählen Sie Pakete aus, um Edge-Funktionen für Ihr SiteWise Edge-Gateway zu aktivieren.

- Das Datenerfassungspaket ermöglicht es Ihrem SiteWise Edge-Gateway, Daten von mehreren OPC UA-Servern zu sammeln und die Daten dann vom Edge in die AWS Cloud zu exportieren. Es wird aktiv, sobald Sie Ihrem SiteWise Edge-Gateway Datenquellen hinzugefügt haben.
- Das Datenverarbeitungspaket ermöglicht es Ihrem SiteWise Edge-Gateway, Ihre Gerätedaten am Edge zu verarbeiten. Sie können beispielsweise Asset-Modelle verwenden, um Metriken und Transformationen zu berechnen. Weitere Informationen zu Anlagenmodellen und Vermögenswerten finden Sie unter [Modellieren Sie Industrieanlagen](#).

 Note

- Das Datenverarbeitungspaket ist nur auf x86-Plattformen verfügbar.

Um Edge-Funktionen zu konfigurieren

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie das SiteWise Edge-Gateway aus, für das Sie Edge-Funktionen aktivieren möchten.
4. Wählen Sie im Abschnitt Edge-Funktionen die Option Bearbeiten
5. Wählen Sie im Abschnitt Edge-Funktionen die Option Datenverarbeitungspaket aktivieren aus (es fallen zusätzliche Gebühren an).
6. (Optional) Im Abschnitt Edge-LDAP-Verbindung können Sie Benutzergruppen in Ihrem Unternehmensverzeichnis Zugriff auf dieses SiteWise Edge-Gateway gewähren. Die Benutzergruppen können die LDAP-Anmeldeinformationen (Lightweight Directory Access Protocol) verwenden, um auf das SiteWise Edge-Gateway zuzugreifen. Anschließend können sie die AWS OpsHub für AWS IoT SiteWise Anwendungen, AWS IoT SiteWise API-Operationen oder andere Tools verwenden, um das SiteWise Edge-Gateway zu verwalten. Weitere Informationen finden Sie unter [SiteWise Edge-Gateways verwalten](#).

 Note

Sie können auch Linux oder Microsoft Windows Anmeldeinformationen verwenden, um auf das SiteWise Edge-Gateway zuzugreifen. Weitere Informationen finden Sie unter [Greifen Sie mit den Anmeldeinformationen für das Linux-Betriebssystem auf Ihr SiteWise Edge-Gateway zu](#).

- a. Wählen Sie Aktiviert aus.
- b. Geben Sie unter Anbietername einen Namen für Ihren LDAP-Anbieter ein.
- c. Geben Sie für Hostname oder IP-Adresse den Hostnamen oder die IP-Adresse Ihres LDAP-Servers ein.
- d. Geben Sie für Port eine Portnummer ein.
- e. Geben Sie für Base Distinguished Name (DN) einen definierten Namen (DN) für die Basis ein.

Die folgenden Attributtypen werden unterstützt: CommonName (CN), LocalityName (L), state (ST), OrganizationName OrProvinceName (O), organizational (OU), countryName UnitName (C), streetAddress (STREET), domainComponent (DC) und userid (UID).

- f. Geben Sie für Admin-Gruppen-DN einen DN ein.
- g. Geben Sie für Benutzergruppen-DN einen DN ein.

7. Wählen Sie Speichern.

Nachdem Sie die Edge-Funktionen auf Ihrem SiteWise Edge-Gateway aktiviert haben, müssen Sie Ihr Asset-Modell für das Edge konfigurieren. Die Edge-Konfiguration Ihres Asset-Modells gibt an, wo Ihre Asset-Eigenschaften berechnet werden. Sie können alle Eigenschaften am Rand berechnen oder Sie können die Eigenschaften Ihres Asset-Modells separat konfigurieren. Zu den Eigenschaften des Anlagenmodells gehören [Metriken](#), [Transformationen](#) und [Messungen](#).

Weitere Informationen zu Asset-Eigenschaften finden Sie unter [the section called “Definieren Sie Dateneigenschaften”](#).

Nachdem Sie Ihr Asset-Modell erstellt haben, können Sie es für den Edge konfigurieren. Weitere Informationen zur Konfiguration Ihres Asset-Modells für den Edge finden Sie unter [the section called “Erstellen Sie ein Asset-Modell \(Konsole\)”](#).

Note

Asset-Modelle und Dashboards werden automatisch alle 10 Minuten zwischen der AWS Cloud und Ihrem SiteWise Edge-Gateway synchronisiert. Sie können die Synchronisierung auch manuell über die lokale SiteWise Edge-Gateway-Anwendung durchführen.

Konfigurieren Sie die Edge-Datenverarbeitung für AWS IoT SiteWise Modelle und Anlagen

Note

Das Feature Data Processing Pack (DPP) ist für Neukunden nicht mehr verfügbar. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter [Data processing pack availability change](#).

Sie können AWS IoT SiteWise Edge verwenden, um Gerätedaten lokal zu sammeln, zu speichern, zu organisieren und zu überwachen. Sie können SiteWise Edge verwenden, um Ihre Industriedaten zu modellieren und lokal zu visualisieren. Sie können Ihre Daten lokal verarbeiten und an die AWS Cloud senden oder sie mithilfe der AWS IoT SiteWise API vor Ort verarbeiten.

Mit AWS IoT SiteWise Edge können Sie Rohdaten lokal verarbeiten und sich dafür entscheiden, nur aggregierte Daten an die AWS Cloud zu senden, um Ihre Bandbreitennutzung und Ihre Cloud-Speicherkosten zu optimieren.

Note

- AWS IoT SiteWise speichert Ihre Edge-Daten bis zu 30 Tage auf Ihren SiteWise Edge-Gateways. Die Aufbewahrungsdauer Ihrer Daten hängt vom verfügbaren Speicherplatz Ihres Geräts ab.
- Wenn Ihr SiteWise Edge-Gateway 30 Tage lang nicht mit der AWS Cloud verbunden war, [Richten Sie eine OPC UA-Quelle in SiteWise Edge ein](#) wird es automatisch deaktiviert.

Konfigurieren Sie ein Asset-Modell für die Datenverarbeitung auf Edge SiteWise

Sie müssen Ihr Asset-Modell für den Edge konfigurieren, bevor Sie Ihre SiteWise Edge-Gateway-Daten am Edge verarbeiten können. Die Edge-Konfiguration Ihres Asset-Modells gibt an, wo Ihre Asset-Eigenschaften berechnet werden. Sie können wählen, ob Sie alle Eigenschaften am Edge berechnen und die Ergebnisse an die AWS Cloud senden möchten, oder Sie können festlegen, wo die einzelnen Asset-Eigenschaften separat berechnet werden sollen. Weitere Informationen finden Sie unter [Konfigurieren Sie die Edge-Datenverarbeitung für AWS IoT SiteWise Modelle und Anlagen](#).

Zu den Asset-Eigenschaften gehören Metriken, Transformationen und Messungen:

- Metriken sind die aggregierten Daten der Anlage über einen bestimmten Zeitraum. Sie können neue Metriken berechnen, indem Sie vorhandene Metrikdaten verwenden. AWS IoT SiteWise sendet Ihre Metriken immer zur Langzeitspeicherung in die AWS Cloud. AWS IoT SiteWise berechnet standardmäßig Metriken in der AWS Cloud. Sie können Ihr Asset-Modell so konfigurieren, dass Ihre Metriken am Edge berechnet werden. AWS IoT SiteWise sendet verarbeitete Ergebnisse an die AWS Cloud.
- Transformationen sind mathematische Ausdrücke, die die Datenpunkte einer Komponenteneigenschaft aus einer Form in eine andere Form abbilden. Transformationen können Metriken als Eingabedaten verwenden und müssen am selben Ort wie ihre Eingaben berechnet und gespeichert werden. Wenn Sie eine metrische Eingabe so konfigurieren, dass sie am Rand berechnet wird, wird AWS IoT SiteWise auch die zugehörige Transformation am Rand berechnet.
- Messungen werden standardmäßig als Rohdaten formatiert, die Ihr Gerät sammelt und an die AWS Cloud sendet. Sie können Ihr Asset-Modell so konfigurieren, dass diese Daten auf Ihrem lokalen Gerät gespeichert werden.

Weitere Informationen zu den Eigenschaften von Vermögenswerten finden Sie unter [the section called “Definieren Sie Dateneigenschaften”](#).

Nachdem Sie Ihr Asset-Modell erstellt haben, können Sie es für den Edge konfigurieren. Weitere Informationen zur Konfiguration Ihres Asset-Modells für den Edge finden Sie unter [the section called “Erstellen Sie ein Asset-Modell \(Konsole\)”](#).

 Note

Asset-Modelle und Dashboards werden automatisch alle 10 Minuten zwischen der AWS Cloud und Ihrem SiteWise Edge-Gateway synchronisiert. Sie können die Synchronisierung auch manuell von der aus durchführen. [SiteWise Edge-Gateways verwalten](#)

Sie können die AWS IoT SiteWise REST-APIs und die AWS Command Line Interface (AWS CLI) verwenden, um Ihr SiteWise Edge-Gateway nach Daten am Edge abzufragen. Bevor Sie Ihr SiteWise Edge-Gateway nach Daten am Edge abfragen, müssen Sie die folgenden Voraussetzungen erfüllen:

- Ihre Anmeldeinformationen müssen für die REST-APIs festgelegt werden. Weitere Informationen zum Einrichten von Anmeldeinformationen finden Sie unter [the section called “Gateways verwalten”](#).
- Der SDK-Endpunkt muss auf die IP-Adresse Ihres SiteWise Edge-Gateways verweisen. Weitere Informationen finden Sie in der Dokumentation zu Ihrem SDK. Weitere Informationen finden

Sie beispielsweise unter [Spezifizieren benutzerdefinierter Endpunkte](#) im AWS SDK for Java 2.x-Entwicklerhandbuch.

- Ihr SiteWise Edge-Gateway-Zertifikat muss registriert sein. Weitere Informationen zur Registrierung Ihres SiteWise Edge-Gateway-Zertifikats finden Sie in der Dokumentation zu Ihrem SDK. Weitere Informationen finden Sie beispielsweise unter [Registrierung von Zertifikatspaketen Node.js im AWS SDK for Java 2.x](#) Entwicklerhandbuch.

Weitere Informationen zum Abfragen von Daten mit finden Sie AWS IoT SiteWise unter. [Daten abfragen von AWS IoT SiteWise](#)

Fügen Sie Datenquellen zu Ihrem hinzuAWS IoT SiteWiseEdge-Gateway

Nach der Einrichtung eines AWS IoT SiteWise Edge-Gateways können Sie Datenquellen hinzufügen und konfigurieren, in die Daten von lokalen Industrieanlagen aufgenommen werden. AWS IoT SiteWise SiteWise Edge unterstützt verschiedene Protokolle, darunter OPC UA, und viele andere Protokolle, die über Partnerdatenquellen verfügbar sind. Diese Quellen ermöglichen es Ihrem Gateway, eine Verbindung zu lokalen Servern herzustellen und Ihre Industriedaten abzurufen. Durch die Konfiguration von Datenquellen können Sie Daten aus einer Vielzahl von Datenquellen aufnehmen und die Datenströme dann mit Anlageneigenschaften verknüpfen, was eine umfassende Modellierung und Datenkartierung von Industrieanlagen ermöglicht. AWS IoT SiteWise

Themen

- [OPC UA-Datenquellen fürAWS IoT SiteWiseEdge-Gateways](#)
- [Partnerdatenquellen auf SiteWise Edge-Gateways](#)

OPC UA-Datenquellen fürAWS IoT SiteWiseEdge-Gateways

Nachdem Sie ein AWS IoT SiteWise Edge-Gateway eingerichtet haben, können Sie Datenquellen so konfigurieren, dass Ihr SiteWise Edge-Gateway Daten von lokalen Industrieanlagen aufnehmen kann. AWS IoT SiteWise Jede Quelle steht für einen lokalen Server, z. B. einen OPC UA-Server, mit dem Ihr SiteWise Edge-Gateway eine Verbindung herstellt und industrielle Datenströme abrufen. Weitere Informationen zum Einrichten eines SiteWise Edge-Gateways finden Sie unter. [Erstellen Sie ein selbst gehostetes SiteWise Edge-Gateway](#)

Der Gateway-Typ MQTT-enabled, V3-Gateways im Vergleich zu klassischen Stream-V2-Gateways, beeinflusst, wie OPC UA-Daten verarbeitet werden. In Classic Stream werden V2-Gateways und

OPC UA-Datenquellen direkt zur SiteWise Gateway-IoT-Publisher-Konfiguration hinzugefügt. Jede Datenquelle ist mit dem Gateway gekoppelt, und das Datenrouting wird für jede Quelle individuell konfiguriert. Im Gegensatz dazu werden OPC UA-Datenquellen mithilfe MQTT-enabled von V3-Gateways in MQTT-Themen konvertiert und über zentrale Ziele verwaltet. Weitere Informationen zu den einzelnen Typen finden Sie unter und [MQTT-enabled, V3-Gateways fürAWS IoT SiteWiseEdge](#)
[Klassische Streams, V2-Gateways fürAWS IoT SiteWiseEdge](#)

 Note

AWS IoT SiteWise startet Ihr SiteWise Edge-Gateway jedes Mal neu, wenn Sie eine Quelle hinzufügen oder bearbeiten. Ihr SiteWise Edge-Gateway nimmt keine Daten auf, während es die Quellkonfiguration aktualisiert. Die Zeit für den Neustart Ihres SiteWise Edge-Gateways hängt von der Anzahl der Tags in den Quellen Ihres SiteWise Edge-Gateways ab. Die Neustartzeit kann zwischen einigen Sekunden (für ein SiteWise Edge-Gateway mit wenigen Tags) und mehreren Minuten (für ein SiteWise Edge-Gateway mit vielen Tags) liegen.

Nachdem Sie Quellen erstellt haben, können Sie Ihre Datenströme mit Asset-Eigenschaften verknüpfen. Weitere Informationen zum Erstellen und Verwenden von Komponenten finden Sie unter [Modellieren Sie Industrieanlagen](#).

Sie können CloudWatch Metriken anzeigen, um zu überprüfen, ob eine Datenquelle verbunden ist AWS IoT SiteWise. Weitere Informationen finden Sie unter [AWS IoT Greengrass Version 2 Gateway-Metriken](#).

AWS IoT SiteWise unterstützt derzeit die folgenden Datenquellenprotokolle:

- [OPC UA](#) — Ein Machine-to-Machine-Kommunikationsprotokoll (M2M) für die industrielle Automatisierung.

Support für zusätzliche Industrieprotokolle

SiteWise Edge unterstützt durch die Integration mit Datenquellenpartnern eine Vielzahl von Industrieprotokollen. Diese Partnerschaften ermöglichen Konnektivität mit über 200 verschiedenen Protokollen für verschiedene industrielle Systeme und Geräte.

Eine Liste der verfügbaren Datenquellenpartner finden Sie unter [SiteWise Datenquellenoptionen für Edge-Gateway-Partner](#).

Richten Sie eine OPC UA-Quelle in SiteWise Edge ein

Sie können die AWS IoT SiteWise Konsole oder eine SiteWise Edge-Gateway-Funktion verwenden, um eine OPC-UA-Quelle zu definieren und zu Ihrem SiteWise Edge-Gateway hinzuzufügen, die einen lokalen OPC UA-Server darstellt.

Themen

- [Konfigurieren Sie eine OPC UA-Quelle \(Konsole\)](#)
- [Konfigurieren Sie eine OPC UA-Quelle \(AWS CLI\)](#)

Konfigurieren Sie eine OPC UA-Quelle (Konsole)

Sie können die Konsole verwenden, um die OPC UA-Quelle mit dem folgenden Verfahren zu konfigurieren.

Note

Warnung: Doppelte TQVs können zu einer doppelten Aufladung führen.

Um eine OPC UA-Quelle mit dem zu konfigurierenAWS IoT SiteWiseKonsole

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie in der linken Navigationsleiste im Abschnitt Edge die Option Edge-Gateways aus.
3. Wählen Sie das SiteWise Edge-Gateway aus, um eine OPC UA-Quelle hinzuzufügen.
4. Wählen Sie Datenquelle hinzufügen aus.
5. Geben Sie einen Namen für die Quelle ein.
6. Geben Sie den Local endpoint (Lokaler Endpunkt) des Datenquellenservers ein. Der Endpunkt kann die IP-Adresse oder der Hostname sein. Sie können dem lokalen Endpunkt auch eine Portnummer hinzufügen. Ihr lokaler Endpunkt könnte beispielsweise so aussehen:
opc.tcp://203.0.113.0:49320
7. (Optional) Fügen Sie unter Knoten-ID zur Auswahl Knotenfilter hinzu, um einzuschränken, welche Datenströme in die AWS Cloud aufgenommen werden. Standardmäßig verwenden SiteWise Edge-Gateways den Stammknoten eines Servers, um alle Datenströme aufzunehmen. Sie können Knotenfilter verwenden, um die Startzeit und die CPU-Auslastung Ihres SiteWise Edge-Gateways zu reduzieren, indem Sie nur Pfade zu Daten einbeziehen, die Sie modellieren.

AWS IoT SiteWise Standardmäßig laden SiteWise Edge-Gateways alle OPC UA-Pfade hoch, außer denen, die mit `/Server/` beginnen. Um OPC UA-Knotenfilter zu definieren, können Sie Knotenpfade und die `*` Platzhalterzeichen verwenden. ** Weitere Informationen finden Sie unter [Verwenden Sie OPC UA-Knotenfilter in Edge SiteWise](#).

8. Die Ziele variieren zwischen MQTT-enabled V3-Gateways und Classic-Streams, V2-Gateways.
 - Klassische Streams und V2-Gateway-Ziele haben eine 1:1 -Beziehung zur Quelle. Jede Quelle sendet Daten an ein bestimmtes Ziel.
 - MQTT-enabled, V3-Gateway-Ziele werden separat eingerichtet, da Sie mit dem Hub-and-Spoke-Modell die Konfiguration und Verwaltung mehrerer Datenquellen über verschiedene Gateways hinweg zentralisieren können. Informationen zum Einrichten von Zielen in einem V3-Gateway finden Sie unter [Verstehen Sie AWS IoT SiteWise Edge-Ziele](#)

Classic streams, V2 gateway destinations

- **AWS IoT SiteWiseEchtzeit** — Wählen Sie diese Option, um Daten direkt an den AWS IoT SiteWise Speicher zu senden. Erfassen und überwachen Sie Daten in Echtzeit am Netzwerkrand.
- **AWS IoT SiteWiseMit Amazon S3 gepuffert** — Daten im Parquet-Format an Amazon S3 senden und dann in den AWS IoT SiteWise Speicher importieren. Wählen Sie diese Option, um Daten stapelweise aufzunehmen und historische Daten kostengünstig zu speichern. Sie können Ihren bevorzugten Amazon S3-Bucket-Standort und die Häufigkeit, mit der Daten auf Amazon S3 hochgeladen werden sollen, konfigurieren. Sie können auch wählen, was mit den Daten nach der Aufnahme geschehen soll. AWS IoT SiteWise Sie können wählen, ob die Daten AWS IoT SiteWise sowohl in Amazon S3 als auch in Amazon S3 verfügbar sein sollen, oder Sie können festlegen, dass sie nach dem Import automatisch aus Amazon S3 gelöscht werdenAWS IoT SiteWise.
 - Der Amazon S3 S3-Bucket ist ein Staging- und Puffermechanismus und unterstützt Dateien im Parquet-Format.
 - Wenn Sie das Kontrollkästchen Daten in AWS IoT SiteWise Speicher importieren aktivieren, werden Daten zuerst in Amazon S3 und dann in den AWS IoT SiteWise Speicher hochgeladen.
 - Wenn Sie das Kontrollkästchen Daten aus Amazon S3 löschen aktivieren, werden Daten aus Amazon S3 gelöscht, nachdem sie in den SiteWise Speicher importiert wurden.

- Wenn Sie das Kontrollkästchen Daten aus Amazon S3 löschen deaktivieren, werden Daten sowohl in Amazon S3 als auch im SiteWise Speicher gespeichert.
- Wenn Sie das Kontrollkästchen Daten in AWS IoT SiteWise Speicher importieren deaktivieren, werden Daten nur in Amazon S3 gespeichert. Sie werden nicht in den SiteWise Speicher importiert.

Einzelheiten [Verwalten des Datenspeichers](#) zu den verschiedenen Speicheroptionen finden Sie AWS IoT SiteWise unter. Weitere Informationen zu den Preisoptionen finden Sie unter [AWS IoT SiteWisePreise](#).

- AWS IoT GreengrassStream-Manager — Verwenden Sie den AWS IoT Greengrass Stream-Manager, um Daten an die folgenden AWS Cloud-Ziele zu senden: Kanäle in AWS IoT Analytics, Streams in Amazon Kinesis Data Streams, Asset-Eigenschaften in AWS IoT SiteWise oder Objekte in Amazon Simple Storage Service (Amazon S3). Weitere Informationen finden Sie im AWS IoT Greengrass Version 2Entwicklerhandbuch unter [Datenstreams auf dem AWS IoT Greengrass Core verwalten](#).

Geben Sie einen Namen für den AWS IoT Greengrass Stream ein.

MQTT-enabled, V3 gateway destinations

1. Informationen [MQTT-enabled, V3-Gateways fürAWS IoT SiteWiseEdge](#) zum Hinzufügen Ihrer relevanten Ziele finden Sie unter.
 2. Kehren Sie nach dem Hinzufügen Ihrer Quellziele zu diesem Verfahren zurück.
9. Im Bereich Erweiterte Konfiguration können Sie wie folgt vorgehen:
- a. Wählen Sie einen Nachrichtensicherheitsmodus für Verbindungen und Daten, die zwischen Ihrem Quellserver und Ihrem SiteWise Edge-Gateway übertragen werden. Dieses Feld ist die Kombination aus der OPC UA-Sicherheitsrichtlinie und dem Nachrichtensicherheitsmodus. Wählen Sie dieselbe Sicherheitsrichtlinie und denselben Nachrichtensicherheitsmodus, die Sie für Ihren OPC UA-Server angegeben haben.
 - b. Wenn Ihre Quelle eine Authentifizierung erfordert, wählen Sie ein AWS Secrets Manager Geheimnis aus der Authentifizierungskonfigurationsliste aus. Das SiteWise Edge-Gateway verwendet die Authentifizierungsanmeldeinformationen in diesem Geheimnis, wenn es eine Verbindung zu dieser Datenquelle herstellt. Sie müssen Geheimnisse an die AWS IoT Greengrass Komponente Ihres SiteWise Edge-Gateways anhängen, um sie für die

Datenquellenauthentifizierung verwenden zu können. Weitere Informationen finden Sie unter [the section called “Konfigurieren Sie die Datenquellenauthentifizierung”](#).

 Tip

Ihr Datenserver verfügt möglicherweise über die Option Allow anonymous login (Anonyme Anmeldung zulassen). Wenn diese Option Yes (Ja) zeigt, ist für Ihre Quelle keine Authentifizierung erforderlich.

- c. (Optional) Sie können ein Datenstream-Präfix aktivieren, indem Sie Datenstream-Präfix aktivieren — optional auswählen.
 - Geben Sie ein Datenstream-Präfix ein. Das SiteWise Edge-Gateway fügt dieses Präfix allen Datenströmen aus dieser Quelle hinzu. Verwenden Sie ein Datenstrom-Präfix, um zwischen Datenströmen mit demselben Namen aus verschiedenen Quellen zu unterscheiden. Jeder Datenstrom sollte einen eindeutigen Namen in Ihrem Konto haben.
- d. (Optional) Wählen Sie eine Option zur Konvertierung von Datentypen, um nicht unterstützte OPC UA-Datentypen in Zeichenketten zu konvertieren, bevor Sie sie aufnehmen. AWS IoT SiteWise Konvertiert Array-Werte mit einfachen Datentypen in JSON-Zeichenketten und DateTime Datentypen in ISO-8601-Zeichenketten. Weitere Informationen finden Sie unter [Nicht unterstützte Datentypen werden konvertiert](#).
- e. Wählen Sie einen Standardauslöser für Datenänderungen für Knoten, die nicht in einer benutzerdefinierten Eigenschaftsgruppe enthalten sind. Der Standard-Trigger für Datenänderungen bestimmt, wann der OPC UA-Server aktualisierte Werte an das Gateway sendet. Sie können eine der folgenden Optionen wählen:
 - Status — um Daten nur zu empfangen, wenn sich ein Status ändert.
 - StatusValue— um Daten zu empfangen, wenn sich ein Status oder Wert ändert.
 - StatusValueTimestamp— um Daten zu empfangen, wenn sich ein Status, ein Wert oder ein Zeitstempel ändert.
- f. (Optional) Auf einem MQTT-enabled V3-Gateway können Sie die Discovery-Konfiguration verwenden, um den OPC UA-Knotenerkennungsprozess zu konfigurieren. Die Discovery-Konfiguration ersetzt das bisherige Dateisystem zur Überschreibung der Konfiguration für diese Optionen durch konsolenbasierte Einstellungen, die dynamisch aktualisiert werden, ohne dass das Gateway neu gestartet werden muss.

 Note

Für den standardmäßigen Datenänderungs-Trigger ist Version 3.1.0 oder höher der IoT SiteWise OPC UA-Collector-Komponente erforderlich. Weitere Informationen finden Sie unter [Aktualisieren Sie die Version einer Komponente AWS IoT SiteWise](#).

- i. Geben Sie für Maximale Anzahl gleichzeitiger Suchanfragen die maximale Anzahl von Suchanfragen ein, die Ihr OPC UA-Server gleichzeitig verarbeiten kann. Sie können bis zu 500 gleichzeitige Suchanfragen pro Datenquelle konfigurieren.
- ii. Geben Sie für Maximale Knotenanzahl pro Suchanforderung die maximale Anzahl von Knoten ein, die in jeder Suchanforderung an den OPC UA-Server gesendet werden sollen. Sie können bis zu 1.000 Knoten pro Suchanforderung senden.
- iii. Wählen Sie Node Tree Loops vermeiden, um zu verhindern, dass das Gateway beim Durchsuchen der Struktur des OPC UA-Servers in Zirkelverweisen hängen bleibt. Wenn diese Option ausgewählt ist, verfolgt das Gateway besuchte Standorte, um Endlosschleifen zu vermeiden, die entstehen können, wenn Serverknoten in einem kreisförmigen Muster aufeinander verweisen.
- iv. Wählen Sie „Node Traversal aktivieren“, damit das Gateway die gesamte Struktur Ihres OPC UA-Servers untersuchen kann, um alle verfügbaren Datenpunkte Ihrer Geräte und Geräte zu ermitteln. Wenn diese Option ausgewählt ist, navigiert das Gateway durch die Datenorganisation Ihrer Geräte über die Stammebene hinaus, um alle Sensoren, Steuerungen und Messpunkte automatisch zu finden.
- v. Wählen Sie Periodische Erkennung aktivieren, um in regelmäßigen Abständen automatisch Erkennungsvorgänge auszuführen, um Änderungen in der Struktur des OPC UA-Servers zu erkennen. Wenn diese Option ausgewählt ist, sucht das Gateway kontinuierlich nach neu hinzugefügten Geräten oder Datenpunkten, um sicherzustellen, dass diese automatisch erkannt und für die Datenerfassung zur Verfügung gestellt werden.
 - A. Geben Sie für das periodische Erkennungsintervall das Zeitintervall zwischen automatischen Erkennungsvorgängen an, wenn die periodische Erkennung ausgeführt wird. Das minimale periodische Erkennungsintervall beträgt 30 Sekunden und das Maximum 30 Tage.

- B. Legen Sie für Maximale Anzahl erkannter Knoten pro Intervall die maximale Anzahl von Knoten fest, die pro Erkennungsintervall erkannt werden sollen. Dies hilft, die Last sowohl auf dem Gateway als auch auf dem OPC UA-Server während der Erkennungsvorgänge zu kontrollieren.
- g. (Optional) Wählen Sie für Eigenschaftsgruppen die Option Neue Gruppe hinzufügen aus.
 - i. Geben Sie einen Namen für die Eigenschaftsgruppe ein.
 - ii. Für Eigenschaften:
 - 1. Fügen Sie für Knotenpfade OPC UA-Knotenfilter hinzu, um einzuschränken, in welche OPC UA-Pfade hochgeladen werden. AWS IoT SiteWise Das Format ähnelt der Node-ID für die Auswahl.
 - iii. Gehen Sie für Gruppeneinstellungen wie folgt vor:
 - 1. Wählen Sie unter Datenqualitätseinstellung den Datenqualitätstyp aus, den AWS IoT SiteWise Collector aufnehmen soll.
 - 2. Konfigurieren Sie für die Einstellung für den Scanmodus die Eigenschaften des Standardabonnements im Scanmodus. Sie können „Abonnieren“ oder „Umfrage“ wählen. Weitere Informationen zum Scanmodus finden Sie unter [the section called „Filtern Sie Datenaufnahmebereiche“](#).

Subscribe

Um jeden Datenpunkt zu senden

- i. Wählen Sie Abonnieren und stellen Sie Folgendes ein:
 - A. [Auslöser für Datenänderungen](#) — Die Bedingung, die eine Warnung bei Datenänderungen auslöst.
 - B. [Größe der Abonnement-Warteschlange](#) — Die Tiefe der Warteschlange auf einem OPC UA-Server für eine bestimmte Metrik, in der Benachrichtigungen für überwachte Elemente in die Warteschlange gestellt werden.
 - C. [Veröffentlichungsintervall für Abonnements](#) — Das Intervall (in Millisekunden) des Veröffentlichungszyklus, das bei der Erstellung des Abonnements angegeben wurde.

- D. Snapshot-Intervall — Optional — Die Timeout-Einstellung für die Snapshot-Frequenz, um sicherzustellen, dass AWS IoT SiteWise Edge einen stetigen Datenstrom aufnimmt.
- E. Scanrate — Die Geschwindigkeit, mit der das SiteWise Edge-Gateway Ihre Register lesen soll. AWS IoT SiteWise berechnet automatisch die minimal zulässige Scanrate für Ihr SiteWise Edge-Gateway.
- F. Zeitstempel — Der Zeitstempel, der Ihren OPC UA-Datenpunkten beigelegt werden soll. Sie können den Serverzeitstempel oder den Zeitstempel Ihres Geräts verwenden.

**Note**

Verwenden Sie Version 2.5.0 oder höher der IoT SiteWise OPC UA-Collector-Komponente. Wenn Sie die Zeitstempelfunktion mit früheren Versionen verwenden, schlagen Konfigurationsupdates fehl. Weitere Informationen finden Sie unter [Aktualisieren Sie die Version einer Komponente AWS IoT SiteWise](#).

- ii. Konfigurieren Sie in den Deadband-Einstellungen einen Deadband-Typ. Der Deadband-Typ steuert, welche Daten Ihre Quelle an Sie sendet und welche Daten sie verwirft. AWS IoT SiteWise Weitere Informationen zur Einstellung der Deadband-Einstellung finden Sie unter [the section called “Filtern Sie Datenaufnahmebereiche”](#)
 - Keine — Der zugehörige Server sendet alle Datenpunkte für diese Eigenschaftengruppe.
 - Prozentsatz — Der zugehörige Server sendet nur Daten, die außerhalb eines bestimmten Prozentsatzes des Datenbereichs liegen. Dieser Bereich wird vom Server auf der Grundlage der für jeden Knoten definierten Mindest- und Höchstwerte der technischen Einheit berechnet. Wenn der Server keine prozentualen Deadbands unterstützt oder keine definierten technischen Einheiten definiert sind, berechnet das Gateway den Bereich anhand der unten angegebenen Mindest- und Höchstwerte.
 - Absolut — Der zugehörige Server sendet nur Daten, die außerhalb eines bestimmten Bereichs liegen.

- A. Legen Sie den Wert für „Deadband“ als Prozentsatz des Datenbereichs fest, in dem das Deadband erreicht ist.
- B. (Optional) Geben Sie mit Minimaler Bereich — optional und Maximaler Bereich — optional einen Mindest - und Höchstwert für den Totbandbereich an.

Poll

Um Datenpunkte in einem bestimmten Intervall zu senden

- Wählen Sie Umfrage und legen Sie Folgendes fest:
 - A. Scanrate — Die Geschwindigkeit, mit der das SiteWise Edge-Gateway Ihre Register lesen soll. AWS IoT SiteWise berechnet automatisch die minimal zulässige Scanrate für Ihr SiteWise Edge-Gateway.
 - B. Zeitstempel — Der Zeitstempel, der Ihren OPC UA-Datenpunkten beigelegt werden soll. Sie können den Serverzeitstempel oder den Zeitstempel Ihres Geräts verwenden.



Note

Verwenden Sie Version 3.1.0 oder höher der IoT SiteWise OPC UA-Collector-Komponente. Wenn Sie die Zeitstempelfunktion mit früheren Versionen verwenden, schlagen Konfigurationsupdates fehl. Weitere Informationen finden Sie unter [Aktualisieren Sie die Version einer Komponente AWS IoT SiteWise](#).



Note

Die Deadband-Einstellungen gelten, wenn Sie in den Einstellungen für den Scanmodus die Option Abonnieren ausgewählt haben.

10. Wählen Sie Speichern.

Konfigurieren Sie eine OPC UA-Quelle (AWS CLI)

Sie können OPC UA-Datenquellen für ein SiteWise Edge-Gateway mit dem definieren. AWS CLI Erstellen Sie dazu eine JSON-Datei mit der OPC-UA-Fähigkeitskonfiguration und verwenden Sie den Befehl [update-gateway-capability-configuration](#), um die [Edge-Gateway-Konfiguration](#) zu aktualisieren. SiteWise Sie müssen alle Ihre OPC UA-Quellen in einer einzigen Funktionskonfiguration definieren.

MQTT-enabled, V3 gateway

Diese Fähigkeit hat den folgenden Namespace.

- `iotsitewise:opcuacollector:3`

```
{
  "sources": [
    {
      "name": "string",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny" | "X509",
          "certificateBody": "string",
          "certificateChain": "string",
        },
        "endpointUri": "string",
        "securityPolicy": "NONE" | "BASIC128_RSA15" | "BASIC256" | "BASIC256_SHA256"
        | "AES128_SHA256_RSA0AEP" | "AES256_SHA256_RSAPSS",
        "messageSecurityMode": "NONE" | "SIGN" | "SIGN_AND_ENCRYPT",
        "identityProvider": {
          "type": "Anonymous" | "Username",
          "usernameSecretArn": "string"
        },
        "nodeFilterRules": [
          {
            "action": "INCLUDE",
            "definition": {
              "type": "OpcUaRootPath",
              "rootPath": "string"
            }
          }
        ]
      },
      "measurementDataStreamPrefix": "string",
    }
  ]
}
```

```

"typeConversions": {
  "array": "JsonArray",
  "datetime": "ISO8601String"
},
"destination": {
  {
    "type": "MQTT"
  }
},
"defaultPropertyGroupConfig": {
  "dataChangeTrigger": "STATUS" | "STATUS_VALUE" | "STATUS_VALUE_TIMESTAMP"
},
"discoveryConfig": {
  "enableNodeTraversal": true | false,
  "avoidNodeTreeLoops": true | false,
  "maxConcurrentBrowseRequests": integer,
  "maxNodesPerBrowseRequest": integer,
  "periodicDiscovery": {
    "interval": "string",
    "maxNodesDiscoveredPerInterval": integer
  }
},
"propertyGroups": [
  {
    "name": "string",
    "nodeFilterRuleDefinitions": [
      {
        "type": "OpcUaRootPath",
        "rootPath": "string"
      }
    ],
    "deadband": {
      "type": "PERCENT" | "ABSOLUTE",
      "value": double,
      "eguMin": double,
      "eguMax": double,
      "timeoutMilliseconds": integer
    },
    "scanMode": {
      "type": "EXCEPTION" | "POLL",
      "rate": integer,
      "timestampToReturn": "SOURCE_TIME" | "SERVER_TIME"
    },
    "dataQuality": {

```

```

        "allowGoodQuality": true | false,
        "allowBadQuality": true | false,
        "allowUncertainQuality": true | false
    },
    "subscription": {
        "dataChangeTrigger": "STATUS" | "STATUS_VALUE" |
"STATUS_VALUE_TIMESTAMP",
        "queueSize": integer,
        "publishingIntervalMilliseconds": integer,
        "snapshotFrequencyMilliseconds": integer
    }
}
]
}
]
}

```

Classic streams, V2 gateway

Diese Fähigkeit hat den folgenden Namespace.

- `iotsitewise:opcuacollector:2`

Erforderliche Syntax

```

{
  "sources": [
    {
      "name": "string",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny" | "X509",
          "certificateBody": "string",
          "certificateChain": "string",
        },
        "endpointUri": "string",
        "securityPolicy": "NONE" | "BASIC128_RSA15" | "BASIC256" | "BASIC256_SHA256"
| "AES128_SHA256_RSA0AEP" | "AES256_SHA256_RSAPSS",
        "messageSecurityMode": "NONE" | "SIGN" | "SIGN_AND_ENCRYPT",
        "identityProvider": {
          "type": "Anonymous" | "Username",
          "usernameSecretArn": "string"
        }
      },
    }
  ],
}

```

```

    "nodeFilterRules": [
      {
        "action": "INCLUDE",
        "definition": {
          "type": "OpcUaRootPath",
          "rootPath": "string"
        }
      }
    ],
    "measurementDataStreamPrefix": "string",
    "typeConversions": {
      "array": "JsonArray",
      "datetime": "ISO8601String"
    },
    "destination": {
      "type": "StreamManager",
      "streamName": "string",
      "streamBufferSize": integer,
    },
    "propertyGroups": [
      {
        "name": "string",
        "nodeFilterRuleDefinitions": [
          {
            "type": "OpcUaRootPath",
            "rootPath": "string"
          }
        ],
        "deadband": {
          "type": "PERCENT" | "ABSOLUTE",
          "value": double,
          "eguMin": double,
          "eguMax": double,
          "timeoutMilliseconds": integer
        },
        "scanMode": {
          "type": "EXCEPTION" | "POLL",
          "rate": integer,
          "timestampToReturn": "SOURCE_TIME" | "SERVER_TIME"
        },
        "dataQuality": {
          "allowGoodQuality": true | false,
          "allowBadQuality": true | false,

```

```

        "allowUncertainQuality": true | false
    },
    "subscription": {
        "dataChangeTrigger": "STATUS" | "STATUS_VALUE" |
"STATUS_VALUE_TIMESTAMP",
        "queueSize": integer,
        "publishingIntervalMilliseconds": integer,
        "snapshotFrequencyMilliseconds": integer
    }
}
]
}
]
}

```

Anforderungstext

sources

Eine Liste von OPC UA-Quelldefinitionsstrukturen, die jeweils die folgenden Informationen enthalten:

name

Ein eindeutiger und aussagekräftiger Name für die Quelle.

endpoint

Eine Endpunktstruktur, die die folgenden Informationen enthält:

certificateTrust

Eine Zertifikatvertrauensrichtlinienstruktur, die die folgenden Informationen enthält:

type

Der Zertifikatvertrauensmodus für die Quelle. Wählen Sie eine der folgenden Optionen:

- **TrustAny**— Das SiteWise Edge-Gateway vertraut jedem Zertifikat, wenn es eine Verbindung zur OPC UA-Quelle herstellt.
- **X509**— Das SiteWise Edge-Gateway vertraut einem X.509 Zertifikat, wenn es eine Verbindung zur OPC UA-Quelle herstellt. Wenn Sie diese Option wählen, müssen Sie **certificateBody** in **certificateTrust** definieren. Sie können auch **certificateChain** in **certificateTrust** definieren.

certificateBody

(Optional) Der Hauptteil eines X.509 Zertifikats.

Dieses Feld ist erforderlich, wenn Sie X509 für type in certificateTrust auswählen.

certificateChain

(Optional) Die Vertrauenskette für ein X.509 Zertifikat.

Dieses Feld wird nur verwendet, wenn Sie X509 für type in certificateTrust auswählen.

endpointUri

Der lokale Endpunkt der OPC UA-Quelle. Der lokale Endpunkt könnte z. B. wie `opc.tcp://203.0.113.0:49320` aussehen.

securityPolicy

Die Sicherheitsrichtlinie, die verwendet werden soll, damit Sie Nachrichten schützen können, die aus der OPC UA-Quelle gelesen werden. Wählen Sie eine der folgenden Optionen:

- **NONE**— Das SiteWise Edge-Gateway schützt keine Nachrichten von der OPC UA-Quelle. Wir empfehlen Ihnen, eine andere Sicherheitsrichtlinie zu wählen. Wenn Sie diese Option wählen, müssen Sie auch NONE für messageSecurityMode auswählen.
- **BASIC256_SHA256**— Die Basic256Sha256 Sicherheitsrichtlinie.
- **AES128_SHA256_RSAPSS**— Die Aes128_Sha256_RsaPss Sicherheitspolitik.
- **AES256_SHA256_RSAPSS**— Die Aes256_Sha256_RsaPss Sicherheitspolitik.
- **BASIC128_RSA15**— (Veraltet) Die Basic128Rsa15 Sicherheitsrichtlinie ist in der OPC UA-Spezifikation veraltet, da sie nicht mehr als sicher gilt. Wir empfehlen Ihnen, eine andere Sicherheitsrichtlinie zu wählen. Weitere Informationen finden Sie unter [Profil SecurityPolicy — Basic128Rsa15](#).
- **BASIC256**— (Veraltet) Die Basic256 Sicherheitsrichtlinie ist in der OPC UA-Spezifikation veraltet, da sie nicht mehr als sicher gilt. Wir empfehlen Ihnen, eine andere Sicherheitsrichtlinie zu wählen. Weitere Informationen finden Sie unter [SecurityPolicy — Basic256](#).

 Important

Wenn Sie eine andere Sicherheitsrichtlinie als wählen NONE, müssen Sie SIGN oder SIGN_AND_ENCRYPT für wählen. `messageSecurityMode` Sie müssen Ihren Quellserver auch so konfigurieren, dass er dem SiteWise Edge-Gateway vertraut. Weitere Informationen finden Sie unter [Richten Sie OPC UA-Server so ein, dass sie dem vertrauen AWS IoT SiteWise Edge-Gateway](#).

`messageSecurityMode`

Der Sicherheitsmodus für Nachrichten, der verwendet werden soll, um Verbindungen zur OPC UA-Quelle zu sichern. Wählen Sie eine der folgenden Optionen:

- NONE— Das SiteWise Edge-Gateway sichert keine Verbindungen zur OPC UA-Quelle. Wir empfehlen, dass Sie einen anderen Nachrichtensicherheitsmodus wählen. Wenn Sie diese Option wählen, müssen Sie auch NONE für `securityPolicy` auswählen.
- SIGN— Daten, die zwischen dem SiteWise Edge-Gateway und der OPC UA-Quelle übertragen werden, sind signiert, aber nicht verschlüsselt.
- SIGN_AND_ENCRYPT— Daten, die zwischen dem Gateway und der OPC UA-Quelle übertragen werden, sind signiert und verschlüsselt.

 Important

Wenn Sie einen anderen Nachrichtensicherheitsmodus als wählen NONE, müssen Sie einen `securityPolicy` anderen als NONE wählen. Sie müssen Ihren Quellserver auch so konfigurieren, dass er dem SiteWise Edge-Gateway vertraut. Weitere Informationen finden Sie unter [Richten Sie OPC UA-Server so ein, dass sie dem vertrauen AWS IoT SiteWise Edge-Gateway](#).

`identityProvider`

Eine Identitätsanbieterstruktur, die die folgenden Informationen enthält:

`type`

Der Typ der Authentifizierungsanmeldeinformationen, die von der Quelle erfordert werden. Wählen Sie eine der folgenden Optionen:

- **Anonymous**— Die Quelle benötigt keine Authentifizierung, um eine Verbindung herzustellen.
- **Username**— Die Quelle benötigt einen Benutzernamen und ein Passwort, um eine Verbindung herzustellen. Wenn Sie diese Option wählen, müssen Sie `usernameSecretArn` in `identityProvider` definieren.

`usernameSecretArn`

(Optional) Der ARN eines AWS Secrets Manager Geheimnisses. Das SiteWise Edge-Gateway verwendet die Authentifizierungsanmeldeinformationen in diesem geheimen Schlüssel, wenn es eine Verbindung zu dieser Quelle herstellt. Sie müssen Geheimnisse an den SiteWise IoT-Connector Ihres SiteWise Edge-Gateways anhängen, um sie für die Quellauthentifizierung zu verwenden. Weitere Informationen finden Sie unter [Konfigurieren Sie die Datenquellenauthentifizierung für SiteWise Edge](#).

Dieses Feld ist erforderlich, wenn Sie `Username` für `type` in `identityProvider` auswählen.

`nodeFilterRules`

Eine Liste von Knotenfilterregelstrukturen, die die OPC UA-Datenstream-Pfade definieren, die an die AWS Cloud gesendet werden sollen. Sie können Knotenfilter verwenden, um die Startzeit und die CPU-Auslastung Ihres SiteWise Edge-Gateways zu reduzieren, indem Sie nur Pfade zu Daten einbeziehen, die Sie modellieren. AWS IoT SiteWise Standardmäßig laden SiteWise Edge-Gateways alle OPC UA-Pfade hoch, außer denen, die mit `beginnen. /Server/` Um OPC UA-Knotenfilter zu definieren, können Sie Knotenpfade und die `*` Platzhalterzeichen und verwenden. `**` Weitere Informationen finden Sie unter [Verwenden Sie OPC UA-Knotenfilter in Edge SiteWise](#).

Jede Struktur in der Liste muss folgende Informationen enthalten:

`action`

Die Aktion für diese Knotenfilterregel. Sie können die folgenden Optionen auswählen:

- **INCLUDE**— Das SiteWise Edge-Gateway umfasst nur Datenströme, die dieser Regel entsprechen.

`definition`

Eine Knotenfilterregelstruktur, die die folgenden Informationen enthält:

type

Der Typ des Knotenfilterpfads für diese Regel. Sie können die folgenden Optionen auswählen:

- **OpcUaRootPath**— Das SiteWise Edge-Gateway bewertet diesen Knotenfilterpfad anhand des Stammverzeichnisses der OPC-UA-Pfadhierarchie.

rootPath

Der Knotenfilterpfad, der anhand der Wurzel der OPC UA-Pfadhierarchie ausgewertet werden soll. Dieser Pfad muss mit / beginnen.

measurementDataStreamPrefix

Eine Zeichenfolge, die allen Datenströmen aus der Quelle vorangestellt wird. Das SiteWise Edge-Gateway fügt dieses Präfix allen Datenströmen aus dieser Quelle hinzu. Verwenden Sie ein Datenstrom-Präfix, um zwischen Datenströmen mit demselben Namen aus verschiedenen Quellen zu unterscheiden. Jeder Datenstrom sollte einen eindeutigen Namen in Ihrem Konto haben.

typeConversions

Die Konvertierungstypen, die für nicht unterstützte OPC UA-Datentypen verfügbar sind. Jeder Datentyp wird in Zeichenketten konvertiert. Weitere Informationen finden Sie unter [Nicht unterstützte Datentypen werden konvertiert](#).

array

Der einfache Array-Datentyp, der in Zeichenketten konvertiert wird. Sie können die folgenden Optionen auswählen:

- **JsonArray**— Zeigt an, dass Sie Ihre einfachen Array-Datentypen in Zeichenketten konvertieren möchten.

datetime

Der DateTime Datentyp, der in Zeichenketten konvertiert wird. Sie können die folgenden Optionen auswählen:

- **ISO8601String**— Zeigt an, dass Sie ISO 8601-Datentypen in Zeichenketten konvertieren möchten.

destination

Konfiguration für das Ziel von OPC UA-Tags. Klassische Stream-, v2- und MQTT-enabled V3-Gateways haben unterschiedliche Konfigurationen für Ziele.

type

Der Typ des Ziels.

`streamName`— nur für Classic-Streams, V2-Gateways

Name des -Streams. Der Streamname sollte eindeutig sein.

`streamBufferSize`— nur für Classic-Streams, V2-Gateways

Die Puffergröße des Streams. Dies ist wichtig für die Verwaltung des Datenflusses aus OPC UA-Quellen.

`defaultPropertyGroupConfig`— MQTT-enabled, nur V3-Gateways

(Optional) Konfiguration für die Standard-Eigenschaftsgruppe. Die Standard-Eigenschaftsgruppe enthält alle Knoten, die sonst nicht in einer benutzerdefinierten Eigenschaftsgruppe enthalten sind.

`dataChangeTrigger`

Der Standardauslöser für Datenänderungen, der in der Standard-Eigenschaftengruppe verwendet werden soll. Gültige Werte sind `STATUS_VALUE_TIMESTAMP`, `STATUS_VALUE` oder `STATUS`.

Note

`defaultPropertyGroupConfig` erfordert Version 3.1.0 oder höher der IoT SiteWise OPC UA-Kollektorkomponente. Weitere Informationen finden Sie unter [Aktualisieren Sie die Version einer Komponente AWS IoT SiteWise](#).

`discoveryConfig`— MQTT-enabled, nur V3-Gateways

(Optional) Konfiguration für den OPC UA-Knotenerkennungsprozess.

`enableNodeTraversal`

Gibt an, ob weiterhin untergeordnete Knoten des Stammknotens durchsucht werden sollen, der durch den Knotenfilter der Datenquelle definiert ist. Wenn diese Option auf `false` gesetzt ist, stoppt die Erkennung am Stammknoten.

`avoidNodeTreeLoops`

Gibt an, ob beim Durchsuchen des OPC UA-Knotens Endlosschleifen vermieden werden sollen. Wenn diese Option auf `true` gesetzt ist, verfolgt das Gateway besuchte Knoten, um Zirkelverweise zu verhindern.

`maxConcurrentBrowseRequests`

Die maximale Anzahl gleichzeitiger Suchanfragen, die Ihr OPC UA-Server gleichzeitig verarbeiten kann. Der gültige Bereich liegt zwischen 1 und 500.

`maxNodesPerBrowseRequest`

Die maximale Anzahl von Knoten, die in jeder Suchanforderung an den OPC UA-Server gesendet werden sollen. Der gültige Bereich liegt zwischen 1 und 1.000.

`periodicDiscovery`

Konfiguration für die regelmäßige Ausführung der Erkennung in festen Intervallen. Die regelmäßige Erkennung ist aktiviert, wenn diese Konfiguration bereitgestellt wird.

`interval`

Die Zeitspanne zwischen den periodischen Erkennungsvorgängen. Sie können es `m` für Minuten, `h` Stunden und `d` Tage verwenden. Zum Beispiel `90m` oder `1h`. Das Mindestintervall beträgt 30 Sekunden.

`maxNodesDiscoveredPerInterval`

Die maximale Anzahl von Knoten, die pro Erkennungsintervall entdeckt werden sollten. Dies hilft, die Last sowohl auf dem Gateway als auch auf dem OPC UA-Server zu kontrollieren.

Note

`periodicDiscovery` erfordert Version 3.1.0 oder höher der IoT SiteWise OPC UA-Kollektorkomponente. Weitere Informationen finden Sie unter [Aktualisieren Sie die Version einer Komponente AWS IoT SiteWise](#).

 Note

Wenn sich die Erkennung endlos wiederholt, aktivieren Sie diese Option.
`avoidNodeTreeLoops` Überwachen Sie den Fortschritt der Erkennung in den CloudWatch Protokollen unter der `aws.iot.SiteWiseOpcUaCollector` Komponente.

propertyGroups

(Optional) Die Liste der Eigenschaftsgruppen, die das Protokoll definieren deadband und vom Protokoll `scanMode` angefordert werden.

name

Der Name der Eigenschaftsgruppe. Dies sollte ein eindeutiger Bezeichner sein.

deadband

Der deadband Wert definiert die minimale Änderung des Werts eines Datenpunkts, die eintreten muss, bevor die Daten an die Cloud gesendet werden. Sie umfasst die folgenden Informationen:

type

Die unterstützten Typen von Deadband. Sie können die folgenden Optionen auswählen:

- **ABSOLUTE**— Ein fester Wert, der die minimale absolute Änderung angibt, die erforderlich ist, um einen Datenpunkt als signifikant genug anzusehen, um an die Cloud gesendet zu werden.
- **PERCENT**— Ein dynamischer Wert, der die minimal erforderliche Änderung als Prozentsatz des Werts des zuletzt gesendeten Datenpunkts angibt. Diese Art von Totband ist nützlich, wenn die Datenwerte im Laufe der Zeit stark variieren.

value

Der Wert des Totbandes. Wenn `type` `jaABSOLUTE`, ist dieser Wert ein Double ohne Einheit. Wenn `type` `jaPERCENT`, ist dieser Wert ein Doppelter zwischen 1 und 100.

eguMin

(Optional) Die Mindestanzahl an technischen Einheiten, wenn ein PERCENT Totband verwendet wird. Sie legen dies fest, wenn auf dem OPC UA-Server keine technischen Einheiten konfiguriert sind.

eguMax

(Optional) Die maximale Anzahl an technischen Einheiten, wenn ein PERCENT Deadband verwendet wird. Sie legen dies fest, wenn auf dem OPC UA-Server keine technischen Einheiten konfiguriert sind.

timeoutMilliseconds

Die Dauer in Millisekunden vor dem Timeout. Das Minimum ist. 100

scanMode

Die scanMode Struktur, die die folgenden Informationen enthält:

type

Die unterstützten Typen von scanMode. Zulässige Werte sind POLL und EXCEPTION.

rate

Das Abtastintervall für den Scanmodus.

timestampToReturn

Die Quelle des Zeitstempels. Sie können die folgenden Optionen auswählen:

- SOURCE_TIME— Verwendet den Zeitstempel von Ihrem Gerät.
- SERVER_TIME— Verwendet den Zeitstempel von Ihrem Server.

Note

Verwendung TimestampToReturn mit Version 2.5.0 oder höher der IoT SiteWise OPC UA-Collector-Komponente. Wenn Sie diese Funktion mit früheren Versionen verwenden, schlagen Konfigurationsupdates fehl. Weitere Informationen finden Sie unter [Aktualisieren Sie die Version einer Komponente AWS IoT SiteWise](#).

nodeFilterRuleDefinitions

(Optional) Eine Liste von Knotenpfaden, die in die Eigenschaftengruppe aufgenommen werden sollen. Eigenschaftsgruppen dürfen sich nicht überschneiden. Wenn Sie keinen Wert für dieses Feld angeben, enthält die Gruppe alle Pfade unter dem Stamm, und Sie können keine zusätzlichen Eigenschaftsgruppen erstellen. Die nodeFilterRuleDefinitions-Struktur enthält folgende Informationen:

type

OpcUaRootPathist der einzige unterstützte Typ. Dies gibt an, dass der Wert von rootPath ein Pfad relativ zum Stammverzeichnis des OPC UA-Browsingbereichs ist.

rootPath

Eine durch Kommas getrennte Liste, die die Pfade (relativ zum Stamm) angibt, die in die Eigenschaftsgruppe aufgenommen werden sollen.

Beispiele für zusätzliche Funktionskonfigurationen für Classic-Streams, V2-Gateways (AWS CLI)

Das folgende Beispiel definiert eine OPC UA SiteWise Edge-Gateway-Funktionskonfiguration anhand einer in einer JSON-Datei gespeicherten Nutzlast.

```
aws iotsitewise update-gateway-capability-configuration \
--capability-namespace "iotsitewise:opcuacollector:2" \
--capability-configuration file://opc-ua-configuration.json
```

Example: OPC UA-Quellkonfiguration

Die folgende `opc-ua-configuration.json` Datei definiert eine grundlegende, unsichere OPC UA-Quellkonfiguration.

```
{
  "sources": [
    {
      "name": "Wind Farm #1",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny"
        },
        "endpointUri": "opc.tcp://203.0.113.0:49320",
        "securityPolicy": "NONE",
        "messageSecurityMode": "NONE",
        "identityProvider": {
          "type": "Anonymous"
        },
        "nodeFilterRules": []
      },
      "measurementDataStreamPrefix": ""
    }
  ]
}
```

```
}
```

Example: OPC UA-Quellkonfiguration mit definierten Eigenschaftsgruppen

Die folgende `opc-ua-configuration.json` Datei definiert eine grundlegende, unsichere OPC UA-Quellkonfiguration mit definierten Eigenschaftsgruppen.

```
{
  "sources": [
    {
      "name": "source1",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny"
        },
        "endpointUri": "opc.tcp://10.0.0.9:49320",
        "securityPolicy": "NONE",
        "messageSecurityMode": "NONE",
        "identityProvider": {
          "type": "Anonymous"
        },
        "nodeFilterRules": [
          {
            "action": "INCLUDE",
            "definition": {
              "type": "OpcUaRootPath",
              "rootPath": "/Utilities/Tank"
            }
          }
        ]
      },
      "measurementDataStreamPrefix": "propertyGroups",
      "propertyGroups": [
        {
          "name": "Deadband_Abs_5",
          "nodeFilterRuleDefinitions": [
            {
              "type": "OpcUaRootPath",
              "rootPath": "/Utilities/Tank/Temperature/TT-001"
            },
            {
              "type": "OpcUaRootPath",
              "rootPath": "/Utilities/Tank/Temperature/TT-002"
            }
          ]
        }
      ]
    }
  ]
}
```

```

    }
  ],
  "deadband": {
    "type": "ABSOLUTE",
    "value": 5.0,
    "timeoutMilliseconds": 120000
  }
},
{
  "name": "Polling_10s",
  "nodeFilterRuleDefinitions": [
    {
      "type": "OpcUaRootPath",
      "rootPath": "/Utilities/Tank/Pressure/PT-001"
    }
  ],
  "scanMode": {
    "type": "POLL",
    "rate": 10000
  }
},
{
  "name": "Percent_Deadband_Timeout_90s",
  "nodeFilterRuleDefinitions": [
    {
      "type": "OpcUaRootPath",
      "rootPath": "/Utilities/Tank/Flow/FT-*"
    }
  ],
  "deadband": {
    "type": "PERCENT",
    "value": 5.0,
    "eguMin": -100,
    "eguMax": 100,
    "timeoutMilliseconds": 90000
  }
}
]
}
]
}

```


Example: OPC UA-Quellkonfiguration mit Eigenschaften

Das folgende JSON-Beispiel für `opc-ua-configuration.json` definiert eine OPC UA-Quellkonfiguration mit den folgenden Eigenschaften:

- Vertraut jedem Zertifikat.
- Verwendet die BASIC256 Sicherheitsrichtlinie, um Nachrichten zu sichern.
- Verwendet den SIGN_AND_ENCRYPT-Modus zum Sichern von Verbindungen.
- Verwendet Authentifizierungsdaten, die in einem Secrets Manager Manager-Secret gespeichert sind.
- Filtert Datenströme außer denjenigen heraus, deren Pfad mit `/WindFarm/2/WindTurbine/` beginnt.
- Fügt `/Washington` am Anfang jedes Datenstrompfades hinzu, um zwischen diesem „Windpark #2“ und einem „Windpark #2“ in einem anderen Bereich zu unterscheiden.

```
{
  "sources": [
    {
      "name": "Wind Farm #2",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny"
        },
        "endpointUri": "opc.tcp://203.0.113.1:49320",
        "securityPolicy": "BASIC256",
        "messageSecurityMode": "SIGN_AND_ENCRYPT",
        "identityProvider": {
          "type": "Username",
          "usernameSecretArn":
            "arn:aws:secretsmanager:region:123456789012:secret:greenrass-windfarm2-auth-1ABCDE"
        },
        "nodeFilterRules": [
          {
            "action": "INCLUDE",
            "definition": {
              "type": "OpcUaRootPath",
              "rootPath": "/WindFarm/2/WindTurbine/"
            }
          }
        ]
      }
    }
  ]
}
```

```

    },
    "measurementDataStreamPrefix": "/Washington"
  }
]
}

```

Example: OPC UA-Quellkonfiguration mit Zertifikatsvertrauen

Das folgende JSON-Beispiel für `opc-ua-configuration.json` definiert eine OPC UA-Quellkonfiguration mit den folgenden Eigenschaften:

- Vertraut einem bestimmten X.509 Zertifikat.
- Verwendet die BASIC256 Sicherheitsrichtlinie, um Nachrichten zu sichern.
- Verwendet den SIGN_AND_ENCRYPT-Modus zum Sichern von Verbindungen.

```

{
  "sources": [
    {
      "name": "Wind Farm #3",
      "endpoint": {
        "certificateTrust": {
          "type": "X509",
          "certificateBody": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w
0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZ
WF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBA5TC0lBTsBDb25zb2x1MRIw
EAYDVQQDEw1UZXR0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvbi5
jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTEwNDI1MjA0NTIxWjCBiDELMAkGA1UEBh
MCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBb
WF6b24xFDASBgNVBA5TC0lBTsBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWMx
HzAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvbi5jb20wgZ8wDQYJKoZIhvcNAQE
BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLyGVI
k60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
ITx0USQv7c7ugFFDzQGBzZswY6786m86gpEIbb30hjZnzcvcQAaRHhd1QWIMm2nr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEATCu4nUhVVxYUntneD9+h8Mg9q6q+auN
KyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6Guo
EDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjSTbNYiytVbZPQUQ5Yaxu2jXnimvw
3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
          "certificateChain": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w

```

```

0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZ
WF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBASTC0lBTSBDb25zb2x1MRIw
EAYDVQQDEw1UZXR0Q21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvi5
jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTEwNDI1MjA0NTIxWjCBiDELMAkGA1UEBh
MCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBb
WF6b24xFDASBgNVBASTC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWMx
HzAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvi5jb20wgZ8wDQYJKoZIhvcNAQE
BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLygVI
k60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
ITx0USQv7c7ugFFDzQGBzZswY6786m86gpEIbb30hjZnzcVQAaRHhd1QWIMm2nr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4nUhVVxYUntneD9+h8Mg9q6q+auN
KyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6Guo
EDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjSTbNYiytVbZPQUQ5Yaxu2jXnimvw
3rrszlaEXAMPLE=
-----END CERTIFICATE-----"
    },
    "endpointUri": "opc.tcp://203.0.113.2:49320",
    "securityPolicy": "BASIC256",
    "messageSecurityMode": "SIGN_AND_ENCRYPT",
    "identityProvider": {
        "type": "Anonymous"
    },
    "nodeFilterRules": []
  },
  "measurementDataStreamPrefix": ""
}
]
}

```

Richten Sie OPC UA-Server so ein, dass sie dem vertrauen AWS IoT SiteWise Edge-Gateway

Wenn Sie bei der Konfiguration Ihrer OPC UA-Quelle eine `messageSecurityMode` andere Option als Keine wählen, müssen Sie Ihren Quellservern ermöglichen, dem AWS IoT SiteWise Edge-Gateway zu vertrauen. Das SiteWise Edge-Gateway generiert ein Zertifikat, das Ihr Quellserver möglicherweise benötigt. Der Prozess ist je nach Ihren Quellservern unterschiedlich. Weitere Informationen finden Sie in der Dokumentation zu Ihren Servern.

Das folgende Verfahren beschreibt die grundlegenden Schritte.

Um einem OPC UA-Server zu ermöglichen, dem SiteWise Edge-Gateway zu vertrauen

1. Öffnen Sie die Schnittstelle für die Konfiguration Ihres OPC UA-Servers.

2. Geben Sie den Benutzernamen und das Passwort für den OPC UA-Serveradministrator ein.
3. Suchen Sie auf der Benutzeroberfläche nach Trusted Clients (Vertrauenswürdige Clients), und wählen Sie dann AWS IoT SiteWise Gateway Client aus.
4. Wählen Sie Trust (Vertrauensstellung) aus.

Exportieren des OPC UA-Client-Zertifikats

Einige OPC UA-Server benötigen Zugriff auf die OPC UA-Client-Zertifikatsdatei, um dem SiteWise Edge-Gateway zu vertrauen. Wenn dies auf Ihre OPC UA-Server zutrifft, können Sie das OPC UA-Client-Zertifikat mit dem folgenden Verfahren vom Edge-Gateway exportieren. SiteWise Anschließend können Sie das Zertifikat auf Ihren OPC UA-Server importieren.

Um die OPC UA-Client-Zertifikatsdatei für eine Quelle zu exportieren

1. Führen Sie den folgenden Befehl aus, um in das Verzeichnis zu wechseln, das die Zertifikatsdatei enthält. *sitewise-work* Ersetzen Sie durch den lokalen Speicherpfad für den *aws.iot.SiteWiseEdgeCollectorOpcua* Greengrass-Arbeitsordner und *source-name* ersetzen Sie ihn durch den Namen der Datenquelle.

Standardmäßig befindet */greengrass/v2/work/* *aws.iot.SiteWiseEdgeCollectorOpcua* sich der Greengrass-Arbeitsordner unter Linux und *C:/greengrass/v2/work/aws.iot.SiteWiseEdgeCollectorOpcua* auf Microsoft Windows.

```
cd /sitewise-work/source-name/opcua-certificate-store
```

2. Das OPC UA-Client-Zertifikat des SiteWise Edge-Gateways für diese Quelle befindet sich in der *aws-iot-opcua-client.pfx* Datei.

Führen Sie den folgenden Befehl aus, um das Zertifikat in eine *.pem*-Datei namens *aws-iot-opcua-client-certificate.pem* zu exportieren.

```
keytool -exportcert -v -alias aws-iot-opcua-client -keystore aws-iot-opcua-client.pfx -storepass amazon -storetype PKCS12 -rfc > aws-iot-opcua-client-certificate.pem
```

3. Übertragen Sie die Zertifikatsdatei, *aws-iot-opcua-client-certificate.pem*, vom SiteWise Edge-Gateway auf den OPC UA-Server.

Dazu können Sie gängige Software wie das scp Programm verwenden, um die Datei mit dem SSH-Protokoll zu übertragen. Weitere Informationen finden Sie unter [Sichere Kopie](#) auf Wikipedia.

 Note

Wenn Ihr SiteWise Edge-Gateway auf Amazon Elastic Compute Cloud (Amazon EC2) läuft und Sie sich zum ersten Mal damit verbinden, müssen Sie die Voraussetzungen für die Verbindung konfigurieren. Weitere Informationen finden Sie unter [Herstellen einer Verbindung mit Ihrer Linux-Instance mit SSH](#) im Amazon-EC2-Benutzerhandbuch.

4. Importieren Sie die Zertifikatsdatei `aws-iot-opcua-client-certificate.pem`, auf den OPC UA-Server, um dem SiteWise Edge-Gateway zu vertrauen. Die Schritte sind von den von Ihnen verwendeten Quellservern abhängig. Weitere Informationen finden Sie in der Dokumentation für den Server.

Filtern Sie Datenaufnahmebereiche mit OPC UA

Sie können die Art und Weise steuern, wie Sie Daten mit einer OPC UA-Quelle aufnehmen, indem Sie den Scanmodus und die Deadband-Bereiche verwenden. Mit diesen Funktionen können Sie steuern, welche Art von Daten aufgenommen werden sollen und wie und wann Ihr Server und das SiteWise Edge-Gateway diese Informationen austauschen.

Erfassen oder filtern Sie Daten auf der Grundlage der Qualität

Sie können Ihre Datenqualitätseinstellungen so konfigurieren, dass Sie steuern, welche Daten aus der OPC UA-Quelle gesammelt werden. Die Datenquelle enthält die Qualitätsbewertung als Metadaten, wenn sie gesendet wird. Sie können eine oder alle der folgenden Optionen auswählen:

- Good
- Bad
- Uncertain

Behandle NaN- oder Nullwerte

SiteWise Edge unterstützt die Erfassung und Verarbeitung von NaN- und Nullwerten.

- NaN (Not a Number): Stellt undefinierte oder nicht darstellbare numerische Ergebnisse dar.

- Null: Weist auf fehlende Daten hin.

Der SiteWise IoT-OPC-UA-Kollektor erfasst NaN- und Nullwerte mit SCHLECHTER oder UNSICHERER Qualität. Diese speziellen Werte werden in den lokalen Stream geschrieben, was eine umfassendere Datenerfassung ermöglicht.

Steuern Sie die Häufigkeit der Datenerfassung im Scanmodus

Sie können Ihren OPC UA-Scanmodus so konfigurieren, dass er steuert, wie Sie Daten aus Ihrer OPC UA-Quelle sammeln. Sie können den Abonnement- oder Abfragemodus wählen.

- Abonnementmodus — Die OPC UA-Quelle sammelt Daten, die mit der durch Ihre Scanrate definierten Frequenz an Ihr SiteWise Edge-Gateway gesendet werden. Der Server sendet nur Daten, wenn sich der Wert geändert hat. Dies ist also die maximale Frequenz, mit der Ihr SiteWise Edge-Gateway Daten empfängt.
- Abfragemodus — Ihr SiteWise Edge-Gateway fragt die OPC UA-Quelle mit einer festgelegten Frequenz ab, die durch Ihre Scanrate definiert wird. Der Server sendet Daten unabhängig davon, ob sich der Wert geändert hat, sodass Ihr SiteWise Edge-Gateway immer Daten in diesem Intervall empfängt.

 Note

Die Option für den Abfragemodus überschreibt Ihre Deadband-Einstellungen für diese Quelle.

Filtern Sie die OPC UA-Datenaufnahme anhand von Totbandbereichen

Sie können ein Deadband auf Ihre OPC UA-Quelleeigenschaftsgruppen anwenden, um bestimmte Daten herauszufiltern und zu verwerfen, anstatt sie an die Cloud zu senden. AWS Ein Deadband gibt ein Zeitfenster an, in dem zu erwartende Schwankungen der eingehenden Datenwerte aus Ihrer OPC UA-Quelle zu erwarten sind. Wenn die Werte in dieses Fenster fallen, sendet Ihr OPC UA-Server sie nicht an die Cloud. AWS Sie können die Deadband-Filterung verwenden, um die Datenmenge zu reduzieren, die Sie verarbeiten und an die AWS Cloud senden. Informationen zum Einrichten von OPC UA-Quellen für Ihr SiteWise Edge-Gateway finden Sie unter. [OPC UA-Datenquellen für AWS IoT SiteWise Edge-Gateways](#)

 Note

Ihr Server löscht alle Daten, die in das durch Ihr Deadband angegebene Fenster fallen. Sie können diese verworfenen Daten nicht wiederherstellen.

Arten von Deadbands

Sie können zwei Arten von Deadbands für Ihre OPC UA-Servereigenschaftsgruppe angeben. Mit diesen können Sie wählen, wie viele Daten an die AWS Cloud gesendet und wie viele verworfen werden.

- **Prozentsatz** — Sie geben ein Zeitfenster an, in dem ein Prozentsatz der zu erwartenden Fluktuation des Messwerts verwendet wird. Der Server berechnet anhand dieses Prozentsatzes das genaue Zeitfenster und sendet Daten an die AWS Cloud, wenn der Wert außerhalb des Fensters liegt. Wenn Sie beispielsweise einen Totbandwert von 2% für einen Sensor mit einem Bereich von -100 Grad Fahrenheit bis +100 Grad Fahrenheit angeben, wird der Server angewiesen, Daten an die AWS Cloud zu senden, wenn sich der Wert um 4 Grad Fahrenheit oder mehr ändert.

 Note

Sie können optional einen minimalen und einen maximalen Totbandwert für dieses Fenster angeben, wenn Ihr Quellserver keine technischen Einheiten definiert. Wenn kein Bereich für technische Einheiten angegeben wird, verwendet der OPC UA-Server standardmäßig den gesamten Bereich des Messdatentyps.

- **Absolut** — Sie geben ein Fenster mit exakten Einheiten an. Wenn Sie beispielsweise einen Totbandwert von 2 für einen Sensor angeben, wird der Server angewiesen, Daten an die AWS Cloud zu senden, wenn sich der Wert um mindestens 2 Einheiten ändert. Sie können absolutes Deadbanding für dynamische Umgebungen verwenden, in denen während des normalen Betriebs regelmäßig mit Schwankungen zu rechnen ist.

Deadband-Timeouts

Sie können optional eine Deadband-Timeout-Einstellung konfigurieren. Nach diesem Timeout sendet der OPC UA-Server den aktuellen Messwert, auch wenn dieser innerhalb der erwarteten Deadband-Fluktuation liegt. Sie können die Timeout-Einstellung verwenden, um sicherzustellen, dass jederzeit

ein stetiger Datenstrom aufgenommen AWS IoT SiteWise wird, auch wenn die Werte das definierte Deadband-Fenster nicht überschreiten.

Verwenden Sie OPC UA-Knotenfilter in Edge SiteWise

Wenn Sie OPC UA-Datenquellen für ein SiteWise Edge-Gateway definieren, können Sie Knotenfilter definieren. Mit Knotenfiltern können Sie einschränken, welche Datenstream-Pfade das SiteWise Edge-Gateway an die Cloud sendet. Sie können Knotenfilter verwenden, um die Startzeit und die CPU-Auslastung Ihres SiteWise Edge-Gateways zu reduzieren, indem Sie nur Pfade zu Daten einbeziehen, die Sie modellieren AWS IoT SiteWise. Standardmäßig laden SiteWise Edge-Gateways alle OPC UA-Pfade hoch, außer denen, die mit beginnen. /Server/ Sie können die Platzhalterzeichen * und ** in den Knotenfiltern verwenden, um mehrere Daten-Stream-Pfade mit einem Filter einzuschließen. Informationen zum Einrichten von OPC UA-Quellen für Ihr SiteWise Edge-Gateway finden Sie unter. [OPC UA-Datenquellen für AWS IoT SiteWise Edge-Gateways](#)

Note

AWS IoT SiteWise startet Ihr SiteWise Edge-Gateway jedes Mal neu, wenn Sie eine Quelle hinzufügen oder bearbeiten. Ihr SiteWise Edge-Gateway nimmt keine Daten auf, während es die Quellkonfiguration aktualisiert. Die Zeit für den Neustart Ihres SiteWise Edge-Gateways hängt von der Anzahl der Tags in den Quellen Ihres SiteWise Edge-Gateways ab. Die Neustartzeit kann zwischen einigen Sekunden (für ein SiteWise Edge-Gateway mit wenigen Tags) und mehreren Minuten (für ein SiteWise Edge-Gateway mit vielen Tags) liegen.

In der folgenden Tabelle sind die Platzhalter aufgeführt, die Sie zum Filtern von OPC UA-Datenquellen verwenden können.

Platzhalter für den OPC UA-Knotenfilter

Platzhalter	Description
*	Entspricht einer einzelnen Ebene in einem Daten-Stream-Pfad.
**	Entspricht mehreren Ebenen in einem Daten-Stream-Pfad.

 Note

Wenn Sie eine Quelle mit einem breiten Filter konfigurieren und die Quelle später ändern, sodass ein restriktiverer Filter verwendet wird, AWS IoT SiteWise werden keine Daten mehr gespeichert, die dem neuen Filter nicht entsprechen.

Example: Szenario mit Knotenfiltern

Sehen Sie sich beispielsweise die folgenden hypothetischen Daten-Streams an:

- `/WA/Factory 1/Line 1/PLC1`
- `/WA/Factory 1/Line 1/PLC2`
- `/WA/Factory 1/Line 2/Counter1`
- `/WA/Factory 1/Line 2/PLC1`
- `/OR/Factory 1/Line 1/PLC1`
- `/OR/Factory 1/Line 2/Counter2`

Mithilfe der vorherigen Datenströme können Sie Knotenfilter definieren, um einzuschränken, welche Daten aus Ihrer OPC UA-Quelle aufgenommen werden sollen.

- Um in diesem Beispiel alle Knoten auszuwählen, verwenden Sie `/ oder /**/`. Mit den Platzhalterzeichen `**` können Sie mehrere Verzeichnisse oder Ordner einschließen.
- Um alle PLC-Daten-Streams auszuwählen, verwenden Sie `/**/*.PLC*` oder `/**/PLC*`.
- Um in diesem Beispiel alle Leistungsindikatoren auszuwählen, verwenden Sie `/**/*.Counter*` oder `/**/*.Counter*`.
- Wenn Sie alle Zähler aus Line 2 auswählen möchten, verwenden Sie `/**/*.Line 2/Counter*`.

Nicht unterstützte Datentypen werden konvertiert

Aktivieren Sie optional die Datentypkonvertierung AWS IoT SiteWise für einfache Arrays und DateTime Datentypen. AWS IoT SiteWise unterstützt nicht alle OPC UA-Datentypen. Wenn Sie nicht unterstützte Daten an Ihren AWS IoT Greengrass Datenstrom senden, gehen diese Daten verloren. Wenn Sie jedoch die nicht unterstützten systemeigenen Datentypen in Zeichenfolgen konvertieren, können Sie die Daten in Zeichenketten aufnehmen, AWS IoT SiteWise anstatt sie zu verwerfen. AWS IoT SiteWise serialisiert Ihre konvertierten Daten, sodass Sie später Ihre eigenen Funktionen

verwenden können, um die Zeichenketten bei Bedarf wieder in ihren ursprünglichen Datentyp zu konvertieren.

Sie können Ihre Einstellungen für die Datentypkonvertierung für eine Datenquelle jederzeit aktualisieren, und jede Datenquelle kann ihre eigenen Einstellungen haben.

Wenn Sie Datenquellen hinzufügenAWS-IoT-SiteWise-Konsole, gibt es in der erweiterten Konfiguration unter Datentypkonvertierung zwei Kontrollkästchen. Sie können angeben, welche Datentypen in Zeichenfolgen konvertiert werden sollen.

Darüber hinaus kann der IoT SiteWise OPC UA-Collector NaN- oder Nullwerte am Edge akzeptieren.

- Konvertiert Array-Werte mit einfachen Datentypen in JSON-Zeichenketten
- Konvertiert DateTime Werte in ISO 8601-Zeichenketten

Voraussetzung

- Verwenden Sie Version 2.5.0 oder höher des [IoT SiteWise OPC UA](#) Collector.

Einschränkungen

Dies sind die Einschränkungen für die Konvertierung des OPC UA-Datentyps in Zeichenketten in AWS IoT SiteWise

- Die Konvertierung komplexer Datentypen wird nicht unterstützt.
- Die Zeichenkettenbegrenzung nach der Konvertierung beträgt 1024 Byte. Wenn die Zeichenfolge länger als 1024 Byte ist, wird die Zeichenfolge von zurückgewiesenAWS IoT SiteWise.

Konfigurieren Sie die Datenquellenauthentifizierung für SiteWise Edge

Wenn Ihr OPC UA-Server für die Verbindung Authentifizierungsdaten benötigt, können Sie diese verwenden, AWS Secrets Manager um ein Geheimnis zu erstellen und für Ihr SiteWise Edge-Gateway bereitzustellen. AWS Secrets Manager verschlüsselt Geheimnisse auf dem Gerät, um Ihren Benutzernamen und Ihr Passwort zu schützen, bis Sie sie verwenden müssen. Weitere Informationen zur AWS IoT Greengrass Secret Manager-Komponente finden Sie unter [Secret Manager](#) im AWS IoT Greengrass Version 2 Entwicklerhandbuch.

Informationen zur Verwaltung des Zugriffs auf Secrets Manager Manager-Geheimnisse finden Sie unter:

- [Wer hat Berechtigungen für Ihre AWS Secrets Manager Geheimnisse.](#)
- [Feststellen, ob eine Anfrage innerhalb eines Kontos zugelassen oder abgelehnt](#) wird.

Schritt 1: Geheimnisse für die Quellauthentifizierung erstellen

Sie können AWS Secrets Manager verwenden, um ein Authentifizierungsgeheimnis für Ihre Datenquelle zu erstellen. Definieren Sie im Secret Paare **username** und **password** Schlüssel-Wert-Paare, die Authentifizierungsdetails für Ihre Datenquelle enthalten.

Ein Secret erstellen (Konsole)

1. Navigieren Sie zur [AWS Secrets Manager -Konsole](#).
2. Wählen Sie Store a new secret (Ein neues Secret speichern).
3. Wählen Sie unter Geheimtyp die Option Andere Art von Geheimnissen aus.
4. Gehen Sie unter Schlüssel/Wert-Paare wie folgt vor:
 1. Geben Sie in das erste Eingabefeld **username** und im zweiten Eingabefeld den Benutzernamen ein.
 2. Wählen Sie Zeile hinzufügen.
 3. Geben Sie im ersten Eingabefeld das Passwort ein **password** und im zweiten Eingabefeld geben Sie das Passwort ein.
5. Wählen Sie als Verschlüsselungsschlüssel aws/secretsmanager und dann Weiter aus.
6. Geben Sie auf der Seite Neues Geheimnis speichern einen geheimen Namen ein.
7. (Optional) Geben Sie eine Beschreibung ein, anhand derer Sie dieses Geheimnis identifizieren können, und wählen Sie dann Weiter aus.
8. (Optional) Aktivieren Sie auf der Seite Neues Geheimnis speichern die Option Automatische Rotation. Weitere Informationen finden Sie im AWS Secrets Manager Benutzerhandbuch unter [Rotation von Geheimnissen](#).
9. Geben Sie einen Rotationsplan an.
10. Wählen Sie eine Lambda-Funktion aus, die dieses Geheimnis rotieren kann, und wählen Sie dann Weiter.
11. Überprüfen Sie Ihre geheimen Konfigurationen und wählen Sie dann Store aus.

Um Ihr SiteWise Edge-Gateway für die Interaktion zu autorisieren AWS Secrets Manager, muss die IAM-Rolle für Ihr SiteWise Edge-Gateway die `secretsmanager:GetSecretValue` Aktion

zulassen. Sie können das Greengrass-Core-Gerät verwenden, um nach der IAM-Richtlinie zu suchen. Weitere Informationen zur Aktualisierung einer IAM-Richtlinie finden Sie unter [Bearbeiten von IAM-Richtlinien im Benutzerhandbuch](#). AWS Identity and Access Management

Example policy

secret-arn Ersetzen Sie es durch den Amazon-Ressourcennamen (ARN) des Geheimnisses, das Sie im vorherigen Schritt erstellt haben. Weitere Informationen zum Abrufen des ARN eines [Secrets finden Sie unter Find Secrets AWS Secrets Manager in](#) im AWS Secrets Manager Benutzerhandbuch.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:secretsmanager:us-east-1:123456789012:secret/*"
      ]
    }
  ]
}
```

Schritt 2: Stellen Sie Geheimnisse auf Ihrem SiteWise Edge-Gateway-Gerät bereit

Sie können die AWS IoT SiteWise Konsole verwenden, um Geheimnisse auf Ihrem SiteWise Edge-Gateway bereitzustellen.

So stellen Sie ein Geheimnis bereit (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Gateways aus.
3. Wählen Sie aus der Gateways-Liste das SiteWise Edge-Ziel-Gateway aus.

4. Wählen Sie im Abschnitt Gateway-Konfiguration den Link Greengrass Core-Gerät aus, um den mit dem SiteWise Edge-Gateway verknüpften AWS IoT Greengrass Core zu öffnen.
5. Wählen Sie im Navigationsbereich Deployments aus.
6. Wählen Sie die Zielbereitstellung und dann Revise aus.
7. Wählen Sie auf der Seite „Ziel angeben“ die Option Weiter aus.
8. Deaktivieren Sie auf der Seite Komponenten auswählen im Abschnitt Öffentliche Komponenten die Option Nur ausgewählte Komponenten anzeigen.
9. Suchen Sie nach aws.greengrass und wählen Sie es aus. SecretManagerKomponente und wählen Sie dann Weiter.
10. Wählen Sie aus der Liste Ausgewählte Komponenten die Datei aws.greengrass aus. SecretManagerKomponente und wählen Sie dann Komponente konfigurieren.
11. Fügen Sie im Feld Konfiguration zum Zusammenführen das folgende JSON-Objekt hinzu.

 Note

secret-arn Ersetzen Sie es durch den ARN des Geheimnisses, das Sie im vorherigen Schritt erstellt haben. Weitere Informationen zum Abrufen des ARN eines [Secrets finden Sie unter Find Secrets AWS Secrets Manager in](#) im AWS Secrets Manager Benutzerhandbuch.

```
{
  "cloudSecrets":[
    {
      "arn":"secret-arn"
    }
  ]
}
```

12. Wählen Sie Bestätigen aus.
13. Wählen Sie Weiter aus.
14. Wählen Sie auf der Seite Erweiterte Einstellungen konfigurieren die Option Weiter aus.
15. Überprüfen Sie Ihre Bereitstellungsconfigurationen und wählen Sie dann Bereitstellen aus.

Schritt 3: Fügen Sie Authentifizierungskonfigurationen hinzu

Sie können die AWS IoT SiteWise Konsole verwenden, um Ihrem SiteWise Edge-Gateway Authentifizierungskonfigurationen hinzuzufügen.

Um Authentifizierungskonfigurationen hinzuzufügen (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie aus der Gateways-Liste das SiteWise Edge-Ziel-Gateway aus.
3. Wählen Sie aus der Liste Datenquellen die Zieldatenquelle aus, und klicken Sie dann auf Bearbeiten.
4. Wählen Sie auf der Seite Datenquelle hinzufügen die Option Erweiterte Konfiguration aus.
5. Wählen Sie für die Authentifizierungskonfiguration den geheimen Schlüssel aus, den Sie im vorherigen Schritt bereitgestellt haben.
6. Wählen Sie Speichern.

Partnerdatenquellen auf SiteWise Edge-Gateways

Wenn Sie ein AWS IoT SiteWise Edge-Gateway verwenden, können Sie eine Partnerdatenquelle mit Ihrem SiteWise Edge-Gateway verbinden und Daten vom Partner in Ihrem SiteWise Edge-Gateway und in der AWS Cloud empfangen. Bei diesen Partnerdatenquellen handelt es sich um AWS IoT Greengrass Komponenten, die in Partnerschaft zwischen dem Partner AWS und dem Partner entwickelt wurden. Wenn Sie eine Partnerdatenquelle hinzufügen, erstellt AWS IoT SiteWise diese Komponente und stellt sie auf Ihrem SiteWise Edge-Gateway bereit.

Note

Sie können für jeden Partner in jedem Gateway eine Datenquelle hinzufügen.

Gehen Sie wie folgt vor, um eine Partnerdatenquelle hinzuzufügen:

1. [Fügen Sie eine Partnerdatenquelle in SiteWise Edge hinzu](#)
2. Rufen Sie gegebenenfalls das Webportal des Partners auf und konfigurieren Sie die Partnerdatenquelle so, dass sie eine Verbindung zum SiteWise Edge-Gateway herstellt.

Themen

- [Sicherheit](#)
- [Einrichten Docker auf Ihrem SiteWise Edge-Gateway](#)
- [Fügen Sie eine Partnerdatenquelle in SiteWise Edge hinzu](#)
- [SiteWise Datenquellenoptionen für Edge-Gateway-Partner](#)

Sicherheit

Im Rahmen des [Modells der gemeinsamen Verantwortung](#) zwischen AWS unseren Kunden und unseren Partnern wird im Folgenden beschrieben, wer für die verschiedenen Sicherheitsaspekte verantwortlich ist:

Kundenverantwortung

- Überprüfung des Partners.
- Konfiguration des Netzwerkzugangs für den Partner.
- Überwachung der angemessenen Nutzung der Computerressourcen des SiteWise Edge-Gateways (CPU, Arbeitsspeicher und Dateisystem).

AWSVerantwortung

- Isolierung des Partners von den AWS Cloud-Ressourcen des Kunden, mit Ausnahme der Ressourcen, die der Partner benötigt. In diesem Fall die AWS IoT SiteWise Einnahme.
- Beschränkung der Partnerlösung auf eine angemessene Nutzung der Computerressourcen des SiteWise Edge-Gateways (CPU und Arbeitsspeicher).

Verantwortung des Partners

- Verwendung sicherer Standardeinstellungen.
- Sorgen Sie mit Patches und anderen geeigneten Updates dafür, dass die Lösung im Laufe der Zeit sicher bleibt.
- Vertrauliche Behandlung von Kundendaten.

Einrichten Docker auf Ihrem SiteWise Edge-Gateway

AWS IoT SiteWise stellt ein Docker Image bereit, mit dem Sie die SiteWise Edge-Anwendung auf verschiedenen Plattformen und Umgebungen ausführen können. Dieses Docker Image enthält alle erforderlichen Komponenten und Abhängigkeiten, die für die Erfassung, Verarbeitung und Übertragung von Daten aus Ihren Industrieanlagen an die AWS Cloud erforderlich sind. Mithilfe

des Docker-Images können Sie die SiteWise Edge-Anwendung auf Docker-compatible Hosts wie Servern, Edge-Geräten oder Cloud-basierten Containerdiensten bereitstellen und ausführen.

Um eine Partnerdatenquelle hinzuzufügen, muss [Docker Engine](#) 1.9.1 oder höher auf Ihrem lokalen Gerät installiert sein.

 Note

Version 20.10 ist die neueste Version, für die verifiziert wurde, dass sie mit der SiteWise Edge-Gateway-Software funktioniert.

Verify Docker ist installiert

Um zu überprüfen, ob Docker es installiert ist, führen Sie den folgenden Befehl von einem Terminal aus, das mit Ihrem SiteWise Edge-Gateway verbunden ist:

```
docker info
```

Wenn der Befehl ein `docker is not recognized` Ergebnis zurückgibt oder eine ältere Version von installiert Docker ist, [installieren Sie Docker Engine](#), bevor Sie fortfahren.

Richten Sie Docker ein

Der Systembenutzer, der eine Docker Container-Komponente ausführt, muss über Root- oder Administratorrechte verfügen, oder Sie müssen sie so konfigurieren Docker, dass er als Benutzer ausgeführt wird, der kein Root oder kein Administrator ist.

Auf Linux-Geräten müssen Sie der `docker` Gruppe einen `ggc_user` Benutzer hinzufügen, ohne dass Befehle aufgerufen werden sollen. Docker `sudo`

Führen Sie den folgenden Befehl aus `ggc_user`, um der `docker` Gruppe einen Nicht-Root-Benutzer, den Sie zum Ausführen von Docker Container-Komponenten verwenden, hinzuzufügen:

```
sudo usermod -aG docker ggc_user
```

Weitere Informationen finden Sie unter [Linux-Schritte nach der Installation für Docker Engine](#).

Fügen Sie eine Partnerdatenquelle in SiteWise Edge hinzu

Um eine Partnerdatenquelle mit Ihrem SiteWise Edge-Gateway zu verbinden, fügen Sie sie als Datenquelle hinzu. Wenn Sie sie als Datenquelle hinzufügen, AWS IoT SiteWise wird eine private AWS IoT Greengrass Komponente auf Ihrem SiteWise Edge-Gateway bereitgestellt.

Voraussetzungen

Um eine Partnerdatenquelle hinzuzufügen, müssen Sie wie folgt vorgehen:

- Erstellen Sie für EasyEdge und CloudRail ein Konto beim Partner und binden Sie dann die Konten.
- [Einrichten Docker auf Ihrem SiteWise Edge-Gateway](#)

Erstellen Sie ein SiteWise Edge-Gateway mit einer Partnerdatenquelle

Wenn Sie ein neues SiteWise Edge-Gateway erstellen möchten, führen Sie die Schritte unter [Erstellen Sie ein selbst gehostetes SiteWise Edge-Gateway](#). Nachdem Sie das SiteWise Edge-Gateway erstellt haben, folgen Sie [Fügen Sie einem vorhandenen SiteWise Edge-Gateway eine Partnerdatenquelle hinzu](#) den Schritten unter Hinzufügen einer Partnerdatenquelle.

Fügen Sie einem vorhandenen SiteWise Edge-Gateway eine Partnerdatenquelle hinzu

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie in der linken Navigationsleiste im Abschnitt Edge die Option Edge-Gateways aus.
3. Wählen Sie das SiteWise Edge-Gateway aus, mit dem Sie die Partnerdatenquelle verbinden möchten.
4. Wählen Sie unter Datenquellen die Option Datenquelle hinzufügen aus.
5. Wählen Sie auf dem Bildschirm Datenquelle hinzufügen einen Quelltyp aus, um den Partner auszuwählen, der Ihr SiteWise Edge-Gateway verbindet. Jede Datenquelle hat ihre eigenen Konfigurationsoptionen. Es gibt zwei Kategorien von Datenquellen: AWS Quellen und Partnerquellen.

Mithilfe einer Partnerdatenquelle können Sie eine Quelle pro Gateway auswählen. Eine Liste der Integrationsoptionen für Datenquellenpartner finden Sie unter [SiteWise Datenquellenoptionen für Edge-Gateway-Partner](#). Beachten Sie, dass Sie bis zu 100 OPC UA-Datenquellen (AWSQuellen) hinzufügen können. Informationen zu den ersten Schritten mit OPC UA-Datenquellen finden Sie unter [OPC UA-Datenquellen fürAWS IoT SiteWiseEdge-Gateways](#)

6. Geben Sie einen Namen für die Quelle ein.

7. Wählen Sie unten die Registerkarte Ihrer Datenquelle aus und folgen Sie dem Konfigurationsverfahren.

CloudRail

Ein Großteil der CloudRail Konfiguration wird im CloudRail Portal vorgenommen, nachdem die Datenquelle für Ihr SiteWise Edge-Gateway gespeichert wurde. Eine Autorisierung der Verbindung ist jedoch erforderlich.

Note

Die CloudRail Verbindung ist nur unter Linux verfügbar.

1. [Erstellen Sie ein CloudRail Konto](#), um mit der Verbindung zu beginnenAWS IoT SiteWise.
2. Stellen Sie sicher, dass Docker auf Ihrem Gateway installiert ist. Weitere Informationen finden Sie unter [Einrichten Docker auf Ihrem SiteWise Edge-Gateway](#).
3. Lesen Sie die Vereinbarung zur Autorisierung von Zugriff und Bereitstellung und wählen Sie dann Autorisieren aus. Wenn Sie das Kästchen aktivieren, AWS erhält der Partner Zugriff auf Ihre Datenquelle und ermöglicht die Bereitstellung AWS auf der Komponente des Partners.

Note

Das Messpräfix — optional — wird in Ihrem CloudRail Portal festgelegt.

Note

Die Partnersoftware wird vom AWS Partner entwickelt, gewartet und unterstützt. AWS ist nicht verantwortlich für die Schnittstelle, Konfiguration oder Software.

Weitere Informationen finden Sie unter [CloudRail](#).

EasyEdge

Ein Großteil der EasyEdge Konfiguration wird im EasyEdge Portal vorgenommen, nachdem die Datenquelle für Ihr SiteWise Edge-Gateway gespeichert wurde. Eine Autorisierung der Verbindung ist jedoch erforderlich.

Note

Die EasyEdge Verbindung ist nur unter Linux verfügbar.

1. [Erstellen Sie ein EasyEdge Konto](#), um mit der Verbindung zu beginnenAWS IoT SiteWise.
2. Stellen Sie sicher, dass Docker auf Ihrem Gateway installiert ist. Weitere Informationen finden Sie unter [Einrichten Docker auf Ihrem SiteWise Edge-Gateway](#).
3. Lesen Sie die Vereinbarung zur Autorisierung von Zugriff und Bereitstellung und wählen Sie dann Autorisieren aus. Wenn Sie das Kästchen aktivieren, AWS erhält der Partner Zugriff auf Ihre Datenquelle und ermöglicht die Bereitstellung AWS auf der Komponente des Partners.

Note

Das Messpräfix — optional — wird in Ihrem EasyEdge Portal festgelegt.

Note

Die Partnersoftware wird vom AWS Partner entwickelt, gewartet und unterstützt. AWS ist nicht verantwortlich für die Schnittstelle, Konfiguration oder Software.

Weitere Informationen finden Sie unter [EasyEdge](#).

Litmus Edge

Sie können die Litmus Konfiguration auf zwei Arten aktivieren. Litmus EdgeDirekte Aktivierung AWS IoT SiteWise mithilfe der Informationen aus dem Litmus Edge Manager

Portal. Oder Sie können die Aktivierung Litmus Edge manuell AWS IoT SiteWise durchführenLitmus Edge Manager.

 Note

Die Litmus Edge Verbindung ist nur unter Linux verfügbar.

Zur Aktivierung mit einem Litmus Edge Aktivierungscode aufAWS IoT SiteWise

Verwenden Sie dieses Verfahren, wenn Sie eine Litmus Edge Datenquelle mit einem Litmus Edge Aktivierungscode auf dem hinzufügenAWS-IoT-SiteWise-Konsole.

1. Wählen Sie Jetzt mit einem Code aktivieren aus. Zusätzliche Konfigurationsoptionen werden angezeigt.
2. Geben Sie den einLitmus Edge Manager, um eine Verbindung Litmus Edge zu Ihrem SiteWise Edge-Gateway herzustellen. Weitere Informationen finden Sie in der Litmus Edge Manager Dokumentation unter [Schritt 3a: Endpunkt für die Daten- und Geräteverwaltung festlegen](#).
3. Geben Sie den Litmus Edge Manager Aktivierungscode für die Aktivierung Litmus Edge ein AWS IoT SiteWise
4. Geben AWS IoT SiteWise Sie optional das Litmus Edge ManagerCA-Zertifikat an. Das Zertifikat Litmus Edge verhindert die Aktivierung auf einem nicht autorisierten GerätLitmus Edge Manager.
5. Stellen Sie sicher, dass Docker auf Ihrem Gateway installiert ist. Weitere Informationen finden Sie unter [Einrichten Docker auf Ihrem SiteWise Edge-Gateway](#).

 Note

AWS IoT SiteWisestellt die Partneranwendung als Docker-Container bereit. Die Anwendung wird mit NET_ADMIN Funktionen bereitgestellt, über die der Litmus Edge Docker-Container verwaltet werden kann. Litmus Edge Manager Litmus Edgebenötigt diesen privilegierten Zugriff, um auf Ihren Geräten ausgeführt zu werden. Weitere Informationen zu den Litmus Edge Docker-Anforderungen

findest du unter [Docker-Installation](#) im QuickStart Leitfaden in der Litmus Edge-Dokumentation.

6. Lesen Sie die Vereinbarung zur Autorisierung von Zugriff und Bereitstellung und wählen Sie dann Autorisieren aus. Wenn Sie das Kästchen aktivieren, AWS erhält der Partner Zugriff auf Ihre Datenquelle und ermöglicht die Bereitstellung AWS auf der Komponente des Partners.

Zur manuellen Aktivierung über Litmus Edge

1. Wählen Sie Später aktivieren aus. Litmus Edge
2. Stellen Sie sicher, dass Docker auf Ihrem Gateway installiert ist. Weitere Informationen finden Sie unter [Einrichten Docker auf Ihrem SiteWise Edge-Gateway](#).

Note

AWS IoT SiteWise stellt die Partneranwendung als Docker-Container bereit. Die Anwendung wird mit NET_ADMIN Funktionen bereitgestellt, über die der Litmus Edge Docker-Container verwaltet werden kann. Litmus Edge Manager Litmus Edge benötigt diesen privilegierten Zugriff, um auf Ihren Geräten ausgeführt zu werden. Weitere Informationen zu den Litmus Edge Docker-Anforderungen findest du unter [Docker-Installation](#) im QuickStart Leitfaden in der Litmus Edge-Dokumentation.

3. Lesen Sie die Vereinbarung zur Autorisierung von Zugriff und Bereitstellung und wählen Sie dann Autorisieren aus. Wenn Sie das Kästchen aktivieren, AWS erhält der Partner Zugriff auf Ihre Datenquelle und ermöglicht die Bereitstellung AWS auf der Komponente des Partners.
4. Wenn die Bereitstellung abgeschlossen ist, befolge die Anweisungen in [der Litmus Edge QuickStart Guide Dokumentation zu Access the Litmus Edge Web UI](#).

 Note

Die Partnersoftware wird vom AWS Partner entwickelt, gewartet und unterstützt. AWS ist nicht verantwortlich für die Schnittstelle, Konfiguration oder Software.

Weitere Informationen finden Sie unter [Litmus Edge](#).

8. Wählen Sie Save (Speichern) aus.

SiteWise Datenquellenoptionen für Edge-Gateway-Partner

AWS IoT SiteWise ermöglicht es Ihnen, Daten aus verschiedenen Partnerdatenquellen wie Industrieanlagen, Sensoren und anderen Systemen von Drittanbietern zu verbinden und aufzunehmen. Um eine Partnerdatenquelle zu verbinden, müssen Sie einige Schritte ausführen, darunter die Konfiguration der Datenquelle, an die Daten gesendet werden sollen AWS IoT SiteWise, die erforderlichen Berechtigungen und Authentifizierungen einrichten und die Daten Ihren Asset-Modellen zuordnen. Dieser Prozess stellt sicher, dass Ihre Partnerdaten nahtlos in Ihre AWS IoT SiteWise Umgebung integriert werden, sodass Sie sie zusammen mit Ihren anderen Datenquellen überwachen und analysieren können.

In diesem Abschnitt sind die verfügbaren Partner für die Integration von Datenquellen von Drittanbietern auf SiteWise Edge-Gateways aufgeführt. Verwenden Sie die folgenden Informationen, um eine Partnerdatenquelle zu konfigurieren.

 Note

Sie können für jeden Partner in jedem Gateway eine Datenquelle hinzufügen

CloudRail

Portal:

<https://devices.cloudrail.com/>

Voraussetzungen

Weitere Informationen zu den CloudRail Anforderungen finden Sie in den [häufig gestellten Fragen](#) auf der CloudRail Website.

CloudRailDokumentation:

[Edge-Computing: SiteWise Edge](#)

EasyEdge

Portal:

<https://studio.easyedge.io/>

Voraussetzungen

[EasyEdgeAnforderungen](#) — Informationen zu den EasyEdge Anforderungen, einschließlich Endpunkten und Ports, die für die Konfiguration der Firewall erforderlich sind. Hinweis: Sie benötigen ein EasyEdge Konto, um auf diese Dokumentation zugreifen zu können.

EasyEdgeDokumentation:

[EasyEdgefür AWS](#)

Litmus Edge

Zugang zuLitmus Edge Manager:

Um auf Litmus Edge zuzugreifen, richte ein [Litmus Edge Manager-Konto](#) ein.

Voraussetzungen

[Litmus EdgeAnforderungen](#) — Empfohlene Konfigurationen und Systemanforderungen für die Implementierung. Litmus Edge

LitmusDokumentation:

- [Integration zu AWS IoT SiteWise](#)
- [Litmus Edge-Dokumentation](#)

AWS IoT GreengrassKomponenten fürAWS IoT SiteWiseEdge

SiteWise Edge verwendet AWS IoT Greengrass Komponenten, um Industriedaten am Netzwerkrand zu sammeln, zu verarbeiten und zu übertragen. Diese Komponenten arbeiten zusammen, um eine lokale Datenverarbeitung und eine nahtlose Integration mit dem AWS IoT SiteWise Cloud-Dienst zu ermöglichen.

SiteWise IoT-Herausgeber

Die SiteWise IoT-Publisher-Komponente (`aws.iot.SiteWiseEdgePublisher`) ist verantwortlich für:

- Sichere Übertragung gesammelter Daten an den AWS IoT SiteWise Cloud-Dienst
- Verwaltung der Datenpufferung und der Wiederholungsversuche bei Verbindungsproblemen

Weitere Informationen zur Konfiguration des Herausgebers für SiteWise Edge finden Sie unter [Konfigurieren Sie dieAWS IoT SiteWisePublisher-Komponente](#). Weitere Informationen zur Publisher-Komponente finden Sie unter [IoT SiteWise Publisher](#) im AWS IoT Greengrass Version 2Developer Guide.

SiteWise IoT-Prozessor

Die SiteWise IoT-Prozessorkomponente (`aws.iot.SiteWiseEdgeProcessor`) erfüllt die folgenden Aufgaben:

- Ausführung von Datentransformationen und Berechnungen am Edge
- Lokale Implementierung von Definitionen und Berechnungen von Vermögenswerten
- Reduzierung des Datenvolumens durch Aggregation oder Filterung von Daten vor der Übertragung

Weitere Informationen zur Prozessorkomponente finden Sie unter [SiteWise IoT-Prozessor](#) im AWS IoT Greengrass Version 2Entwicklerhandbuch.

SiteWise IoT-OPC-UA-Kollektor

Die IoT-Komponente SiteWise OPC UA Collector (`aws.iot.SiteWiseEdgeCollectorOpcua`) wurde entwickelt, um:

- Connect zu OPC UA-Servern in industriellen Umgebungen her
- Erfassen Sie effizient Daten aus OPC UA-Datenquellen
- Transformieren Sie OPC UA-Daten in ein Format, das kompatibel ist mit AWS IoT SiteWise

Weitere Informationen zur OPC UA-Collector-Komponente finden Sie unter [IoT SiteWise OPC UA Collector](#) im AWS IoT Greengrass Version 2 Developer Guide.

IoT SiteWise OPC UA-Datenquellensimulator

Die SiteWise IoT-OPC UA-Datenquellensimulatorkomponente (`aws.iot.SiteWiseEdgeOpcuaDataSourceSimulator`) bietet die folgenden Funktionen:

- Startet einen lokalen OPC UA-Server, der Beispieldaten generiert
- Simuliert eine Datenquelle, die von der AWS IoT SiteWise OPC UA-Collector-Komponente auf einem Gateway gelesen werden kann AWS IoT SiteWise
- Ermöglicht die Untersuchung von AWS IoT SiteWise Funktionen anhand der generierten Beispieldaten

Diese Komponente ist besonders für Test- und Entwicklungszwecke nützlich, da Sie industrielle Datenquellen simulieren können, ohne dass physische Geräte erforderlich sind.

Weitere Informationen zur Komponente zur Datenquellensimulation finden Sie unter [IoT SiteWise OPC UA-Datenquellensimulator](#) im AWS IoT Greengrass Version 2 Developer Guide.

Diese AWS IoT Greengrass Komponenten ermöglichen die SiteWise Edge-Funktionalität. Der SiteWise IoT-Publisher stellt sicher, dass Daten zuverlässig an die Cloud gesendet werden, der SiteWise IoT-Prozessor kümmert sich um lokale Berechnungen und Datenoptimierungen, und der SiteWise IoT-OPC-UA-Collector erleichtert die Integration mit gängigen Industrieprotokollen.

Note

Um diese Komponenten verwenden zu können, müssen Sie sie auf Ihren Edge-Geräten installiert haben AWS IoT Greengrass V2 oder diese später installiert haben. Die richtige Konfiguration der einzelnen Komponenten ist wichtig für eine optimale Leistung von SiteWise Edge.

Filtern Sie Ressourcen auf einem SiteWise Edge-Gateway

Sie können die Edge-Filterung verwenden, um Ihre Ressourcen effizienter zu verwalten, indem Sie nur eine Teilmenge der Ressourcen zur Datenverarbeitung an ein bestimmtes SiteWise Edge-Gateway senden. Wenn Ihre Ressourcen in einer Baumstruktur oder einer übergeordneten und untergeordneten Struktur angeordnet sind, können Sie eine IAM-Richtlinie einrichten, die der IAM-

Rolle eines SiteWise Edge-Gateways zugeordnet ist, sodass nur der Stamm des Baums oder das übergeordnete Element und seine untergeordneten Elemente an ein bestimmtes Edge-Gateway gesendet werden können. SiteWise

Note

Wenn Sie vorhandene Elemente in einer Baumstruktur anordnen, gehen Sie nach dem Erstellen der Struktur zu jedem vorhandenen Asset, das Sie der Struktur hinzugefügt haben, und wählen Sie Bearbeiten und dann Speichern, um sicherzustellen, dass die neue Struktur AWS IoT SiteWise erkannt wird.

Richten Sie die Kantenfilterung ein

Richten Sie die Edge-Filterung auf Ihrem SiteWise Edge-Gateway ein, indem Sie der IAM-Rolle des SiteWise Edge-Gateways die folgende IAM-Richtlinie hinzufügen und sie durch die ID des Root-Assets *<root-asset-id>* ersetzen, das Sie an das SiteWise Edge-Gateway senden möchten.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssociatedAssets"
      ],
      "Resource": "arn:aws:iotsitewise:*:*:asset/*",
      "Condition": {
        "StringNotLike": {
          "iotsitewise:assetHierarchyPath": "/<root-asset-id>*"
        }
      }
    }
  ]
}
```

Wenn sich auf Ihrem SiteWise Edge-Gateway derzeit Assets befinden, die Sie entfernen möchten, melden Sie sich bei Ihrem SiteWise Edge-Gateway an und führen Sie den folgenden Befehl aus, um die Synchronisierung des SiteWise Edge-Gateways zu erzwingen, AWS IoT SiteWise indem Sie den Cache löschen.

```
sudo rm /greengrass/v2/work/aws.iot.SiteWiseEdgeProcessor/sync-app/  
sync_resource_bundles/edge.json
```

Konfigurieren Sie die Proxyunterstützung und verwalten Sie Trust Stores für AWS IoT SiteWise Edge

Konfigurieren und verwalten Sie in AWS IoT SiteWise Edge Trust Stores, um die Proxyunterstützung für Ihre Edge-Geräte einzurichten. Richten Sie zunächst die Proxykonfiguration ein und konfigurieren Sie dann Trust Stores. Sie können Trust Stores entweder während der Gateway-Installation oder manuell nach der Einrichtung Ihres Gateways konfigurieren.

- **Proxys** — Erleichtern Sie die Konnektivität zwischen Ihren Edge-Geräten und AWS -Diensten in verschiedenen Netzwerkumgebungen.
- **Vertrauensspeicher** — Sorgen Sie für sichere Verbindungen, indem Sie vertrauenswürdige Zertifikate verwalten. Richtige Konfigurationen helfen Ihnen dabei, Ihre Netzwerksicherheitsrichtlinien einzuhalten, die Kommunikation in eingeschränkten Netzwerkumgebungen zu ermöglichen und die Datenübertragung zwischen Edge-Geräten und Cloud-Diensten zu optimieren.

SiteWise Edge verwendet mehrere Trust Stores für verschiedene Komponententypen und gewährleistet so einen sicheren und effizienten Datenfluss von Ihren Edge-Geräten zur Cloud. Sie können Trust Stores und Proxys auf einem vorhandenen Gateway oder während des Installationsvorgangs bei der Erstellung eines neuen Gateways konfigurieren.

Anforderungen für Trust Store- und Proxykonfigurationen

Bevor Sie einen Trust Store konfigurieren oder SiteWise Edge mit Proxyeinstellungen installieren, stellen Sie sicher, dass Sie die Voraussetzungen erfüllen. Es gibt unterschiedliche Implementierungsanforderungen, die auf Ihrer Komponentennutzung und Ihren Funktionsanforderungen basieren.

Anforderungen an die Proxyunterstützung

- Die URL Ihres Proxyservers. Die URL sollte die Benutzerinformationen und die Portnummer für den Host enthalten. Beispiel, `scheme://[userinfo@]host[:port]`.
 - `scheme`— Muss HTTP oder HTTPS sein
 - (Optional) `userinfo` — Informationen zu Benutzername und Passwort
 - `host`— Der Hostname oder die IP-Adresse des Proxyservers
 - `port`— Die Portnummer
- Eine Liste von Adressen zur Umgehung des Proxys.
- (Optional) Die Proxy-CA-Zertifikatsdatei, wenn Sie einen HTTPS-Proxy mit einem selbstsignierten Zertifikat verwenden.

Anforderungen an den Trust Store

- Um die volle Funktionalität des Data Processing Packs mit HTTPS-Proxy zu erhalten, sollten Sie alle drei Trust Stores aktualisieren.
- Wenn Sie nur den IoT SiteWise OPC UA Collector und den IoT SiteWise Publisher verwenden, aktualisieren Sie die Zertifikate AWS IoT Greengrass Core und Java Trust Stores auf die neueste Version.

Bewährte Methoden für Trust Store- und Proxyserver-Edge-Konfigurationen

Für die laufende Wartung und zur Aufrechterhaltung des höchsten Sicherheitsniveaus in Ihrer Edge-Umgebung:

- Überprüfen und aktualisieren Sie die Proxyeinstellungen regelmäßig, um sie an Ihre Netzwerksicherheitsanforderungen anzupassen.
- Überwachen Sie die Gateway-Konnektivität und den Datenfluss, um eine ordnungsgemäße Proxykommunikation sicherzustellen
- Pflegen und aktualisieren Sie Trust Stores gemäß den Zertifikatsverwaltungsrichtlinien Ihres Unternehmens
- Sie können unsere empfohlenen Best Practices für sichere Kommunikation in Edge-Umgebungen implementieren und befolgen, wie z. B.:
- Dokumentieren Sie Ihre Proxy- und Trust Store-Konfigurationen, um die betriebliche Transparenz zu gewährleisten

- Halten Sie sich bei der Verwaltung von Anmeldeinformationen an die Sicherheitspraktiken Ihres Unternehmens

Diese Praktiken tragen dazu bei, einen sicheren und zuverlässigen Betrieb Ihrer SiteWise Edge-Gateways aufrechtzuerhalten und gleichzeitig Ihre umfassenderen Sicherheitsrichtlinien einzuhalten.

Konfigurieren Sie die Proxyeinstellungen während AWS IoT SiteWise Installation des Edge-Gateways

Sie können AWS IoT SiteWise Edge so konfigurieren, dass es während der Gateway-Installation mit einem Proxyserver funktioniert. Das Installationsskript unterstützt sowohl HTTP- als auch HTTPS-Proxys und kann automatisch Vertrauensspeicher für sichere Proxyverbindungen konfigurieren.

Wenn Sie das Installationsskript mit Proxyeinstellungen ausführen, führt es mehrere wichtige Aufgaben aus:

- Überprüft das Format und die Parameter der Proxy-URL, um sicherzustellen, dass sie korrekt angegeben sind.
- Lädt die erforderlichen Abhängigkeiten über den konfigurierten Proxy herunter und installiert sie.
- Wenn ein Proxy-CA-Zertifikat bereitgestellt wird, wird es an das AWS IoT Greengrass Root-CA-Zertifikat angehängt und in Java KeyStore importiert.
- Konfiguriert AWS IoT Greengrass (was SiteWise Edge verwendet), um den Proxy für alle ausgehenden Verbindungen zu verwenden.
- Schließt die SiteWise Edge-Installation mit den entsprechenden Proxy- und Trust Store-Konfigurationen ab.

Um die Proxyeinstellungen bei der Installation der Gateway-Software zu konfigurieren

1. Erstellen Sie ein SiteWise Edge-Gateway. Weitere Informationen erhalten Sie unter [Erstellen Sie ein selbst gehostetes SiteWise Edge-Gateway](#) und [Installieren Sie das AWS IoT SiteWise Edge-Gateway-Software auf Ihrem lokalen Gerät](#).
2. Führen Sie das Installationsskript mit den entsprechenden Proxyeinstellungen für Ihre Umgebung aus. Ersetzen Sie die Platzhalter durch Ihre spezifischen Proxyinformationen

Ersetzen Sie jedes der folgenden Elemente:

- `-p, --proxy-url` — Die URL des Proxyservers. Die URL muss entweder `http` oder `https` lauten.
- `-n, --no-proxy` — Eine durch Kommas getrennte Liste von Adressen zur Umgehung des Proxys.
- (Optional) `-c, --proxy-ca-cert` — Pfad zur Proxy-CA-Zertifikatsdatei.
- (Optional) `-j, --javastorepass` — Das KeyStore Java-Passwort. Das Standardpasswort lautet `changeit`.

Linux

Verwenden Sie für Linux-Systeme die folgende Befehlsstruktur:

```
sudo ./install.sh -p proxy-url -n no-proxy-addresses [-c proxy-ca-cert-path] [-j javastorepass]
```

Windows

Verwenden Sie für Microsoft Windows Systeme PowerShell, die diese Befehlsstruktur verwenden:

```
.\install.ps1 -ProxyUrl proxy-url -NoProxyAddresses no-proxy-addresses [-ProxyCaCertPath proxy-ca-cert-path] [-JavaStorePass javastorepass]
```

Problembehandlung bei der Installation mit einem Proxy

Weitere Informationen zur Lösung von Problemen mit dem Trust Store im Zusammenhang mit einem SiteWise Edge-Gateway finden Sie unter [Probleme bei der Installation mit einem Proxy](#).

Konfigurieren Sie Trust Stores für die HTTPS-Proxyunterstützung manuell in AWS IoT SiteWise Edge

Wenn Sie AWS IoT SiteWise Edge-Komponenten so konfigurieren, dass sie eine Verbindung über einen HTTPS-Proxy herstellen, fügen Sie das Zertifikat des Proxyservers den entsprechenden Trust Stores hinzu. SiteWise Edge verwendet mehrere Trust Stores, um die Kommunikation zu sichern. Es gibt drei Trust Stores, und Ihre Verwendung hängt vom SiteWise Edge-Komponententyp in Ihrer Gateway-Implementierung ab.

Trust Stores werden während des Installationsvorgangs automatisch aktualisiert, wenn die Proxyeinstellungen bereitgestellt werden.

- [Konfigurieren Sie eine AWS IoT Greengrass Trust Store für Kernkomponenten](#)— Das AWS IoT Greengrass Root-CA-Zertifikat ist in den Trust Stores enthalten, um die Authentizität der AWS Dienste zu überprüfen.

Dieser Trust Store hilft AWS IoT Greengrass Komponenten dabei, sicher über den Proxy mit AWS Diensten zu kommunizieren und gleichzeitig die Authentizität dieser Dienste zu überprüfen.

- [Konfigurieren Sie einen Vertrauensspeicher für Java-based Komponenten](#)— Java KeyStore (JKS) ist der wichtigste Vertrauensspeicher, der von Java-based Komponenten für SSL/TLS Verbindungen verwendet wird.

Java-Anwendungen verlassen sich auf den JKS, um sichere Verbindungen herzustellen. Wenn Sie beispielsweise den SiteWise IoT-Publisher oder den IoT SiteWise OPC UA-Collector verwenden Java-based, was dies sind, müssen Sie diesen Trust Store konfigurieren. Dadurch wird sichergestellt, dass diese Komponenten sicher über den HTTPS-Proxy kommunizieren können, wenn sie Daten an die Cloud senden oder Daten von OPC UA-Servern sammeln.

- [System-level Konfiguration des Komponenten-Trust-Stores](#)— Bei der Verwendung von HTTPS-Proxy müssen deren Zertifikate den entsprechenden Trust Stores hinzugefügt werden, um sichere Verbindungen zu ermöglichen.

Bei der Verwendung von HTTPS-Proxy müssen ihre Zertifikate den entsprechenden Trust Stores hinzugefügt werden, um sichere Verbindungen zu ermöglichen. Dies ist notwendig, da Komponenten auf Systemebene, die oft in Sprachen wie Rust oder Go geschrieben sind, eher auf den Trust Store des Systems als auf Javas JKS angewiesen sind. Wenn Sie beispielsweise Systemdienstprogramme verwenden, die über den Proxy kommunizieren müssen (z. B. für Softwareupdates oder Zeitsynchronisierung), müssen Sie den Trust Store auf Systemebene konfigurieren. Dadurch wird sichergestellt, dass diese Komponenten und Dienstprogramme sichere Verbindungen über den Proxy herstellen können.

Konfigurieren Sie eine AWS IoT Greengrass Trust Store für Kernkomponenten

Für AWS IoT Greengrass Kernfunktionen, die die Root-CA von Amazon verwenden:

1. Suchen Sie die Zertifikatsdatei unter `/greengrass/v2/AmazonRootCA1.pem`
2. Hängen Sie das HTTPS-Proxy Stammzertifikat (selbstsigniert) an diese Datei an.

```

-----BEGIN CERTIFICATE-----
MIIEFTCCA v2gAwIQWgIVAMHSAzWG/5YVRYtRQ0xXUTEpHuEmApzGCSqGSIb3DQEK
\nCwUAhuL9MQswCQwJVUzEPMAVUzEYMBYGA1UECgwP1hem9uLmNvbSBjbmuMRww
... content of proxy CA certificate ...
+vHIR1t0e5JAm5\noTIZGoFbK82A0/n07f/t5PSIDAim9V3Gc3pSXxCCAQoFYnui
GaPUlGk1gCE84a0X\n7Rp/1ND/PuMZ/s8Yj1kY2NmYmNjMCAXDTE5MTEyN2cM216
gJMIADggEPADf2/m45hzEXAMPLE=
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDQTCCAimgF6AwIBAgITBmyfz/5mjAo54vB4ikPmljZKyjANJmApzyMZFo6qBg
ADA5MQswCQYDVQQGEwJVUzEPMA0tMVT8QtPHRh8jrdkGA1UEChMGMGV3ZQQDExBBKW
... content of root CA certificate ...
o/ufQJQWUCyziar1hem9uMRkwFwYVPSHCb2XV4cdFyQzR1KldZwgJcIQ6XUDgHaa
5MsI+yMRQ+hDaXJioblDxgjUka642M4UwtBV8oK2xJNDd2ZhwLnoQdeXeGADKkpy
rqXRfKoQnoZsG4q5WTP46EXAMPLE
-----END CERTIFICATE-----

```

Konfigurieren Sie den HTTPS-Proxy auf einem etablierten Gateway

Sie können einem etablierten Gateway Proxyunterstützung hinzufügen, indem Sie eine Verbindung zu Port 443 statt zu Port 8883 herstellen. Weitere Informationen zur Verwendung eines Proxyserver finden Sie unter [Connect über Port 443 oder über einen Netzwerk-Proxy](#) im AWS IoT Greengrass Version 2Entwicklerhandbuch. Wenn Sie ein neues Gateway erstellen, können Sie die Proxykonfiguration während der Gateway-Installation festlegen. Weitere Informationen finden Sie unter [Konfigurieren Sie die Proxyeinstellungen währendAWS IoT SiteWiseInstallation des Edge-Gateways](#).

Wenn Sie einen HTTPS-Proxy AWS IoT Greengrass auf SiteWise Edge verwenden, wählt die Software anhand der angegebenen URL automatisch zwischen HTTP und HTTPS für Proxyverbindungen.

Important

Aktualisieren Sie alle erforderlichen Vertrauensspeicher, bevor Sie versuchen, eine Verbindung über einen HTTPS-Proxy herzustellen.

Konfigurieren Sie einen Vertrauensspeicher für Java-based Komponenten

Für SiteWise IoT-Publisher, SiteWise IoT-OPC-UA-Collector und Java-Dienste im Datenverarbeitungspaket ist der standardmäßige Java-Trust-Store-Speicherort `$JAVA_HOME/jre/lib/security/cacerts`

Um ein Zertifikat hinzuzufügen

1. Erstellen Sie eine Datei zum Speichern des Zertifikats des Proxyservers, z. `proxy.crt` B.

Note

Erstellen Sie die Datei im Voraus mit dem Zertifikat des Proxyservers.

2. Fügen Sie die Datei mit dem folgenden Befehl zum Trust Store von Java hinzu:

```
sudo keytool -import -alias proxyCert -keystore /usr/lib/jvm/java-11-openjdk-amd64/lib/security/cacerts -file proxy.crt
```

3. Wenn Sie dazu aufgefordert werden, verwenden Sie das Standardkennwort: `changeit`

System-level Konfiguration des Komponenten-Trust-Stores

Für Komponenten, die in Rust, Go und anderen Sprachen geschrieben wurden und den System Trust Store verwenden:

Linux

Linux-Systeme: Fügen Sie Zertifikate hinzu `/etc/ssl/certs/ca-certificates.crt`

Windows

Microsoft Windows-Systeme: Um den Trust Store zu konfigurieren, folgen Sie dem [Certificate Store-Verfahren](#) in der Microsoft Ignite-Dokumentation.

Windows bietet mehrere Zertifikatsspeicher, einschließlich separater Speicher für Benutzer- und Computerbereiche, jeweils mit mehreren Unterspeichern. Für die meisten SiteWise Edge-Setups empfehlen wir, dem Store Zertifikate hinzuzufügen. `COMPUTER | Trusted Root Certification Authorities` Abhängig von Ihrer spezifischen Konfiguration und Ihren Sicherheitsanforderungen müssen Sie jedoch möglicherweise einen anderen Store verwenden.

Behebung von Problemen mit dem Trust Store

Weitere Informationen zur Lösung von Trust Store-Problemen im Zusammenhang mit einem SiteWise Edge-Gateway finden Sie unter [Probleme mit dem Trust Store](#).

Verwenden Sie AWS IoT SiteWise APIs am Edge

AWS IoT SiteWise stellt eine Teilmenge seiner APIs zusammen mit Edge-spezifischen APIs bereit und ermöglicht so eine nahtlose Interaktion mit den am Edge bereitgestellten Asset-Modellen und den zugehörigen Ressourcen. Diese Asset-Modelle müssen so konfiguriert werden, dass sie am Edge ausgeführt werden können. Weitere Informationen finden Sie unter [Konfigurieren Sie ein Asset-Modell für die Datenverarbeitung auf Edge SiteWise](#) für detaillierte Anweisungen zu diesem Einrichtungsprozess.

Nachdem Sie diese APIs konfiguriert haben, können Sie umfassende Daten zu Ihren Asset-Modellen und einzelnen Assets abrufen. Durch das Abrufen von Asset-Modell-, Asset-, Dashboard-, Portal- und Projektinformationen können Sie bereitgestellte Portale und Dashboards überwachen und auf Asset-Daten zugreifen, die auf Edge-Ebene gesammelt wurden. Dies bietet einen zentralen Host in Ihrem Netzwerk für Interaktionen, AWS IoT SiteWise ohne dass ein Web-API-Aufruf erforderlich ist.

Themen

- [Alles verfügbar AWS IoT SiteWise APIs für Edge-Geräte](#)
- [Edge-only APIs zur Verwendung mit AWS IoT SiteWise Edge-Geräte](#)
- [Aktivieren Sie CORS auf AWS IoT SiteWise Edge-APIs](#)
- [Konfigurieren Sie Sitzungs-Timeouts für AWS IoT SiteWise Edge](#)
- [Tutorial: Auflisten von Asset-Modellen auf einem AWS IoT SiteWise Edge-Gateway](#)

Alles verfügbar AWS IoT SiteWise APIs für Edge-Geräte

AWS IoT SiteWise bietet eine Vielzahl von APIs zur Verwendung auf Edge-Geräten, sodass Sie Aufgaben lokal auf dem Gerät ausführen können. Einige der verfügbaren Edge-APIs umfassen das Abrufen von Asset-Modellen, das Erstellen und Aktualisieren von Asset-Eigenschaften und das Senden von Datenströmen in die Cloud. Durch die Nutzung dieser APIs können Sie Lösungen entwickeln, die in Umgebungen mit intermittierender oder eingeschränkter Netzwerkkonnektivität eingesetzt werden können.

Available (Verfügbar)AWS IoT SiteWiseAPIs

Die folgenden AWS IoT SiteWise APIs sind auf Edge-Geräten verfügbar:

- [ListAssetModels](#)
- [DescribeAssetModel](#)
- [ListAssets](#)
- [DescribeAsset](#)
- [DescribeAssetProperty](#)
- [ListAssociatedAssets](#)
- [GetAssetPropertyAggregates](#)
- [GetAssetPropertyValue](#)
- [GetAssetPropertyValueHistory](#)
- [ListDashboards](#)
- [ListPortals](#)
- [ListProjectAssets](#)
- [ListProjects](#)
- [DescribeDashboard](#)
- [DescribePortal](#)
- [DescribeProject](#)

Verfügbare APIs nur für Edge-Anwendungen

Die folgenden APIs werden lokal auf Geräten am Edge verwendet:

- [Authentifizieren](#)— Verwenden Sie diese API, um die temporären SigV4-Anmeldeinformationen abzurufen, die Sie für API-Aufrufe verwenden werden.

Edge-only APIs zur Verwendung mitAWS IoT SiteWiseEdge-Geräte

Zusätzlich zu den AWS IoT SiteWise APIs, die am Edge verfügbar sind, gibt es Edge-spezifische APIs. Diese Edge-spezifischen APIs werden im Folgenden beschrieben.

Authentifizieren

Ruft die Anmeldeinformationen vom SiteWise Edge-Gateway ab. Sie müssen lokale Benutzer hinzufügen oder über LDAP oder einen Linux-Benutzerpool eine Verbindung zu Ihrem System herstellen. Weitere Informationen zum Hinzufügen von Benutzern finden Sie unter [LDAP](#) - oder [Linux-Benutzerpool](#).

Erforderliche Syntax

```
POST /authenticate HTTP/1.1
Content-type: application/json
{
  "username": "string",
  "password": "string",
  "authMechanism": "string"
}
```

URI-Anforderungsparameter

Die Anforderung verwendet keine URI-Parameter.

Anforderungstext

Die Anforderung akzeptiert die folgenden Daten im JSON-Format.

username

Der Benutzername, der zur Validierung des Anforderungsaufrufs verwendet wurde.

Typ: Zeichenfolge

Erforderlich: Ja

password

Das Passwort des Benutzers, der Anmeldeinformationen anfordert.

Typ: Zeichenfolge

Erforderlich: Ja

authMechanism

Die Authentifizierungsmethode zur Validierung dieses Benutzers auf dem Host.

Typ: Zeichenfolge

Zulässige Werte: ldap, linux, winnt

Erforderlich: Ja

Antwortsyntax

```
HTTP/1.1 200
Content-type: application/json
{
  "accessKeyId": "string",
  "secretAccessKey": "string",
  "sessionToken": "string",
  "region": "edge"
}
```

Antwortelemente

Wenn die Aktion erfolgreich ist, sendet der Service eine HTTP 200-Antwort zurück.

Die folgenden Daten werden im JSON-Format zurückgegeben.

Zugriff KeyId

Die Zugriffsschlüssel-ID, die die temporären Sicherheitsanmeldedaten identifiziert.

Längenbeschränkungen: Mindestlänge von 16. Maximale Länge beträgt 128 Zeichen.

Pattern: `[\w]*`

geheim AccessKey

Der geheime Zugriffsschlüssel, der zum Signieren von Anfragen verwendet werden kann.

Typ: Zeichenfolge

sessionToken

Das Token, das Benutzer an die Service-API übergeben müssen, um die temporären Anmeldeinformationen verwenden zu können.

Typ: Zeichenfolge

Region

Die Region, auf die Sie für API-Aufrufe abzielen.

Typ: CONSTANT - edge

Fehler

IllegalArgumentException

Die Anfrage wurde abgelehnt, weil das bereitgestellte Hauptdokument falsch formatiert war. Die Fehlermeldung beschreibt den spezifischen Fehler.

HTTP-Statuscode: 400

AccessDeniedException

Der Benutzer verfügt nicht über gültige Anmeldeinformationen, die auf dem aktuellen Identity Provider basieren. Die Fehlermeldung beschreibt den Authentifizierungsmechanismus.

HTTP-Statuscode: 403

TooManyRequestsException

Die Anfrage hat das Limit an Authentifizierungsversuchen erreicht. Die Fehlermeldung gibt an, wie lange gewartet werden soll, bis neue Authentifizierungsversuche unternommen werden.

HTTP-Statuscode: 429

Aktivieren Sie CORS auf AWS IoT SiteWise Edge-APIs

Durch die Aktivierung von CORS (Cross-Origin Resource Sharing) auf AWS IoT SiteWise Edge-APIs können Webanwendungen domänenübergreifend direkt mit den APIs kommunizieren. Dies ermöglicht eine nahtlose Integration, einen Datenaustausch in Echtzeit und einen domänenübergreifenden Datenzugriff ohne Zwischenserver oder Problemumgehungen. Die CORS-Einstellungen können so konfiguriert werden, dass zulässige Ursprünge angegeben werden, wodurch ein kontrollierter Zugriff zwischen verschiedenen Quellen gewährleistet wird.

Note

CORS ist für Version 3.3.1 und höher der Komponente verfügbar. Diese Funktion ist für Version 3.3.1 und höher der Komponente verfügbar. `aws.iot.SiteWiseEdgeProcessor`
Weitere Informationen finden Sie im [AWS IoT SiteWiseEntwicklerhandbuch unter Prozessor](#).
AWS IoT Greengrass Version 2

Um CORS auf SiteWise Edge-APIs zu aktivieren

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie das SiteWise Edge-Gateway aus, für das Sie CORS aktivieren möchten. Sie können CORS für den AWS IoT Greengrass V2 Bereitstellungstyp aktivieren.
4. Wählen Sie im Abschnitt Gateway-Konfiguration das zugehörige Greengrass-Core-Gerät aus.
5. Wählen Sie auf der Registerkarte Bereitstellungen unter Greengrass-Geräte den entsprechenden Bereitstellungslink aus.
6. Wählen Sie unter Aktionen die Option Überarbeiten und dann Bereitstellung überarbeiten aus.

Important

Durch das Erstellen einer überarbeiteten CORS-fähigen Konfiguration wird die aktuelle Konfiguration des Geräts ersetzt.

7. Geben Sie in Schritt 1, Ziel angeben, optional einen Namen an, um die Bereitstellung zu identifizieren.
8. In Schritt 2, Komponenten auswählen — optional, können Sie alle aktuellen Auswahlen unverändert lassen und Weiter wählen.
9. Wählen Sie in Schritt 3, Komponenten konfigurieren — optional, aws.iot aus. SiteWiseEdgeProcessor, und wählen Sie Komponente konfigurieren aus.
10. Geben Sie im Abschnitt Konfigurationsupdate unter Konfiguration zum Zusammenführen den folgenden JSON-Code ein:

```
{
  "AWS_SITEWISE_EDGE_ACCESS_CONTROL_ALLOW_ORIGIN": "*"
}
```

Note

Die Verwendung * als Wert für `AWS_SITEWISE_EDGE_ACCESS_CONTROL_ALLOW_ORIGIN` lässt alle Ursprünge zu. Für Produktionsumgebungen wird aus Sicherheitsgründen empfohlen, genaue Quell-URLs anzugeben.

11. Wählen Sie Bestätigen aus.
12. Wählen Sie Weiter, um mit den verbleibenden Schritten fortzufahren, bis Sie zu Schritt 5, Überprüfen, gelangen.
13. Überprüfen Sie Ihre Konfigurationsänderungen und wählen Sie dann Deploy aus, um die Änderungen auf Ihr SiteWise Edge-Gateway anzuwenden.

 Note

Alternativ können Sie CORS aktivieren, indem Sie die Umgebungsvariable * auf Ihrem AWS IoT SiteWise Gateway global `AWS_SITWISE_EDGE_ACCESS_CONTROL_ALLOW_ORIGIN` auf setzen.

 Note

Bei authentifizierten Proxys `userinfo` muss es in das `url` Feld der Proxykonfiguration aufgenommen werden `username` und `password` nicht als separate UN-Felder.

Nach Abschluss der Bereitstellung wird CORS auf Ihrer SiteWise Edge-API aktiviert, sodass bestimmte Quellen ursprungsübergreifende Anfragen an die API stellen können.

Konfigurieren Sie Sitzungs-Timeouts fürAWS IoT SiteWiseEdge

SiteWise Edge ermöglicht es Ihnen, Sitzungs-Timeouts für die SiteWise Edge-API zu konfigurieren. Diese Funktion erhöht die Sicherheit, indem inaktive Sitzungen nach einem bestimmten Zeitraum automatisch beendet werden. Dieser Abschnitt führt Sie durch den Prozess der Konfiguration des Sitzungs-Timeouts mithilfe von. AWS-IoT-SiteWise-Konsole

 Note

Die Konfiguration des Sitzungs-Timeouts ist für Version 3.4.0 und höher der Komponente verfügbar. `aws.iot.SiteWiseEdgeProcessor` Weitere Informationen finden Sie im AWS IoT Greengrass Version 2Entwicklerhandbuch unter [AWS IoT SiteWiseProzessor](#).

So konfigurieren Sie ein Sitzungs-Timeout für ein SiteWise Edge-Gateway

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie das SiteWise Edge-Gateway aus, für das Sie das Sitzungstimeout konfigurieren möchten.

Note

Sie können das Sitzungs-Timeout für den AWS IoT Greengrass V2 Bereitstellungstyp konfigurieren.

4. Wählen Sie im Abschnitt Gateway-Konfiguration das zugehörige Greengrass-Core-Gerät aus.
5. Wählen Sie auf der Registerkarte Bereitstellungen unter Greengrass-Geräte den entsprechenden Bereitstellungslink aus.
6. Wählen Sie unter Aktionen die Option Überarbeiten aus. Lesen Sie die Warnung und wählen Sie dann Bereitstellung überarbeiten aus.

Important

Durch das Erstellen einer überarbeiteten Konfiguration für das Sitzungs-Timeout wird die aktuelle Konfiguration des Geräts ersetzt.

7. Geben Sie in Schritt 1, Ziel angeben, optional einen Namen an, um die überarbeitete Bereitstellung zu identifizieren, und wählen Sie dann Weiter aus.
8. In Schritt 2, Komponenten auswählen — optional, können Sie alle aktuellen Auswahlen unverändert lassen und Weiter wählen.
9. Wählen Sie in Schritt 3, Komponenten konfigurieren — optional, aws.iot aus. SiteWiseEdgeProcessor, und wählen Sie Komponente konfigurieren aus.
10. Geben Sie im Abschnitt Konfigurationsupdate unter Konfiguration zum Zusammenführen den folgenden JSON-Code ein:

```
{
  "AWS_SITEWISE_EDGE_SESSION_TIMEOUT_MINUTES": "240"
}
```

11. Stellen Sie den Wert für `AWS_SITEWISE_EDGE_SESSION_TIMEOUT_MINUTES` in Minuten ein. Die Werte für das Sitzungs-Timeout können zwischen 1 Minute und 10080 Minuten (7 Tage) liegen. Der Standardwert ist 240 Minuten (4 Stunden).
12. Wählen Sie Bestätigen aus.
13. Wählen Sie Weiter, um mit den verbleibenden Schritten fortzufahren, bis Sie zu Schritt 5, Überprüfen, gelangen.
14. Überprüfen Sie Ihre Konfigurationsänderungen und wählen Sie dann Deploy aus, um die Änderungen auf Ihr SiteWise Edge-Gateway anzuwenden.

 Note

Alternativ können Sie das Sitzungstimeout konfigurieren, indem Sie die globale Umgebungsvariable `AWS_SITEWISE_EDGE_SESSION_TIMEOUT_MINUTES` auf Ihrem SiteWise Edge-Gateway auf den gewünschten Wert (in Minuten) einstellen.

Nach Abschluss der Bereitstellung wird die neue Konfiguration des Sitzungstimeouts auf Ihre SiteWise Edge-API angewendet.

Tutorial: Auflisten von Asset-Modellen auf einem AWS IoT SiteWise Edge-Gateway

Sie können eine Teilmenge der verfügbaren AWS IoT SiteWise APIs zusammen mit Edge-spezifischen APIs verwenden, um mit Asset-Modellen und ihren Assets am Edge zu interagieren. In diesem Tutorial erfahren Sie, wie Sie temporäre Anmeldeinformationen für ein AWS IoT SiteWise Edge-Gateway abrufen und eine Liste der Asset-Modelle auf dem SiteWise Edge-Gateway abrufen.

Voraussetzungen

In den Schritten dieses Tutorials können Sie eine Vielzahl von Tools verwenden. Um diese Tools verwenden zu können, stellen Sie sicher, dass Sie die entsprechenden Voraussetzungen installiert haben.

Zum Durcharbeiten dieses Tutorials ist Folgendes erforderlich:

- Ein bereitgestelltes und laufendes [AWS IoT SiteWise Anforderungen an das selbst gehostete Edge-Gateway](#)
- Zugriff auf Ihr SiteWise Edge-Gateway im selben Netzwerk über Port 443.

- [OpenSSL installiert](#)
- (AWS OpsHub für AWS IoT SiteWise) Die [AWS OpsHub für die Anwendung AWS IoT SiteWise](#)
- (curl) [curl](#) ist installiert
- (Python) [urllib3 installiert](#)
- (Python) [Python3 installiert](#)
- (Python) [Boto3 installiert](#)
- (Python) [BotoCore](#) installiert

Schritt 1: Besorgen Sie sich ein signiertes Zertifikat für den SiteWise Edge-Gateway-Service

Um eine TLS-Verbindung zu den am SiteWise Edge-Gateway verfügbaren APIs herzustellen, benötigen Sie ein vertrauenswürdiges Zertifikat. Sie können dieses Zertifikat mit einem OpenSSL oder AWS OpsHub für AWS IoT SiteWise generieren.

OpenSSL

Note

Um diesen Befehl ausführen zu können, muss [OpenSSL](#) installiert sein.

Öffnen Sie ein Terminal und führen Sie den folgenden Befehl aus, um ein signiertes Zertifikat vom SiteWise Edge-Gateway abzurufen. `<sitewise_gateway_ip>` Ersetzen Sie es durch die IP des SiteWise Edge-Gateways.

```
openssl s_client -connect <sitewise_gateway_ip>:443 </dev/null 2>/dev/null | openssl  
x509 -outform PEM > GatewayCert.pem
```

AWS OpsHub für AWS IoT SiteWise

Sie können AWS OpsHub für verwenden AWS IoT SiteWise. Weitere Informationen finden Sie unter [SiteWise Edge-Gateways verwalten](#).

In diesem Tutorial wird der absolute Pfad zum heruntergeladenen SiteWise Edge-Gateway-Zertifikat verwendet. Führen Sie den folgenden Befehl aus, um den vollständigen Pfad Ihres Zertifikats zu exportieren, und `<absolute_path_to_certificate>` ersetzen Sie ihn durch den Pfad zum Zertifikat:

```
export PATH_TO_CERTIFICATE='<absolute_path_to_certificate>'
```

Schritt 2: Holen Sie sich Ihren SiteWise Edge-Gateway-Hostnamen

 Note

Um diesen Befehl ausführen zu können, muss [OpenSSL](#) installiert sein.

Um das Tutorial abzuschließen, benötigen Sie den Hostnamen Ihres SiteWise Edge-Gateways. Um den Hostnamen Ihres SiteWise Edge-Gateways abzurufen, führen Sie den folgenden Befehl aus und `<sitewise_gateway_ip>` ersetzen Sie ihn durch die IP des SiteWise Edge-Gateways:

```
openssl s_client -connect <sitewise_gateway_ip>:443 </dev/null 2>/dev/null | grep -Po  
'CN = \K.*' | head -1
```

Führen Sie den folgenden Befehl aus, um den Hostnamen für die spätere Verwendung zu exportieren und ihn durch den Hostnamen Ihres SiteWise Edge-Gateways zu `<your_edge_gateway_hostname>` ersetzen:

```
export GATEWAY_HOSTNAME='<your_edge_gateway_hostname>'
```

Schritt 3: Besorgen Sie sich temporäre Anmeldeinformationen für Ihr Edge-Gateway SiteWise

Nachdem Sie das signierte Zertifikat und den Hostnamen Ihres SiteWise Edge-Gateways haben, benötigen Sie temporäre Anmeldeinformationen, damit Sie APIs auf dem Gateway ausführen können. Sie können diese Anmeldeinformationen mithilfe von APIs AWS OpsHub für AWS IoT SiteWise oder direkt vom SiteWise Edge-Gateway abrufen.

 Important

Die Anmeldeinformationen laufen alle 4 Stunden ab. Sie sollten sich die Anmeldeinformationen daher kurz vor der Verwendung der APIs auf Ihrem SiteWise Edge-Gateway besorgen. Speichern Sie Anmeldeinformationen nicht länger als 4 Stunden im Cache.

Holen Sie sich temporäre Anmeldeinformationen mit AWS OpsHub für AWS IoT SiteWise

 Note

Sie müssen die [AWS OpsHub AWS IoT SiteWise FOR-Anwendung](#) installiert haben.

Gehen Sie wie folgt vor, um AWS OpsHub für die AWS IoT SiteWise Anwendung Ihre temporären Anmeldeinformationen abzurufen:

1. Loggen Sie sich in die Anwendung ein.
2. Wählen Sie Einstellungen aus.
3. Wählen Sie für Authentifizierung die Option Anmeldeinformationen kopieren aus.
4. Erweitern Sie die Option, die zu Ihrer Umgebung passt, und wählen Sie Kopieren aus.
5. Speichern Sie die Anmeldeinformationen für die spätere Verwendung.

Rufen Sie mithilfe der SiteWise Edge-Gateway-API temporäre Anmeldeinformationen ab

Um die SiteWise Edge-Gateway-API zum Abrufen der temporären Anmeldeinformationen zu verwenden, können Sie ein Python-Skript oder Curl verwenden. Zunächst benötigen Sie einen Benutzernamen und ein Passwort für Ihr SiteWise Edge-Gateway. Die SiteWise Edge-Gateways verwenden die SigV4-Authentifizierung und -Autorisierung. Weitere Informationen zum Hinzufügen von Benutzern finden Sie unter [LDAP](#) - oder [Linux-Benutzerpool](#). Diese Anmeldeinformationen werden in den folgenden Schritten verwendet, um die lokalen Anmeldeinformationen auf Ihrem SiteWise Edge-Gateway abzurufen, die für die Verwendung der AWS IoT SiteWise APIs erforderlich sind.

Python

 Note

Sie müssen [urllib3 und Python3](#) installiert haben.

Um die Anmeldeinformationen mit Python abzurufen

1. Erstellen Sie eine Datei namens `get_credentials.py` und kopieren Sie dann den folgenden Code hinein.

```
'''
```

The following demonstrates how to get the credentials from the SiteWise Edge gateway. You will need to add local users or connect your system to LDAP/AD <https://docs.aws.amazon.com/iot-sitewise/latest/userguide/manage-gateways-ggv2.html#create-user-pool>

Example usage:

```
python3 get_credentials.py -e https://<gateway_hostname> -c
<path_to_certificate> -u '<gateway_username>' -p '<gateway_password>' -m
'<method>'
```

```
'''
```

```
import urllib3
import json
import urllib.parse
import sys
import os
import getopt
```

```
"""
```

This function retrieves theAWS IoT SiteWiseEdge gateway credentials.

```
"""
```

```
def get_credentials(endpoint, certificatePath, user, password, method):
    http = urllib3.PoolManager(cert_reqs='CERT_REQUIRED', ca_certs=
certificatePath)
    encoded_body = json.dumps({
        "username": user,
        "password": password,
        "authMechanism": method,
    })

    url = urllib.parse.urljoin(endpoint, "/authenticate")

    response = http.request('POST', url,
        headers={'Content-Type': 'application/json'},
        body=encoded_body)

    if response.status != 200:
        raise Exception(f'Failed to authenticate! Response status
{response.status}')

    auth_data = json.loads(response.data.decode('utf-8'))

    accessKeyId = auth_data["accessKeyId"]
```

```

    secretAccessKey = auth_data["secretAccessKey"]
    sessionToken = auth_data["sessionToken"]
    region = "edge"

    return accessKeyId, secretAccessKey, sessionToken, region

def print_help():
    print('Usage:')
    print(f'{os.path.basename(__file__)} -e <endpoint> -c <path/to/certificate> -u <user> -p <password> -m <method> -a <alias>')
    print('')
    print('-e, --endpoint    edge gateway endpoint. Usually the Edge gateway hostname.')
    print('-c, --cert_path path to downloaded gateway certificate')
    print('-u, --user        Edge user')
    print('-p, --password    Edge password')
    print('-m, --method      (Optional) Authentication method (linux, winnt, ldap), default is linux')
    sys.exit()

def parse_args(argv):
    endpoint = ""
    certificatePath = None
    user = None
    password = None
    method = "linux"

    try:
        opts, args = getopt.getopt(argv, "he:c:u:p:m:", ["endpoint=", "cert_path=", "user=", "password=", "method="])
    except getopt.GetoptError:
        print_help()

    for opt, arg in opts:
        if opt == '-h':
            print_help()
        elif opt in ("-e", "--endpoint"):
            endpoint = arg
        elif opt in ("-u", "--user"):
            user = arg
        elif opt in ("-p", "--password"):
            password = arg
        elif opt in ("-m", "--method"):

```

```

        method = arg.lower()
    elif opt in ("-c", "--cert_path"):
        certificatePath = arg

    if method not in ['ldap', 'linux', 'winnt']:
        print("not valid method parameter, required are ldap, linux, winnt")
        print_help()

    if (user == None or password == None):
        print("To authenticate against edge user, password have to be passed
together, and the region has to be set to 'edge'")
        print_help()

    if(endpoint == ""):
        print("You must provide a valid and reachable gateway hostname")
        print_help()

    return endpoint,certificatePath, user, password, method

def main(argv):
    # get the command line args
    endpoint, certificatePath, user, password, method = parse_args(argv)

    accessKeyId, secretAccessKey, sessionToken, region=get_credentials(endpoint,
certificatePath, user, password, method)

    print("Copy and paste the following credentials into the shell, they are
valid for 4 hours:")
    print(f"export AWS_ACCESS_KEY_ID={accessKeyId}")
    print(f"export AWS_SECRET_ACCESS_KEY={secretAccessKey}")
    print(f"export AWS_SESSION_TOKEN={sessionToken}")
    print(f"export AWS_REGION={region}")
    print()

if __name__ == "__main__":
    main(sys.argv[1:])

```

2. Führen Sie `get_credentials.py` vom Terminal aus `<gateway_username>` und `<gateway_password>` ersetzen Sie die von Ihnen erstellten Anmeldeinformationen.


```
python3 get_credentials.py -e https://$GATEWAY_HOSTNAME -c $PATH_TO_CERTIFICATE  
-u '<gateway_username>' -p '<gateway_password>' -m 'linux'
```

curl

 Note

Sie müssen [curl](#) installiert haben.

Um die Anmeldeinformationen mit Curl zu erhalten

1. Führen Sie den folgenden Befehl vom Terminal aus <gateway_username>und <gateway_password>ersetzen Sie dabei die von Ihnen erstellten Anmeldeinformationen.

```
curl --cacert $PATH_TO_CERTIFICATE --location \  
-X POST https://$GATEWAY_HOSTNAME:443/authenticate \  
--header 'Content-Type: application/json' \  
--data-raw '{  
  "username": "<gateway_username>",  
  "password": "<gateway_password>",  
  "authMechanism": "linux"  
}
```

Die Antwort sollte wie folgt aussehen:

```
{  
  "username": "sweuser",  
  "accessKeyId": "<accessKeyId>",  
  "secretAccessKey": "<secretAccessKey>",  
  "sessionToken": "<sessionToken>",  
  "sessionExpiryTime": "2022-11-17T04:51:40.927095Z",  
  "authMechanism": "linux",  
  "role": "edge-user"  
}
```

2. Führen Sie im Terminal den folgenden Befehl aus:

```
export AWS_ACCESS_KEY_ID=<accessKeyId>
```

```
export AWS_SECRET_ACCESS_KEY=<secretAccessKey>
export AWS_SESSION_TOKEN=<sessionToken>
export AWS_REGION=edge
```

Schritt 4: Rufen Sie eine Liste der Asset-Modelle auf dem SiteWise Edge-Gateway ab

Da Sie nun über ein signiertes Zertifikat, Ihren SiteWise Edge-Gateway-Hostnamen und temporäre Anmeldeinformationen für Ihr SiteWise Edge-Gateway verfügen, können Sie mithilfe der `ListAssetModels` API eine Liste der Asset-Modelle auf Ihrem SiteWise Edge-Gateway abrufen.

Python

 Note

Sie benötigen [Python3](#), [Boto3](#) und müssen installiert sein. [BotoCore](#)

Um die Liste der Asset-Modelle mit Python abzurufen

1. Erstellen Sie eine Datei mit dem Namen `list_asset_model.py` und kopieren Sie dann den folgenden Code hinein.

```
import json
import boto3
import botocore
import os

# create the client using the credentials
client = boto3.client("iotsitewise",
    endpoint_url= "https://" + os.getenv("GATEWAY_HOSTNAME"),
    region_name=os.getenv("AWS_REGION"),
    aws_access_key_id=os.getenv("AWS_ACCESS_KEY_ID"),
    aws_secret_access_key=os.getenv("AWS_SECRET_ACCESS_KEY"),
    aws_session_token=os.getenv("AWS_SESSION_TOKEN"),
    verify=os.getenv("PATH_TO_CERTIFICATE"),
    config=botocore.config.Config(inject_host_prefix=False))

# call the api using local credentials
response = client.list_asset_models()
print(response)
```

2. Führen Sie `list_asset_model.py` vom Terminal aus.

```
python3 list_asset_model.py
```

curl

 Note

Sie müssen [curl](#) installiert haben.

Um die Liste der Asset-Modelle abzurufen, die Curl verwenden

Führen Sie den folgenden Befehl vom Terminal aus.

```
curl \
  --request GET https://$GATEWAY_HOSTNAME:443/asset-models \
  --cacert $PATH_TO_CERTIFICATE \
  --aws-sigv4 "aws:amz:edge:iotsitewise" \
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \
  -H "x-amz-security-token:$AWS_SESSION_TOKEN"
```

Die Antwort sollte wie folgt aussehen:

```
{
  "assetModelSummaries": [
    {
      "arn": "arn:aws:iotsitewise:{region}:{account-id}:asset-model/{asset-
model-id}",
      "creationDate": 1.669245291E9,
      "description": "This is a small example asset model",
      "id": "{asset-model-id}",
      "lastUpdateDate": 1.669249038E9,
      "name": "Some Metrics Model",
      "status": {
        "error": null,
        "state": "ACTIVE"
      }
    },
    .
    .
  ]
}
```

```
],  
  "nextToken": null  
}
```

Hosten Sie ein SiteWise Edge-Gateway auf Siemens Industrial Edge

Hosten Sie Ihr Gateway mithilfe der Edge-Anwendung auf Siemens Industrial AWS IoT SiteWise Edge. Genau wie bei können Sie mit AWS IoT Greengrass V2 SiteWise Edge on Fertigungsprozesse optimieren oder betriebliche Arbeitsabläufe verbessernSiemens Industrial Edge.

Sie können Daten von Ihrem Siemens Industrial Edge-Gerät in Ihr AWS Konto aufnehmen, indem Sie ein SiteWise Edge-Gateway auf dem Gerät ausführen. Fordern Sie dazu beim AWS IoT SiteWise Edge-Supportteam Zugriff auf die SiteWise Edge-Anwendung an. Erstellen Sie anschließend eine SiteWise Edge-Gateway-Ressource mit einem Bereitstellungsziel für das Siemens Industrial Edge-Gerät — neu. Laden Sie anschließend die Konfigurationsdatei herunter und laden Sie sie über das Siemens Industrial Edge Management Portal in Ihre Anwendung hoch. Weitere Informationen zum Ausführen von AnwendungenSiemens Industrial Edge, einschließlich der Einrichtung der erforderlichen Siemens Ressourcen, finden Sie unter [Was ist Industrial Edge?](#) in der Siemens-Dokumentation.

Note

Siemens ist kein Anbieter oder Lieferant für SiteWise Edge. Der Siemens Industrial Edge Marketplace ist ein unabhängiger Marktplatz.

Themen

- [Sicherheit](#)
- [Siemens Secure Storageund die AWS IoT SiteWise Edge-Anwendung](#)
- [Ziele für Siemens Industrial Edge Geräte](#)
- [Migrieren Sie von der Vorschau-Anwendung](#)
- [Fehlerbehebung](#)
- [AWS IoT SiteWise Changelog der Edge-Anwendung](#)

- [Anforderungen für die AWS IoT SiteWise Edge-Anwendung](#)
- [Erstellen Sie ein Gateway für Siemens Industrial Edge](#)
- [Erstellen Sie eine Siemens Databus user für die Anwendung](#)
- [Greifen Sie auf die AWS IoT SiteWise Edge-Anwendung zu](#)
- [Installieren Sie die Anwendung auf einem Siemens Gerät](#)
- [Aktualisieren Sie die AWS IoT SiteWise Edge-Anwendungskonfiguration](#)
- [AWS IoT SiteWise — Daten, die durch die Nutzung dieses Dienstes generiert wurden](#)

Sicherheit

Im Rahmen des [Modells der gemeinsamen Verantwortung](#) zwischen AWS unseren Kunden und unseren Partnern wird im Folgenden beschrieben, wer für die verschiedenen Sicherheitsaspekte verantwortlich ist:

Kundenverantwortung

- Überprüfung des Partners.
- Konfiguration des Netzwerkzugangs, der dem Partner gewährt wurde.
- Physisches Sichern des Geräts, auf dem SiteWise Edge ausgeführt wird.

AWS Verantwortung

- Isolierung des Partners von den AWS Cloud-Ressourcen des Kunden.

Verantwortung des Partners

- Verwendung sicherer Standardeinstellungen.
- Sorgen Sie mit Patches und anderen geeigneten Updates dafür, dass die Lösung im Laufe der Zeit sicher bleibt.
- Vertrauliche Behandlung von Kundendaten.
- Überprüfung anderer Anwendungen, die auf dem Partner-Marktplatz verfügbar sind.

Siemens Secure Storage und die AWS IoT SiteWise Edge-Anwendung

Zum Schutz von Anmeldeinformationen und Geheimnissen, die für die Ausführung der AWS IoT SiteWise Edge-Anwendung erforderlich sind, Siemens Industrial Edge bietet es Mechanismen zum sicheren Speichern der Anmeldeinformationen auf dem Gerät. Die AWS IoT SiteWise Edge-Anwendung kann auf einem Gerät nicht ausgeführt werden, wenn sie das sichere Speichern dieser

Anmeldeinformationen nicht unterstützt. Ausführungsfehler, die durch fehlende Secure Storage-Unterstützung verursacht werden, werden in Protokolldateien protokolliert.

Für die Installation und Ausführung der AWS IoT SiteWise Edge-Anwendung sind mindestens die folgenden Betriebssystemversionen erforderlich. Aktualisieren Sie Ihre Geräte auf die neuesten Versionen, um die Anwendung zu installieren.

- Für virtuelle Geräte: IEVD Version 1.19 oder höher
- Für physische Geräte: IED-OS Version 2.2 oder höher

Die AWS IoT SiteWise Edge-Anwendung Siemens Industrial Edge wird erst ausgeführt, wenn Sie Ihr Gerät aktualisiert haben.

Ziele für Siemens Industrial Edge Geräte

Wenn die AWS IoT SiteWise Edge-Anwendung aktiviert ist, helfen Ziele dabei, Daten vorzubereiten, bevor sie AWS IoT SiteWise zur weiteren Analyse und Verteilung an sie gesendet werden. Sie können Datenzeileinstellungen für die gepufferte Datenaufnahme mit Amazon S3 konfigurieren oder die Datenaufnahme in Echtzeit verwenden. Beide ermöglichen es Ihnen, MQTT-Themen mithilfe von Pfadfiltern auf dem Gerätebereitstellungsziel zu abonnieren. Siemens Industrial Edge

Das Siemens Industrial Edge Bereitstellungsziel auf Ihrem Gateway unterstützt zwei primäre Datenverarbeitungsmethoden:

- AWS IoT SiteWise Einstellungen in Echtzeit — Daten werden direkt an sie gesendet, AWS IoT SiteWise sobald sie erfasst wurden
- AWS IoT SiteWise mithilfe der Amazon S3-Einstellungen gepuffert — Daten werden gesammelt und temporär in Stapeln gespeichert, bevor sie an Amazon S3 gesendet werden

Weitere Informationen zur Konfiguration dieser Optionen finden Sie unter [Fügen Sie ein AWS IoT SiteWise gepuffertes Ziel mit Amazon S3 hinzu](#) und [Fügen Sie ein AWS IoT SiteWise Edge-Echtzeitziel hinzu](#)

Präfixe für Pfadfilter

Pfadfilter für Gateways, die Siemens Industrial Edge Bereitstellungsziele verwenden, kombinieren sowohl das Thema als auch den Namen des Datenstroms, um eine eindeutige Kennung für Ihre

Daten zu erstellen. Das kombinierte Thema mit dem Namen des Datenstroms wird in Siemens Industrial Edge Gateways als Präfix bezeichnet. Dies unterscheidet sich von selbst gehosteten Gateways, bei denen Pfadfilter ausschließlich auf MQTT-Themen basieren.

Example Pfadfilterstruktur für Siemens-Datenströme

Ein typischer Pfadfilter für einen Siemens-Datenstrom umfasst sowohl den Themenpfad als auch den Namen des Datenstroms:

```
ie/d/device1/application1/datastream1
```

Wobei Folgendes gilt:

- `ie/d/` ist das erforderliche Präfix für Siemens-Datenströme
- `device1/application1` steht für den hierarchischen Pfad
- `datastream1` ist der spezifische Datenstromname

Note

Achten Sie bei der Arbeit mit Siemens Industrial Edge Datenströmen darauf, dass Sie sowohl die Themen Metadaten (`ie/m/`) als auch Daten (`ie/d/`) in Ihre Pfadfilter aufnehmen, um vollständige Informationen zu Ihren Datenströmen zu erhalten.

Ziele und Pfadfilter

In den folgenden Themen erfahren Sie mehr über Ziele und Pfadfilter in MQTT-fähigen Gateways:

- [Verstehen Sie AWS IoT SiteWise Edge-Ziele](#)
- [Fügen Sie ein AWS IoT SiteWise Edge-Echtzeitziel hinzu](#)
- [Fügen Sie ein AWS IoT SiteWise gepuffertes Ziel mit Amazon S3 hinzu](#)
- [Machen Sie sich mit Pfadfiltern für AWS IoT SiteWise Edge-Ziele vertraut](#)
- [Fügen Sie Pfadfilter zu AWS IoT SiteWise Edge-Zielen hinzu](#)
- [AWS IoT SiteWise Edge-Ziele verwalten](#)

Migrieren Sie von der Vorschau-Anwendung

Wenn Sie SiteWise Edge Siemens Industrial Edge während der Vorschauphase ausgeführt haben, müssen Sie ein Upgrade von der Vorschauversion, Version 1.0.1, auf die neueste Version durchführen. Gehen Sie zur Migration wie folgt vor:

1. Erstellen Sie neue SiteWise Edge-Gateways. Weitere Informationen finden Sie unter [Erstellen Sie ein Gateway für Siemens Industrial Edge](#).
2. Erstellen Sie Siemens Databus user für jedes neue Gateway ein neues. Weitere Informationen finden Sie unter [Erstellen Sie eine Siemens Databus user für die Anwendung](#).
3. Deinstallieren Sie die AWS IoT SiteWise Edge-Gateway-Anwendung Version 1.0.1 auf Ihrem IED.

Note

Bereiten Sie sich auf Unterbrechungen des Datenflusses vor, wenn Sie die AWS IoT SiteWise Ressourcen neu konfigurieren, die zuvor von der Vorschauversion der AWS IoT SiteWise Edge-Anwendung verwendet wurden. Der Datenverlauf wird zwar beibehalten, es besteht jedoch die Möglichkeit eines Datenverlusts, wenn Sie das neue Gateway erneut installieren.

4. Löschen Sie die SiteWise Edge-Gateways, die Sie während der Vorschau im [AWS-IoT-SiteWise-Konsole](#) erstellt haben.
5. Installieren Sie die AWS IoT SiteWise Edge-Gateway-Anwendung auf IED mithilfe der neuen Gateway-Konfigurationsdatei. Weitere Informationen finden Sie unter [Installieren Sie die Anwendung auf einem Siemens Gerät](#).

Important

Durch die Installation des neuen Gateways wird die Vorschauversion der SiteWise Edge-Anwendung überschrieben. Es ist nicht möglich, nach der Installation der aktuellen Version zu Version 1.0.1 zurückzukehren.

Nach der Konfiguration des neuen Gateways und Siemens Databus user fließen Ihre Daten zu Ihren Immobilien.

Sie können Ihre SiteWise Edge-Anwendung auch direkt von der Version aktualisieren. Eine neue Gateway-Konfiguration ist jedoch weiterhin erforderlich.

Fehlerbehebung

Informationen zur Fehlerbehebung beim SiteWise Edge-Gateway auf Ihrem Siemens Industrial Edge Gerät finden Sie unter [Fehlerbehebung bei der AWS IoT SiteWise Edge-Anwendung auf Siemens Industrial Edge](#).

Sie können auch auf [AWS re:POST](#) zugreifen, um Antworten auf Ihre Fragen zu finden.

AWS IoT SiteWise Changelog der Edge-Anwendung

In der folgenden Tabelle werden die Änderungen in den einzelnen Versionen der AWS IoT SiteWise Edge-Anwendung beschrieben.

Version	Änderungen	
3.0.1	Fehlerkorrekturen und Verbesserungen • Wendet die maximale Anzahl eindeutiger Aliase pro AWS IoT SiteWise gepufferter Aufnahme an, um dem Cloud-Limit von 10.000 zu entsprechen. • Behebt ein Problem, das zur doppelten Datenaufnahme führt. • Behebt ein Problem beim Synchronisieren großer Konfigurationen. • Behebt Verzögerungen bei der Initialisierung von Checkpoints, die sich zuvor auf den Startvorgang ausgewirkt hatten.	
3.0.0	Neue Features • Fügt der AWS IoT SiteWise Konsole zusätzliche Konfigurationsoptionen für Gateways hinzu, die auf folgenden Geräten installiert sind: Siemens Industrial Edge • Fügen Sie Amazon S3 S3-Ziele hinzu und konfigurieren Sie sie. • Entfernen Sie das AWS IoT SiteWise Echtzeitziel. • Fügen Sie Filter für jedes Ziel hinzu, um das Datenrouting zu steuern.	

Version	Änderungen	
	• Fügt die Option für eine globale Präfixkonfiguration hinzu, um alle Daten vom SiteWise IoT-Publisher im Siemens IE Installationsprogramm zu senden. Das Präfix wird nach dem Filtern angewendet. Fehlerkorrekturen und Verbesserungen • Fügt die Möglichkeit hinzu, Verbindungen zum Siemens Industrial Edge-Datenbus erneut herzustellen, falls der erste Versuch der AWS IoT SiteWise Edge-Anwendung fehlschlägt.	
2.0.1	Fehlerkorrekturen und Verbesserungen • Behebt ein Problem, bei dem die App in den Fehlerstatus überging und beendet wurde, wenn sie beim Start keine AWS Anmeldeinformationen abrufen konnte. Fügt Unterstützung für Wiederholungsversuche hinzu, bis die Anmeldeinformationen erfolgreich abgerufen wurden.	
2.0.0	• Die AWS IoT SiteWise Edge-Anwendung ist jetzt allgemein verfügbar. • Die Anwendung erfordert Siemens IEVD Version 1.19 oder Siemens IED-OS Version 2.2. • Leistungsverbesserungen: Reduzierte Speicher- und CPU-Auslastung. • Verbesserungen beim Debuggen: Sie können jetzt eine optionale Konfigurationsdatei hochladen, um Debug-Logs zu aktivieren. • Sicherheitsverbesserungen: Die Anwendung verwendet SecureStorage API, um Anmeldeinformationen sicher auf dem Gerät zu speichern. • DockerDigest-Wert: sha256:4a960f29234a190ebb5224c1fd0f3e99faafccc4cb3d93ca13fef247b6656d18	

Version	Änderungen	
1.0.1	Erstversion	

Anforderungen für die AWS IoT SiteWise Edge-Anwendung

Um AWS IoT SiteWise Edge auf laufen zu lassen Siemens Industrial Edge, benötigen Sie Folgendes:

- Ein [Siemens Digital Exchange Platform-Konto](#).
- Ein Siemens Industrial Edge Hub-Konto (iehub).
- Eine Siemens Industrial Edge Management Instanz.
 - Der IE App Configuration Service. Weitere Informationen finden Sie [Installing the IE App Configuration Service manually](#) in der Siemens Industrial Edge Management Dokumentation.
- Zugriff auf Version 2.0.1 oder höher der AWS IoT SiteWise Edge-Anwendung. Weitere Informationen finden Sie unter [Greifen Sie auf die AWS IoT SiteWise Edge-Anwendung zu](#).
- Entweder ein Siemens Industrial Edge Gerät (IED) oder ein Siemens Industrial Edge virtuelles Gerät (IEVD).
 - Mindestens 15 GB Festplattenspeicher für Hardwareanforderungen.
 - 1 GB RAM mit zusätzlichen 1 GB Swap-Speicher.
 - Gerätekonfiguration, um ausgehenden Datenverkehr an den Ports 443 und 8883 zuzulassen.
 - Ein x86-64-Bit-Prozessor.
 - Siemens Industrial Edge Management Version 1.13.10 oder höher.
 - Konformität des Geräts mit den Anforderungen. Siemens Secure Storage
 - Auf virtuellen Geräten IEVD-Version 1.19 oder höher.
 - Auf physischen Geräten IED-OS Version 2.2 oder höher.
 - Die neueste Version von Docker Compose.
 - Docker Engine Version 18.09.1 oder höher.
- Domänenzugriff erforderlich. Weitere Informationen finden Sie unter [AWS IoT SiteWise Endpunkte](#).

Erstellen Sie ein Gateway für Siemens Industrial Edge

Sobald Sie über die richtigen Siemens-Konten und IEM-Instanzen verfügen, können Sie ein SiteWise Edge-Gateway mit dem Bereitstellungstyp Siemens Industrial Edge-Gerät erstellen.

 Note

Stellen Sie sicher, dass Sie alle Anforderungen für den Betrieb eines Geräts erfüllen. Siemens Industrial Edge Management. Weitere Informationen finden Sie unter [Anforderungen für die AWS IoT SiteWise Edge-Anwendung](#).

Um die Konfigurationsdatei zu erstellen

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie Create gateway (Gateway erstellen).
4. Wählen Sie als Bereitstellungstyp Siemens Industrial Edge-Gerät aus.
5. Geben Sie einen Namen für Ihr SiteWise Edge-Gateway ein oder verwenden Sie den von generierten Namen AWS IoT SiteWise.
6. (Optional) Gehen Sie unter „Erweiterte Konfiguration“ wie folgt vor:
 - Geben Sie einen Namen für Ihr AWS IoT Core Ding ein oder verwenden Sie den Namen, der von generiert wurde AWS IoT SiteWise.
7. Wählen Sie Create gateway (Gateway erstellen).
8. Wählen Sie im Dialogfeld SiteWise Edge-Gateway-Konfigurationsdatei generieren die Option Generieren und herunterladen aus. AWS IoT SiteWise generiert automatisch eine Konfigurationsdatei, die Sie zur Konfiguration der AWS IoT SiteWise Edge-Anwendung verwenden.

 Important

Bewahren Sie Ihre Gateway-Konfigurationsdatei als Backup für den Fall auf, dass Sie Ihre AWS IoT SiteWise Edge-Anwendungsinstanz wiederherstellen müssen. Zu diesem Zweck können Sie Ihre SiteWise Edge-Gateway-Konfigurationsdatei sicher speichern. [AWS Secrets Manager](#) Secrets Manager speichert, verwaltet und ruft vertrauliche Informationen sicher ab. Wenn Sie diese Konfigurationsdatei verlegen oder löschen, können Sie Ihre AWS IoT SiteWise Edge-Anwendungsinstanz nicht wieder mit dem ursprünglichen Gateway verbinden, falls Sie sie wiederherstellen müssen. Sie müssen sowohl ein neues Gateway als auch eine neue Konfigurationsdatei erstellen.

Erstellen Sie eine Siemens Databus user für die Anwendung

AWS IoT SiteWise Edge on Siemens Industrial Edge nimmt Daten aus der Siemens Databus Anwendung auf. Um SiteWise Edge mit dem zu verbinden Siemens Databus, benötigen Sie eine, Siemens Databus user die Zugriff auf die Daten bietet, auf die Sie sicher übertragen möchten AWS IoT SiteWise. Erstellen Sie zunächst eine Siemens Databus user und geben Sie dann die Anmeldeinformationen für die SiteWise Edge-Anwendung ein.

So erstellen Sie eine Siemens Databus user

1. Wählen Sie in Ihrer Siemens Industrial Edge Management Instanz im Abschnitt Plattformanwendungen die Option Edge Management aus.
2. Wählen Sie das Symbol für Datenverbindungen.
3. Wählen Sie Datenbus aus. Eine Liste Ihrer verbundenen Geräte wird angezeigt.
4. Wählen Sie das Gerät aus, um eine Verbindung zur AWS IoT SiteWise Edge-Anwendung herzustellen.
5. Wählen Sie Launch (Starten) aus. Das Databus Configurator für Ihr ausgewähltes Gerät wird angezeigt.
6. Erstellen Sie unter Benutzer einen Benutzer für Ihr Edge-Gerät. Weitere Informationen zum Erstellen eines Benutzers finden Sie unter [Benutzer](#) in der Dokumentation zu Siemens Industrial Edge Management.
7. Wählen Sie die Themen aus, auf die Sie Zugriff haben Siemens Databus sollen. Diese Themen schränken ein, worauf AWS IoT SiteWise Edge zugreifen kann.

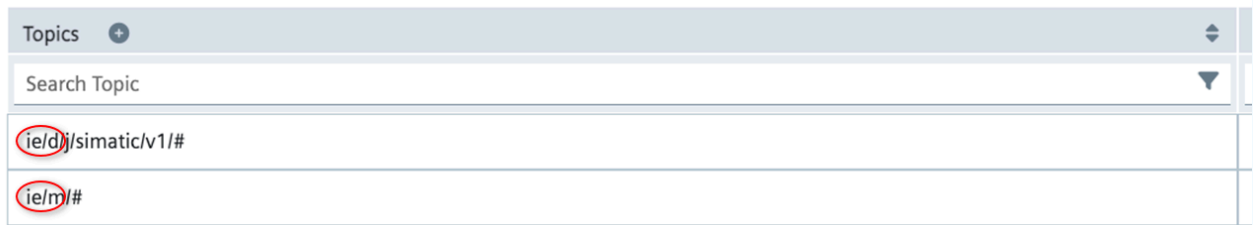
Important

Alle Themen, auf die a Zugriff Siemens Databus user hat, werden veröffentlicht AWS IoT SiteWise.

Note

Siemens Databus users benötigen Zugriff sowohl auf Daten- als auch auf Metadaten Themen. Themen, die mit beginnen, ie/d sind Datenthemen. Und Themen, die mit beginnen, ie/m sind Metadaten Themen. Teilen Sie Themen paarweise, sodass

SiteWise Edge sowohl auf Daten als auch auf Metadaten für das jeweilige Thema zugreifen kann.



8. Legen Sie die entsprechenden Berechtigungen für Ihre Siemens Databus Konfiguration fest.

Nachdem Sie Ihre Siemens Databus Konfiguration erstellt haben, können Sie die AWS IoT SiteWise Edge-Anwendung auf Ihrem installierenSiemens Industrial Edge Management. Weitere Informationen finden Sie unter [Installieren Sie die Anwendung auf einem Siemens Gerät](#).

Sie können optional auch Ziele und Pfadfilter für Ihr Siemens Industrial Edge Gateway konfigurieren. Weitere Informationen finden Sie unter [Ziele und Pfadfilter](#).

Greifen Sie auf die AWS IoT SiteWise Edge-Anwendung zu

Um Zugriff auf die AWS IoT SiteWise Edge-Anwendung zu erhaltenSiemens Industrial Edge, [senden Sie eine E-Mail](#) mit der Bitte um Zugriff an das SiteWise Edge-Supportteam.

Nehmen Sie die folgenden Informationen in Ihre E-Mail auf:

- Ihr Name und Ihre Kontaktinformationen
- Unternehmensname
- Siemens Industrial EdgeMandanten-ID

Installieren Sie die Anwendung auf einem Siemens Gerät

Nachdem Sie Zugriff auf die AWS IoT SiteWise Edge-Anwendung erhalten haben, indem Sie eine E-Mail an das SiteWise Edge-Supportteam sendenSiemens Industrial Edge, weisen Sie die Anwendung einer Instanz von Siemens Industrial Edge Management zu. Anschließend können Sie die AWS IoT SiteWise Edge-Anwendung auf Ihrem Gerät installieren.

Um die AWS IoT SiteWise Edge-Anwendung zu installieren

1. Stellen Sie sicher, dass Docker der darin enthaltene Überblick Siemens Industrial Edge Management mit der neuesten Version übereinstimmt, die in der [AWS IoT SiteWise Changelog der Edge-Anwendung](#) aufgeführt ist.

Weitere Informationen zum Auffinden des Docker Digest-Werts für Siemens finden Sie in der Dokumentation unter [Verwaltung einer App](#) im Siemens Industrial EdgeGerät. Siemens

Siemens Industrial Edge Managementunterstützt jeweils eine Version der AWS IoT SiteWise Edge-Anwendung. Führen Sie diesen Schritt aus, um sicherzustellen, dass Sie die neueste Version der Anwendung verwenden, bevor Sie die AWS IoT SiteWise Edge-Anwendung auf Ihrem Siemens Industrial Edge Gerät installieren.

2. Weisen Sie die AWS IoT SiteWise Edge-Anwendung zuSiemens Industrial Edge Management. Weitere Informationen finden Sie unter [Verwaltung einer App](#) im Abschnitt Industrial Edge Management der Siemens Dokumentation.
3. Durchsuchen Sie in Edge Management den Katalog für den AWS IoT SiteWise Edge und wählen Sie ihn aus.
4. Wählen Sie Installieren aus.

Note

Wenn eine Schaltfläche „Kontaktieren Sie uns“ angezeigt wird, wählen Sie sie aus und folgen Sie den Schritten, um Zugriff auf die AWS IoT SiteWise Edge-Anwendung anzufordernSiemens Industrial Edge. Weitere Informationen finden Sie unter [Greifen Sie auf die AWS IoT SiteWise Edge-Anwendung zu](#).

5. Wählen Sie in den Optionen für Schemakonfigurationen die Option Databus_Configuration aus.
6. Geben Sie den Benutzernamen und das Passwort für die Datenbus-Konfiguration ein. Weitere Informationen zum Erstellen eines finden Sie Siemens Databus user unter[Erstellen Sie eine Siemens Databus user für die Anwendung](#).
7. Wählen Sie das kleine, runde graue Häkchensymbol neben Databus_Configuration, um die Farbe des Symbols grün zu ändern.

Note

Die Eingabekonfigurationen gelten nur, wenn das Häkchensymbol von grau nach grün wechselt. Andernfalls wird die Eingabekonfiguration ignoriert.

Databus_Configuration Invalid Configuration

Provide Databus user credentials for the AWS IoT SiteWise Edge application to collect data from the Databus application. SiteWise Edge will publish all topics to the cloud that the user has access to. You can manage access using the Databus app
NOTE: Make sure to select the round gray checkmark icon next to Databus_Configuration. The icon turns green once selected.

Username*

is a required property

Password*

is a required property

Databus_Configuration

Provide Databus user credentials for the AWS IoT SiteWise Edge application to collect data from the Databus application. SiteWise Edge will publish all topics to the cloud that the user has access to. You can manage access using the Databus app
NOTE: Make sure to select the round gray checkmark icon next to Databus_Configuration. The icon turns green once selected.

Username*

testUser

Password*

.....

8. Wählen Sie Weiter, um zu Andere Konfigurationen zu gelangen, wo Sie Ihre Gateway-Konfigurationsdatei hochladen können.
9. Wählen Sie SiteWise_Edge_Gateway_Config als Speicherort für den Upload der Gateway-Konfigurationsdatei.

Note

Stellen Sie sicher, dass Sie _Edge_Gateway_Config und nicht _Edge_Support_Config_Optional wählen. SiteWise SiteWise

10. Wählen Sie das Gerät aus, auf dem die Anwendung installiert werden soll.
11. Wählen Sie Install Now (Jetzt installieren) aus.

Sie können die Publisher-Komponente optional so konfigurieren, dass Daten in die AWS Cloud exportiert werden. Weitere Informationen finden Sie unter [Konfiguration der AWS IoT SiteWise Publisher-Komponente](#).

Informationen zum Konfigurieren von Zielen für Ihr Siemens Industrial Edge Gateway finden Sie unter [Ziele und Pfadfilter](#).

Aktualisieren Sie die AWS IoT SiteWise Edge-Anwendungskonfiguration

Bei der Aktualisierung einer AWS IoT SiteWise Edge-Anwendungskonfiguration sind einige Dinge zu beachten Siemens Industrial Edge.

Note

Jede Änderung an der AWS IoT SiteWise Edge-Anwendungskonfiguration erfordert einen Neustart der Anwendung.

Gründe für einen Neustart der AWS IoT SiteWise Edge-Anwendung

- Eine neue Anwendung Siemens Databus user für die AWS IoT SiteWise Edge-Anwendung.
- Eine Änderung an der Gateway-Konfigurationsdatei (Ihre SiteWise_Edge_Gateway_Config-Datei).
- Ein Update der Proxykonfiguration (das auch einen vollständigen IEVD-Neustart erfordert)
- Um Debug-Logs für Debugging-Probleme zu aktivieren

Die Anwendung wird neu gestartet

1. Wählen Sie in Ihrer Siemens Industrial Edge Management Instanz im Abschnitt Plattformanwendungen die Option Edge Management aus.
2. Wählen Sie Meine installierten Apps aus.
3. Wählen Sie die AWS IoT SiteWise Edge-Anwendung aus.
4. Wählen Sie Neu starten.

AWS IoT SiteWise — Daten, die durch die Nutzung dieses Dienstes generiert wurden

AWS IoT SiteWise auf Siemens Industrial Edge erweitert Cloud-Funktionen auf industrielle Edge-Umgebungen und ermöglicht so die lokale Datenverarbeitung, Analyse und Entscheidungsfindung. SiteWise Edge lässt sich AWS IoT SiteWise in andere AWS Dienste integrieren, um umfassende industrielle AWS IoT Lösungen bereitzustellen.

Arten von Daten

SiteWise Die Edge-Anwendung auf Siemens Industrial Edge generiert Daten über die Leistung, Nutzung und Interaktionen mit anderen AWS Diensten, insbesondere Amazon S3.

Datenvolumen und Datenerfassung

Die Menge der generierten Daten hängt davon ab, wie Sie Ihre Anwendung und Dienste nutzen.

Datenspeicherung

Daten aus Ihrer Anwendung werden sicher auf AWS Servern gespeichert. Sie werden in maschinenlesbaren Formaten gespeichert.

Datenzugriff

Sie können über Ihr Konto auf Ihre Daten zugreifen. AWS Sie können eine Kopie Ihrer Daten herunterladen, indem Sie den Anweisungen folgen, die in dieser Bedienungsanleitung [SiteWise Edge on Siemens](#) aufgeführt sind. Für Konfigurationsdaten für die Anwendung können Sie den Anweisungen in diesem Benutzerhandbuch [Query Industrial Data](#) folgen. Weitere Anweisungen für den Massenzugriff auf und den Export von Daten finden Sie [unter Bulk Operations Export ausführen](#).

Datenverwaltung

Um mehr über die Datenpraktiken Ihrer Anwendungen zu erfahren, lesen Sie bitte unsere [Servicebedingungen](#), unseren [Datenschutzhinweis](#) und die [Servicedokumentation](#), die Materialien zur Verwaltung Ihrer Daten enthält.

Löschung von Daten

Informationen zu den Optionen zur Aufbewahrung und Löschung von Daten finden Sie auf den folgenden Seiten im Benutzerhandbuch — [Datenspeicher verwalten](#), [Datenströme löschen und Modelle und Ressourcen löschen](#).

Teilen von Daten mit anderen

Sie können Dritte über unsere [Identity and Access Management-Prozesse autorisieren, auf Ihre AWS Ressourcen zuzugreifen](#). AWS gibt personenbezogene Daten in begrenzten Fällen, die in der [AWS Datenschutzerklärung](#) aufgeführt sind, an Dritte weiter.

Brauchen Sie Hilfe?

Besuchen Sie [den Kundensupport](#), um unser Support-Team zu erreichen. Dies gilt unbeschadet Ihres Rechts, nach geltendem Recht eine Beschwerde einzureichen.

Inhaber der Daten

Amazon Web Services EMEA SARL, 38 Avenue John F. Kennedy, L-1855, Luxemburg

Ziele und Pfadfilter

Destinationen in AWS IoT SiteWise Edge bieten eine flexible und effiziente Möglichkeit, den Fluss Ihrer Industriedaten von Edge-Geräten in die Cloud zu verwalten. In diesem Abschnitt wird erklärt, wie Sie Ziele konfigurieren, Pfadfilter verwenden, um bestimmte Datenströme weiterzuleiten, und den richtigen Zieltyp für Ihren Anwendungsfall auswählen.

Sie können Ziele und Pfadfilter auf selbst gehosteten MQTT-fähigen V3-Gateways und Gateways verwenden, die in Verbindung mit einer Edge-Anwendung verwendet werden, auf der gehostet wird. AWS IoT SiteWise Siemens Industrial Edge Ziele und Pfadfilter funktionieren nicht mit Classic Streams, V2-Gateways.

Themen

- [Verstehen Sie AWS IoT SiteWise Edge-Ziele](#)
- [Machen Sie sich mit Pfadfiltern für AWS IoT SiteWise Edge-Ziele vertraut](#)
- [Fügen Sie ein AWS IoT SiteWise Edge-Echtzeitziel hinzu](#)
- [Fügen Sie ein AWS IoT SiteWise gepuffertes Ziel mit Amazon S3 hinzu](#)
- [Fügen Sie Pfadfilter zu AWS IoT SiteWise Edge-Zielen hinzu](#)
- [AWS IoT SiteWise Edge-Ziele verwalten](#)

Verstehen Sie AWS IoT SiteWise Edge-Ziele

Verwenden Sie AWS IoT SiteWise Edge-Ziele, um zu bestimmen, wohin Ihre Quelldaten gesendet werden sollen. Sie können Ihr Datenziel auf der Grundlage bestimmter Merkmale auswählen, die Sie

benötigen, wie z. B. Wirtschaftlichkeit, geringe Latenz oder Speicheranforderungen. Integrieren Sie Gerätedaten AWS IoT SiteWise, die von unseren Partnern oder benutzerdefinierten Anwendungen erfasst wurden, um Pfadfilter (Themen) am Netzwerkrand zu veröffentlichen und zu abonnieren. Anschließend können Sie Ihre Gerätedaten modellieren, übertragen und in der Cloud speichern.

Note

Um alle Zielfunktionen auf selbst gehosteten Gateways in vollem Umfang nutzen zu können, führen Sie ein Upgrade auf die neuesten Versionen des SiteWise IoT-Publishers und des IoT SiteWise OPC UA-Collectors durch. Die Stream-Unterstützung wird auf Classic-Streams und V2-Gateways fortgesetzt, um die Kompatibilität mit bestehenden Setups aufrechtzuerhalten. Weitere Informationen finden Sie unter [Klassische Streams, V2-Gateways für AWS IoT SiteWise Edge](#).

Topics

- [Wie SiteWise Edge-Ziele das Datenmanagement verbessern](#)
- [Zieltypen](#)
- [Vergleichen Sie die Zielfunktionen zwischen den Gateway-Versionen](#)
- [Einschränkungen bei der Destination](#)
- [Anwendungsfälle für SiteWise Edge-Ziele](#)

Wie SiteWise Edge-Ziele das Datenmanagement verbessern

Exportieren Sie Daten vom Edge AWS IoT SiteWise in Echtzeit oder stapelweise mit Amazon S3.

Ziele verbessern die Flexibilität und Skalierbarkeit in Ihrer AWS IoT SiteWise Umgebung.

Destinations implementieren ein zentralisiertes Datenverwaltungsmodell, bei dem Quellen Daten in einem zentralen System veröffentlichen. Ziele bestimmen mithilfe von Pfadfiltern, wohin Ihre Daten gesendet werden. Ziele können mehrere Pfadfilter abonnieren.

MQTT-fähige Gateways, unabhängig davon, ob sie selbst gehostet werden oder auf denen sie laufen, verwenden MQTT für die lokale Kommunikation und verfügen über ein standardmäßiges Echtzeitziel, für das Filter eingestellt sind. # Das bedeutet, dass standardmäßig alle Nachrichten zu allen Themen im Echtzeitziel veröffentlicht werden. AWS IoT SiteWise Weitere Informationen finden Sie unter [Machen Sie sich mit Pfadfiltern für AWS IoT SiteWise Edge-Ziele vertraut](#). Sie können in jedem Gateway ein Echtzeitziel hinzufügen.

Zieltypen

Bei der Konfiguration eines Ziels für Ihr Gateway haben Sie zwei Hauptoptionen: Echtzeitkonfiguration mithilfe von AWS IoT SiteWise Amazon S3 und eine gepufferte Konfiguration mit Amazon S3. Jeder Zieltyp hat seine eigenen Einstellungen und Überlegungen.

AWS IoT SiteWise Einstellungen in Echtzeit

Wählen Sie diese Option, um Daten direkt an AWS IoT SiteWise Hot-Tier-Speicher zu senden und so die Erfassung und Überwachung von Daten in Echtzeit zu erleichtern. Die Echtzeiteinstellungen verwalten den Datenfluss, insbesondere wenn bei einem Gateway Verbindungsprobleme mit der Cloud auftreten. Während eines Verbindungsverlusts werden Daten vorübergehend lokal auf dem Gateway gespeichert. Sobald die Verbindung wieder hergestellt ist, werden die gespeicherten Daten automatisch an die Cloud gesendet.

Sie können verschiedene Aspekte des Datenveröffentlichungsprozesses anpassen, z. B. die maximale Datenmenge, die lokal gespeichert werden soll, die Geschwindigkeit, mit der Daten bei der erneuten Verbindung an die Cloud gesendet werden, und den Zeitpunkt, zu dem Daten gelöscht werden sollen, wenn der Speicher seine Kapazität erreicht hat.

Weitere Informationen zu AWS IoT SiteWise Speicherstufen finden Sie unter: [Datenspeicher verwalten in AWS IoT SiteWise](#)

AWS IoT SiteWise mit Amazon S3 S3-Einstellungen gepuffert

Dieser Zieltyp ermöglicht es Ihnen, Daten lokal auf dem Gateway zu puffern und sie regelmäßig stapelweise an einen Amazon S3 S3-Bucket zu senden. Die Daten werden im effizienten Parquet-Format gespeichert, das für analytische Workloads optimiert ist. Sobald sich die Daten in Amazon S3 befinden, können Sie sie AWS IoT SiteWise zur Speicherung, Verarbeitung und Analyse in Amazon S3 importieren.

Wählen Sie diese Option, um Daten stapelweise aufzunehmen und historische Daten auf kostengünstige Weise zu speichern. Sie können Ihren bevorzugten Amazon S3-Bucket-Standort und die Häufigkeit, mit der Daten auf Amazon S3 hochgeladen werden sollen, konfigurieren. Sie können auch wählen, was mit den Daten nach der Aufnahme geschehen soll. AWS IoT SiteWise Sie können wählen, ob die Daten SiteWise sowohl in Amazon S3 als auch in Amazon S3 verfügbar sein sollen, oder Sie können wählen, ob sie automatisch aus Amazon S3 gelöscht werden sollen.

Vergleichen Sie die Zielfunktionen zwischen den Gateway-Versionen

Die Zielfunktion in MQTT-fähigen Gateways optimiert das Datenflussmanagement. Ziele vereinfachen das Datenmanagement durch die zentrale Konfiguration des Datenroutings zu verschiedenen Endpunkten. Dieser Ansatz macht komplexe individuelle Stream-Setups überflüssig, wodurch das Gesamtsystem flexibler und einfacher zu verwalten ist.

Im Vergleich dazu überträgt das Classic Streams, das V2-Gateway und SiteWise Edge Daten von Datenquellen über AWS IoT Greengrass Streams an Publisher, wobei die Datenziele für jede Datenquelle individuell konfiguriert werden.

Mit der AWS IoT SiteWise Zielfunktion wird die Routing-Konfiguration des Herausgebers konsolidiert. Mit der Zielkonfiguration können Sie Ziele und Pfadfilter zentral verwalten. Sie können ganz einfach ein Ziel hinzufügen, Pfadfilter verwalten, unnötige Filter oder Ziele löschen, je nach Ihren Bedürfnissen.

Darüber hinaus verwendet die Zielfunktion MQTT (Message Queuing Telemetry Transport), ein Industriestandardprotokoll, das in industriellen IoT-Anwendungen weit verbreitet ist. Diese Einführung von MQTT trägt dazu bei, AWS IoT SiteWise die Integration mit verschiedenen Geräten und Systemen zu vereinfachen.

Einschränkungen bei der Destination

Zu den aktuellen Einschränkungen für Ziele auf SiteWise Edge-Gateways gehören:

- Das Datenverarbeitungspaket wird auf MQTT-fähigen Gateways nicht unterstützt.
- Die Unterstützung von Datentypen ist auf Datentypen beschränkt. AWS IoT SiteWise Informationen zur Aktivierung der Datentypkonvertierung finden Sie unter [Nicht unterstützte Datentypen werden konvertiert](#).

Anwendungsfälle für SiteWise Edge-Ziele

SiteWise Edge-Destinationen werden in verschiedenen Anwendungen eingesetzt. Hier sind einige wichtige Beispiele:

Industrielle Automatisierung, Überwachung und vorausschauende Wartung in Echtzeit

In industriellen Umgebungen können Sensoren und Geräte in der Fabrik Daten auf SiteWise Edge veröffentlichen. Ziele können so konfiguriert werden, dass relevante Daten gefiltert und weitergeleitet werden, sodass die Maschinenleistung in Echtzeit überwacht und analysiert

werden kann. Sie können relevante MQTT-Themen mithilfe von Pfadfiltern abonnieren, die Daten verarbeiten und dann die verarbeiteten Daten veröffentlichen. Auf diese Weise können Sie verarbeitete Daten selektiv an AWS Cloud-Analysedienste oder lokale Systeme weiterleiten. Hersteller können dann Strategien zur vorausschauenden Wartung implementieren, Produktionsprozesse optimieren und Ausfallzeiten reduzieren.

Intelligente Gebäude, Energieeffizienz und Nutzungsoptimierung

Gebäudeautomationssysteme generieren Datenströme zur Überwachung und Steuerung verschiedener Aspekte eines Gebäudes, wie z. B. HLK-Systeme, Beleuchtung und Zutrittskontrolle. Mit SiteWise Edge können diese Datenströme aufgenommen, verarbeitet und an verschiedene Ziele weitergeleitet werden. Facility Manager können Ziele so konfigurieren, dass sie relevante Daten filtern und weiterleiten, was erweiterte Funktionen wie Energieeffizienzmaßnahmen und Belegungsoptimierung ermöglicht und gleichzeitig Datenschutz und Compliance gewährleistet.

Diese Anwendungsfälle zeigen, wie die Destinationsfunktion in SiteWise Edge in verschiedenen Branchen genutzt werden kann, um Daten effizient aufzunehmen, zu verarbeiten und weiterzuleiten. Dies ermöglicht erweiterte Funktionen wie Echtzeitüberwachung, vorausschauende Wartung, Energieeffizienz und Ferndiagnose und gewährleistet gleichzeitig Datenschutz und Compliance.

Machen Sie sich mit Pfadfiltern für AWS IoT SiteWise Edge-Ziele vertraut

Themen

- [Anforderungen an Pfadfilter](#)
- [Bewährte Methoden für Pfadfilter](#)
- [Pfadfilter für OPC UA-Server](#)
- [Sonderzeichen in Pfadfilternamen](#)

Jedes Ziel ist so konfiguriert, dass es Daten an AWS IoT SiteWise oder Amazon S3 weiterleitet. Pfadfilter ermöglichen es Ihnen, bestimmte Daten auszuwählen, die beim Empfang von MQTT-Nachrichten für ein Ziel gefiltert werden sollen. Pfadfilter stellen die logischen Namen Ihrer Datenströme dar und dienen als Abonnements für die gewünschten MQTT-Themen.

In MQTT sind Daten in Topics organisiert, bei denen es sich um hierarchische Zeichenketten handelt, die durch Schrägstriche (/) getrennt sind. / Beispielsweise könnte ein Gerät Temperaturdaten zu dem Thema veröffentlichen. `home/livingroom/sensor1/temperature` `home/livingroom/`

`sensor1` stellt hier den Pfad oder den logischen Namen des Sensors dar und `temperature` ist der Datentyp, der veröffentlicht wird.

Sie können Pfadfilter verwenden, um bestimmte Themen oder eine Reihe von Themen mithilfe von Platzhaltern (+und#) zu abonnieren. Der + Platzhalter entspricht einer einzelnen Ebene in der Themenhierarchie. `home/+sensor1/temperature` würde zum Beispiel mit `home/livingroom/sensor1/temperature` und `home/bedroom/sensor1/temperature` übereinstimmen. Wenn der # Platzhalter am Ende eines Filters verwendet wird, entspricht er mehreren Ebenen.

Sie können innerhalb eines Pfadfilternamens auch eine Vielzahl von Zeichen verwenden, die in der MQTT-Spezifikation normalerweise nicht zulässig sind. Diese Zeichen funktionieren nicht als Platzhalter, wenn sie innerhalb eines Namens verwendet werden. AWS IoT SiteWise konvertiert diese Zeichen mithilfe von Kodierung, um die MQTT-Konformität zu gewährleisten und gleichzeitig Ihre ursprüngliche Namensstruktur beizubehalten. Diese Funktion ist besonders nützlich, um bestehende Namenskonventionen aus anderen Systemen zu berücksichtigen. Weitere Informationen finden Sie unter [Sonderzeichen in Pfadfilternamen](#).

Durch die sorgfältige Auswahl der entsprechenden Pfadfilter können Sie steuern, welche Daten an ein bestimmtes Ziel gesendet werden. Passen Sie den Datenfluss mithilfe von Pfadfiltern an die Anforderungen Ihres IoT-Systems an.

Anforderungen an Pfadfilter

Beachten Sie bei der AWS-IoT-SiteWise-Konsole Eingabe von Pfadfiltern mithilfe von Folgendes:

- Pfadfilter werden durch eine neue Linie begrenzt, wobei jede Zeile einen separaten Pfadfilter darstellt.
- Einzelne Pfadfilter können zwischen 1 und 65.535 Byte haben.
- Ein Pfadfilter darf nicht leer sein.
- Nullwerte (U+0000) sind nicht zulässig.
- Sie können bis zu 100 Pfadfilter oder 65.535 Zeichen gleichzeitig eingeben, je nachdem, welcher Grenzwert zuerst erreicht wird.
- Das Gesamtlimit liegt bei 20.000 Pfadfiltern für alle Ziele auf einem Gateway zusammen.
- Sie können die \$ Zeichen%, #+, und in Pfadfilternamen verwenden, konvertiert sie jedoch AWS IoT SiteWise automatisch in die URI-Kodierung.

Bewährte Methoden für Pfadfilter

Beachten Sie bei der Erstellung von Pfadfiltern für Ihre AWS IoT SiteWise Ziele die folgenden Strategien, um Ihre Daten effektiv zu verwalten.

- Strukturieren Sie Ihre Filter so, dass sie Ihre Gerätehierarchie widerspiegeln. Erfasst beispielsweise in einer `factory/+/machine/#` Produktionsumgebung Daten von allen Maschinen in verschiedenen Produktionslinien.
- Verwenden Sie spezifische Ebenen für Gerätetypen, Standorte oder Funktionen. Beispiel, `factory/assembly-line/robot/temperature`. Oder, in der intelligenten Landwirtschaft `farm/+/crop/+/moisture`, um den Feuchtigkeitsgehalt verschiedener Nutzpflanzen auf verschiedenen Feldern zu überwachen.
- Setzen Sie Platzhalter strategisch ein: Verwenden Sie `+` für Variationen auf einer einzelnen Ebene und `#` zur Erfassung aller nachfolgenden Ebenen. Verfolgt `building/+/+/energy-consumption` beispielsweise den Energieverbrauch in verschiedenen Zonen und Stockwerken eines Gebäudes. Dabei wird davon ausgegangen, dass die erste `+` Methode alle Stockwerke und die zweite alle Zonen `+` erfasst.
- Sorgen Sie für ein ausgewogenes Verhältnis zwischen Spezifität und Flexibilität, indem Sie Filter erstellen, die spezifisch genug sind, um relevante Daten zu erfassen, aber flexibel genug sind, um future Änderungen Rechnung zu tragen. `site/+/equipment-type/+/measurement` ermöglicht beispielsweise das Hinzufügen neuer Standorte oder Gerätetypen, ohne die Filterstruktur zu ändern.

Testen Sie Ihre Filter gründlich, um sicherzustellen, dass sie die beabsichtigten Daten erfassen und mit der Architektur und den Zielen Ihres IoT-Systems übereinstimmen.

Pfadfilter für OPC UA-Server

Für OPC UA-Server müssen Ihre Pfadfilter den OPC UA-Tagnamen entsprechen. Die letzte Ebene Ihres Pfadfilters muss exakt mit dem OPC UA-Tagnamen übereinstimmen. Wenn Ihr OPC UA-Tag beispielsweise lautet `Device1.Temperature`, könnte es Ihr Pfadfilter sein. `factory/line1/Device1.Temperature` In den vorherigen Ebenen können Sie Platzhalter verwenden, `factory/+/Device1.Temperature` um beispielsweise das Tag über mehrere Produktionslinien hinweg zu erfassen. Falls Ihre Pfadfilternamen Sonderzeichen enthalten, finden Sie [Sonderzeichen in Pfadfilternamen](#) weitere Informationen unter.

Sonderzeichen in Pfadfilternamen

AWS IoT SiteWise berücksichtigt Zeichen, die häufig in Industrieprotokollen wie OPC UA verwendet werden und die normalerweise in Standard-MQTT-Themennamen nicht zulässig sind. Diese Funktion ermöglicht eine reibungslosere Integration von Industriesystemen mit MQTT-basierten Architekturen.

Note

Unsere Behandlung von Sonderzeichen ist zwar hilfreich für die Integration und Migration, es wird jedoch empfohlen, sich bei neuen Implementierungen nach Möglichkeit an die standardmäßigen MQTT-Namenskonventionen zu halten, um eine umfassendere Kompatibilität zu gewährleisten.

AWS IoT SiteWise Normalisiert beim Empfang von Daten aus industriellen Quellen Themennamen mithilfe der URI-Kodierung für Sonderzeichen:

- %wird %25 (zuerst als Escape-Zeichen kodiert)
- #wird %23
- +wird %2B
- \$wird %24 (nur wenn am Anfang eines Themas)

Diese Kodierung stellt sicher, dass Quelldaten, die diese speziellen MQTT-Zeichen enthalten, sicher als MQTT-Themennamen verwendet werden können, wobei die ursprünglichen industriellen Namenskonventionen beibehalten werden.

Example: Sonderzeichen in Pfadfilternamen

Im Folgenden finden Sie Beispiele dafür, wie Namen von Industriethemen in AWS IoT SiteWise Pfadfiltern erscheinen könnten:

- Factory1/Line#2/Sensor+3wird Factory1/Line%232/Sensor%2B3
- Plant%A/Unit\$1/Tempwird Plant%25A/Unit%241/Temp
- Site1/#Section/+Nodewird Site1/%23Section/%2BNode

Wenn Sie Abonnements erstellen oder Themennamen in ansehen AWS IoT SiteWise, werden Ihnen die unverschlüsselten Originalversionen angezeigt. Die Kodierung erfolgt automatisch, um die MQTT-Konformität sicherzustellen.

Fügen Sie ein AWS IoT SiteWise Edge-Echtzeitziel hinzu

Mit dem Echtzeit-Zieltyp können Sie IoT-Daten in Echtzeit direkt von Ihren Geräten und Gateways in den AWS IoT SiteWise Speicher streamen. Diese Option ist ideal für Anwendungsfälle, bei denen Daten sofort bei der Generierung aufgenommen und verarbeitet werden müssen, ohne dass eine Stapelverarbeitung oder Pufferung erforderlich ist. Sie können in jedem Gateway nur ein Echtzeitziel konfigurieren, da es kontinuierlich Daten streamt. AWS IoT SiteWise

Note

Ein Duplikat TQVs kann zu einer doppelten Aufladung führen.

Um ein Ziel in Echtzeit hinzuzufügen

Verwenden Sie die AWS IoT SiteWise Konsole oder fügen AWS CLI Sie Ihrem SiteWise Edge-MQTT-fähigen Gateway ein Echtzeitziel hinzu.

Console

1. Öffnen Sie die [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie das Gateway aus, zu dem Sie ein Ziel hinzufügen möchten.
4. Wählen Sie im Abschnitt Ziele die Option Ziel hinzufügen aus.
5. Geben Sie auf der Seite Ziel hinzufügen die Zieldetails ein:
 - a. Geben Sie im Feld Zielname einen Namen für Ihr Ziel ein.
 - b. Wählen Sie die AWS IoT SiteWise Echtzeitoption für den Zieltyp aus.
6. Konfigurieren Sie die Veröffentlichungsreihenfolge des Gateways, indem Sie die Veröffentlichungsreihenfolge auf „Ältere Daten zuerst veröffentlichen“ oder „Neueste Daten zuerst veröffentlichen“ festlegen. Standardmäßig veröffentlicht das Gateway die ältesten Daten zuerst.

7. Verwenden Sie Maximale Batch-Wartezeit, um eine maximale Wartezeit festzulegen, an die der Herausgeber warten soll, bevor er einen Datenstapel sendet AWS IoT SiteWise. Diese Einstellung gilt für jeden Alias. Die Daten werden lokal gespeichert, bis entweder:
 - Die eingestellte Zeit ist abgelaufen, oder
 - Es wurden 10 time-quality-value (TQV) Einträge für den Alias empfangen

Unabhängig davon, welche Bedingung zuerst erfüllt ist, wird der Batch an die Cloud gesendet.

8. Um hochgeladene Daten zu komprimieren, aktivieren Sie das Kontrollkästchen Komprimierung beim Hochladen von Daten aktivieren. Wenn Sie das Gateway Ihre Daten komprimieren lassen, bevor Sie sie in die Cloud hochladen, wird die Bandbreitennutzung reduziert.
9. Um abgelaufene Herausgeberdaten herauszufiltern, aktivieren Sie das Kontrollkästchen Abgelaufene Daten ausschließen. Mit dieser Auswahl werden nur aktive und aktuelle Daten an gesendet AWS IoT SiteWise.
10. Geben Sie im Feld Stichtag die Häufigkeit ein, mit der Daten innerhalb Ihres Datensatzes als abgelaufen gelten sollen. Sie können festlegen, ob die Daten in Minuten oder Tagen gezählt werden. Die minimale Sperrfrist beträgt fünf Minuten. Die maximale Ausschlussfrist beträgt sieben Tage.
11. Konfigurieren Sie optional die Einstellungen für den lokalen Speicher:
 - a. Legen Sie die Häufigkeit des Aufbewahrungszeitraums fest — Der Zeitraum, für den das Gateway Daten lokal speichert, die älter sind als der Sperrzeitraum. Die Mindestaufbewahrungsdauer beträgt eine Minute.

Die maximale Aufbewahrungsdauer beträgt 30 Tage und ist größer oder gleich der Rotationsdauer.

- b. Legen Sie den Rotationszeitraum fest — Das Zeitintervall, das beim Speichern von Daten angegeben werden soll, die älter sind als der Grenzzeitraum für eine einzelne Datei. Das Gateway überträgt am Ende jeder Rotationsperiode einen Datenstapel in das folgende lokale Verzeichnis:`/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports`.

Die Aufbewahrung muss länger als eine Minute sein und der Aufbewahrungsdauer entsprechen.

- c. Geben Sie den Wert für die Speicherkapazität (GB) an, um die maximale Größe der lokal gespeicherten Daten in GB festzulegen. Wenn die Daten die festgelegte maximale lokale Speichergröße überschreiten, beginnt das Gateway zuerst mit dem Löschen der ältesten Daten. Das Gateway löscht weiter, bis die Größe der lokal gespeicherten Daten dem Kontingent entspricht oder dieses unterschreitet.

Die Speicherkapazität muss größer oder gleich einem GB sein.

12. Fügen Sie Ihrem Ziel Pfadfilter hinzu. Weitere Informationen finden Sie unter [Fügen Sie Pfadfilter zu AWS IoT SiteWise Edge-Zielen hinzu](#).

Weitere Informationen finden Sie unter [Zieltypen](#).

AWS CLI

Example: Erstelle ein neues Ziel AWS IoT SiteWise in Echtzeit

Verwenden Sie die [UpdateGatewayCapabilityConfiguration](#) API, um den Herausgeber zu konfigurieren.

Stellen Sie den Parameter `capabilityNamespace` auf `iotsitewise:publisher:3` ein.

```
{
  "sources": [
    {
      "type": "MQTT"
    }
  ],
  "destinations": [
    {
      "type": "SITEWISE_REALTIME",
      "name": "your-destination-name",
      "config": {
        "publishingOrder": "TIME_ORDER",
        "enableCompression": true,
        "maxBatchWaitTime": "10s"
      },
      "filters": [
        {
          "type": "PATH",
          "config": {
            "paths": [
              "#"
            ]
          }
        }
      ]
    }
  ]
}
```

```

    ]
  }
}
]
}
]
}
}

```

Um ein vorhandenes AWS IoT SiteWise Echtzeitziel zu aktualisieren, verwenden Sie zunächst die `DescribeGatewayCapabilityConfiguration` API, um das zu findende `destinationId`.

Example: Aktualisieren Sie ein AWS IoT SiteWise Echtzeit-Ziel

Verwenden Sie die [UpdateGatewayCapabilityConfiguration](#) API, um den Herausgeber zu konfigurieren.

Stellen Sie den Parameter `capabilityNamespace` auf `iotsitewise:publisher:3` ein.

```

{
  "sources": [
    {
      "type": "MQTT"
    }
  ],
  "destinations": [
    {
      "id": "your-existing-destination-id",
      "type": "SITEWISE_REALTIME",
      "name": "your-destination-name",
      "config": {
        "publishingOrder": "TIME_ORDER",
        "enableCompression": true,
        "dropPolicy": {
          "cutoffAge": "7d",
          "exportPolicy": {
            "retentionPeriod": "7d",
            "rotationPeriod": "6h",
            "exportSizeLimitGB": 10
          }
        }
      },
      "maxBatchWaitTime": "10s"
    }
  ],
}

```

```

    "filters": [
      {
        "type": "PATH",
        "config": {
          "paths": [
            "#"
          ]
        }
      }
    ]
  }
]
}

```

Die folgenden Konfigurationsoptionen sind spezifisch für Gateways, die den `iotsitewise:publisher:3` Namespace verwenden.

`sources`

Definiert Datenquellen, zu denen Daten von Ihren Industrieanlagen übertragen werden sollen. AWS IoT SiteWise Verwenden Sie für MQTT-fähige Gateways. MQTT

Typ: Array von -Objekten

Erforderlich: Ja

`destinations`

Definiert, wohin Daten gesendet werden sollen. Ziele werden entweder in Echtzeit oder mit Amazon S3 gepuffert. Es ist mindestens ein Zielobjekt erforderlich, aber Sie können ein leeres Array hinzufügen. Sie können für jedes Gateway ein Echtzeitziel einrichten. Weitere Informationen finden Sie unter [Verstehen Sie AWS IoT SiteWise Edge-Ziele](#).

Typ: Array von -Objekten

Erforderlich: Ja

`id`

Die eindeutige Kennung für das Ziel. Sie können entweder eine vorhandene Ziel-ID angeben oder das Feld leer lassen. Wenn Sie keine ID angeben, wird standardmäßig eine UUID generiert.

Typ: Zeichenfolge

Erforderlich: Nein

type

Zieltyp Zu den Optionen gehören: SITEWISE_REALTIME und. SITEWISE_BUFFERED

- SITEWISE_REALTIME— Senden Sie Daten in Echtzeit direkt an den AWS IoT SiteWise Speicher.
- SITEWISE_BUFFERED— Senden Sie Daten stapelweise im Parquet-Format an Amazon S3 und importieren Sie sie dann in den AWS IoT SiteWise Speicher.

Typ: Zeichenfolge

Erforderlich: Ja

name

Ein eindeutiger Name für das Ziel.

Typ: Zeichenfolge

Erforderlich: Ja

config

Spezifische Konfiguration für den Zieltyp im JSON-Format. Die Konfiguration variiert zwischen Echtzeit- und gepufferten Zielen.

Typ: Objekt

Erforderlich: Ja

Reihenfolge der Veröffentlichung

Legt die Reihenfolge fest, in der Daten veröffentlicht werden. Daten werden auf der Grundlage ihres Zeitstempels veröffentlicht. Zu den Optionen gehören TIME_ORDER und. RECENT_DATA

- TIME_ORDER(Standard) — Veröffentlicht zuerst ältere Daten.
- RECENT_DATA— Publiziert die neuesten Daten zuerst.

Typ: Zeichenfolge

Erforderlich: Nein

Aktiviert die Komprimierung

Wenn auf `gesetztttrue`, wird die Datenkomprimierung vor dem Senden an aktiviert. AWS IoT SiteWise Wenn Sie das Gateway Ihre Daten komprimieren lassen, bevor Sie sie in die Cloud hochladen, wird die Bandbreitennutzung reduziert. Der Standardwert ist `true`.

Typ: Boolesch

Erforderlich: Nein

DropPolicy

Definiert, wie mit älteren Daten umgegangen werden soll.

Typ: Objekt

Erforderlich: Nein

- `cutoffAge`

Das maximale Alter der zu veröffentlichenden Daten, angegeben in Tagen, Stunden und Minuten. Zum Beispiel 7d oder 1d7h16m. Daten, die älter sind als die von Ihnen angegebenen, werden nicht gesendet AWS IoT SiteWise.

Daten, die vor dem Stichtag liegen, werden nicht in der Cloud veröffentlicht. Das Mindestalter muss zwischen fünf Minuten und sieben Tagen liegen.

Sie können, und verwenden `mh`, `d` wenn Sie ein Mindestalter angeben. Hinweis, `m` der Minuten, `h` Stunden und Tage `d` darstellt.

Typ: Zeichenfolge

Erforderlich: Ja

- `exportPolicy`

Definiert, wie mit Daten umgegangen wird, die das Mindestalter überschreiten.

Typ: Objekt

Erforderlich: Nein

- `retentionPeriod`

Ihr SiteWise Edge-Gateway löscht alle Daten am Edge, die vor dem Sperrzeitraum liegen, aus dem lokalen Speicher, nachdem sie für den angegebenen Aufbewahrungszeitraum gespeichert wurden. Die Aufbewahrungsdauer muss zwischen einer Minute und 30 Tagen liegen und mindestens dem Rotationszeitraum entsprechen.

Sie können `m`, und verwenden `h`, `d` wenn Sie einen Aufbewahrungszeitraum angeben. Hinweis, `m` der Minuten, `h` Stunden und Tage `d` darstellt.

Typ: Zeichenfolge

Erforderlich: Nein

- `rotationPeriod`

Das Zeitintervall, über das Daten, die vor dem Stichtag liegen, gebündelt und in einer einzigen Datei gespeichert werden sollen. Das SiteWise Edge-Gateway überträgt am Ende jeder Rotationsperiode einen Datenstapel in das folgende lokale Verzeichnis: `/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports`. Der Rotationszeitraum muss länger als eine Minute und gleich oder kürzer als der Aufbewahrungszeitraum sein.

Sie können `m`, und verwenden `h`, `d` wenn Sie einen Rotationszeitraum angeben. Hinweis, `m` der Minuten, `h` Stunden und Tage `d` darstellt.

Typ: Zeichenfolge

Erforderlich: Nein

- `exportSizeLimitGB`

Die maximal zulässige Größe der lokal gespeicherten Daten in GB. Wenn dieses Kontingent überschritten wird, beginnt das SiteWise Edge-Gateway mit dem Löschen der frühesten Daten, bis die Größe der lokal gespeicherten Daten dem Kontingent entspricht oder darunter liegt. Der Wert dieses Parameters muss größer oder gleich 1 sein.

Typ: Ganzzahl

Erforderlich: Nein

maxBatchWaitTime

Legt eine maximale Wartezeit für den Herausgeber fest, bevor er einen Datenstapel an sendet AWS IoT SiteWise. Diese Einstellung gilt für jeden Alias. Die Daten werden lokal gespeichert, bis entweder:

- Die eingestellte Zeit ist abgelaufen, oder
- Es wurden 10 time-quality-value (TQV) Einträge für den Alias empfangen

Verwenden Sie m, h, und, d um einen Annahmeschluss anzugeben. Hinweis, m der Minuten, h Stunden und Tage d darstellt.

Typ: Zeichenfolge

Erforderlich: Nein

filters

Filter, die auf die Daten angewendet werden sollen. Es ist mindestens ein Filter erforderlich.

Typ: Zeichenfolge

Erforderlich: Ja

type

Art des Filters. Verwenden Sie PATH.

Typ: Zeichenfolge

Erforderlich: Ja

config

Spezifische Konfiguration für den Filtertyp im JSON-Format. Es ist mindestens ein Objekt erforderlich, aber das Array kann leer sein.

Typ: Objekt

Erforderlich: Ja

- paths

Ein Array von Pfadfiltern. Weitere Informationen finden Sie unter [Machen Sie sich mit Pfadfiltern für AWS IoT SiteWise Edge-Ziele vertraut](#). Der Standardpfad ist #.

Typ: Zeichenfolgen-Array

Erforderlich: Ja

Fügen Sie ein AWS IoT SiteWise gepuffertes Ziel mit Amazon S3 hinzu

Der Zieltyp „Gepuffert“ ermöglicht es Ihnen, Aufnahmekosten zu sparen, AWS IoT SiteWise wenn Sie die Daten nicht in Echtzeit benötigen. Es ermöglicht Ihnen, Ihre IoT-Daten vorübergehend in einem Amazon S3 S3-Bucket zu speichern, bevor Sie sie importieren AWS IoT SiteWise. Oder Sie können Ihre Daten einfach zur Speicherung auf S3 hochladen, unabhängig davon, ob Sie sie importieren möchten AWS IoT SiteWise. Dies ist nützlich, um Daten von Ihren Geräten und Gateways zu stapeln und zu puffern, bevor Sie sie in sie aufnehmen. AWS IoT SiteWise Mit dieser Option werden Daten mit einer konfigurierten Frequenz im Parquet-Format in den angegebenen S3-Bucket hochgeladen. Sie können diese Daten dann zur weiteren Analyse und Verarbeitung in den AWS IoT SiteWise Speicher importieren.

Um ein mit Amazon S3 gepuffertes Ziel hinzuzufügen

Verwenden Sie die AWS IoT SiteWise Konsole oder fügen AWS CLI Sie Ihrem SiteWise Edge-Gateway ein Ziel hinzu, das Daten mithilfe von Amazon S3 zwischenspeichert.

Console

Verwenden Sie den AWS-Managementkonsole , um ein AWS IoT SiteWise Ziel hinzuzufügen, das mit Amazon S3 gepuffert wurde.

1. Öffnen Sie die [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie das Gateway aus, zu dem Sie ein Ziel hinzufügen möchten.
4. Wählen Sie im Abschnitt Ziele die Option Ziel hinzufügen aus.
5. Geben Sie auf der Seite Ziel hinzufügen die Zieldetails ein:
 - a. Geben Sie im Feld Zielname einen Namen für Ihr Ziel ein.
 - b. Wählen Sie unter Zieltyp die Option Mit Amazon S3 AWS IoT SiteWise gepuffert aus. AWS IoT SiteWise gepuffert mit Amazon S3 sendet Daten stapelweise im Parquet-Format an Amazon Simple Storage Service und importiert die Daten anschließend in AWS IoT SiteWise den Speicher.

6. Geben Sie die Amazon S3 S3-URL für den Standort ein, an dem Sie Ihre Gateway-Daten speichern möchten. Sie können nach dem Pfad suchen, indem Sie Browse S3 wählen. Sobald ein Bucket hinzugefügt wurde, können Sie den Bucket auch anzeigen, indem Sie View wählen.
7. Geben Sie an, wie oft Ihr Gateway Daten auf Amazon S3 hochladen soll, indem Sie einen Zeitrahmen eingeben und eine Zeitspanne für die Häufigkeit des Daten-Uploads auswählen. Der Frequenzwert sollte größer als 0 und kleiner als oder gleich 30 Tagen sein.
8. Legen Sie in den Datenspeichereinstellungen fest, was mit Ihren Gateway-Daten geschehen soll, nachdem Sie sie importiert haben AWS IoT SiteWise. In Bezug auf die Datenspeicherung müssen zwei Entscheidungen getroffen werden:
 - Wenn Sie importierte Daten in den AWS IoT SiteWise Speicher kopieren möchten, aktivieren Sie das Kontrollkästchen Daten in den Speicher kopieren. Diese Option dupliziert die importierten Daten aus Ihrem konfigurierten Amazon S3 S3-Bucket in den AWS IoT SiteWise Speicher.
 - Wenn Sie Ihre Daten aus Ihrem Amazon S3 S3-Bucket in den AWS IoT SiteWise Speicher importieren möchten, können Sie auch angeben, ob die importierten Daten nach Abschluss des Imports gelöscht werden sollen. Aktivieren Sie das Kontrollkästchen Daten aus Amazon S3 löschen, um das importierte Datum nach dem Import in den AWS IoT SiteWise Speicher aus dem konfigurierten Amazon S3 S3-Bucket zu löschen.
9. Fügen Sie Ihrem Ziel Pfadfilter hinzu. Weitere Informationen finden Sie unter [Fügen Sie Pfadfilter zu AWS IoT SiteWise Edge-Zielen hinzu](#).

AWS CLI

Example: Erstellen Sie ein neues AWS IoT SiteWise Ziel, das mit Amazon S3 gepuffert wird

Verwenden Sie die [UpdateGatewayCapabilityConfiguration](#)API, um den Herausgeber zu konfigurieren.

Stellen Sie den Parameter `capabilityNamespace` auf `iotsitewise:publisher:3` ein.

```
{
  "sources": [
    {
      "type": "MQTT"
    }
  ],

```

```

"destinations": [
  {
    "type": "SITEWISE_BUFFERED",
    "name": "your-s3-destination-name",
    "config": {
      "targetBucketArn": "arn:aws:s3:::amzn-s3-demo-bucket/Optional/SomeFolder",
      "publishPolicy": {
        "publishFrequency": "15m",
        "localSizeLimitGB": 10
      },
      "siteWiseImportPolicy": {
        "enableSiteWiseStorageImport": true,
        "enableDeleteAfterImport": true,
        "bulkImportJobRoleArn": "arn:aws:iam::123456789012:role/your-role-name"
      }
    },
    "filters": [
      {
        "type": "PATH",
        "config": {
          "paths": [
            "#"
          ]
        }
      }
    ]
  }
]
}

```

Example: Ein mit Amazon AWS IoT SiteWise S3 gepuffertes Ziel aktualisieren

Um ein vorhandenes AWS IoT SiteWise Echtzeit-Ziel zu aktualisieren, verwenden Sie zunächst die `DescribeGatewayCapabilityConfiguration` API, um das `destinationId` zu finden.

Der Publisher-Namespace: `iotsitewise:publisher:3`

```

{
  "sources": [
    {
      "type": "MQTT"
    }
  ]
}

```

```

],
"destinations": [
  {
    "id": "your-existing-destination-id",
    "type": "SITEWISE_BUFFERED",
    "name": "your-s3-destination-name",
    "config": {
      "targetBucketArn": "arn:aws:s3:::amzn-s3-demo-bucket/Optional/SomeFolder",
      "publishPolicy": {
        "publishFrequency": "15m",
        "localSizeLimitGB": 10
      },
      "siteWiseImportPolicy": {
        "enableSiteWiseStorageImport": true,
        "enableDeleteAfterImport": true,
        "bulkImportJobRoleArn": "arn:aws:iam::123456789012:role/your-role-name"
      }
    },
    "filters": [
      {
        "type": "PATH",
        "config": {
          "paths": [
            "#"
          ]
        }
      }
    ]
  }
]
}

```

Die folgenden Konfigurationsoptionen sind spezifisch für MQTT-fähige Gateways, die den Namespace verwenden. `iotsitewise:publisher:3`

sources

Definiert Datenquellen für die Übertragung von Daten von Ihren Industrieanlagen zu AWS IoT SiteWise. Verwenden Sie für MQTT-fähige Gateways. MQTT

Typ: Array von -Objekten

Erforderlich: Ja

destinations

Definiert, wohin Daten gesendet werden sollen. Ziele werden entweder in Echtzeit oder mit Amazon S3 gepuffert. Es ist mindestens ein Zielobjekt erforderlich, aber Sie können ein leeres Array hinzufügen. Sie können für jedes Gateway ein Echtzeitziel einrichten. Weitere Informationen finden Sie unter [Verstehen Sie AWS IoT SiteWise Edge-Ziele](#).

Typ: Array von -Objekten

Erforderlich: Ja

id

Die eindeutige Kennung für das Ziel. Sie können entweder eine bestehende Ziel-ID angeben oder das Feld leer lassen, damit automatisch eine neue ID für das Ziel generiert wird.

Typ: Zeichenfolge

Erforderlich: Nein

type

Zieltyp Zu den Optionen gehören: SITEWISE_REALTIME und SITEWISE_BUFFERED. Wählen Sie SITEWISE_BUFFERED.

- SITEWISE_REALTIME(Standard) — Daten werden in Echtzeit direkt an den AWS IoT SiteWise Speicher gesendet. Weitere Informationen finden Sie unter [Fügen Sie ein AWS IoT SiteWise Edge-Echtzeitziel hinzu](#).
- SITEWISE_BUFFERED— Senden Sie Daten stapelweise im Parquet-Format an Amazon S3 und importieren Sie sie dann in den AWS IoT SiteWise Speicher.

Typ: Zeichenfolge

Erforderlich: Ja

name

Ein eindeutiger Name für das Ziel.

Typ: Zeichenfolge

Erforderlich: Ja

config

Spezifische Konfiguration für den Zieltyp im JSON-Format. Die Konfiguration variiert zwischen Echtzeit- und gepufferten Zielen.

Typ: Objekt

Erforderlich: Ja

targetBucketArn

Der BucketARN, in dem veröffentlicht werden soll. Wählen Sie dasselbe AWS-Region für beide AWS IoT SiteWise und Amazon S3. Wenn ein Präfix ausgewählt wird, muss es zwischen 1 und 255 Zeichen lang sein.

Note

AWS IoT SiteWise, einschließlich des Gateways, wird Zugriff auf den gesamten angegebenen S3-Bucket haben. Wir empfehlen die Verwendung eines speziellen Buckets für die gepufferte Datenaufnahme.

Typ: Zeichenfolge

Erforderlich: Ja

publishPolicy

Einzelheiten der Veröffentlichungsrichtlinie.

Typ: Objekt

Erforderlich: Ja

publishFrequency

Die Häufigkeit, mit der das SiteWise Edge-Gateway im Amazon S3 S3-Bucket veröffentlicht. Die Häufigkeit, mit der Daten auf Amazon S3 hochgeladen werden, muss mehr als 0 Minuten und weniger als 30 Tage oder weniger als 30 Tage betragen. Sie können, und verwenden mh, d wenn Sie ein Alter für die Veröffentlichungshäufigkeit angeben. Hinweis, m der Minuten, h Stunden und Tage d darstellt. Der Standardwert ist 15 Minuten.

Typ: Zeichenfolge

Erforderlich: Ja

`localSizeLimitGB`

Die maximale Größe der auf die lokale Festplatte geschriebenen Dateien in GB. Wenn dieser Schwellenwert überschritten wird, veröffentlicht der Herausgeber alle gepufferten Daten an seinem Ziel.

Typ: Ganzzahl

Erforderlich: Ja

`siteWiseImportPolicy`

Einzelheiten der Importrichtlinie für den Import von Daten in AWS IoT SiteWise

Typ: Objekt

Erforderlich: Ja

`enableSiteWiseStorageImport`

Stellen Sie dies auf `true`, um Daten aus einem Amazon S3 S3-Bucket in den AWS IoT SiteWise Speicher zu importieren. Es erstellt zunächst eine Kopie der Daten in AWS IoT SiteWise. Wenn Sie dann `enableDeleteAfterImport` auf `true` setzen, werden die Daten in S3 nach dem Kopieren nach AWS IoT SiteWise gelöscht. Dies hat Auswirkungen auf die Preisgestaltung. Der Standardwert ist `true`.

Typ: Boolescher Wert

Erforderlich: Ja

`enableDeleteAfterImport`

Stellen Sie diese Option auf `true`, um die Datei im Amazon S3 S3-Bucket nach der Aufnahme in den AWS IoT SiteWise Speicher zu löschen. Der Standardwert ist `true`.

Typ: Boolescher Wert

Erforderlich: Ja

bulkImportJobRoleArn

Der ARN der IAM-Rolle, die AWS IoT SiteWise davon ausgeht, während der Datenaufnahme gepufferte Daten aus Amazon S3 zu lesen. Diese Rolle wird verwendet, wenn ein Edge-Gerät anruft, um den AWS IoT SiteWise APIs Massenimportprozess einzuleiten.



Note

Wenn auf gesetzt `enableSiteWiseStorageImport` ist `true`, ist dieser Parameter erforderlich.

Typ: Zeichenfolge

Erforderlich: Nein

Fügen Sie Pfadfilter für Ihr Ziel hinzu. Weitere Informationen finden Sie unter [Fügen Sie Pfadfilter zu AWS IoT SiteWise Edge-Zielen hinzu](#).

Fügen Sie Pfadfilter zu AWS IoT SiteWise Edge-Zielen hinzu

Fügen Sie Pfadfilter zu einem Ziel hinzu. Pfadfilter verwenden die MQTT-Themensyntax, wobei # es sich um ein Platzhalterzeichen handelt, das einer beliebigen Anzahl von Ebenen entspricht, und + um ein Platzhalterzeichen, das einer einzelnen Ebene entspricht. Sie können einem Gateway mehrere Ziele hinzufügen, von denen jedes über einen eigenen Satz von Pfadfiltern verfügt, die Ihre Gerätelemetrie abonnieren.

Siemens Industrial EdgeGateways verwenden aus Kompatibilitätsgründen ein Präfix. Weitere Informationen finden Sie unter [Präfixe für Pfadfilter](#).

Console

Um Pfadfilter hinzuzufügen

1. Öffnen Sie die [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie das Gateway aus, dem Sie Pfadfilter hinzufügen möchten.

4. Wählen Sie im Abschnitt Pfadfilter unter Ziel hinzufügen die Option Pfadfilter hinzufügen aus.
5. Geben Sie den Pfadfilter ein, den dieses Ziel abonnieren soll. Sie können Platzhalterzeichen (#und+) verwenden, um mehrere Pfade zu abonnieren.
6. Wählen Sie Pfadfilter hinzufügen, um den Pfadfilter zur Liste hinzuzufügen.
7. Wiederholen Sie die Schritte, um bei Bedarf weitere Pfadfilter hinzuzufügen.
8. Nachdem Sie alle erforderlichen Pfadfilter hinzugefügt haben, wählen Sie Erstellen aus.

AWS CLI for self-hosted gateways

Example: Konfiguration des Pfadfilters

```
{
  "destinations": [
    {
      ...
    }
  ],
  "filters": [
    {
      "type": "PATH",
      "config": {
        "paths": [
          "home/+/sensor1/temperature",
          "home/livingroom/sensor1/temperature",
          "home/livingroom/sensor1/temperature",
          "building/#"
        ]
      }
    }
  ]
}
```

AWS CLI for Siemens IEGateways

Example: Präfixkonfiguration für Pfadfilter

Erfassen Sie alle Daten, indem Sie sowohl die Präfixe data (ie/d) als auch die metadata (ie/m) - Präfixe für jeden Pfadfilter verwenden.

```
{
  "destinations": [
    {
      ...
    }
  ],
  "filters": [
    {
      "type": "PATH",
      "config": {
        "paths": [
          "ie/d/home/+/sensor12/temperature",
          "ie/m/home/livingroom/sensor12/temperature",
          "ie/d/home/livingroom/sensor13/temperature2",
          "ie/m/home/livingroom/sensor13/temperature2",
          "ie/d/building/#",
          "ie/m/building/#"
        ]
      }
    }
  ]
}
```

Note

Kopieren Sie Pfadfilter zwischen Zielen, indem Sie eine Liste der Pfadfilter herunterladen. Weitere Informationen finden Sie unter [Laden Sie alle Pfadfilter in einem Ziel \(Konsole\) herunter](#).

Laden Sie Pfadfilter in großen Mengen hoch

Verwenden Sie eine CSV- oder Textdatei, um Pfadfilter in großen Mengen hochzuladen. AWS IoT SiteWise entfernt automatisch exakte Duplikate, wenn Sie Dateien hochladen. Zum Beispiel `windfarm/site1/` sind es exakte Duplikate, die AWS IoT SiteWise auffallen, weil die Zeichenfolge exakt dieselbe ist. `windfarm/site1/` Teilweise Duplikate werden nicht entfernt und führen zu zusätzlichen Gebühren. Zum Beispiel, `windfarm/#` und überschneiden `windfarm/site1` sich Themen, weil dies bereits von umfasst `windfarm/site1` ist. `windfarm/#`

 Note

Vermeiden Sie Duplikate, um zusätzliche Gebühren zu vermeiden. Die hochgeladene Datei muss entweder im CSV- oder im TXT-Format vorliegen. Sie darf keine Überschriften enthalten und sollte aus einer einzigen Spalte bestehen. In der Spalte listen Sie Ihre Pfadfilter auf, wobei sich jeder Filter in einer separaten Zeile befindet. Die Datei sollte keine weiteren Informationen enthalten.

Anforderungen für das Hochladen von Dateien

Dies sind zusätzliche Anforderungen an den Pfadfilter.

- Sie können eine CSV- oder TXT-Datei hochladen. Andere Dateiformate werden nicht unterstützt.
- CSV-Dateien (.csv) können keine Kopfzeilen haben und sollten nur eine Spalte enthalten.
- Sie können in jeder Zeile einen Pfadfilter verwenden.
- Die hochgeladenen Dateien dürfen nicht leer sein.
- Bei Verwendung von # als Platzhalter muss dies das letzte Zeichen im Themenfilter sein. Zum Beispiel `topic/#` oder als eigenständiges Zeichen auf einer bestimmten Themenebene. Beachten Sie jedoch, dass # dies auch als reguläres Zeichen innerhalb eines Namens einer Themenebene verwenden kann, z. B. `factory/machine#1/topic`. Weitere Informationen finden Sie unter [Sonderzeichen in Pfadfilternamen](#)
- Sie können den + Charakter auch verwenden. Verwenden Sie zum Beispiel, `factory/+/temp` um alle Temperaturen für Fabriken anstelle von `factory/machine2/temp` und `factory/machine3/temp` einzeln abzurufen.

AWS IoT SiteWise Edge-Ziele verwalten

Nachdem Sie Ziele hinzugefügt haben, können Sie sie mit verschiedenen Vorgängen verwalten, z. B. Zielkonfigurationen bearbeiten, Ziele löschen und Pfadfilter verwalten.

Bearbeiten Sie ein Ziel

Wählen Sie das Optionsfeld neben dem Ziel in der Tabelle und klicken Sie auf die Schaltfläche Bearbeiten, um ein Ziel zu bearbeiten.

Console

Um ein Ziel mit der AWS IoT SiteWise Konsole zu bearbeiten

1. Öffnen Sie die [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie in der linken Navigationsleiste im Abschnitt Edge die Option Edge-Gateways aus.
3. Wählen Sie das entsprechende Gateway aus.
4. Wählen Sie im Abschnitt Ziele das Ziel aus, das Sie bearbeiten möchten, und klicken Sie dann auf Bearbeiten.
5. Ändern Sie das Ziel und wählen Sie dann Speichern.

AWS CLI

Um ein Ziel zu bearbeiten, verwenden Sie AWS CLI

- Sie können ein Ziel bearbeiten, indem Sie die Informationen zur Konfiguration der JSON-Fähigkeit ändern.

```
aws iotsitewise update-gateway-capability-configuration \
--gateway-id your-gateway-id \
--capability-namespace "iotsitewise:publisher:3" \
--capability-configuration '{
  "sources": [
    {
      "type": "MQTT"
    }
  ],
  "destinations": [
    {
      "id": "your-existing-destination-id",
      "type": "SITEWISE_REALTIME",
      "name": "your-updated-destination-name",
      "config": {
        "publishingOrder": "TIME_ORDER",
        "enableCompression": true,
        "dropPolicy": {
          "cutoffAge": "10d",
          "exportPolicy": {
            "retentionPeriod": "10d",
            "rotationPeriod": "6h",
```

```
        "exportSizeLimitGB": 10
      },
    },
    "maxBatchWaitTime": "15s"
  },
  "filters": [
    {
      ...
    }
  ]
}
]
```

Note

Sie können das Ziel nicht aktualisieren `type` oder `capability-namespace`. Sie können beispielsweise nicht von einem Typ von `SITewise_REALTIME` zu `wechsellnSITewise_BUFFERED`. Sie können für jedes MQTT-fähige Gateway ein Echtzeitziel einrichten.

Löschen Sie ein Ziel

Wenn Sie ein Ziel nicht mehr benötigen, können Sie es von Ihrem SiteWise Edge-Gateway löschen.

Console

Um ein Ziel mit der AWS IoT SiteWise Konsole zu löschen

1. Öffnen Sie die [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie in der linken Navigationsleiste im Abschnitt Edge die Option Edge-Gateways aus.
3. Wählen Sie das entsprechende Gateway aus.
4. Wählen Sie im Abschnitt Ziele das Ziel aus, das Sie löschen möchten, und klicken Sie dann auf Löschen. Ein Bestätigungsbildschirm wird angezeigt.
5. Um zu bestätigen, dass Sie das Ziel löschen möchten, geben Sie „Löschen“ in das Bestätigungsfeld ein.

AWS CLI

Um ein Ziel zu löschen, verwenden Sie AWS CLI

- Löschen Sie die Gateway-Fähigkeitskonfiguration, indem Sie die Gateway-ID angeben und die Funktionskonfiguration so ändern, dass das Ziel, das Sie löschen möchten, entfernt wird.

```
aws iotsitewise update-gateway-capability-configuration \  
  --gateway-id your-gateway-id \  
  --capability-namespace "iotsitewise:publisher:3" \  
  --capability-configuration '{  
    "sources": [  
      {  
        "type": "MQTT"  
      }  
    ],  
    "destinations": []  
  }'
```

Note

Das Zielarray kann leer sein ([]), aber das Zielobjekt selbst muss in der Funktionskonfiguration enthalten sein.

Laden Sie alle Pfadfilter in einem Ziel (Konsole) herunter

Laden Sie eine CSV-Datei mit all Ihren Pfadfiltern in der AWS IoT SiteWise Konsole herunter. Sie können eine heruntergeladene Liste von Pfadfiltern verwenden, um Pfadfilterlisten einfach zwischen Gateway-Zielen auszutauschen.

Um eine CSV-Datei mit allen Pfadfiltern mithilfe der AWS IoT SiteWise Konsole herunterzuladen

1. Öffnen Sie die [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie in der linken Navigationsleiste im Abschnitt Edge die Option Edge-Gateways aus.
3. Wählen Sie das Gateway aus, das Ihre Pfadfilter enthält.
4. Wählen Sie entweder Ziel hinzufügen oder Ziel bearbeiten.
5. Navigieren Sie zum Abschnitt Pfadfilter und wählen Sie CSV herunterladen aus.

 Note

Die CSV-Datei enthält alle Pfadfilter an einem bestimmten Ziel, unabhängig davon, welche Sie aus der Liste der Pfadfilter ausgewählt haben.

Bearbeiten Sie einen Pfadfilter

Sie können einzelne Pfadfilter bearbeiten, um zu verfeinern, welche Daten Ihr Ziel empfängt.

Console

Mithilfe der AWS IoT SiteWise Konsole können Sie jeden einzelnen Pfadfilter in den jeweiligen Textfeldern bearbeiten.

Um einen Pfadfilter mit der AWS IoT SiteWise Konsole zu bearbeiten

1. Öffnen Sie die [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie in der linken Navigationsleiste im Abschnitt Edge die Option Edge-Gateways aus.
3. Wählen Sie das Gateway aus, das Ihre Pfadfilter enthält.
4. Wählen Sie das entsprechende Ziel aus.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie das Textfeld für die Zeile aus, die den Pfadfilter enthält, den Sie bearbeiten möchten.
7. Aktualisieren Sie den Text des Pfadfilters und stellen Sie sicher, dass das Kontrollkästchen des bearbeiteten Pfadfilters aktiviert ist.
8. Wählen Sie Speichern.

AWS CLI

Um Pfadfilter für ein Ziel mithilfe von zu bearbeiten AWS CLI, rufen Sie zuerst die aktuelle Konfiguration ab, ändern Sie sie und aktualisieren Sie sie dann mit dem `update-gateway-capability-configuration` Befehl.

Um einen Pfadfilter zu bearbeiten, verwenden Sie AWS CLI

1. Rufen Sie die aktuelle Funktionskonfiguration ab:

```
aws iotsitewise describe-gateway-capability-configuration \
  --gateway-id your-gateway-id \
  --capability-namespace "iotsitewise:publisher:3" \
  --query "capabilityConfiguration"
```

2. Bearbeiten Sie den JSON-Code, um die Pfadfilter nach Bedarf zu ändern.
3. Aktualisieren Sie die Funktionskonfiguration mit den geänderten Pfadfiltern:

```
aws iotsitewise update-gateway-capability-configuration \
  --gateway-id your-gateway-id \
  --capability-namespace "iotsitewise:publisher:3" \
  --capability-configuration json-containing-your-updated-path-filters
```

Löschen Sie einen Pfadfilter

Sie können Pfadfilter für ein Ziel löschen, um zu kontrollieren, welche Daten es von MQTT-Quellen und Datenverarbeitungspipelines empfängt.

Console

Um einen Pfadfilter mit der Konsole zu löschen AWS IoT SiteWise

1. Öffnen Sie die [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie in der linken Navigationsleiste im Abschnitt Edge die Option Edge-Gateways aus.
3. Wählen Sie das Gateway aus, das Ihre Pfadfilter enthält.
4. Wählen Sie das entsprechende Ziel aus.
5. Wählen Sie Bearbeiten aus.
6. Wählen Sie auf dem Bildschirm Ziel bearbeiten im Abschnitt Pfadfilter einen oder mehrere Pfadfilter aus, die Sie löschen möchten.
7. Wählen Sie Löschen aus. Eine Bestätigungsnachricht für den Löschvorgang wird angezeigt. Wenn Sie mit dem Löschen der Pfadfilter fortfahren möchten, wählen Sie auf dem Bestätigungsbildschirm Löschen aus.

AWS CLI

Um ein Ziel zu löschen, verwenden Sie AWS CLI

- Löschen Sie einen Pfadfilter, indem Sie ihn aus der Funktionskonfiguration entfernen.

```
aws iotsitewise update-gateway-capability-configuration \
  --gateway-id your-gateway-id \
  --capability-namespace "iotsitewise:publisher:3" \
  --capability-configuration '{
    "sources": [
      {
        "type": "MQTT"
      }
    ],
    "destinations": [
      {
        "id": "your-destination-id",
        "type": "SITEWISE_REALTIME",
        "name": "your-destination-name",
        "config": {
          ...
        },
        "filters": [
          {
            "type": "PATH",
            "config": {
              "paths": [
                "/path1",
                "/path2",
                "/delete-a-path-to-remove-it"
              ]
            }
          }
        ]
      }
    ]
  }
```

 Note

Das Filter-Array kann leer sein ([]), aber das Filterobjekt selbst muss in der Funktionskonfiguration enthalten sein.

SiteWise Edge-Gateways verwalten

Sie können die AWS IoT SiteWise Konsolen- und API-Operationen verwenden, um AWS IoT SiteWise Edge-Gateways zu verwalten. Sie können auch die Anwendung [AWS OpsHub for AWS IoT SiteWise for Windows](#) verwenden, um einige Aspekte Ihres SiteWise Edge-Gateways von Ihrem lokalen Gerät aus zu verwalten.

Wir empfehlen Ihnen dringend, die AWS IoT SiteWise Anwendung AWS OpsHub for zu verwenden, um die Festplattennutzung auf Ihrem lokalen Gerät zu überwachen. Sie können auch die `Gateway.AvailableDiskSpace` und `Gateway.UsedPercentageDiskSpace` CloudWatch Amazon-Metriken überwachen und Alarmer erstellen, um benachrichtigt zu werden, wenn der Festplattenspeicher knapp wird. Weitere Informationen zu CloudWatch Amazon-Alarmen finden Sie unter [Erstellen eines CloudWatch Alarms auf der Grundlage eines statischen Schwellenwerts](#).

Stellen Sie sicher, dass Ihr Gerät über ausreichend Speicherplatz für zukünftige Daten verfügt. Wenn der Speicherplatz auf Ihrem lokalen Gerät knapp wird, löscht der Dienst automatisch eine kleine Datenmenge mit den ältesten Zeitstempeln, um Platz für zukünftige Daten zu schaffen.

Gehen Sie wie folgt vor, um zu überprüfen, ob der Dienst Ihre Daten gelöscht hat:

1. Melden Sie sich AWS OpsHub bei der AWS IoT SiteWise Anwendung an.
2. Wählen Sie Einstellungen aus.
3. Geben Sie für Protokolle einen Zeitraum an, und wählen Sie dann Herunterladen aus.
4. Entpacken Sie die Protokolldatei.
5. Wenn die Protokolldatei die folgende Meldung enthält, hat der Dienst Ihre Daten gelöscht:
numberDatenbytes wurden gelöscht, um zu verhindern, dass dem SiteWise Edge-Gateway-Speicher der Speicherplatz ausgeht.

Verwalten Sie Ihr SiteWise Edge-Gateway mit der AWS IoT SiteWise Konsole

Sie können die AWS IoT SiteWise Konsole verwenden, um alle SiteWise Edge-Gateways in Ihrem AWS Konto zu konfigurieren, zu aktualisieren und zu überwachen.

[Sie können Ihre SiteWise Edge-Gateways anzeigen, indem Sie in der Konsole zur Edge-Gateways-Seite navigieren.](#)[AWS IoT SiteWise](#) Um auf die Edge-Gateway-Detailseite für ein bestimmtes Gateway zuzugreifen, wählen Sie den Namen eines Edge-Gateways aus.

Auf der Registerkarte „Übersicht“ der Edge-Gateway-Detailseite können Sie Folgendes tun:

- Aktualisieren Sie im Abschnitt Datenquellen die Datenquellenkonfiguration und konfigurieren Sie zusätzliche Datenquellen
- Wählen Sie CloudWatch Metriken öffnen aus, um die Anzahl der pro Datenquelle aufgenommenen Datenpunkte in der CloudWatch Metrikkonsole anzuzeigen
- Fügen Sie im Abschnitt Edge-Funktionen Datenpakete zu Ihrem SiteWise Edge-Gateway hinzu, indem Sie auf Bearbeiten klicken
- Sehen Sie sich im Abschnitt Gateway-Konfiguration den Konnektivitätsstatus Ihrer SiteWise Edge-Gateways an
- Sehen Sie sich im Abschnitt Publisher-Konfiguration den Synchronisierungsstatus und die Konfiguration der AWS IoT SiteWise Publisher-Komponente des SiteWise Edge-Gateways an

Auf der Registerkarte Updates der Edge-Gateway-Detailseite können Sie die aktuellen Komponenten- und Paketversionen sehen, die auf dem Edge-Gateway bereitgestellt werden. Hier stellen Sie auch neue Versionen bereit, sobald sie verfügbar sind.

Verwalten Sie SiteWise Edge-Gateways mit AWS OpsHub AWS IoT SiteWise

Sie verwenden die AWS IoT SiteWise Anwendung AWS OpsHub for, um Ihre selbst gehosteten SiteWise Edge-Gateways zu verwalten und zu überwachen. Diese Anwendung bietet die folgenden Überwachungs- und Verwaltungsoptionen:

- Unter Überblick können Sie Folgendes tun:

- Sehen Sie sich SiteWise Edge-Gateway-Details an, die Ihnen helfen, Einblicke in Ihre SiteWise Edge-Gateway-Gerätedaten zu erhalten, Probleme zu identifizieren und die Leistung des SiteWise Edge-Gateways zu verbessern.
- Sehen Sie sich SiteWise Monitor-Portale an, die die Daten von lokalen Servern und Geräten am Edge überwachen. Weitere Informationen finden Sie unter [Was ist AWS IoT SiteWise Monitor](#) im AWS IoT SiteWise Monitor Anwendungshandbuch enthalten.
- Unter Health gibt es ein Dashboard, das Daten von Ihrem SiteWise Edge-Gateway anzeigt. Fachexperten, z. B. Verfahrenstechniker, können das Dashboard verwenden, um sich einen Überblick über das Verhalten des SiteWise Edge-Gateways zu verschaffen.
- Sehen Sie sich unter Assets die auf dem lokalen Gerät bereitgestellten Ressourcen und den zuletzt erfassten oder berechneten Wert für Asset-Eigenschaften an.
- Unter Einstellungen können Sie Folgendes tun:
 - Wenn das Data Processing Pack installiert ist, sehen Sie sich die Informationen zur SiteWise Edge-Gateway-Konfiguration an und synchronisieren Sie die Ressourcen mit der AWS Cloud.
 - Laden Sie die Authentifizierungsdateien herunter, mit denen Sie mithilfe anderer Tools auf das SiteWise Edge-Gateway zugreifen können.
 - Laden Sie Protokolle herunter, die Sie zur Fehlerbehebung am SiteWise Edge-Gateway verwenden können.
 - Sehen Sie sich die auf dem SiteWise Edge-Gateway bereitgestellten AWS IoT SiteWise Komponenten an.

 Important

Folgendes ist AWS OpsHub für die Verwendung erforderlich AWS IoT SiteWise:

- Ihr lokales Gerät und AWS OpsHub die AWS IoT SiteWise FOR-Anwendung müssen mit demselben Netzwerk verbunden sein.
- Das Datenverarbeitungspaket muss aktiviert sein.

Um SiteWise Edge-Gateways zu verwalten mit AWS OpsHub

1. Laden Sie die Anwendung [AWS OpsHubAWS IoT SiteWise für Windows](#) herunter und installieren Sie sie.
2. Öffnen Sie die Anwendung .

3. Wenn Sie keine lokalen Anmeldeinformationen für Ihr Gateway eingerichtet haben, folgen Sie den Schritten unter, [Greifen Sie mit lokalen Betriebssystemanmeldedaten auf Ihr SiteWise Edge-Gateway zu](#) um sie einzurichten.
4. Sie können sich mit Ihren Linux- oder LDAP-Anmeldeinformationen (Lightweight Directory Access Protocol) bei Ihrem SiteWise Edge-Gateway anmelden. Gehen Sie wie folgt vor, um sich bei Ihrem SiteWise Edge-Gateway anzumelden:

Linux

1. Geben Sie für Hostname oder IP-Adresse den Hostnamen oder die IP-Adresse Ihres lokalen Geräts ein.
2. Wählen Sie für Authentifizierung Linux.
3. Geben Sie unter Benutzername den Benutzernamen Ihres Linux-Betriebssystems ein.
4. Geben Sie unter Passwort das Passwort Ihres Linux-Betriebssystems ein.
5. Klicken Sie auf Sign in.

LDAP

1. Geben Sie für Hostname oder IP-Adresse den Hostnamen oder die IP-Adresse Ihres lokalen Geräts ein.
2. Wählen Sie für Authentifizierung LDAP.
3. Geben Sie als Benutzername den Benutzernamen Ihres LDAP ein.
4. Geben Sie unter Passwort das Passwort Ihres LDAP ein.
5. Klicken Sie auf Sign in.

Greifen Sie mit lokalen Betriebssystemanmeldedaten auf Ihr SiteWise Edge-Gateway zu

Neben dem Lightweight Directory Access Protocol (LDAP) können Sie die Linux- oder Windows-Anmeldeinformationen verwenden, um auf Ihr selbst gehostetes SiteWise Edge-Gateway zuzugreifen.

 Important

Um mit Linux-Anmeldeinformationen auf Ihr SiteWise Edge-Gateway zuzugreifen, müssen Sie das Datenverarbeitungspaket für Ihr SiteWise Edge-Gateway aktivieren.

Greifen Sie mit den Anmeldeinformationen für das Linux-Betriebssystem auf Ihr SiteWise Edge-Gateway zu

Bei den folgenden Schritten wird davon ausgegangen, dass Sie ein Gerät mit Ubuntu verwenden. Wenn Sie eine andere Linux-Distribution verwenden, schlagen Sie in der entsprechenden Dokumentation für Ihr Gerät nach.

Um einen Linux-Benutzerpool zu erstellen

1. Führen Sie den folgenden Befehl aus, um eine Admin-Gruppe zu erstellen.

```
sudo groupadd --system SWE_ADMIN_GROUP
```

Benutzer in der SWE_ADMIN_GROUP Gruppe können Administratorzugriff für das SiteWise Edge-Gateway gewähren.

2. Führen Sie den folgenden Befehl aus, um eine Benutzergruppe zu erstellen.

```
sudo groupadd --system SWE_USER_GROUP
```

Benutzer in der SWE_USER_GROUP Gruppe können schreibgeschützten Zugriff für das SiteWise Edge-Gateway gewähren.

3. Führen Sie den folgenden Befehl aus, um der Admin-Gruppe einen Benutzer hinzuzufügen. Ersetzen Sie *user-name* und *password* durch den Benutzernamen und das Passwort, die Sie hinzufügen möchten.

```
sudo useradd -p $(openssl passwd -1 password) user-name
```

4. Um einen Benutzer zu einem SWE_ADMIN_GROUP oder hinzuzufügen SWE_USER_GROUP, *user-name* ersetzen Sie ihn durch den Benutzernamen, den Sie im vorherigen Schritt hinzugefügt haben.

```
sudo usermod -a -G SWE_ADMIN_GROUP user-name
```

Sie können jetzt den Benutzernamen und das Passwort verwenden, um sich in der AWS IoT SiteWise FOR-Anwendung am SiteWise AWS OpsHub Edge-Gateway anzumelden.

Greifen Sie mit Windows-Anmeldeinformationen auf Ihr SiteWise Edge-Gateway zu

Bei den folgenden Schritten wird davon ausgegangen, dass Sie ein Gerät mit Windows verwenden.

 Important

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Erstellen Sie eine sichere Passworrichtlinie mit mindestens 12 Zeichen und einer Kombination aus Groß- und Kleinbuchstaben, Zahlen und Symbolen. Stellen Sie außerdem die Windows-Firewallregeln so ein, dass eingehender Verkehr an Port 443 zugelassen und eingehender Verkehr an allen anderen Ports blockiert wird.

Um einen Windows Server-Benutzerpool zu erstellen

1. Führen Sie PowerShell es als Administrator aus.
 - a. Melden Sie sich auf dem Windows-Server, auf dem Sie SiteWise Edge Gateway installieren möchten, als Administrator an.
 - b. Geben Sie PowerShell in die Windows-Suchleiste ein.
 - c. Klicken Sie in den Suchergebnissen mit der rechten Maustaste auf die PowerShell Windows-App. Wählen Sie Als Administrator ausführen aus.
2. Führen Sie den folgenden Befehl aus, um eine Admin-Gruppe zu erstellen.

```
net localgroup SWE_ADMIN_GROUP /add
```

Sie müssen ein Benutzer in der SWE_ADMIN_GROUP Gruppe sein, um Administratorzugriff für das SiteWise Edge-Gateway zu gewähren.

3. Führen Sie den folgenden Befehl aus, um eine Benutzergruppe zu erstellen.

```
net localgroup SWE_USER_GROUP /add
```

Sie müssen ein Benutzer in der SWE_USER_GROUP Gruppe sein, um dem SiteWise Edge-Gateway reinen Zugriff zu gewähren.

4. Führen Sie den folgenden Befehl aus, um einen Benutzer hinzuzufügen. Ersetzen Sie *user-name* und *password* durch den Benutzernamen und das Passwort, das Sie erstellen möchten.

```
net user user-name password /add
```

5. Führen Sie den folgenden Befehl aus, um der Admin-Gruppe einen Benutzer hinzuzufügen. *user-name* Ersetzen Sie ihn durch den Benutzernamen, den Sie hinzufügen möchten.

```
net localgroup SWE_ADMIN_GROUP user-name /add
```

Sie können jetzt den Benutzernamen und das Passwort verwenden, um sich in der AWS IoT SiteWise For-Anwendung am SiteWise AWS OpsHub Edge-Gateway anzumelden.

Das SiteWise Edge-Gateway-Zertifikat verwalten

Sie können SiteWise Monitor und Anwendungen von Drittanbietern wie Grafana auf Ihren SiteWise Edge-Gateway-Geräten verwenden. Für diese Anwendungen ist eine TLS-Verbindung zum Dienst erforderlich. SiteWise Edge-Gateways verwenden derzeit ein selbstsigniertes Zertifikat. Wenn Sie zum Öffnen der Anwendungen einen Browser verwenden, z. B. ein SiteWise Monitor-Portal, erhalten Sie möglicherweise eine Warnung wegen eines nicht vertrauenswürdigen Zertifikats.

Im Folgenden wird gezeigt, wie Sie das vertrauenswürdige Zertifikat aus der AWS OpsHub AWS IoT SiteWise FOR-Anwendung herunterladen.

1. Melden Sie sich bei der Anwendung an.
2. Wählen Sie Einstellungen aus.
3. Wählen Sie für Authentifizierung die Option Zertifikat herunterladen aus.

Im Folgenden wird davon ausgegangen, dass Sie Google Chrome oder verwenden FireFox. Wenn Sie einen anderen Browser verwenden, lesen Sie in der entsprechenden Dokumentation zu Ihrem Browser nach. Gehen Sie wie folgt vor, um das Zertifikat, das Sie im vorherigen Schritt heruntergeladen haben, einem Browser hinzuzufügen:

- Wenn Sie Google Chrome verwenden, folgen Sie den Anweisungen zum [Einrichten von Zertifikaten](#) in der Google Chrome Enterprise-Hilfe.
- Wenn Sie Firefox verwenden, folgen Sie den Anweisungen [So laden Sie das Zertifikat in den Mozilla- oder Firefox-Browser](#) in der Oracle-Dokumentation.

Ändern Sie die Version der SiteWise Edge Gateway-Komponentenpakete

Sie können die AWS IoT SiteWise Konsole verwenden, um die Version der Komponentenpakete auf Ihren SiteWise Edge-Gateways zu ändern.

So ändern Sie die Version eines SiteWise Edge-Gateway-Komponentenpakets

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Gateways aus.
3. Wählen Sie das SiteWise Edge-Gateway aus, für das Sie die Packversionen ändern möchten.
4. Wählen Sie unter Gateway-Konfiguration die Option Softwareversionen anzeigen aus.
5. Wählen Sie auf der Seite Softwareversionen bearbeiten für das Paket, dessen Version Sie aktualisieren möchten, die Version aus, die Sie bereitstellen möchten, und wählen Sie Bereitstellen aus.
6. Wählen Sie Fertig aus.

SiteWise Edge-Gateways auflisten

Console

Um SiteWise Edge-Gateways aufzulisten

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Sehen Sie sich die Liste all Ihrer SiteWise Edge-Gateways an.

AWS CLI

Gehen Sie folgendermaßen vor AWS CLI, um Ihre verwendeten Gateways aufzulisten:

- Verwenden Sie den Befehl `list-gateways`, um alle Ihre Gateways anzuzeigen:

```
aws iotsitewise list-gateways
```

Dieser Befehl gibt eine Liste Ihrer Gateways mit ihren IDs Namen und anderen Informationen zurück.

Sie können auch Paginierungsparameter angeben:

```
aws iotsitewise list-gateways --max-results 10 --next-token your-token
```

Weitere Informationen finden Sie unter [list-gateways](#) in der Befehlsreferenz. AWS CLI

Beschreiben Sie ein Edge-Gateway SiteWise

Console

Um Gateway-Details anzuzeigen

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie den Namen des Gateways aus, für das Sie Details anzeigen möchten.
4. Sehen Sie sich die Gateway-Details auf der Edge-Gateway-Detailseite an.

AWS CLI

Gehen Sie folgendermaßen vor, um detaillierte Informationen zu einem bestimmten Gateway zu erhalten AWS CLI, das Sie verwenden:

- Verwenden Sie den Befehl `describe-gateway` mit der Gateway-ID:

```
aws iotsitewise describe-gateway --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Dieser Befehl gibt detaillierte Informationen über das Gateway zurück.

Weitere Informationen finden Sie unter [describe-gateway](#) in der AWS CLI Befehlsreferenz.

Erstellen Sie ein Edge-Gateway SiteWise

Console

Um ein SiteWise Edge-Gateway zu erstellen

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie Create gateway (Gateway erstellen).
4. Geben Sie einen Namen für Ihr Gateway ein.
5. Wählen Sie die Greengrass-Gruppe für Ihr Gateway aus.
6. Fügen Sie Ihrem Gateway optional Tags hinzu.
7. Wählen Sie Erstellen aus.

AWS CLI

Gehen Sie folgendermaßen vor AWS CLI, um ein neues SiteWise IoT-Gateway mit zu erstellen:

- Verwenden Sie den Befehl `create-gateway`, um ein neues Gateway zu erstellen:

```
aws iotsitewise create-gateway \
  --gateway-name "NewSiteWiseGateway" \
  --gateway-platform '{
    "greengrass": {
      "groupArn": "arn:aws:greengrass:us-east-1:123456789012:group/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
    }
  }' \
  --tags '{
    "Environment": "Production",
    "Location": "Factory1"
  }'
```

Dieser Befehl gibt die ID und den ARN des neuen Gateways zurück:

```
{
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
  "gatewayArn": "arn:aws:iotsitewise:us-east-1:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
```

```
}
```

Weitere Informationen finden Sie unter [create-gateway](#) in der AWS CLI Befehlsreferenz.

Aktualisieren Sie ein Edge-Gateway SiteWise

Console

Um ein SiteWise Edge-Gateway zu aktualisieren

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie das Gateway aus, das Sie aktualisieren möchten.
4. Wählen Sie Bearbeiten aus.
5. Aktualisieren Sie den Gateway-Namen oder andere Einstellungen nach Bedarf.
6. Wählen Sie Speichern.

AWS CLI

Gehen Sie folgendermaßen vor AWS CLI, um ein vorhandenes Gateway mit zu aktualisieren:

- Verwenden Sie den Befehl `update-gateway`, um den Namen eines Gateways zu aktualisieren:

```
aws iotsitewise update-gateway \  
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \  
  --gateway-name "UpdatedGatewayName"
```

Dieser Befehl erzeugt keine Ausgabe, wenn er erfolgreich ist.

Weitere Informationen finden Sie unter [update-gateway](#) in der Befehlsreferenz. AWS CLI

Aktualisieren Sie die Gateway-Funktionskonfiguration

Console

Um die Gateway-Fähigkeitskonfiguration zu aktualisieren

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie den Namen des Gateways aus, das Sie aktualisieren möchten.
4. Wählen Sie im Abschnitt Datenquellen die Option Bearbeiten aus.
5. Aktualisieren Sie die Datenquellenkonfiguration nach Bedarf.
6. Wählen Sie Speichern.

AWS CLI

Gehen Sie folgendermaßen vor AWS CLI, um die Funktionskonfiguration eines Gateways mithilfe von zu aktualisieren:

- Verwenden Sie den `update-gateway-capability-configuration` Befehl, um die Funktionskonfiguration zu aktualisieren:

```
aws iotsitewise update-gateway-capability-configuration \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
  --capability-namespace "iotsitewise:opcuacollector:1" \
  --capability-configuration '{
    "sources": [
      {
        "name": "OPC-UA Server",
        "endpoint": {
          "certificateTrust": {
            "type": "TrustAny"
          },
          "endpointUri": "opc.tcp://10.0.0.1:4840",
          "securityPolicy": "NONE",
          "messageSecurityMode": "NONE",
          "identityProvider": {
            "type": "Anonymous"
          }
        },
        "measurementDataStreamPrefix": ""
      }
    ]
  }
```



```
}  
]  
'
```

Dieser Befehl gibt den Capability-Namespace und den Synchronisierungsstatus zurück:

```
{  
  "capabilityNamespace": "iotsitewise:opcuacollector:1",  
  "capabilitySyncStatus": "CONFIGURING"  
}
```

Weitere Informationen finden Sie [update-gateway-capability-configuration](#) in der AWS CLI Befehlsreferenz.

Tag-Gateway-Ressourcen

Console

Um eine Gateway-Ressource zu taggen

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie den Namen des Gateways aus, das Sie taggen möchten.
4. Wählen Sie die Registerkarte Tags aus.
5. Wählen Sie Tags verwalten aus.
6. Wählen Sie Neues Tag hinzufügen und geben Sie für jedes Tag einen Schlüssel und einen Wert ein.
7. Wählen Sie Speichern.

AWS CLI

Gehen Sie folgendermaßen vor AWS CLI, um einem Gateway mithilfe von Tags hinzuzufügen:

- Verwenden Sie den Befehl `tag-resource`, um einem Gateway Tags hinzuzufügen:

```
aws iotsitewise tag-resource \
```

```
--resource-arn "arn:aws:iotsitewise:us-east-1:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE" \
--tags '{
  "Department": "Operations",
  "Project": "FactoryAutomation"
}'
```

Dieser Befehl erzeugt keine Ausgabe, wenn er erfolgreich ist.

Weitere Informationen finden Sie unter [tag-resource](#) in der Befehlsreferenz. AWS CLI

Listet die Tags für ein Gateway auf

Console

Um Tags für ein Gateway aufzulisten

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie den Namen des Gateways aus, für das Sie Tags anzeigen möchten.
4. Wählen Sie die Registerkarte Tags aus.
5. Sehen Sie sich die Liste der mit dem Gateway verknüpften Tags an.

AWS CLI

Gehen Sie wie folgt vor, um die Tags aufzulisten, die einem Gateway zugeordnet sind AWS CLI, das verwendet:

- Verwenden Sie den `list-tags-for-resource` Befehl, um Tags für ein Gateway aufzulisten:

```
aws iotsitewise list-tags-for-resource \
  --resource-arn "arn:aws:iotsitewise:us-east-1:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
```

Dieser Befehl gibt die mit dem Gateway verknüpften Tags zurück:

```
{
  "tags": {
```

```
"Environment": "Production",  
"Location": "Factory1",  
"Department": "Operations",  
"Project": "FactoryAutomation"  
}  
}
```

Weitere Informationen finden Sie [list-tags-for-resource](#) in der AWS CLI Befehlsreferenz.

Tags von einem Gateway entfernen

Console

Um Tags von einem Gateway zu entfernen

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie den Namen des Gateways aus, von dem Sie Tags entfernen möchten.
4. Wählen Sie die Registerkarte Tags aus.
5. Wählen Sie Tags verwalten aus.
6. Wählen Sie das Entfernen-Symbol (X) neben jedem Tag, den Sie entfernen möchten.
7. Wählen Sie Speichern.

AWS CLI

Gehen Sie wie folgt vor AWS CLI, um Tags mit einem Gateway zu entfernen:

- Verwenden Sie den Befehl `untag-resource`, um Tags von einem Gateway zu entfernen:

```
aws iotsitewise untag-resource \  
  --resource-arn "arn:aws:iotsitewise:us-east-1:123456789012:gateway/  
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE" \  
  --tag-keys '["Project", "Department"]'
```

Dieser Befehl erzeugt keine Ausgabe, wenn er erfolgreich ist.

Weitere Informationen finden Sie unter [untag-resource](#) in der Befehlsreferenz. AWS CLI

Aktualisieren Sie die Version einer Komponente AWS IoT SiteWise

Aktualisieren Sie die AWS IoT SiteWise Gateway-Komponente auf Ihrem AWS IoT Greengrass Kerngerät, um sicherzustellen, dass Sie auf die neuesten Funktionen, Leistungsverbesserungen und Sicherheitspatches zugreifen können.

Um eine AWS IoT SiteWise Komponente zu aktualisieren auf AWS IoT Greengrass

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Edge-Gateways aus.
3. Wählen Sie das zu bearbeitende Gateway aus und klicken Sie auf Bearbeiten.
4. Wählen Sie unter Edge-Funktionen unter Softwareversionen die Option Verfügbare Softwareupdates aus. Die Seite Softwareversionen bearbeiten wird angezeigt.
5. Wählen Sie die Komponentenversion aus.

Note

Es wird empfohlen, die neueste verfügbare Version auszuwählen. Durch die Beibehaltung der Gateway-Komponenten up-to-date können Sie die optimale Funktionalität für die industrielle Datenerfassung und -verarbeitung aufrechterhalten.

6. Wählen Sie Bereitstellen. Dadurch wird eine AWS IoT Greengrass V2 Bereitstellung zur Aktualisierung der AWS IoT SiteWise Komponente auf dem Gateway gestartet.

Löschen Sie ein SiteWise Edge-Gateway

Console

Um das SiteWise Edge-Gateway zu löschen

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie das Gateway aus, das Sie löschen möchten.
4. Wählen Sie Löschen aus.

5. Um zu bestätigen, dass Sie das Gateway löschen möchten, geben Sie „Löschen“ ein und wählen Sie dann im angezeigten Fenster Löschen.

AWS CLI

Gehen Sie folgendermaßen vor AWS CLI, um ein Gateway mit zu löschen:

1. Führen Sie Ihre Gateways auf, um die Gateway-ID des Gateways zu identifizieren, das Sie löschen möchten.

```
aws iotsitewise list-gateways
```

Dieser Befehl gibt eine Liste Ihrer Gateways mit ihren IDs Namen und anderen Informationen zurück:

```
{
  "gatewaySummaries": [
    {
      "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
      "gatewayName": "ExampleCorpGateway",
      "gatewayCapabilitySummaries": [
        {
          "capabilityNamespace": "iotsitewise:opcuacollector:1",
          "capabilitySyncStatus": "IN_SYNC"
        }
      ],
      "creationDate": 1588369971.457,
      "lastUpdateDate": 1588369971.457
    }
  ]
}
```

2. Löschen Sie das Gateway, indem Sie seine ID angeben:

```
aws iotsitewise delete-gateway --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Dieser Befehl erzeugt keine Ausgabe, wenn er erfolgreich ist.

Note

Wenn Sie ein Gateway löschen, verbleiben einige Dateien des Gateways im Dateisystem Ihres Gateways.

3. Um zu überprüfen, ob das Gateway gelöscht wurde, können Sie Ihre Gateways erneut auflisten:

```
aws iotsitewise list-gateways
```

Das gelöschte Gateway sollte nicht mehr in der Liste erscheinen.

Weitere Informationen finden Sie unter [delete-gateway](#) in der AWS CLI Befehlsreferenz.

SiteWise Edge-Gateways sichern und wiederherstellen

In diesem Thema wird beschrieben, wie Sie SiteWise Edge-Gateways wiederherstellen und Ihre Metrikdaten sichern. Wenn Sie Probleme mit einem defekten SiteWise Edge-Gateway auf demselben Computer haben und das Problem beheben müssen, lesen Sie bitte die AWS IoT SiteWise Dokumentation [Fehlerbehebung bei SiteWise Edge-Gateway-Problemen](#).

Note

Die in diesem Thema behandelten Anleitungen gelten für SiteWise Edge-Gateways, die auf AWS IoT Greengrass V2 Version 2.1.0 oder höher installiert sind.

Tägliche Backups von metrischen Daten

Das Erstellen eines Backups ist wichtig, wenn Sie die Daten auf einen neuen Computer übertragen oder wiederherstellen möchten. Durch die Sicherung Ihrer Daten wird das Risiko eines Verlusts von Betriebsdaten während eines Übertragungs- oder Wiederherstellungsvorgangs erheblich reduziert.

Dieser Abschnitt bezieht sich auf Gateways, die das Data Processing Pack verwenden. Weitere Informationen zum Datenverarbeitungspaket finden Sie unter [Konfigurieren Sie ein Asset-Modell für die Datenverarbeitung auf Edge SiteWise](#).

Der Influxdb-Ordnerpfad lautet wie folgt:

Linux

```
/greengrass/v2/work/aws.iot.SiteWiseEdgeProcessor/influxdb
```

Windows

```
C:\greengrass\v2\work\aws.iot.SiteWiseEdgeProcessor\influxdb
```

Wir empfehlen, dass Sie den gesamten Ordner mit allem, was sich darunter befindet, sichern.

Wir empfehlen Ihnen, Ihre Messdaten regelmäßig vom 1.0 SiteWise Edge entweder auf einer externen Festplatte oder in der AWS Cloud zu sichern.

Stellen Sie ein SiteWise Edge-Gateway wieder her

Bevor Sie versuchen, ein SiteWise Edge-Gateway wiederherzustellen, stellen Sie sicher, dass alle mit dem Gateway verbundenen Edge-Geräte gestoppt oder getrennt sind.

Gehen Sie wie folgt vor, um ein SiteWise Edge-Gateway wiederherzustellen:

1. Verwenden Sie das Installationsskript, das beim Erstellen des SiteWise Edge-Gateways heruntergeladen wurde, um das SiteWise Edge-Gateway auf der neuen Maschine wiederherzustellen. Lesen Sie das Verfahren [zur Installation der SiteWise Edge-Gateway-Software auf Ihrem lokalen Gerät](#), um das SiteWise Edge-Gateway einzurichten.

Wenn Sie das Installationsskript verlieren oder nicht finden können, wenden Sie sich bitte an den [AWS Kundensupport](#).

2. Melden Sie sich nach der Installation des SiteWise Edge-Gateways an der [AWS IoT Greengrass Konsole](#) an.
3. Um die Komponenten erneut bereitzustellen, navigieren Sie zu Verwalten und wählen Sie dann unter AWS IoT Greengrass Geräte die Option Core-Geräte aus.
4. Wählen Sie in der Tabelle mit den AWS IoT Greengrass Kerngeräten das Kerngerät aus, das Ihrem SiteWise Edge-Gateway entspricht.
5. Öffnen Sie auf der Geräteseite die Registerkarte Bereitstellungen und wählen Sie Ihre Bereitstellungs-ID aus. Dadurch wird die Seite Bereitstellungen mit Ihrer ausgewählten ID geöffnet.

6. Sobald Sie sich auf der Seite Bereitstellungen befinden, drücken Sie oben rechts auf die Schaltfläche Aktionen und wählen Sie die Option Überarbeiten aus, um eine neue Bereitstellung zu starten. Konfigurieren Sie die Bereitstellung. Wenn Sie die Bereitstellung unverändert lassen möchten, fahren Sie mit Überprüfen und Bereitstellen fort.
7. Warten Sie, bis der Bereitstellungsstatus lautet `Completed`.

 Note

Außerdem dauert es einige Minuten, bis alle Komponenten auf dem SiteWise Edge vollständig eingerichtet und ausgeführt sind.

AWS IoT SiteWise Daten wiederherstellen

Gehen Sie wie folgt vor, um Daten auf einem neuen Computer wiederherzustellen.

1. Kopieren Sie den `influxdb` Ordner auf das neue Gerät.
2. Stoppen Sie die SiteWise EdgeProcessor Komponente, indem Sie den folgenden Befehl in Ihrem Terminal ausführen:

Linux

```
sudo /greengrass/v2/bin/greengrass-cli component stop -n  
aws.iot.SiteWiseEdgeProcessor
```

Windows

```
C:\greengrass\v2\bin\greengrass-cli component stop -n  
aws.iot.SiteWiseEdgeProcesso
```

3. Suchen Sie den Pfad, in dem Sie Ihre Daten gesichert haben, und führen Sie den folgenden Befehl aus:

Linux

```
sudo yes | sudo cp -rf <influxdb_backup_path> /greengrass/v2/work/  
aws.iot.SiteWiseEdgeProcessor/influxdb
```


PowerShell

```
Copy-Item -Recurse -Force <influxdb_backup_path>\* C:\greengrass  
v2\work\aws.iot.SiteWiseEdgeProcessor\
```

Windows

```
robocopy <influxdb_backup_path> C:\greengrass\v2\work  
\aws.iot.SiteWiseEdgeProcessor\ /E
```

4. Starten Sie die SiteWiseEdgeProcessor Komponente neu:

Linux

```
sudo /greengrass/v2/bin/greengrass-cli component restart -n  
aws.iot.SiteWiseEdgeProcessor
```

Windows

```
C:\greengrass\v2\bin\greengrass-cli component restart -n  
aws.iot.SiteWiseEdgeProcessor
```

Bestätigen Sie erfolgreiche Backups und Wiederherstellungen

Verwenden Sie dieses Verfahren, um Ihre gesicherten Daten und Edge-Gateway-Wiederherstellungen zu validieren. SiteWise

Note

Dieses Verfahren setzt voraus, dass Sie für installiert haben. AWS OpsHub AWS IoT SiteWise Weitere Informationen finden Sie unter [Verwalten von SiteWise Edge-Gateways mithilfe von AWS OpsHub](#). AWS IoT SiteWise

1. Öffnen AWS OpsHub für AWS IoT SiteWise.
2. Überprüfen Sie auf der Seite mit den SiteWise Edge-Gateway-Einstellungen den Status der einzelnen Komponenten, die in der Tabelle Komponenten aufgeführt sind. Vergewissern Sie sich, dass die Statusfarbe grün ist und dass auf der Anzeige RUNNING (LÄUFT) angezeigt wird.
3. Überprüfen Sie Ihre früheren Daten im Portal-Dashboard, um zu überprüfen, ob sowohl die alten als auch die neuen Daten ordnungsgemäß eingerichtet sind. Zwischen vergangenen und neuen

Daten wird es zu Ausfallzeiten kommen. Sie sollten davon ausgehen, dass eine Dauer angezeigt wird, für die keine Datenpunkte erfasst werden.

Wenn Sie Probleme beim Sichern oder Wiederherstellen eines SiteWise Edge-Gateways haben, finden Sie weitere Informationen zu den folgenden Themen zur [Fehlerbehebung bei einem AWS IoT SiteWise Edge-Gateway](#).

Ältere Gateways (AWS IoT Greengrass Version 1)

Note

SiteWise Edge-Gateways, AWS IoT Greengrass V1 die auf laufen, sind nur verfügbar, wenn Sie diese Funktion vor dem 29. Juli 2021 verwendet haben. Weitere Informationen zum Betrieb eines AWS IoT SiteWise Gateways mit finden Sie AWS IoT Greengrass V2 unter [Self-host ein AWS IoT SiteWise Edge-Gateway mit AWS IoT Greengrass V2](#).

SiteWise Edge-Gateways laufen jetzt ausschließlich darauf AWS IoT Greengrass V2 und bieten erweiterte Funktionalität und verbesserte Leistung für Ihre industriellen IoT-Anwendungen. Diese neueste Version AWS IoT Greengrass V2 stellt eine architektonische Weiterentwicklung dar, die auf einem modernen komponentenbasierten Framework basiert und die modulare Softwarebereitstellung ermöglicht. Sie optimiert die Installation durch ein einheitliches Installationsprogramm und bietet Entwicklern gleichzeitig mehr Flexibilität bei der Bereitstellung benutzerdefinierter Komponenten und der Durchführung lokaler Tests. Das komponentenbasierte Modell ermöglicht ein effizienteres Ressourcenmanagement und bietet einen vereinfachten Konfigurationsansatz anhand von Komponentenrezepten. Dieses Design ermöglicht eine bessere Handhabung von Abhängigkeiten zwischen Komponenten, unterstützt kontinuierliche Bereitstellungspraktiken und bietet erweiterte CLI-Funktionen für die lokale Entwicklung. Darüber hinaus AWS IoT Greengrass V2 zentralisiert es das Konfigurationsmanagement AWS IoT Core und bietet verbesserte Protokollierungs- und Überwachungsfunktionen, die alle durch ein detaillierteres Sicherheitsberechtigungsmodell geschützt sind.

Weitere Informationen zu den ersten Schritten mit SiteWise Edge-Gateways finden Sie unter AWS IoT Greengrass V2 [AWS IoT SiteWise Anforderungen an das selbst gehostete Edge-Gateway](#). Diese Ressourcen enthalten step-by-step Anweisungen zur Einrichtung Ihrer Gateways, zur Konfiguration von Datenquellen und zur Verwaltung Ihrer industriellen IoT-Infrastruktur.

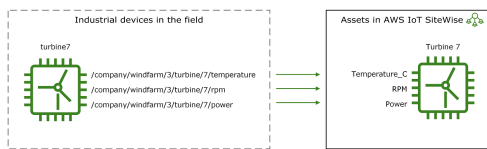
 Note

Im AWS Zuge der kontinuierlichen Innovation und Verbesserung seiner IoT-Dienste wird empfohlen, über die neuesten Funktionen und Verbesserungen auf dem Laufenden zu bleiben. Suchen Sie regelmäßig in der AWS IoT SiteWise AWS IoT Greengrass Dokumentation nach neuen Funktionen, mit denen Sie Ihre industriellen IoT-Lösungen weiter optimieren können.

Modellieren Sie Industrieanlagen

Überblick über die Vermögenswerte

Sie können virtuelle Darstellungen Ihres Industriebetriebs mit AWS IoT SiteWise Anlagen erstellen. Ein Asset steht für ein Gerät, eine Ausrüstung oder einen Prozess, der einen oder mehrere Datenströme in die AWS Cloud hochlädt. Ein Komponentengerät kann beispielsweise eine Windturbine sein, die Lufttemperatur, Drehzahl der Propeller und Zeitreihen für die Leistungsausgabe an Komponenteneigenschaften in AWS IoT SiteWise sendet.

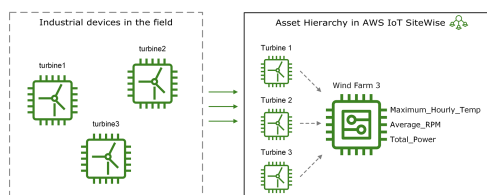


Eigenschaftsalias identifizieren Gerätedatenströme

Jeder Daten-Stream entspricht einem eindeutigen Alias der Eigenschaft. Beispielsweise dient der Alias `/company/windfarm/3/turbine/7/temperature` zur eindeutigen Identifizierung des Temperaturdaten-Streams von Turbine 7 in Windpark 3. Sie können AWS IoT SiteWise Anlagen so konfigurieren, dass eingehende Messdaten mithilfe mathematischer Ausdrücke transformiert werden, z. B. um Temperaturdaten von Celsius in Fahrenheit umzuwandeln.

Anlagenhierarchien stellen Gerätebeziehungen dar

Eine Komponente kann auch eine logische Gruppierung von Geräten darstellen, etwa einen gesamten Windpark. Sie können Anlagen mit anderen Anlagen verknüpfen, um Anlagenhierarchien zu erstellen, die komplexe Industriebetriebe repräsentieren. Anlagen können auf die Daten in ihren zugehörigen untergeordneten Anlagen zugreifen. Auf diese Weise können Sie AWS IoT SiteWise Ausdrücke verwenden, um aggregierte Kennzahlen zu berechnen, z. B. die Nettoleistung eines Windparks.



Anlagenmodelle standardisieren die Darstellung von Geräten

Sie müssen jedes Asset anhand eines Asset-Modells erstellen. Komponentenmodelle sind deklarative Strukturen zur Standardisierung des Formats Ihrer Komponenten. Inventarmodelle setzen konsistente Informationen für mehrere Anlagen desselben Typs voraus, sodass Sie Daten in Anlagen verarbeiten können, die Gerätegruppen repräsentieren. Eine Produktionsanlage könnte beispielsweise über ein Anlagenmodell für CNC-Maschinen verfügen, das Eigenschaften wie Temperatur, Ausfallzeit und Produktionsrate definiert. Im obigen Diagramm verwenden Sie dasselbe Anlagenmodell für alle drei Turbinen, da sie dieselben Eigenschaften haben.

Modellierungsoptionen für Industrieanlagen

Berücksichtigen Sie bei der Gestaltung Ihrer Darstellung von Industrieanlagen die folgenden Optionen:

- Anlagenmodelle stehen für bestimmte Arten von Anlagen oder Prozessen. Sie müssen jede physische Anlage anhand eines Anlagenmodells erstellen. Beispielsweise kann eine chemische Verarbeitungsanlage separate Anlagenmodelle für Reaktoren, Mischer und Lagertanks haben.
- Komponentenmodelle definieren wiederverwendbare Unterbaugruppen, die Sie in Anlagenmodelle oder andere Komponentenmodelle aufnehmen können. Sie könnten beispielsweise ein Temperatursensor-Komponentenmodell in mehrere Anlagenmodelle in einer Fabrik einbeziehen.
- Schnittstellen für Anlagenmodelle wenden Standards für verschiedene Anlagenmodelle an. Beispielsweise könnte eine Schnittstelle für „rotierende Geräte“ Standardeigenschaften für Vibration, Temperatur und Drehzahl definieren, die für Pumpen, Turbinen und Motoren gelten, obwohl jede Maschine über ein eigenes Anlagenmodell verfügt.

Anlagen erstellen und verwalten

Nachdem Sie Ihre Komponentenmodelle definiert haben, können Sie Ihre industriellen Komponenten erstellen. Wählen Sie zum Erstellen einer Komponente ein ACTIVE-Komponentenmodell aus, um eine Komponente anhand von diesem Modell zu erstellen. Anschließend können Sie komponentenspezifische Informationen wie Daten-Stream-Aliase und Attribute eintragen. Im obigen Diagramm erstellen Sie drei Turbinenkomponenten von einem Komponentenmodell ausgehend und ordnen dann Daten-Stream-Aliase wie `/company/windfarm/3/turbine/7/temperature` für jede Turbine zu.

Sie können auch vorhandene Assets, Asset-Modelle und Komponentenmodelle aktualisieren und löschen. Wenn Sie ein Komponentenmodell aktualisieren, spiegelt jede Komponente, die auf diesem Komponentenmodell basiert, alle Änderungen wider, die Sie am zugrunde liegenden Modell vornehmen. Wenn Sie ein Komponentenmodell aktualisieren, gilt dies für jedes Asset, das auf jedem Asset-Modell basiert, das auf das Komponentenmodell verweist.

Verwaltung komplexer Anlagenmodelle

Ihre Anlagenmodelle können sehr komplex sein, z. B. wenn Sie ein kompliziertes Gerät mit vielen Unterkomponenten modellieren. Um solche Anlagenmodelle zu organisieren und zu verwalten, können Sie benutzerdefinierte Verbundmodelle verwenden, um zusammengehörige Eigenschaften zu gruppieren oder gemeinsam genutzte Komponenten wiederzuverwenden. Weitere Informationen finden Sie unter [Benutzerdefinierte zusammengesetzte Modelle \(Komponenten\)](#).

Komponenten- und Modellzustände

Wenn Sie ein Asset, ein Assetmodell oder ein Komponentenmodell erstellen, aktualisieren oder löschen, dauert es einige Zeit, bis die Änderungen übernommen werden. AWS IoT SiteWise löst diese Vorgänge asynchron auf und aktualisiert den Status jeder Ressource. Jedes Asset-, Asset- und Komponentenmodell verfügt über ein Statusfeld, das den Status der Ressource und etwaige Fehlermeldungen enthält. Der Status kann einer der folgenden Werte sein:

- **ACTIVE**— Die Ressource ist aktiv. Dies ist der einzige Status, in dem Sie Ressourcen, Anlagenmodelle und Komponentenmodelle abfragen und mit ihnen interagieren können.
- **CREATING**— Die Ressource wird gerade erstellt.
- **UPDATING**— Die Ressource wird aktualisiert.
- **DELETING**— Die Ressource wird gelöscht.
- **PROPAGATING**— (Nur Asset-Modelle und Komponentenmodelle) Die Änderungen werden auf alle abhängigen Ressourcen übertragen (vom Asset-Modell zu den Assets oder vom Komponentenmodell zu den Asset-Modellen).
- **FAILED**— Die Ressource konnte während eines Erstellungs- oder Aktualisierungsvorgangs nicht validiert werden, möglicherweise aufgrund eines Zirkelverweises in einem Ausdruck. Sie können Ressourcen löschen, die sich im FAILED Status befinden.

Bei einigen Vorgängen zum Erstellen, Aktualisieren und Löschen wird ein AWS IoT SiteWise Objekt, ein Anlagenmodell oder ein Komponentenmodell in einen anderen Zustand versetzt, als ACTIVE

wenn der Vorgang aufgelöst wird. Um eine Ressource abzufragen oder mit ihr zu interagieren, nachdem Sie einen dieser Vorgänge ausgeführt haben, müssen Sie warten, bis sich der Status auf ACTIVE ändert. Andernfalls schlagen Ihre Anfragen fehl.

Themen

- [Überprüfen Sie den Status eines Assets](#)
- [Überprüfen Sie den Status eines Asset- oder Komponentenmodells](#)

Überprüfen Sie den Status eines Assets

Sie können die AWS IoT SiteWise Konsole oder API verwenden, um den Status eines Assets zu überprüfen.

Themen

- [Überprüfen Sie den Status eines Assets \(Konsole\)](#)
- [Überprüfen Sie den Status eines Assets \(AWS CLI\)](#)

Überprüfen Sie den Status eines Assets (Konsole)

Gehen Sie wie folgt vor, um den Status einer Komponente in der AWS IoT SiteWise-Konsole zu überprüfen.

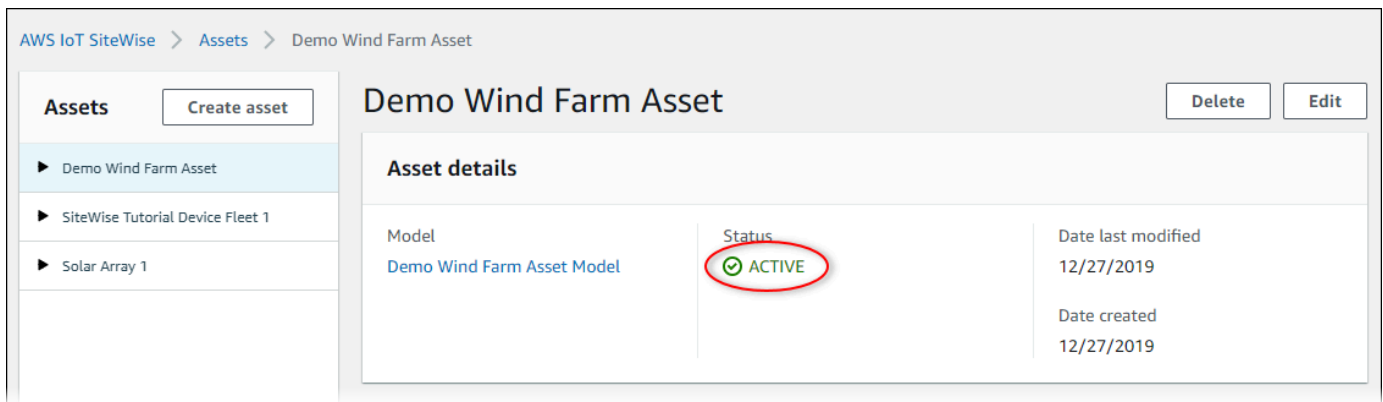
So überprüfen Sie den Status einer Komponente (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im linken Navigationsbereich Komponenten aus.
3. Wählen Sie die zu prüfende Komponente aus.

Tip

Sie können eine Komponentenhierarchie mithilfe des Pfeilsymbols erweitern, um nach Ihrer Komponente zu suchen.

4. Suchen Sie den Status im Bereich Komponentendetails.



Überprüfen Sie den Status eines Assets (AWS CLI)

Sie können das AWS Command Line Interface (AWS CLI) verwenden, um den Status eines Assets zu überprüfen.

Um den Status eines Assets zu überprüfen, verwenden Sie die [DescribeAsset](#) Operation mit dem `assetId` Parameter.

Um den Status eines Assets zu überprüfen (AWS CLI)

- Verwenden Sie den folgenden Befehl, um die Komponente zu beschreiben. `asset-id` Ersetzen Sie ihn durch die ID oder die externe ID des Assets. Die externe ID ist eine benutzerdefinierte ID. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.

```
aws iotsitewise describe-asset --asset-id asset-id
```

Die Operation gibt eine Antwort zurück, die Details der Komponente enthält. Die Antwort enthält ein `assetStatus` Objekt mit der folgenden Struktur:

```
{
  ...
  "assetStatus": {
    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  }
}
```



```
}
```

Der Status der Komponente befindet sich in `assetStatus.state` im JSON-Objekt.

Überprüfen Sie den Status eines Asset- oder Komponentenmodells

Sie können die AWS IoT SiteWise Konsole oder API verwenden, um den Status eines Asset- oder Komponentenmodells zu überprüfen.

Themen

- [Überprüfen Sie den Status eines Asset- oder Komponentenmodells \(Konsole\)](#)
- [Überprüfen Sie den Status eines Asset- oder Komponentenmodells \(AWS CLI\)](#)

Überprüfen Sie den Status eines Asset- oder Komponentenmodells (Konsole)

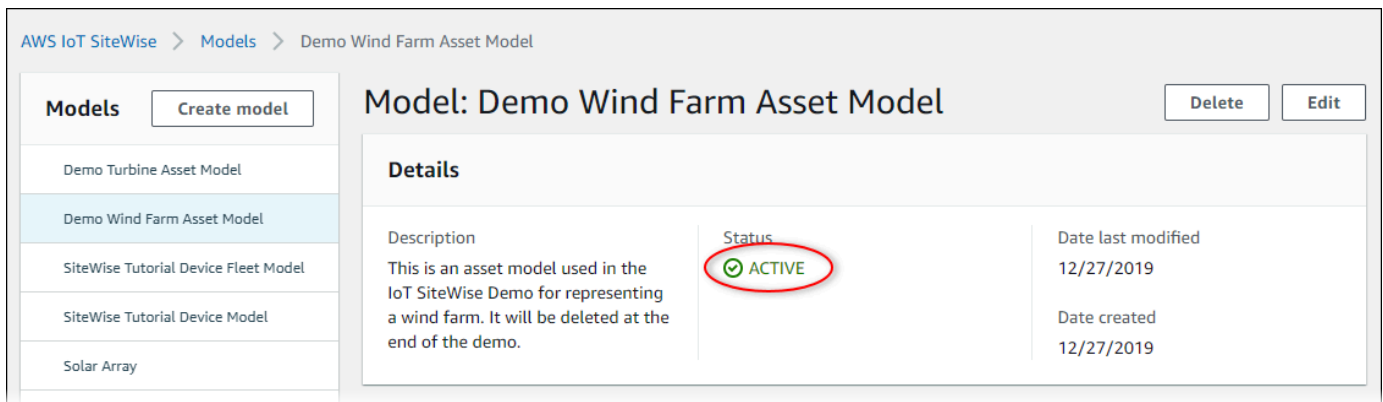
Gehen Sie wie folgt vor, um den Status eines Asset- oder Komponentenmodells in der AWS IoT SiteWise Konsole zu überprüfen.

Tip

Objektmodelle und Komponentenmodelle werden beide im Navigationsbereich unter Modelle aufgeführt. Der Bereich „Details“ des ausgewählten Asset- oder Komponentenmodells gibt an, um welchen Typ es sich handelt.

Um den Status eines Asset- oder Komponentenmodells (Konsole) zu überprüfen

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Klicken Sie im Navigationsbereich auf Models (Modelle).
3. Wählen Sie das zu überprüfende Modell aus.
4. Suchen Sie den Status im Bereich Details.



Überprüfen Sie den Status eines Asset- oder Komponentenmodells (AWS CLI)

Sie können den verwenden AWS CLI, um den Status eines Asset- oder Komponentenmodells zu überprüfen.

Um den Status eines Asset- oder Komponentenmodells zu überprüfen, verwenden Sie die [DescribeAssetModel](#) Operation mit dem `assetModelId` Parameter.

Tip

Der AWS CLI definiert Komponentenmodelle als eine Art von Anlagenmodell. Daher verwenden Sie dieselbe [DescribeAssetModel](#) Operation für beide Modelltypen. Das `assetModelType` Feld in der Antwort gibt an, ob es sich um ein `ASSET_MODEL` oder ein `COMPONENT_MODEL` handelt.

Um den Status eines Asset- oder Komponentenmodells zu überprüfen (AWS CLI)

- Führen Sie den folgenden Befehl aus, um das Modell zu beschreiben. `asset-model-id` Ersetzen Sie es durch die ID oder die externe ID des Asset- oder Komponentenmodells. Die externe ID ist eine benutzerdefinierte ID. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

Die Operation gibt eine Antwort zurück, die die Details des Modells enthält. Die Antwort enthält ein `assetModelStatus`-Objekt, das die folgende Struktur aufweist.

```
{
  ...
  "assetModelStatus": {
    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  }
}
```

Der Status des Modells befindet sich `assetModelStatus.state` im JSON-Objekt in.

Versionen von Asset-Modellen

AWS IoT SiteWise unterstützt die asynchrone Verarbeitung von Erstellungs- und Aktualisierungsvorgängen für Anlagenmodelle und Komponentenmodelle. Außerdem wird der Status des Modells aktualisiert.

AWS IoT SiteWise überträgt die Änderungen eines gültigen Modells in den Erstellungs- und Aktualisierungsanforderungen an seine abhängigen Ressourcen (vom Asset-Modell zu den Assets oder vom Komponentenmodell zu den Asset-Modellen). Anschließend wird das Modell in den ACTIVE Status versetzt.

Wenn die angegebene Modelldefinition ungültig ist, AWS IoT SiteWise wird das Modell in einen FAILED Status versetzt. Die Änderungen werden nicht auf die abhängigen Ressourcen übertragen. Die abhängigen Ressourcen beziehen sich auf die letzte Modelldefinition, die propagiert wurde, als sich das Modell in einem ACTIVE bestimmten Zustand befand.

Basierend auf den obigen Informationen gibt es für Modelldefinitionen zwei Arten von Modellversionen:

1. Letzte Version — Die neueste Definition, die als Teil einer Erstellungs- oder Aktualisierungsanforderung akzeptiert wurde.
2. Aktive Version — Die neueste Definition wurde erfolgreich verarbeitet, und der Modellstatus ist ACTIVE.

Standardmäßig werden Details zur neuesten Version des Modells zurückgegeben, wenn Describe-APIs für ein Asset- oder Komponentenmodell aufgerufen werden. Es gibt Szenarien, in denen die aktive Version des Asset- oder Komponentenmodells benötigt wird. Nachfolgend finden Sie Beispielszenarien:

- Ein Aktualisierungsvorgang mit einer ungültigen Definition versetzt Ihr Asset-Modell in einen FAILED Zustand. Sie müssen Ihre Änderungen rückgängig machen, indem Sie die aktive Version des Asset-Modells abrufen und eine weitere Aktualisierungsanforderung erstellen, die auf diese gültige Definition verweist.
- AWS IoT SiteWiseEs gibt eine Anwendung, in der Kunden Anlagen und ihre entsprechenden Anlagenmodelle einsehen können. Wenn ein Benutzer auf die Definition des Anlagenmodells verweist, die einer bestimmten Anlage entspricht, und sich das Anlagenmodell in einem vorübergehenden FAILED Zustand befindet UPDATINGPROPAGATING, gibt die neueste Version die Anlagenmodelldefinition zurück, die noch nicht auf die Anlagen übertragen wurde. In diesem Fall müssen Sie die aktive Version des Anlagenmodells für Kunden abrufen.

Themen

- [Rufen Sie die aktive Version eines Asset- oder Komponentenmodells \(Konsole\) ab](#)
- [Rufen Sie die aktive Version eines Asset- oder Komponentenmodells ab \(AWS CLI\)](#)

Rufen Sie die aktive Version eines Asset- oder Komponentenmodells (Konsole) ab

Gehen Sie wie folgt vor, um die aktive Version eines Asset- oder Komponentenmodells in der AWS IoT SiteWise Konsole abzurufen.

Tip

Objektmodelle und Komponentenmodelle werden beide im Navigationsbereich unter Modelle aufgeführt. Der Bereich „Details“ des ausgewählten Asset- oder Komponentenmodells gibt an, um welchen Typ es sich handelt.

Um die aktive Version eines Asset- oder Komponentenmodells (Konsole) abzurufen

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).

2. Klicken Sie im Navigationsbereich auf Models (Modelle).
3. Wählen Sie das Modell aus, dessen aktive Version abgerufen werden soll.
 - a. Wenn sich das Modell in einem ACTIVE Status befindet, wird die aktive Version angezeigt.
 - b. Wenn sich das Modell in einem vorübergehenden FAILED Zustand befindet
UPDATINGPROPAGATING, suchen Sie im Bereich Details unter Status nach der Option
Aktive Version anzeigen.

Rufen Sie die aktive Version eines Asset- oder Komponentenmodells ab (AWS CLI)

Verwenden Sie die AWS CLI, um die aktive Version eines Asset- oder Komponentenmodells abzurufen.

Verwenden Sie die [DescribeAssetModel](#) Operation mit dem `assetModelVersion` Parameter, um die aktive Version eines Asset- oder Komponentenmodells abzurufen.

Tip

Der AWS CLI definiert Komponentenmodelle als eine Art von Asset-Modell. Daher verwenden Sie dieselbe [DescribeAssetModel](#) Operation für beide Modelltypen. Das `assetModelType` Feld in der Antwort gibt an, ob es sich um ein `ASSET_MODEL` oder ein `COMPONENT_MODEL` handelt.

Um die aktive Version eines Asset- oder Komponentenmodells abzurufen (AWS CLI)

- Führen Sie den folgenden Befehl aus, um das Modell zu beschreiben. *asset-model-id* Ersetzen Sie es durch die ID oder die externe ID des Asset- oder Komponentenmodells. Die externe ID ist eine benutzerdefinierte ID. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id --asset-model-version ACTIVE
```

Die Operation gibt eine Antwort mit den Details des Modells zurück. Die Antwort enthält ein `assetModelStatus` Objekt mit der folgenden Struktur.

```
{
  ...
  "assetModelName": "string",
  "assetModelProperties": [ ... ],
  ...,
  "assetModelVersion": "string"
}
```

Benutzerdefinierte zusammengesetzte Modelle (Komponenten)

Wenn Sie eine besonders komplexe Industrieanlage modellieren, z. B. eine komplizierte Maschine mit vielen Teilen, kann es zu einer Herausforderung werden, Ihre Anlagenmodelle zu organisieren und zu warten.

In solchen Fällen können Sie Ihren vorhandenen Anlagen- und Komponentenmodellen benutzerdefinierte Verbundmodelle oder Komponenten hinzufügen, wenn Sie die Konsole verwenden. Diese helfen Ihnen, organisiert zu bleiben, indem sie verwandte Eigenschaften gruppieren und Unterkomponentendefinitionen wiederverwenden.

Es gibt zwei Arten von benutzerdefinierten Verbundmodellen:

- Benutzerdefinierte Verbundmodelle definieren eine Reihe von gruppierten Eigenschaften, die für das Asset- oder Komponentenmodell gelten, zu dem das benutzerdefinierte Verbundmodell gehört. Sie verwenden sie, um verwandte Eigenschaften zu gruppieren. Sie bestehen aus einem Namen, einer Beschreibung und einer Reihe von Asset-Modelleigenschaften. Sie sind nicht wiederverwendbar.
- Component-model-basedbenutzerdefinierte Verbundmodelle verweisen auf ein Komponentenmodell, das Sie in Ihr Asset- oder Komponentenmodell aufnehmen möchten. Sie verwenden sie, um Standard-Unterbaugruppen in Ihr Modell aufzunehmen. Sie bestehen aus einem Namen, einer Beschreibung und der ID des Komponentenmodells, auf das sie verweisen. Sie haben keine eigenen Eigenschaften; das referenzierte Komponentenmodell stellt die zugehörigen Eigenschaften allen erstellten Objekten zur Verfügung.

In den folgenden Abschnitten wird veranschaulicht, wie Sie benutzerdefinierte Verbundmodelle in Ihren Entwürfen verwenden können.

Themen

- [Integrierte benutzerdefinierte Verbundmodelle](#)
- [Component-model-based benutzerdefinierte Verbundmodelle](#)
- [Verwenden Sie Pfade, um auf benutzerdefinierte Eigenschaften von Verbundmodellen zu verweisen](#)

Integrierte benutzerdefinierte Verbundmodelle

Benutzerdefinierte zusammengesetzte Inline-Modelle bieten eine Möglichkeit, Ihr Asset-Modell zu organisieren, indem verwandte Eigenschaften gruppiert werden.

Nehmen wir zum Beispiel an, Sie möchten ein Roboter-Asset modellieren. Der Roboter umfasst einen Servomotor, eine Stromversorgung und eine Batterie. Jeder dieser Bestandteile hat seine eigenen Eigenschaften, die Sie in das Modell aufnehmen möchten. Sie könnten ein Objektmodell mit dem Namen definieren `robot_model`, das Eigenschaften wie die folgenden hat.

- `robot_model`
 - `servo_status` (Ganzzahl)
 - `servo_position` (doppelt)
 - `powersupply_status` (Ganzzahl)
 - `powersupply_temperature` (doppelt)
 - `battery_status` (Ganzzahl)
 - `battery_charge` (doppelt)

In einigen Fällen kann es jedoch viele Unterbaugruppen geben, oder die Unterbaugruppen selbst können viele Eigenschaften haben. In diesen Fällen sind möglicherweise so viele Eigenschaften vorhanden, dass es schwierig wird, sie zu referenzieren und sie in einer einzigen flachen Liste im Modellstamm zu verwalten, wie im vorherigen Beispiel.

Um mit solchen Situationen umzugehen, können Sie ein benutzerdefiniertes Verbundmodell verwenden, um Eigenschaften zu gruppieren. Ein benutzerdefiniertes Verbundmodell ist ein benutzerdefiniertes Verbundmodell, das seine eigenen Eigenschaften definiert. Sie könnten Ihren Roboter beispielsweise wie folgt modellieren.

- `robot_model`

- `servo`
 - `status(Ganzzahl)`
 - `position(doppelt)`
- `powersupply`
 - `status(Ganzzahl)`
 - `temperature (doppelt)`
- `battery`
 - `status(Ganzzahl)`
 - `charge(doppelt)`

Im vorherigen Beispiel `battery` sind `servo` `powersupply`, und die Namen von benutzerdefinierten Verbundwerkstoffmodellen, die innerhalb des `robot_model` Asset-Modells definiert sind. Jedes dieser zusammengesetzten Modelle definiert dann seine eigenen Eigenschaften.

Note

In diesem Fall definiert jedes benutzerdefinierte Verbundmodell seine eigenen Eigenschaften, sodass alle Eigenschaften Teil des Asset-Modells selbst sind (`robot_model` in diesem Fall). Diese Eigenschaften werden nicht mit anderen Asset- oder Komponentenmodellen gemeinsam genutzt. Wenn Sie beispielsweise ein anderes Asset-Modell erstellt haben, für das auch ein benutzerdefiniertes Inline-Verbundmodell aufgerufen wurde `servo`, `robot_model` hätte eine Änderung an der `servo` Innenseite keinen Einfluss auf die `servo` Definition des anderen Asset-Modells.

Wenn Sie eine solche gemeinsame Nutzung implementieren möchten (z. B. um nur eine Definition für ein Servo zu haben, die alle Ihre Asset-Modelle gemeinsam nutzen können), würden Sie stattdessen ein Komponentenmodell dafür erstellen und dann komponentenmodellbasierte Verbundmodelle erstellen, die darauf verweisen. Einzelheiten finden Sie im folgenden Abschnitt.

Informationen zum Erstellen benutzerdefinierter Inline-Verbundmodelle finden Sie unter [Erstellen Sie benutzerdefinierte Verbundmodelle \(Komponenten\)](#).

Component-model-based benutzerdefinierte Verbundmodelle

Sie können ein Komponentenmodell erstellen, AWS IoT SiteWise um eine wiederverwendbare Standardunterbaugruppe zu definieren. Sobald Sie ein Komponentenmodell erstellt haben, können Sie Referenzen darauf in Ihren anderen Objektmodellen und Komponentenmodellen hinzufügen. Dazu fügen Sie jedem Modell, in dem Sie die Komponente referenzieren möchten, ein auf einem Komponentenmodell basierendes benutzerdefiniertes Verbundmodell hinzu. Sie können Verweise auf Ihre Komponente aus vielen Modellen oder mehrfach innerhalb desselben Modells hinzufügen.

Auf diese Weise können Sie vermeiden, dass dieselben Definitionen modellübergreifend dupliziert werden. Es vereinfacht auch die Verwaltung Ihrer Modelle, da alle Änderungen, die Sie an einem Komponentenmodell vornehmen, in allen Asset-Modellen, die es verwenden, übernommen werden.

Nehmen wir beispielsweise an, dass Ihre Industrieanlage über viele Arten von Geräten verfügt, die alle dieselbe Art von Servomotor verwenden. Einige von ihnen haben viele Servomotoren in einem einzigen Gerät. Sie erstellen für jeden Gerätetyp ein Anlagenmodell, möchten aber nicht `servo` jedes Mal die Definition duplizieren. Sie möchten es nur einmal modellieren und es in Ihren verschiedenen Anlagenmodellen verwenden. Wenn Sie später eine Änderung an der Definition von `vornehmenseervo` vornehmen, wird sie für alle Ihre Modelle und Anlagen aktualisiert.

Um den Roboter aus dem vorherigen Beispiel auf diese Weise zu modellieren, könnten Sie Servomotoren, Stromversorgungen und Batterien wie folgt als Komponentenmodelle definieren.

- `servo_component_model`
 - `status(Ganzzahl)`
 - `position(doppelt)`

- `powersupply_component_model`
 - `status(Ganzzahl)`
 - `temperature (doppelt)`

- `battery__component_model`
 - `status(Ganzzahl)`
 - `charge(doppelt)`

Anschließend könnten Sie Asset-Modelle definieren, die z. `robot_model` B. auf diese Komponenten verweisen. Mehrere Anlagenmodelle können auf dasselbe Komponentenmodell verweisen. Sie können dasselbe Komponentenmodell auch mehrfach in einem Anlagenmodell referenzieren, z. B. wenn Ihr Roboter mehrere Servomotoren enthält.

- `robot_model`
 - `servo1(Referenz: servo_component_model`
 - `servo2(Referenz:servo_component_model )`
 - `servo3(Referenz:servo_component_model )`
 - `powersupply (Referenz:powersupply_component_model )`
 - `battery(Referenz:battery_component_model )`

Informationen zum Erstellen von Komponentenmodellen finden Sie unter [Erstellen Sie Komponentenmodelle](#).

Informationen darüber, wie Sie Ihre Komponentenmodelle in anderen Modellen referenzieren können, finden Sie unter [Erstellen Sie benutzerdefinierte Verbundmodelle \(Komponenten\)](#).

Verwenden Sie Pfade, um auf benutzerdefinierte Eigenschaften von Verbundmodellen zu verweisen

Wenn Sie eine Eigenschaft in einem Objektmodell, Komponentenmodell oder benutzerdefinierten Verbundmodell erstellen, können Sie sie von anderen Eigenschaften aus referenzieren, die ihren Wert verwenden, z. B. [Transformationen](#) und [Metriken](#).

AWS IoT SiteWise bietet Ihnen verschiedene Möglichkeiten, auf Ihre Immobilie zu verweisen. Am einfachsten ist es oft, die zugehörige Eigenschafts-ID zu verwenden. Wenn sich die Eigenschaft, auf die Sie verweisen möchten, jedoch in einem benutzerdefinierten Verbundmodell befindet, ist es möglicherweise sinnvoller, sie stattdessen über einen Pfad zu referenzieren.

Ein Pfad ist eine geordnete Abfolge von Pfadsegmenten, die eine Eigenschaft in Bezug auf ihre Position innerhalb der verschachtelten Verbundmodelle innerhalb eines Objektmodells und eines Verbundmodells spezifiziert.

Ermitteln Sie Eigenschaftspfade

Den Pfad einer Eigenschaft können Sie ihrem `path` Feld entnehmen [AssetModelProperty](#).

Nehmen wir beispielsweise an, Sie haben ein Anlagenmodell `robot_model`, das ein benutzerdefiniertes zusammengesetztes Modell enthält `servo`, das über eine Eigenschaft verfügt `position`. Wenn Sie diese [DescribeAssetModelCompositeModel](#) Option aufrufen `servo`, würde die `position` Eigenschaft ein `path` Feld auflisten, das wie folgt aussieht:

```
"path": [
  {
    "id": "asset model ID",
    "name": "robot_model"
  },
  {
    "id": "composite model ID",
    "name": "servo"
  },
  {
    "id": "property ID",
    "name": "position"
  }
]
```

Eigenschaftspfade verwenden

Sie können einen Eigenschaftspfad verwenden, wenn Sie eine Eigenschaft definieren, die auf andere Eigenschaften verweist, z. B. eine Transformation oder eine Metrik.

Eine Eigenschaft verwendet eine Variable, um auf eine andere Eigenschaft zu verweisen. Weitere Hinweise zum Arbeiten mit Variablen finden Sie unter [Verwenden Sie Variablen in Formelausdrücken](#).

Wenn Sie eine Variable definieren, die auf eine Eigenschaft verweist, können Sie entweder die ID der Eigenschaft oder ihren Pfad verwenden.

Um eine Variable zu definieren, die den Pfad der referenzierten Eigenschaft verwendet, geben Sie das `propertyPath` Feld ihres Werts an.

Um beispielsweise ein Asset-Modell mit einer Metrik zu definieren, die mithilfe eines Pfads auf eine Eigenschaft verweist, könnten Sie eine Payload wie die folgende übergeben an [CreateAssetModel](#):

```
{
  ...
  "assetModelProperties": [
    {
```

```

...
"type": {
  "metric": {
    ...
    "variables": [
      {
        "name": "variable name",
        "value": {
          "propertyPath": [
            path segments
          ]
        }
      }
    ]
  },
  ...
},
...
],
...
}

```

Schnittstellen für das Asset-Modell

AWS IoT SiteWise Schnittstellen setzen Maßstäbe für verschiedene Anlagemodelle. Sie definieren eine gemeinsame Struktur, die Konsistenz gewährleistet und gleichzeitig Variationen bei der Implementierung zulässt.

Schnittstellen haben dieselbe Struktur wie Asset-Modelle (Eigenschaften, zusammengesetzte Modelle und Hierarchien), aber es ist nicht möglich, Objekte direkt aus ihnen zu erstellen. Stattdessen werden Schnittstellen auf bestehende Anlagenmodelle angewendet, um die Standardisierung sicherzustellen. Komponentenmodelle werden für Schnittstellen nicht unterstützt.

Die Verwendung von Schnittstellen bietet mehrere Vorteile:

- Standardisierte Eigenschaften und Metriken für verschiedene Varianten des Anlagenmodells
- Vereinfachte Metrikdefinitionen auf Schnittstellenebene
- Effizienteres Management komplexer Anlagenhierarchien

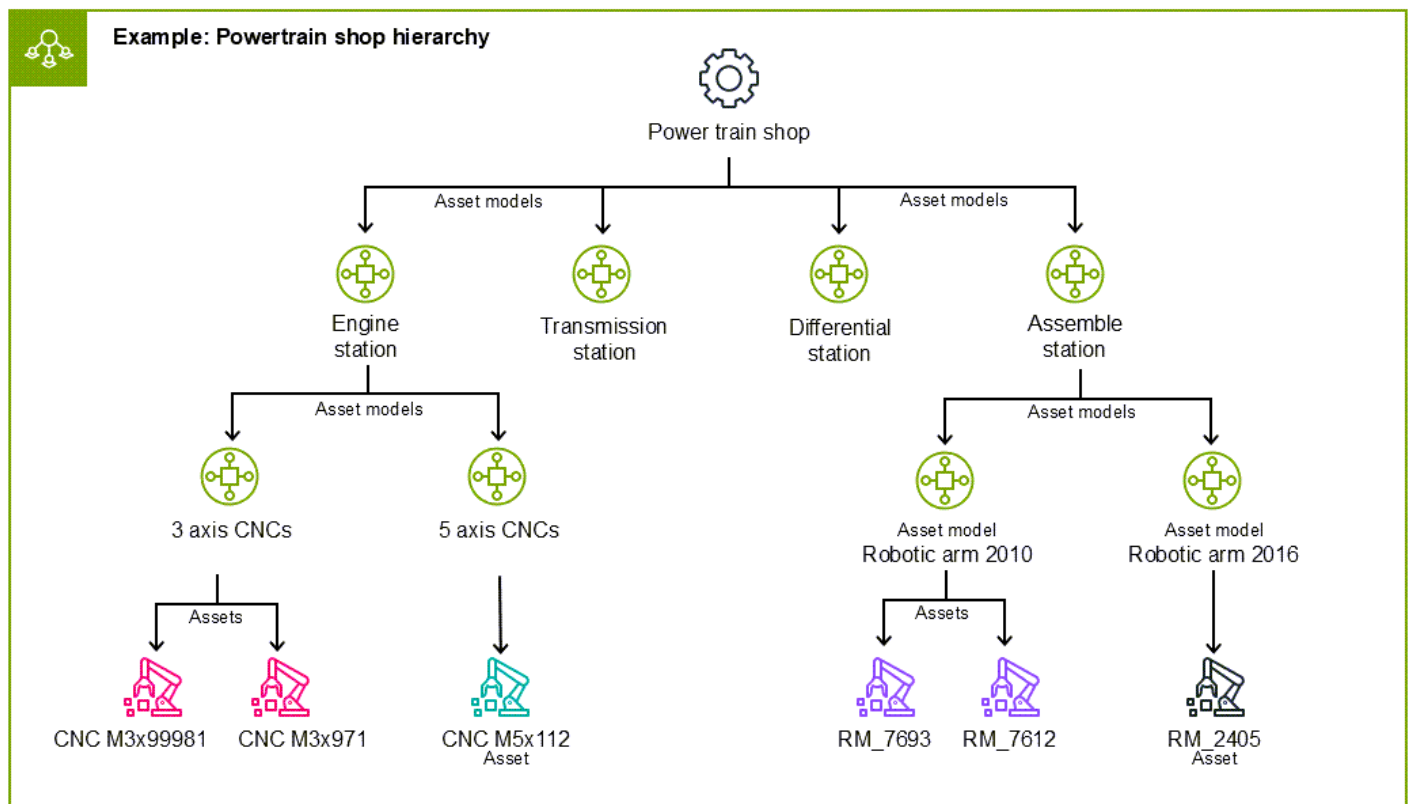
- Unabhängiges Immobilienlebenszyklusmanagement für jede Variante des Anlagensmodells
- Verbesserte teamübergreifende Zusammenarbeit, bei der sich die Betriebsteams auf die Darstellung der physischen Anlagen konzentrieren, während die Datenteams geräteübergreifende Standards festlegen

Wir empfehlen, zuerst Ihre Anlagenmodelle zu erstellen, um Ihre realen Industrieanlagen zu modellieren. Jeder Gerätetyp mit seinen eigenen Eigenschaften kann durch seine eigenen Anlagenmodelle dargestellt werden.

Anwendungsfall zur Standardisierung von Anlagenmodellen

Schnittstellen helfen dabei, Immobilien verschiedener Anlagenmodelle zu standardisieren und gleichzeitig ihre einzigartigen Eigenschaften beizubehalten.

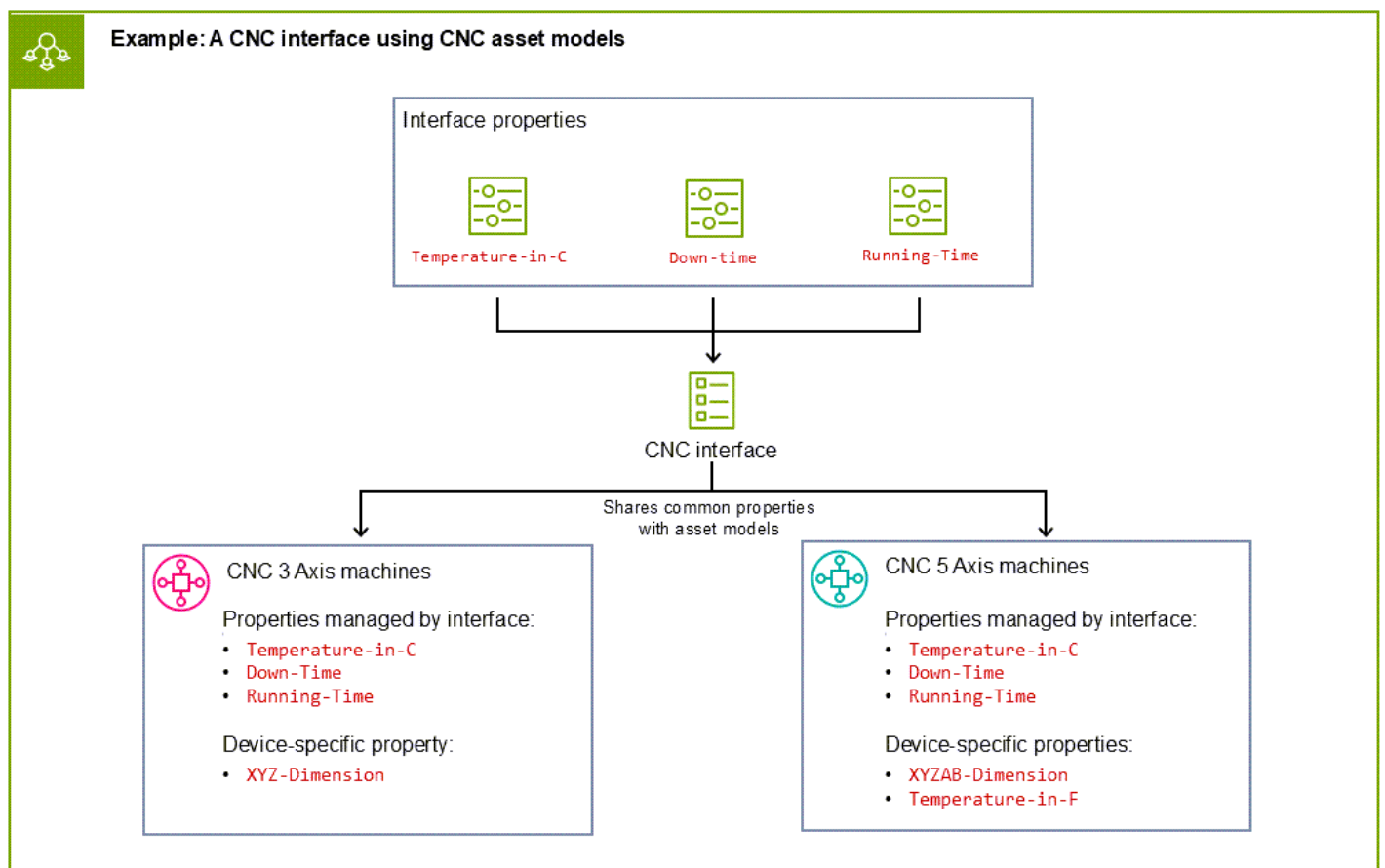
In einer Werkstatt für Antriebssysteme gibt es beispielsweise vier Stationen: Motor, Getriebe, Differential und Montage. Jede Station enthält verschiedene Ausrüstungstypen. Die Motorstation umfasst beispielsweise CNC-Maschinen, die sich jedoch in den Spezifikationen unterscheiden: Einige haben 3 Achsen, andere 5 Achsen.



Mithilfe von Schnittstellen können Sie jedoch Standards für die Gemeinsamkeiten der CNC-Maschinen festlegen. Sie können die wiederholbaren Eigenschaften in einer Schnittstelle verwenden, anstatt für jede Eigenschaft Objektmodelle zu erstellen.

Beispielsweise ist Folgendes möglich:

1. Erstellen Sie separate Anlagenmodelle für jede Kategorie von Maschinentypen. In diesem Beispiel sind das die „CNC 3-Achsen-Maschinen“ und die „CNC 5-Achsen-Maschinen“.
2. Definieren Sie eine Standardschnittstelle mit gemeinsamen Eigenschaften und Metriken. In diesem Beispiel Running-time sind Temperature-in-C Down-time, und alle gemeinsamen Eigenschaften, die für beide CNC-Maschinen gelten.
3. Wenden Sie diese Schnittstelle auf alle Objektmodelle von CNC-Maschinen an und berücksichtigen Sie dabei die gerätespezifischen Eigenschaften der einzelnen Anlagenmodelle.



Sie können Verfügbarkeitsmetriken auch auf Schnittstellenebene definieren. $Avail = avg(Down-time, Running-time)$ Berechnet beispielsweise die Verfügbarkeit auf der Grundlage der Werte für Ausfallzeit und Laufzeit.

Die Verwendung von Schnittstellen vereinfacht die Verwaltung Ihres Anlagenmodells, da konsistente Eigenschaftsdefinitionen und Kennzahlen für die jeweiligen Geräte gewährleistet werden und gleichzeitig die einzigartigen Eigenschaften jedes Maschinentyps beibehalten werden.

Struktur und Komponenten

Schnittstellen beinhalten dieselben Eigenschaftstypen wie Asset-Modelle: Attribute, Messungen, Transformationen und Metriken. Wenn ein Asset-Modell überlagert wird, ordnen Sie vorhandene Eigenschaften ihren Gegenständen in der Benutzeroberfläche zu. Nicht zugeordnete Schnittstelleneigenschaften werden automatisch im Asset-Modell erstellt.

Schnittstellenhierarchien definieren Rollup-Metriken, während Asset-Modellhierarchien Asset-Zuordnungen ermöglichen. Wenn Sie eine Schnittstelle verwenden, ordnet der Service bei der Berechnung von Rollup-Metriken automatisch Asset-Modellhierarchien den Schnittstellenhierarchien zu. Nach dem Anwenden einer Schnittstelle werden die Rollup-Metriken anhand der Schnittstellenhierarchie und nicht anhand der eigenen Hierarchie des Asset-Modells definiert.

Überlegungen

Beachten Sie bei der Arbeit mit Schnittstellen die folgenden Überlegungen:

- Objektmodell- und Schnittstelleneigenschaften können automatisch nach Namen oder manuell zugeordnet werden. Hierarchien werden vom Service bei der Berechnung von Rollup-Metriken automatisch zugeordnet.
- Sie können im verknüpften Asset-Modell keine zusätzlichen Metriken definieren, die Schnittstellenmetriken als Eingaben verwenden.
- Ein Asset-Modell kann nur mit einer Schnittstelle verknüpft werden. Sie können jedoch mehrere Asset-Modelle auf dieselbe Schnittstelle anwenden.
- Alarmer werden in Schnittstellen nicht unterstützt.
- Komponentenmodelle werden für Schnittstellen nicht unterstützt.

Themen

- [Verstehen Sie die Beziehung zwischen Schnittstelle und Asset-Modell](#)

- [Erstellen Sie eine Schnittstelle](#)
- [Wenden Sie eine Schnittstelle auf ein Asset-Modell an](#)
- [Verwaltet Schnittstellen, verknüpfte Asset-Modelle und Eigenschaften](#)
- [Weitere Beispiele für Benutzeroberflächen](#)

Verstehen Sie die Beziehung zwischen Schnittstelle und Asset-Modell

Schnittstellen und Anlagenmodelle arbeiten in einer sich ergänzenden Beziehung zusammen:

Schnittstellen im Vergleich zu Anlagenmodellen

Aspekt	Schnittstellen	Anlagenmodelle
Zweck	Definieren Sie Standards und wenden Sie Konsistenz an	Stellen Sie physische oder logische Ressourcen dar
Schaffung von Vermögenswerten	Assets können nicht direkt erstellt werden	Wird zum Erstellen von Assets verwendet
Eigenschaften	Definieren Sie Standardeigenschaften, die in Modellen implementiert werden müssen	Kann über eine Schnittstelle angewendet werden und über einzigartige Eigenschaften verfügen
Kennzahlen	Definieren Sie Standardberechnungen	Implementieren Sie Schnittstellenmetriken und können zusätzliche Metriken verwenden
Hierarchien	Definieren Sie hierarchische Beziehungen zur Datenberechnung für Rollup-Metriken	Definieren Sie physische hierarchische Beziehungen für Anlagenzuordnungen

Wenn Sie eine Schnittstelle auf ein Anlagenmodell anwenden:

- Das Asset-Modell muss alle in der Schnittstelle definierten Eigenschaften zuordnen.
- Eigenschaftszuordnungen definieren, wie Schnittstelleneigenschaften den Eigenschaften des Objektmodells entsprechen.

- Zugeordnete Objektmodelleigenschaften müssen mit den entsprechenden Schnittstelleneigenschaften synchronisiert bleiben und dürfen nicht so geändert werden, dass es zu Inkonsistenzen zwischen den beiden Eigenschaften kommen würde.
- Nicht zugeordnete Schnittstelleneigenschaften werden automatisch im Asset-Modell erstellt.
- Das Asset-Modell kann zusätzliche Eigenschaften haben, die über die in der Schnittstelle definierten Eigenschaften hinausgehen.
- Das Asset-Modell implementiert Schnittstellenmetriken. Änderungen an Schnittstellenmetriken werden auf alle Asset-Modelle übertragen, die die Schnittstelle verwenden.
- Schnittstellenhierarchien werden für die Berechnung von Rollup-Metriken verwendet. Asset-Modellhierarchien können unabhängig voneinander definiert werden, und der Service ordnet sie bei der Berechnung von Rollup-Metriken automatisch zu.

Diese Beziehung gewährleistet die Standardisierung und ermöglicht gleichzeitig die Flexibilität, die für die Darstellung verschiedener Gerätetypen erforderlich ist.

Standardisieren Sie bestehende Anlagenmodelle

Schnittstellen sind zwar nützlich, wenn es darum geht, neue Anlagenmodelle von Grund auf neu zu entwerfen, aber sie sind ebenso leistungsfähig, um bestehende Anlagenmodelle zu standardisieren, die sich im Laufe der Zeit möglicherweise unabhängig voneinander weiterentwickelt haben.

Bei der Arbeit mit bestehenden Anlagenmodellen können Sie Schnittstellen verwenden, um Kennzahlen und Eigenschaften zu standardisieren:

1. Identifizieren Sie gemeinsame Metriken und Eigenschaften in Ihren bestehenden Anlagenmodellen
2. Erstellen Sie eine Schnittstelle, die diese Standardeigenschaften und -metriken definiert
3. Wenden Sie die Schnittstelle mithilfe von Property Mapping auf Ihre vorhandenen Asset-Modelle an
4. Verwenden Sie Rollup-Metriken, um Daten in Ihrer gesamten Anlagenhierarchie zu aggregieren

Wenn Sie beispielsweise über bestehende Anlagenmodelle für CNC-Maschinen mit unterschiedlichen Eigenschaftsnamen, aber ähnlichen Daten wie `temp_celsius`, `machine_temp` und `temperature_c`, verfügen, können Sie:

1. Erstellen Sie eine CNC-INTERFACE mit einer standardisierten `Temperature-in-C` Eigenschaft

2. Wenden Sie diese Schnittstelle auf jedes CNC-Asset-Modell an und ordnen Sie die vorhandenen Temperatureigenschaften den `Temperature-in-C` Eigenschaften der Schnittstelle zu
3. Definieren Sie in der Benutzeroberfläche Rollup-Metriken, mit denen Statistiken für alle Maschinen berechnet werden (z. B. die Durchschnittstemperatur)

Dieser Ansatz ermöglicht es Ihnen, Ihre bestehenden Anlagenmodelle beizubehalten und gleichzeitig die Vorteile der Standardisierung und der vereinfachten Metrikberechnung zu nutzen.

Hierarchische Beziehungen

Schnittstellenhierarchie

Definiert Beziehungen für die Berechnung und Aggregation von Daten über verschiedene Schnittstellen hinweg. In einer Werkseinstellung könnte eine Schnittstellenhierarchie beispielsweise Schnittstellen zur Temperaturüberwachung auf verschiedenen Ebenen miteinander verbinden, um Durchschnittstemperaturen zu berechnen. Zum Beispiel: Maschine, Produktionslinie und Anlage. Wenn Sie eine Rollup-Metrik wie `defineAverageTemperature` definieren, bestimmt die Schnittstellenhierarchie, wie diese Metrik Daten von niedrigeren Ebenen zu höheren Ebenen aggregiert.

Hierarchie des Asset-Modells

Stellt die tatsächliche physische oder logische Struktur Ihrer Vermögenswerte dar. Beispielsweise kann ein Anlagenmodell einer CNC-Maschine Teil eines Anlagenmodells für eine Produktionslinie sein, das wiederum zu einem Anlagenmodell einer Fabrik gehört. Diese Hierarchie spiegelt reale Beziehungen wider und hilft dabei, Anlagen so zu organisieren, dass sie ihrer physischen Anordnung oder Geschäftsstruktur entsprechen. In Kombination mit Schnittstellenhierarchien helfen Anlagenmodellhierarchien dem System zu verstehen, welche Anlagen in die Rollup-Berechnungen einbezogen werden sollten.

Diese beiden Hierarchietypen arbeiten zusammen: Schnittstellenhierarchien definieren, wie aggregierte Kennzahlen berechnet werden, während Anlagenmodellhierarchien definieren, welche spezifischen Vermögenswerte in diese Berechnungen einbezogen werden sollen.

Schnittstellenmetriken und Rollup-Berechnungen

Schnittstellen zeichnen sich durch die Definition standardisierter Kennzahlen aus, die auf verschiedene Anlagemodelle angewendet werden können. Dies ist besonders nützlich für Rollup-Metriken, die Daten aus mehreren Anlagen aggregieren.

Wenn Sie Metriken in einer Schnittstelle definieren, werden sie automatisch auf alle Asset-Modelle angewendet, die die Schnittstelle implementieren. Die Metriken können auf in der Schnittstelle definierte Eigenschaften verweisen, Aggregationsfunktionen zur Berechnung von Statistiken für alle Anlagen verwenden und konsistente Berechnungen für alle implementierenden Anlagenmodelle sicherstellen. Sie können beispielsweise eine Verfügbarkeitsmetrik in einer Schnittstelle definieren, die das Verhältnis von Laufzeit zu Gesamtzeit berechnet:

```
{
  "name": "Availability",
  "dataType": "DOUBLE",
  "type": {
    "metric": {
      "expression": "Running-time / (Running-time + Down-time) * 100",
      "variables": [
        {
          "name": "Running-time",
          "value": {
            "propertyId": "${Running-time}"
          }
        },
        {
          "name": "Down-time",
          "value": {
            "propertyId": "${Down-time}"
          }
        }
      ],
      "window": {
        "tumbling": {
          "interval": "1h"
        }
      }
    }
  },
  "unit": "Percent"
}
```

Wenn diese Schnittstelle auf mehrere Anlagenmodelle angewendet wird, wird die Verfügbarkeitsmetrik für alle Modelle konsistent berechnet, auch wenn sich die zugrundeliegenden Eigenschaftsnamen unterscheiden (dank der Eigenschaftenzuweisung).

Weitere Informationen zur Definition von Metriken und zur Verwendung von Aggregationsfunktionen finden Sie unter [Aggregieren Sie Daten aus Immobilien und anderen Vermögenswerten \(Metriken\)](#).

Metriken mit Schnittstellen zusammenfassen

Schnittstellen können auch Rollup-Metriken definieren, mit denen Daten über Vermögenswerte hinweg in einer Hierarchie zusammengefasst werden. Wenn Sie eine Hierarchie in einer Schnittstelle definieren und sie auf ein Asset-Modell anwenden, können Sie Metriken erstellen, die Daten aus untergeordneten Anlagen aggregieren.

Sie können beispielsweise eine Metrik definieren, die die Durchschnittstemperatur aller CNC-Maschinen in einer Fabrik berechnet:

```
{
  "name": "AverageTemperature",
  "dataType": "DOUBLE",
  "type": {
    "metric": {
      "expression": "avg(Temperature-in-C)",
      "variables": [
        {
          "name": "Temperature-in-C",
          "value": {
            "propertyId": "${Temperature-in-C}",
            "hierarchyId": "${CNC-machines}"
          }
        }
      ],
      "window": {
        "tumbling": {
          "interval": "1h"
        }
      }
    },
    "unit": "Celsius"
  }
}
```

Diese Metrik verwendet die `avg()` Aggregationsfunktion, um die Durchschnittstemperatur aller CNC-Maschinen in der Hierarchie zu berechnen. Der `hierarchyId` Parameter gibt an, welche Hierarchie für die Aggregation verwendet werden soll.

Wenn diese Schnittstelle auf ein Asset-Modell angewendet wird, aggregiert die Rollup-Metrik automatisch Daten aus allen untergeordneten Anlagen, die der Hierarchiezuzuordnung entsprechen.

Erstellen Sie eine Schnittstelle

Sie können Schnittstellen entweder mit der AWS IoT SiteWise Konsole oder mit dem erstellenAWS CLI.

Console

1. Navigieren Sie zur [AWS IoT SiteWiseKonsole](#) und wählen Sie im Navigationsbereich Modelle aus.
2. Wählen Sie „Schnittstelle erstellen“.
3. Geben Sie einen eindeutigen Namen und optional eine Beschreibung für Ihre Schnittstelle ein. Sie können optional auch eine externe ID Ihrer Wahl hinzufügen.
4. Fügen Sie Ihrer Schnittstelle Eigenschaften hinzu. Sie können Attribute, Messungen, Transformationen und Metriken hinzufügen, genau wie bei Asset-Modellen. Weitere Informationen finden Sie unter [Erstellen Sie ein Asset-Modell \(Konsole\)](#).
5. Wählen Sie Schnittstelle erstellen, um die Schnittstelle zu erstellen.
6. Wenn Sie über Hierarchien verfügen, um Eltern-Kind-Beziehungen zwischen Schnittstellen zu definieren, wählen Sie Hierarchie hinzufügen und geben Sie die entsprechenden Details ein.

AWS CLI

Um eine Schnittstelle zu erstellen, verwenden Sie die `CreateAssetModel` Operation, bei der der `assetModelType` Parameter wie folgt gesetzt ist: `INTERFACE`

```
aws iotsitewise create-asset-model \
  --asset-model-name "CNC-INTERFACE" \
  --asset-model-description "Standard interface for CNC machines" \
  --asset-model-type "INTERFACE" \
  --asset-model-properties '[
    {
      "name": "Temperature-in-C",
      "dataType": "DOUBLE",
      "type": {
        "measurement": {}
      },
      "unit": "Celsius"
    }
  ]'
```

```
    },
    {
      "name": "Down-time",
      "dataType": "DOUBLE",
      "type": {
        "measurement": {}
      },
      "unit": "Minutes"
    },
    {
      "name": "Running-time",
      "dataType": "DOUBLE",
      "type": {
        "measurement": {}
      },
      "unit": "Minutes"
    },
    {
      "name": "Availability",
      "dataType": "DOUBLE",
      "type": {
        "metric": {
          "expression": "Running-time / (Running-time + Down-time) * 100",
          "variables": [
            {
              "name": "Running-time",
              "value": {
                "propertyId": "${Running-time}"
              }
            },
            {
              "name": "Down-time",
              "value": {
                "propertyId": "${Down-time}"
              }
            }
          ]
        },
        "window": {
          "tumbling": {
            "interval": "1h"
          }
        }
      }
    }
  ],
}
```

```
    "unit": "Percent"
  }
]'
```

Wenden Sie eine Schnittstelle auf ein Asset-Modell an

Wenn Sie eine Schnittstelle auf ein Anlagenmodell anwenden, ordnen Sie die Eigenschaften und Hierarchien des Anlagenmodells den entsprechenden Schnittstellen zu. Für Schnittstelleneigenschaften ohne Zuordnung werden die entsprechenden Eigenschaften automatisch im Asset-Modell erstellt. Nach dem Verknüpfen verhindert der Service Änderungen am Asset-Modell, die gegen Schnittstellenstandards verstoßen würden.

Sie können einer Schnittstelle jeweils ein Asset-Modell hinzufügen. Es können jedoch mehrere Asset-Modelle mit einer einzigen Schnittstelle verknüpft werden.

Console

1. Navigieren Sie zur [AWS IoT SiteWiseKonsole](#) und wählen Sie im Navigationsbereich Modelle aus.
2. Wählen Sie das Asset-Modell aus, auf das Sie eine Schnittstelle anwenden möchten.
3. Wählen Sie im Abschnitt Asset-Modelle verknüpfen die Option Asset-Modell verknüpfen aus. Daraufhin wird die Seite Link-Schnittstelle geöffnet.
4. Wählen Sie im Abschnitt Asset-Modelle und Schnittstellen ein Asset-Modell aus dem Drop-down-Menü Select a model to link aus.
5. Ordnen Sie im Abschnitt Eigenschaftszuordnungen jede Schnittstelleneigenschaft einer vorhandenen Objektmodelleigenschaft zu oder erstellen Sie eine neue Eigenschaft. AWS IoT SiteWiseverknüpft automatisch Eigenschaften mit übereinstimmenden Namen im Objektmodell und in der Benutzeroberfläche.
6. Überprüfen Sie die Eigenschaftszuordnungen und wählen Sie Schnittstelle verknüpfen.

AWS CLI

Gehen Sie wie folgt vor, um eine Schnittstelle auf ein Asset-Modell anzuwenden:
`PutAssetModelInterfaceRelationship`

```
aws iotsitewise put-asset-model-interface-relationship \
```

```
--asset-model-id "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \
--interface-asset-model-id "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE" \
--property-mapping-configuration '{
  "createMissingProperty": true,
  "matchByPropertyName": true,
  "overrides": [
    {
      "assetModelPropertyId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
      "interfaceAssetModelPropertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE"
    }
  ]
}'
```

Verwenden Sie die folgende `DescribeAssetModelInterfaceRelationship` Operation, um Informationen über eine Schnittstellenbeziehung abzurufen:

```
aws iotsitewise describe-asset-model-interface-relationship \
--asset-model-id "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \
--interface-asset-model-id "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"
```

Verwenden Sie die folgende `ListInterfaceRelationships` Operation, um alle Asset-Modelle aufzulisten, auf die eine bestimmte Schnittstelle angewendet wurde:

```
aws iotsitewise list-interface-relationships \
--interface-asset-model-id "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE" \
--max-results 10
```

Verwenden Sie die folgende `DeleteAssetModelInterfaceRelationship` Operation, um eine Schnittstellenbeziehung zu löschen:

```
aws iotsitewise delete-asset-model-interface-relationship \
--asset-model-id "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \
--interface-asset-model-id "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"
```

Verwaltet Schnittstellen, verknüpfte Asset-Modelle und Eigenschaften

Nachdem Sie Schnittstellen erstellt und sie mit Objektmodellen verknüpft haben, können Sie Beziehungen verwalten und Schnittstellen über die Konsole bearbeiten und löschen oder AWS CLI.

Ändern Sie die Beziehung zwischen einer Schnittstelle und einem Asset-Modell

Um die Beziehung einer Schnittstelle zu einem Asset-Modell zu ändern, gehen Sie entweder in der AWS IoT SiteWise Konsole oder über die Konsole wie folgt vor AWS CLI:

Console

1. Navigieren Sie zur [AWS IoT SiteWiseKonsole](#) und wählen Sie im Navigationsbereich Modelle aus.
2. Wählen Sie die Schnittstelle aus, die Sie ändern möchten.
3. Wählen Sie das Asset-Modell aus, das Sie ändern und bearbeiten möchten.

Sie können den [Wenden Sie eine Schnittstelle auf ein Asset-Modell an](#) Anweisungen folgen, um ein anderes Asset-Modell zu verknüpfen.

4. Wählen Sie Benutzeroberfläche anwenden, um Ihre Änderungen zu speichern.

AWS CLI

Verwenden Sie die `PutAssetModelInterfaceRelationship` Aktion, um eine Beziehung zwischen Schnittstelle und Anlagenmodell zu bearbeiten. Ersetzen Sie *your-asset-model-id* und *your-interface-asset-model-id* durch Ihre eigenen Werte. Weitere Informationen finden Sie unter [PutAssetModelInterfaceRelationship](#) in der AWS IoT SiteWise-API-Referenz.

```
aws iotsitewise put-asset-model-interface-relationship \
  --asset-model-id your-asset-model-id \
  --interface-asset-model-id your-interface-asset-model-id
```

Ändern Sie die Zuordnung einer Schnittstelleneigenschaft

Gehen Sie entweder in der AWS IoT SiteWise Konsole oder über die Konsole wie folgt vor, um die Eigenschaft einer Schnittstelle zu ändern AWS CLI:

Console

1. Navigieren Sie zur [AWS IoT SiteWiseKonsole](#) und wählen Sie im Navigationsbereich Modelle aus.
2. Wählen Sie die Schnittstelle aus, für die Sie Eigenschaftszuordnungen ändern möchten. Die Seite Eigenschaftszuordnungen bearbeiten wird angezeigt.

3. Filtern Sie im Abschnitt Eigenschaftszuordnungen die Liste, um die entsprechenden Eigenschaftszuordnungen zu finden.
4. Ändern Sie die Eigenschaften mithilfe der Spalte Modelleigenschaft.

AWS CLI

Verwenden Sie die `PutAssetModelInterfaceRelationship` Aktion, um eine Beziehung zwischen Schnittstelle und Asset-Modell zu bearbeiten. Ersetzen Sie *your-asset-model-id* und *your-interface-asset-model-id* durch Ihre eigenen Werte. Weitere Informationen finden Sie unter [PutAssetModelInterfaceRelationship](#) in der AWS IoT SiteWise-API-Referenz.

```
aws iotsitewise put-asset-model-interface-relationship \  
  --asset-model-id your-asset-model-id \  
  --interface-asset-model-id your-interface-asset-model-id \  
  --interface-id your-interface-id \  
  --model-id your-model-id \  
  --role-arn your-role-arn \  
  --tags your-tags \  
  --dry-run your-dry-run
```

Listet Schnittstellen auf, die mit einem Asset-Modell verknüpft sind

Um eine Liste der Schnittstellen zu erhalten, die auf ein Asset-Modell angewendet wurden, gehen Sie entweder in der AWS IoT SiteWise Konsole oder über AWS CLI:

Console

1. Navigieren Sie zur [AWS IoT SiteWiseKonsole](#) und wählen Sie im Navigationsbereich Modelle aus.
2. Wählen Sie im Abschnitt Modelle das entsprechende Asset-Modell oder die entsprechende Schnittstelle aus. Sie können eine Liste der verwendeten Schnittstellen oder der verknüpften Asset-Modelle auf der entsprechenden Detailseite des Modells einsehen.
 - Informationen zur Anzeige einer bestimmten Schnittstelle finden Sie im Abschnitt Verknüpfte Asset-Modelle.
 - Informationen zur Anzeige eines bestimmten Asset-Modells finden Sie im Abschnitt Angewandte Schnittstellen.

AWS CLI

Um Schnittstellen aufzulisten, können Sie die `ListInterfaceRelationships` Operation verwenden. Ersetzen Sie *your-interface-asset-model-id* durch Ihren eigenen Wert.

Weitere Informationen finden Sie unter [ListInterfaceRelationships](#) in der AWS IoT SiteWise-API-Referenz.

```
aws iotsitewise list-interface-relationships \
  --interface-asset-model-id your-interface-asset-model-id \
  [--next-token your-next-token] \
  [--max-results 20]
```

Sehen Sie sich die Details einer Beziehung zwischen Schnittstelle und Anlagenmodell an

Um die Details einer Schnittstelle zu sehen, die auf ein Asset-Modell angewendet wurde, gehen Sie entweder in der AWS IoT SiteWise Konsole oder über die Konsole wie folgt vor AWS CLI:

Console

Sehen Sie sich die Details der verwendeten Schnittstellen und verknüpften Asset-Modelle an.

1. Navigieren Sie zur [AWS IoT SiteWiseKonsole](#) und wählen Sie im Navigationsbereich Modelle aus.
2. Suchen Sie im Bereich Modelle nach dem entsprechenden Asset-Modell oder der entsprechenden Schnittstelle. Wählen Sie den Namen des Modells oder der Schnittstelle aus, um eine Seite mit weiteren Details zu öffnen.

AWS CLI

Verwenden Sie die `DescribeAssetModelInterfaceRelationship` Aktion, um Schnittstellendetails für eine Beziehung zwischen Schnittstelle und Anlagenmodell anzuzeigen. Ersetzen Sie *your-asset-model-id* und *your-interface-asset-model-id* durch Ihre eigenen Werte. Weitere Informationen finden Sie unter [DescribeAssetModelInterfaceRelationship](#) in der AWS IoT SiteWise-API-Referenz.

```
aws iotsitewise describe-asset-model-interface-relationship \
  --asset-model-id your-asset-model-id \
  --interface-asset-model-id your-interface-asset-model-id
```

Entfernen Sie eine Schnittstelle, die auf ein Asset-Modell angewendet wurde

Um eine Schnittstelle zu entfernen, die auf ein Asset-Modell angewendet wurde, gehen Sie entweder in der AWS IoT SiteWise Konsole oder über die Konsole wie folgt vor AWS CLI:

Console

Wir empfehlen, eine Schnittstelle über das Asset-Modell zu entfernen. Sie können eine Schnittstelle auch über die Seite einer bestimmten Schnittstelle löschen oder die Verknüpfung zu einer Schnittstelle aufheben.

1. Navigieren Sie zur [AWS IoT SiteWiseKonsole](#) und wählen Sie im Navigationsbereich Modelle aus.
2. Wählen Sie das entsprechende Asset-Modell aus, aus dem Sie die Schnittstellenbeziehung entfernen möchten.
3. Wählen Sie „Asset-Modell trennen“.

AWS CLI

Um eine Schnittstellenbeziehung aus einem Asset-Modell zu entfernen, können Sie die `DeleteAssetModelInterfaceRelationship` Aktion verwenden. Ersetzen Sie *your-interface-asset-model-id* durch Ihren eigenen Wert. Weitere Informationen finden Sie unter [DeleteAssetModelInterfaceRelationship](#) in der AWS IoT SiteWise-API-Referenz.

```
aws iotsitewise delete-asset-model-interface-relationship \  
  --asset-model-id your-asset-model-id \  
  --interface-asset-model-id your-interface-asset-model-id
```

Weitere Beispiele für Benutzeroberflächen

Hier sind weitere Beispiele dafür, wie Schnittstellen in verschiedenen industriellen Szenarien verwendet werden können:

Anlagen zur Energieerzeugung

Ein Energieerzeugungsunternehmen kann Schnittstellen verwenden, um Kennzahlen für verschiedene Arten von Erzeugungsanlagen zu standardisieren:

```
{
```

```

"assetModelName": "GENERATOR-INTERFACE",
"assetModelDescription": "Standard interface for power generators",
"assetModelType": "INTERFACE",
"assetModelProperties": [
  {
    "name": "ActivePower",
    "dataType": "DOUBLE",
    "type": { "measurement": {} },
    "unit": "MW"
  },
  {
    "name": "ReactivePower",
    "dataType": "DOUBLE",
    "type": { "measurement": {} },
    "unit": "MVAR"
  },
  {
    "name": "Frequency",
    "dataType": "DOUBLE",
    "type": { "measurement": {} },
    "unit": "Hz"
  },
  {
    "name": "PowerFactor",
    "dataType": "DOUBLE",
    "type": {
      "metric": {
        "expression": "cos(atan(ReactivePower / ActivePower))",
        "variables": [
          {
            "name": "ActivePower",
            "value": { "propertyId": "${ActivePower}" }
          },
          {
            "name": "ReactivePower",
            "value": { "propertyId": "${ReactivePower}" }
          }
        ]
      },
      "window": { "tumbling": { "interval": "5m" } }
    }
  },
  {
    "unit": "None"
  }
]

```

```
}
```

Diese Schnittstelle kann auf verschiedene Generatoranlagenmodelle (Gasturbinen, Dampfturbinen, Windturbinen) angewendet werden, um konsistente Leistungskennzahlen für die gesamte Flotte sicherzustellen.

Anlagen zur Wasseraufbereitung

Ein Wasserversorgungsunternehmen kann Schnittstellen verwenden, um die Überwachung aller Behandlungsanlagen zu standardisieren:

```
{
  "assetModelName": "WATER-QUALITY-INTERFACE",
  "assetModelDescription": "Standard interface for water quality monitoring",
  "assetModelType": "INTERFACE",
  "assetModelProperties": [
    {
      "name": "pH",
      "dataType": "DOUBLE",
      "type": { "measurement": {} },
      "unit": "pH"
    },
    {
      "name": "Turbidity",
      "dataType": "DOUBLE",
      "type": { "measurement": {} },
      "unit": "NTU"
    },
    {
      "name": "DissolvedOxygen",
      "dataType": "DOUBLE",
      "type": { "measurement": {} },
      "unit": "mg/L"
    },
    {
      "name": "QualityIndex",
      "dataType": "DOUBLE",
      "type": {
        "metric": {
          "expression": "(pH >= 6.5 && pH <= 8.5 ? 100 : 50) * (Turbidity < 1 ? 1 : 0.8) * (DissolvedOxygen > 5 ? 1 : 0.7)",
          "variables": [
            {
```

```

        "name": "pH",
        "value": { "propertyId": "${pH}" }
    },
    {
        "name": "Turbidity",
        "value": { "propertyId": "${Turbidity}" }
    },
    {
        "name": "DissolvedOxygen",
        "value": { "propertyId": "${DissolvedOxygen}" }
    }
],
"window": { "tumbling": { "interval": "1h" } }
}
},
"unit": "Score"
}
]
}

```

Diese Schnittstelle stellt sicher, dass die Wasserqualität in allen Aufbereitungsanlagen einheitlich gemessen wird, unabhängig von der jeweiligen Gerätekonfiguration.

Hierarchische Schnittstellen

Schnittstellen können hierarchisch organisiert werden, um aggregierte Kennzahlen auf verschiedenen Ebenen Ihres Betriebs zu unterstützen:

1. Equipment-level Schnittstelle (zum Beispiel PUMP-INTERFACE)
 - Eigenschaften: Durchflussrate, Druck, Stromverbrauch, Vibration
 - Kennzahlen: Effizienz, Gesundheitswert
2. Process-level Schnittstelle (zum Beispiel PUMPING-STATION-INTERFACE)
 - Eigenschaften: Gesamtdurchfluss, Durchschnittsdruck, Gesamtleistung
 - Kennzahlen: Effizienz der Station, Betriebskosten pro Volumen
 - Hierarchie: Enthält PUMP-INTERFACE Ausrüstung
3. Facility-level Schnittstelle (zum Beispiel WATER-FACILITY-INTERFACE)
 - Eigenschaften: Durchsatz der Anlage, Energieverbrauch, Chemikalienverbrauch
 - Kennzahlen: Anlageneffizienz, Kosten pro Volumeneinheit, CO₂-Fußabdruck

- Hierarchie: Enthält PUMPING-STATION-INTERFACE Prozesse

Dieser hierarchische Ansatz ermöglicht die Berechnung von Kennzahlen auf jeder Ebene und gewährleistet gleichzeitig die Konsistenz in Ihrem gesamten Betrieb.

Einrichten AWS IoT SiteWise Objekt-IDs

AWS IoT SiteWise definiert verschiedene Typen persistenter Objekte, wie z. B. Vermögenswerte, Objektmodelle, Eigenschaften und Hierarchien. All diese Objekte verfügen über eindeutige Kennungen, mit denen Sie sie abrufen, aktualisieren und löschen können.

AWS IoT SiteWise bietet Kunden verschiedene Optionen für die ID-Erstellung. AWS IoT SiteWise generiert standardmäßig eine für Sie zum Zeitpunkt der Objekterstellung. Benutzer können Ihren Objekten auch ihre eigenen IDs zuweisen.

Themen

- [Arbeiten Sie mit Objekt-UUIDs](#)
- [Verwenden Sie externe IDs](#)

Arbeiten Sie mit Objekt-UUIDs

Jedes persistente Objekt AWS IoT SiteWise hat eine [UUID](#), um es zu identifizieren. Asset-Modelle haben beispielsweise eine Asset-Modell-ID, Assets haben eine Asset-ID und so weiter. Diese ID wird bei der Erstellung des Objekts zugewiesen und bleibt während der gesamten Lebensdauer des Objekts unverändert.

Wenn Sie ein neues Objekt erstellen, AWS IoT SiteWise generiert standardmäßig eine eindeutige ID für Sie. Sie können bei der Erstellung auch Ihre eigene ID im UUID-Format angeben.

Note

UUIDs müssen innerhalb der AWS Region, in der sie erstellt wurden, und für denselben Objekttyp global eindeutig sein. Wenn AWS IoT SiteWise automatisch eine ID für Sie generiert wird, ist sie immer einzigartig. Wenn Sie Ihre eigene ID wählen, stellen Sie sicher, dass sie eindeutig ist.

Wenn Sie beispielsweise durch einen Anruf ein neues Asset-Modell erstellen [CreateAssetModel](#), können Sie Ihre eigene UUID in das optionale `assetModelId` Feld der Anfrage eingeben.

Wenn Sie dagegen in der Anfrage nichts angeben, AWS IoT SiteWise generiert es `assetModelId` eine UUID für das neue Asset-Modell.

Verwenden Sie externe IDs

Um Ihre eigene ID in einem anderen Format als UUID zu definieren, können Sie eine externe ID zuweisen. Sie können dies beispielsweise tun, wenn Sie eine ID, die Sie verwenden, in einem System wiederverwenden, das dies nicht ist AWS, oder um sie für Menschen lesbarer zu machen. Externe IDs haben ein flexibleres Format. Sie können sie verwenden, um bei AWS IoT SiteWise API-Vorgängen auf Ihre Objekte zu verweisen, bei denen Sie sonst die UUID verwenden würden.

Wie bei den UUIDs muss jede externe ID in ihrem Kontext eindeutig sein. Sie können beispielsweise nicht zwei Asset-Modelle mit derselben externen ID haben. Ebenso wie die UUIDs kann ein Objekt während seiner Lebensdauer nur eine externe ID haben, die sich nicht ändern kann.

Unterschiede zwischen externen IDs und UUIDs

Externe IDs unterscheiden sich in folgenden Punkten von UUIDs:

- Jedes Objekt hat eine UUID, externe IDs sind jedoch optional.
- AWS IoT SiteWise generiert niemals externe IDs. Sie stellen diese selbst zur Verfügung.
- Falls das Objekt noch keine hat, können Sie jederzeit eine externe ID vergeben.

Format der externen IDs

Eine gültige externe ID hat die folgenden Eigenschaften:

- Ist zwischen 2 und 128 Zeichen lang.
- Das erste und das letzte Zeichen müssen alphanumerisch sein (A-Z, a-z, 0-9).
- Andere Zeichen als das erste und das letzte müssen entweder alphanumerisch oder eines der folgenden Zeichen sein: `_ - . :`

Eine externe ID muss beispielsweise dem folgenden regulären Ausdruck entsprechen:

```
[a-zA-Z0-9][a-zA-Z0-9_\-.:]*[a-zA-Z0-9]+
```

Referenzobjekte mit externen IDs

An vielen Stellen, an denen Sie mithilfe seiner UUID auf ein Objekt verweisen könnten, können Sie stattdessen dessen externe ID verwenden, falls es eine hat. Hängen Sie dazu die externe ID an die Zeichenfolge an. `externalId`:

Nehmen wir beispielsweise an, Sie haben ein Asset-Modell, dessen UUID (Asset Model ID) lautet `a1b2c3d4-5678-90ab-cdef-11111EXAMPLE`, das auch die externe ID enthält. `myExternalId` Rufen Sie an [DescribeAssetModel](#), um weitere Informationen zu erhalten. Sie könnten einen der folgenden Werte als Wert von `verwendenassetModelId` verwenden:

- Mit der Asset Model ID (UUID) selbst: `a1b2c3d4-5678-90ab-cdef-11111EXAMPLE`
- Mit der externen ID: `externalId:myExternalId`

```
aws iotsitewise describe-asset-model --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
aws iotsitewise describe-asset-model --asset-model-id externalId:myExternalId
```

Note

Das `externalId`: Präfix selbst ist nicht Teil der externen ID. Sie müssen das Präfix nur angeben, wenn Sie eine externe ID für einen API-Vorgang angeben, der entweder UUIDs oder externe IDs akzeptiert. Geben Sie beispielsweise das Präfix an, wenn Sie ein vorhandenes Objekt abfragen oder aktualisieren.

Wenn Sie eine externe ID für ein Objekt definieren, z. B. wenn Sie ein Asset-Modell erstellen, geben Sie das Präfix nicht an.

Auf diese Weise können Sie externe IDs anstelle von UUIDs für viele API-Operationen verwenden AWS IoT SiteWise, aber nicht für alle. Beispielsweise muss der [GetAssetPropertyValue](#) UUIDs verwenden; die Verwendung externer IDs wird nicht unterstützt.

Informationen darüber, ob ein bestimmter API-Vorgang diese Verwendung unterstützt, finden Sie in der [API-Referenz](#).

Erstellen Sie Asset-Modelle, Komponentenmodelle und Schnittstellen für AWS IoT SiteWise

AWS IoT SiteWise Anlagenmodelle, Komponentenmodelle und Schnittstellen fördern die Standardisierung Ihrer Industriedaten. Anlagenmodelle definieren die Gesamtanlage, z. B. eine Windturbine oder eine Fertigungslinie. Komponentenmodelle stellen die einzelnen Komponenten dar, aus denen die Anlage besteht, wie Rotorblätter, Generatoren oder Sensoren. Durch Schnittstellen werden Standards für verschiedene Anlagenmodelle durchgesetzt. Durch die Erstellung dieser Modelle können Sie Ihre Anlagendaten so organisieren und strukturieren, dass sie die realen Beziehungen und Hierarchien Ihrer Industrieanlagen widerspiegeln, wodurch sie einfacher zu überwachen, zu analysieren und zu warten sind.

Ein Anlagen- oder Komponentenmodell enthält einen Namen, eine Beschreibung, Anlageneigenschaften und (optional) benutzerdefinierte Verbundmodelle, die Eigenschaften gruppieren oder auf Komponentenmodelle für Unterbaugruppen verweisen.

In AWS IoT SiteWise können Sie Anlagenmodelle, Komponentenmodelle und Schnittstellen erstellen, um die Struktur und Eigenschaften Ihrer Industrieanlagen und ihrer Komponenten darzustellen.

- Sie verwenden ein Anlagenmodell, um Anlagen zu erstellen. Zusätzlich zu den oben aufgeführten Funktionen kann ein Anlagenmodell auch Hierarchiedefinitionen enthalten, die Beziehungen zwischen Anlagen definieren.
- Ein Komponentenmodell stellt eine Unterbaugruppe innerhalb eines Anlagenmodells oder eines anderen Komponentenmodells dar. Wenn Sie ein Komponentenmodell erstellen, können Sie Referenzen darauf in Objektmodellen und anderen Komponentenmodellen hinzufügen. Sie können Objekte jedoch nicht direkt aus Komponentenmodellen erstellen.
- Eine Schnittstelle setzt Standards für verschiedene Asset-Modelle durch. Schnittstellen definieren gemeinsame Eigenschaften, Metriken und Hierarchien, die durch Anlagenmodelle implementiert werden müssen. Sie können Assets nicht direkt über Schnittstellen erstellen, aber sie tragen dazu bei, die Konsistenz zwischen ähnlichen Asset-Typen sicherzustellen.

Nachdem Sie ein Objekt- oder Komponentenmodell erstellt haben, können Sie benutzerdefinierte Verbundmodelle erstellen, um Eigenschaften zu gruppieren oder auf bestehende Komponentenmodelle zu verweisen. Sie können auch Schnittstellen mit Objektmodellen verknüpfen, um die Standardisierung durchzusetzen.

Einzelheiten zur Erstellung von Anlagenmodellen, Komponentenmodellen und Schnittstellen finden Sie in den folgenden Abschnitten.

Themen

- [Erstellen Sie Asset-Modelle inAWS IoT SiteWise](#)
- [Erstellen Sie Komponentenmodelle](#)
- [Definieren Sie Dateneigenschaften](#)
- [Erstellen Sie benutzerdefinierte Verbundmodelle \(Komponenten\)](#)

Erstellen Sie Asset-Modelle inAWS IoT SiteWise

AWS IoT SiteWiseAnlagenmodelle fördern die Standardisierung Ihrer Industriedaten. Ein Komponentenmodell enthält einen Namen, eine Beschreibung, Komponenteneigenschaften und Definitionen der Komponentenhierarchie. Sie können beispielsweise ein Windturbinenmodell mit Temperatur, Umdrehungen pro Minute (RPM) und Leistungseigenschaften definieren. Anschließend können Sie ein Windparkmodell mit einer Nettoleistungseigenschaft und einer Windturbinenhierarchiedefinition definieren.

Note

- Es empfiehlt sich, bei der Modellierung mit den Knoten der untersten Ebene zu beginnen. Erstellen Sie das Windturbinenmodell beispielsweise vor dem Windparkmodell. Komponentenhierarchiedefinitionen enthalten Verweise auf vorhandene Komponentenmodelle. Wenn Sie diesen Ansatz verfolgen, können Sie Komponentenhierarchien bei der Modellerstellung definieren.
- Anlagenmodelle können keine anderen Anlagenmodelle enthalten. Wenn Sie ein Modell definieren müssen, das Sie als Unterbaugruppe in einem anderen Modell referenzieren können, sollten Sie stattdessen ein Komponenten--> Modell erstellen. Weitere Informationen finden Sie unter [Erstellen Sie Komponentenmodelle](#).

In den folgenden Abschnitten wird beschrieben, wie Sie die AWS IoT SiteWise Konsole oder API verwenden, um Objektmodelle zu erstellen. In den folgenden Abschnitten werden auch die verschiedenen Arten von Komponenteneigenschaften und Komponentenhierarchien beschrieben, die Sie zum Erstellen von Modellen verwenden können.

Themen

- [Erstellen Sie ein Asset-Modell \(Konsole\)](#)
- [Erstellen Sie ein Asset-Modell \(AWS CLI\)](#)
- [Beispiel für Komponentenmodelle](#)
- [Definieren Sie die Hierarchien der Anlagenmodelle](#)

Erstellen Sie ein Asset-Modell (Konsole)

Sie können die AWS IoT SiteWise Konsole verwenden, um ein Asset-Modell zu erstellen. Die AWS IoT SiteWise Konsole bietet verschiedene Funktionen, z. B. die auto Vervollständigung von Formeln, mit denen Sie gültige Anlagenmodelle definieren können.

So erstellen Sie ein Komponentenmodell (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Klicken Sie im Navigationsbereich auf Models (Modelle).
3. Wählen Sie Asset-Modell erstellen aus.
4. Gehen Sie auf der Seite Modell erstellen wie folgt vor:
 - a. Geben Sie unter Name einen Namen für das Komponentenmodell ein, z. B. **Wind Turbine** oder **Wind Turbine Model**. Dieser Name muss für alle Modelle in Ihrem Konto in dieser Region eindeutig sein.
 - b. (Optional) Fügen Sie eine externe ID für das Modell hinzu. Dies ist eine benutzerdefinierte ID. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.
 - c. (Optional) Fügen Sie Messungsdefinitionen für das Modell hinzu. Messungen stellen Datenströme von Ihren Geräten dar. Weitere Informationen finden Sie unter [Definieren Sie Datenströme von Geräten \(Messungen\)](#).
 - d. (Optional) Fügen Sie Transformationsdefinitionen für das Modell hinzu. Transformationen sind Formeln, die Daten von einem Formular auf ein anderes abbilden. Weitere Informationen finden Sie unter [Daten transformieren \(transformiert\)](#).
 - e. (Optional) Fügen Sie Metrik-Definitionen für das Modell hinzu. Metriken sind Formeln, die Daten über Zeitintervalle aggregieren. Mit Metriken können Daten aus zugehörigen Anlagen eingegeben werden, sodass Sie Werte berechnen können, die Ihren Betrieb oder einen Teil

Ihres Betriebs repräsentieren. Weitere Informationen finden Sie unter [Aggregieren Sie Daten aus Immobilien und anderen Vermögenswerten \(Metriken\)](#).

- f. (Optional) Fügen Sie Hierarchiedefinitionen für das Modell hinzu. Hierarchien sind Beziehungen zwischen Anlagen. Weitere Informationen finden Sie unter [Definieren Sie die Hierarchien der Anlagenmodelle](#).
- g. (Optional) Fügen Sie Tags für das Komponentenmodell hinzu. Weitere Informationen finden Sie unter [Kennzeichnen Sie Ihre AWS IoT SiteWise Ressourcen](#).
- h. Wählen Sie Modell erstellen aus.

Wenn Sie ein Asset-Modell erstellen, navigiert die AWS IoT SiteWise Konsole zur Seite des neuen Modells. Auf dieser Seite sehen Sie den Status, des Modells, der anfänglich WIRD ERSTELLT lautet. Diese Seite wird automatisch aktualisiert. Sie können daher einfach abwarten, bis der Status des Modells aktualisiert wird.

 Note

Das Erstellen von Komponentenmodellen kann für komplexe Modelle einige Minuten in Anspruch nehmen. Wenn der Status des Asset-Modells AKTIV ist, können Sie das Asset-Modell verwenden, um Assets zu erstellen. Weitere Informationen finden Sie unter [Komponenten- und Modellzustände](#).

- 5. (Optional) Nachdem Sie Ihr Asset-Modell erstellt haben, können Sie Ihr Asset-Modell für den Edge konfigurieren. Weitere Informationen zu SiteWise Edge finden Sie unter [Konfigurieren Sie Edge-Funktionen auf AWS IoT SiteWise Edge](#).
- a. Wählen Sie auf der Modellseite Configure for Edge aus.

 Note

Das Feature Data Processing Pack (DPP) ist für Neukunden nicht mehr verfügbar. Vorhandene Kunden können den Service weiterhin wie gewohnt verwenden. Weitere Informationen finden Sie unter [Data processing pack availability change](#).

- b. Wählen Sie auf der Seite mit der Modellkonfiguration die Edge-Konfiguration für Ihr Modell aus. Dadurch wird gesteuert, AWS IoT SiteWise wo die mit diesem Asset-Modell verknüpften Eigenschaften berechnet und gespeichert werden können. Weitere

Informationen zur Konfiguration Ihres Modells für den Edge finden Sie unter [Richten Sie eine OPC UA-Quelle in SiteWise Edge ein](#).

- c. Wählen Sie für die benutzerdefinierte Kantenkonfiguration den Standort aus, AWS IoT SiteWise an dem Sie die einzelnen Eigenschaften Ihres Asset-Modells berechnen und speichern möchten.

 Note

Die zugehörigen Transformationen und Metriken müssen für denselben Standort konfiguriert werden. Weitere Informationen zur Konfiguration Ihres Modells für den Edge finden Sie unter [Richten Sie eine OPC UA-Quelle in SiteWise Edge ein](#).

- d. Wählen Sie Speichern. Auf der Modellseite sollte Ihre Edge-Konfiguration jetzt konfiguriert sein.

Erstellen Sie ein Asset-Modell (AWS CLI)

Sie können das AWS Command Line Interface (AWS CLI) verwenden, um ein Asset-Modell zu erstellen.

Verwenden Sie die Operation [CreateAssetModel](#), um ein Komponentenmodell mit Eigenschaften und Hierarchien zu erstellen. Diese Operation erwartet eine Nutzlast mit der folgenden Struktur.

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition
}
```

Um ein Asset-Modell zu erstellen (AWS CLI)

1. Erstellen Sie eine Datei namens `asset-model-payload.json` und kopieren Sie dann das folgende JSON-Objekt in die Datei.

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "",
```

```
"assetModelDescription": "",
"assetModelProperties": [

],
"assetModelHierarchies": [

],
"assetModelCompositeModels": [

]
}
```

2. Verwenden Sie Ihren bevorzugten JSON-Texteditor, um die Datei `asset-model-payload.json` für Folgendes zu bearbeiten:
 - a. Geben Sie einen Namen (`assetModelName`) für das Komponentenmodell ein, z. B. **Wind Turbine** oder **Wind Turbine Model**. Dieser Name muss in diesem Fall für alle Asset- und Komponentenmodelle in Ihrem Konto eindeutig sein **AWS-Region**.
 - b. (Optional) Geben Sie eine externe ID (`assetModelExternalId`) für das Asset-Modell ein. Dies ist eine benutzerdefinierte ID. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.
 - c. (Optional) Geben Sie eine Beschreibung (`assetModelDescription`) für das Komponentenmodell ein oder entfernen Sie das `assetModelDescription`-Schlüssel-Wert-Paar.
 - d. (Optional) Definieren Sie Komponenteneigenschaften (`assetModelProperties`) für das Modell. Weitere Informationen finden Sie unter [Definieren Sie Dateneigenschaften](#).
 - e. (Optional) Definieren Sie Komponentenhierarchien (`assetModelHierarchies`) für das Modell. Weitere Informationen finden Sie unter [Definieren Sie die Hierarchien der Anlagenmodelle](#).
 - f. (Optional) Definieren Sie Alarmer für das Modell. Alarmer überwachen andere Eigenschaften, sodass Sie erkennen können, wann Geräte oder Prozesse besondere Aufmerksamkeit erfordern. Jede Alarmdefinition ist ein zusammengesetztes Modell (`assetModelCompositeModels`), das die vom Alarm verwendeten Eigenschaften standardisiert. Weitere Informationen erhalten Sie unter [Überwachen Sie Daten mit Alarmen](#) **AWS IoT SiteWise** und [Definieren Sie Alarmer für Asset-Modelle in AWS IoT SiteWise](#).
 - g. (Optional) Fügen Sie Tags (`tags`) für das Komponentenmodell hinzu. Weitere Informationen finden Sie unter [Kennzeichnen Sie Ihre AWS IoT SiteWise Ressourcen](#).

3. Führen Sie den folgenden Befehl aus, um aus der Definition in der JSON-Datei ein Komponentenmodell zu erstellen.

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

Die Operation gibt eine Antwort zurück, die die `assetModelId` enthält, auf die Sie beim Erstellen einer Komponente verweisen. Die Antwort enthält auch den Zustand des Modells (`assetModelStatus.state`), der anfänglich `CREATING` lautet. Der Status des Komponentenmodells ist `CREATING`, bis die Änderungen weitergegeben werden.

Note

Das Erstellen von Komponentenmodellen kann für komplexe Modelle einige Minuten in Anspruch nehmen. Um den aktuellen Status Ihres Asset-Modells zu überprüfen, verwenden Sie den [DescribeAssetModel](#) Vorgang, indem Sie den `assetModelId` angeben. Wenn der Status des Komponentenmodells „ACTIVE“ lautet, können mit dem Komponentenmodell Komponenten erstellen. Weitere Informationen finden Sie unter [Komponenten- und Modellzustände](#).

4. (Optional) Erstellen Sie benutzerdefinierte Verbundmodelle für Ihr Anlagenmodell. Mit benutzerdefinierten Verbundmodellen können Sie Eigenschaften innerhalb des Modells gruppieren oder eine Unterbaugruppe einbeziehen, indem Sie auf ein Komponentenmodell verweisen. Weitere Informationen finden Sie unter [Erstellen Sie benutzerdefinierte Verbundmodelle \(Komponenten\)](#).

Beispiel für Komponentenmodelle

Dieser Abschnitt enthält Beispielfinitionen für Objektmodelle, die Sie verwenden können, um Objektmodelle mit den SDKs AWS CLI und AWS IoT SiteWise zu erstellen. Diese Anlagenmodelle stellen eine Windturbine und einen Windpark dar. Windkraftanlagen nehmen Sensorrohdaten auf und berechnen Werte wie Leistung und durchschnittliche Windgeschwindigkeit. Windparkanlagen berechnen Werte wie die Gesamtleistung aller Windturbinen im Windpark.

Themen

- [Windturbinen-Komponentenmodell](#)
- [Windpark-Komponentenmodell](#)

Windturbinen-Komponentenmodell

Das folgende Komponentenmodell stellt eine Turbine in einem Windpark dar. Die Windturbine nimmt Sensordaten auf, um Werte wie Leistung und durchschnittliche Windgeschwindigkeit zu berechnen.

Note

Dieses Beispielmmodell ähnelt dem Windturbinenmodell aus der AWS IoT SiteWise Demo. Weitere Informationen finden Sie unter [Benutze die AWS IoT SiteWise Demo](#).

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "Wind Turbine Asset Model",
  "assetModelDescription": "Represents a turbine in a wind farm.",
  "assetModelProperties": [
    {
      "name": "Location",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "Renton"
        }
      }
    },
    {
      "name": "Make",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "Amazon"
        }
      }
    },
    {
      "name": "Model",
      "dataType": "INTEGER",
      "type": {
        "attribute": {
          "defaultValue": "500"
        }
      }
    }
  ]
}
```

```
},
{
  "name": "Torque (KiloNewton Meter)",
  "dataType": "DOUBLE",
  "unit": "kNm",
  "type": {
    "measurement": {}
  }
},
{
  "name": "Wind Direction",
  "dataType": "DOUBLE",
  "unit": "Degrees",
  "type": {
    "measurement": {}
  }
},
{
  "name": "RotationsPerMinute",
  "dataType": "DOUBLE",
  "unit": "RPM",
  "type": {
    "measurement": {}
  }
},
{
  "name": "Wind Speed",
  "dataType": "DOUBLE",
  "unit": "m/s",
  "type": {
    "measurement": {}
  }
},
{
  "name": "RotationsPerSecond",
  "dataType": "DOUBLE",
  "unit": "RPS",
  "type": {
    "transform": {
      "expression": "rpm / 60",
      "variables": [
        {
          "name": "rpm",
          "value": {
```

```

        "propertyId": "RotationsPerMinute"
      }
    }
  ]
}
},
{
  "name": "Overdrive State",
  "dataType": "DOUBLE",
  "type": {
    "transform": {
      "expression": "gte(torque, 3)",
      "variables": [
        {
          "name": "torque",
          "value": {
            "propertyId": "Torque (KiloNewton Meter)"
          }
        }
      ]
    }
  }
},
{
  "name": "Average Power",
  "dataType": "DOUBLE",
  "unit": "Watts",
  "type": {
    "metric": {
      "expression": "avg(torque) * avg(rps) * 2 * 3.14",
      "variables": [
        {
          "name": "torque",
          "value": {
            "propertyId": "Torque (Newton Meter)"
          }
        },
        {
          "name": "rps",
          "value": {
            "propertyId": "RotationsPerSecond"
          }
        }
      ]
    }
  }
}
}

```

```

    ],
    "window": {
      "tumbling": {
        "interval": "5m"
      }
    }
  },
  {
    "name": "Average Wind Speed",
    "dataType": "DOUBLE",
    "unit": "m/s",
    "type": {
      "metric": {
        "expression": "avg(windspeed)",
        "variables": [
          {
            "name": "windspeed",
            "value": {
              "propertyId": "Wind Speed"
            }
          }
        ],
        "window": {
          "tumbling": {
            "interval": "5m"
          }
        }
      }
    }
  },
  {
    "name": "Torque (Newton Meter)",
    "dataType": "DOUBLE",
    "unit": "Nm",
    "type": {
      "transform": {
        "expression": "knm * 1000",
        "variables": [
          {
            "name": "knm",
            "value": {
              "propertyId": "Torque (KiloNewton Meter)"
            }
          }
        ]
      }
    }
  }
]

```

```

    }
  }
]
}
},
{
  "name": "Overdrive State Time",
  "dataType": "DOUBLE",
  "unit": "Seconds",
  "type": {
    "metric": {
      "expression": "statetime(overdrive_state)",
      "variables": [
        {
          "name": "overdrive_state",
          "value": {
            "propertyId": "Overdrive State"
          }
        }
      ],
      "window": {
        "tumbling": {
          "interval": "5m"
        }
      }
    }
  }
},
{
  "assetModelHierarchies": []
}

```

Windpark-Komponentenmodell

Das folgende Komponentenmodell stellt einen Windpark dar, der aus mehreren Windturbinen besteht. Dieses Anlagenmodell definiert eine [Hierarchie](#) für das Windturbinenmodell. Auf diese Weise kann der Windpark Werte (z. B. die Durchschnittsleistung) anhand von Daten für alle Windturbinen im Windpark berechnen.

 Note

Dieses Beispielmodell ähnelt dem Windparkmodell aus der AWS IoT SiteWise Demo. Weitere Informationen finden Sie unter [Benutze die AWS IoT SiteWise Demo](#).

Dieses Komponentenmodell hängt von der [Windturbinen-Komponentenmodell](#) ab. Ersetzen Sie die Werte `propertyId` und `childAssetModelId` durch die Werte eines vorhandenen Komponentenmodells für Windturbinen.

```
{
  "assetModelName": "Wind Farm Asset Model",
  "assetModelDescription": "Represents a wind farm.",
  "assetModelProperties": [
    {
      "name": "Code",
      "dataType": "INTEGER",
      "type": {
        "attribute": {
          "defaultValue": "300"
        }
      }
    },
    {
      "name": "Location",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "Renton"
        }
      }
    },
    {
      "name": "Reliability Manager",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "Mary Major"
        }
      }
    }
  ],
  {
```

```

    "name": "Total Overdrive State Time",
    "dataType": "DOUBLE",
    "unit": "seconds",
    "type": {
      "metric": {
        "expression": "sum(overdrive_state_time)",
        "variables": [
          {
            "name": "overdrive_state_time",
            "value": {
              "propertyId": "ID of Overdrive State Time property in Wind Turbine
Asset Model",
              "hierarchyId": "Turbine Asset Model"
            }
          }
        ],
        "window": {
          "tumbling": {
            "interval": "5m"
          }
        }
      }
    }
  },
  {
    "name": "Total Average Power",
    "dataType": "DOUBLE",
    "unit": "Watts",
    "type": {
      "metric": {
        "expression": "sum(turbine_avg_power)",
        "variables": [
          {
            "name": "turbine_avg_power",
            "value": {
              "propertyId": "ID of Average Power property in Wind Turbine Asset
Model",
              "hierarchyId": "Turbine Asset Model"
            }
          }
        ],
        "window": {
          "tumbling": {
            "interval": "5m"
          }
        }
      }
    }
  }
]

```



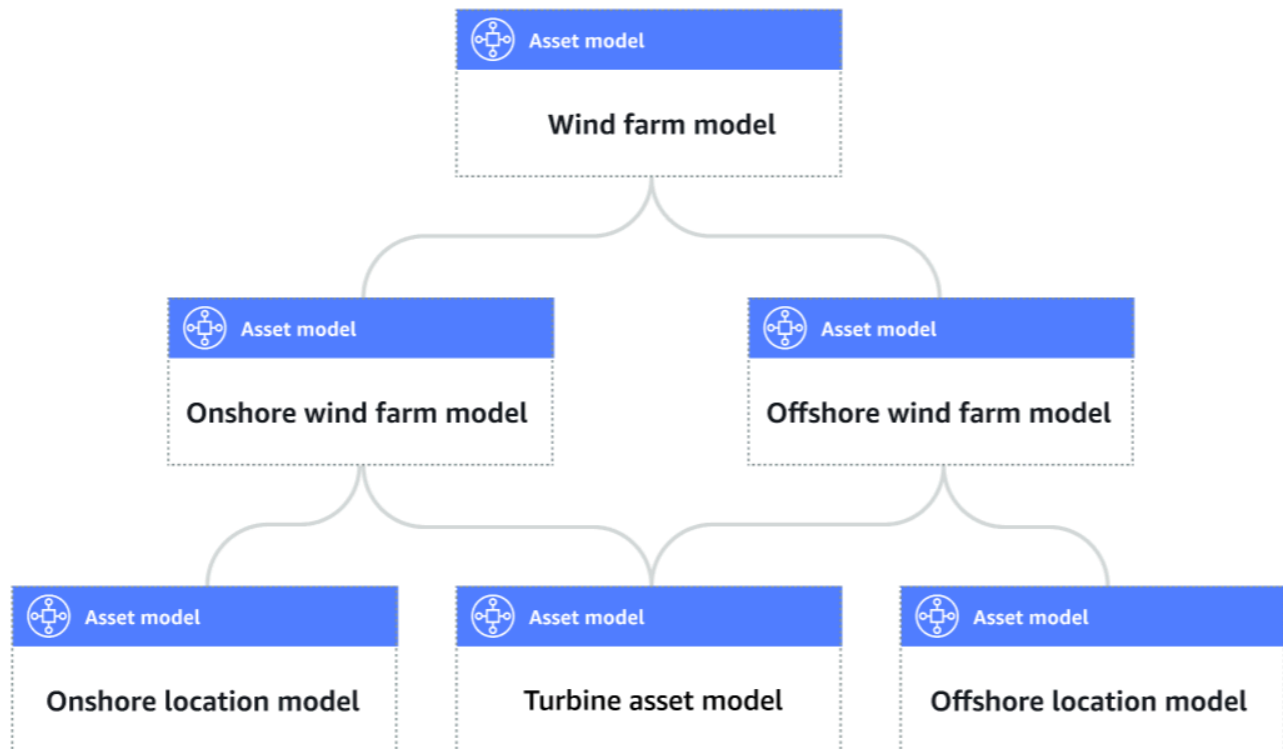
```
    }
  }
}
},
],
"assetModelHierarchies": [
  {
    "name": "Turbine Asset Model",
    "childAssetModelId": "ID of Wind Turbine Asset Model"
  }
]
}
```

Definieren Sie die Hierarchien der Anlagenmodelle

Sie können Anlagenmodellhierarchien definieren, um logische Verknüpfungen zwischen den Anlagenmodellen in Ihrem Industriebetrieb herzustellen. Sie können beispielsweise einen Windpark definieren, der aus Onshore- und Offshore-Windparks besteht. Ein Onshore-Windpark umfasst eine Turbine und einen Standort an Land. Ein Offshore-Windpark umfasst eine Turbine und einen Offshore-Standort.



Asset model hierarchy



Wenn Sie ein untergeordnetes Anlagenmodell über eine Hierarchie einem übergeordneten Anlagenmodell zuordnen, können die Metriken des übergeordneten Anlagenmodells Daten aus den Kennzahlen des untergeordneten Anlagenmodells eingeben. Sie können die Hierarchien und Kennzahlen des Anlagenmodells verwenden, um Statistiken zu berechnen, die Aufschluss über Ihren Betrieb oder einen Teil Ihres Betriebs geben. Weitere Informationen finden Sie unter [Aggregieren Sie Daten aus Immobilien und anderen Vermögenswerten \(Metriken\)](#).

Jede Hierarchie definiert eine Beziehung zwischen einem übergeordneten Anlagemodell und einem untergeordneten Anlagenmodell. In einem übergeordneten Anlagenmodell können Sie mehrere Hierarchien für dasselbe untergeordnete Anlagemodell definieren. Wenn Sie beispielsweise in Ihren Windparks über zwei verschiedene Typen von Windturbinen verfügen, bei denen alle Windturbinen durch dasselbe Anlagenmodell repräsentiert werden, können Sie für jeden Typ eine Hierarchie definieren. Anschließend können Sie im Windparkmodell Metriken definieren, um unabhängige und kombinierte Statistiken für jeden Windturbinentyp zu berechnen.

Ein übergeordnetes Anlagenmodell kann mehreren untergeordneten Vermögensmodellen zugeordnet werden. Wenn Sie beispielsweise einen Onshore-Windpark und einen Offshore-Windpark haben, die durch zwei verschiedene Anlagenmodelle repräsentiert werden, können Sie diese Anlagenmodelle demselben übergeordneten Windpark-Anlagenmodell zuordnen.

Ein untergeordnetes Anlagenmodell kann auch mehreren übergeordneten Vermögensmodellen zugeordnet werden. Wenn Sie beispielsweise über zwei verschiedene Arten von Windparks verfügen, bei denen alle Windturbinen durch dasselbe Anlagenmodell repräsentiert werden, können Sie das Anlagenmodell der Windturbine unterschiedlichen Windpark-Assetmodellen zuordnen.

Note

Wenn Sie eine Anlagenmodellhierarchie definieren, muss es sich bei dem untergeordneten Anlagenmodell um eine frühere Version handeln ACTIVE oder eine frühere ACTIVE Version haben. Weitere Informationen finden Sie unter [Komponenten- und Modellzustände](#).

Nachdem Sie hierarchische Komponentenmodelle definiert und Komponenten erstellt haben, können Sie die Komponenten zuordnen, um die Beziehung zwischen über- und untergeordneten Komponenten herzustellen. Weitere Informationen erhalten Sie unter [Erstellen Sie Objekte für Objektmodelle in AWS IoT SiteWise](#) und [Anlagen zuordnen und deren Zuordnung aufheben](#).

Themen

- [Definieren Sie die Hierarchien des Anlagenmodells \(Konsole\)](#)
- [Definieren Sie Asset-Hierarchien \(AWS CLI\)](#)

Definieren Sie die Hierarchien des Anlagenmodells (Konsole)

Wenn Sie in der AWS IoT SiteWise Konsole eine Hierarchie für ein Asset-Modell definieren, geben Sie die folgenden Parameter an:

- Hierarchiename — Der Name der Hierarchie, z. **Wind Turbines B**.
- Hierarchiemodell — Das untergeordnete Vermögensmodell.
- Externe Hierarchie-ID (optional) — Dies ist eine benutzerdefinierte ID. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.

Weitere Informationen finden Sie unter [Erstellen Sie ein Asset-Modell \(Konsole\)](#).

Definieren Sie Asset-Hierarchien (AWS CLI)

Wenn Sie mit der AWS IoT SiteWise API eine Hierarchie für ein Asset-Modell definieren, geben Sie die folgenden Parameter an:

- **name**— Der Name der Hierarchie, z. **Wind Turbines B**.
- **childAssetModelId**— Die ID oder die externe ID des untergeordneten Asset-Modells für die Hierarchie. Sie können die [ListAssetModels](#) Operation verwenden, um die ID eines vorhandenen Asset-Modells zu finden.

Example Beispiel für eine Hierarchiedefinition

Das folgende Beispiel zeigt eine Anlagenmodellhierarchie, die die Beziehung eines Windparks zu Windturbinen darstellt. Dieses Objekt ist ein Beispiel für ein [AssetModelHierarchy](#). Weitere Informationen finden Sie unter [Erstellen Sie ein Asset-Modell \(AWS CLI\)](#).

```
{
  ...
  "assetModelHierarchies": [
    {
      "name": "Wind Turbines",
      "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-1111EXAMPLE"
    },
  ],
}
```

Erstellen Sie Komponentenmodelle

Verwenden Sie AWS IoT SiteWise Komponentenmodelle, um Unterbaugruppen zu definieren, auf die Sie anhand von Objektmodellen oder anderen Komponentenmodellen verweisen können. Auf diese Weise können Sie die Definition der Komponente in mehreren anderen Modellen oder mehrfach innerhalb desselben Modells wiederverwenden.

Der Prozess der Definition eines Komponentenmodells ist der Definition eines Asset-Modells sehr ähnlich. Wie ein Asset-Modell hat auch ein Komponentenmodell einen Namen, eine Beschreibung und Asset-Eigenschaften. Komponentenmodelle können jedoch keine Definitionen der Asset-Hierarchie enthalten, da die Komponentenmodelle selbst nicht zur direkten Erstellung von Objekten verwendet werden können. Komponentenmodelle können auch keine Alarme definieren.

Sie können beispielsweise eine Komponente für einen Servomotor mit Eigenschaften für Motortemperatur, Encodertemperatur und Isolationswiderstand definieren. Anschließend können Sie ein Anlagenmodell für Geräte definieren, die Servomotoren enthalten, z. B. eine CNC-Maschine.

Note

- Es empfiehlt sich, bei der Modellierung mit den Knoten der untersten Ebene zu beginnen. Erstellen Sie beispielsweise Ihre Servomotorkomponente, bevor Sie das Anlagenmodell Ihrer CNC-Maschine erstellen. Objektmodelle enthalten Verweise auf bestehende Komponentenmodelle.
- Sie können ein Asset nicht direkt aus einem Komponentenmodell erstellen. Um ein Asset zu erstellen, das Ihre Komponente verwendet, müssen Sie ein Asset-Modell für Ihr Asset erstellen. Anschließend erstellen Sie dafür ein benutzerdefiniertes Verbundmodell, das auf Ihre Komponente verweist. Weitere Informationen zum Erstellen von Objektmodellen finden Sie unter [Weitere Informationen](#) [Erstellen Sie Asset-Modelle in AWS IoT SiteWise](#) zum Erstellen von benutzerdefinierten Verbundmodellen finden Sie unter [Erstellen Sie benutzerdefinierte Verbundmodelle \(Komponenten\)](#).

In den folgenden Abschnitten wird beschrieben, wie Sie die AWS IoT SiteWise API zum Erstellen von Komponentenmodellen verwenden.

Themen

- [Erstellen Sie ein Komponentenmodell \(AWS CLI\)](#)
- [Beispiel für ein Komponentenmodell](#)

Erstellen Sie ein Komponentenmodell (AWS CLI)

Sie können das AWS Command Line Interface (AWS CLI) verwenden, um ein Komponentenmodell zu erstellen.

Verwenden Sie die [CreateAssetModel](#) Operation, um ein Komponentenmodell mit Eigenschaften zu erstellen. Für diesen Vorgang wird eine Nutzlast mit der folgenden Struktur erwartet:

```
{
  "assetModelType": "COMPONENT_MODEL",
  "assetModelName": "String",
```

```
"assetModelDescription": "String",  
"assetModelProperties": Array of AssetModelProperty,  
}
```

Um ein Komponentenmodell zu erstellen (AWS CLI)

1. Erstellen Sie eine Datei mit dem Namen `component-model-payload.json` und kopieren Sie dann das folgende JSON-Objekt in die Datei:

```
{  
  "assetModelType": "COMPONENT_MODEL",  
  "assetModelName": "",  
  "assetModelDescription": "",  
  "assetModelProperties": [  
  
  ]  
}
```

2. Verwenden Sie Ihren bevorzugten JSON-Texteditor, um die Datei `component-model-payload.json` für Folgendes zu bearbeiten:
 - a. Geben Sie einen Namen (`assetModelName`) für das Komponentenmodell ein, z. B. **Servo Motor** oder **Servo Motor Model**. Dieser Name muss in diesem Fall für alle Asset- und Komponentenmodelle in Ihrem Konto eindeutig seinAWS-Region.
 - b. (Optional) Geben Sie eine externe ID (`assetModelExternalId`) für das Komponentenmodell ein. Dies ist eine benutzerdefinierte ID. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.
 - c. (Optional) Geben Sie eine Beschreibung (`assetModelDescription`) für das Komponentenmodell ein oder entfernen Sie das `assetModelDescription`-Schlüssel-Wert-Paar.
 - d. (Optional) Definieren Sie Asset-Eigenschaften (`assetModelProperties`) für das Komponentenmodell. Weitere Informationen finden Sie unter [Definieren Sie Dateneigenschaften](#).
 - e. (Optional) Fügen Sie Tags (`tags`) für das Komponentenmodell hinzu. Weitere Informationen finden Sie unter [Kennzeichnen Sie Ihre AWS IoT SiteWise Ressourcen](#).
3. Führen Sie den folgenden Befehl aus, um ein Komponentenmodell aus der Definition in der JSON-Datei zu erstellen.

```
aws iotsitewise create-asset-model --cli-input-json file://component-model-  
payload.json
```

Der Vorgang gibt eine Antwort zurück, die die Antwort enthält, auf `assetModelId` die Sie sich beziehen, wenn Sie einen Verweis auf Ihr Komponentenmodell in einem Asset-Modell oder einem anderen Komponentenmodell hinzufügen. Die Antwort enthält auch den Zustand des Modells (`assetModelStatus.state`), der anfänglich `CREATING` lautet. Der Status des Komponentenmodells ist so `CREATING` lange gültig, bis die Änderungen übernommen werden.

Note

Die Erstellung des Komponentenmodells kann bei komplexen Modellen bis zu einigen Minuten dauern. Um den aktuellen Status Ihres Komponentenmodells zu überprüfen, verwenden Sie den [DescribeAssetModel](#) Vorgang, indem Sie den `assetModelId` angeben. Sobald der Status des Komponentenmodells lautet `ACTIVE`, können Sie Verweise auf Ihr Komponentenmodell in Objektmodellen oder anderen Komponentenmodellen hinzufügen. Weitere Informationen finden Sie unter [Komponenten- und Modellzustände](#).

4. (Optional) Erstellen Sie benutzerdefinierte Verbundmodelle für Ihr Komponentenmodell. Bei benutzerdefinierten Verbundmodellen können Sie Eigenschaften innerhalb des Modells gruppieren oder eine Unterbaugruppe einbeziehen, indem Sie auf ein anderes Komponentenmodell verweisen. Weitere Informationen finden Sie unter [Erstellen Sie benutzerdefinierte Verbundmodelle \(Komponenten\)](#).

Beispiel für ein Komponentenmodell

Dieser Abschnitt enthält eine Beispielfinition für ein Komponentenmodell, mit der Sie ein Komponentenmodell mit den AWS IoT SiteWise SDKs AWS CLI und erstellen können. Dieses Komponentenmodell stellt einen Servomotor dar, der in einem anderen Gerät, z. B. einer CNC-Maschine, verwendet werden kann.

Themen

- [Komponentenmodell des Servomotors](#)

Komponentenmodell des Servomotors

Das folgende Komponentenmodell stellt einen Servomotor dar, der in Geräten wie CNC-Maschinen verwendet werden kann. Der Servomotor bietet verschiedene Messwerte, z. B. für Temperaturen und elektrischen Widerstand. Diese Messungen sind als Eigenschaften für Objekte verfügbar, die aus Objektmodellen erstellt wurden, die auf das Komponentenmodell des Servomotors verweisen.

```
{
  "assetModelName": "ServoMotor",
  "assetModelType": "COMPONENT_MODEL",
  "assetModelProperties": [
    {
      "dataType": "DOUBLE",
      "name": "Servo Motor Temperature",
      "type": {
        "measurement": {}
      },
      "unit": "Celsius"
    },
    {
      "dataType": "DOUBLE",
      "name": "Spindle speed",
      "type": {
        "measurement": {}
      },
      "unit": "rpm"
    }
  ]
}
```

Definieren Sie Dateneigenschaften

Asset-Eigenschaften sind die Strukturen innerhalb jedes Assets, die Asset-Daten enthalten. Bei den Komponenteneigenschaften kann es sich um folgende Typen handeln:

- **Attribute** — Die im Allgemeinen statischen Eigenschaften eines Assets, z. B. Gerätehersteller oder geografische Region. Weitere Informationen finden Sie unter [Definieren Sie statische Daten \(Attribute\)](#).
- **Messungen** — Die Sensordatenströme eines Geräts im Rohformat, z. B. mit Zeitstempel versehene Drehzahlwerte oder Temperaturwerte mit Zeitstempel in Celsius. Eine Messung wird durch einen

Daten-Stream-Alias definiert. Weitere Informationen finden Sie unter [Definieren Sie Datenströme von Geräten \(Messungen\)](#).

- Transformationen — Die transformierten Zeitreihenwerte eines Assets, z. B. Temperaturwerte mit Zeitstempel in Fahrenheit. Eine Transformation wird durch einen Ausdruck und die Variablen definiert, die mit diesem Ausdruck verwendet werden sollen. Weitere Informationen finden Sie unter [Daten transformieren \(transformiert\)](#).
- Metriken — Die Daten einer Anlage, die über ein bestimmtes Zeitintervall aggregiert wurden, z. B. die stündliche Durchschnittstemperatur. Eine Metrik wird durch ein Zeitintervall, einen Ausdruck und die Variablen definiert, die mit diesem Ausdruck verwendet werden sollen. Metrische Ausdrücke können die metrischen Eigenschaften der zugehörigen Anlagen eingeben, sodass Sie Metriken berechnen können, die Ihren Betrieb oder eine Teilmenge Ihres Betriebs repräsentieren. Weitere Informationen finden Sie unter [Aggregieren Sie Daten aus Immobilien und anderen Vermögenswerten \(Metriken\)](#).

Weitere Informationen finden Sie unter [Erstellen Sie Asset-Modelle in AWS IoT SiteWise](#).

Ein Beispiel für die Verwendung von Messungen, Transformationen und Metriken zur Berechnung der Gesamtanlageneffektivität (Overall Equipment Effectiveness, OEE) finden Sie unter [Berechnung der Gesamtanlageneffektivität in AWS IoT SiteWise](#).

Themen

- [Definieren Sie statische Daten \(Attribute\)](#)
- [Definieren Sie Datenströme von Geräten \(Messungen\)](#)
- [Daten transformieren \(transformiert\)](#)
- [Aggregieren Sie Daten aus Immobilien und anderen Vermögenswerten \(Metriken\)](#)
- [Verwenden Sie Formelausdrücke](#)

Definieren Sie statische Daten (Attribute)

Asset-Attribute stellen Informationen dar, die im Allgemeinen statisch sind, z. B. Gerätehersteller oder geografischer Standort. Jede Komponente, die Sie anhand eines Komponentenmodells erstellen, enthält die Attribute dieses Modells.

Themen

- [Definieren Sie Attribute \(Konsole\)](#)

- [Definieren Sie Attribute \(AWS CLI\)](#)

Definieren Sie Attribute (Konsole)

Wenn Sie in der AWS IoT SiteWise Konsole ein Attribut für ein Asset-Modell definieren, geben Sie die folgenden Parameter an:

- Name — Der Name der Immobilie.
- Standardwert — (Optional) Der Standardwert für dieses Attribut. Aus dem Modell erstellte Objekte haben diesen Wert für das Attribut. Weitere Informationen zum Überschreiben des Standardwerts in einer aus einem Modell erstellten Komponente finden Sie unter [Attributwerte aktualisieren](#).
- Datentyp — Der Datentyp der Eigenschaft, der einer der folgenden ist:
 - Zeichenfolge — Eine Zeichenfolge mit bis zu 1024 Byte.
 - Integer — Eine 32-Bit-Ganzzahl mit Vorzeichen und einem Bereich von [-2.147.483.648, 2.147.483.647].
 - Double — Eine Gleitkommazahl mit einem Bereich [-10¹⁰⁰, 10¹⁰⁰] und einer doppelten IEEE-754-Genauigkeit.
 - **false**Boolean — **true** oder.
- Externe ID — (Optional) Dies ist eine benutzerdefinierte ID. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.

Weitere Informationen finden Sie unter [Erstellen Sie ein Asset-Modell \(Konsole\)](#).

Definieren Sie Attribute (AWS CLI)

Wenn Sie mit der AWS IoT SiteWise API ein Attribut für ein Asset-Modell definieren, geben Sie die folgenden Parameter an:

- name— Der Name der Immobilie.
- defaultValue— (Optional) Der Standardwert für dieses Attribut. Aus dem Modell erstellte Objekte haben diesen Wert für das Attribut. Weitere Informationen zum Überschreiben des Standardwerts in einer aus einem Modell erstellten Komponente finden Sie unter [Attributwerte aktualisieren](#).
- dataType— Der Datentyp der Eigenschaft, der einer der folgenden ist:
 - STRING— Eine Zeichenfolge mit bis zu 1024 Byte.
 - INTEGER— Eine 32-Bit-Ganzzahl mit Vorzeichen im Bereich [-2.147.483.648, 2.147.483.647].

- **DOUBLE**— Eine Fließkommazahl mit einem Bereich $[-10^{100}, 10^{100}]$ und einer doppelten IEEE-754-Genauigkeit.
- **BOOLEAN**— `true` oder `false`
- **externalId**— (Optional) Dies ist eine benutzerdefinierte ID. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.

Example Beispiel für eine Attributdefinition

Im folgenden Beispiel wird ein Attribut veranschaulicht, das die Modellnummer einer Komponente mit einem Standardwert darstellt. Dieses Objekt ist ein Beispiel für ein [AssetModelProperty](#), das ein [Attribut](#) enthält. Sie können dieses Objekt als Teil der [CreateAssetModel](#)-Anforderungs-Nutzlast angeben, um eine Attributeigenschaft zu erstellen. Weitere Informationen finden Sie unter [Erstellen Sie ein Asset-Modell \(AWS CLI\)](#).

```
{
  ...
  "assetModelProperties": [
    {
      "name": "Model number",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "BLT123"
        }
      }
    }
  ],
  ...
}
```

Definieren Sie Datenströme von Geräten (Messungen)

Eine Messung stellt den rohen Sensordatenstrom eines Geräts dar, z. B. Temperaturwerte mit Zeitstempel oder Werte für Umdrehungen pro Minute (U/min) mit Zeitstempel.

Themen

- [Definieren Sie Messungen \(Konsole\)](#)
- [Maße definieren \(AWS CLI\)](#)

Definieren Sie Messungen (Konsole)

Wenn Sie in der AWS IoT SiteWise Konsole eine Messung für ein Anlagenmodell definieren, geben Sie die folgenden Parameter an:

- **Name** — Der Name der Immobilie.
- **Einheit** — (Optional) Die wissenschaftliche Einheit für die Eigenschaft, z. B. mm oder Celsius.
- **Datentyp** — Der Datentyp der Eigenschaft, der einer der folgenden ist:
 - **Zeichenfolge** — Eine Zeichenfolge mit bis zu 1024 Byte.
 - **Integer** — Eine 32-Bit-Ganzzahl mit Vorzeichen und einem Bereich von [-2.147.483.648, 2.147.483.647].
 - **Double** — Eine Gleitkommazahl mit einem Bereich [-10¹⁰⁰, 10¹⁰⁰] und einer doppelten IEEE-754-Genauigkeit.
 - **false** Boolean — **true** oder.
- **Externe ID** — (Optional) Dies ist eine benutzerdefinierte ID. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.

Weitere Informationen finden Sie unter [Erstellen Sie ein Asset-Modell \(Konsole\)](#).

Maße definieren (AWS CLI)

Wenn Sie mit der AWS IoT SiteWise API eine Messung für ein Asset-Modell definieren, geben Sie die folgenden Parameter an:

- **name** — Der Name der Immobilie.
- **dataType** — Der Datentyp der Eigenschaft, der einer der folgenden ist:
 - **STRING** — Eine Zeichenfolge mit bis zu 1024 Byte.
 - **INTEGER** — Eine 32-Bit-Ganzzahl mit Vorzeichen im Bereich [-2.147.483.648, 2.147.483.647].
 - **DOUBLE** — Eine Fließkommazahl mit einem Bereich [-10¹⁰⁰, 10¹⁰⁰] und einer doppelten IEEE-754-Genauigkeit.
 - **BOOLEAN** — **true** oder **false**
- **unit** — (Optional) Die wissenschaftliche Einheit für die Eigenschaft, z. B. mm oder Celsius.
- **externalId** — (Optional) Dies ist eine benutzerdefinierte ID. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.

Example Beispiel für eine Messdefinition

Das folgende Beispiel zeigt eine Messung, die die Messwerte der Temperatursensoren einer Komponente darstellt. Dieses Objekt ist ein Beispiel für ein Objekt [AssetModelProperty](#), das eine [Messung](#) enthält. Sie können dieses Objekt als Teil der [CreateAssetModel](#)-Anforderungs-Nutzlast angeben, um eine Messungseigenschaft zu erstellen. Weitere Informationen finden Sie unter [Erstellen Sie ein Asset-Modell \(AWS CLI\)](#).

Die [Messstruktur](#) ist eine leere Struktur, wenn Sie ein Asset-Modell definieren, da Sie später jedes Asset so konfigurieren, dass es eindeutige Gerätedatenströme verwendet. Weitere Informationen darüber, wie Sie die Messeigenschaft einer Anlage mit dem Sensordatenstrom eines Geräts verbinden, finden Sie unter [Datenströme verwalten für AWS IoT SiteWise](#).

```
{
  ...
  "assetModelProperty": [
    {
      "name": "Temperature C",
      "dataType": "DOUBLE",
      "type": {
        "measurement": {}
      },
      "unit": "Celsius"
    }
  ],
  ...
}
```

Daten transformieren (transformiert)

Transformationen sind mathematische Ausdrücke, die die Datenpunkte von Komponenteneigenschaften aus einer Form in einer anderen Form abbilden. Ein Transformationsausdruck besteht aus Variablen, Literalen, Operatoren und Funktionen für Asset-Eigenschaften. Die transformierten Datenpunkte stehen in einer Eins-zu-Eins-Beziehung zu den Eingabedatenpunkten. AWS IoT SiteWise berechnet jedes Mal, wenn eine der Eingabeeigenschaften einen neuen Datenpunkt erhält, einen neuen transformierten Datenpunkt.

 Note

Bei Eigenschaftenaktualisierungen mit demselben Zeitstempel können Ausgabewerte durch Aktualisierungen anderer eingehender Eigenschaften überschrieben werden.

Wenn Ihre Komponente beispielsweise über einen Temperaturmessungs-Stream namens `Temperature_C` mit Einheiten in Celsius verfügt, können Sie jeden Datenpunkt mit der Formel $\text{Temperature\_F} = 9/5 * \text{Temperature\_C} + 32$ in Fahrenheit konvertieren. Jedes Mal, wenn ein Datenpunkt im `Temperature_C` Messstream AWS IoT SiteWise empfangen wird, wird der entsprechende `Temperature_F` Wert innerhalb weniger Sekunden berechnet und ist als Eigenschaft verfügbar. `Temperature_F`

Wenn Ihre Transformation mehr als eine Variable enthält, leitet der Datenpunkt, der früher eintrifft, die Berechnung sofort ein. Stellen Sie sich ein Beispiel vor, bei dem ein Teilehersteller eine Transformation verwendet, um die Produktqualität zu überwachen. Der Hersteller verwendet je nach Bauteiltyp eine andere Norm und verwendet die folgenden Maße, um den Prozess darzustellen:

- `Part_Number`- Eine Zeichenfolge, die den Teiletyp identifiziert.
- `Good_Count`- Eine Ganzzahl, die um eins erhöht wird, wenn das Teil der Norm entspricht.
- `Bad_Count`- Eine Ganzzahl, die um eins erhöht wird, wenn das Teil nicht der Norm entspricht.

Der Hersteller erstellt auch eine Transformation, `Quality_Monitor`, die entspricht.

```
if(eq(Part_Number, "BLT123") and (Bad_Count / (Good_Count + Bad_Count) > 0.1), "Caution", "Normal")
```

Diese Transformation überwacht den Prozentsatz fehlerhafter Teile, die für einen bestimmten Teiletyp hergestellt wurden. Wenn die Artikelnummer BLT123 lautet und der Prozentsatz der fehlerhaften Teile 10 Prozent (0,1) übersteigt, gibt die Transformation zurück. "Caution" Andernfalls kehrt die Transformation zurück. "Normal"

 Note

- Wenn vor anderen Messungen ein neuer Datenpunkt `Part_Number` empfangen wird, verwendet die `Quality_Monitor` Transformation den neuen `Part_Number` Wert und die neuesten `Good_Count` `Bad_Count` UND-Werte. Um Fehler zu vermeiden, setzen `Good_Count` Sie das `Bad_Count` Gerät vor dem nächsten Fertigungslauf zurück.

- Verwenden Sie [Metriken](#), wenn Sie Ausdrücke erst auswerten möchten, nachdem alle Variablen neue Datenpunkte erhalten haben.

Themen

- [Transformationen definieren \(Konsole\)](#)
- [Definieren Sie Transformationen \(AWS CLI\)](#)

Transformationen definieren (Konsole)

Wenn Sie in der AWS IoT SiteWise Konsole eine Transformation für ein Asset-Modell definieren, geben Sie die folgenden Parameter an:

- Name — Der Name der Immobilie.
- Einheit — (Optional) Die wissenschaftliche Einheit für die Eigenschaft, z. B. mm oder Celsius.
- Datentyp — Der Datentyp der Transformation, der Double oder String sein kann.
- Externe ID — (Optional) Dies ist eine benutzerdefinierte ID. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.
- Formel — Der Transformationsausdruck. Transformationsausdrücke können keine Aggregationsfunktionen oder Temporalfunktionen verwenden. Um die Funktion zur auto Vervollständigung zu öffnen, beginnen Sie mit der Eingabe oder drücken Sie die NACH-UNTEN-TASTE. Weitere Informationen finden Sie unter [Verwenden Sie Formelausdrücke](#).

Important

Transformationen können Eigenschaften vom Typ Integer, Double, Boolean oder Zeichenfolge eingeben. Boolesche Werte werden in 0 (falsch) und (wahr) konvertiert. 1 Transformationen müssen eine oder mehrere Eigenschaften, die keine Attribute sind, und eine beliebige Anzahl von Attributeigenschaften eingeben. AWS IoT SiteWise berechnet jedes Mal einen neuen transformierten Datenpunkt, wenn die Eingabeeigenschaft, bei der es sich nicht um ein Attribut handelt, einen neuen Datenpunkt erhält. Neue Attributwerte starten keine Transformationsaktualisierungen. Für Ergebnisse der Transformationsberechnung gilt dieselbe Anforderungsrate für API-Operationen mit Objektdaten.

Formelausdrücke können nur Doppelwerte oder Zeichenkettenwerte ausgeben.

Verschachtelte Ausdrücke können andere Datentypen ausgeben, z. B. Zeichenfolgen,

aber die Formel als Ganzes muss eine Zahl oder Zeichenfolge ergeben. Sie können die [Funktion jp](#) verwenden, um eine Zeichenfolge in eine Zahl umzuwandeln. Der boolesche Wert muss 1 (wahr) oder 0 (falsch) sein. Weitere Informationen finden Sie unter [Undefinierte, unendliche und Überlaufwerte](#).

Weitere Informationen finden Sie unter [Erstellen Sie ein Asset-Modell \(Konsole\)](#).

Definieren Sie Transformationen (AWS CLI)

Wenn Sie mit der AWS IoT SiteWise API eine Transformation für ein Asset-Modell definieren, geben Sie die folgenden Parameter an:

- **name**— Der Name der Immobilie.
- **unit**— (Optional) Die wissenschaftliche Einheit für die Eigenschaft, z. B. mm oder Celsius.
- **dataType**— Der Datentyp der Transformation, der DOUBLE oder sein muss STRING.
- **externalId**— (Optional) Dies ist eine benutzerdefinierte ID. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.
- **expression**— Der Transformationsausdruck. Transformationsausdrücke können keine Aggregationsfunktionen oder Temporalfunktionen verwenden. Weitere Informationen finden Sie unter [Verwenden Sie Formelausdrücke](#).
- **variables**— Die Liste der Variablen, die die anderen Eigenschaften Ihres Assets definiert, die im Ausdruck verwendet werden sollen. Jede Variablenstruktur enthält einen einfachen Namen, der in dem Ausdruck verwendet werden soll, sowie eine **value**-Struktur zur Identifizierung der mit dieser Variablen zu verknüpfenden Eigenschaft. Die **value**-Struktur enthält folgende Informationen:
 - **propertyId**— Die ID der Eigenschaft, aus der Werte eingegeben werden sollen. Sie können den Namen der Eigenschaft anstelle der ID verwenden.

Important

Transformationen können Eigenschaften vom Typ Integer, Double, Boolean oder String eingeben. Boolesche Werte werden in 0 (falsch) und 1 (wahr) konvertiert.

Transformationen müssen eine oder mehrere Eigenschaften, die keine Attribute sind, und eine beliebige Anzahl von Attributeigenschaften eingeben. AWS IoT SiteWise berechnet jedes Mal einen neuen transformierten Datenpunkt, wenn die Eingabeeigenschaft, bei der es sich nicht um ein Attribut handelt, einen neuen Datenpunkt erhält. Neue Attributwerte starten keine Transformationsaktualisierungen. Für Ergebnisse der

Transformationsberechnung gilt dieselbe Anforderungsrate für API-Operationen mit Objektdaten.

Formelausdrücke können nur Doppelwerte oder Zeichenkettenwerte ausgeben.

Verschachtelte Ausdrücke können andere Datentypen ausgeben, z. B. Zeichenfolgen, aber die Formel als Ganzes muss eine Zahl oder Zeichenfolge ergeben. Sie können die [Funktion jp](#) verwenden, um eine Zeichenfolge in eine Zahl umzuwandeln. Der boolesche Wert muss 1 (wahr) oder 0 (falsch) sein. Weitere Informationen finden Sie unter [Undefinierte, unendliche und Überlaufwerte](#).

Example Definition transformieren

Das folgende Beispiel zeigt eine Transformationseigenschaft, die die Temperaturmessdaten einer Komponente von Celsius in Fahrenheit konvertiert. Dieses Objekt ist ein Beispiel für ein [AssetModelProperty](#), das eine [Transformation](#) enthält. Sie können dieses Objekt als Teil der [CreateAssetModel](#)-Anforderungs-Nutzlast angeben, um eine Transformationseigenschaft zu erstellen. Weitere Informationen finden Sie unter [Erstellen Sie ein Asset-Modell \(AWS CLI\)](#).

```
{
  ...
  "assetModelProperty": [
    ...
    {
      "name": "Temperature F",
      "dataType": "DOUBLE",
      "type": {
        "transform": {
          "expression": "9/5 * temp_c + 32",
          "variables": [
            {
              "name": "temp_c",
              "value": {
                "propertyId": "Temperature C"
              }
            }
          ]
        }
      },
      "unit": "Fahrenheit"
    }
  ],
}
```

```
...
}
```

Example Transformationsdefinition, die drei Variablen enthält

Das folgende Beispiel zeigt eine Transform-Eigenschaft, die eine Warnmeldung ("Caution") zurückgibt, wenn mehr als 10 Prozent der BLT123-Teile nicht dem Standard entsprechen. Andernfalls wird eine Informationsmeldung () "Normal" zurückgegeben.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "Quality_Monitor",
      "dataType": "STRING",
      "type": {
        "transform": {
          "expression": "if(eq(Part_Number,\"BLT123\") and (Bad_Count / (Good_Count +
Bad_Count) > 0.1), \"Caution\", \"Normal\")",
          "variables": [
            {
              "name": "Part_Number",
              "value": {
                "propertyId": "Part Number"
              }
            },
            {
              "name": "Good_Count",
              "value": {
                "propertyId": "Good Count"
              }
            },
            {
              "name": "Bad_Count",
              "value": {
                "propertyId": "Bad Count"
              }
            }
          ]
        }
      }
    }
  ]
}
```

```
...  
}
```

Aggregieren Sie Daten aus Immobilien und anderen Vermögenswerten (Metriken)

Metriken sind mathematische Ausdrücke, die Aggregierungsfunktionen verwenden, um alle Eingabedatenpunkte zu verarbeiten und einen einzelnen Datenpunkt pro angegebenem Zeitintervall auszugeben. Eine Metrik kann beispielsweise die stündliche Durchschnittstemperatur aus einem Temperaturdaten-Stream berechnen.

Metriken können Daten aus Metriken zugehöriger Komponenten eingeben, sodass Sie Statistiken berechnen können, die einen Einblick in die Operation oder eine Teilmenge der Operation gewähren. Beispielsweise kann eine Metrik die durchschnittliche stündliche Temperatur für alle Windturbinen in einem Windpark berechnen. Weitere Informationen zum Definieren von Verknüpfungen zwischen Komponenten finden Sie unter [Definieren Sie die Hierarchien der Anlagenmodelle](#).

Mit Metriken können auch Daten aus anderen Eigenschaften eingegeben werden, ohne dass Daten für jedes Zeitintervall aggregiert werden müssen. Wenn Sie ein [Attribut](#) in einer Formel angeben, AWS IoT SiteWise verwendet es bei der Berechnung der Formel den [neuesten](#) Wert für dieses Attribut. Wenn Sie eine Metrik in einer Formel angeben, AWS IoT SiteWise verwendet es den [letzten](#) Wert für das Zeitintervall, über das die Formel berechnet wird. Das bedeutet, dass Sie Metriken wie $OEE = Availability * Quality * Performance$ definieren können, wo und wie alle anderen Metriken für dasselbe Asset-Modell Performance sind, definieren können.

AWS IoT SiteWise berechnet außerdem automatisch eine Reihe grundlegender Aggregationsmetriken für alle Asset-Eigenschaften. Um Berechnungskosten zu reduzieren, können Sie diese Aggregate verwenden, anstatt benutzerdefinierte Metriken für grundlegende Berechnungen zu definieren. Weitere Informationen finden Sie unter [Abfragen von Asset-Eigenschaftenaggregaten in AWS IoT SiteWise](#).

Themen

- [Metriken definieren \(Konsole\)](#)
- [Definieren Sie Metriken \(AWS CLI\)](#)

Metriken definieren (Konsole)

Wenn Sie in der AWS IoT SiteWise Konsole eine Metrik für ein Asset-Modell definieren, geben Sie die folgenden Parameter an:

- Name — Der Name der Immobilie.
- Datentyp — Der Datentyp der Transformation, der Double oder String sein kann.
- Externe ID — (Optional) Dies ist eine benutzerdefinierte ID. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.
- Formel — Der metrische Ausdruck. Metrische Ausdrücke können [Aggregationsfunktionen](#) verwenden, um Daten aus einer Eigenschaft für alle zugehörigen Anlagen in einer Hierarchie einzugeben. Beginnen Sie mit der Eingabe oder drücken Sie die Abwärtspfeiltaste, um die Funktion zur auto Vervollständigung zu öffnen. Weitere Informationen finden Sie unter [Verwenden Sie Formelausdrücke](#).

 Important

Bei Metriken kann es sich nur um Eigenschaften vom Typ Integer, Double, Boolean oder Zeichenfolge handeln. Boolesche Werte werden in 0 (falsch) und 1 (wahr) konvertiert. Wenn Sie Metrikeingabevariablen im Ausdruck einer Metrik definieren, muss für diese Eingaben dasselbe Zeitintervall wie für die Ausgabemetrik gelten.

Formelausdrücke können nur Doppelwerte oder Zeichenkettenwerte ausgeben.

Verschachtelte Ausdrücke können andere Datentypen ausgeben, z. B. Zeichenfolgen, aber die Formel als Ganzes muss eine Zahl oder Zeichenfolge ergeben. Sie können die [Funktion jp](#) verwenden, um eine Zeichenfolge in eine Zahl umzuwandeln. Der boolesche Wert muss 1 (wahr) oder 0 (falsch) sein. Weitere Informationen finden Sie unter [Undefinierte, unendliche und Überlaufwerte](#).

- Zeitintervall — Das metrische Zeitintervall. AWS IoT SiteWise unterstützt die folgenden Zeitintervalle im Taumelfenster, wobei jedes Intervall beginnt, wenn das vorherige endet:
 - 1 Minute — 1 Minute, berechnet am Ende jeder Minute (00:00:00 Uhr, 12:01:00 Uhr, 12:02:00 Uhr usw.).
 - 5 Minuten — 5 Minuten, berechnet am Ende aller fünf Minuten, beginnend mit der vollen Stunde (00:00:00 Uhr, 12:05:00 Uhr, 00:10:00 Uhr usw.).
 - 15 Minuten — 15 Minuten, berechnet am Ende aller fünfzehn Minuten, beginnend mit der vollen Stunde (00:00:00 Uhr, 00:15:00 Uhr, 12:30:00 Uhr usw.).
 - 1 Stunde — 1 Stunde (60 Minuten), berechnet am Ende jeder Stunde in UTC (12:00:00 Uhr, 01:00:00 Uhr, 02:00:00 Uhr usw.).
 - 1 Tag — 1 Tag (24 Stunden), berechnet am Ende eines jeden Tages in UTC (Montag 12:00:00 Uhr, Dienstag 12:00:00 Uhr usw.).

- 1 Woche — 1 Woche (7 Tage), berechnet am Ende jedes Sonntags in UTC (jeden Montag um 00:00:00 Uhr).
- Benutzerdefiniertes Intervall — Sie können ein beliebiges Zeitintervall zwischen einer Minute und einer Woche eingeben.
- Offsetdatum — (Optional) Das Referenzdatum, ab dem Daten aggregiert werden sollen.
- Offsetzeit — (Optional) Die Referenzzeit, ab der Daten aggregiert werden sollen. Die Offsetzeit muss zwischen 00:00:00 und 23:59:59 liegen.
- Offset-Zeitzone — (Optional) Die Zeitzone für den Offset. Wenn sie nicht angegeben ist, ist die standardmäßige Offset-Zeitzone die koordinierte Weltzeit (UTC). Sehen Sie sich die folgenden unterstützten Zeitzone an.

Unterstützte Zeitzone

- (UTC+ 00:00) Koordinierte Weltzeit
- (UTC+ 01:00) Europäische Zentralzeit
- (UTC+ 02:00) Osteuropäische
- (UTC+03+:00) Ostafrikanische Zeit
- (UTC+ 04:00) Nahöstliche Zeit
- (UTC+ 05:00) Pakistan Lahore-Zeit
- (UTC+ 05:30) Indien Normalzeit
- (UTC+ 06:00) Normalzeit in Bangladesch
- (UTC+ 07:00) Vietnam Normalzeit
-
- (UTC+ 09:00) Japan Normalzeit
- (UTC+ 09:30) Australien Zentralzeit
- (UTC+ 10:00) Australien Ostzeit
- (UTC+ 11:00) Salomonische Normalzeit
- (UTC+ 12:00) Neuseeland Normalzeit
- (UTC-11:00) Zeit auf den Midway-Inseln
- (UTC-10:00) Hawaii-Standardzeit
- (UTC-09:00) Alaska-Standardzeit

- (UTC-08:00) Pazifische Standardzeit
- (UTC-07:00) Phoenix-Standardzeit
- (UTC-06:00) Zentrale Standardzeit
- (UTC-05:00) Östliche Standardzeit
- (UTC-04:00) Zeit in Puerto Rico und den Amerikanischen Jungferninseln
- (UTC-03:00) Argentinien Normalzeit
- (UTC-02:00) Südgeorgische Zeit
- (UTC-01:00) Zentralafrikanische Zeit

Example benutzerdefiniertes Zeitintervall mit einem Offset (Konsole)

Das folgende Beispiel zeigt Ihnen, wie Sie ein 12-Stunden-Zeitintervall mit einem Offset am 20. Februar 2021 um 18:30:30 Uhr (PST) definieren.

Um ein benutzerdefiniertes Intervall mit einem Offset zu definieren

1. Wählen Sie für Zeitintervall die Option Benutzerdefiniertes Intervall aus.
2. Führen Sie für Zeitintervall einen der folgenden Schritte aus:
 - Geben Sie Stunden ein **12**, und wählen Sie dann aus.
 - Geben Sie ein **720**, und wählen Sie dann Minuten aus.
 - Geben Sie ein **43200**, und wählen Sie dann Sekunden.

Important

Das Zeitintervall muss unabhängig von der Einheit eine Ganzzahl sein.

3. Wählen Sie für Offset-Datum die Option 2021/02/20 aus.
4. Geben **18:30:30** Sie für Offsetzeit den Wert ein.
5. Wählen Sie für Offset-Zeitzone die Option (UTC-08:00) Pacific Standard Time.

Wenn Sie die Metrik am 1. Juli 2021 vor oder um 18:30 Uhr (PST) erstellen, erhalten Sie das erste Aggregationsergebnis am 1. Juli 2021 um 18:30 Uhr (PST). Das zweite Aggregationsergebnis wird am 2. Juli 2021 um 06:30:30 Uhr (PST) usw. angezeigt.

Definieren Sie Metriken (AWS CLI)

Wenn Sie mit der AWS IoT SiteWise API eine Metrik für ein Asset-Modell definieren, geben Sie die folgenden Parameter an:

- **name**— Der Name der Immobilie.
- **dataType**— Der Datentyp der Metrik, der `DOUBLE` oder sein kann `STRING`.
- **externalId**— (Optional) Dies ist eine benutzerdefinierte ID. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.
- **expression**— Der metrische Ausdruck. Metrische Ausdrücke können [Aggregationsfunktionen](#) verwenden, um Daten aus einer Eigenschaft für alle zugehörigen Anlagen in einer Hierarchie einzugeben. Weitere Informationen finden Sie unter [Verwenden Sie Formelausdrücke](#).
- **window**— Das Zeitintervall und der Offset für das Taumelfenster der Metrik, wobei jedes Intervall beginnt, wenn das vorherige endet:
 - **interval**— Das Zeitintervall für das Taumelfenster. Das Zeitintervall muss zwischen einer Minute und einer Woche liegen.
 - **offsets**— Der Offset für das Taumelfenster.

Weitere Informationen finden Sie unter [TumblingWindow](#) in der AWS IoT SiteWise-API-Referenz.

Example benutzerdefiniertes Zeitintervall mit einem Offset (AWS CLI)

Das folgende Beispiel zeigt Ihnen, wie Sie ein 12-Stunden-Zeitintervall mit einem Offset am 20. Februar 2021 um 18:30:30 Uhr (PST) definieren.

```
{
  "window": {
    "tumbling": {
      "interval": "12h",
      "offset": " 2021-07-23T18:30:30-08"
    }
  }
}
```

Wenn Sie die Metrik am 1. Juli 2021 vor oder um 18:30 Uhr (PST) erstellen, erhalten Sie das erste Aggregationsergebnis am 1. Juli 2021 um 18:30 Uhr (PST). Das zweite Aggregationsergebnis wird am 2. Juli 2021 um 06:30:30 Uhr (PST) usw. angezeigt.

- **variables**— Die Variablenliste, die die anderen Eigenschaften Ihrer Anlage oder Ihrer untergeordneten Anlagen definiert, die in dem Ausdruck verwendet werden sollen. Jede Variablenstruktur enthält einen einfachen Namen, der in dem Ausdruck verwendet werden soll, sowie eine `value`-Struktur zur Identifizierung der mit dieser Variablen zu verknüpfenden Eigenschaft. Die `value`-Struktur enthält folgende Informationen:
 - **propertyId**— Die ID der Eigenschaft, aus der Werte abgerufen werden sollen. Sie können den Namen der Eigenschaft anstelle der ID verwenden, wenn die Eigenschaft im aktuellen Modell (und nicht in einem Modell aus einer Hierarchie) definiert ist.
 - **hierarchyId**— (Optional) Die ID der Hierarchie, aus der untergeordnete Vermögenswerte für die Eigenschaft abgefragt werden sollen. Sie können den Namen der Hierarchiedefinition anstelle der ID verwenden. Wenn Sie diesen Wert weglassen, AWS IoT SiteWise wird die Eigenschaft im aktuellen Modell gesucht.

Important

Bei Metriken kann es sich nur um Eigenschaften vom Typ Integer, Double, Boolean oder Zeichenfolge handeln. Boolesche Werte werden in 0 (falsch) und 1 (wahr) konvertiert. Wenn Sie Metrikeingabevariablen im Ausdruck einer Metrik definieren, muss für diese Eingaben dasselbe Zeitintervall wie für die Ausgabemetrik gelten. Formelausdrücke können nur Doppelwerte oder Zeichenkettenwerte ausgeben. Verschachtelte Ausdrücke können andere Datentypen ausgeben, z. B. Zeichenfolgen, aber die Formel als Ganzes muss eine Zahl oder Zeichenfolge ergeben. Sie können die [Funktion jp](#) verwenden, um eine Zeichenfolge in eine Zahl umzuwandeln. Der boolesche Wert muss 1 (wahr) oder 0 (falsch) sein. Weitere Informationen finden Sie unter [Undefinierte, unendliche und Überlaufwerte](#).

- **unit**— (Optional) Die wissenschaftliche Einheit für die Eigenschaft, z. B. mm oder Celsius.

Example Beispiel für eine Metrik-Definition

Das folgende Beispiel zeigt eine Metrikeigenschaft, die die Temperaturmessdaten einer Komponente aggregiert, um die maximale Durchschnittstemperatur in Fahrenheit zu berechnen. Dieses Objekt ist ein Beispiel für ein Objekt [AssetModelProperty](#), das eine [Metrik](#) enthält. Sie können dieses Objekt als Teil der [CreateAssetModel](#)-Anforderungs-Nutzlast angeben, um eine Metrik-Eigenschaft zu erstellen. Weitere Informationen finden Sie unter [Erstellen Sie ein Asset-Modell \(AWS CLI\)](#).


```

{
    ...
    "assetModelProperties": [
        ...
        {
            "name": "Max temperature",
            "dataType": "DOUBLE",
            "type": {
                "metric": {
                    "expression": "max(temp_f)",
                    "variables": [
                        {
                            "name": "temp_f",
                            "value": {
                                "propertyId": "Temperature F"
                            }
                        }
                    ],
                    "window": {
                        "tumbling": {
                            "interval": "1h"
                        }
                    }
                }
            },
            "unit": "Fahrenheit"
        }
    ],
    ...
}

```

Example Beispiel für eine Metrikdefinition, die Daten aus zugehörigen Anlagen eingibt

Das folgende Beispiel zeigt eine metrische Eigenschaft, die die durchschnittlichen Leistungsdaten mehrerer Windturbinen aggregiert, um die durchschnittliche Gesamtleistung für einen Windpark zu berechnen. [Dieses Objekt ist ein Beispiel für ein Objekt AssetModelProperty, das eine Metrik enthält.](#) Sie können dieses Objekt als Teil der [CreateAssetModel](#)-Anforderungs-Nutzlast angeben, um eine Metrik-Eigenschaft zu erstellen.

```

{
    ...
    "assetModelProperties": [
        ...

```

```

{
  "name": "Total Average Power",
  "dataType": "DOUBLE",
  "type": {
    "metric": {
      "expression": "avg(power)",
      "variables": [
        {
          "name": "power",
          "value": {
            "propertyId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "hierarchyId": "Turbine Asset Model"
          }
        }
      ],
      "window": {
        "tumbling": {
          "interval": "5m"
        }
      }
    },
    "unit": "kWh"
  },
  ...
}

```

Verwenden Sie FormelAusdrücke

Mit FormelAusdrücken können Sie die mathematischen Funktionen definieren, um Ihre industriellen Rohdaten zu transformieren und zu aggregieren, sodass Sie Einblicke in Ihre Operation gewinnen. FormelAusdrücke kombinieren Literale, Operatoren, Funktionen und Variablen, um Daten zu verarbeiten. Weitere Informationen zur Definition von Asset-Eigenschaften, die FormelAusdrücke verwenden, finden Sie unter [Daten transformieren \(transformiert\)](#) und [Aggregieren Sie Daten aus Immobilien und anderen Vermögenswerten \(Metriken\)](#). Transformationen und Metriken sind Formeleigenschaften.

Themen

- [Verwenden Sie Variablen in FormelAusdrücken](#)
- [Verwenden Sie Literale in FormelAusdrücken](#)

- [Verwenden Sie Operatoren in Formelausdrücken](#)
- [Verwenden Sie Konstanten in Formelausdrücken](#)
- [Verwenden Sie Funktionen in Formelausdrücken](#)
- [Tutorials zu Formelausdrücken](#)

Verwenden Sie Variablen in Formelausdrücken

Variablen stellen AWS IoT SiteWise Asset-Eigenschaften in Formelausdrücken dar. Verwenden Sie Variablen, um Werte aus anderen Objekteigenschaften in Ihre Ausdrücke einzugeben, sodass Sie Daten aus konstanten Eigenschaften ([Attributen](#)), Rohdatenströmen ([Messungen](#)) und anderen Formeleigenschaften verarbeiten können.

Variablen können Asset-Eigenschaften aus demselben Asset-Modell oder aus zugehörigen untergeordneten Asset-Modellen darstellen. Nur metrische Formeln können Variablen aus untergeordneten Vermögensmodellen eingeben.

Sie identifizieren Variablen in der Konsole und in der API mit unterschiedlichen Namen.

- AWS IoT SiteWiseKonsole — Verwenden Sie die Namen von Asset-Eigenschaften als Variablen in Ihren Ausdrücken.
- AWS IoT SiteWiseAPI (AWS CLI, AWS SDKs) — Definieren Sie Variablen mit der [ExpressionVariable](#)Struktur, die einen Variablennamen und einen Verweis auf eine Asset-Eigenschaft erfordert. Der Variablenname kann Kleinbuchstaben, Zahlen und Unterstriche enthalten. Verwenden Sie dann Variablennamen, um in Ihren Ausdrücken auf Asset-Eigenschaften zu verweisen.

Bei Variablennamen wird zwischen Groß- und Kleinschreibung unterschieden.

Weitere Informationen finden Sie unter [Transformationen definieren](#) und [Metriken definieren](#).

Verwenden Sie Variablen, um auf Eigenschaften zu verweisen

Der Wert einer Variablen definiert die Eigenschaft, auf die sie verweist. AWS IoT SiteWisebietet verschiedene Möglichkeiten, dies zu tun.

- Nach Eigenschafts-ID: Sie können die eindeutige ID (UUID) der Immobilie angeben, um sie zu identifizieren.

- Nach Namen: Wenn sich die Immobilie auf demselben Objektmodell befindet, können Sie ihren Namen im Feld Eigenschafts-ID angeben.
- Nach Pfad: Ein Variablenwert kann anhand seines Pfads auf eine Eigenschaft verweisen. Weitere Informationen finden Sie unter [Verwenden Sie Pfade, um auf benutzerdefinierte Eigenschaften von Verbundmodellen zu verweisen](#).

Note

Variablen werden von der AWS IoT SiteWise Konsole nicht unterstützt. Sie werden von der AWS IoT SiteWise API (einschließlich der AWS Command Line Interface AWS CLI) und AWS SDKs verwendet.

Eine Variable, von der Sie in einer Antwort erhalten, AWS IoT SiteWise enthält vollständige Informationen über den Wert, einschließlich der ID und des Pfads.

Wenn Sie jedoch eine Variable an übergeben AWS IoT SiteWise (z. B. bei einem „create“ - oder „update“-Aufruf), müssen Sie nur eine dieser Variablen angeben. Wenn Sie beispielsweise den Pfad angeben, müssen Sie die ID nicht angeben.

Verwenden Sie Literale in Formelausdrücken

AWS IoT SiteWise unterstützt die Verwendung von Literalen in Ausdrücken und Formeln. Literale sind feste Werte, die einen bestimmten Datentyp repräsentieren. In AWS IoT SiteWise können Sie Zahlen- und Zeichenkettenliterals in Formelausdrücken definieren. Literale können in verschiedenen Kontexten verwendet werden, einschließlich Datentransformationen, Alarmbedingungen und Visualisierungsberechnungen.

- Zahlen

Verwenden Sie Zahlen und wissenschaftliche Schreibweise, um ganze Zahlen und Doppelzahlen zu definieren. Sie können die [E-Notation](#) verwenden, um Zahlen in wissenschaftlicher Schreibweise auszudrücken.

Beispiele: 12.0, .9, -23.1, 7.89e3, 3.4E-5

- Zeichenfolgen

Verwenden Sie die Zeichen ' (Anführungszeichen) und " (doppelte Anführungszeichen), um Zeichenketten zu definieren. Der Zitattyp für Anfang und Ende muss übereinstimmen. Um ein Anführungszeichen zu maskieren, das dem entspricht, das Sie zur Deklaration einer Zeichenfolge verwenden, fügen Sie dieses Anführungszeichen zweimal ein. Dies ist das einzige Escape-Zeichen in AWS IoT SiteWise Zeichenketten.

Beispiele: 'active', "inactive", '{"temp": 52}', '{"temp": "high"}'

Verwenden Sie Operatoren in Formelausdrücken

Sie können die folgenden gängigen Operatoren in Formelausdrücken verwenden.

Operator	Description
+	Wenn beide Operanden Zahlen sind, addiert dieser Operator den linken und den rechten Operanden. Wenn einer der Operanden eine Zeichenfolge ist, verkettet dieser Operator den linken und den rechten Operanden als Zeichenketten. Der Ausdruck wird beispielsweise zu ausgewertet <code>1 + 2 + " is three"</code> <code>"3 is three"</code> Die verkettete Zeichenfolge kann bis zu 1024 Zeichen enthalten. Wenn die Zeichenfolge 1024 Zeichen überschreitet, wird AWS IoT SiteWise kein Datenpunkt für diese Berechnung ausgegeben.
-	Subtrahiert den rechten Operanden vom linken Operanden Sie können diesen Operator nur mit numerischen Operanden verwenden.
/	Dividiert den linken Operanden durch den rechten Operanden

Operator	Description
	Sie können diesen Operator nur mit numerischen Operanden verwenden.
*	Multipliziert die linken und rechten Operanden. Sie können diesen Operator nur mit numerischen Operanden verwenden.
^	Hebt den linken Operanden auf die Potenz des rechten Operanden (Exponentiation). Sie können diesen Operator nur mit numerischen Operanden verwenden.
%	Gibt den Rest zurück, der beim Dividieren des linken Operanden durch den rechten Operanden entsteht. Das Ergebnis hat das gleiche Zeichen wie der linke Operand. Dieses Verhalten unterscheidet sich von der Modulo-Operation. Sie können diesen Operator nur mit numerischen Operanden verwenden.
$x < y$	Gibt zurück 1, wenn kleiner als x ist, andernfalls 0.
$x > y$	Gibt zurück 1, wenn größer als x ist, andernfalls 0.
$x \leq y$	Gibt zurück 1, ob kleiner oder gleich x ist, andernfalls 0.
$x \geq y$	Gibt zurück 1, ob größer als oder gleich x ist, andernfalls 0.
$x == y$	Gibt zurück 1, ob gleich x ist, andernfalls 0.

Operator	Description
$x \neq y$	Gibt zurück 1, wenn nicht gleich x ist, andernfalls 0.
! x	Gibt zurück 1, ob als 0 (falsch) ausgewertet x wird, andernfalls 0. x wird als falsch bewertet, wenn: • x ist ein numerischer Operand und wird als ausgewertet. 0 • x wird als leere Zeichenfolge ausgewertet. • x wird als leeres Array ausgewertet. • x wird als ausgewertet None.
x and y	Gibt zurück 0, ob als 0 (falsch) ausgewertet x wird. Andernfalls wird das ausgewertete Ergebnis von zurückgegeben y. x oder y wird als falsch bewertet, wenn: • x oder y ist ein numerischer Operand und wird als ausgewertet. 0 • x oder y wird als leere Zeichenfolge ausgewertet. • x oder y wird als leeres Array ausgewertet. • x oder y wird als ausgewertet None.

Operator	Description
<code>x or y</code>	Gibt zurück 1, ob als 1 (wahr) ausgewertet <code>x</code> wird. Andernfalls wird das ausgewertete Ergebnis von zurückgegeben <code>y</code>. <code>x</code> oder <code>y</code> wird als falsch bewertet, wenn: • <code>x</code> oder <code>y</code> ist ein numerischer Operand und wird als ausgewertet. 0 • <code>x</code> oder <code>y</code> wird als leere Zeichenfolge ausgewertet. • <code>x</code> oder <code>y</code> wird als leeres Array ausgewertet. • <code>x</code> oder <code>y</code> wird als ausgewertetNone.
<code>not x</code>	Gibt zurück 1, ob als 0 (falsch) ausgewertet <code>x</code> wird, andernfalls 0. <code>x</code> wird als falsch bewertet, wenn: • <code>x</code> ist ein numerischer Operand und wird als ausgewertet. 0 • <code>x</code> wird als leere Zeichenfolge ausgewertet. • <code>x</code> wird als leeres Array ausgewertet. • <code>x</code> wird als ausgewertetNone.
<code>[]</code> <code>s[index]</code>	Gibt das Zeichen an einem Index <code>index</code> der Zeichenfolge zurück. Dies entspricht der Indexsyntax in Python. Example Beispiele • <code>"Hello!"[1]</code> gibt e zurück. • <code>"Hello!"[-2]</code> gibt o zurück.

Operator	Description
[] s[start:end:step]	Gibt einen Teil der Zeichenfolge zurück. Dies entspricht der Slice-Syntax in Python. Dieser Operator hat die folgenden Argumente: • start— (Optional) Der inklusive Startindex des Slice. Standardeinstellung: 0. • end— (Optional) Der exklusive Endindex des Slice. Standardmäßig wird die Länge der Zeichenfolge verwendet. • step— (Optional) Die Zahl, die für jeden Schritt im Slice erhöht werden soll. Sie können beispielsweise angeben 2, dass ein Segment mit jedem zweiten Zeichen zurückgegeben werden soll, oder Sie können angeben, -1 dass das Segment umgekehrt werden soll. Standardeinstellung: 1. Sie können das <code>step</code> Argument weglassen, um seinen Standardwert zu verwenden. Beispiel: <code>s[1:4:1]</code> ist gleichbedeutend mit <code>s[1:4]</code>. Bei den Argumenten muss es sich um ganze Zahlen oder um die Konstante None handeln. Wenn Sie angeben <code>none</code>, AWS IoT SiteWise wird der Standardwert für dieses Argument verwendet. Example Beispiele • <code>"Hello!"[1:4]</code> gibt <code>"ell"</code> zurück. • <code>"Hello!"[:2]</code> gibt <code>"He"</code> zurück. • <code>"Hello!"[3:]</code> gibt <code>"lo!"</code> zurück. • <code>"Hello!"[:-4]</code> gibt <code>"He"</code> zurück. • <code>"Hello!"[:2]</code> gibt <code>"Hl"</code> zurück.

Operator	Description
	"Hello!"[::-1] gibt "!olleH" zurück.

Verwenden Sie Konstanten in Formelausdrücken

In AWS IoT SiteWise können Sie Konstanten in Ihren Ausdrücken und Formeln verwenden, um feste Werte oder vordefinierte Parameter darzustellen. Konstanten können in verschiedenen Kontexten verwendet werden, z. B. bei Datentransformationen, Alarmbedingungen oder Visualisierungsberechnungen. Durch die Verwendung von Konstanten können Sie Ihre Ausdrücke vereinfachen und sie lesbarer und verwaltbarer machen.

Sie können die folgenden allgemeinen mathematischen Konstanten in Ihren Ausdrücken verwenden. Bei allen Konstanten wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Note

Wenn Sie eine Variable mit demselben Namen wie eine Konstante definieren, überschreibt die Variable die Konstante.

Konstante	Description
pi	Die Zahl pi (π): 3.141592653589793
e	Die Zahl e: 2.718281828459045
true	Entspricht der Zahl 1. In werden AWS IoT SiteWise Boolesche Werte in ihre Zahlenäquivalente umgewandelt.
false	Entspricht der Zahl 0. In werden AWS IoT SiteWise Boolesche Werte in ihre Zahlenäquivalente umgewandelt.
none	Entspricht keinem Wert. Sie können diese Konstante verwenden, um nichts als Ergebnis eines bedingten Ausdrucks auszugeben.

Verwenden Sie Funktionen in Formelausdrücken

Sie können die folgenden Funktionen verwenden, um mit Daten in Ihren Formelausdrücken zu arbeiten.

Transformationen und Metriken unterstützen verschiedene Funktionen. Die folgende Tabelle zeigt, welche Funktionstypen mit den einzelnen Typen von Formeleigenschaften kompatibel sind.

Note

Sie können maximal 10 Funktionen in einen Formelausdruck aufnehmen.

Typ der Funktion	Transformiert	Kennzahlen
Verwenden Sie allgemeine Funktionen in Formelausdrücken	✔ Ja	✔ Ja
Verwenden Sie Vergleichsfunktionen in Formelausdrücken	✔ Ja	✔ Ja
Verwenden Sie bedingte Funktionen in Formelausdrücken	✔ Ja	✔ Ja
Verwenden Sie Zeichenkettenfunktionen in Formelausdrücken	✔ Ja	✔ Ja
Verwenden Sie Aggregationsfunktionen in Formelausdrücken	✘ Nein	✔ Ja
Verwenden Sie temporale Funktionen in Formelausdrücken	✔ Ja	✔ Ja

Typ der Funktion	Transformiert	Kennzahlen
Verwenden Sie Datums- und Uhrzeitfunktionen in Formelausdrücken	✔ Ja	✔ Ja

Syntax der Funktion

Sie können die folgende Syntax verwenden, um Funktionen zu erstellen:

Reguläre Syntax

Bei der regulären Syntax folgen auf den Funktionsnamen Klammern mit null oder mehr Argumenten.

function_name(argument1, argument2, argument3, ...). Funktionen mit der regulären Syntax könnten beispielsweise wie `log(x)` und `aussehencontains(s, substring)` aussehen.

Einheitliche Syntax für Funktionsaufrufe (UFCS)

Mit UFCS können Sie Funktionen mithilfe der Syntax für Methodenaufrufen in der objektorientierten Programmierung aufrufen. Bei UFCS folgt auf das erste Argument ein Punkt (`.`), dann der Funktionsname und die verbleibenden Argumente (falls vorhanden) in Klammern.

argument1.function_name(argument2, argument3, ...). Funktionen mit UFCS könnten beispielsweise wie `x.log()` und `s.contains(substring)` aussehen.

Sie können UFCS auch verwenden, um nachfolgende Funktionen zu verketteten. AWS IoT SiteWise verwendet das Auswertungsergebnis der aktuellen Funktion als erstes Argument für die nächste Funktion.

Sie können beispielsweise `message.jp('$ .status').lower().contains('fail')` anstelle von `verwendencontains(lower(jp(message, '$ .status')), 'fail')` verwenden.

Weitere Informationen finden Sie auf der Website der [Programmiersprache D](#).

Note

Sie können UFCS für alle AWS IoT SiteWise Funktionen verwenden.

AWS IoT SiteWiseBei Funktionen wird nicht zwischen Groß- und Kleinschreibung unterschieden. Sie können „lower(s) und“ beispielsweise Lower(s) synonym verwenden.

Verwenden Sie allgemeine Funktionen in Formelausdrücken

In [Transformationen](#) und [Metriken](#) können Sie die folgenden Funktionen verwenden, um allgemeine mathematische Funktionen in Transformationen und Metriken zu berechnen.

Funktion	Description
<code>abs(x)</code>	Gibt den absoluten Wert von x zurück.
<code>acos(x)</code>	Gibt den Arkuskosinus von x zurück.
<code>asin(x)</code>	Gibt den Arkussinus von x zurück.
<code>atan(x)</code>	Gibt den Arkustangens von x zurück.
<code>cbrt(x)</code>	Gibt die Kubikwurzel von x zurück.
<code>ceil(x)</code>	Gibt die nächste Ganzzahl zurück, die größer als x ist.
<code>cos(x)</code>	Gibt den Kosinus von x zurück.
<code>cosh(x)</code>	Gibt den hyperbolischen Kosinus von x zurück.
<code>cot(x)</code>	Gibt den Kotangens von zurück. x
<code>exp(x)</code>	Gibt e hoch x zurück.
<code>expm1(x)</code>	Gibt $\exp(x) - 1$ zurück. Verwenden Sie diese Funktion, um kleinere Werte von genauer $\exp(x) - 1$ zu berechnen. x
<code>floor(x)</code>	Gibt die nächste ganze Zahl zurück, die kleiner als x ist.
<code>log(x)</code>	Gibt $\log_e$ (Basis e) von x zurück.

Funktion	Description
<code>log10(x)</code>	Gibt $\log_{10}$ (Basis 10) von x zurück.
<code>log1p(x)</code>	Gibt $\log(1 + x)$ zurück. Verwenden Sie diese Funktion, um kleinere Werte von genauer zu berechnen $\log(1 + x)x$.
<code>log2(x)</code>	Gibt $\log_2$ (Basis 2) von x zurück.
<code>pow(x, y)</code>	Gibt x hoch y zurück. Das entspricht x^y .
<code>signum(x)</code>	Gibt das Vorzeichen von x (-1 für negative Eingaben, 0 für Nulleingaben, +1 für positive Eingaben) zurück.
<code>sin(x)</code>	Gibt den Sinus von x zurück.
<code>sinh(x)</code>	Gibt den hyperbolischen Sinus von x zurück.
<code>sqrt(x)</code>	Gibt die Quadratwurzel von x zurück.
<code>tan(x)</code>	Gibt den Tangens von x zurück.
<code>tanh(x)</code>	Gibt den hyperbolischen Tangens von x zurück.

Verwenden Sie Vergleichsfunktionen in FormelAusdrücken

In [Transformationen](#) und [Metriken](#) können Sie die folgenden Vergleichsfunktionen verwenden, um zwei Werte zu vergleichen und 1 (wahr) oder 0 (falsch) auszugeben. AWS IoT SiteWise vergleicht Zeichenketten in [lexikografischer](#) Reihenfolge.

Funktion	Description
<code>gt(x, y)</code>	Gibt 1 zurück, wenn x größer als y ist, andernfalls 0 ($x > y$).

Funktion	Description
	Diese Funktion gibt keinen Wert zurück, wenn x es sich um inkompatible Typen y handelt, z. B. eine Zahl und eine Zeichenfolge.
<code>gte(x, y)</code>	Gibt 1 zurück, wenn x größer oder gleich y ist, andernfalls 0 ($x \geq y$). AWS IoT SiteWise betrachtet die Argumente als gleich, wenn sie innerhalb einer relativen Toleranz von $1E-9$ liegen. Dies verhält sich ähnlich wie die Funktion isclose in Python. Diese Funktion gibt keinen Wert zurück, wenn x es sich um inkompatible Typen y handelt, z. B. eine Zahl und eine Zeichenfolge.
<code>eq(x, y)</code>	Gibt 1 zurück, wenn x gleich y ist, andernfalls 0 ($x == y$). AWS IoT SiteWise betrachtet die Argumente als gleich, wenn sie innerhalb einer relativen Toleranz von $1E-9$ liegen. Dies verhält sich ähnlich wie die Funktion isclose in Python. Diese Funktion gibt keinen Wert zurück, wenn x es sich um inkompatible Typen y handelt, z. B. eine Zahl und eine Zeichenfolge.
<code>lt(x, y)</code>	Gibt 1 zurück, wenn x kleiner als y ist, andernfalls 0 ($x < y$). Diese Funktion gibt keinen Wert zurück, wenn x es sich um inkompatible Typen y handelt, z. B. eine Zahl und eine Zeichenfolge.

Funktion	Description
<code>lte(x, y)</code>	Gibt 1 zurück, wenn x kleiner oder gleich y ist, andernfalls 0 ($x \leq y$). AWS IoT SiteWise betrachtet die Argumente als gleich, wenn sie innerhalb einer relativen Toleranz von $1E-9$ liegen. Dies verhält sich ähnlich wie die Funktion isclose in Python. Diese Funktion gibt keinen Wert zurück, wenn x es sich um inkompatible Typen y handelt, z. B. eine Zahl und eine Zeichenfolge.
<code>isnan(x)</code>	Gibt zurück 1, ob gleich x ist NaN, andernfalls 0. Diese Funktion gibt keinen Wert zurück, wenn x es sich um eine Zeichenfolge handelt.

Verwenden Sie bedingte Funktionen in Formelausdrücken

In [Transformationen](#) und [Metriken](#) können Sie die folgende Funktion verwenden, um eine Bedingung zu überprüfen und unterschiedliche Ergebnisse zurückzugeben, unabhängig davon, ob die Bedingung als wahr oder falsch ausgewertet wird.

Funktion	Description
<code>if(condition, result_if_true, result_if_false)</code>	Wertet das aus <code>condition</code> und gibt zurück, <code>result_if_true</code> ob die Bedingung als wahr ausgewertet wird oder <code>result_if_false</code> ob die Bedingung als falsch ausgewertet wird. <code>condition</code> muss eine Zahl sein. Diese Funktion betrachtet 0 eine leere Zeichenfolge als <code>false</code> und alles andere (einschließlich NaN) als <code>true</code>. Boolesche Werte werden in 0 (falsch) und 1 (wahr) umgewandelt.

Funktion	Description
	Sie können die Konstante none aus dieser Funktion zurückgeben, um die Ausgabe für eine bestimmte Bedingung zu verwerfen. Das bedeutet, dass Sie Datenpunkte herausfiltern können, die eine Bedingung nicht erfüllen. Weitere Informationen finden Sie unter Datenpunkte filtern. Example Beispiele • <code>if(0, x, y)</code> gibt die Variable zurücky. • <code>if(5, x, y)</code> gibt die Variable zurückx. • <code>if(gt(temp, 300), x, y)</code> gibt die Variable zurückx, wenn die Variable größer als temp ist300. • <code>if(gt(temp, 300), temp, none)</code> gibt die Variable zurück, temp wenn sie größer oder gleich ist300, oder none (kein Wert), wenn sie kleiner als temp ist300. Es wird empfohlen, UFCS für verschachtelte bedingte Funktionen zu verwenden, bei denen es sich bei einem oder mehreren Argumenten um bedingte Funktionen handelt. Sie können <code>if(condition, result_if_true)</code> es verwenden, um eine Bedingung und zusätzliche Bedingungen <code>elif(condition, result_if_true, result_if_false)</code> auszuwerten. Sie können beispielsweise <code>if(condition1, result1_if_true).elif(condition2, result2_if_true, result2_if_false)</code> anstelle von verwenden <code>if(condition1, result1_if_true,</code>

Funktion	Description
	<pre>if(condition2, result2_if_true, result2_if_false)) .</pre> Sie können auch zusätzliche bedingte Zwischenfunktionen verketten. Sie können beispielsweise mehrere if Anweisungen verwenden, <code>if(condition1, result1_if_true).elif(condition2, result2_if_true).elif(condition3, result3_if_true, result3_if_false)</code> anstatt sie zu verschachteln, wie <code>if(condition1, result1_if_true, if(condition2, result2_if_true, if(condition3, result3_if_true, result3_if_false)))</code> z.  Important Sie müssen es <code>elif(condition, result_if_true, result_if_false)</code> mit UFCS verwenden.

Verwenden Sie Zeichenkettenfunktionen in Formelausdrücken

In [Transformationen](#) und [Metriken](#) können Sie die folgenden Funktionen verwenden, um mit Zeichenketten zu arbeiten. Weitere Informationen finden Sie unter [Verwenden Sie Zeichenketten in Formeln](#).

 Important

Formelausdrücke können nur Doppel- oder Zeichenkettenwerte ausgeben. Verschachtelte Ausdrücke können andere Datentypen ausgeben, z. B. Zeichenfolgen, aber die Formel als Ganzes muss eine Zahl oder Zeichenfolge ergeben. Sie können die [Funktion jp](#) verwenden, um eine Zeichenfolge in eine Zahl umzuwandeln. Der boolesche Wert muss 1 (wahr)

oder 0 (falsch) sein. Weitere Informationen finden Sie unter [Undefinierte, unendliche und Überlaufwerte](#).

Funktion	Description
<code>len(s)</code>	Gibt die Länge der Zeichenfolge zurück. <code>s</code>
<code>find(s, substring)</code>	Gibt den Index der Zeichenfolge <code>substring</code> in der Zeichenfolge zurück.
<code>contains(s, substring)</code>	Gibt zurück 1, ob die Zeichenfolge die Zeichenfolge <code>s</code> enthält <code>substring</code> , andernfalls 0.
<code>upper(s)</code>	Gibt die Zeichenfolge <code>s</code> in Großbuchstaben zurück.
<code>lower(s)</code>	Gibt die Zeichenfolge <code>s</code> in Kleinbuchstaben zurück.
<code>jp(s, json_path)</code>	Wertet die Zeichenfolge <code>s</code> mit dem JsonPath-Ausdruck aus <code>json_path</code> und gibt das Ergebnis zurück. Verwenden Sie diese Funktion, um Folgendes zu tun: • Extrahieren Sie einen Wert, ein Array oder ein Objekt aus einer serialisierten JSON-Struktur. • Konvertiert eine Zeichenfolge in eine Zahl. Die Formel <code>jp('111', '\$')</code> gibt 111 beispielsweise eine Zahl zurück. Um einen Zeichenkettenwert aus einer JSON-Struktur zu extrahieren und ihn als Zahl zurückzugeben, müssen Sie mehrere verschachtelte <code>jp</code> Funktionen verwenden. Die

Funktion	Description
	äußere jp Funktion extrahiert die Zeichenfolge aus der JSON-Struktur, und die innere jp Funktion konvertiert die Zeichenfolge in eine Zahl. Die Zeichenfolge <code>json_path</code> muss ein Zeichenfolgenliteral enthalten. Das bedeutet, dass es <code>json_path</code> sich nicht um einen Ausdruck handeln kann, der zu einer Zeichenfolge ausgewertet wird. Example Beispiele • <code>jp('{"status":"active","value":15}', '\$.value')</code> gibt 15 zurück. • <code>jp('{"measurement":{"reading":25,"confidence":0.95}}', '\$.measurement.reading')</code> gibt 25 zurück. • <code>jp('[2,8,23]', '\$[2]')</code> gibt 23 zurück. • <code>jp('{"values":[3,6,7]}', '\$.values[1]')</code> gibt 6 zurück. • <code>jp('111', '\$')</code> gibt 111 zurück. • <code>jp(jp('{"measurement":{"reading":25,"confidence":"0.95"}}', '\$.measurement.confidence'), '\$')</code> gibt 0.95 zurück.

Funktion	Description
<code>join(s0, s1, s2, s3, ...)</code>	Gibt eine verkettete Zeichenfolge mit einem Trennzeichen zurück. Diese Funktion verwendet die erste Eingabezeichenfolge als Trennzeichen und verbindet die verbleibenden Eingabezeichenfolgen miteinander. Dies verhält sich ähnlich wie die Funktion join (CharSequence delimiter, CharSequence... elements) in Java. Example Beispiele • <code>join("-", "aa", "bb", "cc")</code> gibt zurück <code>aa-bb-cc</code>
<code>format(expression: "format")</code> oder <code>format("format", expression)</code>	Gibt eine Zeichenfolge im angegebenen Format zurück. Diese Funktion ergibt <code>expression</code> einen Wert und gibt den Wert dann im angegebenen Format zurück. Dies verhält sich ähnlich wie die Funktion format (String-Format, Object... args) in Java. Weitere Informationen zu unterstützten Formaten finden Sie unter Konvertierungen unter Class Formatter in der API-Spezifikation für Java Platform, Standard Edition 7. Example Beispiele • <code>format(100+1: "d")</code> gibt eine Zeichenfolge zurück, <code>101</code>. • <code>format("The result is %d", 100+1)</code> gibt eine Zeichenfolge zurück, <code>The result is 101</code>.

Funktion	Description
f'expression'	Gibt eine verkettete Zeichenfolge zurück. Mit dieser formatierten Funktion können Sie einen einfachen Ausdruck verwenden, um Zeichenketten zu verketteten und zu formatieren. Diese Funktionen können verschachtelte Ausdrücke enthalten. Sie können {} (geschweifte Klammern) verwenden, um Ausdrücke zu interpolieren. Dies verhält sich ähnlich wie die formatierten Zeichenkettenliterals in Python. Example Beispiele • f'abc{1+2: "f"}d' gibt abc3.000000d zurück. Gehen Sie wie folgt vor, um diesen Beispielausdruck auszuwerten: 1. format(1+2: "f") gibt eine Fließkommazahl zurück,3.000000. 2. join(' ', "abc", 1+2, 'd')gibt eine Zeichenfolge zurück,abc3.000000d . Sie können den Ausdruck auch auf folgende Weise schreiben:join(' ', "abc", format(1+2: "f"), 'd') .

Verwenden Sie Aggregationsfunktionen in Formelausdrücken

Nur in [Metriken](#) können Sie die folgenden Funktionen verwenden, um Eingabewerte für jedes Zeitintervall zu aggregieren und einen einzelnen Ausgabewert zu berechnen. Einige Aggregationsfunktionen können keine Daten aus zugeordneten Komponenten aggregieren.

Argumente für Aggregationsfunktionen können [Variablen](#), [Zahlenliterals](#), [zeitliche Funktionen](#), verschachtelte Ausdrücke oder Aggregationsfunktionen sein. Die Formel `max(latest(x), latest(y), latest(z))` verwendet eine Aggregationsfunktion als Argument und gibt den größten aktuellen Wert der x Eigenschaften, und zurück. y z

Sie können verschachtelte Ausdrücke in Aggregationsfunktionen verwenden. Wenn Sie verschachtelte Ausdrücke verwenden, gelten die folgenden Regeln:

- Jedes Argument kann nur eine Variable haben.

Example

Zum Beispiel $\text{avg}(x \cdot (x-1))$ und $\text{sum}(x/2) / \text{avg}(y^2)$ werden unterstützt.

Wird beispielsweise $\min(x/y)$ nicht unterstützt.

- Jedes Argument kann verschachtelte Ausdrücke mit mehreren Ebenen haben.

Example

Wird beispielsweise $\text{sum}(\text{avg}(x^2)/2)$.

- Verschiedene Argumente können unterschiedliche Variablen haben.

Example

Zum Beispiel $\text{sum}(x/2, y \cdot 2)$ wird unterstützt.

Note

- Wenn Ihre Ausdrücke Messungen enthalten, AWS IoT SiteWise verwendet die letzten Werte im aktuellen Zeitintervall für die Messungen, um Aggregate zu berechnen.
- Wenn Ihre Ausdrücke Attribute enthalten, AWS IoT SiteWise verwendet die neuesten Werte für die Attribute, um Aggregate zu berechnen.

Funktion	Description
$\text{avg}(x_0, \dots, x_n)$	Gibt den Mittelwert der angegebenen Variablenwerte über das aktuelle Zeitintervall zurück. Diese Funktion gibt nur dann einen Datenpunkt aus, wenn die angegebenen Variablen im aktuellen Zeitintervall mindestens einen Datenpunkt haben.

Funktion	Description
$\text{sum}(x_0, \dots, x_n)$	Gibt die Summe der angegebenen Variablenwerte über das aktuelle Zeitintervall zurück. Diese Funktion gibt nur dann einen Datenpunkt t aus, wenn die angegebenen Variablen im aktuellen Zeitintervall mindestens einen Datenpunkt haben.
$\text{min}(x_0, \dots, x_n)$	Gibt den Mindestwert der angegebenen Variablenwerte über das aktuelle Zeitintervall zurück. Diese Funktion gibt nur dann einen Datenpunkt t aus, wenn die angegebenen Variablen im aktuellen Zeitintervall mindestens einen Datenpunkt haben.
$\text{max}(x_0, \dots, x_n)$	Gibt den Höchstwert der angegebenen Variablenwerte über das aktuelle Zeitintervall zurück. Diese Funktion gibt nur dann einen Datenpunkt t aus, wenn die angegebenen Variablen im aktuellen Zeitintervall mindestens einen Datenpunkt haben.
$\text{count}(x_0, \dots, x_n)$	Gibt die Gesamtanzahl der Datenpunkte für die angegebenen Variablen über das aktuelle Zeitintervall zurück. Weitere Informationen zum Zählen der Datenpunkte, die eine Bedingung erfüllen, finden Sie unter Zählen Sie Datenpunkte, die einer Bedingung entsprechen. Diese Funktion berechnet einen Datenpunkt für jedes Zeitintervall.

Funktion	Description
<code>stdev($x_0, \dots, x_n$)</code>	Gibt die Standardabweichung der Werte der angegebenen Variablen über das aktuelle Zeitintervall zurück. Diese Funktion gibt nur dann einen Datenpunkt <code>t</code> aus, wenn die angegebenen Variablen im aktuellen Zeitintervall mindestens einen Datenpunkt haben.

Verwenden Sie temporale Funktionen in Formelausdrücken

Verwenden Sie temporale Funktionen, um Werte zurückzugeben, die auf Zeitstempeln von Datenpunkten basieren.

Verwenden Sie zeitliche Funktionen in Metriken

Nur in [Metriken](#) können Sie die folgenden Funktionen verwenden, die Werte auf der Grundlage von Zeitstempeln von Datenpunkten zurückgeben.

Bei den Argumenten für temporale Funktionen muss es sich um Eigenschaften aus dem lokalen Objektmodell oder um verschachtelte Ausdrücke handeln. Das bedeutet, dass Sie keine Eigenschaften aus untergeordneten Vermögensmodellen in temporalen Funktionen verwenden können.

Sie können verschachtelte Ausdrücke in zeitlichen Funktionen verwenden. Wenn Sie verschachtelte Ausdrücke verwenden, gelten die folgenden Regeln:

- Jedes Argument kann nur eine Variable haben.

Zum Beispiel `latest( t*9/5 + 32 )` wird unterstützt.

- Argumente können keine Aggregationsfunktionen sein.

Wird beispielsweise `first( sum(x) )` nicht unterstützt.

Funktion	Description
<code>first(x)</code>	Gibt den Wert der angegebenen Variablen mit dem frühesten Zeitstempel über das aktuelle Zeitintervall hinweg zurück.
<code>last(x)</code>	Gibt den Wert der angegebenen Variablen mit dem spätesten Zeitstempel über das aktuelle Zeitintervall hinweg zurück.
<code>earliest(x)</code>	Gibt den letzten Wert der angegebenen Variablen vor dem Beginn des aktuellen Zeitintervalls zurück. Diese Funktion berechnet einen Datenpunkt für jedes Zeitintervall, wenn die Eingabeeigenschaft mindestens einen Datenpunkt im Verlauf hat. Details dazu finden Sie unter time-range-definition.
<code>latest(x)</code>	Gibt den letzten Wert der angegebenen Variablen mit dem letzten Zeitstempel vor dem Ende des aktuellen Zeitintervalls zurück. Diese Funktion berechnet einen Datenpunkt für jedes Zeitintervall, wenn die Eingabeeigenschaft mindestens einen Datenpunkt im Verlauf hat. Details dazu finden Sie unter time-range-definition.
<code>statetime(x)</code>	Gibt die Zeitspanne in Sekunden zurück, in der die angegebenen Variablen im aktuellen Zeitintervall positiv sind. Sie können die Vergleichsfunktionen verwenden, um eine Transformationseigenschaft zu erstellen, die von der <code>statetime</code> Funktion verwendet werden soll.

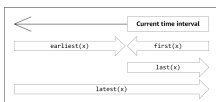
Funktion	Description
	Wenn Sie beispielsweise eine Idle-Eigenschaft haben, für die 0 oder 1 gilt, können Sie die Leerlaufzeit pro Zeitintervall mit diesem Ausdruck berechnen: <code>IdleTime = statetime(Idle)</code> . Weitere Informationen finden Sie im Beispielzustandszeitszenario. Diese Funktion unterstützt keine Metrikeigenschaften als Eingabevariablen. Diese Funktion berechnet einen Datenpunkt für jedes Zeitintervall, wenn die Eingabeeigenschaft mindestens einen Datenpunkt im Verlauf hat.

Funktion	Description
<code>TimeWeightedAvg(x, [interpolation])</code>	Gibt den Durchschnitt der Eingabedaten zurück, gewichtet mit Zeitintervallen zwischen Punkten. Einzelheiten zur Berechnung und zu den Intervallen finden Sie unter Parameter für zeitgewichtete Funktionen. Das optionale Argument <code>interpolation</code> muss eine Zeichenkettenkonstante sein: • <code>locf</code>— Dies ist die Standardeinstellung. Bei der Berechnung wird der Berechnungsalgorithmus Last Observed Carry Forward für Intervalle zwischen Datenpunkten verwendet. Bei diesem Ansatz wird der Datenpunkt als letzter beobachteter Wert bis zum Zeitstempel des nächsten Eingabedatenpunkts berechnet. Der Wert nach einem guten Datenpunkt wird als Wert bis zum nächsten Datenpunkt-Zeitstempel extrapoliert. • <code>linear</code>— Die Berechnung verwendet den Berechnungsalgorithmus der linearen Interpolation für Intervalle zwischen Datenpunkten. Der Wert zwischen zwei guten Datenpunkten wird als lineare Interpolation zwischen den Werten dieser Datenpunkte extrapoliert. Der Wert zwischen guten und schlechten Datenpunkten oder der Wert nach dem letzten guten Datenpunkt wird als guter Datenpunkt extrapoliert.

Funktion	Description
<code>TimeWeightedStDev(x, [algo])</code>	Gibt die Standardabweichung der Eingabedaten zurück, gewichtet mit Zeitintervallen zwischen Punkten. Einzelheiten zur Berechnung und zu den Intervallen finden Sie unter Parameter für zeitgewichtete Funktionen. Bei der Berechnung wird der Berechnungsalgorithmus Last Observed Carry Forward für Intervalle zwischen Datenpunkten verwendet. Bei diesem Ansatz wird der Datenpunkt als letzter beobachteter Wert bis zum Zeitstempel des nächsten Eingabedatenpunkts berechnet. Die Gewichtung wird als Zeitintervall in Sekunden zwischen Datenpunkten oder Fenstergrenzen berechnet. Das optionale Argument <code>algo</code> muss eine Zeichenkettenkonstante sein: • <code>f</code>— Dies ist die Standardeinstellung. Sie gibt eine unvoreingenommene gewichtete Stichprobenvarianz mit Frequenzgewichten zurück, die in Sekunden berechnet wird. Dieser Algorithmus wird in der Regel unter Berücksichtigung der Standardabweichung angenommen und wird bei gewichteten Stichproben als Bessel-Korrektur der Standardabweichung bezeichnet. • <code>p</code>— Gibt die verzerrte gewichtete Stichprobenvarianz zurück, die auch als Populationsvarianz bezeichnet wird.

Funktion	Description
	Die folgenden Formeln werden für Berechnungen verwendet, wobei: • S_p = Standardabweichung der Population • S_f = Frequenzstandardabweichung • X_i = eingehende Daten • ω_i = Gewicht, das dem Zeitintervall in Sekunden entspricht • μ^* = ein gewichteter Mittelwert der eingehenden Daten Gleichung für die Standardabweichung der Grundgesamtheit: $S_p^2 = \frac{\sum_{i=1}^N \omega_i (x_i - \mu^*)^2}{\sum_{i=1}^N \omega_i}$ Gleichung für die Frequenzstandardabweichung: $S_f^2 = \frac{\sum_{i=1}^N \omega_i (x_i - \mu^*)^2}{\sum_{i=1}^N \omega_i - 1}$

Das folgende Diagramm zeigt, wie die zeitlichen Funktionen `first`, `last`, `earliest` und `latest`, relativ zum aktuellen Zeitintervall AWS IoT SiteWise berechnet werden.



Note

- Der Zeitraum für `first(x)`, `last(x)` ist [aktueller Fensterstart, aktuelles Fensterende].
- Der Zeitraum für `latest(x)` ist [Beginn der Zeit, Ende des aktuellen Fensters].
- Der Zeitraum für `earliest(x)` ist [Beginn der Zeit, Ende des vorherigen Fensters].

Time-weighted Funktionen, Parameter.

Time-weighted Funktionen, die für das Aggregatfenster berechnet wurden, berücksichtigen Folgendes:

- Datenpunkte innerhalb des Fensters
- Zeitintervalle zwischen Datenpunkten
- Letzter Datenpunkt vor dem Fenster
- Erster Datenpunkt nach dem Fenster (für einige Algorithmen)

Bedingungen:

- Schlechter Datenpunkt — Jeder Datenpunkt mit nicht guter Qualität oder schlechtem Zahlenwert. Dies wird bei der Berechnung der Fensterergebnisse nicht berücksichtigt.
- Fehlerhaftes Intervall — Das Intervall nach einem fehlerhaften Datenpunkt. Das Intervall vor dem ersten bekannten Datenpunkt wird ebenfalls als fehlerhaftes Intervall angesehen.
- Guter Datenpunkt — Jeder Datenpunkt mit guter Qualität und numerischem Wert.

Note

- AWS IoT SiteWise verbraucht nur GOOD hochwertige Daten, wenn es Transformationen und Metriken berechnet. Es ignoriert alle Datenpunkte. UNCERTAIN BAD
- Das Intervall vor dem ersten bekannten Datenpunkt wird als schlechtes Intervall angesehen. Weitere Informationen finden Sie unter [the section called “Tutorials zu Formelausdrücken”](#).

Das Intervall nach dem letzten bekannten Datenpunkt dauert unbegrenzt und wirkt sich auf alle folgenden Fenster aus. Wenn ein neuer Datenpunkt eintrifft, berechnet die Funktion das Intervall neu.

Gemäß den obigen Regeln wird das aggregierte Fensterergebnis berechnet und auf Fenstergrenzen beschränkt. Standardmäßig sendet die Funktion das Fensterergebnis nur, wenn das gesamte Fenster ein gutes Intervall aufweist.

Wenn das Intervall für den Fensterwert kleiner als die Fensterlänge ist, sendet die Funktion das Fenster nicht.

Wenn sich die Datenpunkte, die das Fensterergebnis beeinflussen, ändern, berechnet die Funktion das Fenster neu, auch wenn sich die Datenpunkte außerhalb des Fensters befinden.

Wenn die Eingabeeigenschaft mindestens einen Datenpunkt in ihrer Historie hat und eine Berechnung eingeleitet wurde, berechnet die Funktion die zeitgewichteten Aggregatfunktionen für jedes Zeitintervall.

Example Beispielzustandszeitszenario

Betrachten Sie ein Beispiel mit einer Komponente mit den folgenden Eigenschaften:

- **Idle**— Eine Messung, die oder ist. 0 1 Wenn der Wert „1“ ist, befindet sich die Maschine im Leerlauf.
- **Idle Time**— Eine Metrik, die anhand der Formel `statetime(Idle)` die Zeit in Sekunden berechnet, in der sich die Maschine im Leerlauf befindet, pro 1-Minuten-Intervall.

Die **Idle**-Eigenschaft verfügt über die folgenden Datenpunkte.

Zeitstempel	14:00:00 Uhr	14:00:30 Uhr	14:01:15 Uhr	14:02:45 Uhr	14:04:00 Uhr
Idle	0	1	1	0	0

AWS IoT SiteWise berechnet die **Idle Time** Eigenschaft jede Minute aus den Werten von **Idle**. Nach Abschluss dieser Berechnung verfügt die **Idle Time**-Eigenschaft über die folgenden Datenpunkte.

Zeitstempel	14:00:00 Uhr	14:01:00 Uhr	14:02:00 Uhr	14:03:00 Uhr	14:04:00 Uhr
Idle Time	N/A	30	60	45	0

AWS IoT SiteWise führt am Ende jeder Minute die folgenden Berechnungen durch. **Idle Time**

- Um 14:00 Uhr (für 13:59 Uhr bis 14:00 Uhr)
 - Vor 14:00 Uhr liegen keine Daten für **Idle** vor, daher wird kein Datenpunkt berechnet.
- Um 14:01 Uhr (für 14:00 Uhr bis 14:01 Uhr)
 - Um 14:00:00 Uhr ist die Maschine aktiv (**Idle** ist 0).

- Um 14:00:30 Uhr befindet sich die Maschine im Leerlauf (Idle ist 1).
- Idle ändert sich vor dem Ende des Intervalls um 14:01:00 Uhr nicht mehr, Idle Time ist also 30 Sekunden.
- Um 14:02 Uhr (für 14:01 bis 14:02 Uhr)
 - Um 14:01:00 Uhr befindet sich die Maschine im Leerlauf (entsprechend des letzten Datenpunkts um 14:00:30 Uhr).
 - Um 14:01:15 Uhr befindet sich die Maschine noch im Leerlauf.
 - Idle ändert sich vor dem Ende des Intervalls um 14:02:00 Uhr nicht mehr, Idle Time ist also 60 Sekunden.
- Um 14:03 Uhr (für 14:02 Uhr bis 14:03 Uhr)
 - Um 14:02:00 Uhr befindet sich die Maschine im Leerlauf (entsprechend des letzten Datenpunkts um 14:01:15 Uhr).
 - Um 14:02:45 Uhr ist die Maschine aktiv.
 - Idle ändert sich vor dem Ende des Intervalls um 14:03:00 Uhr nicht mehr, Idle Time ist also 45 Sekunden.
- Um 14:04 Uhr (für 14:03 Uhr bis 14:04 Uhr)
 - Um 14:03:00 Uhr ist die Maschine aktiv (entsprechend des letzten Datenpunkts um 14:02:45 Uhr).
 - Idle ändert sich vor dem Ende des Intervalls um 14:04:00 Uhr nicht mehr, Idle Time ist also 0 Sekunden.

Example Beispiel TimeWeightedAvg und TimeWeightedStDev Szenario

Die folgenden Tabellen enthalten Beispielergebnisse und -ausgaben für diese einminütigen Fenstermetriken: `Avg(x)`, `TimeWeightedAvg(x)`, `TimeWeightedAvg(x, "linear")`, `stDev(x)`, `timeWeightedStDev(x)`, `timeWeightedStDev(x, 'p')`.

Beispielergebnisse für ein einminütiges Aggregatfenster:

Note

Diese Datenpunkte sind alle GOOD qualitativ hochwertig.

03:00:00	4,0
03:01:00	2.0
03:01:10	8.0
03:01:50	20.0
03:02:00	14,0
03:02:05	10.0
03:02:10	3.0
03:02:30	20.0
03:03:30	0.0

Ausgabe aggregierter Ergebnisse:

 Note

Keine — Für dieses Fenster wurde kein Ergebnis erzeugt.

Zeit	Avg(x)	TimeWeightedAvg(x)	TimeWeightedAvg(X, "linear")	stDev(X)	timeWeightedStDev(x)	timeWeightedStDev(x, 'p')
3:00:00	4	Keine	Keine	0	Keine	Keine
3:01:00	2	4	3	0	0	0
3:02:00	14	9	13	6	5,4306100 41581775	5,3851648 07134504

Zeit	Avg(x)	TimeWeightedAvg(x)	TimeWeightedAvg(X, "linear")	stDev(X)	timeWeightedStDev(x)	timeWeightedStDev(x, 'p')
3:03:00	11	13	12,875	8,5440037 4531753	7,7240544 37220943	7,6594168 62050705
3:04:00	0	10	2.5	0	10,084389 681792215	10
3:05:00	Keine	0	0	Keine	0	0

Verwenden Sie temporale Funktionen in Transformationen

Nur bei [Transformationen](#) können Sie die `pretrigger()` Funktion verwenden, um den GOOD Qualitätswert für eine Variable vor der Eigenschaftenaktualisierung abzurufen, die die aktuelle Transformationsberechnung ausgelöst hat.

Stellen Sie sich ein Beispiel vor AWS IoT SiteWise, bei dem ein Hersteller den Status einer Maschine überwacht. Der Hersteller verwendet die folgenden Messungen und Transformationen, um den Prozess darzustellen:

- Eine Messung `current_state`, die 0 oder 1 sein kann.
 - Wenn sich die Maschine im Reinigungszustand befindet, `current_state` entspricht dies 1.
 - Wenn sich die Maschine im Fertigungszustand befindet, `current_state` entspricht 0.
- Eine Transformation `cleaning_state_duration`, das entspricht `if(pretrigger(current_state) == 1, timestamp(current_state) - timestamp(pretrigger(current_state)), none)`. Diese Transformation gibt im Unix-Epochenformat in Sekunden zurück, wie lange sich die Maschine im Reinigungszustand befunden hat. Weitere Informationen finden Sie unter [Verwenden Sie bedingte Funktionen in Formelausdrücken](#) und zur Funktion [timestamp\(\)](#).

Bleibt das Gerät länger als erwartet im Reinigungszustand, überprüft der Hersteller das Gerät möglicherweise.

Sie können die `pretrigger()` Funktion auch in multivariaten Transformationen verwenden. Sie haben beispielsweise zwei Messungen mit dem Namen `x` und `y` eine Transformation, die entspricht $x + y + \text{pretrigger}(y)$. Die folgende Tabelle zeigt die Werte für `xy`, und `z` von 9:00 Uhr bis 9:15 Uhr.

 Note

- In diesem Beispiel wird davon ausgegangen, dass die Werte für die Messungen chronologisch eintreffen. Beispielsweise kommt der Wert `x` für 09:00 Uhr vor dem Wert `x` für 09:05 Uhr an.
- Wenn die Datenpunkte für 9:05 Uhr vor den Datenpunkten für 9:00 Uhr ankommen, wird um 9:05 Uhr `z` nicht berechnet.
- Wenn der Wert `x` für 9:05 Uhr vor dem Wert für 09:00 Uhr eintrifft und die Werte `x` für chronologisch `y` eintreffen, `z` entspricht $22 = 20 + 1 + 1$ dies um 9:05 Uhr.

	09:00 UHR	09:05 UHR	09:10 UHR	09:15 UHR
<code>x</code>	10	20		30
<code>y</code>	1	2	3	
$z = x + y + \text{pretrigger}(y)$	y empfängt vor 09:00 Uhr keinen Datenpunkt. Wird daher <code>z</code> nicht um 09:00 Uhr berechnet.	$23 = 20 + 2 + 1$ pretrigger(<code>y</code>) entspricht 1.	$25 = 20 + 3 + 2$ x empfängt keinen neuen Datenpunkt. pretrigger(<code>y</code>) entspricht 2.	$36 = 30 + 3 + 3$ y empfängt keinen neuen Datenpunkt. Entspricht pretrigger(<code>y</code>) also 3 um 09:15 Uhr.

Verwenden Sie Datums- und Uhrzeitfunktionen in Formelausdrücken

In [Transformationen](#) und [Metriken](#) können Sie die Datums- und Uhrzeitfunktionen auf folgende Weise verwenden:

- Ruft den aktuellen Zeitstempel eines Datenpunkts in UTC oder in der lokalen Zeitzone ab.
- Konstruieren Sie Zeitstempel mit Argumenten wie `yearmonth`, und. `day_of_month`
- Extrahieren Sie mit dem `unix_time` Argument einen Zeitraum, z. B. ein Jahr oder einen Monat.

Funktion	Description
<code>now()</code>	Gibt das aktuelle Datum und die aktuelle Uhrzeit in Sekunden im Unix-Epochenformat zurück.
<code>timestamp()</code>	• Bei Transformationen gibt die Funktion den Zeitstempel der Eingabenachricht in Sekunden im Unix-Epochenformat zurück. Nur bei Transformationen können Sie einen der folgenden Schritte ausführen: • Geben Sie eine Variable als Argument für die Funktion an. Die <code>timestamp (<i>variable-name</i>)</code> Funktion gibt den Zeitstempel des letzten GOOD Qualitäts werts für die angegebene Variable in Sekunden im Unix-Epochenformat zurück. Wenn Ihr Asset beispielsweise eine Transformationseigenschaft mit dem Namen hat, die die <code>9/5 * Temperature_C</code> Formel verwendet <code>Temperature_F</code>, um jeden Temperaturdatenpunkt von Celsius in Fahrenheit umzurechnen, können Sie die <code>timestamp(Temperature_F)</code> Funktion verwenden, um den Zeitstempel des neuesten GOOD Qualitäts werts für die Eigenschaft abzurufen. <code>Temperature_F</code> • Verwenden Sie die <code>pretrigger()</code> Funktion als Argument für die

Funktion	Description
	Funktion. Die <code>timestamp(pretrigger(<i>variable-name</i>))</code> Funktion gibt den Zeitstempel des GOOD Qualitäts werts für die angegebene Variable vor der Eigenschaftenaktualisierung, die die aktuelle Transformationsberechnung ausgelöst hat, in Sekunden im Unix-Epochenformat zurück. Weitere Informationen finden Sie unter Verwenden Sie temporale Funktionen in Transformationen. • Bei Metriken gibt die Funktion den am Ende des aktuellen Fensters abgerufenen Zeitstempel in Sekunden im Unix-Epochenformat zurück.

Funktion	Description
<code>mktime(time_zone, year, month, day_of_month, hour, minute, second)</code>	Gibt die Eingabezeit in Sekunden im Unix-Epochenformat zurück. Für die Verwendung dieser Funktion gelten die folgenden Anforderungen: • Das Zeitzoneargument muss eine Zeichenfolge (' UTC ') in Anführungszeichen sein. Wenn nicht angegeben, ist die Standardzeitzone UTC. Das Zeitzoneargument kann das erste oder letzte Argument sein. • Die Argumente Jahr, Monat, Tag des Monats, Stunde, Minute und Sekunde müssen in der richtigen Reihenfolge angegeben werden. • Die Argumente Jahr, Monat und Datum sind erforderlich. Für die Verwendung dieser Funktion gelten die folgenden Einschränkungen: • <code>year</code>- Gültige Werte liegen zwischen 1970 und 2250. • <code>month</code>- Gültige Werte liegen zwischen 1 und 12. • <code>day-of-month</code> - Gültige Werte liegen zwischen 1 und 31. • <code>hour</code>- Gültige Werte liegen zwischen 0 und 23. • <code>minute</code>- Gültige Werte liegen zwischen 0 und 59.

Funktion	Description
	• second- Gültige Werte liegen zwischen 0 und 60. Es kann sich um eine Fließkommazahl handeln. Beispiele: • <code>mtime(2020, 2, 29)</code> • <code>mtime('UTC+3', 2021, 12, 31, 22)</code> • <code>mtime(2022, 10, 13, 2, 55, 13.68, 'PST')</code>

Funktion	Description
<code>localtime(unix_time, time_zone)</code>	Gibt das Jahr, den Tag des Monats, den Wochentag, den Tag des Jahres, die Stunde, die Minute oder die Sekunde in der angegebenen Zeitzone aus der Unix-Zeit zurück. Für die Verwendung dieser Funktion gelten die folgenden Anforderungen: • Das Zeitzenenargument muss eine Zeichenfolge (' UTC ') in Anführungszeichen sein. Wenn nicht angegeben, ist die Standardzeitzone UTC. • Das Unix-Zeitargument ist die Zeit in Sekunden im Unix-Epochenformat. Der gültige Bereich liegt zwischen 1-31556889864403199. Es kann eine Fließkommazahl sein. Beispiel für eine Antwort: 2007-12-03T10:15:30+01:00[Europe/Paris] <code>localtime(unix_time, time_zone)</code> ist keine eigenständige Funktion. Die <code>sec()</code> Funktionen <code>year()</code>, <code>mon()</code>, <code>mday()</code>, <code>wday()</code>, <code>yday()</code>, <code>hour()</code>, <code>minute()</code>, und nehmen <code>localtime(unix_time, time_zone)</code> als Argument an. Beispiele: • <code>year(localtime('GMT', 1605898608.8113723))</code> • <code>now().localtime().year()</code> • <code>timestamp().localtime('PST').year()</code>

Funktion	Description
	<code>localtime(1605289736, 'Europe/London').year()</code>
<code>year(localtime(unix_time, time_zone))</code>	Gibt das Jahr von zurücklocaltime(unix_time, time_zone) .
<code>mon(localtime(unix_time, time_zone))</code>	Gibt den Monat von zurücklocaltime(unix_time, time_zone) .
<code>mday(localtime(unix_time, time_zone))</code>	Gibt den Tag des Monats von zurücklocaltime(unix_time, time_zone) .
<code>wday(localtime(unix_time, time_zone))</code>	Gibt den Wochentag von zurücklocaltime(unix_time, time_zone) .
<code>yday(localtime(unix_time, time_zone))</code>	Gibt den Tag des Jahres von zurücklocaltime(unix_time, time_zone) .
<code>hour(localtime(unix_time, time_zone))</code>	Gibt die Stunde von zurücklocaltime(unix_time, time_zone) .
<code>minute(localtime(unix_time, time_zone))</code>	Gibt die Minute von zurücklocaltime(unix_time, time_zone) .
<code>sec(localtime(unix_time, time_zone))</code>	Gibt die Sekunde von zurücklocaltime(unix_time, time_zone) .

Unterstützte Zeitonenformate

Sie können das Zeitonenargument auf folgende Weise angeben:

- Zeitonen-Offset — Geben Sie 'Z' UTC oder einen Offset ('+2' oder '-5') an.
- Offset-IDs — Kombinieren Sie eine Abkürzung für eine Zeitzone und einen Offset. Beispiel: 'GMT+2' und 'UTC-01:00'. Die Abkürzung für die Zeitzone darf nur drei Buchstaben enthalten.
- Regionsbasierte IDs — Zum Beispiel 'Etc/GMT+12' und 'Pacific/Pago_Pago'.

Unterstützte Abkürzungen für Zeitzonen

Die Datums- und Uhrzeitfunktionen unterstützen die folgenden aus drei Buchstaben bestehenden Abkürzungen für Zeitzonen:

- BESTE ZEIT - - 05:00
- DONNERSTAG - - 10:00
- MST - - 07:00
- AKT - Australia/Darwin
- ESSEN - Australia/Sydney
- AGT - America/Argentina/Buenos _Aires
- KUNST - Africa/Cairo
- SCHNELL - America/Anchorage
- WETTE - America/Sao _Paulo
- BST - Asia/Dhaka
- KATZE - Africa/Harare
- MEZ - Europe/Paris
- CNT - _Johns America/St
- CST - America/Chicago
- CTT - Asia/Shanghai
- ESSEN - _Abeba Africa/Addis
- DIÄT - America/Indiana/Indianapolis
- IST - Asia/Kolkata
- NUR - Asia/Tokyo
- MIT - Pacific/Apia
- NETZ - Asia/Yerevan
- NET - Pacific/Auckland
- PLT - Asia/Karachi
- PRT - _Rico America/Puerto
- PST - _Angeles America/Los
- SST - Pacific/Guadalcanal

- VST - _ Asia/Ho Chi_Minh

Unterstützte IDs Region-based

Die Datums- und Uhrzeitfunktionen unterstützen die folgenden Region-based IDs, geordnet nach ihrer Beziehung zu UTC+ 00:00:

- Etc/GMT+12 () UTC-12:00
- Pacific/Pago_Zahlung () UTC-11:00
- Pacific/Samoa (UTC-11:00)
- Pacific/Niue (UTC-11:00)
- US/Samoa (UTC-11:00)
- Etc/GMT+11 () UTC-11:00
- Pacific/Midway (UTC-11:00)
- Pacific/Honolulu (UTC-10:00)
- Pacific/Rarotonga (UTC-10:00)
- Pacific/Tahiti (UTC-10:00)
- Pacific/Johnston (UTC-10:00)
- US/Hawaii (UTC-10:00)
- SystemV/HST10 (UTC-10:00)
- Etc/GMT+10 () UTC-10:00
- Pacific/Marquesas (UTC-09:30)
- Etc/GMT+9 () UTC-09:00
- Pacific/Gambier (UTC-09:00)
- America/Atka (UTC-09:00)
- SystemV/YST9 (UTC-09:00)
- America/Adak (UTC-09:00)
- US/Aleutian (UTC-09:00)
- Etc/GMT+8 () UTC-08:00
- US/Alaska (UTC-08:00)
- America/Juneau (UTC-08:00)

- America/Metlakatla (UTC-08:00)
- America/Yakutat (UTC-08:00)
- Pacific/Pitcairn (UTC-08:00)
- America/Sitka (UTC-08:00)
- America/Anchorage (UTC-08:00)
- SystemV/PST8 (UTC-08:00)
- America/Nome (UTC-08:00)
- SystemV/YST9YDT (UTC-08:00)
- Canada/Yukon (UTC-07:00)
- US/Pacific-New (UTC-07:00)
- Etc/GMT+7 () UTC-07:00
- US/Arizona (UTC-07:00)
- America/Dawson_Bach () UTC-07:00
- Canada/Pacific (UTC-07:00)
- PST8PDT () UTC-07:00
- SystemV/MST7 (UTC-07:00)
- America/Dawson (UTC-07:00)
- Mexico/BajaNorte (UTC-07:00)
- America/Tijuana (UTC-07:00)
- America/Creston (UTC-07:00)
- America/Hermosillo (UTC-07:00)
- America/Santa_Isabel (UTC-07:00)
- America/Vancouver (UTC-07:00)
- America/Ensenada (UTC-07:00)
- America/Phoenix (UTC-07:00)
- America/Whitehorse (UTC-07:00)
- America/Fort_Nelson () UTC-07:00
- SystemV/PST8PDT (UTC-07:00)
- America/Los_Angeles () UTC-07:00

- US/Pacific (UTC-07:00)
- America/El_Salvador () UTC-06:00
- America/Guatemala (UTC-06:00)
- America/Belize (UTC-06:00)
- America/Managua (UTC-06:00)
- America/Tegucigalpa (UTC-06:00)
- Etc/GMT+6 () UTC-06:00
- Pacific/Easter (UTC-06:00)
- Mexico/BajaSur (UTC-06:00)
- America/Regina (UTC-06:00)
- America/Denver (UTC-06:00)
- Pacific/Galapagos (UTC-06:00)
- America/Yellowknife (UTC-06:00)
- America/Swift_Aktuell () UTC-06:00
- America/Inuvik (UTC-06:00)
- America/Mazatlan (UTC-06:00)
- America/Boise (UTC-06:00)
- America/Costa_Rica () UTC-06:00
- MST7 MDT () UTC-06:00
- SystemV/CST6 (UTC-06:00)
- America/Chihuahua (UTC-06:00)
- America/Ojinaga (UTC-06:00)
- Chile/EasterIsland (UTC-06:00)
- US/Mountain (UTC-06:00)
- America/Edmonton (UTC-06:00)
- Canada/Mountain (UTC-06:00)
- America/Cambridge_Bucht (UTC-06:00)
- Navajo () UTC-06:00
- SystemV/MST7MDT (UTC-06:00)

- Canada/Saskatchewan (UTC-06:00)
- America/Shiprock (UTC-06:00)
- America/Panama (UTC-05:00)
- America/Chicago (UTC-05:00)
- America/Eirunepe (UTC-05:00)
- Etc/GMT+5 () UTC-05:00
- Mexico/General (UTC-05:00)
- America/Porto_Hektar () UTC-05:00
- America/Guayaquil (UTC-05:00)
- America/Rankin_Einlass () UTC-05:00
- US/Central (UTC-05:00)
- America/Rainy_Fluss () UTC-05:00
- America/Indiana/Knox (UTC-05:00)
- America/North_Dakota/Beulah (UTC-05:00)
- America/Monterrey (UTC-05:00)
- America/Jamaica (UTC-05:00)
- America/Atikokan (UTC-05:00)
- America/Coral_Hafen () UTC-05:00
- America/North_Dakota/Center (UTC-05:00)
- America/Cayman (UTC-05:00)
- America/Indiana/Tell_Stadt () UTC-05:00
- America/Mexico_Stadt () UTC-05:00
- America/Matamoros (UTC-05:00)
- CST6CDT () UTC-05:00
- America/Knox_REIN (UTC-05:00)
- America/Bogota (UTC-05:00)
- America/Menominee (UTC-05:00)
- America/Resolute (UTC-05:00)
- SystemV/EST5 (UTC-05:00)

- Canada/Central (UTC-05:00)
- Brazil/Acre (UTC-05:00)
- America/Cancun (UTC-05:00)
- America/Lima (UTC-05:00)
- America/Bahia_Flaggen () UTC-05:00
- US/Indiana-Starke (UTC-05:00)
- America/Rio_Weiß () UTC-05:00
- SystemV/CST6CDT (UTC-05:00)
- Jamaika (UTC-05:00)
- America/Merida (UTC-05:00)
- America/North_Dakota/New_Salem (UTC-05:00)
- America/Winnipeg (UTC-05:00)
- America/Cuiaba (UTC-04:00)
- America/Marigot (UTC-04:00)
- America/Indiana/Petersburg (UTC-04:00)
- Chile/Continental (UTC-04:00)
- America/Grand_Türke () UTC-04:00
- Kuba () UTC-04:00
- Etc/GMT+4 () UTC-04:00
- America/Manaus (UTC-04:00)
- America/Fort_Wayne () UTC-04:00
- America/St_Thomas () UTC-04:00
- America/Anguilla (UTC-04:00)
- America/Havana (UTC-04:00)
- US/Michigan (UTC-04:00)
- America/Barbados (UTC-04:00)
- America/Louisville (UTC-04:00)
- America/Curacao (UTC-04:00)
- America/Guyana (UTC-04:00)

- America/Martinique (UTC-04:00)
- America/Puerto_Rico () UTC-04:00
- America/Port_von_Spanien () UTC-04:00
- SystemV/AST4 (UTC-04:00)
- America/Indiana/Vevay (UTC-04:00)
- America/Indiana/Vincennes (UTC-04:00)
- America/Kralendijk (UTC-04:00)
- America/Antigua (UTC-04:00)
- America/Indianapolis (UTC-04:00)
- America/Iqaluit (UTC-04:00)
- America/St_Vincent () UTC-04:00
- America/Kentucky/Louisville (UTC-04:00)
- America/Dominica (UTC-04:00)
- America/Asuncion (UTC-04:00)
- EST 5 EDT () UTC-04:00
- America/Nassau (UTC-04:00)
- America/Kentucky/Monticello (UTC-04:00)
- Brazil/West (UTC-04:00)
- America/Aruba (UTC-04:00)
- America/Indiana/Indianapolis (UTC-04:00)
- America/Santiago (UTC-04:00)
- America/La_Paz () UTC-04:00
- America/Thunder_Bucht () UTC-04:00
- America/Indiana/Marengo (UTC-04:00)
- America/Blanc-Sablon (UTC-04:00)
- America/Santo_Sonntag () UTC-04:00
- US/Eastern (UTC-04:00)
- Canada/Eastern (UTC-04:00)
- America/Port-au-Prince (UTC-04:00)

- America/St_Barthélemy () UTC-04:00
- America/Nipigon (UTC-04:00)
- US/East-Indiana (UTC-04:00)
- America/St_Lucia (UTC-04:00)
- America/Montserrat (UTC-04:00)
- America/Lower_Prinzen () UTC-04:00
- America/Detroit (UTC-04:00)
- America/Tortola (UTC-04:00)
- America/Porto_Groß () UTC-04:00
- America/Campo_Groß () UTC-04:00
- America/Virgin (UTC-04:00)
- America/Pangnirtung (UTC-04:00)
- America/Montreal (UTC-04:00)
- America/Indiana/Winamac (UTC-04:00)
- America/Boa_Vista () UTC-04:00
- America/Grenada (UTC-04:00)
- America/New_York () UTC-04:00
- America/St_Kitts () UTC-04:00
- America/Caracas (UTC-04:00)
- America/Guadeloupe (UTC-04:00)
- America/Toronto (UTC-04:00)
- SystemV/EST5EDT (UTC-04:00)
- America/Argentina/Catamarca (UTC-03:00)
- Canada/Atlantic (UTC-03:00)
- America/Argentina/Cordoba (UTC-03:00)
- America/Araguaina (UTC-03:00)
- America/Argentina/Salta (UTC-03:00)
- Etc/GMT+3 () UTC-03:00
- America/Montevideo (UTC-03:00)

- Brazil/East (UTC-03:00)
- America/Argentina/Mendoza (UTC-03:00)
- America/Argentina/Rio_Gallegos () UTC-03:00
- America/Catamarca (UTC-03:00)
- America/Cordoba (UTC-03:00)
- America/Sao_Paulo () UTC-03:00
- America/Argentina/Jujuy (UTC-03:00)
- America/Cayenne (UTC-03:00)
- America/Recife (UTC-03:00)
- America/Buenos_Aires () UTC-03:00
- America/Paramaribo (UTC-03:00)
- America/Moncton (UTC-03:00)
- America/Mendoza (UTC-03:00)
- America/Santarem (UTC-03:00)
- Atlantic/Bermuda (UTC-03:00)
- America/Maceio (UTC-03:00)
- Atlantic/Stanley (UTC-03:00)
- America/Halifax (UTC-03:00)
- Antarctica/Rothera (UTC-03:00)
- America/Argentina/San_Luis () UTC-03:00
- America/Argentina/Ushuaia (UTC-03:00)
- Antarctica/Palmer (UTC-03:00)
- America/Punta_Arenen () UTC-03:00
- America/Glace_Bucht () UTC-03:00
- America/Fortaleza (UTC-03:00)
- America/Thule (UTC-03:00)
- America/Argentina/La_Rioja () UTC-03:00
- America/Belem (UTC-03:00)
- America/Jujuy (UTC-03:00)

- America/Bahia (UTC-03:00)
- America/Goose_Bucht () UTC-03:00
- America/Argentina/San_Juan () UTC-03:00
- America/Argentina/ComodRivadavia (UTC-03:00)
- America/Argentina/Tucuman (UTC-03:00)
- America/Rosario (UTC-03:00)
- SystemV/AST4ADT (UTC-03:00)
- America/Argentina/Buenos_Aires () UTC-03:00
- America/St_Johns () UTC-02:30
- Canada/Newfoundland (UTC-02:30)
- America/Miquelon (UTC-02:00)
- Etc/GMT+2 () UTC-02:00
- America/Godthab (UTC-02:00)
- America/Noronha (UTC-02:00)
- Brazil/DeNoronha (UTC-02:00)
- Atlantic/South_Georgien () UTC-02:00
- Etc/GMT+1 () UTC-01:00
- Atlantic/Cape_Grün () UTC-01:00
- Pacific/Kiritimati (UTC+ 14:00)
- Etc/GMT-14 (UTC+ 14:00)
- Pacific/Fakaofu (UTC+ 13:00)
- Pacific/Enderbury (UTC+ 13:00)
- Pacific/Apia (UTC+ 13:00)
- Pacific/Tongatapu (UTC+ 13:00)
- Etc/GMT-13 (UTC+ 13:00)
- NZ-CHAT (UTC+ 12:45)
- Pacific/Chatham (UTC+ 12:45)
- Pacific/Kwajalein (UTC+ 12:00)
- Antarctica/McMurdo (UTC+ 12:00)

- Pacific/Wallis (UTC+ 12:00)
- Pacific/Fiji (UTC+ 12:00)
- Pacific/Funafuti (UTC+ 12:00)
- Pacific/Nauru (UTC+ 12:00)
- Kwajalein (UTC+ 12:00)
- NEUSEELAND (UTC+ 12:00)
- Pacific/Wake (UTC+ 12:00)
- Antarctica/South_Pol (UTC+ 12:00)
- Pacific/Tarawa (UTC+ 12:00)
- Pacific/Auckland (UTC+ 12:00)
- Asia/Kamchatka (UTC+ 12:00)
- Etc/GMT-12 (UTC+ 12:00)
- Asia/Anadyr (UTC+ 12:00)
- Pacific/Majuro (UTC+ 12:00)
- Pacific/Ponape (UTC+ 11:00)
- Pacific/Bougainville (UTC+ 11:00)
- Antarctica/Macquarie (UTC+ 11:00)
- Pacific/Pohnpei (UTC+ 11:00)
- Pacific/Efate (UTC+ 11:00)
- Pacific/Norfolk (UTC+ 11:00)
- Asia/Magadan (UTC+ 11:00)
- Pacific/Kosrae (UTC+ 11:00)
- Asia/Sakhalin (UTC+ 11:00)
- Pacific/Noumea (UTC+ 11:00)
- Etc/GMT-11 (UTC+ 11:00)
- Asia/Srednekolymsk (UTC+ 11:00)
- Pacific/Guadalcanal (UTC+ 11:00)
- Australia/Lord_Howe (UTC+ 10:30)
- Australia/LHI (UTC+ 10:30)

- Australia/Hobart (UTC+ 10:00)
- Pacific/Yap (UTC+ 10:00)
- Australia/Tasmania (UTC+ 10:00)
- Pacific/Port_Moresby (UTC+ 10:00)
- Australia/ACT (UTC+ 10:00)
- Australia/Victoria (UTC+ 10:00)
- Pacific/Chuuk (UTC+ 10:00)
- Australia/Queensland (UTC+ 10:00)
- Australia/Canberra (UTC+ 10:00)
- Australia/Currie (UTC+ 10:00)
- Pacific/Guam (UTC+ 10:00)
- Pacific/Truk (UTC+ 10:00)
- Australia/NSW (UTC+ 10:00)
- Asia/Vladivostok (UTC+ 10:00)
- Pacific/Saipan (UTC+ 10:00)
- Antarctica/DumontDUrville (UTC+ 10:00)
- Australia/Sydney (UTC+ 10:00)
- Australia/Brisbane (UTC+ 10:00)
- Etc/GMT-10 (UTC+ 10:00)
- Asia/Ust-Nera (UTC+ 10:00)
- Australia/Melbourne (UTC+ 10:00)
- Australia/Lindeman (UTC+ 10:00)
- Australia/North (UTC+ 09:30)
- Australia/Yancowinna (UTC+ 09:30)
- Australia/Adelaide (UTC+ 09:30)
- Australia/Broken_Hill (UTC+ 09:30)
- Australia/South (UTC+ 09:30)
- Australia/Darwin (UTC+ 09:30)
- Etc/GMT-9 (UTC+ 09:00)

- Pacific/Palau (UTC+ 09:00)
- Asia/Chita (UTC+ 09:00)
- Asia/Dili (UTC+ 09:00)
- Asia/Jayapura (UTC+ 09:00)
- Asia/Yakutsk (UTC+ 09:00)
- Asia/Pyongyang (UTC+ 09:00)
- ROCK (UTC+ 09:00)
- Asia/Seoul (UTC+ 09:00)
- Asia/Khandyga (UTC+ 09:00)
- Japan (UTC+ 09:00)
- Asia/Tokyo (UTC+ 09:00)
- Australia/Eucla (UTC+ 08:45)
- Asia/Kuching (UTC+ 08:00)
- Asia/Chungking (UTC+ 08:00)
- Etc/GMT-8 (UTC+ 08:00)
- Australia/Perth (UTC+ 08:00)
- Asia/Macao (UTC+ 08:00)
- Asia/Macau (UTC+ 08:00)
- Asia/Choibalsan (UTC+ 08:00)
- Asia/Shanghai (UTC+ 08:00)
- Antarctica/Casey (UTC+ 08:00)
- Asia/Ulan_Bator (UTC+ 08:00)
- Asia/Chongqing (UTC+ 08:00)
- Asia/Ulaanbaatar (UTC+ 08:00)
- Asia/Taipei (UTC+ 08:00)
- Asia/Manila (UTC+ 08:00)
- PRC (UTC+ 08:00)
- Asia/Ujung_Pandang (UTC+ 08:00)
- Asia/Harbin (UTC+ 08:00)

- Singapur (UTC+ 08:00)
- Asia/Brunei (UTC+ 08:00)
- Australia/West (UTC+ 08:00)
- Asia/Hong_Kong (UTC+ 08:00)
- Asia/Makassar (UTC+ 08:00)
- Hongkong (UTC+ 08:00)
- Asia/Kuala_Lumpur (UTC+ 08:00)
- Asia/Irkutsk (UTC+ 08:00)
- Asia/Singapore (UTC+ 08:00)
- Asia/Pontianak (UTC+ 07:00)
- Etc/GMT-7 (UTC+ 07:00)
- Asia/Phnom_Penh (UTC+ 07:00)
- Asia/Novosibirsk (UTC+ 07:00)
- Antarctica/Davis (UTC+ 07:00)
- Asia/Tomsk (UTC+ 07:00)
- Asia/Jakarta (UTC+ 07:00)
- Asia/Barnaul (UTC+ 07:00)
- Indian/Christmas (UTC+ 07:00)
- Asia/Ho_Chi_Minh (UTC+ 07:00)
- Asia/Hovd (UTC+ 07:00)
- Asia/Bangkok (UTC+ 07:00)
- Asia/Vientiane (UTC+ 07:00)
- Asia/Novokuznetsk (UTC+ 07:00)
- Asia/Krasnoyarsk (UTC+ 07:00)
- Asia/Saigon (UTC+ 07:00)
- Asia/Yangon (UTC+ 06:30)
- Asia/Rangoon (UTC+ 06:30)
- Indian/Cocos (UTC+ 06:30)
- Asia/Kashgar (UTC+ 06:00)

- Etc/GMT-6 (UTC+ 06:00)
- Asia/Almaty (UTC+ 06:00)
- Asia/Dacca (UTC+ 06:00)
- Asia/Omsk (UTC+ 06:00)
- Asia/Dhaka (UTC+ 06:00)
- Indian/Chagos (UTC+ 06:00)
- Asia/Qyzylorda (UTC+ 06:00)
- Asia/Bishkek (UTC+ 06:00)
- Antarctica/Vostok (UTC+ 06:00)
- Asia/Urumqi (UTC+ 06:00)
- Asia/Thimbu (UTC+ 06:00)
- Asia/Thimphu (UTC+ 06:00)
- Asia/Kathmandu (UTC+ 05:45)
- Asia/Katmandu (UTC+ 05:45)
- Asia/Kolkata (UTC+ 05:30)
- Asia/Colombo (UTC+ 05:30)
- Asia/Calcutta (UTC+ 05:30)
- Asia/Aqtau (UTC+ 05:00)
- Etc/GMT-5 (UTC+ 05:00)
- Asia/Samarkand (UTC+ 05:00)
- Asia/Karachi (UTC+ 05:00)
- Asia/Yekaterinburg (UTC+ 05:00)
- Asia/Dushanbe (UTC+ 05:00)
- Indian/Maldives (UTC+ 05:00)
- Asia/Oral (UTC+ 05:00)
- Asia/Tashkent (UTC+ 05:00)
- Antarctica/Mawson (UTC+ 05:00)
- Asia/Aqtobe (UTC+ 05:00)
- Asia/Ashkhabad (UTC+ 05:00)

- Asia/Ashgabat (UTC+ 05:00)
- Asia/Atyrau (UTC+ 05:00)
- Indian/Kerguelen (UTC+ 05:00)
- Iran (UTC+ 04:30)
- Asia/Tehran (UTC+ 04:30)
- Asia/Kabul (UTC+ 04:30)
- Asia/Yerevan (UTC+ 04:00)
- Etc/GMT-4 (UTC+ 04:00)
- Etc/GMT-4 (UTC+ 04:00)
- Asia/Dubai (UTC+ 04:00)
- Indian/Reunion (UTC+ 04:00)
- Europe/Saratov (UTC+ 04:00)
- Europe/Samara (UTC+ 04:00)
- Indian/Mahe (UTC+ 04:00)
- Asia/Baku (UTC+ 04:00)
- Asia/Muscat (UTC+ 04:00)
- Europe/Volgograd (UTC+ 04:00)
- Europe/Astrakhan (UTC+ 04:00)
- Asia/Tbilisi (UTC+ 04:00)
- Europe/Ulyanovsk (UTC+ 04:00)
- Asia/Aden (UTC+ 03:00)
- Africa/Nairobi (UTC+ 03:00)
- Europe/Istanbul (UTC+ 03:00)
- Etc/GMT-3 (UTC+ 03:00)
- Europe/Zaporozhye (UTC+ 03:00)
- Israel (UTC+ 03:00)
- Indian/Comoro (UTC+ 03:00)
- Antarctica/Syowa (UTC+ 03:00)
- Africa/Mogadishu (UTC+ 03:00)

- Europe/Bucharest (UTC+ 03:00)
- Africa/Asmera (UTC+ 03:00)
- Europe/Mariehamn (UTC+ 03:00)
- Asia/Istanbul (UTC+ 03:00)
- Europe/Tiraspol (UTC+ 03:00)
- Europe/Moscow (UTC+ 03:00)
- Europe/Chisinau (UTC+ 03:00)
- Europe/Helsinki (UTC+ 03:00)
- Asia/Beirut (UTC+ 03:00)
- Asia/Tel_Aviv (UTC+ 03:00)
- Africa/Djibouti (UTC+ 03:00)
- Europe/Simferopol (UTC+ 03:00)
- Europe/Sofia (UTC+ 03:00)
- Asia/Gaza (UTC+ 03:00)
- Africa/Asmara (UTC+ 03:00)
- Europe/Riga (UTC+ 03:00)
- Asia/Baghdad (UTC+ 03:00)
- Asia/Damascus (UTC+ 03:00)
- Africa/Dar_ES_Salaam (UTC+ 03:00)
- Africa/Addis_Abeba (UTC+ 03:00)
- Europe/Uzhgorod (UTC+ 03:00)
- Asia/Jerusalem (UTC+ 03:00)
- Asia/Riyadh (UTC+ 03:00)
- Asia/Kuwait (UTC+ 03:00)
- Europe/Kirov (UTC+ 03:00)
- Africa/Kampala (UTC+ 03:00)
- Europe/Minsk (UTC+ 03:00)
- Asia/Qatar (UTC+ 03:00)
- Europe/Kiev (UTC+ 03:00)

- Asia/Bahrain (UTC+ 03:00)
- Europe/Vilnius (UTC+ 03:00)
- Indian/Antananarivo (UTC+ 03:00)
- Indian/Mayotte (UTC+ 03:00)
- Europe/Tallinn (UTC+ 03:00)
- Türkei (UTC+ 03:00)
- Africa/Juba (UTC+ 03:00)
- Asia/Nicosia (UTC+ 03:00)
- Asia/Famagusta (UTC+ 03:00)
- W-SU (UTC+ 03:00)
- FÜSSE (UTC+ 03:00)
- Asia/Hebron (UTC+ 03:00)
- Asia/Amman (UTC+ 03:00)
- Europe/Nicosia (UTC+ 03:00)
- Europe/Athens (UTC+ 03:00)
- Africa/Cairo (UTC+ 02:00)
- Africa/Mbabane (UTC+ 02:00)
- Europe/Brussels (UTC+ 02:00)
- Europe/Warsaw (UTC+ 02:00)
- MEZ (UTC+ 02:00)
- Europe/Luxembourg (UTC+ 02:00)
- Etc/GMT-2 (UTC+ 02:00)
- Libyen (UTC+ 02:00)
- Africa/Kigali (UTC+ 02:00)
- Africa/Tripoli (UTC+ 02:00)
- Europe/Kaliningrad (UTC+ 02:00)
- Africa/Windhoek (UTC+ 02:00)
- Europe/Malta (UTC+ 02:00)
- Europe/Busingen (UTC+ 02:00)

-
- Europe/Skopje (UTC+ 02:00)
- Europe/Sarajevo (UTC+ 02:00)
- Europe/Rome (UTC+ 02:00)
- Europe/Zurich (UTC+ 02:00)
- Europe/Gibraltar (UTC+ 02:00)
- Africa/Lubumbashi (UTC+ 02:00)
- Europe/Vaduz (UTC+ 02:00)
- Europe/Ljubljana (UTC+ 02:00)
- Europe/Berlin (UTC+ 02:00)
- Europe/Stockholm (UTC+ 02:00)
- Europe/Budapest (UTC+ 02:00)
- Europe/Zagreb (UTC+ 02:00)
- Europe/Paris (UTC+ 02:00)
- Africa/Ceuta (UTC+ 02:00)
- Europe/Prague (UTC+ 02:00)
- Antarctica/Troll (UTC+ 02:00)
- Africa/Gaborone (UTC+ 02:00)
- Europe/Copenhagen (UTC+ 02:00)
- Europe/Vienna (UTC+ 02:00)
- Europe/Tirane (UTC+ 02:00)
- GETROFFEN (UTC+ 02:00)
- Europe/Amsterdam (UTC+ 02:00)
- Africa/Maputo (UTC+ 02:00)
- Europe/San_Marino (UTC+ 02:00)
- Polen (UTC+ 02:00)
- Europe/Andorra (UTC+ 02:00)
- Europe/Oslo (UTC+ 02:00)
- Europe/Podgorica (UTC+ 02:00)
- Africa/Bujumbura (UTC+ 02:00)

- Atlantic/Jan_Mayen (UTC+ 02:00)
- Africa/Maseru (UTC+ 02:00)
- Europe/Madrid (UTC+ 02:00)
- Africa/Blantyre (UTC+ 02:00)
- Africa/Lusaka (UTC+ 02:00)
- Africa/Harare (UTC+ 02:00)
- Africa/Khartoum (UTC+ 02:00)
- Africa/Johannesburg (UTC+ 02:00)
- Europe/Belgrade (UTC+ 02:00)
- Europe/Bratislava (UTC+ 02:00)
- Arctic/Longyearbyen (UTC+ 02:00)
- Ägypten (UTC+ 02:00)
- Europe/Vatican (UTC+ 02:00)
- Europe/Monaco (UTC+ 02:00)
- Europe/London (UTC+ 01:00)
- Etc/GMT-1 (UTC+ 01:00)
- Europe/Jersey (UTC+ 01:00)
- Europe/Guernsey (UTC+ 01:00)
- Europe/Isle_of_Man (UTC+ 01:00)
- Africa/Tunis (UTC+ 01:00)
- Africa/Malabo (UTC+ 01:00)
- GB-Eire (UTC+ 01:00)
- Africa/Lagos (UTC+ 01:00)
- Africa/Algiers (UTC+ 01:00)
- GB (UTC+ 01:00)
- Portugal (UTC+ 01:00)
- Africa/Sao_Uhrzeit (UTC+ 01:00)
- Africa/Ndjamena (UTC+ 01:00)
- Atlantic/Faeroe (UTC+ 01:00)
- Irland (UTC+ 01:00)

- Atlantic/Faroe (UTC+ 01:00)
- Europe/Dublin (UTC+ 01:00)
- Africa/Libreville (UTC+ 01:00)
- Africa/El_Aaiun (UTC+ 01:00)
- Africa/El_Aaiun (UTC+ 01:00)
- Africa/Douala (UTC+ 01:00)
- Africa/Brazzaville (UTC+ 01:00)
- Africa/Porto-Novo (UTC+ 01:00)
- Atlantic/Madeira (UTC+ 01:00)
- Europe/Lisbon (UTC+ 01:00)
- Atlantic/Canary (UTC+ 01:00)
- Africa/Casablanca (UTC+ 01:00)
- Europe/Belfast (UTC+ 01:00)
- Africa/Luanda (UTC+ 01:00)
- Africa/Kinshasa (UTC+ 01:00)
- Africa/Bangui (UTC+ 01:00)
- NASS (UTC+ 01:00)
- Africa/Niamey (UTC+ 01:00)
- GMT (UTC+ 00:00)
- Etc/GMT-0 (UTC+ 00:00)
- Atlantic/St_Helena (UTC+ 00:00)
- Etc/GMT+0 (UTC+ 00:00)
- Africa/Banjul (UTC+ 00:00)
- Etc/GMT (UTC+ 00:00)
- Africa/Freetown (UTC+ 00:00)
- Africa/Bamako (UTC+ 00:00)
- Africa/Conakry (UTC+ 00:00)
- Universell (UTC+ 00:00)
- Africa/Nouakchott (UTC+ 00:00)
- UTC (UTC+ 00:00)

- Etc/Universal (UTC+ 00:00)
- Atlantic/Azores (UTC+ 00:00)
- Africa/Abidjan (UTC+ 00:00)
- Africa/Accra (UTC+ 00:00)
- Etc/UCT (UTC+ 00:00)
- GMT0 (UTC+ 00:00)
- Zulu (UTC+ 00:00) Zulu (UTC+ 00:00)
- Africa/Ouagadougou (UTC+ 00:00)
- Atlantic/Reykjavik (UTC+ 00:00)
- Etc/Zulu (UTC+ 00:00)
- Island (UTC+ 00:00)
- Africa/Lome (UTC+ 00:00)
- Greenwich (UTC+ 00:00)
- Etc/GMT0 (UTC+ 00:00)
- America/Danmarkshavn (UTC+ 00:00)
- Africa/Dakar (UTC+ 00:00)
- Africa/Bissau (UTC+ 00:00)
- Etc/Greenwich (UTC+ 00:00)
- Africa/Timbuktu (UTC+ 00:00)
- UTC (UTC+ 00:00)
- Africa/Monrovia (UTC+ 00:00)
- Etc/UTC (UTC+ 00:00)

Tutorials zu Formelausdrücken

Sie können diesen Anleitungen folgen, um Formelausdrücke in zu verwenden. AWS IoT SiteWise

Themen

- [Verwenden Sie Zeichenketten in Formeln](#)
- [Datenpunkte filtern](#)
- [Zählen Sie Datenpunkte, die einer Bedingung entsprechen](#)

- [Späte Daten in Formeln](#)
- [Datenqualität in Formeln](#)
- [Undefinierte, unendliche und Überlaufwerte](#)

Verwenden Sie Zeichenketten in Formeln

Sie können mit Zeichenketten in Ihren Formelausdrücken arbeiten. Sie können auch Zeichenketten aus Variablen eingeben, die auf Attribut- und Messeigenschaften verweisen.

Important

Formelausdrücke können nur Doppel- oder Zeichenkettenwerte ausgeben. Verschachtelte Ausdrücke können andere Datentypen ausgeben, z. B. Zeichenfolgen, aber die Formel als Ganzes muss eine Zahl oder Zeichenfolge ergeben. Sie können die [Funktion jp](#) verwenden, um eine Zeichenfolge in eine Zahl umzuwandeln. Der boolesche Wert muss 1 (wahr) oder 0 (falsch) sein. Weitere Informationen finden Sie unter [Undefinierte, unendliche und Überlaufwerte](#).

AWS IoT SiteWise stellt die folgenden Funktionen für Formelausdrücke bereit, mit denen Sie Zeichenfolgen bearbeiten können:

- [Zeichenkettenlitterale](#)
- Der [Indexoperator](#) (`s[index]`)
- Der [Slice-Operator](#) (`s[start:end:step]`)
- [Vergleichsfunktionen](#), mit denen Sie Zeichenketten in [lexikografischer](#) Reihenfolge vergleichen können
- [Zeichenkettenfunktionen](#), zu denen auch die `jp` Funktion gehört, die serialisierte JSON-Objekte analysieren und Zeichenketten in Zahlen konvertieren kann

Datenpunkte filtern

Sie können die [if-Funktion](#) verwenden, um Datenpunkte herauszufiltern, die eine Bedingung nicht erfüllen. Die `if` Funktion wertet eine Bedingung aus und gibt unterschiedliche Werte `true` und `false` Ergebnisse zurück. Sie können die [Konstante none](#) als Ausgabe für einen Fall einer `if` Funktion verwenden, um den Datenpunkt für diesen Fall zu verwerfen.

Um Datenpunkte herauszufiltern, die einer Bedingung entsprechen

- Erstellen Sie eine Transformation, die die `if` Funktion verwendet, um eine Bedingung zu definieren, die prüft, ob eine Bedingung erfüllt ist, und entweder den `result_if_false` Wert `result_if_true` oder zurückgibt `none`.

Example Beispiel: Filtert Datenpunkte heraus, an denen das Wasser nicht kocht

Stellen Sie sich ein Szenario vor, in dem Sie eine Messung durchführentemp_c, die die Temperatur (in Celsius) des Wassers in einer Maschine angibt. Sie können die folgende Transformation definieren, um Datenpunkte herauszufiltern, an denen das Wasser nicht kocht:

- Transformation: `boiling_temps = if(gte(temp_c, 100), temp_c, none)` — Gibt die Temperatur zurück, wenn sie größer oder gleich 100 Grad Celsius ist, andernfalls wird kein Datenpunkt zurückgegeben.

Zählen Sie Datenpunkte, die einer Bedingung entsprechen

Sie können [Vergleichsfunktionen](#) und [sum\(\)](#) verwenden, um die Anzahl der Datenpunkte zu zählen, für die eine Bedingung zutrifft.

Um Datenpunkte zu zählen, die einer Bedingung entsprechen

1. Erstellen Sie eine Transformation, die eine Vergleichsfunktion verwendet, um eine Filterbedingung für eine andere Eigenschaft zu definieren.
2. Erstellen Sie eine Metrik, die die Datenpunkte summiert, für die diese Bedingung erfüllt ist.

Example Beispiel: Zählen Sie die Anzahl der Datenpunkte, bei denen Wasser kocht

Stellen Sie sich ein Szenario vor, in dem Sie über eine Messung verfügentemp_c, die die Temperatur (in Celsius) des Wassers in einer Maschine angibt. Sie können die folgenden Transformations- und Metrikeigenschaften definieren, um die Anzahl der Datenpunkte zu zählen, bei denen das Wasser kocht:

- Transformation: `is_boiling = gte(temp_c, 100)` — Gibt zurück, 1 ob die Temperatur größer oder gleich 100 Grad Celsius ist, andernfalls wird zurückgegeben 0.
- Metrisch: `boiling_count = sum(is_boiling)` — Gibt die Anzahl der Datenpunkte zurück, an denen Wasser kocht.

Späte Daten in Formeln

AWS IoT SiteWise unterstützt die späte Datenaufnahme von Daten, die bis zu 7 Tage alt sind. Wenn verspätete Daten AWS IoT SiteWise empfangen werden, werden vorhandene Werte für jede Metrik neu berechnet, die die verspäteten Daten in einem vergangenen Fenster eingibt. Diese Neuberechnungen führen zu Gebühren für die Datenverarbeitung.

Note

Bei der AWS IoT SiteWise Berechnung von Eigenschaften, die verspätete Daten eingeben, wird der aktuelle Formelausdruck jeder Eigenschaft verwendet.

Nachdem ein vergangenes Fenster für eine Metrik AWS IoT SiteWise neu berechnet wurde, ersetzt es den vorherigen Wert für dieses Fenster. Wenn Sie Benachrichtigungen für diese Metrik aktiviert haben, wird AWS IoT SiteWise auch eine Benachrichtigung über den Eigenschaftswert ausgegeben. Dies bedeutet, dass Sie eine neue Benachrichtigung zum Aktualisieren von Eigenschaftswerten für dieselbe Eigenschaft und denselben Zeitstempel erhalten können, für die Sie zuvor bereits eine Benachrichtigung erhalten haben. Wenn Ihre Anwendungen oder Data Lakes Eigenschaftswertbenachrichtigungen verwenden, müssen Sie den vorherigen Wert mit dem neuen Wert aktualisieren, damit die Daten weiterhin korrekt sind.

Datenqualität in Formeln

AWS IoT SiteWise hat jeder Datenpunkt einen Qualitätscode, der einer der folgenden sein kann:

- GOOD— Die Daten sind von keinen Problemen betroffen.
- BAD— Die Daten sind von einem Problem wie einem Sensorausfall betroffen.
- UNCERTAIN— Die Daten sind von einem Problem wie einer Sensorungenauigkeit betroffen.

AWS IoT SiteWise verbraucht bei der Berechnung von Transformationen und Metriken nur GOOD hochwertige Daten. AWS IoT SiteWise gibt nur GOOD Qualitätsdaten für erfolgreiche Berechnungen aus. Wenn eine Berechnung nicht erfolgreich ist, wird AWS IoT SiteWise kein Datenpunkt für diese Berechnung ausgegeben. Dies kann auftreten, wenn eine Berechnung zu einem undefinierten, unendlichen oder Überlaufwert führt.

Weitere Informationen zum Abfragen von Daten und zum Filtern nach Datenqualität finden Sie unter [Daten abfragen von AWS IoT SiteWise](#).

Undefinierte, unendliche und Überlaufwerte

Einige Formelausdrücke (wie $x / 0$, $\sqrt{-1}$, oder $\log(0)$) berechnen Werte, die in einem reellen Zahlensystem undefiniert, unendlich oder außerhalb des von unterstützten Bereichs liegen. AWS IoT SiteWise Wenn der Ausdruck einer Anlageneigenschaft einen undefinierten, unendlichen Wert oder einen Überlaufwert berechnet, wird AWS IoT SiteWise kein Datenpunkt für diese Berechnung ausgegeben.

AWS IoT SiteWise gibt auch keinen Datenpunkt aus, wenn ein nicht numerischer Wert als Ergebnis eines Formelausdrucks berechnet wird. Das bedeutet, dass, wenn Sie eine Formel definieren, die eine Zeichenfolge, ein Array oder die [Konstante none](#) berechnet, AWS IoT SiteWise kein Datenpunkt für diese Berechnung ausgegeben wird.

Example Beispiele

Jeder der folgenden Formelausdrücke führt zu einem Wert, der nicht als Zahl dargestellt AWS IoT SiteWise werden kann. AWS IoT SiteWise gibt bei der Berechnung dieser Formelausdrücke keinen Datenpunkt aus.

- $x / 0$ ist undefiniert.
- $\log(0)$ ist undefiniert.
- $\sqrt{-1}$ ist in einem reellen Zahlensystem undefiniert.
- "hello" + " world" ist eine Zeichenfolge.
- `jp({'values':[3,6,7]}, '$.values')` ist ein Array.
- `if(gte(temp, 300), temp, none)` ist none wenn temp ist weniger als 300.

Erstellen Sie benutzerdefinierte Verbundmodelle (Komponenten)

Benutzerdefinierte zusammengesetzte Modelle oder Komponenten, wenn Sie die Konsole verwenden, bieten eine weitere Organisationsebene für Ihre Asset- und Komponentenmodelle. Sie können sie verwenden, um Ihre Modelle zu strukturieren, indem Sie Eigenschaften gruppieren oder auf andere Modelle verweisen. Weitere Informationen zum Arbeiten mit benutzerdefinierten Verbundmodellen finden Sie unter [Benutzerdefinierte zusammengesetzte Modelle \(Komponenten\)](#)

Sie erstellen ein benutzerdefiniertes Verbundmodell innerhalb eines vorhandenen Objekt- oder Komponentenmodells. Es gibt zwei Arten von benutzerdefinierten Verbundmodellen. Um verwandte Eigenschaften innerhalb eines Modells zu gruppieren, können Sie ein benutzerdefiniertes Verbundmodell erstellen. Um in Ihrem Objekt- oder Komponentenmodell auf ein Komponentenmodell

zu verweisen, können Sie ein auf einem Komponentenmodell basierendes benutzerdefiniertes Verbundmodell erstellen.

In den folgenden Abschnitten wird beschrieben, wie Sie mithilfe der AWS IoT SiteWise API benutzerdefinierte Verbundmodelle erstellen.

Themen

- [Erstellen Sie eine Inline-Komponente \(Konsole\)](#)
- [Erstellen Sie ein benutzerdefiniertes Inline-Verbundmodell \(AWS CLI\)](#)
- [Erstellen Sie eine auf einem Komponentenmodell basierende Komponente \(Konsole\)](#)
- [Erstellen Sie ein auf einem Komponentenmodell basierendes benutzerdefiniertes Verbundmodell \(AWS CLI\)](#)

Erstellen Sie eine Inline-Komponente (Konsole)

Sie können die AWS IoT SiteWise Konsole verwenden, um eine Inline-Komponente zu erstellen, die ihre eigenen Eigenschaften definiert.

Note

Da es sich um eine Inline-Komponente handelt, gelten diese Eigenschaften nur für das aktuelle Asset-Modell und werden nirgendwo anders gemeinsam genutzt.

Wenn Sie ein wiederverwendbares Modell erstellen müssen (z. B. um mehrere Objektmodelle gemeinsam zu nutzen oder um mehrere Instanzen in ein Objektmodell einzubeziehen), sollten Sie stattdessen eine Komponente erstellen, die auf einem Komponentenmodell basiert. Einzelheiten finden Sie im folgenden Abschnitt.

Um eine Komponente (Konsole) zu erstellen

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Klicken Sie im Navigationsbereich auf Models (Modelle).
3. Wählen Sie das Asset-Modell aus, zu dem Sie eine Komponente hinzufügen möchten.
4. Wählen Sie auf der Registerkarte Eigenschaften die Option Komponenten aus.
5. Wählen Sie Komponente erstellen.
6. Gehen Sie auf der Seite Komponente erstellen wie folgt vor:

- a. Geben Sie einen Namen für die Komponente ein, z. B. **ServoMotor** oder **ServoMotor Model1**. Dieser Name muss für alle Komponenten in Ihrem Konto in dieser Region eindeutig sein.
- b. (Optional) Fügen Sie Attributdefinitionen für das Modell hinzu. Attribute stellen Informationen dar, die sich selten ändern. Weitere Informationen finden Sie unter [Definieren Sie statische Daten \(Attribute\)](#).
- c. (Optional) Fügen Sie Messungsdefinitionen für das Modell hinzu. Messungen stellen Datenströme von Ihren Geräten dar. Weitere Informationen finden Sie unter [Definieren Sie Datenströme von Geräten \(Messungen\)](#).
- d. (Optional) Fügen Sie Transformationsdefinitionen für das Modell hinzu. Transformationen sind Formeln, die Daten von einem Formular auf ein anderes abbilden. Weitere Informationen finden Sie unter [Daten transformieren \(transformiert\)](#).
- e. (Optional) Fügen Sie Metrik-Definitionen für das Modell hinzu. Metriken sind Formeln, die Daten über Zeitintervalle aggregieren. Mit Metriken können Daten aus zugehörigen Anlagen eingegeben werden, sodass Sie Werte berechnen können, die Ihren Betrieb oder einen Teil Ihres Betriebs repräsentieren. Weitere Informationen finden Sie unter [Aggregieren Sie Daten aus Immobilien und anderen Vermögenswerten \(Metriken\)](#).
- f. Wählen Sie Komponente erstellen aus.

Erstellen Sie ein benutzerdefiniertes Inline-Verbundmodell (AWS CLI)

Sie können das AWS Command Line Interface (AWS CLI) verwenden, um ein benutzerdefiniertes Inline-Verbundmodell zu erstellen, das seine eigenen Eigenschaften definiert.

Verwenden Sie die [CreateAssetModelCompositeModel](#) Operation, um ein Inline-Modell mit Eigenschaften zu erstellen. Diese Operation erwartet eine Nutzlast mit der folgenden Struktur.

Note

Da es sich um ein zusammengesetztes Inline-Modell handelt, gelten diese Eigenschaften nur für das aktuelle Anlagenmodell und werden nirgendwo anders verwendet. Was es „inline“ macht, ist, dass es keinen Wert für das `composedAssetModelId` Feld bereitstellt. Wenn Sie ein wiederverwendbares Modell erstellen müssen (z. B. um es von mehreren Asset-Modellen gemeinsam zu nutzen oder um mehrere Instanzen in ein Asset-Modell

einzubeziehenden), sollten Sie stattdessen ein auf einem Komponentenmodell basierendes Verbundmodell erstellen. Einzelheiten finden Sie im folgenden Abschnitt.

```
{
  "assetModelCompositeModelName": "CNCLathe_ServoMotorA",
  "assetModelCompositeModelType": "CUSTOM",
  "assetModelCompositeModelProperties": [
    {
      "dataType": "DOUBLE",
      "name": "Servo Motor Temperature",
      "type": {
        "measurement": {}
      },
      "unit": "Celsius"
    },
    {
      "dataType": "DOUBLE",
      "name": "Spindle speed",
      "type": {
        "measurement": {}
      },
      "unit": "rpm"
    }
  ]
}
```

Erstellen Sie eine auf einem Komponentenmodell basierende Komponente (Konsole)

Sie können die AWS IoT SiteWise Konsole verwenden, um eine Komponente zu erstellen, die auf einem Komponentenmodell basiert.

Um eine Komponente zu erstellen, die auf einem Komponentenmodell basiert (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Klicken Sie im Navigationsbereich auf Models (Modelle).
3. Wählen Sie das Asset-Modell aus, dem Sie eine Komponente hinzufügen möchten.
4. Wählen Sie auf der Registerkarte Eigenschaften die Option Komponenten aus.
5. Wählen Sie Komponente erstellen.
6. Gehen Sie auf der Seite Komponente erstellen wie folgt vor:

- a. Wählen Sie das Komponentenmodell aus, auf dem die Komponente basieren soll.
- b. Geben Sie einen Namen für die Komponente ein, z. B. **ServoMotor** oder **ServoMotor Model1**. Dieser Name muss für alle Komponenten in Ihrem Konto in dieser Region eindeutig sein.
- c. Wählen Sie Komponente erstellen aus.

Erstellen Sie ein auf einem Komponentenmodell basierendes benutzerdefiniertes Verbundmodell (AWS CLI)

Sie können das verwenden AWS CLI, um ein auf einem Komponentenmodell basierendes benutzerdefiniertes Verbundmodell innerhalb Ihres Anlagenmodells zu erstellen. Ein auf einem Komponentenmodell basierendes benutzerdefiniertes Verbundmodell ist ein Verweis auf ein Komponentenmodell, das Sie bereits an anderer Stelle definiert haben.

Verwenden Sie den [CreateAssetModelCompositeModel](#) Vorgang, um ein auf einem Komponentenmodell basierendes benutzerdefiniertes Verbundmodell zu erstellen. Diese Operation erwartet eine Nutzlast mit der folgenden Struktur.

Note

In diesem Beispiel `composedAssetModelId` ist der Wert von die Objektmodell-ID oder die externe ID eines vorhandenen Komponentenmodells. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch. Ein Beispiel für die Erstellung eines Komponentenmodells finden Sie unter [Erstellen Sie ein Komponentenmodell \(AWS CLI\)](#).

```
{
  "assetModelCompositeModelName": "CNCLathe_ServoMotorA",
  "assetModelCompositeModelType": "CUSTOM",
  "composedAssetModelId": component model ID
}
```

Da es sich nur um eine Referenz handelt, hat ein auf einem Komponentenmodell basierendes benutzerdefiniertes Verbundmodell außer einem Namen keine eigenen Eigenschaften.

Wenn Sie Ihrem Anlagenmodell mehrere Exemplare derselben Komponente hinzufügen möchten (z. B. eine CNC-Maschine mit mehreren Servomotoren), können Sie mehrere auf Komponentenmodellen basierende benutzerdefinierte Verbundmodelle hinzufügen, die jeweils einen eigenen Namen haben, aber alle auf denselben Namen verweisen. `composedAssetModelId`

Sie können Komponenten innerhalb anderer Komponenten verschachteln. Dazu können Sie einem Ihrer Komponentenmodelle ein auf einem Komponentenmodell basierendes Verbundmodell hinzufügen, wie in diesem Beispiel gezeigt.

Erstellen Sie Objekte für Objektmodelle inAWS IoT SiteWise

Sie können eine Komponente aus einem Komponentenmodell erstellen. Sie müssen über ein Komponentenmodell verfügen, bevor Sie eine Komponente erstellen können. Wenn Sie noch kein Komponentenmodell erstellt haben, beachten Sie die Informationen im Abschnitt [Erstellen Sie Asset-Modelle inAWS IoT SiteWise](#).

Note

Sie können nur Komponenten anhand von Modellen mit dem Status ACTIVE erstellen. Wenn der Status Ihres Modells nicht ACTIVE lautet, müssen Sie möglicherweise einige Minuten warten, bevor Sie Komponenten von diesem Modell ausgehend erstellen können. Weitere Informationen finden Sie unter [Komponenten- und Modellzustände](#).

Themen

- [Erstellen Sie eine Anlage \(Konsole\)](#)
- [Erstellen Sie ein Asset \(AWS CLI\)](#)
- [Konfigurieren Sie ein neues Asset](#)

Erstellen Sie eine Anlage (Konsole)

Sie können die AWS IoT SiteWise Konsole verwenden, um ein Asset zu erstellen.

So erstellen Sie eine Komponente (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).

2. Wählen Sie im linken Navigationsbereich Komponenten aus.
3. Wählen Sie dann Create asset (Komponente erstellen) aus.
4. Gehen Sie auf der Seite Komponente erstellen wie folgt vor:
 - a. Wählen Sie unter Modell das Komponentenmodell aus, aus dem eine Komponente erstellt werden soll.

 Note

Wenn Ihr Modell nicht AKTIV ist, müssen Sie warten, bis es aktiv ist, oder Probleme beheben, wenn es FEHLGESCHLAGEN ist.

- b. Geben Sie unter Name einen Namen für Ihre Komponente ein.
- c. (Optional) Fügen Sie Tags für Ihre Komponente hinzu. Weitere Informationen finden Sie unter [Kennzeichnen Sie Ihre AWS IoT SiteWise Ressourcen](#).
- d. Wählen Sie dann Create asset (Komponente erstellen) aus.

Wenn Sie ein Asset erstellen, navigiert die AWS IoT SiteWise Konsole zur Seite des neuen Assets. Auf dieser Seite sehen Sie den Status der Komponente, der anfänglich WIRD ERSTELLT lautet. Diese Seite wird automatisch aktualisiert. Sie können daher einfach abwarten, bis der Status der Komponente aktualisiert wird.

 Note

Die Komponentenerstellung kann bis zu einer Minute dauern. Wenn der Status AKTIV ist, können Sie Aktualisierungsvorgänge für Ihr Asset durchführen. Weitere Informationen finden Sie unter [Komponenten- und Modellzustände](#).

Nachdem Sie eine Komponente erstellt haben, finden Sie weitere Informationen unter [Konfigurieren Sie ein neues Asset](#).

Erstellen Sie ein Asset (AWS CLI)

Sie können das AWS Command Line Interface (AWS CLI) verwenden, um ein Asset aus einem Asset-Modell zu erstellen.

Sie müssen über eine `assetModelId` verfügen, um eine Komponente zu erstellen. Wenn Sie ein Asset-Modell erstellt haben, es aber nicht kennen, verwenden Sie die [ListAssetModels](#) API, um alle Ihre Asset-Modelle anzuzeigen.

Verwenden Sie die [CreateAsset](#) API mit den folgenden Parametern, um ein Asset aus einem Asset-Modell zu erstellen:

- `assetName`— Der Name des neuen Assets. Geben Sie Ihrem Asset einen Namen, damit Sie es leichter identifizieren können.
- `assetModelId`— Die ID des Vermögenswerts. Dies ist die tatsächliche ID im UUID-Format, oder die, `externalId:myExternalId` falls sie eine hat. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.

Um ein Asset zu erstellen (AWS CLI)

- Führen Sie den folgenden Befehl aus, um eine Komponente zu erstellen. `asset-name` Ersetzen Sie es durch einen Namen für das Asset und `asset-model-id` durch die ID oder die externe ID des Asset-Modells.

```
aws iotsitewise create-asset \  
  --asset-name asset-name \  
  --asset-model-id asset-model-id
```

Die Operation gibt eine Antwort zurück, in der die Angaben und der Status der neuen Komponente im folgenden Format enthalten sind.

```
{  
  "assetId": "String",  
  "assetArn": "String",  
  "assetStatus": {  
    "state": "String",  
    "error": {  
      "code": "String",  
      "message": "String"  
    }  
  }  
}
```

Der `state` der Komponente ist `CREATING`, bis die Komponente erstellt wird.

 Note

Die Komponentenerstellung kann bis zu einer Minute dauern. Um den Status Ihres Assets zu überprüfen, verwenden Sie den [DescribeAsset](#) Vorgang mit der ID Ihres Assets als `assetId` Parameter. Sobald das Asset fertig `state` ist `ACTIVE`, können Sie Aktualisierungsvorgänge an Ihrem Asset durchführen. Weitere Informationen finden Sie unter [Komponenten- und Modellzustände](#).

Nachdem Sie eine Komponente erstellt haben, finden Sie weitere Informationen unter [Konfigurieren Sie ein neues Asset](#).

Konfigurieren Sie ein neues Asset

Nachdem Sie eine Anlage in erstellt haben AWS IoT SiteWise, können Sie mehrere weitere Schritte unternehmen, um die Anlage und ihre Daten vollständig zu nutzen. Diese Schritte können die Konfiguration von Datenströmen zur Aufnahme von Daten aus der Anlage, die Einrichtung von Alarmen und Benachrichtigungen zur Überwachung der Leistung der Anlage, die Erstellung von Visualisierungen und Dashboards zur Anzeige der Anlagendaten und die Integration der Anlage in andere AWS Dienste oder Drittanbieteranwendungen zur weiteren Analyse oder Automatisierung umfassen.

Schließen Sie die Konfiguration Ihres Assets mit den folgenden optionalen Aktionen ab:

- [Datenströme verwalten für AWS IoT SiteWise](#), wenn Ihre Komponente über Messeigenschaften verfügt.
- [Attributwerte aktualisieren](#), wenn Ihre Komponente über eindeutige Attributwerte verfügt.
- [Anlagen zuordnen und deren Zuordnung aufheben](#), wenn Ihre Komponente eine übergeordnete Komponente ist.

Suchen Sie nach Assets auf AWS-IoT-SiteWise-Konsole

Verwenden Sie die AWS-IoT-SiteWise-Konsole Suchfunktion, um Assets auf der Grundlage von Metadaten und Echtzeitfiltern für Eigenschaftswerte zu finden.

Voraussetzungen

AWS IoT SiteWise erfordert Genehmigungen für die Integration von Industriedaten AWS IoT TwinMaker, um sie besser organisieren und modellieren zu können. Wenn Sie Berechtigungen dafür erteilt haben AWS IoT SiteWise, verwenden Sie die [ExecuteQuery](#) API. Wenn Sie noch keine Berechtigungen erteilt haben und Hilfe bei den AWS IoT SiteWise ersten Schritten benötigen, finden Sie weitere Informationen unter [Integrieren AWS IoT SiteWise und AWS IoT TwinMaker](#).

Erweiterte Suche auf AWS-IoT-SiteWise-Konsole

Suche nach Metadaten

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im Navigationsbereich unter Assets die Option Erweiterte Suche aus.
3. Wählen Sie unter Erweiterte Suche die Option Metadatensuche aus.
4. Füllen Sie die Parameter aus. Füllen Sie für eine effiziente Suche so viele Felder wie möglich aus.
 - a. Assetname — Geben Sie einen vollständigen Asset-Namen oder einen Teil des Namens für eine umfassende Suche ein.
 - b. Eigenschaftsname — Geben Sie einen vollständigen oder einen Teil des Namens für eine umfassende Suche ein.
 - c. Operator — Wählen Sie einen Operator aus:
 - =
 - <
 - >
 - <=
 - >=
 - d. Immobilienwert — Dieser Wert wird mit dem aktuellen Wert der Immobilie verglichen.
 - e. Eigenschaftswerttyp — Der Datentyp der Eigenschaft. Wählen Sie eine der folgenden Optionen aus:
 - Doppelt
 - Ganzzahl
 - Zeichenfolge

- Boolesch
- Wählen Sie Search (Suchen) aus.
 - Wählen Sie in der Tabelle mit den Suchergebnissen das Asset aus der Spalte Name aus. Dadurch gelangen Sie zur detaillierten Asset-Seite für dieses Asset.

Assets

Assets represent Industrial devices and processes that send data streams to SiteWise. Models are structures that enforce a specific model of properties and hierarchies for all instances of each asset. You must create every asset from a model.

Advanced search
Use advanced search to find assets based on specific metadata. In addition, you can enter SQL queries directly in the query builder.

Metadata search | Query builder

Asset name:

Property name:

Operator:

Property value:

Property value type:

Search results (2)

Name	Asset id	Description
Level-2-asset-1	d0e9019b-9c38-4316-b574-38317aa38143	
Level-2-asset-2	b9c0d2fc-1527-42ce-8ba2-d1a4e8ff43de	Example description

Partial search (Partielle Suche)

Für eine Asset-Suche müssen nicht alle Parameter angegeben werden. Hier sind einige Beispiele für partielle Suchen mit der Metadatensuchoption:

- Finden Sie Assets anhand ihres Namens:
 - Geben Sie einen Wert nur in das Feld Assetname ein.
 - Die Felder Eigenschaftsname und Eigenschaftswert sind leer.
- Suchen Sie nach Assets, die Eigenschaften mit einem bestimmten Namen enthalten:
 - Geben Sie einen Wert nur in das Feld Eigenschaftsname ein.
 - Die Felder „Assetname“ und „Eigenschaftswert“ sind leer.
- Finden Sie Vermögenswerte anhand der neuesten Werte ihrer Eigenschaften:

- Geben Sie Werte in die Felder Eigenschaftsname und Eigenschaftswert ein.
- Wählen Sie einen Operator und einen Eigenschaftswerttyp aus.

Suche im Query Builder

1. Navigieren Sie zur AWS-IoT-SiteWise-Konsole.
2. Wählen Sie im Navigationsbereich unter Assets die Option Erweiterte Suche aus.
3. Wählen Sie unter Erweiterte Suche die Option Query Builder aus.
4. Schreiben Sie im Bereich Query Builder Ihre SQL-Abfrage, um einasset_name, asset_id und abzurufenasset_description.
5. Wählen Sie Search (Suchen) aus.
6. Wählen Sie in der Tabelle mit den Suchergebnissen das Asset aus der Spalte Name aus. Dadurch gelangen Sie zur detaillierten Asset-Seite für dieses Asset.

The screenshot shows the AWS IoT SiteWise console interface. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and a region dropdown set to 'N. Virginia'. Below this, the 'Assets' section is active. The 'Assets' page has a 'Create asset' button and a description: 'Assets represent Industrial devices and processes that send data streams to SiteWise. Models are structures that enforce a specific model of properties and hierarchies for all instances of each asset. You must create every asset from a model.'

The 'Advanced search' section is expanded, showing two tabs: 'Metadata search' and 'Query builder'. The 'Query builder' tab is selected, displaying a SQL query in a text area:


```
SELECT a.asset_id, a.asset_name, a.asset_description
FROM asset a, asset_property p, latest_value_time_series ts
WHERE a.asset_name LIKE '%asset-2%' AND a.property_name = 'temperature_f' AND ts.double_value > 50.0
```

 Below the query area are 'Clear' and 'Search' buttons. The 'Search' button has been clicked, resulting in 'Search results (2)'. The results are displayed in a table with columns: Name, Asset id, and Description. Two results are shown: 'Level-2a-asset-2' with asset_id '4fed596d-e903-4338-86db-34ca9301233a' and description 'Generator #3', and 'Level-2b-asset-2' with asset_id 'b4ac2b24-4fce-4a72-9fea-ef6d0f741e8d' and description 'Generator #2'.

Name	Asset id	Description
Level-2a-asset-2	4fed596d-e903-4338-86db-34ca9301233a	Generator #3
Level-2b-asset-2	b4ac2b24-4fce-4a72-9fea-ef6d0f741e8d	Generator #2

 Note

- Die SELECT Klausel in der SQL-Abfrage muss die `asset_id` Felder `asset_name` und enthalten, um sicherzustellen, dass die Tabelle mit den Suchergebnissen ein gültiges Asset enthält.
- Der Abfrage-Generator zeigt in der Ergebnistabelle nur den Namen, die Asset-ID und die Beschreibung an. Durch das Hinzufügen weiterer Felder zur SELECT Klausel werden der Ergebnistabelle keine weiteren Spalten hinzugefügt

Attributwerte aktualisieren

Komponenten übernehmen die Attribute ihres Komponentenmodells, einschließlich des Standardwerts des Attributs. In bestimmten Fällen möchten Sie u. U. das Standardattribut des Komponentenmodells beibehalten, z. B. für die Eigenschaft des Komponentenherstellers. In anderen Fällen möchten Sie u. U. das übernommene Attribut aktualisieren, beispielsweise für den Breiten- und Längengrad einer Komponente.

Updating an attribute value (console)

Sie können die AWS IoT SiteWise Konsole verwenden, um den Wert einer Attribut-Asset-Eigenschaft zu aktualisieren.

So aktualisieren Sie den Wert eines Attributs (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im linken Navigationsbereich Komponenten aus.
3. Wählen Sie die Komponente aus, für die Sie ein Attribut aktualisieren möchten.

 Tip

Sie können eine Komponentenhierarchie mithilfe des Pfeilsymbols erweitern, um nach Ihrer Komponente zu suchen.

4. Wählen Sie Bearbeiten aus.
5. Suchen Sie das zu aktualisierende Attribut und geben Sie dann den neuen Wert ein.

6. Wählen Sie Speichern.

Updating an attribute value (AWS CLI)

Sie können das AWS Command Line Interface (AWS CLI) verwenden, um einen Attributwert zu aktualisieren.

Um dieses Verfahren abzuschließen, müssen Sie die `assetId` Ihrer Komponenten und die `propertyId` Ihrer Eigenschaft kennen. Sie können auch die externe ID verwenden. Wenn Sie ein Asset erstellt haben und es nicht kennen `assetId`, verwenden Sie die [ListAssets](#) API, um alle Assets für ein bestimmtes Modell aufzulisten. Verwenden Sie den [DescribeAsset](#) Vorgang, um die Eigenschaften Ihres Assets einschließlich der Eigenschafts-IDs anzuzeigen.

Verwenden Sie die Operation [BatchPutAssetPropertyValue](#), um Attributwerte zu Ihrer Komponente zuzuweisen. Mit dieser Operation können Sie mehrere Attribute gleichzeitig festlegen. Die Nutzlast dieser Operation enthält eine Liste von Einträgen, jeweils mit der Komponenten-ID, der Eigenschafts-ID und dem Attributwert.

Um den Wert eines Attributs zu aktualisieren (AWS CLI)

1. Erstellen Sie eine Datei namens `batch-put-payload.json` und kopieren Sie das folgende JSON-Objekt in die Datei. In diesem Nutzlast-Beispiel wird veranschaulicht, wie der Breiten- und Längengrad einer Windturbine festgelegt wird. Aktualisieren Sie die IDs, Werte und Zeitstempel, um die Nutzlast für Ihren Anwendungsfall zu ändern.

```
{
  "entries": [
    {
      "entryId": "windfarm3-turbine7-latitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "propertyValues": [
```

```

    {
      "value": {
        "doubleValue": 47.6204
      },
      "timestamp": {
        "timeInSeconds": 1575691200
      }
    }
  ],
  {
    "entryId": "windfarm3-turbine7-longitude",
    "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE",
    "propertyValues": [
      {
        "value": {
          "doubleValue": 122.3491
        },
        "timestamp": {
          "timeInSeconds": 1575691200
        }
      }
    ]
  }
]
}

```

- Jeder Eintrag in der Nutzlast enthält eine `entryId`, die Sie als eindeutige Zeichenfolge definieren können. Bei fehlgeschlagenen Anforderungseinträgen enthält jeder Fehler die `entryId` der entsprechenden Anforderung, woran Sie erkennen können, welche Anforderungen zu wiederholen sind.
- Um einen Attributwert festzulegen, können Sie für jede Attributeigenschaft eine Zeitstempelqualitätswert-Struktur (TQV) in die Liste von `propertyValues` aufnehmen. Diese Struktur muss den neuen `value` und den aktuellen `timestamp` enthalten.
 - `value`— Eine Struktur, die je nach Typ der festgelegten Eigenschaft eines der folgenden Felder enthält:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`

- `stringValue`
- `nullValue`
- `timestamp`— Eine Struktur, die die aktuelle Unix-Epoche in Sekunden enthält, `timeInSeconds`. AWS IoT SiteWise lehnt alle Datenpunkte mit Zeitstempeln ab, die länger als 7 Tage in der Vergangenheit oder neuer als 5 Minuten in der future existierten.

Weitere Informationen zum Vorbereiten einer Nutzlast für [BatchPutAssetPropertyValue](#) finden Sie unter [Daten aufnehmen mit AWS IoT SiteWise APIs](#).

2. Führen Sie den folgenden Befehl aus, um die Attributwerte an zu senden: AWS IoT SiteWise

```
aws iotsitewise batch-put-asset-property-value -\cli-input-json file://batch-put-payload.json
```

Anlagen zuordnen und deren Zuordnung aufheben

Wenn das Modell Ihrer Komponente Hierarchien für untergeordnete Komponentenmodelle definiert, können Sie Ihrer Komponente untergeordnete Komponenten zuordnen. Übergeordnete Komponenten können von zugehörigen Komponenten aus auf Daten zugreifen und diese aggregieren. Weitere Informationen zu hierarchischen Komponentenmodellen finden Sie unter [Definieren Sie die Hierarchien der Anlagenmodelle](#). Wenn Sie Schnittstellen verwenden, werden die in der Schnittstelle definierten Hierarchien auf die Anlagenmodelle angewendet, die die Schnittstelle implementieren. Weitere Informationen zu Schnittstellen finden Sie unter [Schnittstellen für das Asset-Modell](#)

Themen

- [Elemente zuordnen und deren Zuordnung aufheben \(Konsole\)](#)
- [Vermögenswerte zuordnen und deren Zuordnung aufheben \(AWS CLI\)](#)

Elemente zuordnen und deren Zuordnung aufheben (Konsole)

Sie können die AWS IoT SiteWise Konsole verwenden, um Assets zuzuordnen und zu trennen.

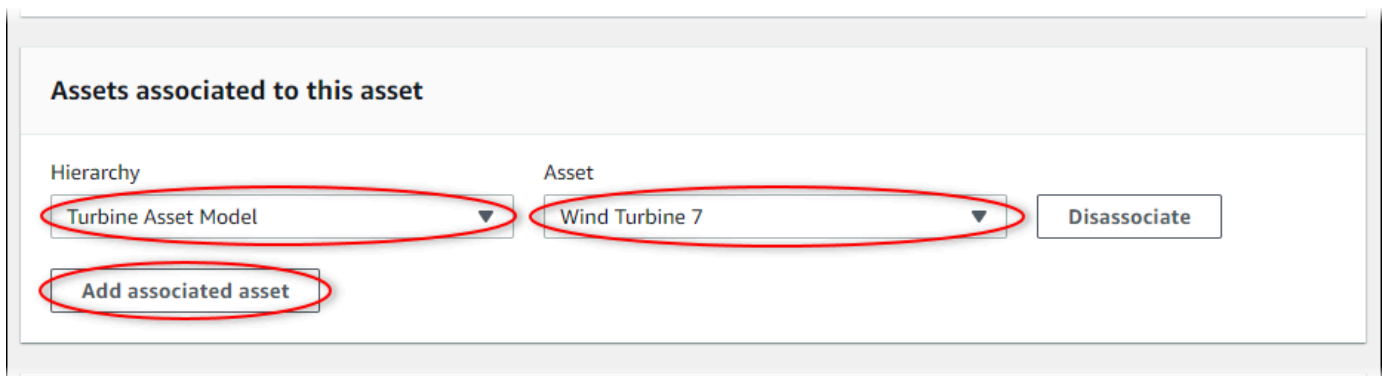
So ordnen Sie eine Komponente zu (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im linken Navigationsbereich Komponenten aus.
3. Wählen Sie die übergeordnete Komponente aus, der Sie eine untergeordnete Komponente zuordnen möchten.

Tip

Sie können eine Komponentenhierarchie mithilfe des Pfeilsymbols erweitern, um nach Ihrer Komponente zu suchen.

4. Wählen Sie Bearbeiten aus.
5. Wählen Sie unter Mit dieser Komponente verknüpfte Komponente die Option Zugehörige Komponente hinzufügen aus.



6. Wählen Sie für Hierarchie die Hierarchie aus, durch die die Beziehung zwischen der übergeordneten Komponente und der untergeordneten Komponente definiert wird.
7. Wählen Sie unter Komponente die untergeordnete Komponente aus, die zugeordnet werden soll.
8. Wählen Sie Speichern.

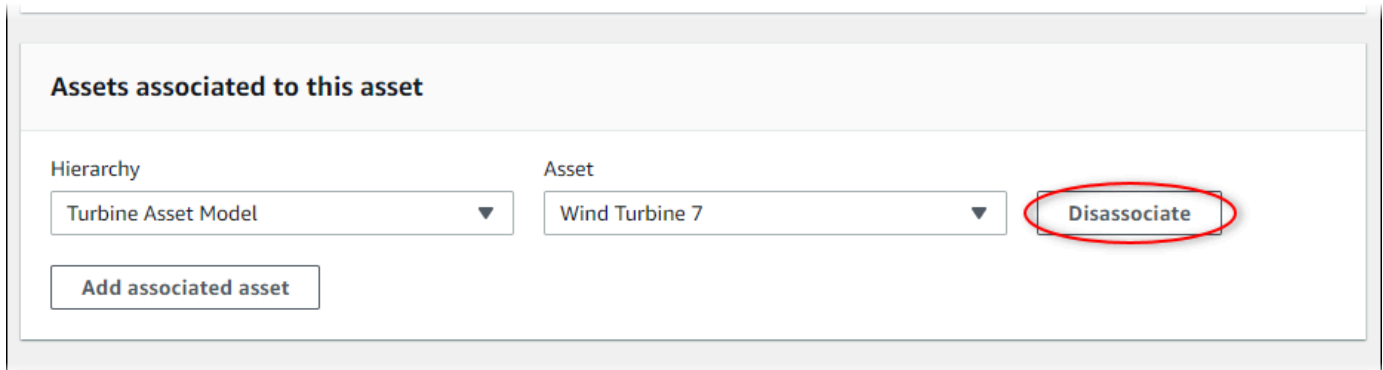
So heben Sie die Zuordnung einer Komponente auf (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im linken Navigationsbereich Komponenten aus.
3. Wählen Sie die übergeordnete Komponente aus, für die Sie die Zuordnung einer untergeordneten Komponente aufheben möchten.

Tip

Sie können eine Komponentenhierarchie mithilfe des Pfeilsymbols erweitern, um nach Ihrer Komponente zu suchen.

4. Wählen Sie Bearbeiten aus.
5. Wählen Sie unter Mit dieser Komponente verknüpfte Komponenten für die Komponente die Option Zuordnung aufheben aus.



Assets associated to this asset

Hierarchy: Turbine Asset Model ▼

Asset: Wind Turbine 7 ▼

Disassociate

Add associated asset

6. Wählen Sie Speichern.

Vermögenswerte zuordnen und deren Zuordnung aufheben (AWS CLI)

Sie können das AWS Command Line Interface (AWS CLI) verwenden, um Anlagen zuzuordnen oder zu trennen.

Für dieses Verfahren müssen Sie die ID der Hierarchie (`hierarchyId`) im übergeordneten Komponentenmodell kennen, durch die die Beziehung zum untergeordneten Komponentenmodell definiert wird. Verwenden Sie die [DescribeAsset](#) Operation, um die Hierarchie-ID in der Antwort zu finden.

So suchen Sie eine Hierarchie-ID

- Führen Sie den folgenden Befehl aus, um das übergeordnete Komponente zu beschreiben.
parent-asset-id Ersetzen Sie durch die ID oder externe ID des übergeordneten Assets.

```
aws iotsitewise describe-asset --asset-id parent-asset-id
```

Die Operation gibt eine Antwort zurück, die Details der Komponente enthält. Die Antwort enthält eine `assetHierarchies` Liste mit der folgenden Struktur:

```
{
  ...
  "assetHierarchies": [
    {
      "id": "String",
      "name": "String"
    }
  ],
  ...
}
```

Die Hierarchie-ID ist der `id`-Wert für eine Hierarchie in der Liste der Komponentenhierarchien.

Wenn Sie über die Hierarchie-ID verfügen, können Sie eine Komponente dieser Hierarchie zuordnen oder ihre Zuordnung zu dieser Hierarchie aufheben.

Verwenden Sie die [AssociateAssets](#) Operation, um eine untergeordnete Anlage einer übergeordneten Anlage zuzuordnen. Verwenden Sie die [DisassociateAssets](#) Operation, um eine untergeordnete Anlage von einer übergeordneten Anlage zu trennen. Geben Sie die folgenden Parameter an, die für beide Operationen identisch sind:

- `assetId`— Die ID oder externe ID der übergeordneten Anlage.
- `hierarchyId`— Die Hierarchie-ID oder externe ID im übergeordneten Asset.
- `childAssetId`— Die ID oder externe ID der untergeordneten Anlage.

Um ein Asset zuzuordnen (AWS CLI)

- Führen Sie den folgenden Befehl aus, um eine untergeordnete Komponente einer übergeordneten Komponente zuzuordnen. Ersetzen Sie *parent-asset-id* *hierarchy-id*, und *child-asset-id* durch die entsprechenden IDs:

```
aws iotsitewise associate-assets \
  --asset-id parent-asset-id \
  --hierarchy-id hierarchy-id \
  --child-asset-id child-asset-id
```

Um die Zuordnung zu einem Asset aufzuheben (AWS CLI)

- Führen Sie den folgenden Befehl aus, um die Zuordnung einer untergeordneten Komponente zu einer übergeordneten Komponente aufzuheben. Ersetzen Sie *parent-asset-id* durch die entsprechende ID, und *child-asset-id* durch die entsprechenden IDs:

```
aws iotsitewise disassociate-assets \
  --asset-id parent-asset-id \
  --hierarchy-id hierarchy-id \
  --child-asset-id child-asset-id
```

Aktualisieren Sie Ressourcen und Modelle

Sie können Ihre Anlagen, Anlagenmodelle, Komponentenmodelle und Schnittstellen aktualisieren, AWS IoT SiteWise um deren Namen und Definitionen zu ändern. Diese Aktualisierungsvorgänge sind asynchron und es dauert einige Zeit, bis sie weitergegeben werden. AWS IoT SiteWise Überprüfen Sie den Status des Assets oder Modells, bevor Sie weitere Änderungen vornehmen. Sie müssen warten, bis die Änderungen weitergegeben werden, bevor Sie die aktualisierte Komponente oder das aktualisierte Modell weiterhin verwenden können.

Themen

- [Aktualisieren Sie die Anlagen in AWS IoT SiteWise](#)
- [Aktualisieren Sie Objektmodelle, Komponentenmodelle und Schnittstellen](#)
- [Aktualisieren Sie benutzerdefinierte Verbundmodelle \(Komponenten\)](#)
- [Optimistisches Sperren für Asset-Modell-Schreibvorgänge](#)

Aktualisieren Sie die Anlagen in AWS IoT SiteWise

Sie können die AWS IoT SiteWise Konsole oder API verwenden, um den Namen einer Anlage zu aktualisieren.

Wenn Sie ein Asset aktualisieren, bleibt der Status des Assets so lange erhalten, `UPDATING` bis die Änderungen übernommen werden. Weitere Informationen finden Sie unter [Komponenten- und Modellzustände](#).

Themen

- [Aktualisieren Sie ein Asset \(Konsole\)](#)
- [Ein Asset aktualisieren \(AWS CLI\)](#)

Aktualisieren Sie ein Asset (Konsole)

Sie können die AWS IoT SiteWise Konsole verwenden, um die Asset-Details zu aktualisieren.

So aktualisieren Sie eine Komponente (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im linken Navigationsbereich Komponenten aus.
3. Wählen Sie die zu aktualisierende Komponente aus.

Tip

Sie können eine Komponentenhierarchie mithilfe des Pfeilsymbols erweitern, um nach Ihrer Komponente zu suchen.

4. Wählen Sie Bearbeiten aus.
5. Aktualisieren Sie den Eintrag für Name der Komponente.
6. (Optional) Aktualisieren Sie auf dieser Seite andere Informationen für die Komponente. Weitere Informationen finden Sie hier:
 - [Datenströme verwalten für AWS IoT SiteWise](#)
 - [Attributwerte aktualisieren](#)
 - [Interagiere mit anderen AWS Diensten](#)
7. Wählen Sie Speichern.

Ein Asset aktualisieren (AWS CLI)

Sie können das AWS Command Line Interface (AWS CLI) verwenden, um den Namen eines Assets zu aktualisieren.

Verwenden Sie die [UpdateAsset](#) Operation, um ein Asset zu aktualisieren. Geben Sie die folgenden Parameter an:

- **assetId**— Die ID des Assets. Dies ist die tatsächliche ID im UUID-Format, oder die, **externalId:myExternalId** falls sie eine hat. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.
- **assetName**— Der neue Name des Assets.

Um den Namen eines Assets zu aktualisieren (AWS CLI)

- Führen Sie den folgenden Befehl aus, um den Namen einer Komponente zu aktualisieren. **asset-id** Ersetzen Sie ihn durch die ID oder externe ID des Assets. Aktualisieren Sie das **asset-name** mit dem neuen Namen für das Asset.

```
aws iotsitewise update-asset \  
  --asset-id asset-id \  
  --asset-name asset-name
```

Aktualisieren Sie Objektmodelle, Komponentenmodelle und Schnittstellen

Verwenden Sie die AWS IoT SiteWise Konsole oder API, um ein Anlagenmodell, ein Komponentenmodell oder eine Schnittstelle zu aktualisieren.

Sie können den Typ oder den Datentyp einer vorhandenen Eigenschaft oder das Fenster einer vorhandenen Metrik nicht ändern. Sie können den Modelltyp auch nicht von einem Asset-Modell in ein Komponentenmodell oder eine Schnittstelle oder umgekehrt ändern.

Important

- Wenn Sie eine Eigenschaft aus einem Asset- oder Komponentenmodell entfernen, AWS IoT SiteWise werden alle vorherigen Daten für diese Eigenschaft gelöscht. Bei Komponentenmodellen wirkt sich dies auf alle Anlagenmodelle aus, die dieses Komponentenmodell verwenden. Achten Sie also besonders darauf, zu verstehen, wie umfassend Ihre Änderung sein kann.
- Wenn Sie eine Hierarchiedefinition aus einem Anlagenmodell entfernen, AWS IoT SiteWise wird die Zuordnung aller Anlagen in dieser Hierarchie aufgehoben.

Wenn Sie ein Komponentenmodell aktualisieren, spiegelt jede Komponente, die auf diesem Modell basiert, alle Änderungen wider, die Sie am zugrunde liegenden Modell vornehmen. Bis die Änderungen weitergegeben werden, hat jede Komponente den Status UPDATING. Sie müssen warten, bis diese Komponenten wieder in den Zustand ACTIVE zurückkehren, bevor Sie mit ihnen interagieren können. Während dieser Zeit hat das aktualisierte Komponentenmodell den Status PROPAGATING.

Wenn Sie ein Komponentenmodell aktualisieren, spiegelt jedes Anlagenmodell, das dieses Komponentenmodell enthält, die Änderungen wider. Bis die Änderungen am Komponentenmodell wirksam werden, hat jedes betroffene Asset-Modell den UPDATING Status, gefolgt von PROPAGATING der Aktualisierung der zugehörigen Assets, wie im vorherigen Absatz beschrieben. Sie müssen warten, bis diese Asset-Modelle wieder in den gleichen ACTIVE Zustand zurückkehren, bevor Sie mit ihnen interagieren. Während dieser Zeit wird der Status des aktualisierten Komponentenmodells beibehalten PROPAGATING.

Weitere Informationen finden Sie unter [Komponenten- und Modellzustände](#).

Themen

- [Aktualisierung eines Asset-Modells, Komponentenmodells oder einer Schnittstelle \(Konsole\)](#)
- [Aktualisieren Sie ein Objektmodell, ein Komponentenmodell oder eine Schnittstelle \(AWS CLI\)](#)

Aktualisierung eines Asset-Modells, Komponentenmodells oder einer Schnittstelle (Konsole)

Sie können die AWS IoT SiteWise Konsole verwenden, um ein Asset-Modell, ein Komponentenmodell oder eine Schnittstelle zu aktualisieren.

Um ein Asset-Modell, ein Komponentenmodell oder eine Schnittstelle (Konsole) zu aktualisieren

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Klicken Sie im Navigationsbereich auf Models (Modelle).
3. Wählen Sie das zu aktualisierende Asset-Modell, Komponentenmodell oder die Schnittstelle aus.
4. Wählen Sie Bearbeiten aus.
5. Führen Sie auf der Seite Modell bearbeiten einen der folgenden Schritte aus:
 - Ändern Sie unter Modelldetails die Angabe unter Name für das Modell.

- Ändern Sie eine der Attributdefinitionen. Sie können den Datentyp vorhandener Attribute nicht ändern. Weitere Informationen finden Sie unter [Definieren Sie statische Daten \(Attribute\)](#).
- Ändern Sie eine der Messungsdefinitionen. Sie können den Datentyp vorhandener Messungen nicht ändern. Weitere Informationen finden Sie unter [Definieren Sie Datenströme von Geräten \(Messungen\)](#).
- Ändern Sie eine der Transformationsdefinitionen. Weitere Informationen finden Sie unter [Daten transformieren \(transformiert\)](#).
- Ändern Sie eine der Metrikdefinitionen. Sie können das Zeitintervall vorhandener Metriken nicht ändern. Weitere Informationen finden Sie unter [Aggregieren Sie Daten aus Immobilien und anderen Vermögenswerten \(Metriken\)](#).
- (Nur Asset-Modelle) Ändern Sie eine der Hierarchiedefinitionen. Sie können das Hierarchiemodell vorhandener Hierarchien nicht ändern. Weitere Informationen finden Sie unter [Definieren Sie die Hierarchien der Anlagenmodelle](#).

6. Wählen Sie Save (Speichern) aus.

 Note

In der Konsole gestellte Aktualisierungsanforderungen werden abgelehnt, wenn ein anderer Benutzer das Asset-Modell seit dem letzten Öffnen der Seite Modell bearbeiten erfolgreich aktualisiert hat. Die Konsole fordert den Benutzer auf, die Seite Modell bearbeiten zu aktualisieren, um das aktualisierte Modell abzurufen. Sie müssen Ihre Aktualisierungen erneut vornehmen und den Speichervorgang erneut versuchen. Weitere Details finden Sie unter [Optimistisches Sperren für Asset-Modell-Schreibvorgänge](#).

Aktualisieren Sie ein Objektmodell, ein Komponentenmodell oder eine Schnittstelle (AWS CLI)

Verwenden Sie AWS Command Line Interface (AWS CLI), um ein Asset-Modell, ein Komponentenmodell oder eine Schnittstelle zu aktualisieren.

Verwenden Sie die [UpdateAssetModel](#)API, um den Namen, die Beschreibung und die Eigenschaften eines Assetmodells, Komponentenmodells oder einer Schnittstelle zu aktualisieren. Nur für Asset-Modelle können Sie Hierarchien aktualisieren. Bei Schnittstellen können Sie Eigenschaften und Hierarchien aktualisieren. Geben Sie die folgenden Parameter an:

- `assetModelId`— Die ID des Assets. Dies ist die tatsächliche ID im UUID-Format, oder die, `externalId:myExternalId` falls sie eine hat. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.

Geben Sie das aktualisierte Modell in der Payload an. Weitere Informationen zum erwarteten Format eines Asset- oder Komponentenmodells finden Sie unter [Erstellen Sie Asset-Modelle in AWS IoT SiteWise](#).

 Warning

Die [UpdateAssetModel](#) API überschreibt das vorhandene Modell mit dem Modell, das Sie in der Payload angeben. Um zu verhindern, dass die Eigenschaften oder Hierarchien Ihres Modells gelöscht werden, müssen Sie deren IDs und Definitionen in die aktualisierte Modellnutzlast aufnehmen. Informationen dazu, wie Sie die bestehende Struktur Ihres Modells abfragen können, finden Sie unter Vorgang. [DescribeAssetModel](#)

 Note

Mit dem folgenden Verfahren können nur zusammengesetzte Modelle des Typs aktualisiert AWS/ALARM werden. Wenn Sie CUSTOM zusammengesetzte Modelle aktualisieren möchten, verwenden Sie [UpdateAssetModelCompositeModel](#) stattdessen. Weitere Informationen finden Sie unter [Aktualisieren Sie benutzerdefinierte Verbundmodelle \(Komponenten\)](#).

Um ein Asset- oder Komponentenmodell zu aktualisieren (AWS CLI)

1. Führen Sie den folgenden Befehl aus, um die vorhandene Modelldefinition abzurufen. `asset-model-id` Ersetzen Sie es durch die ID oder die externe ID des Asset- oder Komponentenmodells, das aktualisiert werden soll.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

Der obige Befehl gibt die Modelldefinition zurück, die der neuesten Version des Modells entspricht.

Für einen Anwendungsfall, in dem sich ein Asset-Modell in einem FAILED Status befindet, rufen Sie die gültige Modelldefinition ab, die der aktiven Version entspricht, um Ihre

Aktualisierungsanforderung zu erstellen. Details dazu finden Sie unter [Versionen von Asset-Modellen](#). Führen Sie den folgenden Befehl aus, um die aktive Modelldefinition abzurufen:

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id --asset-model-version ACTIVE
```

Der Vorgang gibt eine Antwort zurück, die die Details des Modells enthält. Die Antwort weist die folgenden Struktur auf.

```
{
  "assetModelId": "String",
  "assetModelArn": "String",
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition,
  "assetModelCompositeModels": Array of AssetModelCompositeModel,
  "assetModelCompositeModelSummaries": Array of AssetModelCompositeModelSummary,
  "assetModelCreationDate": "String",
  "assetModelLastUpdateDate": "String",
  "assetModelStatus": {
    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  },
  "assetModelType": "String",
  "assetModelVersion": "String",
  "eTag": "String"
}
```

Weitere Informationen finden Sie unter dem Vorgang [DescribeAssetModel](#).

- Erstellen Sie eine Datei namens `update-asset-model.json` und kopieren Sie die Antwort des vorherigen Befehls in die Datei.
- Entfernen Sie die folgenden Schlüssel-Wert-Paare aus dem JSON-Objekt in `update-asset-model.json`:
 - `assetModelId`
 - `assetModelArn`

- `assetModelCompositeModelSummaries`
- `assetModelCreationDate`
- `assetModelLastUpdateDate`
- `assetModelStatus`
- `assetModelType`
- `assetModelVersion`
- `eTag`

Die [UpdateAssetModel](#) Operation erwartet eine Nutzlast mit der folgenden Struktur:

```
{
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition,
  "assetModelCompositeModels": Array of AssetModelCompositeModel
}
```

4. Führen Sie in `update-asset-model.json` eine der folgenden Aufgaben durch:
 - Ändern des Namens des Komponentenmodells (`assetModelName`).
 - Ändern, Hinzufügen oder Entfernen der Beschreibung des Komponentenmodells (`assetModelDescription`).
 - Ändern, Hinzufügen oder Entfernen der Eigenschaften des Komponentenmodells (`assetModelProperties`). Sie können den `dataType` der vorhandenen Eigenschaften oder das `window` vorhandener Metriken nicht ändern. Weitere Informationen finden Sie unter [Definieren Sie Dateneigenschaften](#).
 - Ändern, Hinzufügen oder Entfernen einer der Hierarchien des Komponentenmodells (`assetModelHierarchies`). Sie können die `childAssetModelId` von vorhandenen Hierarchien nicht ändern. Weitere Informationen finden Sie unter [Definieren Sie die Hierarchien der Anlagenmodelle](#).
 - Sie können eines der zusammengesetzten Modelle des Typs `AWS/ALARM` (`assetModelCompositeModels`) des Asset-Modells ändern, hinzufügen oder entfernen. Alarmer überwachen andere Eigenschaften, sodass Sie erkennen können, wann Geräte oder Prozesse besondere Aufmerksamkeit erfordern. Jede Alarmdefinition ist ein zusammengesetztes Modell, das die vom Alarm verwendeten Eigenschaften standardisiert.

Weitere Informationen erhalten Sie unter [Überwachen Sie Daten mit Alarmen](#) [AWS IoT SiteWise](#) und [Definieren Sie Alarme für Asset-Modelle in AWS IoT SiteWise](#).

5. Führen Sie den folgenden Befehl aus, um das Komponentenmodell mit der in `update-asset-model.json` gespeicherten Definition zu aktualisieren. `asset-model-id` Ersetzen Sie es durch die ID des Asset-Modells:

```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --cli-input-json file://model-payload.json
```

Important

Wenn mehrere Benutzer ein Asset-Modell gleichzeitig aktualisieren, können die Änderungen eines Benutzers versehentlich von einem anderen Benutzer überschrieben werden. Um dies zu verhindern, müssen Sie eine bedingte Aktualisierungsanforderung definieren. Siehe [Optimistisches Sperren für Asset-Modell-Schreibvorgänge](#).

Aktualisieren Sie benutzerdefinierte Verbundmodelle (Komponenten)

Sie können die AWS IoT SiteWise API verwenden, um ein benutzerdefiniertes Verbundmodell zu aktualisieren, oder die AWS IoT SiteWise Konsole, um Komponenten zu aktualisieren.

Themen

- [Aktualisieren Sie eine Komponente \(Konsole\)](#)
- [Aktualisieren Sie ein benutzerdefiniertes Verbundmodell \(AWS CLI\)](#)

Aktualisieren Sie eine Komponente (Konsole)

Sie können die AWS IoT SiteWise Konsole verwenden, um eine Komponente zu aktualisieren.

Um eine Komponente (Konsole) zu aktualisieren

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Klicken Sie im Navigationsbereich auf Models (Modelle).
3. Wählen Sie das Asset-Modell aus, in dem sich die Komponente befindet.

4. Wählen Sie auf der Registerkarte Eigenschaften die Option Komponenten aus.
5. Wählen Sie die Komponente aus, die Sie aktualisieren möchten.
6. Wählen Sie Bearbeiten aus.
7. Führen Sie auf der Seite Komponente bearbeiten einen der folgenden Schritte aus:
 - Ändern Sie unter Modelldetails die Angabe unter Name für das Modell.
 - Ändern Sie eine der Attributdefinitionen. Sie können den Datentyp vorhandener Attribute nicht ändern. Weitere Informationen finden Sie unter [Definieren Sie statische Daten \(Attribute\)](#).
 - Ändern Sie eine der Messungsdefinitionen. Sie können den Datentyp vorhandener Messungen nicht ändern. Weitere Informationen finden Sie unter [Definieren Sie Datenströme von Geräten \(Messungen\)](#).
 - Ändern Sie eine der Transformationsdefinitionen. Weitere Informationen finden Sie unter [Daten transformieren \(transformiert\)](#).
 - Ändern Sie eine der Metrikdefinitionen. Sie können das Zeitintervall vorhandener Metriken nicht ändern. Weitere Informationen finden Sie unter [Aggregieren Sie Daten aus Immobilien und anderen Vermögenswerten \(Metriken\)](#).
8. Wählen Sie Save (Speichern) aus.

Aktualisieren Sie ein benutzerdefiniertes Verbundmodell (AWS CLI)

Verwenden Sie AWS Command Line Interface (AWS CLI), um ein benutzerdefiniertes Verbundmodell zu aktualisieren.

Verwenden Sie die [UpdateAssetModelCompositeModel](#) Operation, um den Namen oder die Beschreibung zu aktualisieren. Nur für benutzerdefinierte Verbundwerkstoffmodelle können Sie auch die Eigenschaften aktualisieren. Sie können die Eigenschaften eines auf einem Komponentenmodell basierenden benutzerdefinierten Verbundmodells nicht aktualisieren, da das referenzierte Komponentenmodell die zugehörigen Eigenschaften bereitstellt.

Important

Wenn Sie eine Eigenschaft aus einem benutzerdefinierten Verbundmodell entfernen, werden alle vorherigen Daten für diese Eigenschaft AWS IoT SiteWise gelöscht. Sie können den Typ oder den Datentyp einer vorhandenen Eigenschaft nicht ändern.

Gehen Sie wie folgt vor, um eine vorhandene Eigenschaft eines zusammengesetzten Modells durch eine neue Eigenschaft mit derselben name zu ersetzen:

1. Reichen Sie eine `UpdateAssetModelCompositeModel` Anfrage ein, bei der die gesamte vorhandene Eigenschaft entfernt wurde.
2. Reichen Sie eine zweite `UpdateAssetModelCompositeModel` Anfrage ein, die die neue Immobilie umfasst. Die neue Objekteigenschaft hat dieselbe Eigenschaft name wie die vorherige und AWS IoT SiteWise generiert ein neues Unikatid.

Um ein benutzerdefiniertes Verbundmodell zu aktualisieren (AWS CLI)

1. Führen Sie den folgenden Befehl aus, um die bestehende Definition eines zusammengesetzten Modells abzurufen. `composite-model-id` Ersetzen Sie es durch die ID oder die externe ID des benutzerdefinierten Verbundmodells, das aktualisiert werden soll, und durch das Asset-Modell, `asset-model-id` mit dem das benutzerdefinierte Verbundmodell verknüpft ist. Weitere Informationen finden Sie im AWS IoT SiteWise-Benutzerhandbuch.

- a. Führen Sie den folgenden Befehl aus:

```
aws iotsitewise describe-asset-model-composite-model \  
--asset-model-composite-model-id composite-model-id \  
--asset-model-id asset-model-id
```

- b. Der obige Befehl gibt die Definition des zusammengesetzten Modells zurück, die der neuesten Version des zugehörigen Modells entspricht. Für einen Anwendungsfall, in dem sich ein Asset-Modell in einem FAILED Status befindet, rufen Sie die gültige Modelldefinition ab, die der aktiven Version entspricht, um Ihre Aktualisierungsanforderung zu erstellen. Details dazu finden Sie unter [Versionen von Asset-Modellen](#).
- c. Führen Sie den folgenden Befehl aus, um die aktive Modelldefinition abzurufen:

```
aws iotsitewise describe-asset-model-composite-model \  
--asset-model-composite-model-id composite-model-id \  
--asset-model-id asset-model-id \  
--asset-model-version ACTIVE
```

- d. Weitere Informationen finden Sie unter dem Vorgang [DescribeAssetModelCompositeModel](#).
2. Erstellen Sie eine Datei mit dem Namen `update-custom-composite-model.json`, und kopieren Sie dann die Antwort des vorherigen Befehls in die Datei.

3. Entfernen Sie alle Schlüssel-Wert-Paare aus dem JSON-Objekt in `update-custom-composite-model.json` mit Ausnahme der folgenden Felder:
 - `assetModelCompositeModelName`
 - `assetModelCompositeModelDescription`(falls vorhanden)
 - `assetModelCompositeModelProperties`(falls vorhanden)
4. Führen Sie in `update-custom-composite-model.json` eine der folgenden Aufgaben durch:
 - Ändern Sie den Wert von `assetModelCompositeModelName`.
 - Fügen Sie den Wert hinzu `assetModelCompositeModelDescription`, entfernen Sie ihn oder ändern Sie ihn.
 - Nur für benutzerdefinierte Inline-Verbundmodelle: Ändern, hinzufügen oder entfernen Sie alle Eigenschaften des Asset-Modells in `assetModelCompositeModelProperties`.

Weitere Informationen zum erforderlichen Format für diese Datei finden Sie in der Anforderungssyntax für [UpdateAssetModelCompositeModel](#).

5. Führen Sie den folgenden Befehl aus, um das benutzerdefinierte Verbundmodell mit der in gespeicherten Definition zu aktualisieren `update-custom-composite-model.json`.
composite-model-id Ersetzen Sie es durch die ID des zusammengesetzten Modells und *asset-model-id* durch die ID des Asset-Modells, in dem es sich befindet.

```
aws iotsitewise update-asset-model-composite-model \  
--asset-model-composite-model-id composite-model-id \  
--asset-model-id asset-model-id \  
--cli-input-json file://update-custom-composite-model.json
```

Important

Wenn mehrere Benutzer ein Asset-Modell gleichzeitig aktualisieren, können die Änderungen eines Benutzers versehentlich von einem anderen Benutzer überschrieben werden. Um dies zu verhindern, müssen Sie eine bedingte Aktualisierungsanforderung definieren. Siehe [Optimistisches Sperren für Asset-Modell-Schreibvorgänge](#).

Optimistisches Sperren für Asset-Modell-Schreibvorgänge

Beim Aktualisieren eines Asset-Modells geht ein Benutzer wie folgt vor:

1. Lesen Sie die aktuelle Definition des Anlagenmodells.
2. Bearbeiten Sie die Definition des Anlagenmodells mit den erforderlichen Änderungen.
3. Aktualisieren Sie das Asset-Modell mit der neuen Definition.

In einem Szenario, in dem zwei Benutzer ein Modell aktualisieren, ist Folgendes möglich:

- Benutzer A liest die Definition des Assetmodells X.
- Benutzer B liest die Definition des Anlagenmodells X und überträgt die Änderungen, wodurch die Definition von X geändert wird.
- Benutzer A übernimmt und überschreibt die von Benutzer B am Anlagenmodell X vorgenommene Änderung, ohne die Änderungen von Benutzer B zu überprüfen oder zu übernehmen.

Optimistisches Sperren ist ein Mechanismus, der verwendet wird, um versehentliche Überschreibungen wie im obigen Szenario zu verhindern. Optimistisches Sperren ist eine Strategie, mit der sichergestellt werden soll, dass die aktuelle Version eines Asset-Modells, das aktualisiert oder gelöscht wird, mit der aktuellen Version in AWS IoT SiteWise übereinstimmt. Dadurch wird verhindert, dass Schreibvorgänge im Asset-Modell durch versehentliche Aktualisierungen überschrieben werden.

Gehen Sie wie folgt vor, um Schreibvorgänge im Assetmodell mit optimistischem Sperren durchzuführen:

Themen

- [Ausführen von Schreibvorgängen im Asset-Modell mit optimistischer Sperre \(Konsole\)](#)
- [Ausführung von Schreibvorgängen im Asset-Modell mit optimistischer Sperre \(AWS CLI\)](#)

Ausführen von Schreibvorgängen im Asset-Modell mit optimistischer Sperre (Konsole)

Im Folgenden wird beschrieben, wie Sie Schreibvorgänge am Objektmodell mit einer optimistischen Sperre für die aktive Version des Objektmodells in der Konsole durchführen.

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).

2. Klicken Sie im Navigationsbereich auf Models (Modelle).
3. Wählen Sie das zu aktualisierende Asset- oder Komponentenmodell aus.
4. Wählen Sie Bearbeiten aus.
5. Nehmen Sie auf der Seite Modell bearbeiten Änderungen vor.
6. Wählen Sie Speichern.

 Note

Manchmal wurden zwischen dem Beginn der Bearbeitung des Modells und dem Speichern der vorgenommenen Änderungen am Modell eine oder mehrere erfolgreiche Modellaktualisierungen durchgeführt.

Um sicherzustellen, dass der Benutzer neue erfolgreiche Aktualisierungen nicht versehentlich überschreibt, wird der Schreibvorgang des Benutzers zurückgewiesen. Die Konsole deaktiviert die Schaltfläche Speichern und fordert den Benutzer auf, die Seite Modell bearbeiten zu aktualisieren. Der Benutzer muss die neue aktive Version des Modells erneut aktualisieren. Der Benutzer muss die folgenden zusätzlichen Schritte ausführen:

7. Wählen Sie Refresh aus.
8. Folgen Sie erneut den Schritten 5 und 6.

Ausführung von Schreibvorgängen im Asset-Modell mit optimistischer Sperre (AWS CLI)

Im Folgenden wird beschrieben, wie Sie Schreibvorgänge im Assetmodell mit optimistischer Sperrung durchführenAWS CLI.

1. Ruft das ETag ab, das der aktuellen Modelldefinition zugeordnet ist

ETag ist ein eindeutiges Token, das für jede neue Darstellung eines Asset-Modells generiert wird. Rufen Sie die [DescribeAssetModel](#) API auf, um die aktuelle Definition des Asset-Modells und die zugehörige Definition ETag aus der Antwort abzurufen.

Bei gleichzeitigen Aktualisierungen führen Benutzer entweder erfolgreiche Aktualisierungen (Model in ACTIVE State) oder erfolglose Updates (Model in FAILED State) durch. Um sicherzustellen, dass ein Benutzer ein erfolgreiches Update nicht versehentlich überschreibt,

müssen Sie die aktive Version des Objektmodells abrufen [Versionen von Asset-Modellen](#) und den ETag Wert abrufen.

Führen Sie den folgenden Befehl aus:

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id \
--asset-model-version ACTIVE
```

Die Antwort gibt die folgende Struktur zurück:

```
{
  "assetModelId": "String",
  "assetModelArn": "String",
  "assetModelName": "String",
  ...
  "eTag": "String"
}
```

 Note

Sie müssen die neueste Version des Asset-Modells und seiner ETag Version abrufen, um keine Aktualisierungen zu überschreiben.

2. Führen Sie die UPDATE- und DELETE-Operationen mit Schreibbedingungen durch

Die folgenden Asset-Model-APIs unterstützen optimistisches Sperren:

- [UpdateAssetModel](#)
- [DeleteAssetModel](#)
- [CreateAssetModelCompositeModel](#)
- [UpdateAssetModelCompositeModel](#)
- [DeleteAssetModelCompositeModel](#)

 Note

In den folgenden Szenarien wird die `UpdateAssetModel` API als Referenz verwendet. Die Bedingungen gelten für alle oben aufgeführten Operationen.

In den folgenden Szenarien werden die unterschiedlichen Schreibbedingungen je nach den Anforderungen an die Parallelitätssteuerung beschrieben:

- Führen Sie den folgenden Befehl aus, um erfolgreiche Updates nicht zu überschreiben. Eine neue aktive Version darf seit der zuletzt gelesenen aktiven Version nicht existieren. Ersetzen Sie durch den in der API ETag zurückgegebenen Vorgang, der beim Lesen der aktiven Version verwendet wurde.

```
aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --if-match e-tag \
  --match-for-version-type ACTIVE \
  --cli-input-json file://model-payload.json
```

- Wenn eine Modellerstellung fehlschlägt, ist für das Modell noch keine aktive Version vorhanden, da es sich in einem FAILED Status befindet. Es ist immer noch möglich, eine neue aktive Version zu überschreiben, die bereits vorhanden ist, bevor Ihre Änderungen übernommen werden. Führen Sie den folgenden Befehl aus, um eine neue aktive Version nicht zu überschreiben, wenn beim letzten Lesevorgang keine aktive Version vorhanden war.

```
aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --if-none-match "*" \
  --match-for-version-type ACTIVE \
  --cli-input-json file://model-payload.json
```

- Führen Sie den folgenden Befehl aus, um zu verhindern, dass erfolgreiche oder erfolglose Updates überschrieben werden. Dieser Befehl definiert eine Schreibbedingung, die sicherstellt, dass seit Ihrer letzten gelesenen letzten Version keine neueste Version erstellt wurde. Ersetzen Sie durch den in der API ETag zurückgegebenen Vorgang, der beim Lesen der aktiven Version verwendet wurde.

tagErsetzen Sie durch den in der API-Operation ETag zurückgegebenen Vorgang, der beim Lesen der aktiven Version verwendet wurde.

```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --if-match eTag \  
  --match-for-version-type LATEST \  
  --cli-input-json file://model-payload.json
```

Ergibt die Schreibbedingung das ErgebnisFALSE, schlägt die Schreibanforderung mit dem PreconditionFailedException fehl.

Löschen Sie Objekte und Modelle inAWS IoT SiteWise

Sie können Ihre Anlagen, Anlagenmodelle, Komponentenmodelle und Schnittstellen löschen, AWS IoT SiteWise sobald Sie mit ihnen fertig sind. Die Löschvorgänge sind asynchron und es dauert einige Zeit, bis sie übertragen werden. AWS IoT SiteWise

Themen

- [Löschen Sie Objekte inAWS IoT SiteWise](#)
- [Löschen Sie Objektmodelle, Komponentenmodelle und Schnittstellen inAWS IoT SiteWise](#)

Löschen Sie Objekte inAWS IoT SiteWise

Sie können die AWS IoT SiteWise Konsole oder API verwenden, um ein Asset zu löschen, das in Ihrer Umgebung nicht mehr benötigt wird. Durch das Löschen eines Asset-Modells werden auch alle zugehörigen Assets und Komponentenmodelle gelöscht. Es ist jedoch wichtig zu beachten, dass das Löschen eines Assets oder Modells eine permanente Aktion ist und dass alle Daten, die mit den gelöschten Ressourcen verknüpft sind, ebenfalls entfernt werden. Es wird empfohlen, vor dem Löschen von Assets oder Modellen alle Abhängigkeiten oder Integrationen zu überprüfen, die betroffen sein könnten, und sicherzustellen, dass Sie über eine Sicherungskopie aller wichtigen Daten verfügen.

Bevor Sie eine Komponente löschen können, müssen Sie zunächst die Zuordnung der ihr untergeordneten Komponenten und ihre Zuordnung zu der ihr übergeordneten Komponente aufheben. Weitere Informationen finden Sie unter [Anlagen zuordnen und deren Zuordnung](#)

[aufheben](#). Wenn Sie AWS Command Line Interface (AWS CLI) verwenden, können Sie die [ListAssociatedAssets](#) Operation verwenden, um die untergeordneten Elemente einer Anlage aufzulisten.

Wenn Sie eine Komponente löschen, ist der Status so lange DELETING, bis die Änderungen weitergegeben werden. Weitere Informationen finden Sie unter [Komponenten- und Modellzustände](#). Nachdem die Komponente gelöscht wurde, können Sie sie nicht mehr abfragen. Wenn Sie dies versuchen, gibt die API eine HTTP-404-Antwort zurück.

 Important

AWS IoT SiteWise löscht alle Eigenschaftsdaten für gelöschte Objekte.

Themen

- [Löscht ein Asset \(Konsole\)](#)
- [Löschen Sie ein Asset \(AWS CLI\)](#)

Löscht ein Asset (Konsole)

Sie können die AWS IoT SiteWise Konsole verwenden, um ein Asset zu löschen.

So löschen Sie ein Asset (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im linken Navigationsbereich Komponenten aus.
3. Wählen Sie die zu löschende Komponente aus.

 Tip

Sie können eine Komponentenhierarchie mithilfe des Pfeilsymbols erweitern, um nach Ihrer Komponente zu suchen.

4. Wenn die Komponente über Zugehörige Komponenten verfügt, löschen Sie jede Komponente. Sie können den Namen einer Komponente auswählen, um zu ihrer Seite zu navigieren, auf der Sie sie löschen können.
5. Wählen Sie auf der Seite der Komponente Löschen aus.

6. Gehen Sie im Dialogfeld „Asset löschen“ wie folgt vor:
 - a. Geben Sie **Delete** ein, um den Löschvorgang zu bestätigen.
 - b. Wählen Sie Löschen aus.

Löschen Sie ein Asset (AWS CLI)

Sie können das AWS Command Line Interface (AWS CLI) verwenden, um ein Asset zu löschen.

Verwenden Sie die [DeleteAsset](#) Operation, um ein Asset zu löschen. Geben Sie den folgenden Parameter an:

- `assetId`— Die ID des Assets. Dies ist die tatsächliche ID im UUID-Format, oder die, `externalId:myExternalId` falls sie eine hat. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.

Um ein Asset zu löschen (AWS CLI)

1. Führen Sie den folgenden Befehl aus, um die Hierarchien der Komponente aufzulisten. *asset-id* Durch die ID oder die externe ID des Assets ersetzen:

```
aws iotsitewise describe-asset --asset-id asset-id
```

Die Operation gibt eine Antwort zurück, die Details der Komponente enthält. Die Antwort enthält eine `assetHierarchies` Liste mit der folgenden Struktur:

```
{
  ...
  "assetHierarchies": [
    {
      "id": "String",
      "name": "String"
    }
  ],
  ...
}
```

Weitere Informationen finden Sie unter dem Vorgang [DescribeAsset](#).

2. Führen Sie für jede Hierarchie den folgenden Befehl aus, um die untergeordneten Komponenten der Komponente aufzulisten, die dieser Hierarchie zugeordnet sind. *asset-id* Ersetzen Sie durch die ID oder externe ID des Assets und *hierarchy-id* durch die ID oder externe ID der Hierarchie.

```
aws iotsitewise list-associated-assets \  
  --asset-id asset-id \  
  --hierarchy-id hierarchy-id
```

Weitere Informationen finden Sie unter dem Vorgang [ListAssociatedAssets](#).

3. Führen Sie den folgenden Befehl aus, um jede zugeordnete Komponente zu löschen und dann die Komponente zu löschen. *asset-id* Ersetzen Sie es durch die ID oder externe ID des Assets.

```
aws iotsitewise delete-asset --asset-id asset-id
```

Löschen Sie Objektmodelle, Komponentenmodelle und Schnittstellen in AWS IoT SiteWise

Sie können die AWS IoT SiteWise Konsole oder API verwenden, um ein Asset-Modell, ein Komponentenmodell oder eine Schnittstelle zu löschen.

Bevor Sie ein Asset-Modell löschen können, müssen Sie zunächst alle Assets löschen, die anhand des Asset-Modells erstellt wurden. Bevor Sie eine Schnittstelle löschen können, müssen Sie sie zunächst von allen Asset-Modellen trennen, die sie implementieren.

Wenn Sie ein Asset-Modell oder eine Schnittstelle löschen, bleibt ihr Status so lange erhalten, DELETING bis die Änderungen übernommen werden. Weitere Informationen finden Sie unter [Komponenten- und Modellzustände](#). Nachdem das Asset-Modell oder die Schnittstelle gelöscht wurde, können Sie dieses Asset-Modell oder diese Schnittstelle nicht mehr abfragen. Wenn Sie dies versuchen, gibt die API eine HTTP-404-Antwort zurück.

Themen

- [Löschen Sie ein Asset-Modell, ein Komponentenmodell oder eine Schnittstelle \(Konsole\)](#)
- [Löschen Sie ein Objektmodell, ein Komponentenmodell oder eine Schnittstelle \(AWS CLI\)](#)

Löschen Sie ein Asset-Modell, ein Komponentenmodell oder eine Schnittstelle (Konsole)

Sie können die AWS IoT SiteWise Konsole verwenden, um ein Asset-Modell, ein Komponentenmodell oder eine Schnittstelle zu löschen.

Um ein Assetmodell, ein Komponentenmodell oder eine Schnittstelle (Konsole) zu löschen

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Klicken Sie im Navigationsbereich auf Models (Modelle).
3. Wählen Sie das zu löschende Asset-Modell, das Komponentenmodell oder die Schnittstelle aus.
4. Wenn Sie ein Asset-Modell löschen und es über Assets verfügt, löschen Sie jedes Asset. Wählen Sie den Namen einer Komponente aus, um zu ihrer Seite zu navigieren, auf der Sie sie löschen können. Weitere Informationen finden Sie unter [Löscht ein Asset \(Konsole\)](#).
5. Wählen Sie auf der Seite des Modells die Option Löschen aus.
6. Gehen Sie im Dialogfeld Modell löschen wie folgt vor:
 - a. Geben Sie **Delete** ein, um den Löschvorgang zu bestätigen.
 - b. Wählen Sie Löschen aus.

Löschen Sie ein Objektmodell, ein Komponentenmodell oder eine Schnittstelle (AWS CLI)

Sie können die AWS Command Line Interface (AWS CLI) verwenden, um ein Asset-Modell, ein Komponentenmodell oder eine Schnittstelle zu löschen.

Verwenden Sie die [DeleteAssetModel](#)Operation, um ein Asset-Modell, ein Komponentenmodell oder eine Schnittstelle zu löschen. Geben Sie den folgenden Parameter an:

- `assetModelId`— Die ID des Assets. Dies ist die tatsächliche ID im UUID-Format, oder die, `externalId:myExternalId` falls sie eine hat. Weitere Informationen finden Sie unter [Referenzobjekte mit externen IDs](#) im AWS IoT SiteWise-Benutzerhandbuch.

Um ein Asset-Modell zu löschen (AWS CLI)

1. Führen Sie den folgenden Befehl aus, um alle Komponenten aufzulisten, die aus dem Modell erstellt wurden. `asset-model-id` Ersetzen Sie es durch die ID oder die externe ID des Asset-Modells.

```
aws iotsitewise list-assets --asset-model-id asset-model-id
```

Weitere Informationen finden Sie unter dem Vorgang [ListAssets](#).

2. Wenn der vorherige Befehl Komponenten aus dem Modell zurückgibt, löschen Sie jede Komponente. Weitere Informationen finden Sie unter [Löschen Sie ein Asset \(AWS CLI\)](#).
3. Führen Sie den folgenden Befehl zum Löschen des Komponentenmodells aus. `asset-model-id` Ersetzen Sie es durch die ID oder externe ID des Asset-Modells.

```
aws iotsitewise delete-asset-model --asset-model-id asset-model-id
```

Important

Um zu verhindern, dass ein Asset-Modell gelöscht wird, das seit dem letzten Lesevorgang gleichzeitig aktualisiert wurde, müssen Sie eine bedingte Löschanforderung definieren. Siehe [Optimistisches Sperren für Asset-Modell-Schreibvorgänge](#).

Massenoperationen mit Anlagen und Modellen

Wenn Sie mit einer großen Anzahl von Objekten oder Anlagenmodellen arbeiten möchten, verwenden Sie Massenoperationen, um Ressourcen massenweise zu importieren und an einen anderen Speicherort zu exportieren. Sie können beispielsweise eine Datendatei erstellen, die Assets oder Asset-Modelle in einem Amazon S3 S3-Bucket definiert, und diese mithilfe des Massenimports erstellen oder aktualisieren AWS IoT SiteWise. Wenn Sie über eine große Anzahl von Assets oder Asset-Modellen verfügen AWS IoT SiteWise, können Sie diese alternativ nach Amazon S3 exportieren.

 Note

Sie führen Massenoperationen durch, AWS IoT SiteWise indem Sie Operationen in der AWS IoT TwinMaker API aufrufen. Sie können dies tun, ohne einen AWS IoT TwinMaker Workspace einzurichten AWS IoT TwinMaker oder zu erstellen. Sie benötigen lediglich einen Amazon S3 S3-Bucket, in dem Sie Ihre AWS IoT SiteWise Inhalte platzieren können.

Themen

- [Wichtige Konzepte und Terminologie](#)
- [Unterstützte Funktionen](#)
- [Voraussetzungen für den Massenvorgang](#)
- [Führen Sie einen Massenimportauftrag aus](#)
- [Führen Sie einen Massenexportauftrag aus](#)
- [Verfolgung des Auftragsfortschritts und Fehlerbehandlung](#)
- [Beispiele für den Import von Metadaten](#)
- [Beispiele für den Export von Metadaten](#)
- [AWS IoT SiteWiseAuftragsschema für die Übertragung von Metadaten](#)

Wichtige Konzepte und Terminologie

AWS IoT SiteWiseFunktionen für Massenimport und -export basieren auf den folgenden Konzepten und Begriffen:

- Import: Die Aktion des Verschiebens von Assets oder Asset-Modellen aus einer Datei in einem Amazon S3 S3-Bucket inAWS IoT SiteWise.
- Export: Die Aktion, bei der Assets oder Asset-Modelle aus AWS IoT SiteWise einem Amazon S3 S3-Bucket verschoben werden.
- Quelle: Der Startort, von dem Sie Inhalte verschieben möchten.

Ein Amazon S3 S3-Bucket ist beispielsweise eine Importquelle und AWS IoT SiteWise eine Exportquelle.

- Ziel: Der gewünschte Ort, an den Sie Ihre Inhalte verschieben möchten.

Ein Amazon S3 S3-Bucket ist beispielsweise ein Exportziel und AWS IoT SiteWise ein Importziel.

- **AWS IoT SiteWiseSchema:** Dieses Schema wird verwendet, um Metadaten von zu importieren und zu exportierenAWS IoT SiteWise.
- **Top-level Ressource:** Eine AWS IoT SiteWise Ressource, die Sie individuell erstellen oder aktualisieren können, z. B. ein Asset oder ein Asset-Modell.
- **Sub-resource:** Eine verschachtelte AWS IoT SiteWise Ressource innerhalb einer Ressource der obersten Ebene. Beispiele hierfür sind Eigenschaften, Hierarchien und zusammengesetzte Modelle.
- **Metadaten:** Wichtige Informationen, die für den erfolgreichen Import oder Export von Ressourcen erforderlich sind. Beispiele für Metadaten sind Definitionen von Vermögenswerten und Asset-Modellen.
- **Metadaten TransferJob:** Das Objekt, das beim Ausführen erstellt wurdeCreateMetadataTransferJob.

Unterstützte Funktionen

In diesem Thema wird erklärt, was Sie tun können, wenn Sie einen Massenvorgang ausführen. Massenvorgänge unterstützen die folgenden Funktionen:

- **Top-level Ressourcenerstellung:** Wenn Sie ein Asset oder ein Asset-Modell importieren, das keine ID definiert oder dessen ID nicht mit der einer vorhandenen ID übereinstimmt, wird es als neue Ressource erstellt.
- **Top-level Ressourcenersatz:** Wenn Sie ein Asset oder ein Asset-Modell importieren, dessen ID mit einer bereits vorhandenen übereinstimmt, ersetzt es die vorhandene Ressource.
- **Erstellen, Ersetzen oder Löschen von Unterressourcen:** Wenn Ihr Import eine Ressource der obersten Ebene ersetzt, z. B. eine Anlage oder ein Anlagenmodell, ersetzt die neue Definition alle Unterressourcen wie Eigenschaften, Hierarchien oder zusammengesetzte Modelle.

Wenn Sie beispielsweise ein Asset-Modell während eines Massenimports aktualisieren und die aktualisierte Version eine Eigenschaft definiert, die im Original nicht vorhanden war, wird eine neue Eigenschaft erstellt. Wenn sie eine Eigenschaft definiert, die bereits vorhanden ist, wird die vorhandene Eigenschaft aktualisiert. Wenn das aktualisierte Objektmodell eine Eigenschaft auslöst, die im Original vorhanden war, wird die Eigenschaft gelöscht.

- **Kein Löschen von Ressourcen auf oberster Ebene:** Bei Massenvorgängen wird kein Asset oder Asset-Modell gelöscht. Bei Massenvorgängen werden sie nur erstellt oder aktualisiert.

Voraussetzungen für den Massenvorgang

In diesem Abschnitt werden die Voraussetzungen für Massenoperationen erläutert, einschließlich AWS Identity and Access Management (IAM-) Berechtigungen für den Austausch von Ressourcen zwischen AWS Diensten und Ihrem lokalen Computer. Bevor Sie einen Massenvorgang starten, müssen Sie die folgenden Voraussetzungen erfüllen:

- Erstellen Sie einen Amazon S3 S3-Bucket zum Speichern von Ressourcen. Weitere Informationen zur Verwendung von Amazon S3 finden Sie unter [Was ist Amazon S3?](#)

IAM-Berechtigungen

Um Massenoperationen durchzuführen, müssen Sie eine AWS Identity and Access Management (IAM-) Richtlinie mit Berechtigungen erstellen, die den Austausch von AWS Ressourcen zwischen Amazon S3 und Ihrem lokalen Computer ermöglichen. AWS IoT SiteWise Weitere Informationen zum Erstellen von benutzerdefinierten Richtlinien finden Sie unter [IAM-Richtlinien erstellen](#).

Um Massenoperationen durchzuführen, benötigen Sie die folgenden Richtlinien.

AWS IoT SiteWisePolicy

Diese Richtlinie ermöglicht den Zugriff auf die erforderlichen AWS IoT SiteWise API-Aktionen für Massenoperationen:

```
{
  "Sid": "SiteWiseApiAccess",
  "Effect": "Allow",
  "Action": [
    "iotsitewise:CreateAsset",
    "iotsitewise:CreateAssetModel",
    "iotsitewise:UpdateAsset",
    "iotsitewise:UpdateAssetModel",
    "iotsitewise:UpdateAssetProperty",
    "iotsitewise:ListAssets",
    "iotsitewise:ListAssetModels",
    "iotsitewise:ListAssetProperties",
    "iotsitewise:ListAssetModelProperties",
    "iotsitewise:ListAssociatedAssets",
    "iotsitewise:DescribeAsset",
    "iotsitewise:DescribeAssetModel",
    "iotsitewise:DescribeAssetProperty",
```

```

    "iotsitewise:AssociateAssets",
    "iotsitewise:DisassociateAssets",
    "iotsitewise:AssociateTimeSeriesToAssetProperty",
    "iotsitewise:DisassociateTimeSeriesFromAssetProperty",
    "iotsitewise:BatchPutAssetPropertyValue",
    "iotsitewise:BatchGetAssetPropertyValue",
    "iotsitewise:TagResource",
    "iotsitewise:UntagResource",
    "iotsitewise:ListTagsForResource",
    "iotsitewise:CreateAssetModelCompositeModel",
    "iotsitewise:UpdateAssetModelCompositeModel",
    "iotsitewise:DescribeAssetModelCompositeModel",
    "iotsitewise>DeleteAssetModelCompositeModel",
    "iotsitewise:ListAssetModelCompositeModels",
    "iotsitewise:ListCompositionRelationships",
    "iotsitewise:DescribeAssetCompositeModel"
  ],
  "Resource": "*"
}
```

AWS IoT TwinMakerpolicy

Diese Richtlinie ermöglicht den Zugriff auf die AWS IoT TwinMaker API-Operationen, die Sie für die Arbeit mit Massenoperationen verwenden:

```

{
  "Sid": "MetadataTransferJobApiAccess",
  "Effect": "Allow",
  "Action": [
    "iottwinmaker:CreateMetadataTransferJob",
    "iottwinmaker:CancelMetadataTransferJob",
    "iottwinmaker:GetMetadataTransferJob",
    "iottwinmaker:ListMetadataTransferJobs"
  ],
  "Resource": "*"
}
```

Amazon S3 S3-Richtlinie

Diese Richtlinie bietet Zugriff auf Amazon S3 S3-Buckets für die Übertragung von Metadaten für Massenoperationen.

For a specific Amazon S3 bucket

Wenn Sie einen bestimmten Bucket für die Arbeit mit Ihren Metadaten für Massenoperationen verwenden, bietet diese Richtlinie Zugriff auf diesen Bucket:

```
{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject",
    "s3:GetObject",
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:AbortMultipartUpload",
    "s3:ListBucketMultipartUploads",
    "s3:ListMultipartUploadParts"
  ],
  "Resource": [
    "arn:aws:s3:::bucket name",
    "arn:aws:s3:::bucket name/*"
  ]
}
```

To allow any Amazon S3 bucket

Wenn Sie viele verschiedene Buckets verwenden, um mit Ihren Metadaten für Massenoperationen zu arbeiten, bietet diese Richtlinie Zugriff auf jeden beliebigen Bucket:

```
{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject",
    "s3:GetObject",
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:AbortMultipartUpload",
    "s3:ListBucketMultipartUploads",
    "s3:ListMultipartUploadParts"
  ],
  "Resource": "*"
}
```

Informationen zur Fehlerbehebung bei Import- und Exportvorgängen finden Sie unter [Problembehandlung beim Massenimport und -export](#).

Führen Sie einen Massenimportauftrag aus

Beim Massenimport werden Metadaten in einen AWS IoT SiteWise Workspace verschoben. Durch den Massenimport können beispielsweise Metadaten aus einer lokalen Datei oder einer Datei in einem Amazon S3 S3-Bucket in einen AWS IoT SiteWise Workspace verschoben werden.

Schritt 1: Bereiten Sie die Datei für den Import vor

Laden Sie die Datei im AWS IoT SiteWise nativen Format herunter, um Assets und Asset-Modelle zu importieren. Weitere Details finden Sie unter [AWS IoT SiteWiseAuftragsschema für die Übertragung von Metadaten](#).

Schritt 2: Laden Sie die vorbereitete Datei auf Amazon S3 hoch

Laden Sie die Datei auf Amazon S3 hoch. Weitere Informationen finden Sie unter [Hochladen einer Datei auf Amazon S3](#) im Amazon Simple Storage Service-Benutzerhandbuch.

Metadaten importieren (Konsole)

Sie können den verwendenAWS-IoT-SiteWise-Konsole, um Metadaten massenweise zu importieren. Folgen Sie [Schritt 1: Bereiten Sie die Datei für den Import vor](#) und bereiten [Schritt 2: Laden Sie die vorbereitete Datei auf Amazon S3 hoch](#) Sie eine Datei vor, die für den Import bereit ist.

Daten von Amazon S3 importieren nachAWS-IoT-SiteWise-Konsole

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im Navigationsbereich die Option Massenoperationen Neu aus.
3. Wählen Sie Neuer Import, um den Importvorgang zu starten.
4. Gehen Sie auf der Seite Metadaten importieren wie folgt vor:
 - Wählen Sie Amazon S3 durchsuchen, um den Amazon S3 S3-Bucket und die Dateien anzuzeigen.
 - Navigieren Sie zu dem Amazon S3 S3-Bucket, der die vorbereitete Importdatei enthält.
 - Wählen Sie die zu importierende Datei aus.
 - Überprüfen Sie die ausgewählte Datei und wählen Sie „Importieren“.

5. Auf der Seite „Massenvorgänge für SiteWise Metadaten“ von AWS-IoT-SiteWise-Konsole wird der neu erstellte Importauftrag in der Fortschrittsstabelle der Jobs angezeigt.

Metadaten importieren (AWS CLI)

Gehen Sie wie folgt vor, um eine Importaktion durchzuführen:

Daten von Amazon S3 importieren nach AWS CLI

1. Erstellen Sie eine Metadaten-Datei, die die Ressourcen angibt, die Sie importieren möchten, und folgen Sie dabei dem [AWS IoT SiteWise Auftragsschema für die Übertragung von Metadaten](#). Speichern Sie diese Datei in Ihrem Amazon S3 S3-Bucket.

Beispiele für zu importierende Metadaten-Dateien finden Sie unter [Beispiele für den Import von Metadaten](#).

2. Erstellen Sie nun eine JSON-Datei mit dem Hauptteil der Anfrage. Der Anforderungstext gibt die Quelle und das Ziel für den Übertragungsjob an. Diese Datei ist von der Datei aus dem vorherigen Schritt getrennt. Stellen Sie sicher, dass Sie Ihren Amazon S3 S3-Bucket als Quelle und `iotsitewise` als Ziel angeben.

Das folgende Beispiel zeigt den Hauptteil der Anfrage:

```
{
  "metadataTransferJobId": "your-transfer-job-Id",
  "sources": [{
    "type": "s3",
    "s3Configuration": {
      "location": "arn:aws:s3:::amzn-s3-demo-bucket/"
    }
  }],
  "destination": {
    "type": "iotSiteWise"
  }
}
```

3. Rufen Sie den auf, `CreateMetadataTransferJob` indem Sie den folgenden AWS CLI Befehl ausführen. In diesem Beispiel wird die Anforderungstextdatei aus dem vorherigen Schritt benannt `createMetadataTransferJobExport.json`.

```
aws iottwinmaker create-metadata-transfer-job --region us-east-1 \  
--cli-input-json file://createMetadataTransferJobImport.json
```

Dadurch wird ein Auftrag zur Übertragung von Metadaten erstellt und der Prozess der Übertragung der ausgewählten Ressourcen gestartet.

Führen Sie einen Massenexportauftrag aus

Beim Massenexport werden Metadaten von einem AWS IoT SiteWise Workspace in einen Amazon S3 S3-Bucket verschoben.

Wenn Sie einen Massenexport Ihrer AWS IoT SiteWise Inhalte nach Amazon S3 durchführen, können Sie Filter angeben, um einzuschränken, welche spezifischen Asset-Modelle und Assets Sie exportieren möchten.

Die Filter müssen in einem `iotSiteWiseConfiguration` Abschnitt im Quellenbereich Ihrer JSON-Anfrage angegeben werden.

Note

Sie können mehrere Filter in Ihre Anfrage aufnehmen. Bei der Massenoperation werden Asset-Modelle und Assets exportiert, die einem der Filter entsprechen.

Wenn Sie keine Filter angeben, exportiert der Massenvorgang alle Ihre Asset-Modelle und Assets.

Example Hauptteil mit Filtern anfordern

```
{  
  "metadataTransferJobId": "your-transfer-job-id",  
  "sources": [  
    {  
      "type": "iotsitewise",  
      "iotSiteWiseConfiguration": {  
        "filters": [  
          {  
            "filterByAssetModel": {  
              "assetModelId": "asset model ID"  
            }  
          }  
        ]  
      }  
    }  
  ]  
}
```

```

    }
  },
  {
    "filterByAssetModel": {
      "assetModelId": "asset model ID",
      "includeAssets": true
    }
  },
  {
    "filterByAssetModel": {
      "assetModelId": "asset model ID",
      "includeOffspring": true
    }
  }
]
}
},
"destination": {
  "type": "s3",
  "s3Configuration": {
    "location": "arn:aws:s3:::amzn-s3-demo-bucket"
  }
}
}
}

```

Metadaten exportieren (Konsole)

Das folgende Verfahren erklärt die Exportaktion der Konsole:

Erstellen Sie einen Exportauftrag im AWS-IoT-SiteWise-Konsole

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im Navigationsbereich die Option Massenoperationen Neu aus.
3. Wählen Sie Neuer Export, um den Exportvorgang zu starten.
4. Gehen Sie auf der Seite Metadaten exportieren wie folgt vor:
 - Geben Sie einen Namen für den Exportjob ein. Dies ist der Name, der für die exportierte Datei in Ihrem Amazon S3 S3-Bucket verwendet wird.
 - Wählen Sie Ihre zu exportierenden Ressourcen aus, wodurch die Filter für den Job festgelegt werden:

- Exportieren Sie alle Assets und Asset-Modelle. Verwenden Sie Filter für Assets und Asset-Modelle.
 - Exportieren Sie Vermögenswerte. Filtern Sie nach Ihren Vermögenswerten.
 - Wählen Sie das Asset aus, das für den Exportfilter verwendet werden soll.
 - (Optional) Fügen Sie den Nachwuchs oder das zugehörige Asset-Modell hinzu.
 - Exportieren Sie Asset-Modelle. Filtern Sie nach Ihren Asset-Modellen.
 - Wählen Sie das Asset-Modell aus, das für den Exportfilter verwendet werden soll.
 - (Optional) Fügen Sie den Nachwuchs oder das zugehörige Asset oder beides hinzu.
 - Wählen Sie Weiter aus.
 - Navigieren Sie zum Amazon S3 S3-Bucket:
 - Wählen Sie Amazon S3 durchsuchen, um den Amazon S3 S3-Bucket und die Dateien anzuzeigen.
 - Navigieren Sie zu dem Amazon S3 S3-Bucket, in dem die Datei platziert werden muss.
 - Wählen Sie Weiter aus.
 - Überprüfen Sie den Exportauftrag und wählen Sie Exportieren.
5. Auf der Seite „Massenoperationen für SiteWise Metadaten“ von AWS-IoT-SiteWise-Konsole wird der neu erstellte Importauftrag in der Fortschrittsstabelle Jobs angezeigt.

Informationen zu den verschiedenen Möglichkeiten, Filter beim Exportieren von Metadaten zu verwenden, finden Sie unter [Beispiele für den Export von Metadaten](#).

Metadaten exportieren (AWS CLI)

Das folgende Verfahren erklärt die AWS CLI Exportaktion:

Daten exportieren von AWS IoT SiteWise zu Amazon S3

1. Erstellen Sie eine JSON-Datei mit Ihrem Anfragetext. Der Anforderungstext gibt die Quelle und das Ziel für den Übertragungsjob an. Das folgende Beispiel zeigt ein Beispiel für einen Anforderungstext:

```
{
  "metadataTransferJobId": "your-transfer-job-Id",
  "sources": [{
    "type": "iotsitewise"
```

```
    }],  
    "destination": {  
      "type": "s3",  
      "s3Configuration": {  
        "location": "arn:aws:s3:::amzn-s3-demo-bucket"  
      }  
    }  
  }  
}
```

Stellen Sie sicher, dass Sie Ihren Amazon S3 S3-Bucket als Ziel des Metadatentransferjobs angeben.

 Note

In diesem Beispiel werden alle Ihre Asset-Modelle und Assets exportiert. Um den Export auf bestimmte Asset-Modelle oder Assets zu beschränken, können Sie Filter in Ihren Anfragetext aufnehmen. Weitere Informationen zum Anwenden von Exportfiltern finden Sie unter [Beispiele für den Export von Metadaten](#).

2. Speichern Sie Ihre Anfragetextdatei, um sie im nächsten Schritt zu verwenden. In diesem Beispiel heißt die Datei `createMetadataTransferJobExport.json`.
3. Rufen Sie die auf, `CreateMetadataTransferJob` indem Sie den folgenden AWS CLI Befehl ausführen:

```
aws iottwinmaker create-metadata-transfer-job --region us-east-1 \  
--cli-input-json file://createMetadataTransferJobExport.json
```

Ersetzen Sie die JSON-Eingabedatei `createMetadataTransferJobExport.json` durch Ihren eigenen Namen der Übertragungsdatei.

Verfolgung des Auftragsfortschritts und Fehlerbehandlung

Die Verarbeitung eines Massenverarbeitungsauftrags nimmt Zeit in Anspruch. Jeder Auftrag wird in der Reihenfolge des AWS IoT SiteWise Eingangs der Anfrage verarbeitet. Sie wird für jedes Konto einzeln bearbeitet. Wenn ein Job abgeschlossen ist, beginnt der nächste in der Warteschlange automatisch mit der Verarbeitung. AWS IoT SiteWise löst die Jobs asynchron auf und aktualisiert den

Status der einzelnen Jobs im Laufe der Bearbeitung. Jeder Auftrag hat ein Statusfeld, das den Status der Ressource und gegebenenfalls eine Fehlermeldung enthält.

Der Status kann einer der folgenden Werte sein:

- **VALIDATING**— Validierung des Jobs einschließlich des übermittelten Dateiformats und seines Inhalts.
- **PENDING**— Der Job befindet sich in einer Warteschlange. Sie können Jobs in diesem Status von der AWS IoT SiteWise Konsole aus stornieren, aber alle anderen Status bleiben bis zum Ende bestehen.
- **RUNNING**— Der Job wird bearbeitet. Es erstellt und aktualisiert Ressourcen, wie in der Importdatei definiert, oder exportiert Ressourcen auf der Grundlage der ausgewählten Exportjobfilter. Wenn der Vorgang abgebrochen wird, werden alle durch diesen Job importierten Ressourcen nicht gelöscht. Weitere Informationen finden Sie unter [Überprüfen Sie den Auftragsfortschritt und die Details \(Konsole\)](#).
- **CANCELLING**— Der Job wird aktiv storniert.
- **ERROR**— Eine oder mehrere Ressourcen konnten nicht verarbeitet werden. Weitere Informationen finden Sie im ausführlichen Auftragsbericht. Weitere Informationen finden Sie unter [Überprüfen Sie die Fehlerdetails \(Konsole\)](#).
- **COMPLETED**— Der Job wurde ohne Fehler abgeschlossen.
- **CANCELLED**— Der Job wurde abgebrochen und befindet sich nicht in der Warteschlange. Wenn Sie einen **RUNNING** Job storniert haben, werden Ressourcen, die zum Zeitpunkt des Abbruchs bereits von diesem Job importiert wurden, nicht gelöscht. AWS IoT SiteWise

Themen

- [Verfolgung des Fortschritts von Aufträgen](#)
- [Untersuchen Sie die Fehler auf AWS IoT SiteWise](#)

Verfolgung des Fortschritts von Aufträgen

Überprüfen Sie den Auftragsfortschritt und die Details (Konsole)

Sehen Sie [Metadaten exportieren \(Konsole\)](#) sich [Metadaten importieren \(Konsole\)](#) oder an, um einen Sammelauftrag zu starten.

Übersicht über den Arbeitsfortschritt in der AWS IoT SiteWise-Konsole:

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im Navigationsbereich die Option Bulk operations New aus.
3. In der Fortschrittstabelle „Aufträge“ in der AWS IoT SiteWise Konsole wird die Liste der Aufträge für Massenvorgänge angezeigt.
4. In der Spalte Jobtyp wird beschrieben, ob es sich um einen Export- oder Importjob handelt. In den Spalten Importdatum wird das Datum angezeigt, an dem der Job gestartet wurde.
5. In der Spalte Status wird der Status des Jobs angezeigt. Sie können einen Job auswählen, um Details zu dem Job zu sehen.
6. Für den ausgewählten Job wird Erfolg angezeigt, wenn er erfolgreich war, oder eine Liste mit Fehlern, wenn der Job fehlgeschlagen ist. Außerdem wird für jeden Ressourcentyp eine Fehlerbeschreibung angezeigt.

Übersicht der Jobdetails in der AWS IoT SiteWise-Konsole:

In der Fortschrittstabelle für Jobs in der AWS IoT SiteWise Konsole wird die Liste der Jobs für Massenvorgänge angezeigt.

1. Wählen Sie einen Job aus, um weitere Details zu sehen.
2. Bei einem Importjob `Data source ARN` steht der für den Amazon S3 S3-Speicherort der Importdatei.
3. Bei einem Exportauftrag `Data destination ARN` steht der für den Amazon S3 S3-Speicherort der Datei nach dem Export.
4. Das `Status` und `Status reason`, geben zusätzliche Details zum aktuellen Job an. Weitere Details finden Sie unter [Verfolgung des Auftragsfortschritts und Fehlerbehandlung](#).
5. Das `Queued position` steht für die Position des Auftrags in der Prozesswarteschlange. Die Jobs werden nacheinander verarbeitet. Eine Position von 1 in der Warteschlange bedeutet, dass der Job als Nächstes verarbeitet wird.
6. Auf der Seite mit den Auftragsdetails werden auch die Anzahl der Auftragsfortschritte angezeigt.
 - Es gibt folgende Typen für die Zählung des Auftragsfortschritts:
 - i. `Total resources`— Gibt die Gesamtzahl der Anlagen an, die sich im Übertragungsprozess befinden.

- ii. **Succeeded**— Gibt die Anzahl der Vermögenswerte an, die während des Prozesses erfolgreich übertragen wurden.
 - iii. **Failed**— Gibt die Anzahl der Anlagen an, die während des Vorgangs ausgefallen sind.
 - iv. **Skipped**— Gibt die Anzahl der Assets an, die während des Vorgangs übersprungen wurden.
7. Ein Auftragsstatus von **PENDING** oder zeigt **anVALIDATING**, dass der gesamte Auftragsfortschritt als – gezählt wird. Dies weist darauf hin, dass die Fortschrittszahlen der Jobs ausgewertet werden.
8. Ein Auftragsstatus von **RUNNING** zeigt die **Total resources** Anzahl an, d. h. den Job, der zur Verarbeitung weitergeleitet wurde. Die detaillierten Zählungen (**SucceededFailed**, und **Skipped**) beziehen sich auf die verarbeiteten Ressourcen. Die Summe der detaillierten Zählungen ist kleiner als die **Total resources** Anzahl, bis der Status des Jobs **COMPLETED** oder lautet **ERROR**.
9. Wenn der Status eines Jobs **COMPLETED** oder lautet **ERROR**, entspricht die **Total resources** Anzahl der Summe der detaillierten Anzahlen (**SucceededFailed**, und **Skipped**).
10. Wenn der Status eines Job lautet **ERROR**, finden Sie in der Tabelle Auftragsfehler Einzelheiten zu den spezifischen Fehlern und Ausfällen. Weitere Details finden Sie unter [Überprüfen Sie die Fehlerdetails \(Konsole\)](#).

Überprüfen Sie den Auftragsfortschritt und die Einzelheiten (AWS CLI)

Nachdem Sie einen Massenvorgang gestartet haben, können Sie seinen Status mithilfe der folgenden API-Aktionen überprüfen oder aktualisieren:

- Verwenden Sie die [GetMetadataTransferJob](#) API-Aktion, um Informationen zu einem bestimmten Job abzurufen.

Rufen Sie Informationen mit der **GetMetadataTransferJob** API ab:

1. Erstellen Sie einen Übertragungsauftrag und führen Sie ihn aus. Rufen Sie die **GetMetadataTransferJob**-API auf.

Example AWS CLIBefehl:

```
aws iottwinmaker get-metadata-transfer-job \  
    --metadata-transfer-job-id your_metadata_transfer_job_id \  
    --output-type text
```

```
--region your_region
```

2. Die `GetMetadataTransferJob` API gibt ein `MetadataTransferJobProgress` Objekt mit den folgenden Parametern zurück:

- `succeededCount` — Gibt die Anzahl der Assets an, die im Prozess erfolgreich übertragen wurden.
- `FailedCount` — Gibt die Anzahl der Assets an, die während des Vorgangs ausgefallen sind.
- `skippedCount` — Gibt die Anzahl der Assets an, die während des Vorgangs übersprungen wurden.
- `TotalCount` — Gibt die Gesamtzahl der Vermögenswerte an, die sich im Übertragungsprozess befinden.

Diese Parameter geben den Status des Auftragsfortschritts an. Wenn der Status lautet `RUNNING`, helfen sie dabei, die Anzahl der Ressourcen nachzuverfolgen, die noch verarbeitet werden müssen.

Wenn bei der Schemavalidierung Fehler auftreten oder wenn `FailedCount` größer oder gleich 1 ist, wechselt der Status des Jobs zu `ERROR`. Ein vollständiger Fehlerbericht für den Job wird in Ihrem Amazon S3 S3-Bucket abgelegt. Weitere Details finden Sie unter [Untersuchen Sie die Fehler auf AWS IoT SiteWise](#).

- Verwenden Sie die [ListMetadataTransferJobs](#) API-Aktion, um aktuelle Jobs aufzulisten.

Verwenden Sie eine JSON-Datei, um die zurückgegebenen Jobs nach ihrem aktuellen Status zu filtern. Sehen Sie sich das folgende Verfahren an:

1. Um die Filter anzugeben, die Sie verwenden möchten, erstellen Sie eine AWS CLI JSON-Eingabedatei. Sie möchten Folgendes verwenden:

```
{
  "sourceType": "s3",
  "destinationType": "iottwinmaker",
  "filters": [{
    "state": "COMPLETED"
  }]
}
```

Eine Liste der gültigen `state` Werte finden Sie [ListMetadataTransferJobsFilter](#) im AWS IoT TwinMaker API-Referenzhandbuch.

2. Verwenden Sie die JSON-Datei als Argument im folgenden AWS CLI Beispielbefehl:

```
aws iottwinmaker list-metadata-transfer-job --region your_region \  
--cli-input-json file://ListMetadataTransferJobsExample.json
```

- Verwenden Sie die [CancelMetadataTransferJob](#) API-Aktion, um einen Job abzuberechnen. Diese API storniert den spezifischen Metadatentransferauftrag, ohne dass sich dies auf bereits exportierte oder importierte Ressourcen auswirkt:

```
aws iottwinmaker cancel-metadata-transfer-job \  
--region your_region \  
--metadata-transfer-job-id job-to-cancel-id
```

Untersuchen Sie die Fehler auf AWS IoT SiteWise

Überprüfen Sie die Fehlerdetails (Konsole)

Fehlerdetails in der AWS IoT SiteWise Konsole:

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Eine Liste der Aufträge AWS-IoT-SiteWise-Konsole für Massenvorgänge finden Sie in der Tabelle mit dem Auftragsfortschritt unter.
3. Wählen Sie einen Auftrag aus, um die Auftragsdetails anzuzeigen.
4. Wenn der Status eines Jobs COMPLETED oder lautet ERROR, entspricht die `Total resources` Anzahl der Summe der detaillierten Anzahlen (`SucceededFailed`, und `Skipped`).
5. Wenn der Status eines Job lautet ERROR, finden Sie in der Tabelle Auftragsfehler Einzelheiten zu den spezifischen Fehlern und Ausfällen.
6. In der Tabelle Auftragsfehler wird der Inhalt des Jobberichts angezeigt. Das `Resource type` Feld gibt den Ort des Fehlers oder der Ausfälle an, z. B. im Folgenden:
 - Ein Validierungsfehler im `Resource type` Feld weist beispielsweise darauf hin, dass die Importvorlage und das Metadaten-Schemadateiformat nicht übereinstimmen. `Bulk operations template` Weitere Informationen finden Sie unter [AWS IoT SiteWise Auftragsschema für die Übertragung von Metadaten](#).

- Ein Fehler Asset im Resource type Feld bedeutet, dass das Asset aufgrund eines Konflikts mit einem anderen Asset nicht erstellt wurde. Informationen zu [AWS IoT SiteWiseRessourcenfehlern und Konflikten finden Sie unter Häufige Fehler](#).

Überprüfen Sie die Fehlerdetails (AWS CLI)

Informationen zur Behandlung und Diagnose von Fehlern, die während eines Übertragungsauftrags auftreten, finden Sie im folgenden Verfahren zur Verwendung der GetMetadataTransferJob API-Aktion:

1. Rufen Sie nach dem Erstellen und Ausführen eines Übertragungsauftrags folgenden Befehl auf [GetMetadataTransferJob](#):

```
aws iottwinmaker get-metadata-transfer-job \  
    --metadata-transfer-job-id your_metadata_transfer_job_id \  
    --region us-east-1
```

2. Sobald der Status des Auftrags angezeigt wird COMPLETED, können Sie mit der Überprüfung der Ergebnisse des Auftrags beginnen.
3. Wenn Sie aufrufen GetMetadataTransferJob, wird ein Objekt zurückgegeben, das aufgerufen wurde [MetadataTransferJobProgress](#).

Das MetadataTransferJobProgress Objekt enthält die folgenden Parameter:

- FailedCount: Gibt die Anzahl der Assets an, die während des Übertragungsvorgangs ausgefallen sind.
 - skippedCount: Gibt die Anzahl der Assets an, die während des Übertragungsvorgangs übersprungen wurden.
 - succeededCount: Gibt die Anzahl der Assets an, die während des Übertragungsvorgangs erfolgreich waren.
 - TotalCount: Gibt die Gesamtzahl der am Übertragungsprozess beteiligten Vermögenswerte an.
4. Darüber hinaus gibt der API-Aufruf ein Element zurück reportUrl, das eine vorsignierte URL enthält. Wenn bei Ihrem Übertragungsauftrag Probleme auftreten, die Sie weiter untersuchen müssen, besuchen Sie diese URL.

Beispiele für den Import von Metadaten

In diesem Abschnitt wird gezeigt, wie Sie Metadateiendateien erstellen, um Asset-Modelle und Assets mit einem einzigen Massenimportvorgang zu importieren.

Beispiel für einen Massenimport

Sie können viele Asset-Modelle und Assets mit einem einzigen Massenimportvorgang importieren. Das folgende Beispiel zeigt, wie Sie zu diesem Zweck eine Metadateiendatei erstellen.

In diesem Beispielszenario haben Sie verschiedene Baustellen, auf denen Industrieroboter in Arbeitszellen installiert sind.

Das Beispiel definiert zwei Anlagenmodelle:

- **RobotModel1**: Dieses Anlagenmodell stellt einen bestimmten Robotertyp dar, den Sie auf Ihren Baustellen einsetzen. Der Roboter hat eine Messeigenschaft, `Temperature`.
- **WorkCell**: Dieses Anlagenmodell stellt eine Sammlung von Robotern auf einer Ihrer Baustellen dar. Das Anlagenmodell definiert eine Hierarchie `robotHierarchyOEM1`, um die Beziehung zwischen Robotern in einer Arbeitszelle darzustellen.

Das Beispiel definiert auch einige Vermögenswerte:

- **WorkCell1**: eine Arbeitszelle an Ihrem Standort in Boston
- **RobotArm123456**: ein Roboter in dieser Arbeitszelle
- **RobotArm987654**: ein weiterer Roboter in dieser Arbeitszelle

Die folgende JSON-Metadateiendatei definiert diese Asset-Modelle und Assets. Wenn Sie einen Massenimport mit diesen Metadaten ausführen, werden die darin enthaltenen Asset-Modelle und Assets AWS IoT SiteWise einschließlich ihrer hierarchischen Beziehungen erstellt.

Metadateiendatei für den Import

```
{
  "assetModels": [
    {
      "assetModelExternalId": "Robot.OEM1.3536",
      "assetModelName": "RobotModel1",
      "assetModelProperties": [
```

```

        {
            "dataType": "DOUBLE",
            "externalId": "Temperature",
            "name": "Temperature",
            "type": {
                "measurement": {
                    "processingConfig": {
                        "forwardingConfig": {
                            "state": "ENABLED"
                        }
                    }
                }
            },
            "unit": "fahrenheit"
        }
    ],
    {
        "assetModelExternalId": "ISA95.WorkCell",
        "assetModelName": "WorkCell",
        "assetModelProperties": [],
        "assetModelHierarchies": [
            {
                "externalId": "workCellHierarchyWithOEM1Robot",
                "name": "robotHierarchyOEM1",
                "childAssetModelExternalId": "Robot.OEM1.3536"
            }
        ]
    }
],
"assets": [
    {
        "assetExternalId": "Robot.OEM1.3536.123456",
        "assetName": "RobotArm123456",
        "assetModelExternalId": "Robot.OEM1.3536"
    },
    {
        "assetExternalId": "Robot.OEM1.3536.987654",
        "assetName": "RobotArm987654",
        "assetModelExternalId": "Robot.OEM1.3536"
    },
    {
        "assetExternalId": "BostonSite.Area1.Line1.WorkCell1",
        "assetName": "WorkCell1",

```

```

    "assetModelExternalId": "ISA95.WorkCell",
    "assetHierarchies": [
      {
        "externalId": "workCellHierarchyWith0EM1Robot",
        "childAssetExternalId": "Robot.0EM1.3536.123456"
      },
      {
        "externalId": "workCellHierarchyWith0EM1Robot",
        "childAssetExternalId": "Robot.0EM1.3536.987654"
      }
    ]
  }
]
}

```

Beispiel für das erste Onboarding von Modellen und Ressourcen

In diesem Beispielszenario gibt es in einem Unternehmen verschiedene Baustellen mit Industrierobotern.

Das Beispiel definiert mehrere Anlagenmodelle:

- **Sample_Enterprise**— Dieses Vermögensmodell steht für das Unternehmen, zu dem die Standorte gehören. Das Anlagenmodell definiert eine Hierarchie `Enterprise to Site`, um die Beziehung der Standorte zum Unternehmen darzustellen.
- **Sample_Site**— Dieses Anlagenmodell repräsentiert die Produktionsstätten innerhalb des Unternehmens. Das Anlagenmodell definiert eine Hierarchie `Site to Line`, um die Beziehung der Linien zum Standort darzustellen.
- **Sample_Welding Line**— Dieses Anlagenmodell stellt eine Montagelinie innerhalb von Baustellen dar. Das Anlagenmodell definiert eine Hierarchie `Line to Robot`, um die Beziehung der Roboter zur Linie darzustellen.
- **Sample_Welding Robot**— Dieses Anlagenmodell steht für einen bestimmten Robotertyp auf Ihren Baustellen.

Das Beispiel definiert auch Vermögenswerte auf der Grundlage der Asset-Modelle.

- **Sample_AnyCompany Motor**— Dieses Asset wird anhand des **Sample_Enterprise** Asset-Modells erstellt.
- **Sample_Chicago**— Dieses Asset wurde anhand des **Sample_Site** Asset-Modells erstellt.

- **Sample_Welding Line 1**— Dieses Asset wurde anhand des **Sample_Welding Line** Asset-Modells erstellt.
- **Sample_Welding Robot 1**— Dieses Asset wurde anhand des **Sample_Welding Robot** Asset-Modells erstellt.
- **Sample_Welding Robot 2**— Dieses Asset wurde anhand des **Sample_Welding Robot** Asset-Modells erstellt.

Die folgende JSON-Metadatendatei definiert diese Asset-Modelle und Assets. Wenn Sie einen Massenimport mit diesen Metadaten ausführen, werden die darin enthaltenen Asset-Modelle und Assets AWS IoT SiteWise einschließlich ihrer hierarchischen Beziehungen erstellt.

JSON-Datei zur Einbindung von Assets und Modellen für den Import

```
{
  "assetModels": [
    {
      "assetModelExternalId": "External_Id_Welding_Robot",
      "assetModelName": "Sample_Welding Robot",
      "assetModelProperties": [
        {
          "dataType": "STRING",
          "externalId": "External_Id_Welding_Robot_Serial_Number",
          "name": "Serial Number",
          "type": {
            "attribute": {
              "defaultValue": "-"
            }
          },
          "unit": "-"
        },
        {
          "dataType": "DOUBLE",
          "externalId": "External_Id_Welding_Robot_Cycle_Count",
          "name": "CycleCount",
          "type": {
            "measurement": {}
          },
          "unit": "EA"
        }
      ]
    }
  ]
}
```

```

        "dataType": "DOUBLE",
        "externalId": "External_Id_Welding_Robot_Joint_1_Current",
        "name": "Joint 1 Current",
        "type": {
            "measurement": {}
        },
        "unit": "Amps"
    },
    {
        "dataType": "DOUBLE",
        "externalId": "External_Id_Welding_Robot_Joint_1_Max_Current",
        "name": "Max Joint 1 Current",
        "type": {
            "metric": {
                "expression": "max(joint1current)",
                "variables": [
                    {
                        "name": "joint1current",
                        "value": {
                            "propertyExternalId":
"External_Id_Welding_Robot_Joint_1_Current"
                        }
                    }
                ],
                "window": {
                    "tumbling": {
                        "interval": "5m"
                    }
                }
            }
        },
        "unit": "Amps"
    }
]
},
{
    "assetModelExternalId": "External_Id_Welding_Line",
    "assetModelName": "Sample_Welding_Line",
    "assetModelProperties": [
        {
            "dataType": "DOUBLE",
            "externalId": "External_Id_Welding_Line_Availability",
            "name": "Availability",
            "type": {

```

```

        "measurement": {}
    },
    "unit": "%"
}
],
"assetModelHierarchies": [
    {
        "externalId": "External_Id_Welding_Line_T0_Robot",
        "name": "Line to Robot",
        "childAssetModelExternalId": "External_Id_Welding_Robot"
    }
]
},
{
    "assetModelExternalId": "External_Id_Site",
    "assetModelName": "Sample_Site",
    "assetModelProperties": [
        {
            "dataType": "STRING",
            "externalId": "External_Id_Site_Street_Address",
            "name": "Street Address",
            "type": {
                "attribute": {
                    "defaultValue": "-"
                }
            },
            "unit": "-"
        }
    ],
    "assetModelHierarchies": [
        {
            "externalId": "External_Id_Site_T0_Line",
            "name": "Site to Line",
            "childAssetModelExternalId": "External_Id_Welding_Line"
        }
    ]
},
{
    "assetModelExternalId": "External_Id_Enterprise",
    "assetModelName": "Sample_Enterprise",
    "assetModelProperties": [
        {
            "dataType": "STRING",
            "name": "Company Name",

```

```

        "externalId": "External_Id_Enterprise_Company_Name",
        "type": {
            "attribute": {
                "defaultValue": "-"
            }
        },
        "unit": "-"
    }
],
"assetModelHierarchies": [
    {
        "externalId": "External_Id_Enterprise_T0_Site",
        "name": "Enterprise to Site",
        "childAssetModelExternalId": "External_Id_Site"
    }
]
},
"assets": [
    {
        "assetExternalId": "External_Id_Welding_Robot_1",
        "assetName": "Sample_Welding Robot 1",
        "assetModelExternalId": "External_Id_Welding_Robot",
        "assetProperties": [
            {
                "externalId": "External_Id_Welding_Robot_Serial_Number",
                "attributeValue": "S1000"
            },
            {
                "externalId": "External_Id_Welding_Robot_Cycle_Count",
                "alias": "AnyCompany/Chicago/Welding Line/S1000/Count"
            },
            {
                "externalId": "External_Id_Welding_Robot_Joint_1_Current",
                "alias": "AnyCompany/Chicago/Welding Line/S1000/1/Current"
            }
        ]
    },
    {
        "assetExternalId": "External_Id_Welding_Robot_2",
        "assetName": "Sample_Welding Robot 2",
        "assetModelExternalId": "External_Id_Welding_Robot",
        "assetProperties": [
            {

```

```

        "externalId": "External_Id_Welding_Robot_Serial_Number",
        "attributeValue": "S2000"
    },
    {
        "externalId": "External_Id_Welding_Robot_Cycle_Count",
        "alias": "AnyCompany/Chicago/Welding Line/S2000/Count"
    },
    {
        "externalId": "External_Id_Welding_Robot_Joint_1_Current",
        "alias": "AnyCompany/Chicago/Welding Line/S2000/1/Current"
    }
]
},
{
    "assetExternalId": "External_Id_Welding_Line_1",
    "assetName": "Sample_Welding Line 1",
    "assetModelExternalId": "External_Id_Welding_Line",
    "assetProperties": [
        {
            "externalId": "External_Id_Welding_Line_Availability",
            "alias": "AnyCompany/Chicago/Welding Line/Availability"
        }
    ],
    "assetHierarchies": [
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_1"
        },
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_2"
        }
    ]
},
{
    "assetExternalId": "External_Id_Site_Chicago",
    "assetName": "Sample_Chicago",
    "assetModelExternalId": "External_Id_Site",
    "assetHierarchies": [
        {
            "externalId": "External_Id_Site_T0_Line",
            "childAssetExternalId": "External_Id_Welding_Line_1"
        }
    ]
}
]

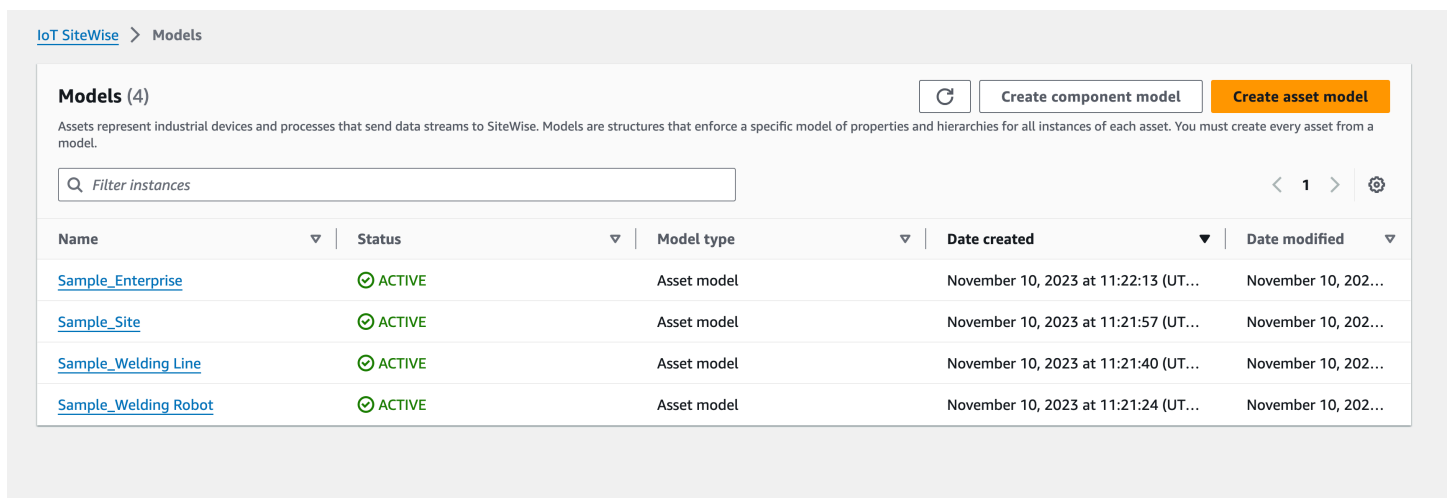
```

```

    },
    {
      "assetExternalId": "External_Id_Enterprise_AnyCompany",
      "assetName": "Sample_AnyEnterprise Motor",
      "assetModelExternalId": "External_Id_Enterprise",
      "assetHierarchies": [
        {
          "externalId": "External_Id_Enterprise_T0_Site",
          "childAssetExternalId": "External_Id_Site_Chicago"
        }
      ]
    }
  ]
}

```

Der folgende Screenshot zeigt Modelle, die AWS-IoT-SiteWise-Konsole nach der Ausführung des vorherigen Codebeispiels angezeigt werden.



The screenshot shows the AWS IoT SiteWise console interface. At the top, there's a breadcrumb navigation: [IoT SiteWise](#) > [Models](#). Below this, the section is titled "Models (4)". To the right of the title are two buttons: "Create component model" and "Create asset model". A descriptive text states: "Assets represent industrial devices and processes that send data streams to SiteWise. Models are structures that enforce a specific model of properties and hierarchies for all instances of each asset. You must create every asset from a model." Below the text is a search bar labeled "Filter instances". To the right of the search bar are navigation controls: "< 1 >" and a settings gear icon. The main part of the screenshot is a table with the following columns: Name, Status, Model type, Date created, and Date modified. The table contains four rows of data, each representing an asset model.

Name	Status	Model type	Date created	Date modified
Sample_Enterprise	ACTIVE	Asset model	November 10, 2023 at 11:22:13 (UT...)	November 10, 202...
Sample_Site	ACTIVE	Asset model	November 10, 2023 at 11:21:57 (UT...)	November 10, 202...
Sample_Welding Line	ACTIVE	Asset model	November 10, 2023 at 11:21:40 (UT...)	November 10, 202...
Sample_Welding Robot	ACTIVE	Asset model	November 10, 2023 at 11:21:24 (UT...)	November 10, 202...

Der folgende Screenshot zeigt Modelle, Anlagen und Hierarchien, die AWS-IoT-SiteWise-Konsole nach der Ausführung des vorherigen Codebeispiels angezeigt werden.

IoT SiteWise > Assets

Assets (1) Refresh Create asset

Assets represent industrial devices and processes that send data streams to SiteWise. Models are structures that enforce a specific model of properties and hierarchies for all instances of each asset. You must create every asset from a model.

Filter top level assets

Name	Description	Status	Date created	Date modified
Sample_AnyEnterprise Motor		ACTIVE	November 10, 2023 at 11:23:06 (UTC-5:00)	November 10, 2023 at 11:23:06 (UTC-5:00)
Sample_Chicago		ACTIVE	November 10, 2023 at 11:22:57 (UTC-5:00)	November 10, 2023 at 11:22:57 (UTC-5:00)
Sample_Welding Line 1		ACTIVE	November 10, 2023 at 11:22:48 (UTC-5:00)	November 10, 2023 at 11:22:48 (UTC-5:00)
Sample_Welding Robot 1		ACTIVE	November 10, 2023 at 11:22:39 (UTC-5:00)	November 10, 2023 at 11:22:39 (UTC-5:00)
Sample_Welding Robot 2		ACTIVE	November 10, 2023 at 11:22:30 (UTC-5:00)	November 10, 2023 at 11:22:30 (UTC-5:00)

Beispiel für das Onboarding zusätzlicher Ressourcen

In diesem Beispiel werden zusätzliche Vermögenswerte definiert, die in ein vorhandenes Vermögensmodell in Ihrem Konto importiert werden sollen:

- Sample_Welding Line 2— Dieses Asset wird anhand des Sample_Welding Line Asset-Modells erstellt.
- Sample_Welding Robot 3— Dieses Asset wurde anhand des Sample_Welding Robot Asset-Modells erstellt.
- Sample_Welding Robot 4— Dieses Asset wurde anhand des Sample_Welding Robot Asset-Modells erstellt.

Informationen zum Erstellen der ersten Anlagen für dieses Beispiel finden Sie unter [Beispiel für das erste Onboarding von Modellen und Ressourcen](#).

Die folgende JSON-Metadatendatei definiert diese Asset-Modelle und Assets. Wenn Sie einen Massenimport mit diesen Metadaten ausführen, werden die darin enthaltenen Asset-Modelle und Assets AWS IoT SiteWise einschließlich ihrer hierarchischen Beziehungen erstellt.

JSON-Datei zum Onboarding zusätzlicher Assets

```
{
  "assets": [
    {
      "assetExternalId": "External_Id_Welding_Robot_3",
      "assetName": "Sample_Welding Robot 3",
```

```

    "assetModelExternalId": "External_Id_Welding_Robot",
    "assetProperties": [
      {
        "externalId": "External_Id_Welding_Robot_Serial_Number",
        "attributeValue": "S3000"
      },
      {
        "externalId": "External_Id_Welding_Robot_Cycle_Count",
        "alias": "AnyCompany/Chicago/Welding Line/S3000/Count"
      },
      {
        "externalId": "External_Id_Welding_Robot_Joint_1_Current",
        "alias": "AnyCompany/Chicago/Welding Line/S3000/1/Current"
      }
    ]
  },
  {
    "assetExternalId": "External_Id_Welding_Robot_4",
    "assetName": "Sample_Welding Robot 4",
    "assetModelExternalId": "External_Id_Welding_Robot",
    "assetProperties": [
      {
        "externalId": "External_Id_Welding_Robot_Serial_Number",
        "attributeValue": "S4000"
      },
      {
        "externalId": "External_Id_Welding_Robot_Cycle_Count",
        "alias": "AnyCompany/Chicago/Welding Line/S4000/Count"
      },
      {
        "externalId": "External_Id_Welding_Robot_Joint_1_Current",
        "alias": "AnyCompany/Chicago/Welding Line/S4000/1/Current"
      }
    ]
  },
  {
    "assetExternalId": "External_Id_Welding_Line_1",
    "assetName": "Sample_Welding Line 1",
    "assetModelExternalId": "External_Id_Welding_Line",
    "assetHierarchies": [
      {
        "externalId": "External_Id_Welding_Line_T0_Robot",
        "childAssetExternalId": "External_Id_Welding_Robot_1"
      }
    ]
  }
]

```



```

        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_2"
        },
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_3"
        }
    ]
},
{
    "assetExternalId": "External_Id_Welding_Line_2",
    "assetName": "Sample_Welding Line 2",
    "assetModelExternalId": "External_Id_Welding_Line",
    "assetHierarchies": [
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_4"
        }
    ]
},
{
    "assetExternalId": "External_Id_Site_Chicago",
    "assetName": "Sample_Chicago",
    "assetModelExternalId": "External_Id_Site",
    "assetHierarchies": [
        {
            "externalId": "External_Id_Site_T0_Line",
            "childAssetExternalId": "External_Id_Welding_Line_1"
        },
        {
            "externalId": "External_Id_Site_T0_Line",
            "childAssetExternalId": "External_Id_Welding_Line_2"
        }
    ]
}
]
}

```

Der folgende Screenshot zeigt Modelle, Assets und Hierarchien, die AWS-IoT-SiteWise-Konsole nach der Ausführung des vorherigen Codebeispiels angezeigt werden.

IoT SiteWise > Assets

Assets (1) ↻ Create asset

Assets represent industrial devices and processes that send data streams to SiteWise. Models are structures that enforce a specific model of properties and hierarchies for all instances of each asset. You must create every asset from a model.

🔍 Filter top level assets < 1 > ⚙️

Name	Description	Status	Date created	Date modified
[-] Sample_AnyCompany Motor		✔️ ACTIVE	November 09, 2023 at 19:18:05 (UTC-5:00)	November 09, 2023 at 19:18:05 (UTC-5:00)
[-] Sample_Chicago		✔️ ACTIVE	November 09, 2023 at 19:17:56 (UTC-5:00)	November 09, 2023 at 19:17:56 (UTC-5:00)
[-] Sample_Welding Line 1		✔️ ACTIVE	November 09, 2023 at 19:17:48 (UTC-5:00)	November 09, 2023 at 19:17:48 (UTC-5:00)
[-] Sample_Welding Robot 2		✔️ ACTIVE	November 09, 2023 at 19:17:39 (UTC-5:00)	November 09, 2023 at 19:51:05 (UTC-5:00)
[-] Sample_Welding Robot 3		✔️ ACTIVE	November 09, 2023 at 20:40:02 (UTC-5:00)	November 09, 2023 at 20:40:02 (UTC-5:00)
[-] Sample_Welding Robot 1		✔️ ACTIVE	November 09, 2023 at 19:17:30 (UTC-5:00)	November 09, 2023 at 19:51:05 (UTC-5:00)
[-] Sample_Welding Line 2		✔️ ACTIVE	November 09, 2023 at 20:40:20 (UTC-5:00)	November 09, 2023 at 20:40:20 (UTC-5:00)
[-] Sample_Welding Robot 4		✔️ ACTIVE	November 09, 2023 at 20:40:11 (UTC-5:00)	November 09, 2023 at 20:40:11 (UTC-5:00)

Beispiel für das Onboarding neuer Immobilien

In diesem Beispiel werden neue Immobilien in bestehenden Anlagemodellen definiert. Erfahren Sie [Beispiel für das Onboarding zusätzlicher Ressourcen](#), wie Sie zusätzliche Anlagen und Modelle integrieren können.

- **Joint 1 Temperature**— Diese Eigenschaft wird dem `Sample_Welding Robot` Asset-Modell hinzugefügt. Diese neue Eigenschaft wird auch auf jedes Asset übertragen, das mit dem `Sample_Welding Robot` Asset-Modell erstellt wurde.

Informationen zum Hinzufügen einer neuen Eigenschaft zu einem vorhandenen Asset-Modell finden Sie im folgenden Beispiel für eine JSON-Metadatendatei. Wie in der JSON-Datei gezeigt, muss die gesamte bestehende `Sample_Welding Robot` Asset-Modelldefinition zusammen mit der neuen Eigenschaft bereitgestellt werden. Wenn die gesamte Eigenschaftsliste aus der vorhandenen Definition nicht bereitgestellt wird, werden die ausgelassenen Eigenschaften AWS IoT SiteWise gelöscht.

JSON-Datei zum Integrieren neuer Eigenschaften

In diesem Beispiel wird dem Asset-Modell eine neue Eigenschaft `Joint 1 Temperature` hinzugefügt.

```
{
```

```

"assetModels": [
  {
    "assetModelExternalId": "External_Id_Welding_Robot",
    "assetModelName": "Sample_Welding Robot",
    "assetModelProperties": [
      {
        "dataType": "STRING",
        "externalId": "External_Id_Welding_Robot_Serial_Number",
        "name": "Serial Number",
        "type": {
          "attribute": {
            "defaultValue": "-"
          }
        },
        "unit": "-"
      },
      {
        "dataType": "DOUBLE",
        "externalId": "External_Id_Welding_Robot_Cycle_Count",
        "name": "CycleCount",
        "type": {
          "measurement": {}
        },
        "unit": "EA"
      },
      {
        "dataType": "DOUBLE",
        "externalId": "External_Id_Welding_Robot_Joint_1_Current",
        "name": "Joint 1 Current",
        "type": {
          "measurement": {}
        },
        "unit": "Amps"
      },
      {
        "dataType": "DOUBLE",
        "externalId": "External_Id_Welding_Robot_Joint_1_Max_Current",
        "name": "Max Joint 1 Current",
        "type": {
          "metric": {
            "expression": "max(joint1current)",
            "variables": [
              {
                "name": "joint1current",

```

```

        "value": {
            "propertyExternalId":
"External_Id_Welding_Robot_Joint_1_Current"
        }
    },
    "window": {
        "tumbling": {
            "interval": "5m"
        }
    }
},
"unit": "Amps"
},
{
    "dataType": "DOUBLE",
    "externalId": "External_Id_Welding_Robot_Joint_1_Temperature",
    "name": "Joint 1 Temperature",
    "type": {
        "measurement": {}
    },
    "unit": "degC"
}
]
}
]
}

```

Beispiel für die Verwaltung von Datenströmen

Dieses Beispiel zeigt zwei Möglichkeiten zur Verwaltung von Datenströmen, die mit einer Anlageneigenschaft verknüpft sind. Beim Umbenennen eines Alias für eine Anlageneigenschaft gibt es zwei Optionen für die historischen Daten, die derzeit im Datenstrom der Anlageneigenschaft gespeichert sind.

- Option eins — Behalten Sie den aktuellen Datenstrom bei und benennen Sie nur den Alias um, sodass auf die historischen Daten mit dem neuen Alias zugegriffen werden kann.

Im Beispiel für die JSON-Metadatendatei `External_Id_Welding_Robot_Cycle_Count` ändert die Asset-Eigenschaft mit der ID ihren Alias in `AnyCompany/Chicago/Welding Line/`

S3000/Count-Updated. Die historischen Daten für diese Asset-Eigenschaft bleiben nach dieser Änderung unverändert.

- Option zwei — Weisen Sie der Asset-Eigenschaft, auf die mit dem neuen Alias zugegriffen werden kann, einen neuen Datenstrom zu. Auf den alten Datenstrom kann zusammen mit seinen historischen Daten weiterhin mit dem alten Alias zugegriffen werden, er ist jedoch keiner Anlageneigenschaft zugeordnet.

Im Beispiel für die JSON-Metadatendatei `External_Id_Welding_Robot_Joint_1_Current` ändert die Asset-Eigenschaft mit der ID ihren Alias in `AnyCompany/Chicago/Welding Line/S4999/1/Current`. Diesmal `retainDataOnAliasChange` ist der zusätzliche Wert vorhanden und auf `false` gesetzt. Mit dieser Einstellung wird der ursprüngliche Datenstrom von der Objekteigenschaft getrennt, und es wird ein neuer Datenstrom erstellt, der keine historischen Daten enthält.

Um auf den alten Datenstream mit den ursprünglichen historischen Daten zuzugreifen, rufen Sie die AWS Console Home Seite Datenströme auf und suchen Sie nach dem alten Alias `AnyCompany/Chicago/Welding Line/S3000/1/Current`.

JSON-Datei zur Aktualisierung von Eigenschafts-Aliasnamen

```
{
  "assetExternalId": "External_Id_Welding_Robot_3",
  "assetName": "Sample_Welding Robot 3",
  "assetModelExternalId": "External_Id_Welding_Robot",
  "assetProperties": [
    {
      "externalId": "External_Id_Welding_Robot_Serial_Number",
      "attributeValue": "S3000"
    },
    {
      "externalId": "External_Id_Welding_Robot_Cycle_Count",
      "alias": "AnyCompany/Chicago/Welding Line/S3000/Count-Updated"
    },
    {
      "externalId": "External_Id_Welding_Robot_Joint_1_Current",
      "alias": "AnyCompany/Chicago/Welding Line/S4999/1/Current",
      "retainDataOnAliasChange": "FALSE"
    }
  ]
}
```

```
}
```

Beispiele für den Export von Metadaten

Wenn Sie einen Massenexport Ihrer AWS IoT SiteWise Inhalte nach Amazon S3 durchführen, können Sie Filter angeben, um einzuschränken, welche spezifischen Asset-Modelle und Assets Sie exportieren möchten.

Sie geben die Filter in einem `iotSiteWiseConfiguration` Abschnitt innerhalb des `sources` Abschnitts Ihres Anfragetextes an.

Note

Sie können mehrere Filter einbeziehen. Bei der Massenoperation werden alle Asset-Modelle oder Assets exportiert, die einem der Filter entsprechen.

Wenn Sie keine Filter angeben, exportiert der Vorgang alle Ihre Asset-Modelle und Assets.

```
{
  "metadataTransferJobId": "your-transfer-job-id",
  "sources": [{
    "type": "iotsitewise",
    "iotSiteWiseConfiguration": {
      "filters": [{
        list of filters
      }]
    }
  ]],
  "destination": {
    "type": "s3",
    "s3Configuration": {
      "location": "arn:aws:s3:::amzn-s3-demo-bucket"
    }
  }
}
```

Filtern Sie nach Asset-Modell

Sie können ein bestimmtes Asset-Modell filtern. Sie können auch alle Anlagen, die dieses Modell verwenden, oder alle Anlagenmodelle innerhalb seiner Hierarchie einbeziehen. Sie können nicht sowohl Vermögenswerte als auch Hierarchien einbeziehen.

Weitere Informationen zu Hierarchien finden Sie unter [Definieren Sie die Hierarchien der Anlagenmodelle](#).

Asset model

Dieser Filter umfasst das angegebene Asset-Modell:

```
"filterByAssetModel": {  
  "assetModelId": "asset model ID"  
}
```

Asset model and its assets

Dieser Filter umfasst das angegebene Asset-Modell sowie alle Assets, die dieses Asset-Modell verwenden:

```
"filterByAssetModel": {  
  "assetModelId": "asset model ID",  
  "includeAssets": true  
}
```

Asset model and its hierarchy

Dieser Filter umfasst das angegebene Asset-Modell zusammen mit allen zugehörigen Asset-Modellen in seiner Hierarchie:

```
"filterByAssetModel": {  
  "assetModelId": "asset model ID",  
  "includeOffspring": true  
}
```

Nach Vermögenswert filtern

Sie können ein bestimmtes Asset filtern. Sie können auch das zugehörige Asset-Modell oder alle zugehörigen Assets in die Hierarchie einbeziehen. Sie können nicht sowohl das Vermögensmodell als auch die Hierarchie einbeziehen.

Weitere Informationen zu Hierarchien finden Sie unter [Definieren Sie die Hierarchien der Anlagenmodelle](#).

Asset

Dieser Filter umfasst das angegebene Asset:

```
"filterByAsset": {  
  "assetId": "asset ID"  
}
```

Asset and its asset model

Dieser Filter umfasst das angegebene Asset zusammen mit dem verwendeten Asset-Modell:

```
"filterByAsset": {  
  "assetId": "asset ID",  
  "includeAssetModel": true  
}
```

Asset and its hierarchy

Dieser Filter umfasst das angegebene Asset zusammen mit allen zugehörigen Assets in seiner Hierarchie:

```
"filterByAsset": {  
  "assetId": "asset ID",  
  "includeOffspring": true  
}
```

AWS IoT SiteWiseAuftragsschema für die Übertragung von Metadaten

Verwenden Sie das Auftragsschema für die Übertragung von AWS IoT SiteWise Metadaten als Referenz, wenn Sie Ihre eigenen Massenimport- und Exportvorgänge durchführen:


```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "IoTSiteWise",
  "description": "Metadata transfer job resource schema for IoTSiteWise",
  "definitions": {
    "Name": {
      "type": "string",
      "minLength": 1,
      "maxLength": 256,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "Description": {
      "type": "string",
      "minLength": 1,
      "maxLength": 2048,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "ID": {
      "type": "string",
      "minLength": 36,
      "maxLength": 36,
      "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$"
    },
    "ExternalId": {
      "type": "string",
      "minLength": 2,
      "maxLength": 128,
      "pattern": "[a-zA-Z0-9_][a-zA-Z_\\-0-9.:]*[a-zA-Z0-9_]+"
    },
    "AttributeValue": {
      "description": "The value of the property attribute.",
      "type": "string",
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "PropertyUnit": {
      "description": "The unit of measure (such as Newtons or RPM) of the asset property.",
      "type": "string",
      "minLength": 1,
      "maxLength": 256,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "PropertyAlias": {

```

```

    "description": "The property alias that identifies the property.",
    "type": "string",
    "minLength": 1,
    "maxLength": 1000,
    "pattern": "[^\\u0000-\\u001F\\u007F]+"
  },
  "AssetProperty": {
    "description": "The asset property's definition, alias, unit, and notification state.",
    "type": "object",
    "additionalProperties": false,
    "anyOf": [
      {
        "required": [
          "id"
        ]
      },
      {
        "required": [
          "externalId"
        ]
      }
    ],
    "properties": {
      "id": {
        "description": "The ID of the asset property.",
        "$ref": "#/definitions/ID"
      },
      "externalId": {
        "description": "The ExternalID of the asset property.",
        "$ref": "#/definitions/ExternalId"
      },
      "alias": {
        "$ref": "#/definitions/PropertyAlias"
      },
      "unit": {
        "$ref": "#/definitions/PropertyUnit"
      },
      "attributeValue": {
        "$ref": "#/definitions/AttributeValue"
      },
      "retainDataOnAliasChange": {
        "type": "string",
        "default": "TRUE",

```

```

        "enum": [
            "TRUE",
            "FALSE"
        ],
    },
    "propertyNotificationState": {
        "description": "The MQTT notification state (ENABLED or DISABLED) for this
asset property.",
        "type": "string",
        "enum": [
            "ENABLED",
            "DISABLED"
        ]
    }
},
"AssetHierarchy": {
    "description": "A hierarchy specifies allowed parent/child asset relationships.",
    "type": "object",
    "additionalProperties": false,
    "anyOf": [
        {
            "required": [
                "id",
                "childAssetId"
            ]
        },
        {
            "required": [
                "externalId",
                "childAssetId"
            ]
        },
        {
            "required": [
                "id",
                "childAssetExternalId"
            ]
        },
        {
            "required": [
                "externalId",
                "childAssetExternalId"
            ]
        }
    ]
}

```

```

    }
  ],
  "properties": {
    "id": {
      "description": "The ID of a hierarchy in the parent asset's model.",
      "$ref": "#/definitions/ID"
    },
    "externalId": {
      "description": "The ExternalID of a hierarchy in the parent asset's model.",
      "$ref": "#/definitions/ExternalId"
    },
    "childAssetId": {
      "description": "The ID of the child asset to be associated.",
      "$ref": "#/definitions/ID"
    },
    "childAssetExternalId": {
      "description": "The ExternalID of the child asset to be associated.",
      "$ref": "#/definitions/ExternalId"
    }
  }
},
"Tag": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "key",
    "value"
  ],
  "properties": {
    "key": {
      "type": "string"
    },
    "value": {
      "type": "string"
    }
  }
},
"AssetModelType": {
  "type": "string",
  "default": null,
  "enum": [
    "ASSET_MODEL",
    "COMPONENT_MODEL"
  ]
}

```

```

    },
    "AssetModelCompositeModel": {
      "description": "Contains a composite model definition in an asset model. This composite model definition is applied to all assets created from the asset model.",
      "type": "object",
      "additionalProperties": false,
      "anyOf": [
        {
          "required": [
            "id"
          ]
        },
        {
          "required": [
            "externalId"
          ]
        }
      ],
      "required": [
        "name",
        "type"
      ],
      "properties": {
        "id": {
          "description": "The ID of the asset model composite model.",
          "$ref": "#/definitions/ID"
        },
        "externalId": {
          "description": "The ExternalID of the asset model composite model.",
          "$ref": "#/definitions/ExternalId"
        },
        "parentId": {
          "description": "The ID of the parent asset model composite model.",
          "$ref": "#/definitions/ID"
        },
        "parentExternalId": {
          "description": "The ExternalID of the parent asset model composite model.",
          "$ref": "#/definitions/ExternalId"
        },
        "composedAssetModelId": {
          "description": "The ID of the composed asset model.",
          "$ref": "#/definitions/ID"
        },
        "composedAssetModelExternalId": {

```

```

        "description": "The ExternalID of the composed asset model.",
        "$ref": "#/definitions/ExternalId"
    },
    "description": {
        "description": "A description for the asset composite model.",
        "$ref": "#/definitions/Description"
    },
    "name": {
        "description": "A unique, friendly name for the asset composite model.",
        "$ref": "#/definitions/Name"
    },
    "type": {
        "description": "The type of the composite model. For alarm composite models,
this type is AWS/ALARM.",
        "$ref": "#/definitions/Name"
    },
    "properties": {
        "description": "The property definitions of the asset model.",
        "type": "array",
        "items": {
            "$ref": "#/definitions/AssetModelProperty"
        }
    }
},
"AssetModelProperty": {
    "description": "Contains information about an asset model property.",
    "type": "object",
    "additionalProperties": false,
    "anyOf": [
        {
            "required": [
                "id"
            ]
        },
        {
            "required": [
                "externalId"
            ]
        }
    ],
    "required": [
        "name",
        "dataType",

```

```

    "type"
  ],
  "properties": {
    "id": {
      "description": "The ID of the asset model property.",
      "$ref": "#/definitions/ID"
    },
    "externalId": {
      "description": "The ExternalID of the asset model property.",
      "$ref": "#/definitions/ExternalId"
    },
    "name": {
      "description": "The name of the asset model property.",
      "$ref": "#/definitions/Name"
    },
    "dataType": {
      "description": "The data type of the asset model property.",
      "$ref": "#/definitions/DataType"
    },
    "dataTypeSpec": {
      "description": "The data type of the structure for this property.",
      "$ref": "#/definitions/Name"
    },
    "unit": {
      "description": "The unit of the asset model property, such as Newtons or
RPM.",
      "type": "string",
      "minLength": 1,
      "maxLength": 256,
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    },
    "type": {
      "description": "The property type",
      "$ref": "#/definitions/PropertyType"
    }
  }
},
"DataType": {
  "type": "string",
  "enum": [
    "STRING",
    "INTEGER",
    "DOUBLE",
    "BOOLEAN",

```

```

    "STRUCT"
  ]
},
"PropertyType": {
  "description": "Contains a property type, which can be one of attribute,
measurement, metric, or transform.",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "attribute": {
      "$ref": "#/definitions/Attribute"
    },
    "transform": {
      "$ref": "#/definitions/Transform"
    },
    "metric": {
      "$ref": "#/definitions/Metric"
    },
    "measurement": {
      "$ref": "#/definitions/Measurement"
    }
  }
},
"Attribute": {
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "defaultValue": {
      "type": "string",
      "pattern": "[^\\u0000-\\u001F\\u007F]+"
    }
  }
},
"Transform": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "expression",
    "variables"
  ],
  "properties": {
    "expression": {
      "description": "The mathematical expression that defines the transformation
function.",

```



```

        "type": "string",
        "minLength": 1,
        "maxLength": 1024
    },
    "variables": {
        "description": "The list of variables used in the expression.",
        "type": "array",
        "items": {
            "$ref": "#/definitions/ExpressionVariable"
        }
    },
    "processingConfig": {
        "$ref": "#/definitions/TransformProcessingConfig"
    }
},
"TransformProcessingConfig": {
    "description": "The processing configuration for the given transform property.",
    "type": "object",
    "additionalProperties": false,
    "required": [
        "computeLocation"
    ],
    "properties": {
        "computeLocation": {
            "description": "The compute location for the given transform property.",
            "$ref": "#/definitions/ComputeLocation"
        },
        "forwardingConfig": {
            "description": "The forwarding configuration for a given property.",
            "$ref": "#/definitions/ForwardingConfig"
        }
    }
},
"Metric": {
    "type": "object",
    "additionalProperties": false,
    "required": [
        "expression",
        "variables",
        "window"
    ],
    "properties": {
        "expression": {

```

```

        "description": "The mathematical expression that defines the metric
aggregation function.",
        "type": "string",
        "minLength": 1,
        "maxLength": 1024
    },
    "variables": {
        "description": "The list of variables used in the expression.",
        "type": "array",
        "items": {
            "$ref": "#/definitions/ExpressionVariable"
        }
    },
    "window": {
        "description": "The window (time interval) over which AWS IoT SiteWise
computes the metric's aggregation expression",
        "$ref": "#/definitions/MetricWindow"
    },
    "processingConfig": {
        "$ref": "#/definitions/MetricProcessingConfig"
    }
}
},
"MetricProcessingConfig": {
    "description": "The processing configuration for the metric.",
    "type": "object",
    "additionalProperties": false,
    "required": [
        "computeLocation"
    ],
    "properties": {
        "computeLocation": {
            "description": "The compute location for the given metric property.",
            "$ref": "#/definitions/ComputeLocation"
        }
    }
},
"ComputeLocation": {
    "type": "string",
    "enum": [
        "EDGE",
        "CLOUD"
    ]
},

```

```
"ForwardingConfig": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "state"
  ],
  "properties": {
    "state": {
      "type": "string",
      "enum": [
        "ENABLED",
        "DISABLED"
      ]
    }
  }
},
"MetricWindow": {
  "description": "Contains a time interval window used for data aggregate computations (for example, average, sum, count, and so on).",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "tumbling": {
      "description": "The tumbling time interval window.",
      "type": "object",
      "additionalProperties": false,
      "required": [
        "interval"
      ],
      "properties": {
        "interval": {
          "description": "The time interval for the tumbling window.",
          "type": "string",
          "minLength": 2,
          "maxLength": 23
        }
      },
      "offset": {
        "description": "The offset for the tumbling window.",
        "type": "string",
        "minLength": 2,
        "maxLength": 25
      }
    }
  }
}
```

```

    }
  },
  "ExpressionVariable": {
    "type": "object",
    "additionalProperties": false,
    "required": [
      "name",
      "value"
    ],
    "properties": {
      "name": {
        "description": "The friendly name of the variable to be used in the
expression.",
        "type": "string",
        "minLength": 1,
        "maxLength": 64,
        "pattern": "^[a-z][a-z0-9_]*$"
      },
      "value": {
        "description": "The variable that identifies an asset property from which to
use values.",
        "$ref": "#/definitions/VariableValue"
      }
    }
  },
  "VariableValue": {
    "type": "object",
    "additionalProperties": false,
    "anyOf": [
      {
        "required": [
          "propertyId"
        ]
      },
      {
        "required": [
          "propertyExternalId"
        ]
      }
    ],
    "properties": {
      "propertyId": {
        "$ref": "#/definitions/ID"
      }
    }
  },

```

```

    "propertyExternalId": {
      "$ref": "#/definitions/ExternalId"
    },
    "hierarchyId": {
      "$ref": "#/definitions/ID"
    },
    "hierarchyExternalId": {
      "$ref": "#/definitions/ExternalId"
    }
  }
},
"Measurement": {
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "processingConfig": {
      "$ref": "#/definitions/MeasurementProcessingConfig"
    }
  }
},
"MeasurementProcessingConfig": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "forwardingConfig"
  ],
  "properties": {
    "forwardingConfig": {
      "description": "The forwarding configuration for the given measurement
property.",
      "$ref": "#/definitions/ForwardingConfig"
    }
  }
},
"AssetModelHierarchy": {
  "description": "Contains information about an asset model hierarchy.",
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "id",
        "childAssetModelId"
      ]
    }
  ]
}

```

```

    },
    {
      "required": [
        "id",
        "childAssetModelExternalId"
      ]
    },
    {
      "required": [
        "externalId",
        "childAssetModelId"
      ]
    },
    {
      "required": [
        "externalId",
        "childAssetModelExternalId"
      ]
    }
  ],
  "required": [
    "name"
  ],
  "properties": {
    "id": {
      "description": "The ID of the asset model hierarchy.",
      "$ref": "#/definitions/ID"
    },
    "externalId": {
      "description": "The ExternalID of the asset model hierarchy.",
      "$ref": "#/definitions/ExternalId"
    },
    "name": {
      "description": "The name of the asset model hierarchy.",
      "$ref": "#/definitions/Name"
    },
    "childAssetModelId": {
      "description": "The ID of the asset model. All assets in this hierarchy must be instances of the child AssetModelId asset model.",
      "$ref": "#/definitions/ID"
    },
    "childAssetModelExternalId": {
      "description": "The ExternalID of the asset model. All assets in this hierarchy must be instances of the child AssetModelId asset model.",

```

```

        "$ref": "#/definitions/ExternalId"
    }
}
},
"AssetModel": {
    "type": "object",
    "additionalProperties": false,
    "anyOf": [
        {
            "required": [
                "assetModelId"
            ]
        },
        {
            "required": [
                "assetModelExternalId"
            ]
        }
    ],
    "required": [
        "assetModelName"
    ],
    "properties": {
        "assetModelId": {
            "description": "The ID of the asset model.",
            "$ref": "#/definitions/ID"
        },
        "assetModelExternalId": {
            "description": "The ID of the asset model.",
            "$ref": "#/definitions/ExternalId"
        },
        "assetModelName": {
            "description": "A unique, friendly name for the asset model.",
            "$ref": "#/definitions/Name"
        },
        "assetModelDescription": {
            "description": "A description for the asset model.",
            "$ref": "#/definitions/Description"
        },
        "assetModelType": {
            "description": "The type of the asset model.",
            "$ref": "#/definitions/AssetModelType"
        },
        "assetModelProperties": {

```

```

        "description": "The property definitions of the asset model.",
        "type": "array",
        "items": {
            "$ref": "#/definitions/AssetModelProperty"
        }
    },
    "assetModelCompositeModels": {
        "description": "The composite asset models that are part of this asset model.
Composite asset models are asset models that contain specific properties.",
        "type": "array",
        "items": {
            "$ref": "#/definitions/AssetModelCompositeModel"
        }
    },
    "assetModelHierarchies": {
        "description": "The hierarchy definitions of the asset model. Each hierarchy
specifies an asset model whose assets can be children of any other assets created from
this asset model.",
        "type": "array",
        "items": {
            "$ref": "#/definitions/AssetModelHierarchy"
        }
    },
    "tags": {
        "description": "A list of key-value pairs that contain metadata for the asset
model.",
        "type": "array",
        "items": {
            "$ref": "#/definitions/Tag"
        }
    }
},
"Asset": {
    "type": "object",
    "additionalProperties": false,
    "anyOf": [
        {
            "required": [
                "assetId",
                "assetModelId"
            ]
        }
    ],
    {

```



```

        "required": [
            "assetExternalId",
            "assetModelId"
        ],
    },
    {
        "required": [
            "assetId",
            "assetModelExternalId"
        ],
    },
    {
        "required": [
            "assetExternalId",
            "assetModelExternalId"
        ],
    }
],
"required": [
    "assetName"
],
"properties": {
    "assetId": {
        "description": "The ID of the asset",
        "$ref": "#/definitions/ID"
    },
    "assetExternalId": {
        "description": "The external ID of the asset",
        "$ref": "#/definitions/ExternalId"
    },
    "assetModelId": {
        "description": "The ID of the asset model from which to create the asset.",
        "$ref": "#/definitions/ID"
    },
    "assetModelExternalId": {
        "description": "The ExternalID of the asset model from which to create the
asset.",
        "$ref": "#/definitions/ExternalId"
    },
    "assetName": {
        "description": "A unique, friendly name for the asset.",
        "$ref": "#/definitions/Name"
    },
    "assetDescription": {

```

```

        "description": "A description for the asset",
        "$ref": "#/definitions/Description"
    },
    "assetProperties": {
        "type": "array",
        "items": {
            "$ref": "#/definitions/AssetProperty"
        }
    },
    "assetHierarchies": {
        "type": "array",
        "items": {
            "$ref": "#/definitions/AssetHierarchy"
        }
    },
    "tags": {
        "description": "A list of key-value pairs that contain metadata for the
asset.",
        "type": "array",
        "uniqueItems": false,
        "items": {
            "$ref": "#/definitions/Tag"
        }
    }
}
},
"additionalProperties": false,
"properties": {
    "assetModels": {
        "type": "array",
        "uniqueItems": false,
        "items": {
            "$ref": "#/definitions/AssetModel"
        }
    },
    "assets": {
        "type": "array",
        "uniqueItems": false,
        "items": {
            "$ref": "#/definitions/Asset"
        }
    }
}
}

```

```
}  

```

Überwachen Sie Daten mit AlarmenAWS IoT SiteWise

Sie können Alarme für Ihre Daten konfigurieren, um Ihr Team zu benachrichtigen, wenn Geräte oder Prozesse nicht optimal funktionieren. Optimale Leistung eines Geräts oder Prozesses bedeutet, dass die Werte für bestimmte Metriken innerhalb des Bereichs zwischen einem unteren und einem oberen Grenzwert liegen sollten. Wenn diese Metriken außerhalb ihres Betriebsbereichs liegen, müssen Gerätebediener benachrichtigt werden, damit sie das Problem beheben können. Verwenden Sie Alarme, um Probleme schnell zu erkennen und die Bediener zu benachrichtigen, um die Leistung Ihrer Geräte und Prozesse zu maximieren.

Themen

- [Arten von Alarmen](#)
- [Alarmzustände](#)
- [Eigenschaften des Alarmstatus](#)
- [Definieren Sie Alarme für Asset-Modelle inAWS IoT SiteWise](#)
- [Konfigurieren Sie Alarme für Anlagen inAWS IoT SiteWise](#)
- [Reagieren Sie auf Alarme inAWS IoT SiteWise](#)
- [Erfassen Sie einen externen Alarmstatus inAWS IoT SiteWise](#)

Arten von Alarmen

Sie können Alarme definieren, die in der AWS Cloud erkannt werden, und Alarme, die Sie bei externen Prozessen erkennen. AWS IoT SiteWiseunterstützt die folgenden Arten von Alarmen:

- AWS IoT EventsAlarme

Note

Hinweis zum Ende des Supports: Der Support für AWS wurde beendetAWS IoT Events. Weitere Informationen finden Sie unter [AWS IoT EventsEnde des Supports](#).

AWS IoT EventsAlarme sind Alarme, die eintreffenAWS IoT Events. AWS IoT SiteWisesendet Eigenschaftswerte einer Anlage an ein Alarmmodell inAWS IoT Events. AWS IoT EventsSendet dann den Alarmstatus anAWS IoT SiteWise. Sie können Optionen konfigurieren, z. B. wann der

Alarm erkannt wird und wen benachrichtigt werden soll, wenn sich der Alarmstatus ändert. Sie können auch die [AWS IoT Events Aktionen definieren, die ausgeführt](#) werden, wenn sich der Alarmstatus ändert.

Alarme in AWS IoT Events sind Instanzen von Alarmmodellen. Das Alarmmodell spezifiziert den Schwellenwert und den Schweregrad des Alarms, was zu tun ist, wenn sich der Alarmstatus ändert, und vieles mehr. Wenn Sie jedes Merkmal des Alarmmodells konfigurieren, geben Sie eine Attributeigenschaft aus dem Asset-Modell an, das der Alarm überwacht. Alle auf dem Asset-Modell basierenden Assets verwenden den Wert des Attributs, wenn sie dieses AWS IoT Events Merkmal des Alarms auswerten. Weitere Informationen finden Sie unter [Verwenden von Alarmen](#) im AWS IoT Events Entwicklerhandbuch.

Sie können auf einen AWS IoT Events Alarm reagieren, wenn er seinen Status ändert. Sie können beispielsweise einen Alarm bestätigen oder die Schlummerfunktion deaktivieren, wenn er aktiv wird. Sie können Alarme auch aktivieren, deaktivieren und zurücksetzen.

SiteWise Monitor-Benutzer können AWS IoT Events Alarme in SiteWise Monitor-Portalen visualisieren, konfigurieren und darauf reagieren. Weitere Informationen finden Sie unter [Überwachung mit Alarmen](#) im AWS IoT SiteWise Monitor Anwendungsleitfaden.

 Note

AWS IoT Events Für die Auswertung dieser Alarme und die Übertragung von Daten zwischen AWS IoT SiteWise und fallen Gebühren an AWS IoT Events. Weitere Informationen finden Sie unter [AWS IoT Events Preise](#).

- Externe Alarme

Externe Alarme sind Alarme, die Sie außerhalb auswerten AWS IoT SiteWise. Verwenden Sie externe Alarme, wenn Sie über eine Datenquelle verfügen, die den Alarmstatus meldet. Der externe Alarm enthält eine Messeigenschaft, in die Sie die Alarmzustandsdaten aufnehmen.

Sie können einen externen Alarm nicht bestätigen oder in den Schlummermodus versetzen, wenn er seinen Status ändert.

SiteWise Monitor-Benutzer können den Status externer Alarme in SiteWise Monitor-Portalen sehen, sie können diese Alarme jedoch nicht konfigurieren oder darauf reagieren.

AWS IoT SiteWise bewertet den Status externer Alarme nicht.

Alarmzustände

Industriearme enthalten Informationen über den Zustand der Geräte oder Prozesse, die sie überwachen, und (optional) Informationen über die Reaktion des Bedieners auf den Alarmzustand.

Wenn Sie einen AWS IoT Events Alarm definieren, geben Sie an, ob der Bestätigungsfluss aktiviert werden soll oder nicht. Der Bestätigungsfluss ist standardmäßig aktiviert. Wenn Sie diese Option aktivieren, können die Bediener den Alarm bestätigen und eine Notiz mit Einzelheiten zum Alarm oder zu den Maßnahmen hinterlassen, die sie zu seiner Behebung ergriffen haben. Wenn ein Bediener einen aktiven Alarm nicht bestätigt, bevor er inaktiv wird, wird der Alarm gesperrt. Der verriegelte Zustand bedeutet, dass der Alarm aktiv wurde und nicht bestätigt wurde. Der Bediener muss also die Ausrüstung oder den Prozess überprüfen und den eingeschalteten Alarm bestätigen.

Alarmer haben die folgenden Zustände:

- **Normal (Normal)** — Der Alarm ist aktiviert, aber inaktiv. Der industrielle Prozess oder die industrielle Ausrüstung funktioniert erwartungsgemäß.
- **Aktiv (Active)** — Der Alarm ist aktiv. Der industrielle Prozess oder die industrielle Ausrüstung befindet sich außerhalb des Betriebsbereichs und erfordert besondere Aufmerksamkeit.
- **Bestätigt (Acknowledged)** — Ein Bediener hat den Zustand des Alarms bestätigt.

Dieser Status gilt nur für Alarmer, bei denen Sie den Bestätigungsfluss aktivieren.

- **Eingeschaltet (Latched)** — Der Alarm wurde wieder normal, war aber aktiv und kein Bediener hat ihn bestätigt. Der industrielle Prozess oder die industrielle Ausrüstung erfordert die Aufmerksamkeit eines Bedieners, um den Alarm wieder in den Normalzustand zu versetzen.

Dieser Status gilt nur für Alarmer, bei denen Sie den Bestätigungsfluss aktivieren.

- **Snoozed (SnoozeDisabled)** — Der Alarm ist deaktiviert, weil ein Bediener den Alarm ausgeschaltet hat. Der Operator definiert die Dauer, für die der Alarm in den Schlummermodus versetzt wird. Nach dieser Dauer kehrt der Alarm in den Normalzustand zurück.
- **Deaktiviert (Disabled)** — Der Alarm ist deaktiviert und wird nicht erkannt.

Eigenschaften des Alarmstatus

AWS IoT SiteWise speichert Alarmzustandsdaten als JSON-Objekt, das in eine Zeichenfolge serialisiert ist. Dieses Objekt enthält den Status und zusätzliche Informationen über den Alarm, z. B. Aktionen zur Reaktion des Bedieners und die Regel, die der Alarm auswertet.

Sie identifizieren die Alarmstatuseigenschaft anhand ihres Namens und Strukturtyps, `AWS/ALARM_STATE`. Weitere Informationen finden Sie unter [Definieren Sie Alarme für Asset-Modelle in AWS IoT SiteWise](#).

Das Datenobjekt für den Alarmstatus enthält die folgenden Informationen:

`stateName`

Der Zustand des Alarms. Weitere Informationen finden Sie unter [Alarmzustände](#).

Datentyp: `STRING`

`customerAction`

(Optional) Ein Objekt, das Informationen über die Reaktion eines Bedieners auf den Alarm enthält. Bediener können Alarme aktivieren, deaktivieren, bestätigen und die Schlummerfunktion aktivieren. Wenn sie dies tun, umfassen die Daten zum Alarmstatus ihre Reaktion und den Hinweis, den sie hinterlassen können, wenn sie reagieren. Dieses Objekt enthält die folgenden Informationen:

`actionName`

Der Name der Aktion, die der Bediener ergreift, um auf den Alarm zu reagieren. Dieser Wert enthält eine der folgenden Zeichenketten:

- `ENABLE`
- `DISABLE`
- `SNOOZE`
- `ACKNOWLEDGE`
- `RESET`

Datentyp: `STRING`

`enable`

(Optional) Ein Objekt, das vorhanden ist, `customerAction` wenn der Bediener den Alarm aktiviert. Wenn ein Bediener den Alarm aktiviert, wechselt der Alarmstatus zu `Normal`. Dieses Objekt enthält die folgenden Informationen:

`note`

(Optional) Die Notiz, die der Kunde hinterlässt, wenn er den Alarm aktiviert.

Datentyp: `STRING`

Maximale Länge: 128 Zeichen

disable

(Optional) Ein Objekt, das vorhanden ist, `customerAction` wenn der Bediener den Alarm deaktiviert. Wenn ein Bediener den Alarm aktiviert, wechselt der Alarmstatus zu `Disabled`. Dieses Objekt enthält die folgenden Informationen:

note

(Optional) Die Notiz, die der Kunde hinterlässt, wenn er den Alarm deaktiviert.

Datentyp: `STRING`

Maximale Länge: 128 Zeichen

acknowledge

(Optional) Ein Objekt, das vorhanden ist, `customerAction` wenn der Bediener den Alarm bestätigt. Wenn ein Bediener den Alarm aktiviert, wechselt der Alarmstatus zu `Acknowledged`. Dieses Objekt enthält die folgenden Informationen:

note

(Optional) Die Notiz, die der Kunde hinterlässt, wenn er den Alarm bestätigt.

Datentyp: `STRING`

Maximale Länge: 128 Zeichen

snooze

(Optional) Ein Objekt, das vorhanden ist, `customerAction` wenn der Bediener den Alarm in den Schlummermodus versetzt. Wenn ein Bediener den Alarm aktiviert, wechselt der Alarmstatus zu `SnoozeDisabled`. Dieses Objekt enthält die folgenden Informationen:

snoozeDuration

Die Dauer in Sekunden, für die der Bediener den Alarm aktiviert. Nach Ablauf dieser Dauer wechselt der Alarm in `Normal` den Status.

Datentyp: `INTEGER`

note

(Optional) Die Notiz, die der Kunde hinterlässt, wenn er den Alarm aktiviert.

Datentyp: `STRING`

Maximale Länge: 128 Zeichen

ruleEvaluation

(Optional) Ein Objekt, das Informationen über die Regel enthält, die den Alarm auswertet. Dieses Objekt enthält die folgenden Informationen:

simpleRule

Ein Objekt, das Informationen über eine einfache Regel enthält, die einen Eigenschaftswert mit einem Schwellenwert anhand eines Vergleichsoperators vergleicht. Dieses Objekt enthält die folgenden Informationen:

inputProperty

Der Wert der Eigenschaft, die dieser Alarm auswertet.

Datentyp: DOUBLE

operator

Der Vergleichsoperator, den dieser Alarm verwendet, um die Eigenschaft mit dem Schwellenwert zu vergleichen. Dieser Wert enthält eine der folgenden Zeichenketten:

- <— Weniger als
- <=— Weniger als oder gleich
- ==— Gleich
- !=— Nicht gleich
- >=— Größer als oder gleich
- >— Größer als

Datentyp: STRING

threshold

Der Schwellenwert, mit dem dieser Alarm den Eigenschaftswert vergleicht.

Datentyp: DOUBLE

Definieren Sie Alarme für Asset-Modelle in AWS IoT SiteWise

Anlagenmodelle fördern die Standardisierung Ihrer industriellen Daten und Alarme. Sie können Alarmdefinitionen für Anlagenmodelle definieren, um die Alarme für alle Anlagen auf der Grundlage eines Anlagenmodells zu standardisieren.

Sie verwenden zusammengesetzte Asset-Modelle, um Alarme für Asset-Modelle zu definieren. Bei zusammengesetzten Anlagenmodellen handelt es sich um Anlagenmodelle, die einen bestimmten Satz von Eigenschaften auf ein anderes Anlagenmodell standardisieren. Zusammengesetzte Anlagenmodelle stellen sicher, dass bestimmte Eigenschaften in einem Anlagenmodell vorhanden sind. Alarme verfügen über Typ-, Status- und (optionale) Quelleneigenschaften, sodass das zusammengesetzte Alarmmodell erzwingt, dass diese Eigenschaften vorhanden sind.

Jedes zusammengesetzte Objektmodell hat einen Typ, der die Eigenschaften für dieses zusammengesetzte Modell definiert. Verbundmodelle für Alarme definieren Eigenschaften für den Alarmtyp, den Alarmstatus und die (optionale) Alarmquelle. Wenn Sie ein Asset aus einem Asset-Modell mit zusammengesetzten Modellen erstellen, enthält das Asset neben den Eigenschaften, die Sie im Asset-Modell angeben, auch die Eigenschaften aus dem Verbundmodell.

Jede Eigenschaft in einem zusammengesetzten Modell muss den Namen haben, der sie für ihren Typ des zusammengesetzten Modells kennzeichnet. Die Eigenschaften eines zusammengesetzten Modells unterstützen Eigenschaften mit komplexen Datentypen. Diese Eigenschaften haben den STRUCT Datentyp und ein dataTypeSpec Merkmal, das den komplexen Datentyp der Eigenschaft angibt. Eigenschaften komplexer Datentypen enthalten JSON-Daten, die als Zeichenfolgen serialisiert sind.

Verbundmodelle von Alarmen haben die folgenden Eigenschaften. Jede Eigenschaft muss den Namen haben, der sie für diesen Typ von Verbundmodell identifiziert.

Typ des Alarms

Der Typ des Alarms. Geben Sie eines der folgenden Elemente an:

- **IOT_EVENTS**— Ein AWS IoT Events Alarm. AWS IoT SiteWise sendet Daten an, AWS IoT Events um den Status dieses Alarms auszuwerten. Sie müssen die Eigenschaft Alarmquelle angeben, um das AWS IoT Events Alarmmodell für diese Alarmdefinition zu definieren.

Note

Hinweis zum Ende des Supports: Der Support für AWS wurde eingestellt AWS IoT Events. Weitere Informationen finden Sie unter [AWS IoT Events Ende des Supports](#).

- **EXTERNAL**— Ein externer Alarm. Sie nehmen den Zustand des Alarms als Messwert auf.

Name der Immobilie: `AWS/ALARM_TYPE`

Art der Immobilie: [Attribut](#)

Datentyp: STRING

Zustand des Alarms

Die Zeitreihendaten für den Status des Alarms. Dies ist ein als Zeichenfolge serialisiertes Objekt, das den Status und andere Informationen über den Alarm enthält. Weitere Informationen finden Sie unter [Eigenschaften des Alarmstatus](#).

Name der Immobilie: AWS/ALARM_STATE

Art der Immobilie: [Messung](#)

Datentyp: STRUCT

Typ der Datenstruktur: AWS/ALARM_STATE

Quelle des Alarms

(Optional) Der Amazon-Ressourcenname (ARN) der Ressource, die den Status des Alarms auswertet. Für AWS IoT Events Alarme ist dies der ARN des Alarmmodells.

Name der Immobilie: AWS/ALARM_SOURCE

Art der Immobilie: [Attribut](#)

Datentyp: STRING

Example Beispiel für ein zusammengesetztes Alarmmodell

Das folgende Anlagenmodell stellt einen Kessel dar, dessen Temperatur über einen Alarm überwacht wird. AWS IoT SiteWise sendet die Temperaturdaten an, AWS IoT Events um den Alarm zu erkennen.

```
{
  "assetModelName": "Boiler",
  "assetModelDescription": "A boiler that alarms when its temperature exceeds its limit.",
  "assetModelProperties": [
    {
      "name": "Temperature",
      "dataType": "DOUBLE",
      "unit": "Celsius",
      "type": {
        "measurement": {}
      }
    }
  ],
}
```

```

{
  "name": "High Temperature",
  "dataType": "DOUBLE",
  "unit": "Celsius",
  "type": {
    "attribute": {
      "defaultValue": "105.0"
    }
  }
},
],
"assetModelCompositeModels": [
{
  "name": "BoilerTemperatureHighAlarm",
  "type": "AWS/ALARM",
  "properties": [
    {
      "name": "AWS/ALARM_TYPE",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "IOT_EVENTS"
        }
      }
    },
    {
      "name": "AWS/ALARM_STATE",
      "dataType": "STRUCT",
      "dataTypeSpec": "AWS/ALARM_STATE",
      "type": {
        "measurement": {}
      }
    },
    {
      "name": "AWS/ALARM_SOURCE",
      "dataType": "STRING",
      "type": {
        "attribute": {}
      }
    }
  ]
}
]

```

```
}
```

Themen

- [Anforderungen für Alarmmeldungen inAWS IoT SiteWise](#)
- [DefinierenAWS IoT EventsAlarme fürAWS IoT SiteWise](#)
- [Definieren Sie externe Alarme inAWS IoT SiteWise](#)

Anforderungen für Alarmmeldungen inAWS IoT SiteWise

AWS IoT Eventsverwendet eine AWS Lambda Funktion in Ihrem AWS Konto, um Alarmbenachrichtigungen zu senden. Sie müssen diese Lambda-Funktion in derselben AWS Region wie Ihre Alarme erstellen, um Alarmbenachrichtigungen zu aktivieren. Diese Lambda-Funktion verwendet [Amazon Simple Notification Service \(Amazon SNS\)](#) zum Senden von Textbenachrichtigungen und [Amazon Simple Email Service \(Amazon SES\)](#) zum Senden von E-Mail-Benachrichtigungen. Wenn Sie den AWS IoT Events Alarm erstellen, konfigurieren Sie die Protokolle und Einstellungen, die der Alarm zum Senden von Benachrichtigungen verwendet.

Note

Hinweis zum Ende des Supports: Der Support für AWS wurde beendetAWS IoT Events. Weitere Informationen finden Sie unter [AWS IoT EventsEnde des Supports](#).

AWS IoT Eventsstellt eine AWS CloudFormation Stack-Vorlage bereit, mit der Sie diese Lambda-Funktion in Ihrem Konto erstellen können. Weitere Informationen finden Sie unter [Lambda-Funktion für Alarmbenachrichtigungen](#) im AWS IoT EventsEntwicklerhandbuch.

DefinierenAWS IoT EventsAlarme fürAWS IoT SiteWise

Note

Hinweis zum Ende des Supports: Der Support für AWS wurde beendetAWS IoT Events. Weitere Informationen finden Sie unter [AWS IoT EventsEnde des Supports](#).

Wenn Sie einen AWS IoT Events Alarm erstellen, AWS IoT SiteWise sendet die Eigenschaftswerte der Anlage anAWS IoT Events, um den Status des Alarms auszuwerten. AWS IoT

EventsAlarmdefinitionen hängen von einem Alarmmodell ab, in dem Sie sie definierenAWS IoT Events. Um einen AWS IoT Events Alarm anhand eines Anlagenmodells zu definieren, definieren Sie ein zusammengesetztes Alarmmodell, das das AWS IoT Events Alarmmodell als Alarmquelleneigenschaft angibt.

AWS IoT EventsAlarme hängen von Eingaben wie Alarmschwellenwerten und Einstellungen für Alarmbenachrichtigungen ab. Sie definieren diese Eingaben als Attribute im Asset-Modell. Sie können diese Eingaben dann für jedes Asset auf der Grundlage des Modells anpassen. Die AWS IoT SiteWise Konsole kann diese Attribute für Sie erstellen. Wenn Sie Alarme mit der API AWS CLI oder definieren, müssen Sie diese Attribute im Asset-Modell manuell definieren.

Sie können auch andere Aktionen definieren, die ausgeführt werden, wenn Ihr Alarm erkannt wird, z. B. benutzerdefinierte Aktionen für Alarmbenachrichtigungen. Sie können beispielsweise eine Aktion konfigurieren, die eine Push-Benachrichtigung an ein Amazon SNS SNS-Thema sendet. Weitere Informationen zu den Aktionen, die Sie definieren können, finden Sie unter [Arbeiten mit anderen AWS Diensten](#) im AWS IoT EventsEntwicklerhandbuch.

Wenn Sie ein Asset-Modell aktualisieren oder löschen, AWS IoT SiteWise kann überprüft werden, ob ein Alarmmodell eine mit diesem Asset-Modell verknüpfte Anlageneigenschaft überwacht. AWS IoT Events Dadurch wird verhindert, dass Sie eine Anlageneigenschaft löschen, die derzeit von einem AWS IoT Events Alarm verwendet wird. Um diese Funktion in zu aktivierenAWS IoT SiteWise, benötigen Sie die `iotevents:ListInputRoutings` entsprechende Genehmigung. Diese Berechtigung AWS IoT SiteWise ermöglicht Aufrufe des [ListInputRoutings](#)API-Vorgangs, der von unterstützt wirdAWS IoT Events. Weitere Informationen finden Sie unter [\(Fakultativ\) ListInputRoutings Erlaubnis](#).

Note

Die Funktion für Alarmbenachrichtigungen ist in der Region China (Peking) nicht verfügbar.

Themen

- [Definieren Sie eineAWS IoT EventsAlarm \(AWS IoT SiteWiseKonsole\)](#)
- [Definieren Sie eineAWS IoT EventsAlarm \(AWS IoT EventsKonsole\)](#)
- [Definieren Sie eineAWS IoT EventsAlarm \(AWS CLI\)](#)

Definieren Sie eine AWS IoT Events Alarm (AWS IoT SiteWise Konsole)

Sie können die AWS IoT SiteWise Konsole verwenden, um einen AWS IoT Events Alarm für ein vorhandenes Anlagenmodell zu definieren. Um einen AWS IoT Events Alarm für ein neues Asset-Modell zu definieren, erstellen Sie das Asset-Modell und führen Sie dann diese Schritte aus. Weitere Informationen finden Sie unter [Erstellen Sie Asset-Modelle in AWS IoT SiteWise](#).

Important

Für jeden Alarm ist ein Attribut erforderlich, das den Schwellenwert angibt, mit dem für den Alarm verglichen werden soll. Sie müssen das Schwellenwertattribut im Asset-Modell definieren, bevor Sie einen Alarm definieren können.

Stellen Sie sich ein Beispiel vor, bei dem Sie einen Alarm definieren möchten, der erkennt, wenn eine Windkraftanlage ihre maximale Nennwindgeschwindigkeit von 50 mph überschreitet. Bevor Sie den Alarm definieren, müssen Sie ein Attribut (Maximale Windgeschwindigkeit) mit dem Standardwert definieren 50.

Um eine zu definieren AWS IoT Events Alarm für ein Asset-Modell

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Klicken Sie im Navigationsbereich auf Models (Modelle).
3. Wählen Sie das Asset-Modell aus, für das Sie einen Alarm definieren möchten.
4. Wählen Sie die Registerkarte Alarm.
5. Wählen Sie Alarm hinzufügen.
6. Wählen Sie im Bereich Optionen für den Alarmtyp die Option AWS IoT Events Alarm aus.
7. Gehen Sie im Abschnitt Alarmdetails wie folgt vor:
 - a. Geben Sie einen Namen für den Alarm ein.
 - b. (Optional) Geben Sie eine Beschreibung für Ihren Alarm ein.
8. Im Abschnitt Schwellenwertdefinitionen legen Sie fest, wann der Alarm erkannt wird und wie schwerwiegend der Alarm ist. Gehen Sie wie folgt vor:
 - a. Wählen Sie die Eigenschaft aus, bei der der Alarm erkannt wird. Jedes Mal, wenn diese Eigenschaft einen neuen Wert erhält, wird der Wert AWS IoT SiteWise an gesendet, AWS IoT Events um den Status des Alarms auszuwerten.

- b. Wählen Sie den Operator aus, der verwendet werden soll, um die Eigenschaft mit dem Schwellenwert zu vergleichen. Wählen Sie aus den folgenden Optionen aus:
 - < weniger als
 - <= kleiner als oder gleich
 - == gleich
 - != nicht gleich
 - >= größer als oder gleich
 - > größer als
 - c. Wählen Sie unter Wert die Attributeigenschaft aus, die als Schwellenwert verwendet werden soll. AWS IoT Events vergleicht den Wert der Eigenschaft mit dem Wert dieses Attributs.
 - d. Geben Sie den Schweregrad des Alarms ein. Verwenden Sie eine Zahl, die Ihr Team versteht, um den Schweregrad dieses Alarms wiederzugeben.
9. (Optional) Gehen Sie im Abschnitt Benachrichtigungseinstellungen — optional wie folgt vor:
 - a. Wählen Sie Aktiv.

 Note

Wenn Sie Inaktiv wählen, erhalten Sie und Ihr Team keine Alarmbenachrichtigungen.

- b. Wählen Sie unter Empfänger den Empfänger aus.

 Important

Sie können Alarmbenachrichtigungen an AWS IAM Identity Center Benutzer senden. Um diese Funktion nutzen zu können, müssen Sie IAM Identity Center aktivieren. Sie können IAM Identity Center jeweils nur in einer AWS Region aktivieren. Das bedeutet, dass Sie Alarmbenachrichtigungen nur in der Region definieren können, in der Sie IAM Identity Center aktivieren. Weitere Informationen finden Sie unter [Erste Schritte](#) im AWS IAM Identity Center-Benutzerhandbuch.

- c. Wählen Sie unter Protokoll eine der folgenden Optionen aus:
 - E-Mail und Text — Der Alarm benachrichtigt IAM Identity Center-Benutzer mit einer SMS-Nachricht und einer E-Mail-Nachricht.

- E-Mail — Der Alarm benachrichtigt IAM Identity Center-Benutzer mit einer E-Mail-Nachricht.
 - Text — Der Alarm benachrichtigt IAM Identity Center-Benutzer mit einer SMS-Nachricht.
- d. Wählen Sie unter Absender den Absender aus.

 Important

Sie müssen die Absender-E-Mail-Adresse in Amazon Simple Email Service (Amazon SES) verifizieren. Weitere Informationen finden Sie unter [Verifizieren der Identität einer E-Mail-Adresse](#) im Amazon Simple Email Service Developer Guide.

10. Im Abschnitt Standard-Asset-Status können Sie den Standardstatus für Alarme festlegen, die mit diesem Asset-Modell erstellt wurden.

 Note

Sie aktivieren oder deaktivieren diesen Alarm für Assets, die Sie in einem späteren Schritt anhand dieses Asset-Modells erstellen.

11. Im Bereich Erweiterte Einstellungen können Sie die Berechtigungen, die zusätzlichen Benachrichtigungseinstellungen, die Alarmstatusaktionen, das Alarmmodell in SiteWise Monitor und den Bestätigungsablauf konfigurieren.

 Note

AWS IoT Events Für Alarme sind die folgenden Servicerollen erforderlich:

- Eine Rolle, die AWS IoT Events davon ausgeht, Alarmstatuswerte an zu senden AWS IoT SiteWise.
- Eine Rolle, die AWS IoT Events davon ausgeht, Daten an Lambda zu senden. Sie benötigen diese Rolle nur, wenn Ihr Alarm Benachrichtigungen sendet.

Gehen Sie im Abschnitt Berechtigungen wie folgt vor:

- a. Verwenden Sie als AWS IoT Events Rolle eine vorhandene Rolle oder erstellen Sie eine Rolle mit den erforderlichen Berechtigungen. Diese Rolle erfordert die `iotsitewise:BatchPutAssetPropertyValue` Erlaubnis und eine

Vertrauensbeziehung, die es `iotevents.amazonaws.com` ermöglicht, die Rolle zu übernehmen.

- b. Verwenden Sie für die AWS IoT EventsLambda-Rolle eine vorhandene Rolle oder erstellen Sie eine Rolle mit den erforderlichen Berechtigungen. Für diese Rolle sind die `sso-directory:DescribeUser` Berechtigungen `lambda:InvokeFunction` und sowie eine Vertrauensbeziehung erforderlich, die es ermöglicht, die Rolle `iotevents.amazonaws.com` zu übernehmen.

12. (Optional) Gehen Sie im Abschnitt **Zusätzliche Benachrichtigungseinstellungen** wie folgt vor:

- a. Für das Empfängerattribut definieren Sie ein Attribut, dessen Wert den Empfänger der Benachrichtigung angibt. Sie können IAM Identity Center-Benutzer als Empfänger auswählen.

Sie können ein Attribut erstellen oder ein vorhandenes Attribut im Asset-Modell verwenden.

- Wenn Sie Neues Empfängerattribut erstellen wählen, geben Sie den Namen des Empfängerattributs und den Standardwert Empfänger an — optional für das Attribut.
- Wenn Sie Bestehendes Empfängerattribut verwenden wählen, wählen Sie das Attribut im Feld Name des Empfängerattributs aus. Der Alarm verwendet den Standardwert des von Ihnen ausgewählten Attributs.

Sie können den Standardwert für jedes Asset, das Sie anhand dieses Asset-Modells erstellen, überschreiben.

- b. Für das benutzerdefinierte Nachrichtenattribut definieren Sie ein Attribut, dessen Wert die benutzerdefinierte Nachricht angibt, die zusätzlich zur Standardnachricht zur Statusänderung gesendet werden soll. Sie können beispielsweise eine Nachricht angeben, die Ihrem Team hilft, zu verstehen, wie mit diesem Alarm umgegangen werden kann.

Sie können wählen, ob Sie ein Attribut erstellen oder ein vorhandenes Attribut im Asset-Modell verwenden möchten.

- Wenn Sie ein neues benutzerdefiniertes Nachrichtenattribut erstellen möchten, geben Sie den Namen des benutzerdefinierten Nachrichtenattributs und den Standardwert für benutzerdefinierte Nachricht an — optional für das Attribut.
- Wenn Sie Ein vorhandenes benutzerdefiniertes Nachrichtenattribut verwenden wählen, wählen Sie das Attribut unter Name des benutzerdefinierten Nachrichtenattributs aus. Der Alarm verwendet den Standardwert des von Ihnen ausgewählten Attributs.

Sie können den Standardwert für jedes Asset, das Sie anhand dieses Asset-Modells erstellen, überschreiben.

- c. Führen Sie für Manage your Lambda function einen der folgenden Schritte aus:
 - Um eine neue Lambda-Funktion AWS IoT SiteWise erstellen zu lassen, wählen Sie Create a new lambda from an AWS managed template.
 - Um eine bestehende Lambda-Funktion zu verwenden, wählen Sie Use an existing lambda und wählen Sie den Namen der Funktion.

Weitere Informationen finden Sie unter [Verwaltung von Alarmbenachrichtigungen](#) im AWS IoT EventsEntwicklerhandbuch.

13. (Optional) Gehen Sie im Abschnitt Statusaktion festlegen wie folgt vor:

- a. Wählen Sie Aktion bearbeiten aus.
- b. Fügen Sie unter Aktionen zum Alarmstatus hinzufügen die Aktionen hinzu und wählen Sie dann Speichern aus.

Sie können bis zu 10 Aktionen hinzufügen.

AWS IoT Events kann Aktionen ausführen, wenn der Alarm aktiv ist. Sie können integrierte Aktionen definieren, um einen Timer zu verwenden oder eine Variable festzulegen oder Daten an andere AWS Ressourcen zu senden. Weitere Informationen finden Sie im AWS IoT EventsEntwicklerhandbuch unter [Unterstützte Aktionen](#).

14. (Optional) Wählen Sie unter Alarmmodell im SiteWise Monitor verwalten — optional die Option Aktiv oder Inaktiv aus.

Verwenden Sie diese Option, damit Sie das Alarmmodell in SiteWise Monitoren aktualisieren können. Diese Option ist standardmäßig aktiviert.

15. Wählen Sie unter Acknowledge-Flow die Option Aktiv oder Inaktiv aus. Weitere Informationen zum Bestätigungsablauf finden Sie unter [Alarmzustände](#).

16. Wählen Sie „Alarm hinzufügen“.

 Note

Die AWS IoT SiteWise Konsole stellt mehrere API-Anfragen, um den Alarm zum Asset-Modell hinzuzufügen. Wenn Sie Alarm hinzufügen wählen, öffnet die Konsole ein Dialogfeld, in dem der Status dieser API-Anfragen angezeigt wird. Bleiben Sie auf dieser Seite, bis jede API-Anfrage erfolgreich ist oder bis eine API-Anfrage fehlschlägt. Wenn eine Anfrage fehlschlägt, schließen Sie das Dialogfeld, beheben Sie das Problem und wählen Sie Alarm hinzufügen, um es erneut zu versuchen.

Definieren Sie eine AWS IoT Events Alarm (AWS IoT Events Konsole)

Sie können die AWS IoT Events Konsole verwenden, um einen AWS IoT Events Alarm für ein vorhandenes Anlagenmodell zu definieren. Um einen AWS IoT Events Alarm für ein neues Asset-Modell zu definieren, erstellen Sie das Asset-Modell und führen Sie dann diese Schritte aus. Weitere Informationen finden Sie unter [Erstellen Sie Asset-Modelle in AWS IoT SiteWise](#).

 Important

Für jeden Alarm ist ein Attribut erforderlich, das den Schwellenwert angibt, mit dem für den Alarm verglichen werden soll. Sie müssen das Schwellenwertattribut im Asset-Modell definieren, bevor Sie einen Alarm definieren können.

Stellen Sie sich ein Beispiel vor, bei dem Sie einen Alarm definieren möchten, der erkennt, wenn eine Windkraftanlage ihre maximale Nennwindgeschwindigkeit von 50 mph überschreitet. Bevor Sie den Alarm definieren, müssen Sie ein Attribut (Maximale Windgeschwindigkeit) mit dem Standardwert definieren 50.

Um eine zu definieren AWS IoT Events Alarm für ein Asset-Modell

1. Navigieren Sie zur [AWS IoT Events-Konsole](#).
2. Wählen Sie im Navigationsbereich die Option Alarmmodelle aus.
3. Wählen Sie Alarmmodell erstellen aus.
4. Geben Sie einen Namen für den Alarm ein.
5. (Optional) Geben Sie eine Beschreibung für Ihren Alarm ein.
6. Gehen Sie im Bereich Alarmziel wie folgt vor:

- a. Wählen Sie unter Zieloptionen die Option AWS IoT SiteWiseAsset-Eigenschaft aus.
 - b. Wählen Sie das Asset-Modell aus, für das Sie den Alarm hinzufügen möchten.
7. Im Abschnitt Schwellenwertdefinitionen legen Sie fest, wann der Alarm erkannt wird und wie schwerwiegend der Alarm ist. Gehen Sie wie folgt vor:
 - a. Wählen Sie die Eigenschaft aus, bei der der Alarm erkannt wird. Jedes Mal, wenn diese Eigenschaft einen neuen Wert erhält, wird der Wert AWS IoT SiteWise an gesendet, AWS IoT Events um den Status des Alarms auszuwerten.
 - b. Wählen Sie den Operator aus, der verwendet werden soll, um die Eigenschaft mit dem Schwellenwert zu vergleichen. Wählen Sie aus den folgenden Optionen aus:
 - < weniger als
 - <= kleiner als oder gleich
 - == gleich
 - != nicht gleich
 - >= größer als oder gleich
 - > größer als
 - c. Wählen Sie unter Wert die Attributeigenschaft aus, die als Schwellenwert verwendet werden soll. AWS IoT Events vergleicht den Wert der Eigenschaft mit dem Wert dieses Attributs.
 - d. Geben Sie den Schweregrad des Alarms ein. Verwenden Sie eine Zahl, die Ihr Team versteht, um den Schweregrad dieses Alarms wiederzugeben.
8. (Optional) Gehen Sie im Abschnitt Benachrichtigungseinstellungen — optional wie folgt vor:
 - a. Wählen Sie unter Protokoll eine der folgenden Optionen aus:
 - E-Mail und Text — Der Alarm benachrichtigt IAM Identity Center-Benutzer mit einer SMS-Nachricht und einer E-Mail-Nachricht.
 - E-Mail — Der Alarm benachrichtigt IAM Identity Center-Benutzer mit einer E-Mail-Nachricht.
 - Text — Der Alarm benachrichtigt IAM Identity Center-Benutzer mit einer SMS-Nachricht.
 - b. Wählen Sie unter Absender den Absender aus.

 Important

Sie müssen die Absender-E-Mail-Adresse in Amazon Simple Email Service (Amazon SES) verifizieren. Weitere Informationen finden Sie unter [Verifizieren von E-Mail-Adressen in Amazon SES](#) im Amazon Simple Email Service Developer Guide.

- c. Wählen Sie das Attribut unter Empfängerattribut — optional aus. Der Alarm verwendet den Standardwert des von Ihnen ausgewählten Attributs.
 - d. Wählen Sie das Attribut unter Benutzerdefiniertes Nachrichtenattribut — optional aus. Der Alarm verwendet den Standardwert des von Ihnen ausgewählten Attributs.
9. Geben Sie im Abschnitt Instanz den Standardstatus für diesen Alarm an. Sie können diesen Alarm in einem späteren Schritt für alle Assets aktivieren oder deaktivieren, die Sie anhand dieses Asset-Modells erstellen.
10. In den erweiterten Einstellungen können Sie die Berechtigungen, die zusätzlichen Benachrichtigungseinstellungen, die Alarmstatusaktionen, das Alarmmodell in SiteWise Monitor und den Bestätigungsablauf konfigurieren.

 Note

AWS IoT EventsFür Alarme sind die folgenden Servicerollen erforderlich:

- Eine Rolle, die AWS IoT Events davon ausgeht, Alarmstatuswerte an zu sendenAWS IoT SiteWise.
- Eine Rolle, die AWS IoT Events davon ausgeht, Daten an Lambda zu senden. Sie benötigen diese Rolle nur, wenn Ihr Alarm Benachrichtigungen sendet.

- a. Wählen Sie im Abschnitt Bestätigungsablauf die Option Aktiviert oder Deaktiviert aus. Weitere Informationen zum Bestätigungsablauf finden Sie unter[Alarmzustände](#).
- b. Gehen Sie im Abschnitt „Berechtigungen“ wie folgt vor:
 - i. Verwenden Sie als AWS IoT EventsRolle eine vorhandene Rolle oder erstellen Sie eine Rolle mit den erforderlichen Berechtigungen. Diese Rolle erfordert die `iotsitewise:BatchPutAssetPropertyValue` Erlaubnis und eine

Vertrauensbeziehung, die es `iotevents.amazonaws.com` ermöglicht, die Rolle zu übernehmen.

- ii. Verwenden Sie für die Lambda-Rolle eine vorhandene Rolle oder erstellen Sie eine Rolle mit den erforderlichen Berechtigungen. Für diese Rolle sind die `sso-directory:DescribeUser` Berechtigungen `lambda:InvokeFunction` und sowie eine Vertrauensbeziehung erforderlich, die es ermöglicht, die Rolle `iotevents.amazonaws.com` zu übernehmen.
- c. (Optional) Gehen Sie im Bereich Zusätzliche Benachrichtigungseinstellungen wie folgt vor:
 - Führen Sie für Manage your Lambda function einen der folgenden Schritte aus:
 - Um eine neue Lambda-Funktion AWS IoT Events erstellen zu lassen, wählen Sie Create a new Lambda-Funktion.
 - Um eine bestehende Lambda-Funktion zu verwenden, wählen Sie Bestehende Lambda-Funktion verwenden und wählen Sie den Namen der Funktion.

Weitere Informationen finden Sie unter [Verwaltung von Alarmbenachrichtigungen](#) im AWS IoT EventsEntwicklerhandbuch.

- d. (Optional) Gehen Sie im Abschnitt Statusaktion festlegen — optional wie folgt vor:
 - Fügen Sie unter Aktionen zum Alarmstatus die Aktionen hinzu und wählen Sie dann Speichern aus.

Sie können bis zu 10 Aktionen hinzufügen.

AWS IoT Events kann Aktionen ausführen, wenn der Alarm aktiv ist. Sie können integrierte Aktionen definieren, um einen Timer zu verwenden oder eine Variable festzulegen oder Daten an andere AWS Ressourcen zu senden. Weitere Informationen finden Sie im AWS IoT EventsEntwicklerhandbuch unter [Unterstützte Aktionen](#).

11. Wählen Sie Erstellen aus.

Note

Die AWS IoT Events Konsole stellt mehrere API-Anfragen, um den Alarm zum Asset-Modell hinzuzufügen. Wenn Sie Alarm hinzufügen wählen, öffnet die Konsole ein Dialogfeld, in dem der Status dieser API-Anfragen angezeigt wird. Bleiben Sie auf dieser

Seite, bis jede API-Anfrage erfolgreich ist oder bis eine API-Anfrage fehlschlägt. Wenn eine Anfrage fehlschlägt, schließen Sie das Dialogfeld, beheben Sie das Problem und wählen Sie Alarm hinzufügen, um es erneut zu versuchen.

Definieren Sie eine AWS IoT Events Alarm (AWS CLI)

Sie können das AWS Command Line Interface (AWS CLI) verwenden, um einen AWS IoT Events Alarm zu definieren, der eine Anlageneigenschaft überwacht. Sie können den Alarm für ein neues oder ein vorhandenes Asset-Modell definieren. Nachdem Sie den Alarm für das Asset-Modell definiert haben, erstellen Sie einen Alarm im Asset-Modell AWS IoT Events und verbinden ihn mit dem Asset-Modell. In diesem Prozess gehen Sie wie folgt vor:

Schritte

- [Schritt 1: Definieren Sie einen Alarm für ein Asset-Modell](#)
- [Schritt 2: Definieren Sie eine AWS IoT Events Alarm-Modell](#)
- [Schritt 3: Aktivieren Sie den Datenfluss zwischen AWS IoT SiteWise und AWS IoT Events](#)

Schritt 1: Definieren Sie einen Alarm für ein Asset-Modell

Fügen Sie eine Alarmdefinition und zugehörige Eigenschaften zu einem neuen oder vorhandenen Anlagenmodell hinzu.

So definieren Sie einen Alarm für ein Asset-Modell (CLI)

1. Erstellen Sie eine Datei mit dem Namen `asset-model-payload.json`. Folgen Sie den Schritten in diesen anderen Abschnitten, um der Datei die Details Ihres Asset-Modells hinzuzufügen, reichen Sie jedoch nicht die Anfrage zur Erstellung oder Aktualisierung des Asset-Modells ein. In diesem Abschnitt fügen Sie den Asset-Modelldetails in der `asset-model-payload.json` Datei eine Alarmdefinition hinzu.
 - Weitere Informationen zum Erstellen eines Asset-Modells finden Sie unter [Erstellen Sie ein Asset-Modell \(AWS CLI\)](#).
 - Weitere Informationen zum Aktualisieren eines vorhandenen Asset-Modells finden Sie unter [Aktualisieren Sie ein Objektmodell, ein Komponentenmodell oder eine Schnittstelle \(AWS CLI\)](#).

 Note

Ihr Anlagenmodell muss mindestens eine Anlageneigenschaft definieren, einschließlich der Anlageneigenschaft, die mit dem Alarm überwacht werden soll.

2. Fügen Sie dem Anlagenmodell ein zusammengesetztes Alarmmodell (`assetModelCompositeModels`) hinzu. Ein zusammengesetztes AWS IoT Events Alarmmodell spezifiziert den `IOT_EVENTS` Typ und gibt eine Alarmquelleneigenschaft an. Sie fügen die Eigenschaft `Alarmquelle` hinzu, nachdem Sie das Alarmmodell in erstellt haben AWS IoT Events.

 Important

Das zusammengesetzte Alarmmodell muss denselben Namen haben wie das AWS IoT Events Alarmmodell, das Sie später erstellen. Namen von Alarmmodellen dürfen nur alphanumerische Zeichen enthalten. Geben Sie einen eindeutigen, alphanumerischen Namen an, sodass Sie denselben Namen für das Alarmmodell verwenden können.

```
{
  ...
  "assetModelCompositeModels": [
    {
      "name": "BoilerTemperatureHighAlarm",
      "type": "AWS/ALARM",
      "properties": [
        {
          "name": "AWS/ALARM_TYPE",
          "dataType": "STRING",
          "type": {
            "attribute": {
              "defaultValue": "IOT_EVENTS"
            }
          }
        },
        {
          "name": "AWS/ALARM_STATE",
          "dataType": "STRUCT",
          "dataTypeSpec": "AWS/ALARM_STATE",

```

```

        "type": {
            "measurement": {}
        }
    }
]
}

```

3. Fügen Sie dem Asset-Modell ein Alarmschwellenwertattribut hinzu. Geben Sie den Standardwert an, der für diesen Schwellenwert verwendet werden soll. Sie können diesen Standardwert für jedes Asset, das auf diesem Modell basiert, überschreiben.

 Note

Das Alarmschwellenwertattribut muss ein INTEGER oder a seinDOUBLE.

```

{
    ...
    "assetModelProperties": [
        ...
        {
            "name": "Temperature Max Threshold",
            "dataType": "DOUBLE",
            "type": {
                "attribute": {
                    "defaultValue": "105.0"
                }
            }
        }
    ]
}

```

4. (Optional) Fügen Sie dem Asset-Modell Attribute für Alarmbenachrichtigungen hinzu. Diese Attribute geben den IAM Identity Center-Empfänger und andere Eingaben an, die zum Senden von Benachrichtigungen AWS IoT Events verwendet werden, wenn sich der Status des Alarms ändert. Sie können diese Standardeinstellungen für jedes Asset, das auf diesem Modell basiert, überschreiben.

⚠ Important

Sie können Alarmbenachrichtigungen an AWS IAM Identity Center Benutzer senden. Um diese Funktion nutzen zu können, müssen Sie IAM Identity Center aktivieren. Sie können IAM Identity Center jeweils nur in einer AWS Region aktivieren. Das bedeutet, dass Sie Alarmbenachrichtigungen nur in der Region definieren können, in der Sie IAM Identity Center aktivieren. Weitere Informationen finden Sie unter [Erste Schritte](#) im AWS IAM Identity Center-Benutzerhandbuch.

Gehen Sie wie folgt vor:

- a. Fügen Sie ein Attribut hinzu, das die ID Ihres IAM Identity Center-Identitätsspeichers angibt. Sie können den IAM Identity Center [ListInstances](#) API-Vorgang verwenden, um Ihre Identitätsspeicher aufzulisten. Dieser Vorgang funktioniert nur in der Region, in der Sie IAM Identity Center aktivieren.

```
aws sso-admin list-instances
```

Geben Sie dann die Identitätsspeicher-ID (z. B. d-123EXAMPLE) als Standardwert für das Attribut an.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "identityStoreId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "d-123EXAMPLE"
        }
      }
    }
  ]
}
```

- b. Fügen Sie ein Attribut hinzu, das die ID des IAM Identity Center-Benutzers angibt, der Benachrichtigungen erhält. Um einen Standardempfänger für Benachrichtigungen zu definieren, fügen Sie eine IAM Identity Center-Benutzer-ID als Standardwert hinzu. Gehen Sie wie folgt vor, um eine IAM Identity Center-Benutzer-ID zu erhalten:
 - i. Sie können die IAM Identity [ListUsers](#) Center-API verwenden, um die ID eines Benutzers abzurufen, dessen Benutzernamen Sie kennen. *d-123EXAMPLE* Ersetzen Sie sie durch die ID Ihres Identitätsspeichers und *Name* ersetzen Sie sie durch den Benutzernamen des Benutzers.

```
aws identitystore list-users \
  --identity-store-id d-123EXAMPLE \
  --filters AttributePath=UserName,AttributeValue=Name
```

- ii. Verwenden Sie die [IAM Identity Center-Konsole](#), um Ihre Benutzer zu durchsuchen und eine Benutzer-ID zu finden.

Geben Sie dann die Benutzer-ID (z. B. *123EXAMPLE-a1b2c3d4-5678-90ab-cdef-3333EXAMPLE*) als Standardwert für das Attribut an, oder definieren Sie das Attribut ohne Standardwert.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "userId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "123EXAMPLE-a1b2c3d4-5678-90ab-cdef-3333EXAMPLE"
        }
      }
    }
  ]
}
```

- c. (Optional) Fügen Sie ein Attribut hinzu, das die Standard-Absender-ID für SMS-Benachrichtigungen (Text) angibt. Die Absender-ID wird in Nachrichten, die Amazon Simple Notification Service (Amazon SNS) sendet, als Nachrichtenabsender angezeigt. Weitere

Informationen finden Sie [AWS End User Messaging SMSim AWS End User Messaging SMSBenutzerhandbuch unter Eine Absender-ID anfordern.](#)

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "senderId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "MyFactory"
        }
      }
    }
  ]
}
```

- d. (Optional) Fügen Sie ein Attribut hinzu, das die Standard-E-Mail-Adresse angibt, die als Absenderadresse in E-Mail-Benachrichtigungen verwendet werden soll.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "fromAddress",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "my.factory@example.com"
        }
      }
    }
  ]
}
```

- e. (Optional) Fügen Sie ein Attribut hinzu, das den Standard-Betreff angibt, der in E-Mail-Benachrichtigungen verwendet werden soll.

```
{
```

```

...
"assetModelProperties": [
  ...
  {
    "name": "emailSubject",
    "dataType": "STRING",
    "type": {
      "attribute": {
        "defaultValue": "[ALERT] High boiler temperature"
      }
    }
  }
]
}

```

- f. (Optional) Fügen Sie ein Attribut hinzu, das eine zusätzliche Nachricht angibt, die in Benachrichtigungen aufgenommen werden soll. Standardmäßig enthalten Benachrichtigungen Informationen über den Alarm. Sie können auch eine zusätzliche Nachricht hinzufügen, die dem Benutzer weitere Informationen gibt.

```

{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "additionalMessage",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "Turn off the power before you check the alarm."
        }
      }
    }
  ]
}

```

5. Erstellen Sie das Asset-Modell oder aktualisieren Sie das bestehende Asset-Modell. Führen Sie eine der folgenden Aktionen aus:
 - Führen Sie den folgenden Befehl aus, um das Asset-Modell zu erstellen.

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

- Führen Sie den folgenden Befehl aus, um das vorhandene Asset-Modell zu aktualisieren.
asset-model-id Ersetzen Sie es durch die ID des Asset-Modells.

```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --cli-input-json file://asset-model-payload.json
```

Nachdem Sie den Befehl ausgeführt haben, notieren Sie sich das `assetModelId` in der Antwort.

Beispiel: Modell der Kesselanlage

Das folgende Anlagenmodell stellt einen Kessel dar, der Temperaturdaten meldet. Dieses Anlagenmodell definiert einen Alarm, der erkennt, wenn der Kessel überhitzt.

```
{  
  "assetModelName": "Boiler Model",  
  "assetModelDescription": "Represents a boiler.",  
  "assetModelProperties": [  
    {  
      "name": "Temperature",  
      "dataType": "DOUBLE",  
      "unit": "C",  
      "type": {  
        "measurement": {}  
      }  
    },  
    {  
      "name": "Temperature Max Threshold",  
      "dataType": "DOUBLE",  
      "type": {  
        "attribute": {  
          "defaultValue": "105.0"  
        }  
      }  
    }  
  ],  
  {  
    "name": "identityStoreId",  
    "dataType": "STRING",  
    "type": {  
      "attribute": {
```

```

        "defaultValue": "d-123EXAMPLE"
    }
},
{
    "name": "userId",
    "dataType": "STRING",
    "type": {
        "attribute": {
            "defaultValue": "123EXAMPLE-a1b2c3d4-5678-90ab-cdef-33333EXAMPLE"
        }
    }
},
{
    "name": "senderId",
    "dataType": "STRING",
    "type": {
        "attribute": {
            "defaultValue": "MyFactory"
        }
    }
},
{
    "name": "fromAddress",
    "dataType": "STRING",
    "type": {
        "attribute": {
            "defaultValue": "my.factory@example.com"
        }
    }
},
{
    "name": "emailSubject",
    "dataType": "STRING",
    "type": {
        "attribute": {
            "defaultValue": "[ALERT] High boiler temperature"
        }
    }
},
{
    "name": "additionalMessage",
    "dataType": "STRING",
    "type": {

```



```

        "attribute": {
            "defaultValue": "Turn off the power before you check the alarm."
        }
    },
],
"assetModelHierarchies": [

],
"assetModelCompositeModels": [
{
    "name": "BoilerTemperatureHighAlarm",
    "type": "AWS/ALARM",
    "properties": [
        {
            "name": "AWS/ALARM_TYPE",
            "dataType": "STRING",
            "type": {
                "attribute": {
                    "defaultValue": "IOT_EVENTS"
                }
            }
        },
        {
            "name": "AWS/ALARM_STATE",
            "dataType": "STRUCT",
            "dataTypeSpec": "AWS/ALARM_STATE",
            "type": {
                "measurement": {}
            }
        }
    ]
}
]
}

```

Schritt 2: Definieren Sie eine AWS IoT Events Alarm-Modell

Erstellen Sie das Alarmmodell in AWS IoT Events. In AWS IoT Events verwenden Sie Ausdrücke, um Werte in Alarmmodellen anzugeben. Sie können Ausdrücke verwenden, um Werte anzugeben AWS IoT SiteWise, die ausgewertet und als Eingaben für den Alarm verwendet werden sollen. Wenn die Eigenschaftswerte einer Anlage AWS IoT SiteWise an das Alarmmodell sendet, AWS IoT Events

wertet sie den Ausdruck aus, um den Wert der Eigenschaft oder die ID der Anlage zu ermitteln. Sie können die folgenden Ausdrücke im Alarmmodell verwenden:

- Werte der Eigenschaften von Vermögenswerten

Verwenden Sie den folgenden Ausdruck, um den Wert einer Anlageneigenschaft zu ermitteln.

assetModelId Ersetzen Sie ihn durch die ID des Asset-Modells und *propertyId* ersetzen Sie ihn durch die ID der Eigenschaft.

```
$sitewise.assetModel.`assetModelId`.`propertyId`.propertyValue.value
```

- Objekt-IDs

Verwenden Sie den folgenden Ausdruck, um die ID des Assets abzurufen.

assetModelId Ersetzen Sie es durch die ID des Asset-Modells und *propertyId* ersetzen Sie es durch die ID der Eigenschaft.

```
$sitewise.assetModel.`assetModelId`.`propertyId`.assetId
```

Note

Wenn Sie das Alarmmodell erstellen, können Sie Literale anstelle von Ausdrücken definieren, die zu AWS IoT SiteWise Werten ausgewertet werden. Dadurch kann die Anzahl der Attribute, die Sie in Ihrem Asset-Modell definieren, reduziert werden. Wenn Sie jedoch einen Wert als Literalwert definieren, können Sie diesen Wert nicht für Anlagen anpassen, die auf dem Anlagemodell basieren. Ihre AWS IoT SiteWise Monitor Benutzer können den Alarm auch nicht anpassen, da sie Alarmeinstellungen nur für Assets konfigurieren können.

Um eine zu erstellenAWS IoT EventsAlarmmodell (CLI)

1. Wenn Sie das Alarmmodell in erstellenAWS IoT Events, müssen Sie die ID jeder Eigenschaft angeben, die der Alarm verwendet. Dazu gehören:
 - Die Eigenschaft „Alarmstatus“ im zusammengesetzten Objektmodell
 - Die Eigenschaft, die der Alarm überwacht
 - Das Schwellenwertattribut
 - (Optional) Das ID-Attribut für den Identitätsspeicher von IAM Identity Center

- (Optional) Das IAM Identity Center-Benutzer-ID-Attribut
- (Optional) Das SMS-Absender-ID-Attribut
- (Optional) Das E-Mail-Absender-Adressattribut
- (Optional) Das E-Mail-Betreff-Attribut
- (Optional) Das zusätzliche Nachrichtenattribut

Führen Sie den folgenden Befehl aus, um die IDs dieser Eigenschaften im Asset-Modell abzurufen. *asset-model-id* Ersetzen Sie es durch die ID des Asset-Modells aus dem vorherigen Schritt.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

Die Operation gibt eine Antwort zurück, die Details des Komponentenmodells enthält. Notieren Sie sich die ID jeder Eigenschaft, die der Alarm verwendet. Sie verwenden diese IDs, wenn Sie das AWS IoT Events Alarmmodell im nächsten Schritt erstellen.

2. Erstellen Sie das Alarmmodell in AWS IoT Events. Gehen Sie wie folgt vor:
 - a. Erstellen Sie eine Datei mit dem Namen `alarm-model-payload.json`.
 - b. Kopieren Sie das folgende JSON-Objekt in die Datei.
 - c. Geben Sie einen Namen (`alarmModelName`), eine Beschreibung (`alarmModelDescription`) und einen Schweregrad (`severity`) für Ihren Alarm ein. Geben Sie für den Schweregrad eine Ganzzahl an, die den Schweregrad Ihres Unternehmens widerspiegelt.

 Important

Das Alarmmodell muss denselben Namen haben wie das zusammengesetzte Alarmmodell, das Sie zuvor für Ihr Anlagenmodell definiert haben. Namen von Alarmmodellen dürfen nur alphanumerische Zeichen enthalten.

```
{  
  "alarmModelName": "BoilerTemperatureHighAlarm",  
  "alarmModelDescription": "Detects when the boiler temperature is high.",  
  "severity": 3
```

```
}
```

- d. Fügen Sie dem Alarm die Vergleichsregel (`alarmRule`) hinzu. Diese Regel definiert die zu überwachende Eigenschaft (`inputProperty`), den zu vergleichenden Schwellenwert (`threshold`) und den zu verwendenden Vergleichsoperator (`comparisonOperator`).
- *`assetModelId`* Ersetzen Sie es durch die ID des Asset-Modells.
 - *`alarmPropertyId`* Ersetzen Sie durch die ID der Immobilie, die der Alarm überwacht.
 - *`thresholdAttributeId`* Ersetzen Sie es durch die ID der Attributeigenschaft „Threshold“.
 - *`GREATER`* Ersetzen Sie es durch den Operator, der verwendet werden soll, um die Eigenschaftswerte mit dem Schwellenwert zu vergleichen. Wählen Sie aus den folgenden Optionen aus:
 - LESS
 - LESS_OR_EQUAL
 - EQUAL
 - NOT_EQUAL
 - GREATER_OR_EQUAL
 - GREATER

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  }
}
```

- e. Fügen Sie eine Aktion (`alarmEventActions`) hinzu, um den Alarmstatus an den AWS IoT SiteWise Zeitpunkt zu senden, an dem sich der Zustand des Alarms ändert.

 Note

Für eine erweiterte Konfiguration können Sie zusätzliche Aktionen definieren, die ausgeführt werden, wenn sich der Zustand des Alarms ändert. Sie können beispielsweise eine AWS Lambda Funktion aufrufen oder zu einem MQTT-Thema veröffentlichen. Weitere Informationen finden Sie unter [Arbeiten mit anderen AWS Diensten](#) im AWS IoT EventsEntwicklerhandbuch.

- *assetModelId* Ersetzen Sie es durch die ID des Asset-Modells.
- *alarmPropertyId* Ersetzen Sie durch die ID der Immobilie, die der Alarm überwacht.
- *alarmStatePropertyId* Ersetzen Sie durch die ID der Eigenschaft „Alarmstatus“ im zusammengesetzten Alarmmodell.

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ]
  }
}
```

}

- f. (Optional) Konfigurieren Sie die Einstellungen für die Alarmbenachrichtigung. Die Alarmbenachrichtigungsaktion verwendet eine Lambda-Funktion in Ihrem Konto, um Alarmbenachrichtigungen zu senden. Weitere Informationen finden Sie unter [Anforderungen für Alarmmeldungen in AWS IoT SiteWise](#). In den Einstellungen für Alarmbenachrichtigungen können Sie SMS- und E-Mail-Benachrichtigungen konfigurieren, die an IAM Identity Center-Benutzer gesendet werden. Gehen Sie wie folgt vor:
- i. Fügen Sie die Konfiguration für Alarmbenachrichtigungen (alarmNotification) zur Payload in hinzu. alarm-model-payload.json
 - *alarmNotificationFunctionArn* Ersetzen Sie es durch den ARN der Lambda-Funktion, die Alarmbenachrichtigungen verarbeitet.

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ]
  },
  "alarmNotification": {
    "notificationActions": [
```

```

    {
      "action": {
        "lambdaAction": {
          "functionArn": "alarmNotificationFunctionArn"
        }
      }
    }
  ]
}

```

- ii. (Optional) Konfigurieren Sie die SMS-Benachrichtigungen (`smsConfigurations`), die an einen IAM Identity Center-Benutzer gesendet werden, wenn sich der Alarmstatus ändert.
 - *identityStoreIdAttributeId* Ersetzen Sie es durch die ID des Attributs, das die ID des IAM Identity Center-Identitätsspeichers enthält.
 - *userIdAttributeId* Ersetzen Sie es durch die ID des Attributs, das die ID des IAM Identity Center-Benutzers enthält.
 - *senderIdAttributeId* Ersetzen Sie es durch die ID des Attributs, das die Amazon SNS Sender-ID enthält, oder entfernen Sie es `senderId` aus der Payload.
 - *additionalMessageAttributeId* Ersetzen Sie es durch die ID des Attributs, das die zusätzliche Nachricht enthält, oder entfernen Sie sie `additionalMessage` aus der Payload.

```

{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
        "$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
        "$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [

```

```

    {
      "iotSiteWise": {
        "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
        "propertyId": "'alarmStatePropertyId'"
      }
    }
  ],
  "alarmNotification": {
    "notificationActions": [
      {
        "action": {
          "lambdaAction": {
            "functionArn": "alarmNotificationFunctionArn"
          }
        },
        "smsConfigurations": [
          {
            "recipients": [
              {
                "ssoIdentity": {
                  "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value",
                  "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
                }
              },
            ],
            "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId`.propertyValue.value",
            "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
          }
        ]
      }
    ]
  }
}

```

- iii. (Optional) Konfigurieren Sie die E-Mail-Benachrichtigungen (emailConfigurations), die an einen IAM Identity Center-Benutzer gesendet werden, wenn sich der Status des Alarms ändert.

- *identityStoreIdAttributeId* Ersetzen Sie es durch die ID der IAM Identity Center Identity Store-ID-Attributeigenschaft.
- *userIdAttributeId* Ersetzen Sie es durch die ID der IAM Identity Center-Benutzer-ID-Attributeigenschaft.
- *fromAddressAttributeId* Ersetzen Sie sie durch die ID der Adressattributeigenschaft „Von“ oder entfernen Sie sie from aus der Payload.
- *emailSubjectAttributeId* Ersetzen Sie es durch die ID der Eigenschaft des E-Mail-Betreff-Attributs oder entfernen Sie es subject aus der Payload.
- *additionalMessageAttributeId* Ersetzen Sie es durch die ID der zusätzlichen Nachrichtenattributeigenschaft oder entfernen Sie sie additionalMessage aus der Payload.

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
        "$sitewise.assetModel.`assetModelId`.`alarmPropertyId` .propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
        "$sitewise.assetModel.`assetModelId`.`thresholdAttributeId` .propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
            "$sitewise.assetModel.`assetModelId`.`alarmPropertyId` .assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ]
  },
  "alarmNotification": {
    "notificationActions": [
      {
```

```

    "action": {
      "lambdaAction": {
        "functionArn": "alarmNotificationFunctionArn"
      }
    },
    "smsConfigurations": [
      {
        "recipients": [
          {
            "ssoIdentity": {
              "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId` .propertyValue.value"
              "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId` .propertyValue.value"
            }
          }
        ],
        "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId` .propertyValue.value",
        "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId` .propertyValue.value"
      }
    ],
    "emailConfigurations": [
      {
        "from":
"$sitewise.assetModel.`assetModelId`.`fromAddressAttributeId` .propertyValue.value"
        "recipients": {
          "to": [
            {
              "ssoIdentity": {
                "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId` .propertyValue.value"
                "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId` .propertyValue.value"
              }
            }
          ]
        },
        "content": {
          "subject":
"$sitewise.assetModel.`assetModelId`.`emailSubjectAttributeId` .propertyValue.value"
          "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId` .propertyValue.value"
        }
      }
    ]
  }
}

```

```

    }
  }
]
}
}

```

- g. (Optional) Fügen Sie die Alarmfunktionen (alarmCapabilities) zur Payload-Eingabe hinzu. `alarm-model-payload.json` In diesem Objekt können Sie angeben, ob der Bestätigungsfluss aktiviert ist, und den standardmäßigen Aktivierungsstatus für Anlagen auf der Grundlage des Asset-Modells festlegen. Weitere Informationen zum Bestätigungsfluss finden Sie unter [Alarmzustände](#).

```

{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ]
  },
  "alarmNotification": {
    "notificationActions": [
      {
        "action": {
          "lambdaAction": {

```

```

        "functionArn": "alarmNotificationFunctionArn"
    },
    "smsConfigurations": [
        {
            "recipients": [
                {
                    "ssoIdentity": {
                        "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId` .propertyValue.value",
                        "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId` .propertyValue.value"
                    }
                }
            ],
            "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId` .propertyValue.value",
            "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId` .propertyValue.value"
        }
    ],
    "emailConfigurations": [
        {
            "from":
"$sitewise.assetModel.`assetModelId`.`fromAddressAttributeId` .propertyValue.value",
            "recipients": {
                "to": [
                    {
                        "ssoIdentity": {
                            "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId` .propertyValue.value",
                            "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId` .propertyValue.value"
                        }
                    }
                ]
            },
            "content": {
                "subject":
"$sitewise.assetModel.`assetModelId`.`emailSubjectAttributeId` .propertyValue.value",
                "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId` .propertyValue.value"
            }
        }
    ]
}

```

```

    ]
  }
]
},
"alarmCapabilities": {
  "initializationConfiguration": {
    "disabledOnInitialization": false
  },
  "acknowledgeFlow": {
    "enabled": true
  }
}
}
}

```

- h. Fügen Sie die IAM-Dienstrolle (roleArn) hinzu, die davon ausgehen AWS IoT Events kann, Daten an zu AWS IoT SiteWise senden. Für diese Rolle sind die `iotsitewise:BatchPutAssetPropertyValue` Genehmigung und eine Vertrauensbeziehung erforderlich, die es ermöglichen, die Rolle `iotevents.amazonaws.com` zu übernehmen. Zum Senden von Benachrichtigungen benötigt diese Rolle auch die `sso-directory:DescribeUser` Berechtigungen `lambda:InvokeFunction` und. Weitere Informationen finden Sie unter [Alarm-Dienstrollen](#) im AWS IoT EventsEntwicklerhandbuch.
- Ersetzen Sie das `roleArn` durch den ARN der Rolle, die diese Aktionen ausführen AWS IoT Events kann.

```

{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId` .propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId` .propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [

```

```

    {
      "iotSiteWise": {
        "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
        "propertyId": "'alarmStatePropertyId'"
      }
    }
  ],
  "alarmNotification": {
    "notificationActions": [
      {
        "action": {
          "lambdaAction": {
            "functionArn": "alarmNotificationFunctionArn"
          }
        },
        "smsConfigurations": [
          {
            "recipients": [
              {
                "ssoIdentity": {
                  "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value",
                  "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
                }
              }
            ],
            "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId`.propertyValue.value",
            "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
          }
        ],
        "emailConfigurations": [
          {
            "from":
"$sitewise.assetModel.`assetModelId`.`fromAddressAttributeId`.propertyValue.value",
            "recipients": {
              "to": [
                {
                  "ssoIdentity": {

```

```

        "identityStoreId":
        "$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value"
        "userId":
        "$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
    }
  ],
  },
  "content": {
    "subject":
    "$sitewise.assetModel.`assetModelId`.`emailSubjectAttributeId`.propertyValue.value",
    "additionalMessage":
    "$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
  }
}
]
}
],
},
"alarmCapabilities": {
  "initializationConfiguration": {
    "disabledOnInitialization": false
  },
  "acknowledgeFlow": {
    "enabled": false
  }
},
"roleArn": "arn:aws:iam::123456789012:role/MyIoTEventsAlarmRole"
}

```

- i. Führen Sie den folgenden Befehl aus, um das AWS IoT Events Alarmmodell aus der Payload in `alarm-model-payload.json` zu erstellen.

```
aws iotevents create-alarm-model --cli-input-json file://alarm-model-payload.json
```

- j. Die Operation gibt eine Antwort zurück, die den ARN des Alarmmodells enthält, `alarmModelArn`. Kopieren Sie diesen ARN, um ihn im nächsten Schritt in der Alarmdefinition Ihres Asset-Modells festzulegen.

Schritt 3: Aktivieren Sie den Datenfluss zwischenAWS IoT SiteWiseundAWS IoT Events

Nachdem Sie die erforderlichen Ressourcen in AWS IoT SiteWise und erstellt habenAWS IoT Events, können Sie den Datenfluss zwischen den Ressourcen aktivieren, um Ihren Alarm zu aktivieren. In diesem Abschnitt aktualisieren Sie die Alarmdefinition im Asset-Modell, um das Alarmmodell zu verwenden, das Sie im vorherigen Schritt erstellt haben.

Um den Datenfluss zwischen zu ermöglichenAWS IoT SiteWiseundAWS IoT Events(CLI)

- Stellen Sie das Alarmmodell als Quelle des Alarms im Asset-Modell ein. Gehen Sie wie folgt vor:
 - a. Führen Sie den folgenden Befehl aus, um die vorhandene Komponentenmodelldefinition abzurufen. *asset-model-id* Ersetzen Sie es durch die ID des Asset-Modells.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

Die Operation gibt eine Antwort zurück, die Details des Komponentenmodells enthält.

- b. Erstellen Sie eine Datei namens `update-asset-model-payload.json` und kopieren Sie die Antwort des vorherigen Befehls in die Datei.
- c. Entfernen Sie die folgenden Schlüssel-Wert-Paare aus der `update-asset-model-payload.json` Datei:
 - `assetModelId`
 - `assetModelArn`
 - `assetModelCreationDate`
 - `assetModelLastUpdateDate`
 - `assetModelStatus`
- d. Fügen Sie dem zuvor definierten zusammengesetzten Alarmmodell die Eigenschaft `Alarmquelle` (AWS/ALARM_SOURCE) hinzu. *alarmModelArn* Ersetzen Sie ihn durch den ARN des Alarmmodells, der den Wert der Eigenschaft `Alarmquelle` festlegt.

```
{
  ...
  "assetModelCompositeModels": [
    ...
    {
      "name": "BoilerTemperatureHighAlarm",
      "type": "AWS/ALARM",
```



```

    "properties": [
      {
        "id": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "name": "AWS/ALARM_TYPE",
        "dataType": "STRING",
        "type": {
          "attribute": {
            "defaultValue": "IOT_EVENTS"
          }
        }
      },
      {
        "id": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
        "name": "AWS/ALARM_STATE",
        "dataType": "STRUCT",
        "dataTypeSpec": "AWS/ALARM_STATE",
        "type": {
          "measurement": {}
        }
      },
      {
        "name": "AWS/ALARM_SOURCE",
        "dataType": "STRING",
        "type": {
          "attribute": {
            "defaultValue": "alarmModelArn"
          }
        }
      }
    ]
  }
}

```

- e. Führen Sie den folgenden Befehl aus, um das Asset-Modell mit der in der `update-asset-model-payload.json` Datei gespeicherten Definition zu aktualisieren. *asset-model-id* Ersetzen Sie es durch die ID des Asset-Modells.

```

aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --cli-input-json file://update-asset-model-payload.json

```

Ihr Anlagenmodell definiert jetzt einen Alarm, der eindringt AWS IoT Events. Der Alarm überwacht die Zielimmobilie in allen Anlagen, die auf diesem Anlagenmodell basieren. Sie können den Alarm für jedes Asset konfigurieren, um Eigenschaften wie den Schwellenwert oder den IAM Identity Center-Empfänger für jedes Asset anzupassen. Weitere Informationen finden Sie unter [Konfigurieren Sie Alarme für Anlagen in AWS IoT SiteWise](#).

Definieren Sie externe Alarme in AWS IoT SiteWise

Externe Alarme enthalten den Status eines Alarms, den Sie außerhalb eines AWS IoT SiteWise Alarms erkennen.

Definieren Sie einen externen Alarm (Konsole)

Sie können die AWS IoT SiteWise Konsole verwenden, um einen externen Alarm für ein vorhandenes Anlagenmodell zu definieren. Um einen externen Alarm für ein neues Asset-Modell zu definieren, erstellen Sie das Asset-Modell und führen Sie dann diese Schritte aus. Weitere Informationen finden Sie unter [Erstellen Sie Asset-Modelle in AWS IoT SiteWise](#).

Um einen Alarm für ein Asset-Modell zu definieren

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Klicken Sie im Navigationsbereich auf Models (Modelle).
3. Wählen Sie das Asset-Modell aus, für das Sie einen Alarm definieren möchten.
4. Wählen Sie die Registerkarte Alarmdefinitionen.
5. Wählen Sie Alarm hinzufügen.
6. Wählen Sie in den Optionen für den Alarmtyp die Option Externer Alarm aus.
7. Geben Sie einen Namen für den Alarm ein.
8. (Optional) Geben Sie eine Beschreibung für Ihren Alarm ein.
9. Wählen Sie Alarm hinzufügen.

Definieren Sie einen externen Alarm (CLI)

Sie können den verwenden AWS CLI, um einen externen Alarm für ein neues oder vorhandenes Anlagenmodell zu definieren.

Um einem Asset-Modell einen externen Alarm hinzuzufügen, fügen Sie dem Asset-Modell ein zusammengesetztes Alarmmodell hinzu. Ein zusammengesetztes externes Alarmmodell spezifiziert

den EXTERNAL Typ und keine Eigenschaft der Alarmquelle. Das folgende Beispiel für einen zusammengesetzten Alarm definiert einen externen Temperaturalarm.

```
{
  ...
  "assetModelCompositeModels": [
    {
      "name": "BoilerTemperatureHighAlarm",
      "type": "AWS/ALARM",
      "properties": [
        {
          "name": "AWS/ALARM_TYPE",
          "dataType": "STRING",
          "type": {
            "attribute": {
              "defaultValue": "EXTERNAL"
            }
          }
        },
        {
          "name": "AWS/ALARM_STATE",
          "dataType": "STRUCT",
          "dataTypeSpec": "AWS/ALARM_STATE",
          "type": {
            "measurement": {}
          }
        }
      ]
    }
  ]
}
```

Weitere Informationen zum Hinzufügen eines zusammengesetzten Modells zu einem neuen oder vorhandenen Anlagenmodell finden Sie im Folgenden:

- [Erstellen Sie ein Asset-Modell \(AWS CLI\)](#)
- [Aktualisieren Sie ein Objektmodell, ein Komponentenmodell oder eine Schnittstelle \(AWS CLI\)](#)

Nachdem Sie den externen Alarm definiert haben, können Sie den Alarmstatus auf der Grundlage des Asset-Modells in Anlagen aufnehmen. Weitere Informationen finden Sie unter [Erfassen Sie einen externen Alarmstatus in AWS IoT SiteWise](#).

Konfigurieren Sie Alarme für Anlagen in AWS IoT SiteWise

Nachdem Sie einen AWS IoT Events Alarm für ein Asset-Modell definiert haben, können Sie den Alarm für jedes Asset basierend auf dem Asset-Modell konfigurieren. Sie können den Schwellenwert und die Benachrichtigungseinstellungen für den Alarm bearbeiten. Jeder dieser Werte ist ein Attribut auf dem Asset, sodass Sie den Standardwert des Attributs aktualisieren können, um diese Werte zu konfigurieren.

Note

Sie können diese Werte für AWS IoT Events Alarme konfigurieren, jedoch nicht für externe Alarme.

Themen

- [Konfigurieren Sie einen Schwellenwert \(Konsole\)](#)
- [Konfigurieren Sie einen Schwellenwert \(AWS CLI\)](#)
- [Konfigurieren Sie die Benachrichtigungseinstellungen in AWS IoT SiteWise](#)

Konfigurieren Sie einen Schwellenwert (Konsole)

Sie können die AWS IoT SiteWise Konsole verwenden, um den Wert des Attributs zu aktualisieren, das den Schwellenwert eines Alarms angibt.

Um den Schwellenwert eines Alarms zu aktualisieren (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im linken Navigationsbereich Komponenten aus.
3. Wählen Sie das Asset aus, für das Sie einen Alarmschwellenwert aktualisieren möchten.

Tip

Sie können eine Komponentenhierarchie mithilfe des Pfeilsymbols erweitern, um nach Ihrer Komponente zu suchen.

4. Wählen Sie Bearbeiten aus.

5. Suchen Sie das Attribut, das der Alarm für seinen Schwellenwert verwendet, und geben Sie dann seinen neuen Wert ein.
6. Wählen Sie Speichern.

Konfigurieren Sie einen Schwellenwert (AWS CLI)

Sie können das AWS Command Line Interface (AWS CLI) verwenden, um den Wert des Attributs zu aktualisieren, das den Schwellenwert eines Alarms angibt.

Um dieses Verfahren abzuschließen, müssen Sie die `assetId` Ihrer Komponenten und die `propertyId` Ihrer Eigenschaft kennen. Sie können auch die externe ID verwenden. Wenn Sie ein Asset erstellt haben und es nicht kennen `assetId`, verwenden Sie die [ListAssets](#) API, um alle Assets für ein bestimmtes Modell aufzulisten. Verwenden Sie den [DescribeAsset](#) Vorgang, um die Eigenschaften Ihres Assets einschließlich der Eigenschafts-IDs anzuzeigen.

Verwenden Sie die Operation [BatchPutAssetPropertyValue](#), um Attributwerte zu Ihrer Komponente zuzuweisen. Mit dieser Operation können Sie mehrere Attribute gleichzeitig festlegen. Die Nutzlast dieser Operation enthält eine Liste von Einträgen, jeweils mit der Komponenten-ID, der Eigenschafts-ID und dem Attributwert.

Um den Wert eines Attributs zu aktualisieren (AWS CLI)

1. Erstellen Sie eine Datei namens `batch-put-payload.json` und kopieren Sie das folgende JSON-Objekt in die Datei. In diesem Nutzlast-Beispiel wird veranschaulicht, wie der Breiten- und Längengrad einer Windturbine festgelegt wird. Aktualisieren Sie die IDs, Werte und Zeitstempel, um die Nutzlast für Ihren Anwendungsfall zu ändern.

```
{
  "entries": [
    {
      "entryId": "windfarm3-turbine7-latitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 47.6204
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    }
  ]
}
```

```

    }
  }
]
},
{
  "entryId": "windfarm3-turbine7-longitude",
  "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE",
  "propertyValues": [
    {
      "value": {
        "doubleValue": 122.3491
      },
      "timestamp": {
        "timeInSeconds": 1575691200
      }
    }
  ]
}
]
}
}

```

- Jeder Eintrag in der Nutzlast enthält eine `entryId`, die Sie als eindeutige Zeichenfolge definieren können. Bei fehlgeschlagenen Anforderungseinträgen enthält jeder Fehler die `entryId` der entsprechenden Anforderung, woran Sie erkennen können, welche Anforderungen zu wiederholen sind.
- Um einen Attributwert festzulegen, können Sie für jede Attributeigenschaft eine Zeitstempelqualitätswert-Struktur (TQV) in die Liste von `propertyValues` aufnehmen. Diese Struktur muss den neuen `value` und den aktuellen `timestamp` enthalten.
- `value`— Eine Struktur, die je nach Typ der festgelegten Eigenschaft eines der folgenden Felder enthält:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`

- **timestamp**— Eine Struktur, die die aktuelle Unix-Epoche in Sekunden enthält, `timeInSeconds`. AWS IoT SiteWise lehnt alle Datenpunkte mit Zeitstempeln ab, die länger als 7 Tage in der Vergangenheit oder neuer als 5 Minuten in der future existierten.

Weitere Informationen zum Vorbereiten einer Nutzlast für [BatchPutAssetPropertyValue](#) finden Sie unter [Daten aufnehmen mit AWS IoT SiteWise APIs](#).

2. Führen Sie den folgenden Befehl aus, um die Attributwerte an zu senden: AWS IoT SiteWise

```
aws iotsitewise batch-put-asset-property-value -\-cli-input-json file://batch-put-payload.json
```

Konfigurieren Sie die Benachrichtigungseinstellungen in AWS IoT SiteWise

Sie können die Einstellungen für Alarmbenachrichtigungen entweder mit der AWS IoT SiteWise Konsole oder mit AWS Command Line Interface (AWS CLI) konfigurieren.

Konfigurieren Sie die Benachrichtigungseinstellungen (Konsole)

Sie können die AWS IoT SiteWise Konsole verwenden, um den Wert der Attribute zu aktualisieren, die die Benachrichtigungseinstellungen für einen Alarm angeben.

Um die Benachrichtigungseinstellungen eines Alarms zu aktualisieren (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im linken Navigationsbereich Komponenten aus.
3. Wählen Sie das Asset aus, für das Sie die Alarmeinstellungen aktualisieren möchten.
4. Wählen Sie Bearbeiten aus.
5. Suchen Sie das Attribut, das der Alarm für die Benachrichtigungseinstellung verwendet, die Sie ändern möchten, und geben Sie dann den neuen Wert ein.
6. Wählen Sie Speichern.

Benachrichtigungseinstellungen konfigurieren (CLI)

Sie können die AWS Command Line Interface (AWS CLI) verwenden, um den Wert des Attributs zu aktualisieren, das die Benachrichtigungseinstellungen für einen Alarm angibt.

Um dieses Verfahren abzuschließen, müssen Sie die `assetId` Ihrer Komponenten und die `propertyId` Ihrer Eigenschaft kennen. Sie können auch die externe ID verwenden. Wenn Sie ein Asset erstellt haben und es nicht kennen `assetId`, verwenden Sie die [ListAssets](#) API, um alle Assets für ein bestimmtes Modell aufzulisten. Verwenden Sie den [DescribeAsset](#) Vorgang, um die Eigenschaften Ihres Assets einschließlich der Eigenschafts-IDs anzuzeigen.

Verwenden Sie die Operation [BatchPutAssetPropertyValue](#), um Attributwerte zu Ihrer Komponente zuzuweisen. Mit dieser Operation können Sie mehrere Attribute gleichzeitig festlegen. Die Nutzlast dieser Operation enthält eine Liste von Einträgen, jeweils mit der Komponenten-ID, der Eigenschafts-ID und dem Attributwert.

Um den Wert eines Attributs zu aktualisieren (AWS CLI)

1. Erstellen Sie eine Datei namens `batch-put-payload.json` und kopieren Sie das folgende JSON-Objekt in die Datei. In diesem Nutzlast-Beispiel wird veranschaulicht, wie der Breiten- und Längengrad einer Windturbine festgelegt wird. Aktualisieren Sie die IDs, Werte und Zeitstempel, um die Nutzlast für Ihren Anwendungsfall zu ändern.

```
{
  "entries": [
    {
      "entryId": "windfarm3-turbine7-latitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 47.6204
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    },
    {
      "entryId": "windfarm3-turbine7-longitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE",
      "propertyValues": [
        {
          "value": {
```



```

        "doubleValue": 122.3491
      },
      "timestamp": {
        "timeInSeconds": 1575691200
      }
    ]
  }
}

```

- Jeder Eintrag in der Nutzlast enthält eine `entryId`, die Sie als eindeutige Zeichenfolge definieren können. Bei fehlgeschlagenen Anforderungseinträgen enthält jeder Fehler die `entryId` der entsprechenden Anforderung, woran Sie erkennen können, welche Anforderungen zu wiederholen sind.
- Um einen Attributwert festzulegen, können Sie für jede Attributeigenschaft eine Zeitstempelqualitätswert-Struktur (TQV) in die Liste von `propertyValues` aufnehmen. Diese Struktur muss den neuen `value` und den aktuellen `timestamp` enthalten.
- `value`— Eine Struktur, die je nach Typ der festgelegten Eigenschaft eines der folgenden Felder enthält:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`
- `timestamp`— Eine Struktur, die die aktuelle Unix-Epoche in Sekunden enthält, `timeInSeconds`. AWS IoT SiteWise lehnt alle Datenpunkte mit Zeitstempeln ab, die länger als 7 Tage in der Vergangenheit oder neuer als 5 Minuten in der future existierten.

Weitere Informationen zum Vorbereiten einer Nutzlast für [BatchPutAssetPropertyValue](#) finden Sie unter [Daten aufnehmen mit AWS IoT SiteWise APIs](#).

2. Führen Sie den folgenden Befehl aus, um die Attributwerte an zu senden: AWS IoT SiteWise

```
aws iotsitewise batch-put-asset-property-value -\cli-input-json file://batch-put-payload.json
```

Reagieren Sie auf Alarme in AWS IoT SiteWise

Wenn sich der Status eines AWS IoT Events Alarms ändert, können Sie wie folgt auf den Alarm reagieren:

- Bestätigen Sie einen Alarm, um anzuzeigen, dass Sie sich mit dem Problem befassen.
- Schalten Sie einen Alarm in die Schlummerfunktion, um ihn vorübergehend zu deaktivieren.
- Deaktivieren Sie einen Alarm, um ihn dauerhaft zu deaktivieren, bis Sie ihn wieder aktivieren.
- Aktivieren Sie einen deaktivierten Alarm, um den Alarmstatus zu erkennen.
- Setzen Sie einen Alarm zurück, um seinen Status und seinen letzten Wert zu löschen.

Sie können die AWS IoT SiteWise Konsole oder die AWS IoT Events API verwenden, um auf einen Alarm zu reagieren.

Note

Sie können auf AWS IoT Events Alarme reagieren, aber nicht auf externe Alarme.

Themen

- [Reagieren Sie auf einen Alarm \(Konsole\)](#)
- [Reagieren Sie auf einen Alarm \(API\)](#)

Reagieren Sie auf einen Alarm (Konsole)

Sie können die AWS IoT SiteWise Konsole verwenden, um einen Alarm zu bestätigen, in den Schlummermodus zu versetzen, zu deaktivieren oder zu aktivieren.

Themen

- [Bestätigen Sie einen Alarm \(Konsole\)](#)
- [Alarmanlage ausschalten \(Konsole\)](#)
- [Deaktiviert einen Alarm \(Konsole\)](#)
- [Aktiviert einen Alarm \(Konsole\)](#)
- [Einen Alarm zurücksetzen \(Konsole\)](#)

Bestätigen Sie einen Alarm (Konsole)

Sie können einen Alarm bestätigen, um anzuzeigen, dass Sie sich mit dem Problem befassen.

Note

Sie müssen den Bestätigungsfluss für den Alarm aktivieren, damit Sie den Alarm bestätigen können. Diese Option ist standardmäßig aktiviert, wenn Sie den Alarm von der AWS IoT SiteWise Konsole aus definieren.

Um einen Alarm zu bestätigen (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im linken Navigationsbereich Komponenten aus.
3. Wählen Sie das Asset aus, für das Sie einen Alarm bestätigen möchten.

Tip

Sie können eine Komponentenhierarchie mithilfe des Pfeilsymbols erweitern, um nach Ihrer Komponente zu suchen.

4. Wählen Sie die Registerkarte Alarme.
5. Wählen Sie den Alarm aus, den Sie bestätigen möchten, und wählen Sie dann Aktionen, um das Aktionsmenü für die Reaktion zu öffnen.
6. Wählen Sie Acknowledge (Bestätigen). Der Status des Alarms wechselt zu Bestätigt.

Alarmanlage ausschalten (Konsole)

Sie können einen Alarm in die Schlummerfunktion versetzen, um ihn vorübergehend zu deaktivieren. Geben Sie die Dauer an, für die der Alarm deaktiviert werden soll.

Um einen Alarm in die Schlummerfunktion zu versetzen (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im linken Navigationsbereich Komponenten aus.
3. Wählen Sie das Asset aus, für das Sie einen Alarm deaktivieren möchten.

 Tip

Sie können eine Komponentenhierarchie mithilfe des Pfeilsymbols erweitern, um nach Ihrer Komponente zu suchen.

4. Wählen Sie die Registerkarte Alarme.
5. Wählen Sie den Alarm aus, der in den Schlummermodus versetzt werden soll, und wählen Sie dann Aktionen, um das Aktionsmenü für die Reaktion zu öffnen.
6. Wählen Sie „Schlummern“. Es wird ein Modell geöffnet, in dem Sie die Dauer für den Schlummermodus angeben.
7. Wählen Sie die Schlummerlänge oder geben Sie eine benutzerdefinierte Schlummerlänge ein.
8. Wählen Sie Speichern. Der Alarmstatus wechselt zu Snoozed.

Deaktiviert einen Alarm (Konsole)

Sie können einen Alarm deaktivieren, sodass er nicht mehr erkannt wird. Nachdem Sie den Alarm deaktiviert haben, müssen Sie ihn erneut aktivieren, wenn der Alarm erkannt werden soll.

Um einen Alarm zu deaktivieren (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im linken Navigationsbereich Komponenten aus.
3. Wählen Sie das Asset aus, für das Sie einen Alarm deaktivieren möchten.

 Tip

Sie können eine Komponentenhierarchie mithilfe des Pfeilsymbols erweitern, um nach Ihrer Komponente zu suchen.

4. Wählen Sie die Registerkarte Alarme.
5. Wählen Sie den Alarm aus, den Sie deaktivieren möchten, und wählen Sie dann Aktionen, um das Aktionsmenü für die Reaktion zu öffnen.
6. Wählen Sie Disable (deaktivieren) aus. Der Status des Alarms ändert sich zu Deaktiviert.

Aktiviert einen Alarm (Konsole)

Sie können einen Alarm so einrichten, dass er erneut erkannt wird, nachdem Sie ihn deaktiviert oder die Schlummerfunktion aktiviert haben.

Um einen Alarm zu aktivieren (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im linken Navigationsbereich Komponenten aus.
3. Wählen Sie das Asset aus, für das Sie einen Alarm aktivieren möchten.

Tip

Sie können eine Komponentenhierarchie mithilfe des Pfeilsymbols erweitern, um nach Ihrer Komponente zu suchen.

4. Wählen Sie die Registerkarte Alarme.
5. Wählen Sie den Alarm aus, den Sie aktivieren möchten, und wählen Sie dann Aktionen, um das Aktionsmenü für die Reaktion zu öffnen.
6. Wählen Sie Enable (Aktivieren) aus. Der Zustand des Alarms wechselt zu Normal.

Einen Alarm zurücksetzen (Konsole)

Sie können einen Alarm zurücksetzen, um seinen Status und seinen letzten Wert zu löschen.

Um einen Alarm zurückzusetzen (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im linken Navigationsbereich Komponenten aus.
3. Wählen Sie das Asset aus, für das Sie einen Alarm zurücksetzen möchten.

Tip

Sie können eine Komponentenhierarchie mithilfe des Pfeilsymbols erweitern, um nach Ihrer Komponente zu suchen.

4. Wählen Sie die Registerkarte Alarme.

5. Wählen Sie den Alarm aus, den Sie aktivieren möchten, und wählen Sie dann Aktionen, um das Aktionsmenü für die Reaktion zu öffnen.
6. Klicken Sie auf Reset (Zurücksetzen). Der Zustand des Alarms wechselt zu Normal.

Reagieren Sie auf einen Alarm (API)

Sie können die AWS IoT Events API verwenden, um einen Alarm zu bestätigen, in den Schlummermodus zu versetzen, zu deaktivieren, zu aktivieren oder zurückzusetzen. Weitere Informationen finden Sie in der AWS IoT EventsAPI-Referenz zu den folgenden Vorgängen:

- [BatchAcknowledgeAlarm](#)
- [BatchSnoozeAlarm](#)
- [BatchDisableAlarm](#)
- [BatchEnableAlarm](#)
- [BatchResetAlarm](#)

Weitere Informationen finden Sie unter [Reagieren auf Alarme](#) im AWS IoT EventsEntwicklerhandbuch.

Erfassen Sie einen externen Alarmstatus inAWS IoT SiteWise

Externe Alarme sind Alarme, die Sie außerhalb von auswertenAWS IoT SiteWise. Sie können externe Alarme verwenden, wenn Sie über eine Datenquelle verfügen, die den Alarmstatus meldet und in die Sie Daten aufnehmen möchtenAWS IoT SiteWise.

Für die Eigenschaften des Alarmstatus ist ein bestimmtes Format für die Datenwerte des Alarmstatus erforderlich. Jeder Datenwert muss ein JSON-Objekt sein, das in eine Zeichenfolge serialisiert ist. Anschließend nehmen Sie die serialisierte Zeichenfolge als Zeichenkettenwert auf. Weitere Informationen finden Sie unter [Eigenschaften des Alarmstatus](#).

Example Beispiel für einen Datenwert für den Alarmstatus (nicht serialisiert)

```
{
  "stateName": "Active"
}
```

Example Beispiel für einen Datenwert für den Alarmstatus (serialisiert)

```
{\"stateName\":\"Active\"}
```

Note

Wenn Ihre Datenquelle keine Daten in diesem Format melden kann oder Sie Ihre Daten vor der Aufnahme nicht in dieses Format konvertieren können, entscheiden Sie sich möglicherweise dafür, keine Alarm-Eigenschaft zu verwenden. Stattdessen können Sie die Daten beispielsweise als Messeigenschaft mit dem Datentyp Zeichenfolge aufnehmen. Weitere Informationen erhalten Sie unter [Definieren Sie Datenströme von Geräten \(Messungen\)](#) und [Daten aufnehmen in AWS IoT SiteWise](#).

Ordnen Sie externe Alarmstatus-Streams zu AWS IoT SiteWise

Sie können Eigenschaftsalias definieren, um Ihre Datenströme Ihren Alarmzustandseigenschaften zuzuordnen. Auf diese Weise können Sie beim Aufnehmen oder Abrufen von Daten auf einfache Weise eine Eigenschaft für den Alarmstatus identifizieren. Weitere Informationen zu Eigenschaftsaliasnamen finden Sie unter [Datenströme verwalten für AWS IoT SiteWise](#).

Themen

- [Ordnen Sie externe Alarmstatus-Streams zu \(Konsole\)](#)
- [Ordnen Sie externe Alarmstatus-Streams zu \(AWS CLI\)](#)

Ordnen Sie externe Alarmstatus-Streams zu (Konsole)

Sie können Eigenschaftsalias definieren, um Ihre Datenströme Ihren Alarmzustandseigenschaften zuzuordnen. Auf diese Weise können Sie beim Aufnehmen oder Abrufen von Daten auf einfache Weise eine Eigenschaft für den Alarmstatus identifizieren. Weitere Informationen zu Eigenschaftsaliasnamen finden Sie unter [Datenströme verwalten für AWS IoT SiteWise](#).

Sie können die AWS IoT SiteWise Konsole verwenden, um einen Alias für eine Alarmstatuseigenschaft festzulegen.

Um einen Eigenschaftsalias für eine Alarmstatuseigenschaft festzulegen (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).

2. Wählen Sie im linken Navigationsbereich Komponenten aus.
3. Wählen Sie die Komponente aus, für die Sie einen Eigenschaftsalias festlegen möchten.

 Tip

Sie können eine Komponentenhierarchie mithilfe des Pfeilsymbols erweitern, um nach Ihrer Komponente zu suchen.

4. Wählen Sie Bearbeiten aus.
5. Scrollen Sie zu Alarme und erweitern Sie den Bereich.
6. Geben Sie unter Externe Alarme den Alias im Feld Eigenschaftsalias — optional ein.
7. Wählen Sie Speichern.

Ordnen Sie externe Alarmstatus-Streams zu (AWS CLI)

Sie können Eigenschaftsalias definieren, um Ihre Datenströme Ihren Alarmzustandseigenschaften zuzuordnen. Auf diese Weise können Sie beim Aufnehmen oder Abrufen von Daten auf einfache Weise eine Eigenschaft für den Alarmstatus identifizieren. Weitere Informationen zu Eigenschaftsaliasnamen finden Sie unter [Datenströme verwalten für AWS IoT SiteWise](#)

Sie können das AWS Command Line Interface (AWS CLI) verwenden, um einen Alias für eine Alarmzustandseigenschaft festzulegen.

Um dieses Verfahren abzuschließen, müssen Sie die `assetId` Ihrer Komponenten und die `propertyId` Ihrer Eigenschaft kennen. Sie können auch die externe ID verwenden. Wenn Sie ein Asset erstellt haben und es nicht kennen, verwenden Sie die [ListAssets](#) API, um alle Assets für ein bestimmtes Modell aufzulisten. Verwenden Sie den [DescribeAsset](#) Vorgang, um die Eigenschaften Ihres Assets einschließlich der Eigenschafts-IDs anzuzeigen.

 Note

Die [DescribeAsset](#) Antwort enthält die Liste der zusammengesetzten Asset-Modelle für das Asset. Jeder Alarm ist ein zusammengesetztes Modell. Um das zu finden, suchen Sie das zusammengesetzte Modell für den Alarm und suchen Sie dann die `AWS/ALARM_STATE` Eigenschaft in diesem zusammengesetzten Modell.

Weitere Hinweise zum Festlegen des Eigenschaftsalias finden Sie unter [Aktualisieren Sie den Alias einer Asset-Eigenschaft](#).

Erfassen Sie Alarmstatusdaten in AWS IoT SiteWise

Eigenschaften für den Alarmstatus erwarten den Alarmstatus als serialisierte JSON-Zeichenfolge. Um den Alarmstatus in einen externen Alarm zu übernehmen AWS IoT SiteWise, nehmen Sie diese serialisierte Zeichenfolge als Zeichenkettenwert mit Zeitstempel auf. Das folgende Beispiel zeigt einen Statusdatenwert für einen aktiven Alarm.

```
{\"stateName\": \"Active\"}
```

Um eine Eigenschaft für den Alarmstatus zu identifizieren, können Sie eine der folgenden Optionen angeben:

- Das `assetId` Ende `propertyId` der Alarm-Eigenschaft, an die Sie Daten senden.
- Das `propertyAlias`, was ein Datenstream-Alias ist (z. B. `/company/windfarm/3/turbine/7/temperature/high`). Um diese Option verwenden zu können, müssen Sie zuerst den Alias Ihrer Alarm-Eigenschaft festlegen. Informationen zum Festlegen von Eigenschaftsaliasnamen für Eigenschaften des Alarmstatus finden Sie unter [Ordnen Sie externe Alarmstatus-Streams zu AWS IoT SiteWise](#).

Das folgende Beispiel für eine [BatchPutAssetPropertyValue](#) API-Payload zeigt, wie der Status eines externen Alarms formatiert wird. Dieser externe Alarm meldet, wenn der Wert der Umdrehungen pro Minute (U/min) einer Windturbine zu hoch ist.

Example Beispiel für eine BatchPutAssetPropertyValue Payload für Alarmzustandsdaten

```
{
  "entries": [
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/temperature/high",
      "propertyValues": [
        {
          "value": {
            "stringValue": "{\"stateName\": \"Active\"}"
          },
          "timestamp": {
```

```
    "timeInSeconds": 1607550262
  }
}
]
```

Weitere Hinweise zur Verwendung der BatchPutAssetPropertyValue API zum Erfassen von Daten finden Sie unter. [Daten aufnehmen mit AWS IoT SiteWise APIs](#)

Weitere Hinweise zu anderen Möglichkeiten der Datenaufnahme finden Sie unter. [Daten aufnehmen inAWS IoT SiteWise](#)

AWS IoT SiteWise Assistentin

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Der AWS IoT SiteWise Assistent ist ein generativer KI-gestützter Assistent. Er ermöglicht es Benutzern wie Werksleitern, Qualitätsingenieuren und Wartungstechnikern, direkt aus ihren Betriebs- und Unternehmensdaten Erkenntnisse zu gewinnen, Probleme zu lösen und Maßnahmen zu ergreifen.

Der AWS IoT SiteWise Assistent konsolidiert Informationen aus AWS IoT Daten, Anlagenmodellen, Handbüchern und Dokumentationen in verständlichen Zusammenfassungen kritischer Ereignisse. Darüber hinaus ermöglicht er interaktive, vertiefende Frage-und-Antwort-Sitzungen für einfache Diagnosen, Ursachenforschung und gezielte Empfehlungen.

Themen

- [Konfigurieren Sie den Assistenten AWS IoT SiteWise](#)
- [Erstellen eines Datensatzes](#)
- [Bearbeiten Sie einen Datensatz](#)
- [Löschen Sie einen Datensatz](#)
- [AWS IoT SiteWise Fragen des Assistenten](#)

Konfigurieren Sie den Assistenten AWS IoT SiteWise

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

AWS IoT SiteWise Konfiguration des Assistenten

1. Melden Sie sich bei der [AWS IoT SiteWise -Konsole](#) an.

Note

Erteilen Sie Berechtigungen, um die Integration mit dem AWS IoT TwinMaker Service zu ermöglichen. Dies ist erforderlich, damit der AWS IoT SiteWise Assistent und das Dashboard SQL-Abfragen in AWS IoT SiteWise Ressourcen ausführen können. Siehe [Integrieren AWS IoT SiteWise und AWS IoT TwinMaker](#).

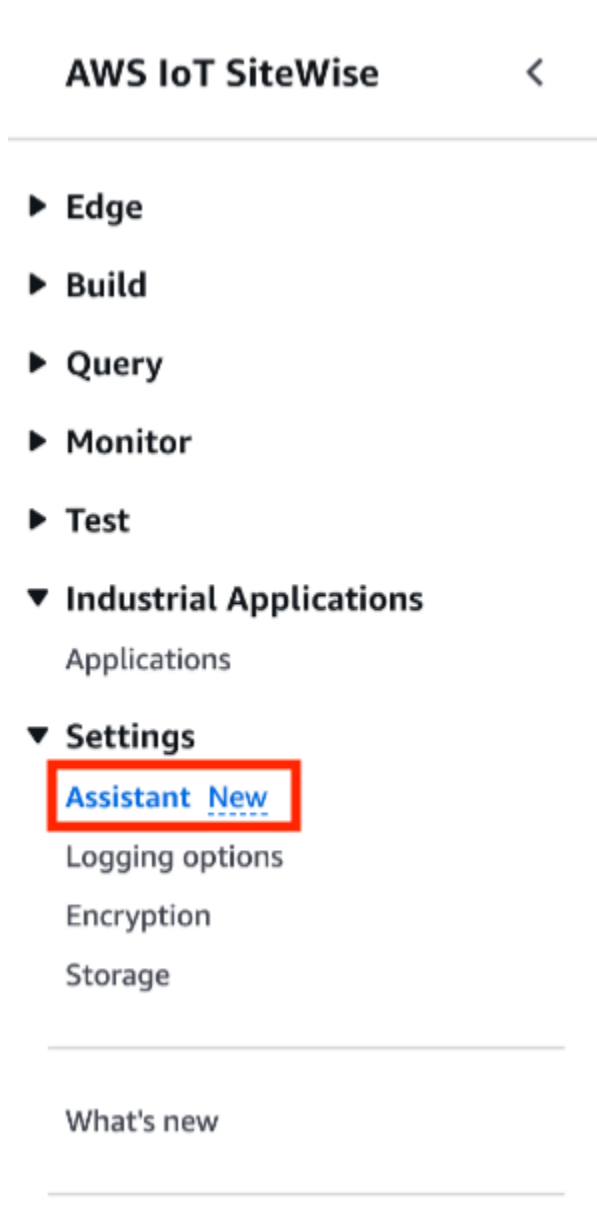
Grant permission

In order to better organize and model industrial data, AWS IoT SiteWise requires permission to integrate with AWS IoT TwinMaker. Choose **Grant permission** to enable this integration. This integration could incur costs. For more information, see [pricing](#) .

[Cancel](#)

Grant permission

2. Wählen Sie im linken Navigationsbereich Assistent aus.



Erstellen eines Datensatzes

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

 Note

Der AWS IoT SiteWise Assistent muss einen Datensatz mit einem [Amazon Kendra Kendra-Index](#) verwenden, um Wissen und Beratung auf Unternehmensebene zu erhalten. Wenn Sie keinen Amazon Kendra Kendra-Index haben, finden Sie Informationen zur [Erstellung eines Indexes unter Index](#) erstellen. Das Hinzufügen eines [Datensatzes](#) verbessert die Qualität der Reaktion des Assistenten und minimiert Halluzinationen.

Console

Erstellen Sie einen Datensatz in der Konsole AWS IoT SiteWise

1. Datensätze werden im Abschnitt Datensätze der AWS IoT SiteWise Assistentenseite angezeigt.
2. Wenn keine Datensätze vorhanden sind, wählen Sie Datensatz erstellen.
3. Wählen Sie auf der Seite mit den Datensatz-Details einen Kendra-Index aus dem Drop-down-Menü aus, den Sie mit dem Datensatz verknüpfen möchten.
4. Der Datensatzname wird durch den in Schritt 3 ausgewählten Kendra-Index aufgefüllt. Bearbeiten Sie den Namen bei Bedarf.
5. (Optional) Die Datensatzbeschreibung wird mit dem in Schritt 3 ausgewählten Kendra-Index gefüllt. Bearbeiten Sie die Beschreibung bei Bedarf.
6. Wählen Sie im Bereich Berechtigungen eine der folgenden Optionen aus:
 - a. Wählen Sie Neue Servicerolle erstellen und verwenden aus. Erstellt standardmäßig AWS IoT SiteWise automatisch eine Servicerolle. Diese Rolle ermöglicht dem AWS IoT SiteWise Assistenten den Zugriff auf Ihre Kendra-Indizes.
 - b. Wählen Sie Bestehende Servicerolle verwenden und wählen Sie dann die Zielrolle aus.
7. Wählen Sie Erstellen aus.

[AWS IoT SiteWise](#) > [Assistant](#) > Create dataset

Create dataset

Create a dataset for the Assistant.

Dataset details [Info](#)

Kendra index

Select a Kendra index for the Assistant dataset.

test-index ▼

[Amazon Kendra](#) [↗](#)



Dataset name

Dataset name is pre-populated from the Kendra index selected. You can modify the dataset name.

Dataset1

Dataset name must be 1-256 characters. Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen).

Dataset description - optional

Dataset description is pre-populated from the Kendra index selected. You can modify the dataset description.

Dataset for AWS IoT SiteWise Assistant

Dataset description must be 1-2048 characters.

Permissions

Assistant must have permissions to access the data. To create a custom role, visit the [IAM console](#) [↗](#)

Choose a method to authorize Assistant

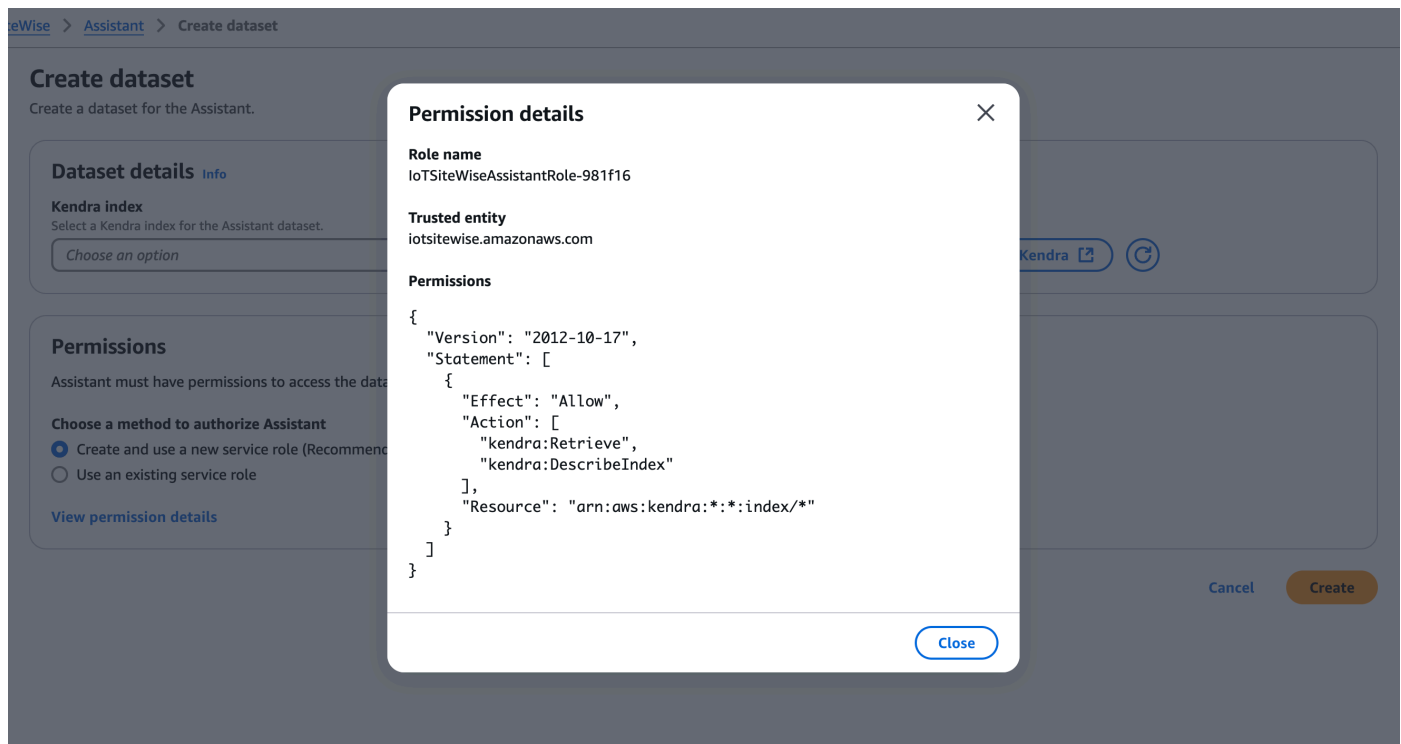
- ☒ Create and use a new service role - *recommended*
- ☐ Use an existing service role

[View permission details](#)

[Cancel](#)

[Create](#)

Die von AWS IoT SiteWise für den Benutzer erstellte Servicerolle, falls der Benutzer eine neue Servicerolle erstellen und verwenden möchte.



AWS CLI

Erstellen Sie einen Datensatz in AWS CLI

1. Erstellen Sie eine IAM-Rolle, die zum Erstellen eines Datensatzes verwendet wird. Verwenden Sie die folgende Berechtigungsrichtlinie:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kendra:Retrieve"
      ],
      "Resource": "arn:aws:kendra:*:*:index/*"
    }
  ]
}
```


Verwenden Sie die folgende Vertrauensbeziehung:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "iotsitewise.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Erstellen Sie eine Datei `create-dataset.json` mit der im Beispiel bereitgestellten Vorlage. Füllen Sie diesen Datensatz aus `kendra knowledgeBaseArn` und stellen Sie `datasetId` eine Verbindung `roleArn` mit ihm her.

```
{
  "datasetId": "<UUID>",
  "datasetName": "DatasetForAssistant",
  "datasetSource": {
    "sourceType": "KENDRA",
    "sourceFormat": "KNOWLEDGE_BASE",
    "sourceDetail": {
      "kendra": {
        "knowledgeBaseArn": "arn:aws:kendra::%s:index/index",
        "roleArn": "arn:aws:iam::%s:role/role"
      }
    }
  }
}
```

3. Erstellen Sie den Datensatz mit dem folgenden Befehl:

```
aws iotsitewise create-dataset --cli-input-json file://create-dataset.json --  
region us-east-1
```

Bearbeiten Sie einen Datensatz

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Console

Bearbeiten Sie einen Datensatz

1. Datensätze werden im Abschnitt Datensätze der Assistentenseite angezeigt. Wählen Sie einen Datensatz zum Bearbeiten aus. Wählen Sie Bearbeiten, um mit der Bearbeitung zu beginnen.
2. Wählen Sie auf der Seite mit den Datensatz-Details einen Kendra-Index aus dem Drop-down-Menü aus, den Sie mit dem Datensatz verknüpfen möchten.
3. Der Datensatzname wird durch den in Schritt 2 ausgewählten Kendra-Index aufgefüllt. Bearbeiten Sie den Namen bei Bedarf.
4. (Optional) Die Datensatzbeschreibung wird mit dem in Schritt 2 ausgewählten Kendra-Index gefüllt. Bearbeiten Sie die Beschreibung bei Bedarf.
5. Wählen Sie im Bereich Berechtigungen eine der folgenden Optionen aus:
 - a. Wählen Sie Neue Servicerolle erstellen und verwenden aus. Erstellt standardmäßig AWS IoT SiteWise automatisch eine Servicerolle. Diese Rolle ermöglicht dem AWS IoT SiteWise Assistenten den Zugriff auf Ihre Kendra-Indizes.
 - b. Wählen Sie Bestehende Servicerolle verwenden und wählen Sie dann die Zielrolle aus.
6. Wählen Sie Änderungen speichern, um Ihre Auswahl zu speichern.

[AWS IoT SiteWise](#) > [Assistant](#) > Edit dataset

Edit dataset

Edit a dataset for the Assistant.

Dataset details [Info](#)

Kendra index

Select a Kendra index for the Assistant dataset.

test-index

[Amazon Kendra](#)



Dataset name

Dataset name is pre-populated from the Kendra index selected. You can modify the dataset name.

Dataset1

Dataset name must be 1-256 characters. Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen).

Dataset description - optional

Dataset description is pre-populated from the Kendra index selected. You can modify the dataset description.

Dataset for AWS IoT SiteWise Assistant

Dataset description must be 1-2048 characters.

Permissions

Assistant must have permissions to access the data. To create a custom role, visit the [IAM console](#).

Choose a method to authorize Assistant

- ☐ Create and use a new service role - *recommended*
- ☒ Use an existing service role

Existing role

IoTSiteWiseAssistantRole-40f994

[View the IoTSiteWiseAssistantRole-40f994 role in the IAM console](#)

[Cancel](#)

[Save changes](#)

AWS CLI

Bearbeiten Sie einen Datensatz in AWS CLI

- Erstellen Sie eine Datei `update-dataset.json` mit der im Beispiel bereitgestellten Vorlage. Füllen Sie diesen Datensatz aus `kendra knowledgeBaseArn` und stellen Sie `datasetId` eine Verbindung `roleArn` mit ihm her.

```
{
  "datasetId": "<UUID>",
  "datasetName": "DatasetForAssistant",
  "datasetSource": {
    "sourceType": "KENDRA",
    "sourceFormat": "KNOWLEDGE_BASE",
    "sourceDetail": {
      "kendra": {
        "knowledgeBaseArn": "arn:aws:kendra::%s:index/index",
```

```
        "roleArn": "arn:aws:iam::%s:role/role"
      }
    }
  }
}
```

2. Aktualisieren Sie den Datensatz mit dem folgenden Befehl:

```
aws iotsitewise update-dataset --cli-input-json file://update-dataset.json --  
region us-east-1
```

Löschen Sie einen Datensatz

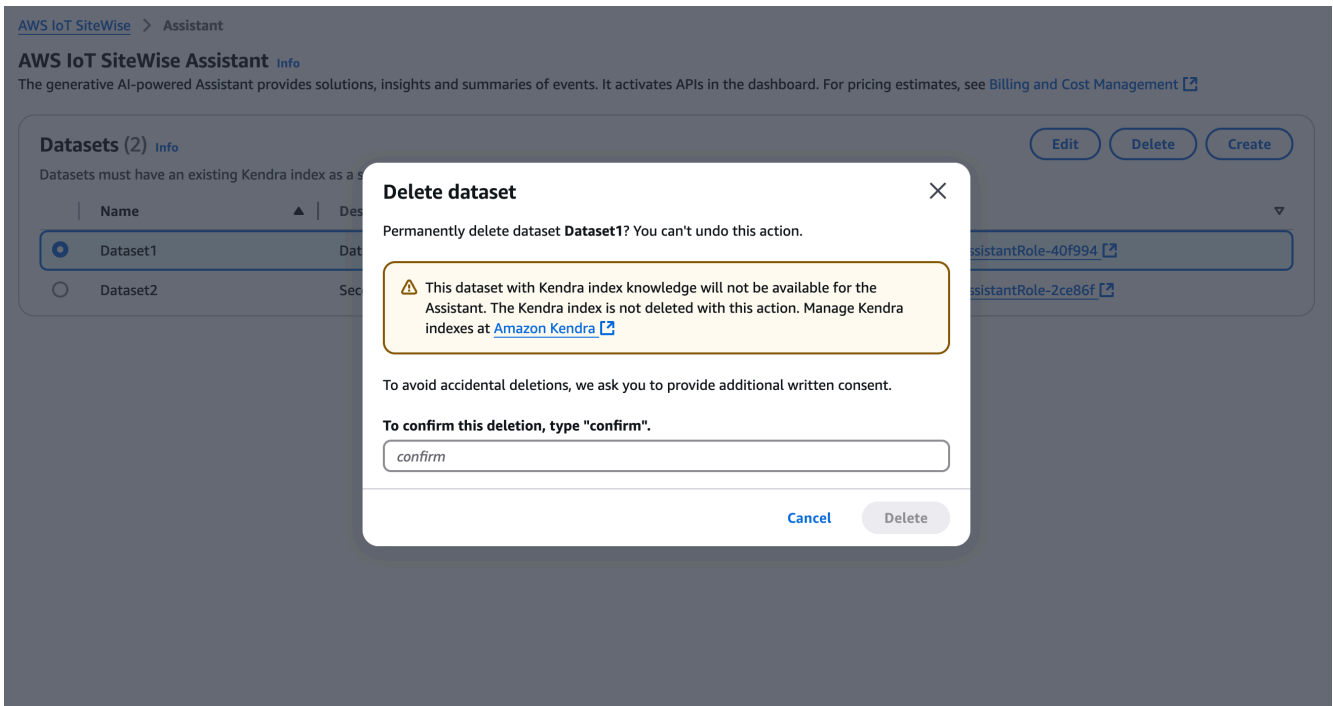
Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Console

Löschen Sie einen Datensatz

1. Datensätze werden im Abschnitt Datensätze der Assistentenseite angezeigt. Wählen Sie einen Datensatz aus. Wählen Sie Löschen aus.
2. Geben Sie in das Popup Bestätigen ein, um das Löschen zu bestätigen.



3. Wählen Sie Löschen aus.

AWS CLI

Löschen Sie einen Datensatz

- Löschen Sie den Datensatz mit `datasetId`.

```
aws iotsitewise delete-dataset --region us-east-1 --dataset-id <UUID>
```

AWS IoT SiteWise Fragen des Assistenten

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Weitere Informationen [Beispielfragen, die Sie dem Assistenten stellen sollten AWS IoT SiteWise](#) zum Abfragen von AWS IoT SiteWise Assistant finden Sie unter.

Überwachen Sie Daten mit AWS IoT SiteWise Monitor

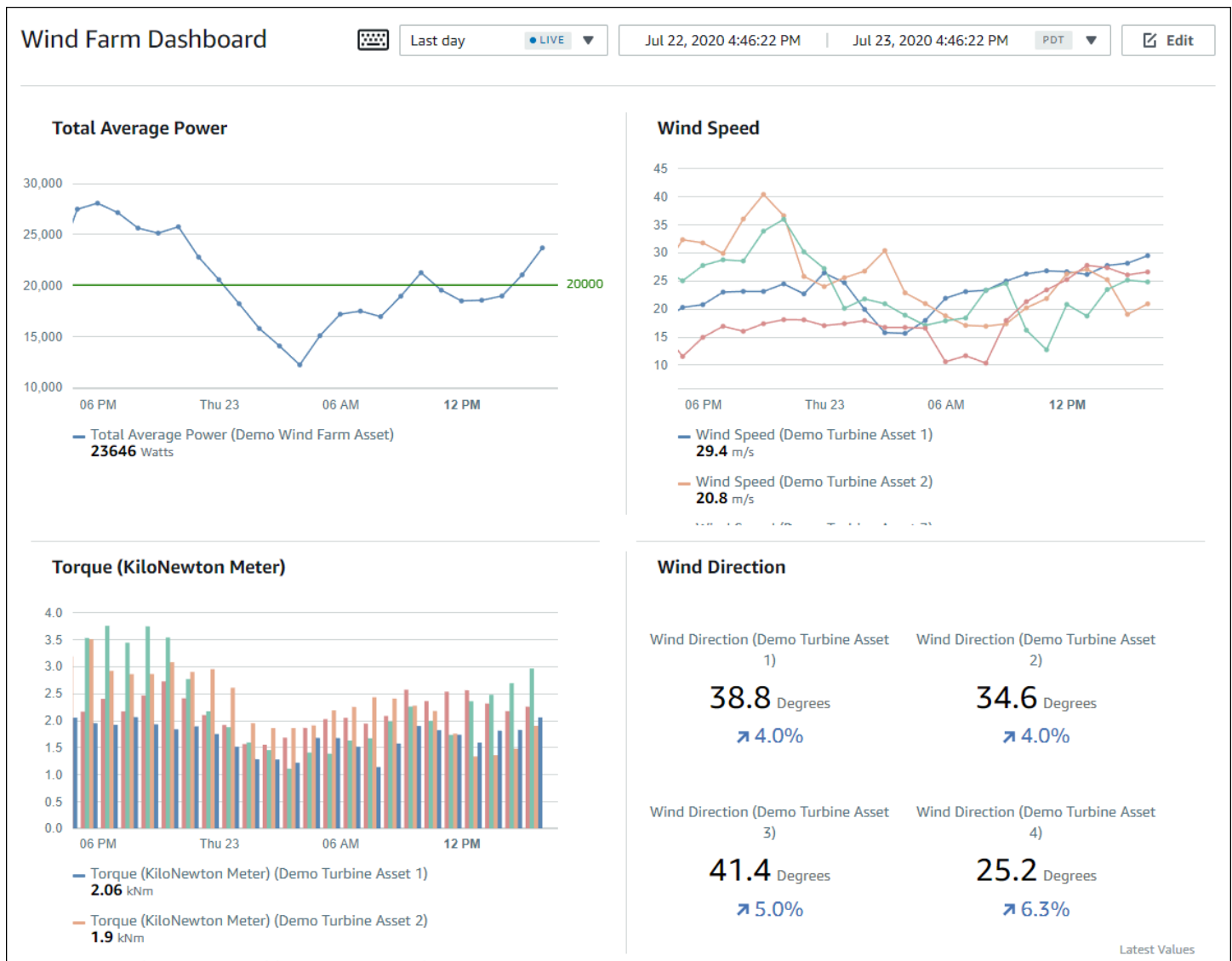
Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Sie können AWS IoT SiteWise verwenden, um die Daten Ihrer Prozesse, Geräte und Anlagen zu überwachen, indem Sie SiteWise Monitor-Webportale erstellen. SiteWise Monitor ist eine Funktion von AWS IoT SiteWise, mit der Sie Portale in Form einer verwalteten Webanwendung erstellen können. Sie können diese Portale dann verwenden, um Ihre Betriebsdaten anzuzeigen und freizugeben. Sie können Projekte mit Dashboards erstellen, um Daten aus Ihren Prozessen, Geräten und Anlagen zu visualisieren, die mit AWS IoT verbunden sind.

Fachexperten, wie z. B. Verfahrenstechniker, können diese Portale nutzen, um schnell Einblicke in ihre Betriebsdaten zu erhalten und damit das Verhalten der Geräte und Anlagen zu verstehen.

Das folgende Beispiel zeigt ein Dashboard, das Daten für eine Windkraftanlage anzeigt.



Da Daten im Zeitverlauf AWS IoT SiteWise erfasst werden, können Sie SiteWise Monitor verwenden, um Betriebsdaten im Zeitverlauf oder die zuletzt gemeldeten Werte zu bestimmten Zeitpunkten anzuzeigen. Auf diese Weise können Sie Erkenntnisse gewinnen, die sonst nur schwer möglich sind.

SiteWise Rollen überwachen

Vier Rollen interagieren mit SiteWise Monitor:

AWS Administrator

Der AWS Administrator verwendet die AWS IoT SiteWise Konsole, um Portale zu erstellen. Der AWS -Administrator kann auch Portaladministratoren zuweisen und Portalbenutzer hinzufügen.

Portaladministratoren weisen Portalbenutzer später Projekten als Eigentümer oder Betrachter zu. Der AWS Administrator arbeitet ausschließlich in der AWS Konsole.

Portaladministrator

Jedes SiteWise Monitor-Portal hat einen oder mehrere Portaladministratoren.

Portaladministratoren verwenden das Portal, um Projekte zu erstellen, die Sammlungen von Komponenten und Dashboards enthalten. Der Portaladministrator weist dann jedem Projekt Komponenten und Eigentümer zu. Durch die Steuerung des Zugriffs auf das Projekt legen Portaladministratoren fest, welche Komponenten von Projekteigentümern und -betrachtern angezeigt werden können.

Projekteigentümer

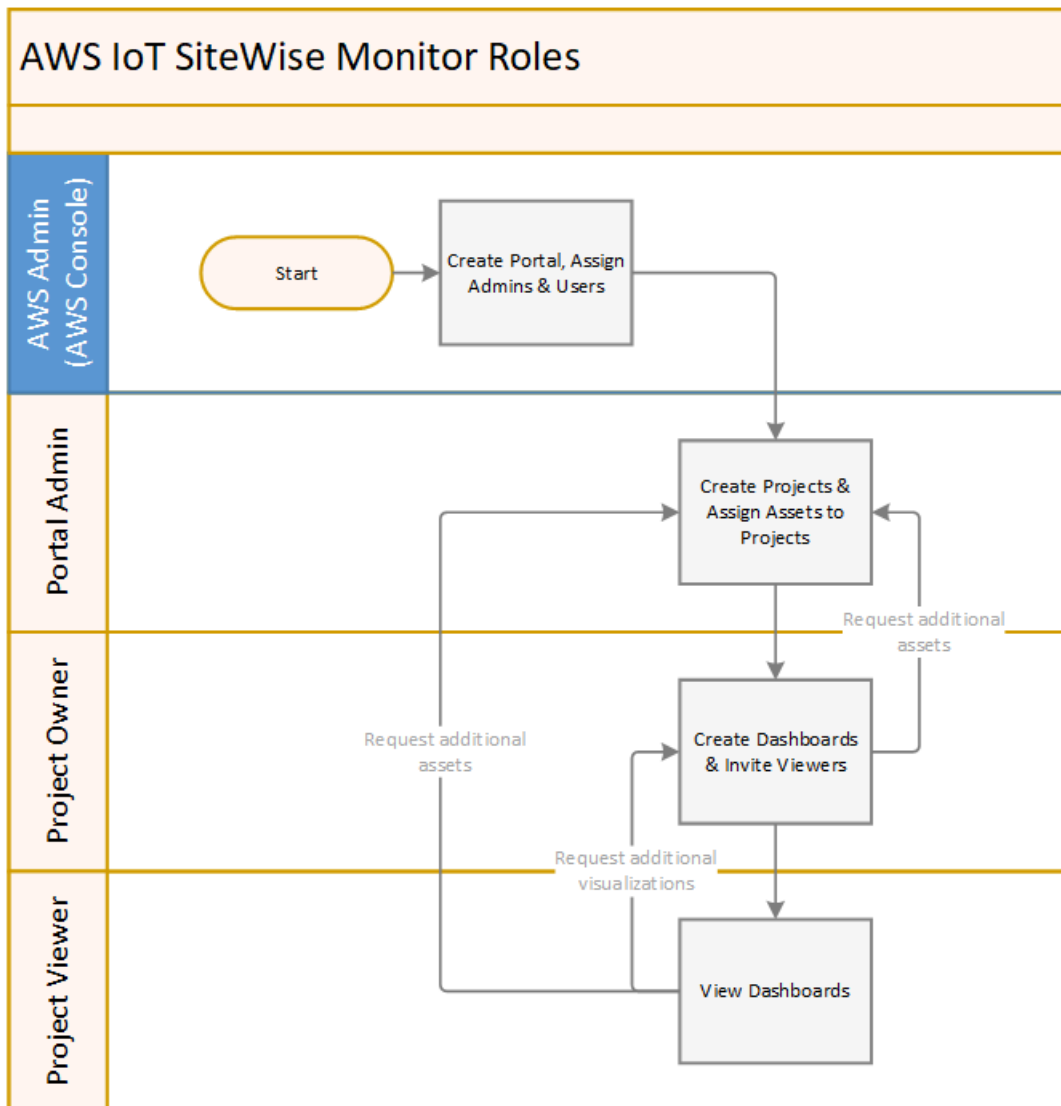
Jedes SiteWise Monitor-Projekt hat Besitzer. Projekteigentümer erstellen Visualisierungen in Form von Dashboards, um Betriebsdaten konsistent darzustellen. Wenn Dashboards zur Freigabe bereit sind, kann der Projekteigentümer Betrachter zu dem Projekt einladen. Projekteigentümer können dem Projekt auch andere Eigentümer zuweisen. Projekteigentümer können Schwellenwerte und Benachrichtigungseinstellungen für Alarmer konfigurieren.

Projektbetrachter

Jedes SiteWise Monitor-Projekt hat Zuschauer. Projektbetrachter können eine Verbindung mit dem Portal herstellen, um die Dashboards anzuzeigen, die Projekteigentümer erstellt haben. In jedem Dashboard können Projektbetrachter den Zeitraum anpassen, um die Betriebsdaten besser zu verstehen. Projektbetrachter können nur Dashboards in den Projekten anzeigen, auf die sie Zugriff haben. Projektbeobachter können Alarmer bestätigen und die Schlummerfunktion aktivieren.

Je nach Organisation kann dieselbe Person mehrere Rollen ausführen.

Die folgende Abbildung zeigt, wie diese vier Rollen im SiteWise Monitor-Portal interagieren.



Sie können mithilfe von AWS IAM Identity Center oder IAM verwalten, wer Zugriff auf Ihre Daten hat. Ihre Datennutzer können sich von einem Desktop- oder mobilen Browser aus mit ihren IAM Identity Center- oder IAM-Anmeldeinformationen bei SiteWise Monitor anmelden.

SAML-Verbund

IAM Identity Center und IAM unterstützen den Identitätsverbund mit [SAML \(Security Assertion Markup Language\) 2.0](#). SAML 2.0 ist ein offener Standard, den viele externe Identitätsanbieter (IdPs) verwenden, um Benutzer zu authentifizieren und ihre Identitäts- und Sicherheitsinformationen an Dienstleister weiterzugeben (). SPs sind in der Regel Anwendungen oder Dienste. Der SAML-Verbund ermöglicht es Ihren SiteWise Monitor-Portaladministratoren und -Benutzern, sich mit externen Anmeldeinformationen, wie z. B. ihren Firmenbenutzernamen und Kennwörtern, bei den ihnen zugewiesenen Portalen anzumelden.

Sie können IAM Identity Center und IAM so konfigurieren, dass sie den SAML-basierten Verbund für den Zugriff auf Ihre Monitor-Portale verwenden. SiteWise

IAM Identity Center

Ihre Portaladministratoren und Benutzer können sich mit ihren Firmenbenutzernamen und AWS Kennwörtern beim Access-Portal anmelden. Sie können dann zu den ihnen zugewiesenen SiteWise Monitor-Portalen navigieren. IAM Identity Center verwendet Zertifikate, um eine SAML-Vertrauensbeziehung zwischen Ihrem Identitätsanbieter und einzurichten.

AWSWeitere Informationen zum [SCIM-Profil und zur SAML 2.0-Implementierung finden Sie im Benutzerhandbuch](#).AWS IAM Identity Center

IAM

Ihre Portaladministratoren und Benutzer können temporäre Sicherheitsanmeldedaten anfordern, um auf die ihnen zugewiesenen SiteWise Monitor-Portale zuzugreifen. Sie erstellen eine SAML-Identitätsanbieter-Identität in IAM, um eine Vertrauensbeziehung zwischen Ihrem Identitätsanbieter und einzurichten. AWSWeitere Informationen finden Sie im [IAM-Benutzerhandbuch unter Verwenden eines SAML-basierten Verbunds für den API-Zugriff AWS auf](#).

Ihre Portaladministratoren und Benutzer können sich beim Portal Ihres Unternehmens anmelden und die Option auswählen, um zur Management-Konsole zu wechseln. AWS Sie können dann zu den ihnen zugewiesenen SiteWise Monitor-Portalen navigieren. Das Portal Ihres Unternehmens kümmert sich um den Vertrauensaustausch zwischen Ihrem Identitätsanbieter und AWS. Weitere Informationen finden Sie unter [Aktivieren des Zugriffs auf die AWS Management Console durch SAML 2.0-Verbundbenutzer](#) im IAM-Benutzerhandbuch.

Note

Vermeiden Sie beim Hinzufügen von Benutzern oder Administratoren zum Portal die Erstellung von IAM-Richtlinien, die Benutzerberechtigungen einschränken, z. B. eingeschränkte IP-Adressen. Alle angehängten Richtlinien mit eingeschränkten Berechtigungen können keine Verbindung zum AWS IoT SiteWise Portal herstellen.

SiteWise Konzepte überwachen

Um SiteWise Monitor verwenden zu können, sollten Sie mit den folgenden Konzepten vertraut sein:

Portal

Ein AWS IoT SiteWise Monitor Portal ist eine Webanwendung, mit der Sie Ihre AWS IoT SiteWise Daten visualisieren und gemeinsam nutzen können. Ein Portal verfügt über einen oder mehrere Administratoren und enthält keine oder mehrere Projekte.

Projekt

Jedes SiteWise Monitor-Portal enthält eine Reihe von Projekten. Jedem Projekt ist eine Teilmenge Ihrer AWS IoT SiteWise -Komponenten zugeordnet. Projekteigentümer erstellen ein oder mehrere Dashboards, um eine konsistente Möglichkeit zum Anzeigen der mit diesen Komponenten verknüpften Daten bereitzustellen. Projekteigentümer können Betrachter zu dem Projekt einladen, damit diese die Komponenten und Dashboards in dem Projekt anzeigen können. Das Projekt ist die grundlegende Einheit für die gemeinsame Nutzung innerhalb von SiteWise Monitor. Projekteigentümer können Benutzer einladen, denen der AWS Administrator Zugriff auf das Portal gewährt hat. Ein Benutzer muss Zugriff auf ein Portal haben, bevor ein Projekt in diesem Portal für diesen Benutzer freigegeben werden kann.

Komponente

Wenn Daten AWS IoT SiteWise aus Ihren Industrieanlagen aufgenommen werden, werden Ihre Geräte, Anlagen und Prozesse jeweils als Vermögenswerte dargestellt. Jeder Anlage sind Eigenschaften und Alarmer zugeordnet. Der Portaladministrator weist jedem Projekt Komponenten zu.

Eigenschaft

Eigenschaften sind Zeitreihendaten, die Objekten zugeordnet sind. Beispielsweise kann ein Gerät eine Seriennummer, einen Standort, eine Marke und ein Modell sowie ein Installationsdatum aufweisen. Es kann auch Zeitreihenwerte für Verfügbarkeit, Leistung, Qualität, Temperatur, Druck usw. enthalten.

Alarm

Alarmer überwachen die Eigenschaften, um zu erkennen, wenn sich Geräte außerhalb ihres Betriebsbereichs befinden. Jeder Alarm definiert einen Schwellenwert und eine zu überwachende Eigenschaft. Wenn die Eigenschaft den Schwellenwert überschreitet, wird der Alarm aktiv und weist darauf hin, dass Sie oder jemand aus Ihrem Team das Problem beheben sollten. Projekteigentümer können die Schwellenwerte und Benachrichtigungseinstellungen für Alarmer anpassen. Projektbeobachter können Alarmer bestätigen und die Schlummerfunktion aktivieren. Außerdem können sie eine Nachricht mit Einzelheiten zum Alarm oder zu den Maßnahmen hinterlassen, die sie zu seiner Behebung ergriffen haben.

Dashboard

Jedes Projekt enthält eine Reihe von Dashboards. Dashboards stellen eine Reihe von Visualisierungen für die Werte einer Gruppe von Komponenten bereit. Projekteigentümer erstellen die Dashboards und die darin enthaltenen Visualisierungen. Wenn ein Projekteigentümer bereit ist, die Gruppe von Dashboards freizugeben, kann der Eigentümer Betrachter zu dem Projekt einladen, wodurch diese Zugriff auf alle Dashboards in dem Projekt erhalten. Wenn Sie eine andere Gruppe von Betrachtern für verschiedene Dashboards wünschen, müssen Sie die Dashboards auf Projekte aufteilen. Wenn sich Zuschauer Dashboards ansehen, können sie den Zeitraum so anpassen, dass sie sich bestimmte Daten ansehen.

Visualisierung

In jedem Dashboard entscheiden die Projekteigentümer, wie die Eigenschaften und Alarmer der mit dem Projekt verknüpften Assets angezeigt werden sollen. Die Verfügbarkeit kann als Liniendiagramm dargestellt werden, während andere Werte als Balkendiagramme oder wichtige Leistungsindikatoren (KPIs) dargestellt werden können. Alarmer lassen sich am besten als Statusraster und Statuszeitleisten anzeigen. Projekteigentümer passen jede Visualisierung an, um die Daten für diese Komponente optimal darzustellen.

Erste Schritte mit AWS IoT SiteWise Monitor (Classic)

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Wenn Sie der AWS Administrator Ihrer Organisation sind, erstellen Sie Portale von der AWS IoT SiteWise Konsole aus. Gehen Sie wie folgt vor, um ein Portal zu erstellen, damit Mitglieder Ihrer Organisation Ihre AWS IoT SiteWise Daten einsehen können:

1. Konfigurieren und erstellen Sie ein Portal.
2. Fügen Sie Portaladministratoren hinzu, und senden Sie Einladungs-E-Mail-Nachrichten.
3. Fügen Sie Portalbenutzer hinzu

Nachdem Sie ein Portal erstellt haben, kann der Portaladministrator Ihre AWS IoT SiteWise Ressourcen anzeigen und sie Projekten im Portal zuweisen. Projekteigentümer können dann Dashboards erstellen, um die Eigenschaften der Komponenten zu visualisieren und den Projektbetrachtern damit ein besseres Verständnis der Leistung Ihrer Geräte, Prozesse und Anlagen zu ermöglichen.

 Note

Vermeiden Sie beim Hinzufügen von Benutzern oder Administratoren zum Portal die Erstellung von AWS Identity and Access Management (IAM-) Richtlinien, die Benutzerberechtigungen einschränken, wie z. B. eingeschränkte IP-Adressen. Alle angehängten Richtlinien mit eingeschränkten Berechtigungen können keine Verbindung zum AWS IoT SiteWise Portal herstellen.

Sie können einem Tutorial folgen, das die Schritte durchläuft, die zum Einrichten eines Portals mit einem Projekt, Dashboards und mehreren Benutzern für ein bestimmtes Szenario unter Verwendung von Windparkdaten erforderlich sind. Weitere Informationen finden Sie unter [Visualisieren und teilen Sie Windparkdaten in SiteWise Monitor](#).

Themen

- [Erstellen Sie ein Portal in SiteWise Monitor](#)
- [Konfigurieren Sie Ihr Portal in SiteWise Monitor](#)
- [Laden Sie Administratoren in SiteWise Monitor ein](#)
- [Fügen Sie Portalbenutzer in SiteWise Monitor hinzu](#)
- [AWS IoT SiteWise Dashboards erstellen \(AWS CLI\)](#)
- [Schalten Sie Alarme für Ihre Portale ein in AWS IoT SiteWise](#)
- [Aktivierung Ihres AWS IoT SiteWise Portals am Edge](#)
- [Verwalten Sie Ihre SiteWise Monitor-Portale](#)

Erstellen Sie ein Portal in SiteWise Monitor

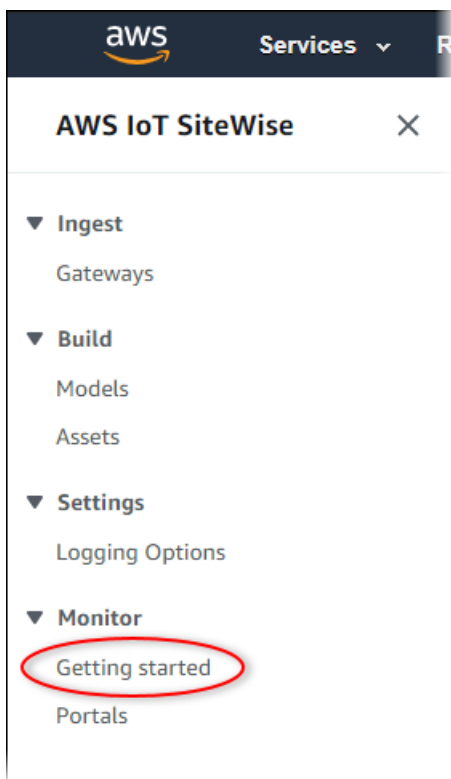
Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

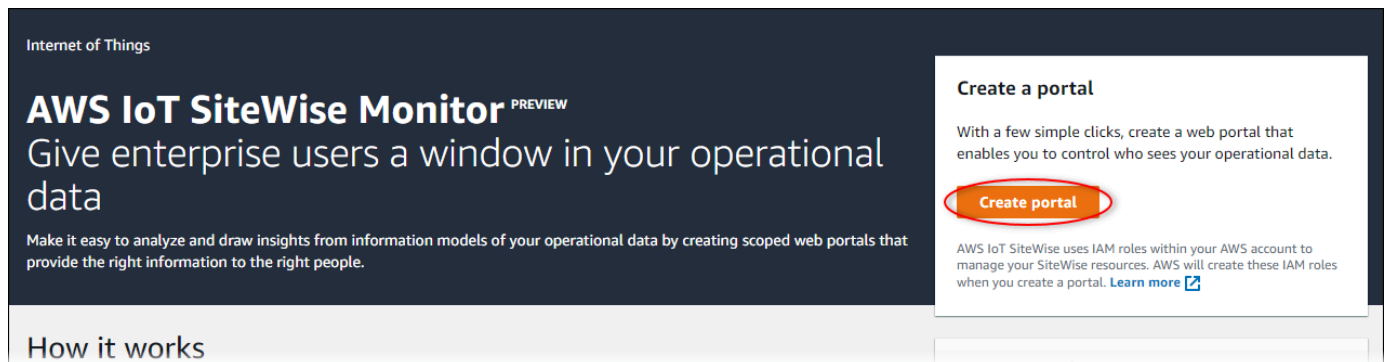
Sie erstellen ein SiteWise Monitor-Portal in der AWS IoT SiteWise Konsole.

So erstellen Sie ein Portal

1. Melden Sie sich bei der [AWS IoT SiteWise -Konsole](#) an.
2. Wählen Sie im Navigationsbereich Monitor (Überwachen), Getting started (Erste Schritte) aus.



3. Wählen Sie Create Portal (Portal erstellen) aus.



Als Nächstes müssen Sie einige grundlegende Informationen zur Konfiguration des Portals angeben.

Konfigurieren Sie Ihr Portal in SiteWise Monitor

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Ihre Benutzer verwenden Portale, um Ihre Daten anzuzeigen. Sie können den Namen, die Beschreibung, das Branding, die Benutzerauthentifizierung, die E-Mail-Adresse des Support-Kontakts und die Berechtigungen eines Portals anpassen.

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configurationStep 2- optional
Additional featuresStep 3
Invite administratorsStep 4
Assign users

Portal configuration

Each web portal provides enterprise users with access to your IoT SiteWise assets. [Learn more](#)

Portal details

Portal name

Choose a portal name to identify the web portal to your users. Company name is recommended.

example-factory-1

Name should be 1-128 characters and only contain A-Z a-z 0-9 _ and -.

Description - optional

Create a description of your portal

Example Corp Factory #1 in Renton, WA

Description should contain a maximum of 2048 characters.

Portal branding

You can provide your logo image to display your brand in this web portal.

Logo image

Upload a square, high-resolution .png file. The image is displayed on a dark background.

Choose file

The file size must be less than 1 MB.

User authentication

Your users can sign in to this portal with their AWS Single Sign-On (AWS SSO) or AWS Identity and Access Management (IAM) credentials. If you choose AWS SSO, you must enable the service for your AWS account.

 You haven't enabled AWS SSO in your account yet. When you create your first portal user, this automatically enables AWS SSO in your AWS account.

Create user

☒ AWS SSO

Your users can sign in to the portal with their corporate usernames and passwords.

☐ IAM

Your users can sign in to the portal with their IAM credentials.

Support contact email

You can provide an email address for cases where there's a problem or issue with this portal and your users need to contact support to resolve.

Email

support@example.com

Tags

This resource doesn't have any tags.

Add tag

You can add up to 50 more tags.

Permissions

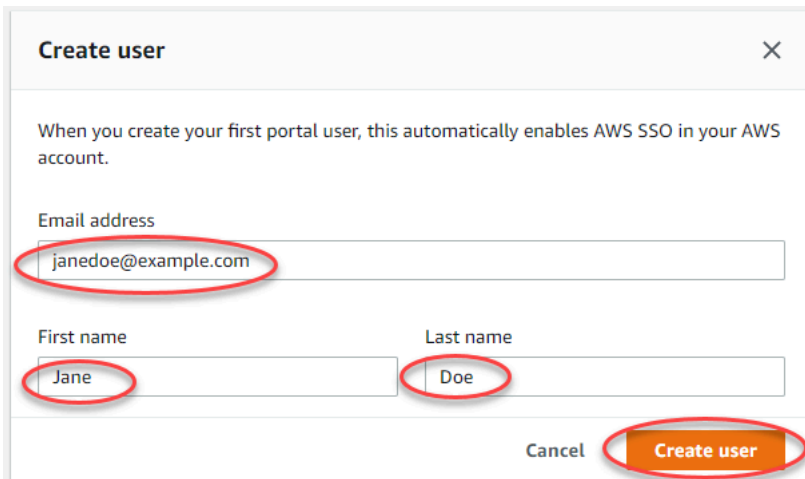
SiteWise Monitor assumes this role to give permissions to your federated users to access AWS IoT SiteWise resources. [Learn](#)

So konfigurieren Sie ein Portal:

1. Geben Sie einen Namen für Ihr Portal ein.
2. (Optional) Geben Sie eine Beschreibung für Ihr Portal ein. Wenn Sie über mehrere Portale verfügen, verwenden Sie aussagekräftige Beschreibungen, um den Überblick über die Inhalte der einzelnen Portale zu behalten.
3. (Optional) Laden Sie ein Bild hoch, um Ihre Marke im Portal anzuzeigen. Wählen Sie ein quadratisches PNG-Bild aus. Wenn Sie ein nicht quadratisches Bild hochladen, skaliert das Portal das Bild zu einem Quadrat.
4. Wählen Sie eine der folgenden Optionen:
 - Wählen Sie IAM Identity Center, wenn sich Ihre Portalbenutzer mit ihren Firmenbenutzernamen und Passwörtern bei diesem Portal anmelden.

Wenn Sie IAM Identity Center in Ihrem Konto nicht aktiviert haben, gehen Sie wie folgt vor:

- a. Wählen Sie Create user (Benutzer erstellen) aus.
- b. Um das erste Portal zu erstellen, geben Sie auf der Seite Benutzer erstellen die E-Mail-Adresse, den Vor- und Nachnamen des Benutzers ein und wählen Sie dann Benutzer erstellen aus.



Note

- AWS aktiviert IAM Identity Center automatisch in Ihrem Konto, wenn Sie den ersten Portalbenutzer erstellen.
- Sie können IAM Identity Center jeweils nur in einer Region konfigurieren. SiteWise Monitor stellt eine Verbindung zu der Region her, die Sie für IAM

Identity Center konfiguriert haben. Das bedeutet, dass Sie eine Region für den Zugriff auf das IAM Identity Center verwenden, aber Sie können Portale in jeder Region erstellen.

- Wählen Sie IAM, wenn sich Ihre Portalbenutzer mit ihren IAM-Anmeldeinformationen bei diesem Portal anmelden.

 Important

Benutzer oder Rollen müssen über die `iotsitewise:DescribePortal` Berechtigung verfügen, sich beim Portal anzumelden.

5. Geben Sie eine E-Mail-Adresse ein, die Portalbenutzer bei Problemen mit dem Portal kontaktieren können, wenn sie Hilfe bei der Fehlerbehebung benötigen.
6. (Optional) Fügen Sie Tags für Ihr Portal hinzu. Weitere Informationen finden Sie unter [Kennzeichnen Sie Ihre AWS IoT SiteWise Ressourcen](#).
7. Wählen Sie eine der folgenden Optionen:
 - Wählen Sie Neue Servicerolle erstellen und verwenden aus. Standardmäßig erstellt SiteWise Monitor automatisch eine Servicerolle für jedes Portal. Diese Rolle ermöglicht Ihren Portalbenutzern den Zugriff auf Ihre AWS IoT SiteWise Ressourcen. Weitere Informationen finden Sie unter [Verwenden Sie Servicerollen für AWS IoT SiteWise Monitor](#).
 - Wählen Sie Bestehende Servicerolle verwenden und wählen Sie dann die Zielrolle aus.
8. Wählen Sie Weiter
9. (Optional) Aktivieren Sie Alarme für Ihr Portal. Weitere Informationen finden Sie unter [Schalten Sie Alarme für Ihre Portale ein in AWS IoT SiteWise](#).
10. Wählen Sie Erstellen. AWS IoT SiteWise wird Ihr Portal erstellen.

 Note

Wenn Sie die Konsole schließen, können Sie zum Abschluss der Einrichtung Administratoren und Benutzer hinzufügen. Weitere Informationen finden Sie unter [Fügen Sie Portaladministratoren hinzu oder entfernen Sie sie in AWS IoT SiteWise](#). Wenn Sie dieses Portal nicht behalten möchten, löschen Sie es, damit es keine Ressourcen

verbraucht. Weitere Informationen finden Sie unter [Löschen Sie ein Portal in AWS IoT SiteWise](#).

Die Spalte Status kann einen der folgenden Werte haben.

- **ERSTELLEN** — AWS IoT SiteWise bearbeitet Ihre Anfrage zur Erstellung des Portals. Dieser Vorgang kann mehrere Minuten in Anspruch nehmen.
- **AKTUALISIERUNG** — AWS IoT SiteWise bearbeitet Ihre Anfrage zur Aktualisierung des Portals. Dieser Vorgang kann mehrere Minuten in Anspruch nehmen.
- **AUSSTEHEND** — AWS IoT SiteWise wartet darauf, dass die Weitergabe des DNS-Eintrags abgeschlossen ist. Dieser Vorgang kann mehrere Minuten in Anspruch nehmen. Sie können das Portal löschen, solange der Status AUSSTEHEND ist.
- **LÖSCHEN** — AWS IoT SiteWise bearbeitet Ihre Anfrage zum Löschen des Portals. Dieser Vorgang kann mehrere Minuten in Anspruch nehmen.
- **AKTIV** — Wenn das Portal aktiv wird, können Ihre Portalbenutzer darauf zugreifen.
- **FEHLGESCHLAGEN** — Ihre Anfrage zum Erstellen, Aktualisieren oder Löschen des Portals AWS IoT SiteWise konnte nicht bearbeitet werden. Wenn Sie AWS IoT SiteWise das Senden von Protokollen an Amazon CloudWatch Logs aktiviert haben, können Sie diese Protokolle zur Behebung von Problemen verwenden. Weitere Informationen finden Sie unter [Überwachung AWS IoT SiteWise mit CloudWatch Protokollen](#).

Wenn Ihr Portal erstellt ist, wird eine Meldung angezeigt.

☑ Successfully created portal URL at https://a1b2c3d4-5678-90ab-cdef-1111EXAMPLE.app.iotsitewise.aws



Als Nächstes müssen Sie einen oder mehrere Portaladministratoren zum Portal einladen. Bislang haben Sie nur ein Portal erstellt, es kann noch niemand darauf zugreifen.

Laden Sie Administratoren in SiteWise Monitor ein

Um mit dem neuen Portal zu beginnen, müssen Sie einen Portaladministrator zuweisen. Der Portaladministrator erstellt Projekte, wählt Projekteigentümer aus und weist Projekten Komponenten zu. Portaladministratoren können all Ihre AWS IoT SiteWise Ressourcen sehen.

Wählen Sie je nach Benutzerauthentifizierungsdienst eine der folgenden Optionen aus:

IAM Identity Center

Wenn Sie SiteWise Monitor zum ersten Mal verwenden, können Sie den Benutzer, den Sie zuvor erstellt haben, als Portaladministrator auswählen. Wenn Sie einen weiteren Benutzer als Portaladministrator hinzufügen möchten, können Sie auf dieser Seite einen IAM Identity Center-Benutzer erstellen. Alternativ können Sie einen externen Identitätsanbieter mit IAM Identity Center verbinden. Weitere Informationen finden Sie im [AWS IAM Identity Center -Benutzerhandbuch](#).

So laden Sie Administratoren ein

1. Aktivieren Sie die Kontrollkästchen für die Benutzer, die Ihre Portaladministratoren sein sollen. Dadurch werden die Benutzer zur Liste der Portaladministratoren hinzugefügt.

Note

Wenn Sie IAM Identity Center als Identitätsspeicher verwenden und bei Ihrem AWS Organizations Verwaltungskonto angemeldet sind, können Sie Benutzer erstellen wählen, um einen IAM Identity Center-Benutzer zu erstellen. IAM Identity Center sendet dem neuen Benutzer eine E-Mail, damit er sein Passwort einrichten kann. Anschließend können Sie den Benutzer dem Portal als Administrator zuweisen. Weitere Informationen finden Sie unter [Identitäten im IAM Identity Center verwalten](#).

2. (Optional) Wählen Sie Send invite to selected users (Einladung an ausgewählte Benutzer senden) aus. Ihr E-Mail-Client wird geöffnet und der Nachrichtentext wird mit einer Einladung gefüllt.

Sie können die E-Mail anpassen, bevor Sie sie an die Portaladministratoren senden. Sie können die E-Mail-Nachricht auch später an Ihre Portaladministratoren senden. Wenn Sie SiteWise Monitor zum ersten Mal ausprobieren und Ihren neuen IAM Identity Center- oder IAM-Benutzer oder Ihre neue Rolle als Portaladministrator hinzufügen, müssen Sie sich keine E-Mail senden.

3. Wenn Sie einen Benutzer hinzufügen, der nicht Administrator sein soll, deaktivieren Sie das Kontrollkästchen für den betreffenden Benutzer.
4. Wenn Sie mit dem Einladen von Portaladministratoren fertig sind, wählen Sie Next (Weiter) aus.

IAM

Sie können einen Benutzer oder eine Rolle als Portaladministrator auswählen. Wenn Sie einen weiteren Benutzer oder eine weitere Rolle als Portaladministrator hinzufügen möchten, können Sie einen Benutzer oder eine Rolle in der IAM-Konsole erstellen. Weitere Informationen finden Sie unter [Einen IAM-Benutzer in Ihrem AWS Konto erstellen und IAM-Rollen erstellen im IAM-Benutzerhandbuch](#).

So laden Sie Administratoren ein

1. Gehen Sie wie folgt vor:
 - Wählen Sie IAM-Benutzer aus, um einen IAM-Benutzer als Portaladministrator hinzuzufügen.
 - Wählen Sie IAM-Rollen aus, um eine IAM-Rolle als Portaladministrator hinzuzufügen.
2. Aktivieren Sie die Kontrollkästchen für die Benutzer oder Rollen, die Sie als Portaladministratoren verwenden möchten. Dadurch werden die Benutzer oder Rollen zur Liste der Portaladministratoren hinzugefügt.
3. Wenn Sie einen Benutzer oder eine Rolle hinzufügen, die Sie nicht als Administrator verwenden möchten, deaktivieren Sie das Kontrollkästchen für diesen Benutzer oder diese Rolle.
4. Wenn Sie mit dem Einladen von Portaladministratoren fertig sind, wählen Sie Next (Weiter) aus.

Important

Benutzer oder Rollen müssen über die `iotsitewise:DescribePortal` Berechtigung verfügen, sich beim Portal anzumelden.

Note

Wenn Sie IAM Identity Center als Identitätsspeicher verwenden und bei Ihrem AWS Organizations Verwaltungskonto angemeldet sind, können Sie Benutzer erstellen wählen, um einen IAM Identity Center-Benutzer zu erstellen. IAM Identity Center sendet dem neuen Benutzer eine E-Mail, damit er sein Passwort einrichten kann. Anschließend

können Sie den Benutzer dem Portal als Administrator zuweisen. Weitere Informationen finden Sie unter [Identitäten im IAM Identity Center verwalten](#).

Sie können die Liste der Portaladministratoren später ändern. Weitere Informationen finden Sie unter [Fügen Sie Portaladministratoren hinzu oder entfernen Sie sie in AWS IoT SiteWise](#).

Note

Da nur ein Portaladministrator Projekte erstellen und ihnen Ressourcen zuweisen kann, sollten Sie mindestens einen Portaladministrator angeben.

Als letzten Schritt fügen Sie Benutzer hinzu, die auf Ihr neues Portal zugreifen können.

Fügen Sie Portalbenutzer in SiteWise Monitor hinzu

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Sie steuern, welche Benutzer Zugriff auf Ihr Portal haben. In jedem Portal erstellen die Portaladministratoren ein oder mehrere Projekte und weisen Portalbenutzer für jedes Projekt als Eigentümer oder Betrachter zu. Jeder Projekteigentümer kann zusätzliche Portalbenutzer als Projekteigentümer oder -betrachter einladen.

Wählen Sie je nach Benutzerauthentifizierungsdienst eine der folgenden Optionen aus:

IAM Identity Center

Wenn Sie der Benutzerliste einen Benutzer hinzufügen möchten, führen Sie die folgenden Schritte aus.

So fügen Sie Portalbenutzer hinzu

1. Wählen Sie Benutzer aus der Benutzerliste aus, die Sie dem Portal hinzufügen möchten. Dadurch werden die Benutzer zur Liste der Portalbenutzer hinzugefügt. Wenn Sie SiteWise

Monitor zum ersten Mal verwenden, müssen Sie Ihren Portaladministrator nicht als Portalbenutzer hinzufügen.

Note

Wenn Sie IAM Identity Center als Identitätsspeicher verwenden und bei Ihrem AWS Organizations Verwaltungskonto angemeldet sind, können Sie Benutzer erstellen wählen, um einen IAM Identity Center-Benutzer zu erstellen. IAM Identity Center sendet dem neuen Benutzer eine E-Mail, damit er sein Passwort einrichten kann. Anschließend können Sie den Benutzer dem Portal als Benutzer zuweisen. Weitere Informationen finden Sie unter [Identitäten im IAM Identity Center verwalten](#).

2. Wenn Sie einen Benutzer hinzufügen, der keinen Zugriff auf das Portal erhalten soll, deaktivieren Sie das Kontrollkästchen für den betreffenden Benutzer.
3. Wenn Sie mit der Auswahl der Benutzer fertig sind, wählen Sie Benutzer zuweisen.

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2
[Invite administrators](#)

Step 3
Assign users

Assign users

Select the users you want to be able to access and view this portal. Portal administrators will send invitations to these users at a later date. [Learn more](#)

Users (2) Create user

	Display name	Email
☐	Jane Doe	janedoe@example.com
☒	John Doe	johndoe@example.com

► Selected users (1)

Cancel Previous **Assign users**

IAM

Wenn Sie den Benutzer oder die Rolle, die Sie hinzufügen möchten, in der Liste der IAM-Benutzer oder IAM-Rollen sehen, führen Sie die folgenden Schritte aus.

So fügen Sie Portalbenutzer hinzu

1. Führen Sie die folgenden Optionen aus:

- Wählen Sie IAM-Benutzer aus, um einen IAM-Benutzer als Portalbenutzer hinzuzufügen.
- Wählen Sie IAM-Rollen aus, um eine IAM-Rolle als Portalbenutzer hinzuzufügen.

Wenn Sie SiteWise Monitor zum ersten Mal verwenden, müssen Sie Ihren Portaladministrator nicht als Portalbenutzer hinzufügen.

2. Aktivieren Sie die Kontrollkästchen für die Benutzer oder Rollen, die Sie als Portalbenutzer verwenden möchten. Dadurch werden die Benutzer oder Rollen zur Liste der Portalbenutzer hinzugefügt.
3. Wenn Sie einen Benutzer hinzufügen, der keinen Zugriff auf das Portal erhalten soll, deaktivieren Sie das Kontrollkästchen für den betreffenden Benutzer.
4. Wenn Sie mit der Auswahl der Benutzer fertig sind, wählen Sie Benutzer zuweisen.

Important

Benutzer oder Rollen müssen über die `iotsitewise:DescribePortal` Berechtigung verfügen, sich beim Portal anzumelden.

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2
[Invite administrators](#)

Step 3
Assign users

Assign users

Select the users you want to be able to access and view this portal. Portal administrators will send invitations to these users at a later date. [Learn more](#)

Users

Roles

IAM users (1) [Manage users in IAM console](#)

Find user name

< 1 >

☒	Name	Date created
☒	raspberrypi-testing	11-08-2019

► Portal users (1)

Remove

Cancel

Previous

Assign users

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2
Invite administrators

Step 3
Assign users

Assign users

Select the users you want to be able to access and view this portal. Portal administrators will send invitations to these users at a later date. [Learn more](#)

Users **Roles**

IAM roles (66) [Manage roles in IAM console](#)

Find role name

	Name	Date created
☐	[REDACTED]	
☐	[REDACTED]	
☒	AWSIoTSiteWiseMonitorServiceRole_4wZigNpA1	03-16-2021
☐	AWSIoTSiteWiseMonitorServiceRole_ECKT-2Oar	03-11-2021
☐	AWSIoTSiteWiseMonitorServiceRole_GTnd004Wr	03-16-2021
☐	AWSIoTSiteWiseMonitorServiceRole_rHINLNCS-	03-11-2021
☐	AWSIoTSiteWiseMonitorServiceRole_XB330QUIO	03-10-2021
☐	[REDACTED]	
☐	[REDACTED]	
☐	[REDACTED]	

► Portal users (2) Remove

Cancel Previous **Assign users**

Herzlichen Glückwunsch! Sie haben erfolgreich ein Portal erstellt, Portaladministratoren zugewiesen und Benutzern zugewiesen, die dieses Portal nutzen können, wenn sie dazu aufgefordert werden. Ihre Portaladministratoren können jetzt Projekte erstellen und diesen Projekten Komponenten zuweisen. Anschließend können Ihre Projekteigentümer Dashboards erstellen, um die Daten für die Komponenten der einzelnen Projekte zu visualisieren.

Sie können die Liste der Portalbenutzer später ändern. Weitere Informationen finden Sie unter [Fügen Sie Portalbenutzer hinzu oder entfernen Sie sie in AWS IoT SiteWise](#).

Wenn Sie Änderungen an dem Portal vornehmen müssen, beachten Sie die Informationen unter [Verwalten Sie Ihre SiteWise Monitor-Portale](#).

Informationen zu den ersten Schritten im Portal finden Sie unter [Erste Schritte](#) im SiteWise Monitor-Anwendungshandbuch.

AWS IoT SiteWise Dashboards erstellen (AWS CLI)

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Wenn Sie Visualisierungen (oder Widgets) in Dashboards mithilfe von definieren AWS CLI, müssen Sie die folgenden Informationen im JSON-Dokument angeben. `dashboardDefinition` Diese Definition ist ein Parameter der [CreateDashboard](#) Operationen und [UpdateDashboard](#).

`widgets`

Eine Liste von Widget-Definitionsstrukturen, die jeweils die folgenden Informationen enthalten:

`type`

Der Typ des Widgets. AWS IoT SiteWise bietet die folgenden Widget-Typen:

- `sc-line-chart`— Ein Liniendiagramm. Weitere Informationen finden Sie unter [Liniendiagramme](#) im AWS IoT SiteWise Monitor Anwendungshandbuch.
- `sc-scatter-chart`— Ein Punktdiagramm. Weitere Informationen finden Sie unter [Streudiagramme](#) im AWS IoT SiteWise Monitor Anwendungsleitfaden.
- `sc-bar-chart`— Ein Balkendiagramm. Weitere Informationen finden Sie unter [Balkendiagramme](#) im AWS IoT SiteWise Monitor Anwendungshandbuch.
- `sc-status-grid`— Ein Status-Widget, das den aktuellen Wert von Asset-Eigenschaften als Raster anzeigt. Weitere Informationen finden Sie unter [Status-Widgets](#) im AWS IoT SiteWise Monitor Anwendungsleitfaden.
- `sc-status-timeline`— Ein Status-Widget, das die historischen Werte von Asset-Eigenschaften als Zeitleiste anzeigt. Weitere Informationen finden Sie unter [Status-Widgets](#) im AWS IoT SiteWise Monitor Anwendungsleitfaden.

- `sc-kpi`— Eine Visualisierung von KPIs (Key Performance Indicator). Weitere Informationen finden Sie unter [KPI-Widgets](#) im AWS IoT SiteWise Monitor Anwendungsleitfaden.
- `sc-table`— Ein Tabellen-Widget. Weitere Informationen finden Sie unter [Tabellen-Widgets](#) im AWS IoT SiteWise Monitor Anwendungshandbuch.

`title`

Der Titel des Widgets.

`x`

Die horizontale Position des Widgets ausgehend von der linken Seite des Rasters. Dieser Wert bezieht sich auf die Position des Widgets im Raster des Dashboards.

`y`

Die vertikale Position des Widgets ausgehend vom oberen Rand des Rasters. Dieser Wert bezieht sich auf die Position des Widgets im Raster des Dashboards.

`width`

Die Breite des Widgets ausgedrückt als Anzahl der Leerzeichen im Raster des Dashboards.

`height`

Die Höhe des Widgets ausgedrückt als Anzahl der Leerzeichen im Raster des Dashboards.

`metrics`

Eine Liste von Metrikstrukturen, die jeweils einen Datenstrom für dieses Widget definieren. Jede Struktur in der Liste muss folgende Informationen enthalten:

`label`

Eine Beschriftung, die für diese Metrik angezeigt werden soll.

`type`

Der Typ der Datenquelle für diese Metrik. AWS IoT SiteWise stellt die folgenden Metriktypen bereit:

- `iotsitewise`— Das Dashboard ruft Daten für eine Anlageeigenschaft in AWS IoT SiteWise ab. Bei Auswahl dieser Option müssen Sie `assetId` und `propertyId` für diese Metrik definieren.

`assetId`

(Optional) Die ID einer Komponente in AWS IoT SiteWise.

Dieses Feld ist erforderlich, wenn Sie `iotsitewise` für `type` in dieser Metrik auswählen.

`propertyId`

(Optional) Die ID einer Komponenteneigenschaft in AWS IoT SiteWise.

Dieses Feld ist erforderlich, wenn Sie `iotsitewise` für `type` in dieser Metrik auswählen.

`analysis`

(Optional) Eine Struktur, die die Analyse definiert, z. B. Trendlinien, die für das Widget angezeigt werden sollen. Weitere Informationen finden Sie im AWS IoT SiteWise Monitor Anwendungsleitfaden unter [Konfiguration von Trendlinien](#). Sie können pro Eigenschaft im Widget eine von jedem Trendlinientyp hinzufügen. Die Analysestruktur enthält die folgenden Informationen:

`trends`

(Optional) Eine Liste von Trendstrukturen, die jeweils eine Trendanalyse für dieses Widget definieren. Jede Struktur in der Liste enthält die folgenden Informationen:

`type`

Der Typ der Trendlinie. Wählen Sie die folgende Option:

- `linear-regression`— Zeigt eine lineare Regressionslinie an. SiteWise Monitor verwendet die Methode der [kleinsten Quadrate](#), um die lineare Regression zu berechnen.

`annotations`

(Optional) Eine Annotationsstruktur, die Schwellenwerte für das Widget definiert. Weitere Informationen finden Sie unter [Konfiguration von Schwellenwerten im Anwendungshandbuch](#).AWS IoT SiteWise Monitor Sie können bis zu sechs Anmerkungen pro Widget hinzufügen. Die Struktur der Anmerkungen enthält die folgenden Informationen:

`y`

(Optional) Eine Liste von Annotationsstrukturen, die jeweils einen horizontalen Schwellenwert für dieses Widget definieren. Jede Struktur in der Liste enthält die folgenden Informationen:

`comparisonOperator`

Der Vergleichsoperator für den Schwellenwert. Wählen Sie eine der folgenden Optionen:

- LT— Hebt Eigenschaften hervor, bei denen mindestens ein Datenpunkt unter dem `liegtvalue`.
- GT— Hebt Eigenschaften hervor, bei denen mindestens ein Datenpunkt größer als der `istvalue`.
- LTE— Hebt Eigenschaften hervor, bei denen mindestens ein Datenpunkt kleiner oder gleich dem `istvalue`.
- GTE— Hebt Eigenschaften hervor, bei denen mindestens ein Datenpunkt größer oder gleich dem `istvalue`.
- EQ— Hebt Eigenschaften hervor, bei denen mindestens ein Datenpunkt gleich dem `istvalue`.

`value`

Der Schwellenwert für den Vergleich von Datenpunkten mit dem `comparisonOperator`.

`color`

(Optional) Der sechsstellige Hexadezimalcode der Schwellenfarbe. In der Visualisierung werden Eigenschaftslegenden in dieser Farbe für Eigenschaften angezeigt, bei denen mindestens ein Datenpunkt die Schwellenwertregel erfüllt. Die Standardeinstellung ist schwarz (`#000000`).

`showValue`

(Optional) Ob der Wert des Schwellenwerts an den Rändern des Widgets angezeigt werden soll oder nicht. Standardeinstellung: `true`.

`properties`

(Optional) Ein einfaches Eigenschaftswörterbuch für das Widget. Die Mitglieder dieser Struktur sind kontextabhängig. AWS IoT SiteWise stellt die folgenden Widgets bereit, die Folgendes verwenden: `properties`

- [Liniendiagramme](#), [Punktdiagramme](#) und [Balkendiagramme](#) haben die folgenden Eigenschaften:

`colorDataAcrossThresholds`

(Optional) Ob die Farbe der Daten, die die Schwellenwerte überschreiten, in diesem Widget geändert werden soll oder nicht. Wenn Sie diese Option aktivieren, werden die Daten, die einen Schwellenwert überschreiten, in der von Ihnen ausgewählten Farbe angezeigt. Standardeinstellung: `true`.

- [Statusraster](#) haben die folgenden Eigenschaften:

labels

(Optional) Eine Struktur, die die Beschriftungen definiert, die in der Statusleiste angezeigt werden sollen. Die Labelstruktur enthält die folgenden Informationen:

showValue

(Optional) Ob die Einheit und der Wert für jede Anlageneigenschaft in diesem Widget angezeigt werden sollen oder nicht. Standardeinstellung: `true`.

Example Dashboard-Beispieldefinition

Im folgenden Beispiel wird ein Dashboard aus einer Nutzlast definiert, die in einer JSON-Datei gespeichert ist.

```
aws iotsitewise create-dashboard \
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE \
  --dashboard-name "Wind Farm Dashboard" \
  --dashboard-definition file://dashboard-definition.json
```

Das folgende JSON-Beispiel für `dashboard-definition.json` definiert ein Dashboard mit den folgenden Visualisierungs-Widgets:

- Ein Liniendiagramm, das die Gesamtleistung des Windparks oben links im Dashboards visualisiert. Dieses Liniendiagramm enthält einen Schwellenwert, der angibt, wann der Windpark weniger Leistung abgibt als die erwartete Mindestleistung. Dieses Liniendiagramm enthält auch eine lineare Regressionstrendlinie.
- Ein Balkendiagramm, das die Windgeschwindigkeit für vier Turbinen oben rechts im Dashboard visualisiert.

Note

Dieses Beispiel stellt Visualisierungen von Linien- und Balkendiagrammen in einem Dashboard dar. Dieses Dashboard ist dem [Beispiel-Dashboard für einen Windpark](#) ähnlich.

```
{
```

```

"widgets": [
  {
    "type": "sc-line-chart",
    "title": "Total Average Power",
    "x": 0,
    "y": 0,
    "height": 3,
    "width": 3,
    "metrics": [
      {
        "label": "Power",
        "type": "iotsitewise",
        "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
        "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
        "analysis": {
          "trends": [
            {
              "type": "linear-regression"
            }
          ]
        }
      }
    ],
    "annotations": {
      "y": [
        {
          "comparisonOperator": "LT",
          "value": 20000,
          "color": "#D13212",
          "showValue": true
        }
      ]
    }
  },
  {
    "type": "sc-bar-chart",
    "title": "Wind Speed",
    "x": 3,
    "y": 3,
    "height": 3,
    "width": 3,
    "metrics": [
      {
        "label": "Turbine 1",

```



```
    "type": "iotsitewise",
    "assetId": "a1b2c3d4-5678-90ab-cdef-2a2a2EXAMPLE",
    "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
  },
  {
    "label": "Turbine 2",
    "type": "iotsitewise",
    "assetId": "a1b2c3d4-5678-90ab-cdef-2b2b2EXAMPLE",
    "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
  },
  {
    "label": "Turbine 3",
    "type": "iotsitewise",
    "assetId": "a1b2c3d4-5678-90ab-cdef-2c2c2EXAMPLE",
    "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
  },
  {
    "label": "Turbine 4",
    "type": "iotsitewise",
    "assetId": "a1b2c3d4-5678-90ab-cdef-2d2d2EXAMPLE",
    "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
  }
]
}
```

Schalten Sie Alarmer für Ihre Portale ein in AWS IoT SiteWise

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Note

Hinweis zum Ende des Supports: Der Support für AWS wurde beendet AWS IoT Events. Weitere Informationen finden Sie unter [AWS IoT Events Ende des Supports](#).

Sie können die von unterstützte Alarmfunktion AWS IoT Events für Ihre Portale aktivieren, sodass Portaladministratoren AWS IoT Events Alarmmodelle in Ihren SiteWise Monitor-Portalen erstellen, bearbeiten und löschen können. Projekteigentümer können Alarmer konfigurieren. Projektbetrachter können Alarmerdetails einsehen. In diesem Abschnitt wird erklärt, wie Sie die AWS IoT SiteWise Konsole verwenden können, um die Alarmfunktion für Ihre Portale zu aktivieren.

 Important

- Sie können in Ihren Portalen keine externen Alarmer erstellen.
- Wenn Sie Alarmerbenachrichtigungen senden möchten, müssen Sie IAM Identity Center für den Benutzerauthentifizierungsdienst auswählen.
- Die Funktion für Alarmerbenachrichtigungen ist in China (Peking) nicht verfügbar AWS-Region.

Wenn Sie ein Portal konfigurieren und erstellen, können Sie Alarmer und Alarmerbenachrichtigungen in Schritt 2 Zusätzliche Funktionen aktivieren. Wählen Sie je nach Benutzerauthentifizierungsdienst eine der folgenden Optionen aus:

IAM Identity Center

[AWS IoT SiteWise](#) > [Monitor](#) > [Portals](#) > Create portal

Step 1
Portal configuration

Step 2 - optional
Additional features

Step 3
Invite administrators

Step 4
Assign users

Additional features - *optional*

Alarms

Your portal users can create alarms in the portal to monitor equipment or processes. They can also get notified when the equipment or processes perform outside specified range.

☒ **Enable alarms**
If enabled, your portal users can define AWS IoT Events alarms in SiteWise Monitor.

AWS IoT SiteWise access role
Choose an IAM role that allows AWS IoT Events to send data to AWS IoT SiteWise. To edit the role, go to the [IAM console](#).

☒ Create a role from an AWS managed template

☐ Use an existing role

☒ **Enable alarm notifications**
If enabled, alarms can send email or SMS notifications.

Sender
Specify the email address that sends alarm notifications. To edit or add a sender, go to the [Amazon SES console](#).

AWS Lambda role
Choose an IAM role that allows AWS Lambda to send data to Amazon SES and Amazon SNS. To edit the role, go to the [IAM console](#).

☒ Create a role from an AWS managed template

☐ Use an existing role

AWS Lambda function
Choose an AWS Lambda function to manage alarm notifications. To edit the function, go to the [AWS Lambda console](#).

☒ Create a lambda from an AWS managed template

☐ Use an existing lambda

[Previous](#) [Create](#)

Um Alarme für ein Portal zu aktivieren

1. (Optional) Wählen Sie Alarme aktivieren.
 - Verwenden Sie für die AWS IoT SiteWise Zugriffsrolle eine vorhandene Rolle oder erstellen Sie eine Rolle mit den erforderlichen Berechtigungen. Diese Rolle erfordert die `iotevents:BatchPutMessage` Erlaubnis und eine Vertrauensbeziehung, die es ermöglicht `iot.amazonaws.com` und `iotevents.amazonaws.com` die Übernahme der Rolle ermöglicht.
2. (Optional) Wählen Sie „Alarmbenachrichtigungen aktivieren“.
 - a. Wählen Sie unter Absender den Absender aus.

⚠ Important

Sie müssen die Absender-E-Mail-Adresse in Amazon SES verifizieren. Weitere Informationen finden Sie unter [Verifizieren von E-Mail-Adressen in Amazon SES](#) im Amazon Simple Email Service Developer Guide.

- b. Verwenden Sie für die AWS Lambda Rolle eine vorhandene Rolle oder erstellen Sie eine Rolle mit den erforderlichen Berechtigungen. Für diese Rolle sind die `sso-directory:DescribeUser` Berechtigungen `lambda:InvokeFunction` und eine Vertrauensbeziehung erforderlich, die es ermöglicht `iotevents.amazonaws.com` und `lambda.amazonaws.com` die Übernahme der Rolle ermöglicht.
- c. Wählen Sie für AWS Lambda Funktionen eine vorhandene Lambda-Funktion aus oder erstellen Sie eine Funktion, die Alarmbenachrichtigungen verwaltet. Weitere Informationen finden Sie unter [Verwaltung von Alarmbenachrichtigungen](#) im AWS IoT Events Entwicklerhandbuch.

IAM

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2- optional
Additional features

Step 3
Invite administrators

Step 4
Assign users

Additional features - optional

Alarms

Your portal users can create alarms in the portal to monitor equipment or processes. They can also get notified when the equipment or processes perform outside specified range.

☒ **Enable alarms**
If enabled, your portal users can define AWS IoT Events alarms in SiteWise Monitor.

AWS IoT SiteWise access role
Choose an IAM role that allows AWS IoT Events to send data to AWS IoT SiteWise. To edit the role, go to the [IAM console](#).

☒ Create a role from an AWS managed template
☐ Use an existing role

i Alarms created in the portal can't send notifications. If you want to send alarm notifications, choose **Previous**. Then, on the **Portal configuration** page, choose **AWS SSO for User authentication**.

Previous **Create**

Um Alarmer für ein Portal zu aktivieren

- (Optional) Wählen Sie Alarmer aktivieren.

- Verwenden Sie für die AWS IoT SiteWise Zugriffsrolle eine vorhandene Rolle oder erstellen Sie eine Rolle mit den erforderlichen Berechtigungen. Diese Rolle erfordert die `iotevents:BatchPutMessage` Erlaubnis und eine Vertrauensbeziehung, die es ermöglicht `iot.amazonaws.com` und `iotevents.amazonaws.com` die Übernahme der Rolle ermöglicht.

Weitere Informationen zu Alarmen in SiteWise Monitor finden Sie unter [Überwachung mit Alarmen](#) im AWS IoT SiteWise Anwendungshandbuch.

Aktivierung Ihres AWS IoT SiteWise Portals am Edge

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Nachdem Sie Ihr Portal am Edge aktiviert haben, ist dieses Portal auf allen SiteWise Edge-Gateways verfügbar, für die das Datenverarbeitungspaket in Ihrem Konto aktiviert ist.

Um das Portal am Edge zu aktivieren

1. Aktivieren Sie im Abschnitt Edge-Konfiguration die Option Dieses Portal am Edge aktivieren.
2. Wählen Sie Erstellen aus.

Verwalten Sie Ihre SiteWise Monitor-Portale

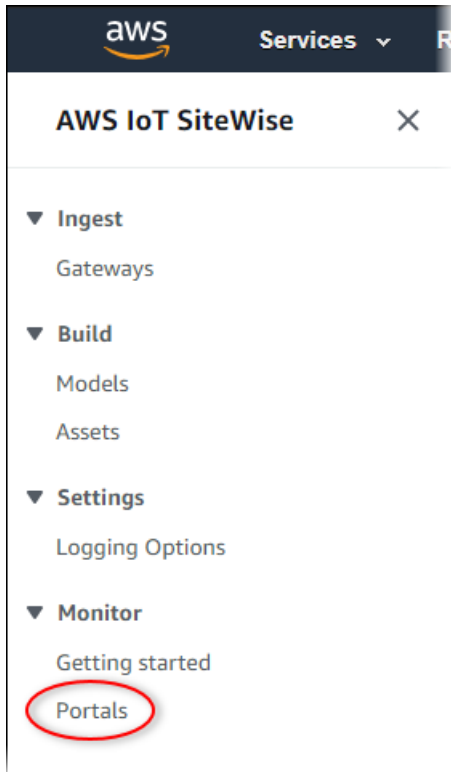
Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Sie haben die Möglichkeit, verschiedene Aspekte des Portals zu verwalten und zu konfigurieren. Dazu gehören das Hinzufügen und Entfernen von Benutzern oder Administratoren, das Festlegen

von Benutzerberechtigungen und Rollen, das Anpassen der URL und des Namens des Portals, das Einrichten von Support-Kontaktinformationen und das Senden von E-Mail-Einladungen an Portaladministratoren.

1. Melden Sie sich bei der [AWS IoT SiteWise -Konsole](#) an.
2. Wählen Sie im Navigationsbereich Monitor (Überwachen), Portals (Portale) aus.



3. Wählen Sie das Portal und dann View details (Details anzeigen) aus (oder wählen Sie den Namen des Portals aus).
4. Sie können eine der folgenden Verwaltungsaufgaben ausführen:
 - [Ändern Sie die Portaldetails in AWS IoT SiteWise](#)
 - [Fügen Sie Portaladministratoren hinzu oder entfernen Sie sie in AWS IoT SiteWise](#)
 - [Senden Sie E-Mail-Einladungen an Portaladministratoren](#)
 - [Fügen Sie Portalbenutzer hinzu oder entfernen Sie sie in AWS IoT SiteWise](#)
 - [Löschen Sie ein Portal in AWS IoT SiteWise](#)

Weitere Informationen zum Erstellen eines Portals finden Sie unter [Erste Schritte mit AWS IoT SiteWise Monitor \(Classic\)](#).

Themen

- [Ändern Sie die Portaldetails in AWS IoT SiteWise](#)
- [Fügen Sie Portaladministratoren hinzu oder entfernen Sie sie in AWS IoT SiteWise](#)
- [Senden Sie E-Mail-Einladungen an Portaladministratoren](#)
- [Fügen Sie Portalbenutzer hinzu oder entfernen Sie sie in AWS IoT SiteWise](#)
- [Löschen Sie ein Portal in AWS IoT SiteWise](#)

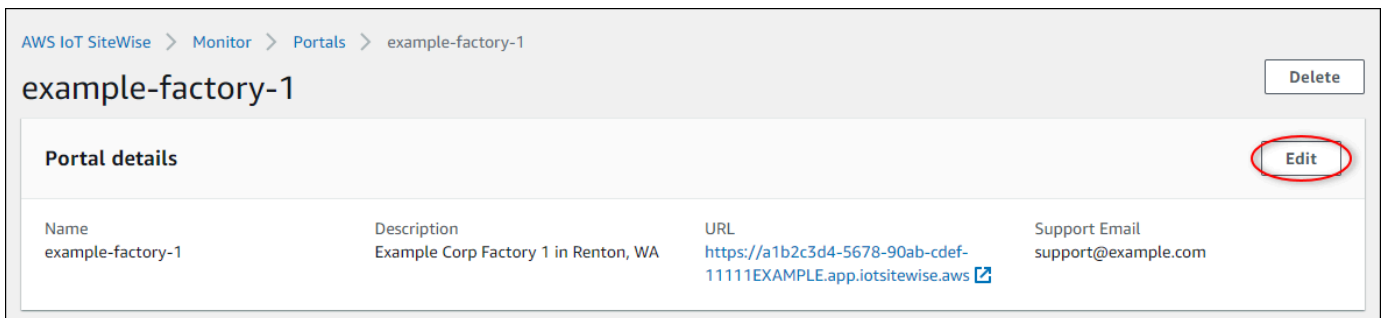
Ändern Sie die Portaldetails in AWS IoT SiteWise

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Sie können den Namen, die Beschreibung, das Branding, die Support-E-Mail-Adresse und die Berechtigungen eines Portals ändern.

1. Wählen Sie auf der Seite mit den Portaldetails im Abschnitt Portal details (Portaldetails) die Option Edit (Bearbeiten) aus.



Name	Description	URL	Support Email
example-factory-1	Example Corp Factory 1 in Renton, WA	https://a1b2c3d4-5678-90ab-cdef-11111EXAMPLE.app.iotsitewise.aws	support@example.com

2. Aktualisieren Sie Name, Description (Beschreibung), Portal Branding, Support contact email (E-Mail-Adresse des Support-Kontakts) oder Permissions (Berechtigungen).
3. Wenn Sie fertig sind, wählen Sie Speichern.

Fügen Sie Portaladministratoren hinzu oder entfernen Sie sie in AWS IoT SiteWise

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Sie können mit wenigen Schritten Benutzer als Administratoren für ein Portal hinzufügen oder entfernen. Wählen Sie je nach Benutzerauthentifizierungsdienst eine der folgenden Optionen aus.

IAM Identity Center

Portal administrators (1)						Remove from portal	Send invitations	Assign administrators
☐	Display name	Type	Email address	Role				
☐	Jane Doe	SSO user	janedoe@example.com	Portal administrator				

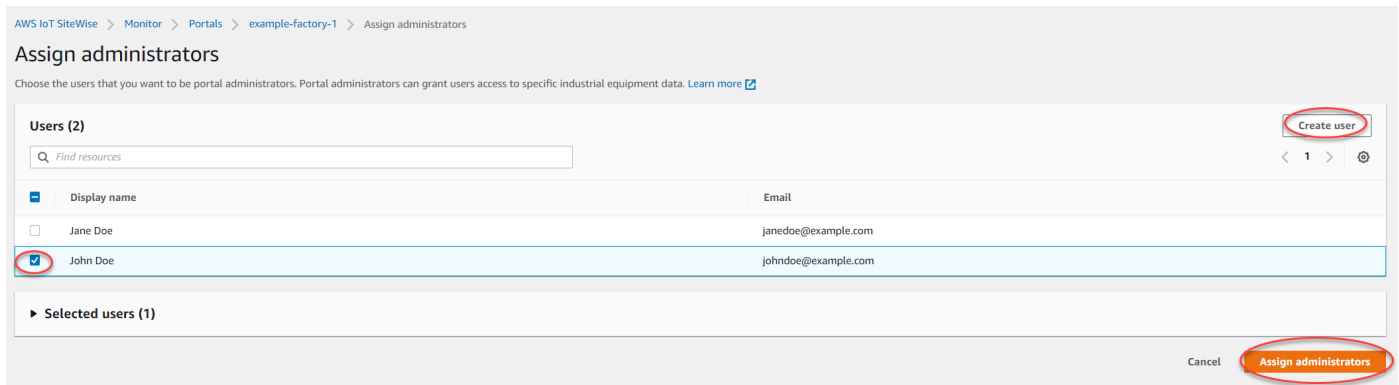
So fügen Sie Portaladministratoren hinzu

1. Wählen Sie auf der Seite mit den Portaldetails im Abschnitt Portaladministratoren die Option Administratoren zuweisen aus.
2. Aktivieren Sie auf der Seite Administratoren zuweisen die Kontrollkästchen für die Benutzer, die dem Portal als Administratoren hinzugefügt werden sollen.

Note

Wenn Sie IAM Identity Center als Identitätsspeicher verwenden und bei Ihrem AWS Organizations Verwaltungskonto angemeldet sind, können Sie Benutzer erstellen wählen, um einen IAM Identity Center-Benutzer zu erstellen. IAM Identity Center sendet dem neuen Benutzer eine E-Mail, damit er sein Passwort einrichten kann. Anschließend können Sie den Benutzer dem Portal als Administrator zuweisen. Weitere Informationen finden Sie unter [Identitäten im IAM Identity Center verwalten](#).

3. Wählen Sie Administratoren zuweisen.



So entfernen Sie Portaladministratoren

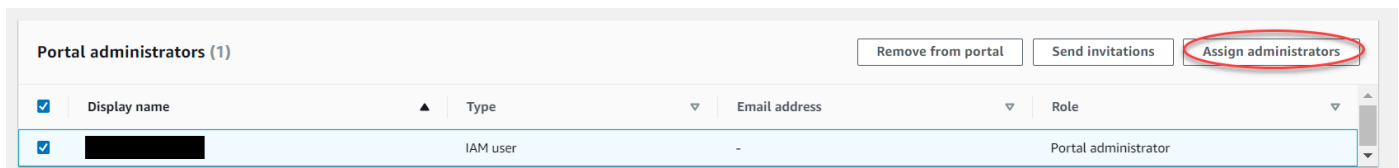
- Aktivieren Sie auf der Seite „Portal details (Portaldetails)“ im Abschnitt Portal administrators (Portaladministratoren) das Kontrollkästchen für jeden zu entfernenden Benutzer und wählen Sie dann Remove from portal (Aus Portal entfernen) aus.



Note

Wir empfehlen, dass Sie mindestens einen Portaladministrator auswählen.

IAM



So fügen Sie Portaladministratoren hinzu

- Wählen Sie auf der Seite mit den Portaldetails im Abschnitt Portaladministratoren die Option Administratoren zuweisen aus.
- Gehen Sie auf der Seite Administratoren zuweisen wie folgt vor:
 - Wählen Sie IAM-Benutzer, wenn Sie einen IAM-Benutzer als Portaladministrator hinzufügen möchten.
 - Wählen Sie IAM-Rollen, wenn Sie eine IAM-Rolle als Portaladministrator hinzufügen möchten.

3. Aktivieren Sie die Kontrollkästchen für die Benutzer oder Rollen, die Sie als Portaladministratoren verwenden möchten. Dadurch werden die Benutzer oder Rollen zur Liste der Portaladministratoren hinzugefügt.
4. Wählen Sie Administratoren zuweisen aus.

Important

Benutzer oder Rollen müssen über die `iotsitewise:DescribePortal` Berechtigung verfügen, sich beim Portal anzumelden.

AWS IoT SiteWise > Monitor > Portals > example-factory-2 > Assign administrators

Assign administrators

Choose the users that you want to be portal administrators. Portal administrators can grant users access to specific industrial equipment data. [Learn more](#)

 IAM users or roles must have the `iotsitewise:DescribePortal` permission to sign in to the portal.

Users Roles

IAM users (1) [Manage users in IAM console](#)

☒	Name	Date created
☒	raspberrypi-testing	11-08-2019

► Portal administrators (1) Remove

Cancel Assign administrators

[AWS IoT SiteWise](#) > [Monitor](#) > [Portals](#) > [example-factory-2](#) > [Assign administrators](#)

Assign administrators

Choose the users that you want to be portal administrators. Portal administrators can grant users access to specific industrial equipment data. [Learn more](#)

ⓘ IAM users or roles must have the `lotsitewise:DescribePortal` permission to sign in to the portal.

Users **Roles**

IAM roles (66) [Manage roles in IAM console](#)

Find role name

	Name	Date created
☐	[REDACTED]	
☒	AWSIoTSiteWiseMonitorServiceRole_4wZigNpA1	03-16-2021
☐	AWSIoTSiteWiseMonitorServiceRole_ECKT-2Oar	03-11-2021
☐	AWSIoTSiteWiseMonitorServiceRole_GTnd0O4Wr	03-16-2021
☐	AWSIoTSiteWiseMonitorServiceRole_rHINLNCs-	03-11-2021
☐	AWSIoTSiteWiseMonitorServiceRole_XB330QUIO	03-10-2021
☐	[REDACTED]	
☐	[REDACTED]	

► Portal administrators (2) Remove

Cancel Assign administrators

So entfernen Sie Portaladministratoren

- Aktivieren Sie auf der Seite „Portal details (Portaldetails)“ im Abschnitt Portal administrators (Portaladministratoren) das Kontrollkästchen für jeden zu entfernenden Benutzer und wählen Sie dann Remove from portal (Aus Portal entfernen) aus.

Note

Es empfiehlt sich nicht, ein Portal ohne Portaladministrator zu belassen.

Senden Sie E-Mail-Einladungen an Portaladministratoren

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Sie können E-Mail-Einladungen an Portaladministratoren senden.

1. Aktivieren Sie auf der Seite mit den Portaldetails im Abschnitt Portal administrators (Portaladministratoren) die Kontrollkästchen für die Portaladministratoren.

☒	Display name	Email address	Role
☒	John Doe	john.doe@example.com	Portal administrator

2. Wählen Sie Send invitations (Einladungen senden) aus. Ihr E-Mail-Client wird geöffnet und der Nachrichtentext wird mit einer Einladung gefüllt.

Sie können die E-Mail anpassen, bevor Sie sie an die Portaladministratoren senden.

Fügen Sie Portalbenutzer hinzu oder entfernen Sie sie in AWS IoT SiteWise

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Sie wählen, welche Benutzer Zugriff auf Ihr Portal haben. Portalbenutzer werden in der Benutzerliste innerhalb eines SiteWise Monitor-Portals angezeigt. Aus dieser Liste können Portaladministratoren Projektbesitzer hinzufügen, und Projektbesitzer können Projektbetrachter hinzufügen.

Note

Ihre Portaladministratoren und Portalbenutzer wenden sich möglicherweise über die Support-E-Mail eines Portals an Sie, wenn Sie einen Benutzer hinzufügen oder entfernen müssen.

Wählen Sie je nach Benutzerauthentifizierungsdienst eine der folgenden Optionen aus.

IAM Identity Center

☐	Display name	Type	Email address	Role
☐	John Doe	SSO user	johndoe@example.com	Portal viewer

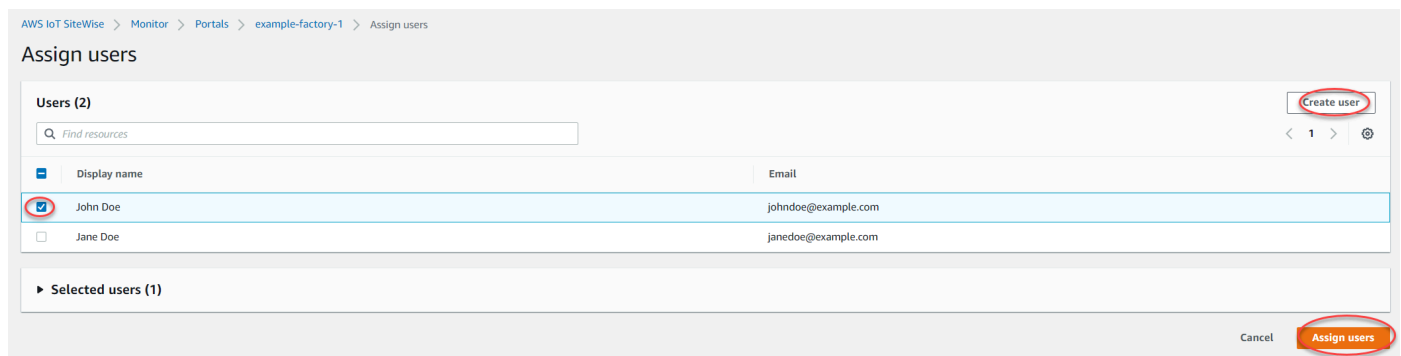
So fügen Sie Portalbenutzer hinzu

1. Wählen Sie auf der Seite „Portal details (Portaldetails)“ im Abschnitt Portal users (Portalbenutzer) die Option Assign users (Benutzer zuweisen) aus.
2. Aktivieren Sie auf der Seite Benutzer zuweisen das Kontrollkästchen für die Benutzer, die dem Portal hinzugefügt werden sollen.

Note

Wenn Sie IAM Identity Center als Identitätsspeicher verwenden und bei Ihrem AWS Organizations Verwaltungskonto angemeldet sind, können Sie Benutzer erstellen auswählen, um einen IAM Identity Center-Benutzer zu erstellen. IAM Identity Center sendet dem neuen Benutzer eine E-Mail, damit er sein Passwort einrichten kann. Anschließend können Sie den Benutzer dem Portal als Benutzer zuweisen. Weitere Informationen finden Sie unter [Identitäten im IAM Identity Center verwalten](#).

3. Wählen Sie Assign users (Benutzer zuweisen) aus.



So entfernen Sie Portalbenutzer

- Aktivieren Sie auf der Seite mit den Portaldetails im Abschnitt Portalbenutzer das Kontrollkästchen für die Benutzer, die aus dem Portal entfernt werden sollen, und wählen Sie dann Aus Portal entfernen aus.

IAM

Portal users (1)					Remove from portal	Assign users
☐	Display name	Type	Email address	Role		
☐	AWSIoTSiteWiseMonitorServiceRole_4wZigNpA1	IAM role	-	Portal viewer		

So fügen Sie Portalbenutzer hinzu

1. Wählen Sie auf der Seite „Portal details (Portaldetails)“ im Abschnitt Portal users (Portalbenutzer) die Option Assign users (Benutzer zuweisen) aus.
2. Gehen Sie auf der Seite Benutzer zuweisen wie folgt vor:
 - Wählen Sie IAM-Benutzer aus, um einen IAM-Benutzer als Ihren Portalbenutzer hinzuzufügen.
 - Wählen Sie IAM-Rollen aus, um eine IAM-Rolle als Portalbenutzer hinzuzufügen.
3. Aktivieren Sie die Kontrollkästchen für die Benutzer oder Rollen, die Sie als Portalbenutzer hinzufügen möchten. Dadurch werden die Benutzer oder Rollen zur Liste der Portalbenutzer hinzugefügt.
4. Wählen Sie Assign users (Benutzer zuweisen) aus.

AWS IoT SiteWise > Monitor > Portals > example-factory-2 > Assign users

Assign users

Users Roles

IAM users (1) [Manage users in IAM console](#)

☒	Name	Date created
☒	[REDACTED]	11-08-2019

Portal users (1) [Remove](#)

Cancel **Assign users**

AWS IoT SiteWise > Monitor > Portals > example-factory-2 > Assign users

Assign users

Users Roles

IAM roles (66)

Find role name

Manage roles in IAM console

	Name	Date created
☐	[REDACTED]	
☒	AWSIoTSiteWiseMonitorServiceRole_4wZigNpA1	03-16-2021
☐	AWSIoTSiteWiseMonitorServiceRole_ECKT-2Oar	03-11-2021
☐	AWSIoTSiteWiseMonitorServiceRole_GTnd0O4Wr	03-16-2021
☐	AWSIoTSiteWiseMonitorServiceRole_rHINLNCs-	03-11-2021
☐	AWSIoTSiteWiseMonitorServiceRole_XB330QUIO	03-10-2021
☐	[REDACTED]	
☐	[REDACTED]	
☐	[REDACTED]	

Portal users (2)

Remove

Cancel Assign users

So entfernen Sie Portalbenutzer

- Aktivieren Sie auf der Seite mit den Portaldetails im Abschnitt Portalbenutzer das Kontrollkästchen für die Benutzer, die aus dem Portal entfernt werden sollen, und wählen Sie dann Aus Portal entfernen aus.

Important

Benutzer oder Rollen müssen über die `iotsitewise:DescribePortal` Berechtigung verfügen, sich beim Portal anzumelden.

Löschen Sie ein Portal in AWS IoT SiteWise

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Sie können ein Portal löschen, wenn Sie es zu Testzwecken erstellt haben, oder wenn Sie ein Duplikat eines bereits erstellten Portals erstellt haben.

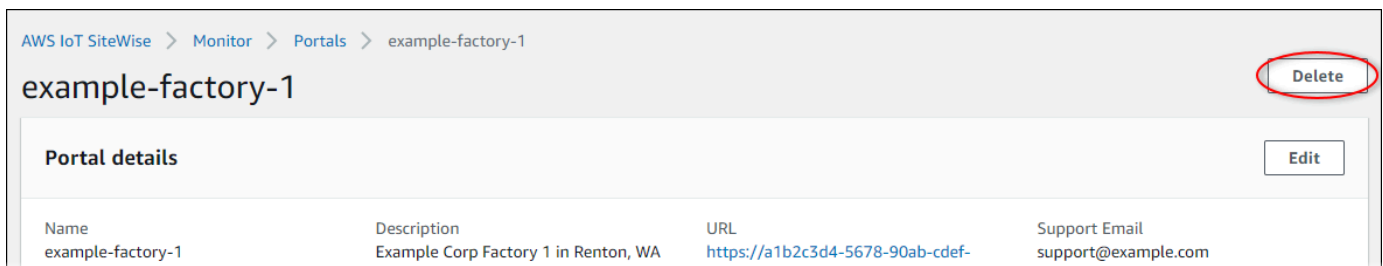
Note

Bevor Sie das Portal löschen können, müssen Sie zunächst alle Dashboards und Projekte in dem Portal manuell löschen. Weitere Informationen finden Sie unter [Löschen von Projekten](#) und [Löschen von Dashboards](#) im SiteWise Monitor-Anwendungshandbuch.

1. Wählen Sie auf der Seite „Portal details (Portaldetails)“ die Option Delete (Löschen) aus.

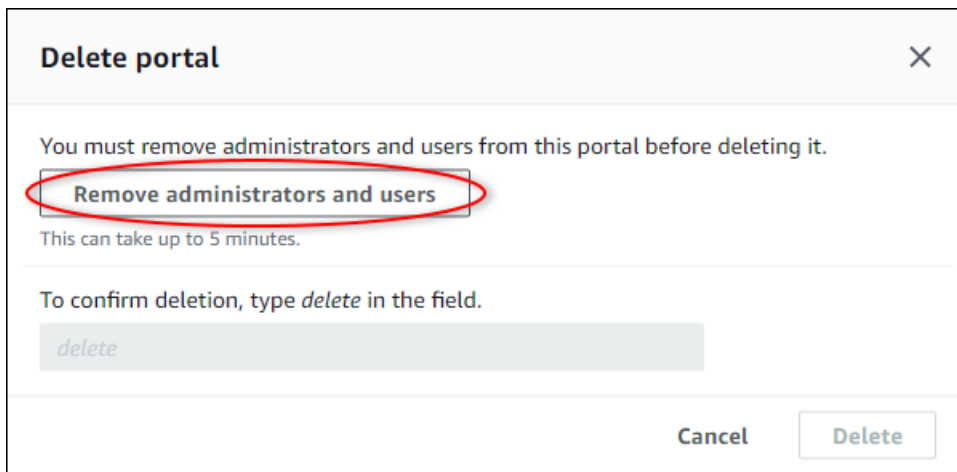
Important

Wenn Sie ein Portal löschen, gehen alle Projekte, die das Portal enthält, und alle Dashboards in den einzelnen Projekten verloren. Diese Aktion kann nicht rückgängig gemacht werden. Ihre Komponentendaten sind nicht betroffen.



2. Wählen Sie im Dialogfeld Portale löschen die Option Admins and users entfernen aus.

Sie müssen die Administratoren und Benutzer aus dem Portal entfernen, bevor Sie das Portal löschen können. Wenn es für Ihr Portal keine Administratoren oder Benutzer gibt, wird die Schaltfläche nicht angezeigt und Sie können mit dem nächsten Schritt fortfahren.



Delete portal ✕

You must remove administrators and users from this portal before deleting it.

Remove administrators and users

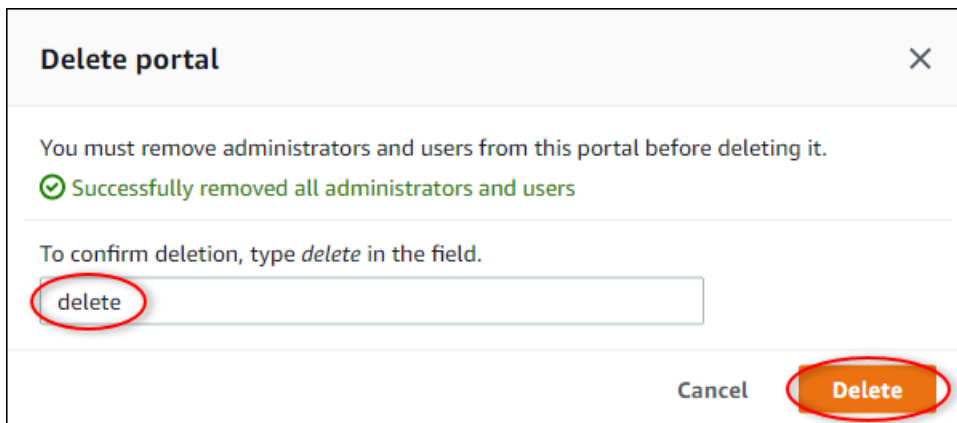
This can take up to 5 minutes.

To confirm deletion, type *delete* in the field.

delete

Cancel Delete

3. Wenn Sie sicher sind, dass Sie das gesamte Portal löschen möchten, geben Sie **delete** in das Feld ein, um das Löschen zu bestätigen.



Delete portal ✕

You must remove administrators and users from this portal before deleting it.

✓ Successfully removed all administrators and users

To confirm deletion, type *delete* in the field.

delete

Cancel Delete

4. Wählen Sie Löschen aus.

Erste Schritte mit AWS IoT SiteWise Monitor (AI-aware) — Vorschau

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Als AWS Administrator Ihrer Organisation können Sie von der AWS IoT SiteWise Konsole aus Portale erstellen, sodass Mitglieder Ihrer Organisation Ihre AWS IoT SiteWise Daten einsehen können. Führen Sie zunächst die folgenden Schritte aus.

1. Konfigurieren und erstellen Sie ein Portal.
2. Fügen Sie Portaladministratoren hinzu und senden Sie Einladungs-E-Mails.
3. Portalbenutzer hinzufügen.

Nachdem Sie ein Portal erstellt haben, kann der Portaladministrator Projekte erstellen und Benutzer zum Projekt hinzufügen. Die Projektmitglieder erstellen dann Dashboards, um die verbundenen Daten zu visualisieren AWS IoT SiteWise, sodass sie die Leistung ihrer angeschlossenen Geräte, Prozesse und Geräte überwachen können.

 Note

Vermeiden Sie beim Hinzufügen von Benutzern oder Administratoren zum Portal die Erstellung von AWS Identity and Access Management (IAM-) Richtlinien, die Benutzerberechtigungen einschränken, wie z. B. eingeschränkte IP-Adressen. Alle angehängten Richtlinien mit eingeschränkten Berechtigungen können keine Verbindung zum AWS IoT SiteWise Portal herstellen.

Erstellen Sie Projekte, um sie mit Ihren Teams zu teilen. Projekteigentümer können dann Dashboards erstellen, um die Eigenschaften der Ressourcen zu visualisieren, sodass die Projektbetrachter besser verstehen, wie Geräte, Prozesse und Geräte funktionieren. Es bietet Ihren Teams auch einen konsistenten Überblick über die Abläufe.

Dashboards helfen Ihnen, Ihre Projektdaten zu visualisieren und zu verstehen. Es hilft Unternehmen und Anwendungsbenutzern, den Überblick über ihre AWS IoT Geräte und Daten zu behalten. Wählen Sie einen Visualisierungstyp, der Ihre Daten für Ihre Bedürfnisse am besten darstellt. Ordnen Sie Visualisierungen neu an und ändern Sie die Größe, um ein Layout zu erstellen, das zu Ihrem Team passt. Erkunden Sie Ihre Geräte-, Prozess- und Ausrüstungsressourcen und -daten, identifizieren Sie Probleme schnell und verbessern Sie die betriebliche Effizienz.

Themen

- [Erstellen Sie ein Portal](#)
- [Konfigurieren Sie Ihr Portal](#)

- [Verwalte deine Portale](#)
- [Löschen Sie ein Portal](#)
- [Erstellen Sie Dashboards mit AWS CLI](#)
- [Anmeldung am Portal](#)
- [Erstellen eines Projekts](#)
- [Projekt aktualisieren](#)
- [Projekt löschen](#)
- [Erstellen eines Dashboards](#)
- [Aktualisieren eines Dashboards](#)
- [Löschen eines Dashboards](#)
- [Dashboard konfigurieren](#)

Erstellen Sie ein Portal

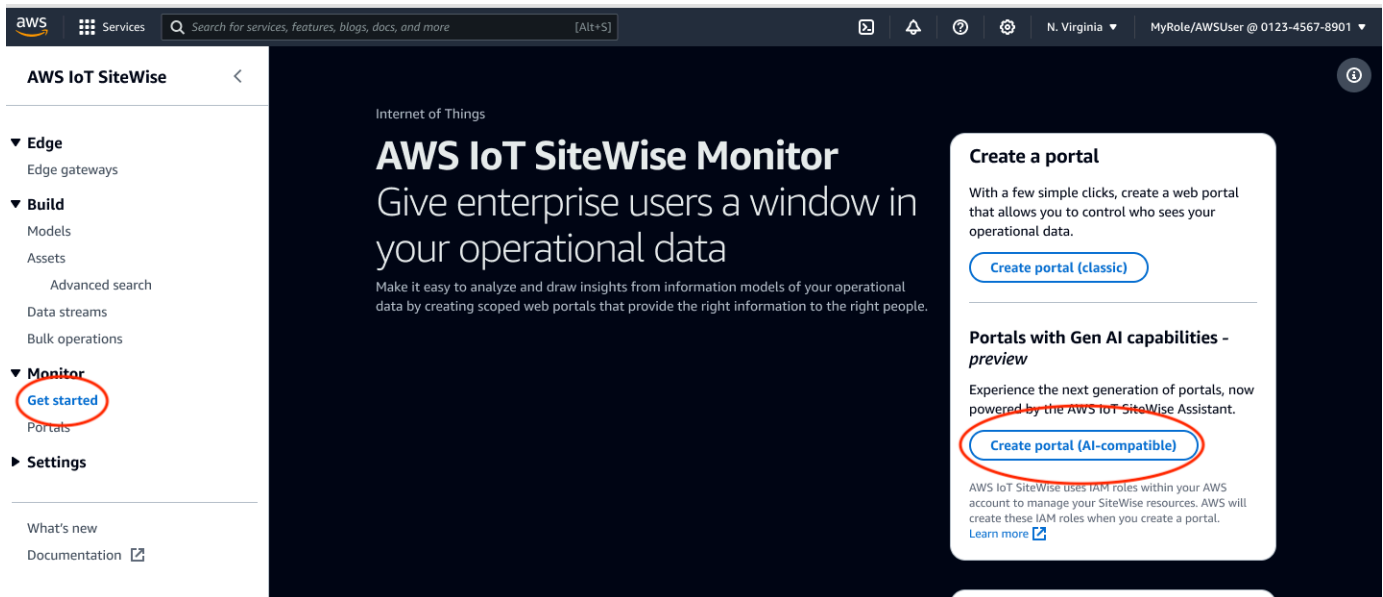
Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Sie erstellen ein SiteWise Monitor-Portal in der AWS IoT SiteWise Konsole.

So erstellen Sie ein Portal

1. Melden Sie sich bei der [AWS IoT SiteWise -Konsole](#) an.
2. Wählen Sie im Navigationsbereich Monitor, Get started aus.
3. Wählen Sie Portal erstellen (KI-fähig) aus.



Als Nächstes müssen Sie einige grundlegende Informationen zur Konfiguration des Portals angeben.

Konfigurieren Sie Ihr Portal

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Ihre Benutzer verwenden Portale, um Ihre Daten anzuzeigen. Sie können den Namen, die Beschreibung, das Branding, die Benutzerauthentifizierung, die E-Mail-Adresse des Support-Kontakts und die Berechtigungen eines Portals anpassen.

Schritte zur Konfiguration eines Portals:

1. Geben Sie einen Namen für Ihr Portal ein.
2. (Optional) Geben Sie eine Beschreibung für Ihr Portal ein. Wenn Sie über mehrere Portale verfügen, verwenden Sie aussagekräftige Beschreibungen, um den Überblick über die Inhalte der einzelnen Portale zu behalten.

3. (Optional) Laden Sie ein Bild hoch, um Ihre Marke im Portal anzuzeigen. Wählen Sie ein quadratisches PNG-Bild aus. Wenn Sie ein nicht quadratisches Bild hochladen, skaliert das Portal das Bild zu einem Quadrat.
4. Geben Sie bei Support-Problemen eine E-Mail-Adresse in das Feld Support-Kontakt-E-Mail ein.
5. Wählen Sie im Feld Benutzerauthentifizierung die folgende Option aus:
 - Wählen Sie IAM Identity Center, wenn sich Ihre Portalbenutzer mit ihren Firmenbenutzernamen und Passwörtern bei diesem Portal anmelden.

Wenn Sie IAM Identity Center in Ihrem Konto nicht aktiviert haben, gehen Sie wie folgt vor:

- a. Wählen Sie Create user (Benutzer erstellen) aus.
- b. Um das erste Portal zu erstellen, geben Sie auf der Seite Benutzer erstellen die E-Mail-Adresse, den Vor- und Nachnamen des Benutzers ein und wählen Sie dann Benutzer erstellen aus.

 Note

Support für IAM-Anmeldeinformationen ist in Kürze verfügbar.

6. Wählen Sie im Bereich Servicezugriff eine der folgenden Optionen aus:
 - Wählen Sie Neue Servicerolle erstellen und verwenden aus. Standardmäßig erstellt SiteWise Monitor automatisch eine Servicerolle für jedes Portal. Diese Rolle ermöglicht Ihren Portalbenutzern den Zugriff auf Ihre AWS IoT SiteWise Ressourcen. Weitere Informationen finden Sie unter [Verwenden Sie Servicerollen fürAWS IoT SiteWise Monitor](#).
 - Wählen Sie Bestehende Servicerolle verwenden und wählen Sie dann die Zielrolle aus.
7. Wählen Sie, ob Sie den AWS IoT SiteWise Assistenten für dieses Portal aktivieren möchten. Der AWS IoT SiteWise Assistent bietet schnelle Datenanalysen, Einblicke in Echtzeit und gezielte Empfehlungen.

 Note

Die Aktivierung des AWS IoT SiteWise Assistenten ist kostenpflichtig. Um Wissenslösungen und Anleitungen auf Unternehmensebene nutzen zu können, benötigen Sie einen Datensatz, der mit dem Amazon Kendra Kendra-Index verknüpft ist.

8. (Optional) Fügen Sie Tags für Ihr Portal hinzu. Weitere Informationen finden Sie unter [Kennzeichnen Sie Ihre AWS IoT SiteWise Ressourcen](#).
9. Wählen Sie Portal erstellen. AWS IoT SiteWise erstellt Ihr Portal.

Note

Wenn Sie die Konsole schließen, können Sie zum Abschluss der Einrichtung Administratoren und Benutzer hinzufügen. Weitere Informationen finden Sie unter [Portaladministratoren hinzufügen oder entfernen](#). Wenn Sie dieses Portal nicht behalten möchten, löschen Sie es, damit es keine Ressourcen verbraucht. Weitere Informationen finden Sie unter [Löschen Sie ein Portal](#).

Wenn Ihr Portal erstellt ist, wird eine Meldung angezeigt.

 Successfully created portal "example portal". 

[AWS IoT SiteWise](#) > [Monitor](#) > [Portals](#) > example portal

example portal Edit Delete Open portal 

Portal details		
Name example portal	Status  Active	URL  https://p-jfnlf2d8.gamma.iotsitewise.aws
Type AI-compatible	AWS IoT SiteWise Assistant  Disabled	Portal branding -
Description -	Last updated November 1, 2024, 15:37 (UTC-07:00)	Support contact email myemail@mycompany.com
ID f5fc93a1-011c-4c5a-81a6-e001b50d2547	Date created November 1, 2024, 15:37 (UTC-07:00)	

Sobald ein Portal erstellt wurde, wird es im Abschnitt Portale aufgeführt. Im Abschnitt Portaldetails werden Name, Beschreibung, ID, URL, Status, Datum der letzten Aktualisierung und Erstellung, das Portal-Branding und die Support-E-Mail für jedes Portal aufgeführt.

Die Spalte Status kann einen der folgenden Werte haben.

- **AWS IoT SiteWise CREATING** - bearbeitet Ihre Anfrage zur Erstellung des Portals. Dieser Vorgang kann mehrere Minuten in Anspruch nehmen.
- **UPDATE** - AWS IoT SiteWise bearbeitet Ihre Anfrage zur Aktualisierung des Portals. Dieser Vorgang kann mehrere Minuten in Anspruch nehmen.

- **PENDING** - AWS IoT SiteWise wartet darauf, dass die Weitergabe des DNS-Eintrags abgeschlossen ist. Dieser Vorgang kann mehrere Minuten in Anspruch nehmen. Sie können das Portal löschen, solange der Status **AUSSTEHEND** ist.
- **LÖSCHEN** - AWS IoT SiteWise bearbeitet Ihre Anfrage zum Löschen des Portals. Dieser Vorgang kann mehrere Minuten in Anspruch nehmen.
- **AKTIV** - Wenn das Portal aktiv wird, können Ihre Portalbenutzer darauf zugreifen.
- **FEHLGESCHLAGEN** — Ihre Anfrage zur Erstellung, Aktualisierung oder Löschung des Portals AWS IoT SiteWise konnte nicht bearbeitet werden. Wenn Sie AWS IoT SiteWise das Senden von Protokollen an Amazon CloudWatch Logs aktiviert haben, können Sie diese Protokolle zur Behebung von Problemen verwenden. Weitere Informationen finden Sie unter [Überwachung AWS IoT SiteWise mit CloudWatch Protokollen](#).

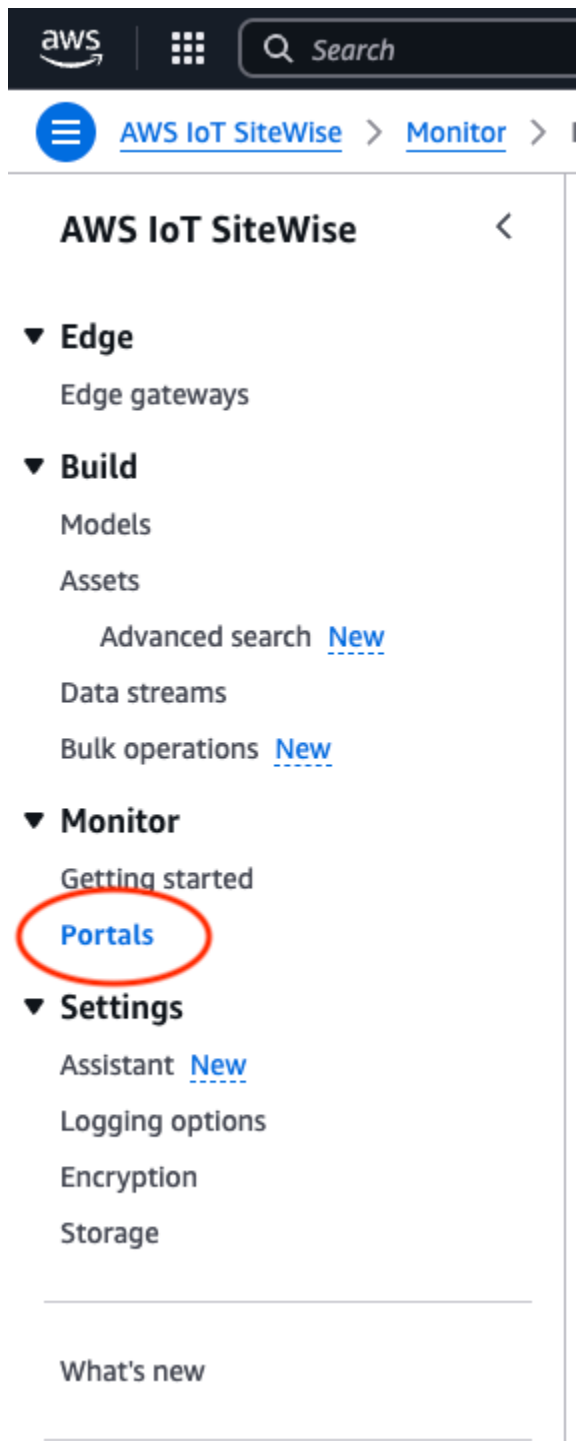
Verwalte deine Portale

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Sie haben die Möglichkeit, verschiedene Aspekte des Portals zu verwalten und zu konfigurieren. Dazu gehören das Hinzufügen und Entfernen von Administratoren, das Festlegen von Berechtigungen und Rollen, das Anpassen des Namens und der Beschreibung, das Einrichten von Support-E-Mails und das Einladen von Portaladministratoren.

1. Melden Sie sich bei der [AWS IoT SiteWise -Konsole](#) an.
2. Wählen Sie im Navigationsbereich Monitor (Überwachen), Portals (Portale) aus.



3. Wählen Sie ein Portal und dann Portal öffnen (oder wählen Sie den Namen des Portals).
4. Sie können eine der folgenden Verwaltungsaufgaben ausführen:
 - [Bearbeiten Sie die Portalattribute](#)
 - [Portaladministratoren hinzufügen oder entfernen](#)

- [Senden Sie E-Mail-Einladungen an Portaladministratoren](#)
- [Löschen Sie ein Portal in AWS IoT SiteWise](#)

Bearbeiten Sie die Portalattribute

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Sie können den Namen, die Beschreibung, das Branding, die Support-E-Mail und den Servicezugriff eines Portals ändern.

1. Wählen Sie auf der Seite mit den Portaldetails im Abschnitt Portal details (Portal details) die Option Edit (Bearbeiten) aus.



[AWS IoT SiteWise](#) > [Monitor](#) > [Portals](#) > example portal

example portal

EditDeleteOpen portal

Portal details

Name example portal	Status Active	URL https://p-jfnlf2d8.gamma.iotsitewise.aws
Type AI-compatible	AWS IoT SiteWise Assistant Disabled	Portal branding -
Description -	Last updated November 1, 2024, 15:37 (UTC-07:00)	Support contact email myemail@mycompany.com
ID f5fc93a1-011c-4c5a-81a6-e001b50d2547	Date created November 1, 2024, 15:37 (UTC-07:00)	

2. Aktualisieren Sie den Namen, die Beschreibung, das Portal-Branding, die Support-Kontakt-E-Mail, den AWS IoT SiteWise Assistenten - oder Servicezugriff.
3. Wenn Sie fertig sind, wählen Sie Änderungen speichern aus.

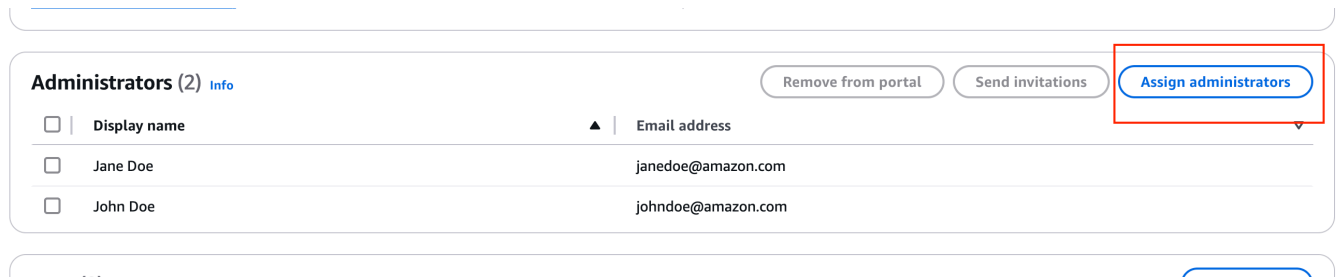
Portaladministratoren hinzufügen oder entfernen

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Sie können mit wenigen Schritten Benutzer als Administratoren für ein Portal hinzufügen oder entfernen. Wählen Sie je nach Benutzerauthentifizierungsdienst eine der folgenden Optionen aus.

IAM Identity Center



So fügen Sie Portaladministratoren hinzu

1. Wählen Sie auf der Seite mit den Portaldetails im Abschnitt Administratoren die Option Administratoren zuweisen aus.
2. Wählen Sie auf der Seite Administratoren zuweisen die Benutzer aus, die dem Portal als Administratoren hinzugefügt werden sollen.

Note

Wenn Sie IAM Identity Center als Identitätsspeicher verwenden und bei Ihrem AWS Organizations Verwaltungskonto angemeldet sind, können Sie Benutzer erstellen wählen, um einen IAM Identity Center-Benutzer zu erstellen. IAM Identity Center sendet dem neuen Benutzer eine E-Mail, damit er sein Passwort einrichten kann. Anschließend können Sie den Benutzer dem Portal als Administrator zuweisen. Weitere Informationen finden Sie unter [Identitäten im IAM Identity Center verwalten](#).

3. Wählen Sie Administratoren zuweisen.

Assign administrators [Info](#)

Choose portal administrators from the users list. Portal administrators grant access to specific industrial equipment data. [Learn more](#)

Users (11) Create user

X 2 matches < 1 > ⚙

☐	Name	Email
☒	johndoe@amazon.com	johndoe@amazon.com
☐	janedoe@amazon.com	janedoe@amazon.com

Cancel Assign administrators

So entfernen Sie Portaladministratoren

- Aktivieren Sie auf der Seite „Portal details (Portaldetails)“ im Abschnitt Portal administrators (Portaladministratoren) das Kontrollkästchen für jeden zu entfernenden Benutzer und wählen Sie dann Remove from portal (Aus Portal entfernen) aus.

Note

Unter Administratoren (#) ist die Anzahl der Administratoren für das Portal aufgeführt. Sie können mehrere Portaladministratoren hinzufügen, um Projekte zu verwalten und daran zu arbeiten.

Senden Sie E-Mail-Einladungen an Portaladministratoren

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Sie können E-Mail-Einladungen an Portaladministratoren senden.

- Aktivieren Sie auf der Seite mit den Portaldetails im Abschnitt Administratoren die Kontrollkästchen für die Portaladministratoren.
- Wählen Sie Send invitations (Einladungen senden) aus. Ihr E-Mail-Client wird geöffnet und der Nachrichtentext wird mit einer Einladung gefüllt.

Sie können die E-Mail anpassen, bevor Sie sie an die Portaladministratoren senden.

Löschen Sie ein Portal

Sie können ein Portal löschen, wenn Sie es zu Testzwecken erstellt haben, oder wenn Sie ein Duplikat eines bereits erstellten Portals erstellt haben.

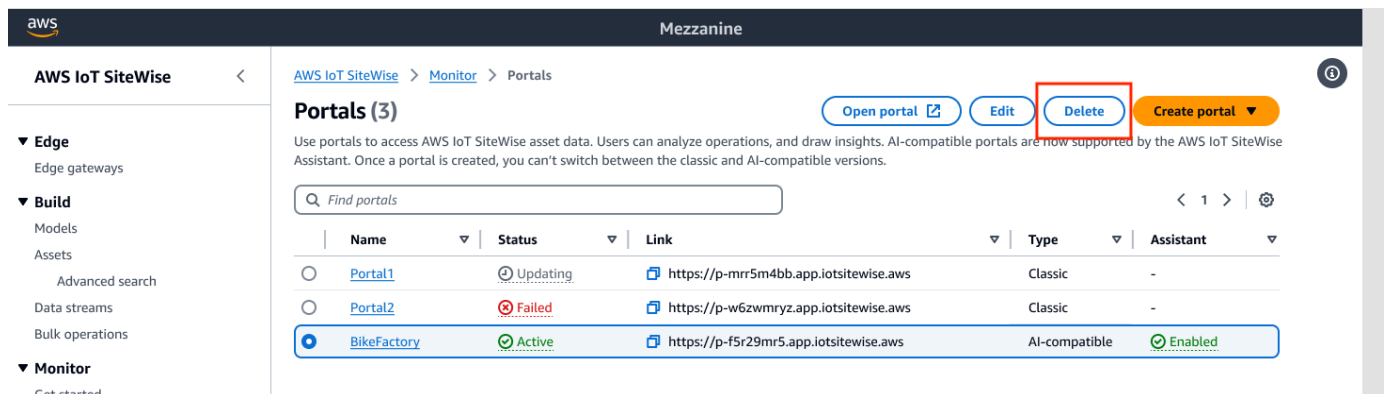
Note

Bevor Sie das Portal löschen können, müssen Sie zunächst alle Dashboards und Projekte in dem Portal manuell löschen.

1. Wählen Sie auf der Seite „Portal details (Portaldetails)“ die Option Delete (Löschen) aus.

Important

Wenn Sie ein Portal löschen, gehen alle Projekte, die das Portal enthält, und alle Dashboards in den einzelnen Projekten verloren. Diese Aktion kann nicht rückgängig gemacht werden. Ihre Komponentendaten sind nicht betroffen.



2. Wählen Sie im Dialogfeld „Portal löschen“ die Option „Administratoren und Benutzer entfernen“ aus.

Sie müssen die Administratoren und Benutzer aus dem Portal entfernen, bevor Sie das Portal löschen können. Wenn es für Ihr Portal keine Administratoren oder Benutzer gibt, wird die Schaltfläche nicht angezeigt und Sie können mit dem nächsten Schritt fortfahren.

3. Wenn Sie sicher sind, dass Sie das gesamte Portal löschen möchten, geben Sie **confirm** in das Feld ein, um das Löschen zu bestätigen.
4. Wählen Sie Löschen aus.

Erstellen Sie Dashboards mit AWS CLI

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Wenn Sie Visualisierungen (oder Widgets) in Dashboards mithilfe von definieren AWS CLI, müssen Sie die folgenden Informationen im JSON-Dokument angeben. `dashboardDefinition` Diese Definition ist ein Parameter der [CreateDashboard](#) Operationen und [UpdateDashboard](#)

`displaySettings`

Die Anzeigeeinstellungen mit den folgenden Parametern:

- `numRows`— Anzahl der Zeilen im Dashboard-Layout. Jede Zeile ist `cellSize`-breit.
- `numColumns`— Anzahl der Spalten im Dashboard-Layout. Jede Spalte ist `cellSize`-breit.
- `cellSize`— (Optional) Die Größe einer Zelle im Layout in Pixeln. Es muss eine positive Zahl sein. Die Standardeinstellung ist 10.
- `significantDigits`— (Optional) Anzahl der signifikanten Ziffern, die im Dashboard angezeigt werden sollen. Die Standardeinstellung ist 4.

`querySettings`

Die Abfrageinformationen mit dem folgenden Parameter:

- `refreshRate`— (Optional) Die Geschwindigkeit, mit der Daten aktualisiert werden, in Millisekunden. Akzeptiert die folgenden Werte: 1000, 5000, 10000, 60000, 300000.

`defaultViewport`

Wenn nicht angegeben, werden standardmäßig die letzten fünf Minuten verwendet. Enthält die folgenden Parameter:

- `duration`— (Optional) Legt fest, wie weit in die Vergangenheit Daten ausgehend von der Gegenwart abgefragt werden sollen.
- `start`— (Optional) Es ist vom Typ Datum. Der Startzeitbereich für die Datenabfrage. Es muss ein `end` Datum angegeben werden.
- `end`— (Optional) Es ist vom Typ Datum. Der Endzeitbereich für die Datenabfrage. Es muss ein `start` Datum angegeben werden.

`widgets`

Eine Liste von Strukturen für Widget-Definitionen, die die folgenden Informationen enthalten:

`type`

Der Typ des Widgets. AWS IoT SiteWise bietet die folgenden Widget-Typen:

- `xy-plot`— Je nach Konfiguration ein Liniendiagramm oder ein Streudiagramm.
- `bar-chart`— Ein Balkendiagramm.
- `kpi-chart`— Ein Diagramm mit wichtigen Leistungsindikatoren.
- `status-timeline`— Ein Status-Widget, das Zeitreihendaten aus einer oder mehreren Datenquellen visualisiert und darin navigiert.
- `text`— Ein Text-Widget.
- `table`— Ein Tabellen-Widget.

`id`

Eine eindeutige Kennung für das Widget.

`x`

Die horizontale Position des Widgets, beginnend von der linken Seite des Dashboards. Dieser Wert bezieht sich auf die Position des Widgets im Raster des Dashboards.

`y`

Die vertikale Position des Widgets, beginnend am oberen Rand des Dashboards. Dieser Wert bezieht sich auf die Position des Widgets im Raster des Dashboards.

Z

Die relative Reihenfolge der Widgets. Ein größeres Z-Wert-Widget wird vor dem Widget mit niedrigerem Z-Wert angezeigt, wenn sie sich überschneiden.

width

Die Breite des Widgets, ausgedrückt in der Anzahl der Zellen auf dem Dashboard.

height

Die Höhe des Widgets, ausgedrückt in der Anzahl der Zellen auf dem Dashboard.

properties

Eine Liste der Eigenschaften des Widgets. Sie variiert je nach Art des Widgets. Einzelheiten finden Sie im [IoT App Kit](#).

Example Dashboard-Beispieldefinition

Im folgenden Beispiel wird ein Dashboard aus einer Nutzlast definiert, die in einer JSON-Datei gespeichert ist.

```
aws iotsitewise create-dashboard \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE \  
  --dashboard-name "Example Dashboard" \  
  --dashboard-definition file://dashboard-definition.json
```

Das folgende JSON-Beispiel für `dashboard-definition.json` definiert ein Dashboard mit den folgenden Visualisierungs-Widgets:

```
{  
  "displaySettings": {  
    "numColumns": 200,  
    "numRows": 1000,  
    "cellSize": 20,  
    "significantDigits": 4  
  },  
  "widgets": [{  
    "id": "0t73JcxUoc6oEXAMPLE",  
    "type": "xy-plot",  
    "width": 33,  
    "height": 20,  
  }  
]
```

```

"x": 0,
"y": 0,
"z": 0,
"properties": {
  "aggregationType": "AVERAGE",
  "queryConfig": {
    "source": "iotsitewise",
    "query": {
      "assets": [{
        "assetId": "97c97abf-e883-47bb-a3f4-EXAMPLE",
        "properties": [{
          "propertyId": "97cc61f4-57a4-4c5f-a82c-EXAMPLE",
          "refId": "692ce941-f3d9-4074-a297-EXAMPLE",
          "aggregationType": "AVERAGE",
          "color": "#7d2105",
          "resolution": "1m"
        }]
      }],
      "properties": [],
      "assetModels": [],
      "alarms": [],
      "alarmModels": []
    }
  },
  "line": {
    "connectionStyle": "linear",
    "style": "solid"
  },
  "symbol": {
    "style": "filled-circle"
  },
  "axis": {
    "yVisible": true,
    "xVisible": true
  },
  "legend": {
    "visible": true,
    "position": "right",
    "width": "30%",
    "height": "30%",
    "visibleContent": {
      "unit": true,
      "asset": true,
      "latestValue": true,

```



```

        "latestAlarmStateValue": true,
        "maxValue": false,
        "minValue": false
    }
}
}, {
    "id": "fto7rF40Ny1EXAMPLE-G",
    "type": "bar-chart",
    "width": 33,
    "height": 20,
    "x": 0,
    "y": 20,
    "z": 0,
    "properties": {
        "aggregationType": "AVERAGE",
        "queryConfig": {
            "source": "iotsitewise",
            "query": {
                "assets": [{
                    "assetId": "97c97abf-e883-47bb-a3f4-EXAMPLE",
                    "properties": [{
                        "propertyId": "c84ca8f3-3dea-478a-afec-EXAMPLE",
                        "aggregationType": "AVERAGE",
                        "refId": "2960b958-2034-4d6e-bcc2-EXAMPLE"
                    }]
                }]
            },
            "properties": [],
            "assetModels": [],
            "alarms": [],
            "alarmModels": [],
            "requestSettings": {
                "aggregation": "AVERAGE"
            }
        }
    },
    "axis": {
        "showX": true,
        "showY": true
    },
    "styleSettings": {
        "2960b958-2034-4d6e-bcc2-360f1f02e505": {
            "color": "#7d2105"
        }
    }
}
}

```

```
    }  
  }  
}],  
"querySettings": {  
  "refreshRate": 5000  
}  
}
```

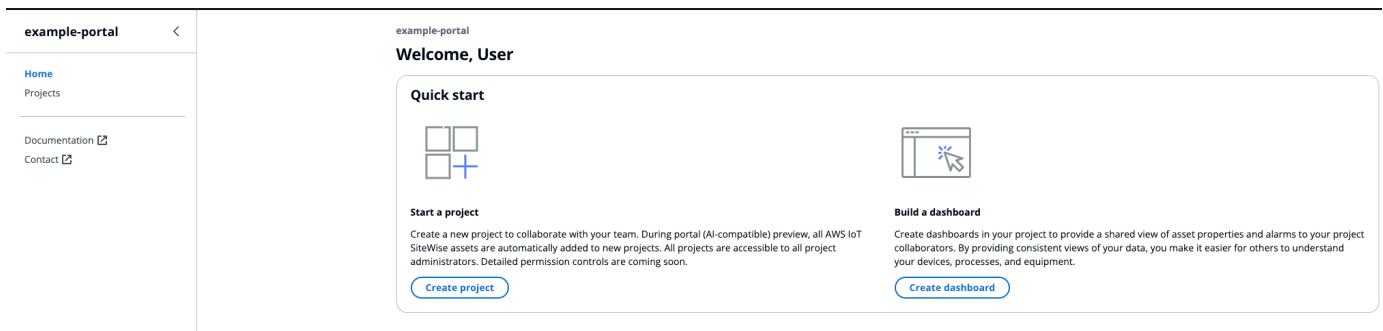
Anmeldung am Portal

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Benutzerlogin

1. Geben Sie in Ihrem Browser die Anwendungs-URL ein.
2. Geben Sie Ihren Benutzernamen und Ihr Passwort ein und klicken Sie auf die Schaltfläche Anmelden.
3. Sie sind jetzt bei der Anwendung angemeldet.



Erstellen eines Projekts

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Projekt erstellen

1. Ein Projekt wird auf zwei Arten erstellt:
 - a. Wählen Sie auf der Startseite im Bereich Willkommen unter Schnellstart die Option Projekt erstellen aus.
 - b. Wählen Sie im linken Navigationsbereich Projekte aus. Wählen Sie in der oberen rechten Ecke Erstellen aus, um ein Projekt zu erstellen.
2. Geben Sie im Abschnitt Projekte erstellen einen Projektnamen und optional eine Beschreibung ein.
3. Wählen Sie Erstellen aus.

example-portal

example-portal <

Home
Projects

Documentation
Contact

Home > Projects > Create

Create project

Create a project to host your dashboards.

Project name

Example project

The project name must contain from 1 and 256 characters. Character count: 15/256.

Project description - optional

An example project.

The project description must contain from 1 and 2048 characters. Character count: 19/2048.

Cancel Create

Projekt aktualisieren

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Projekt bearbeiten

1. Wählen Sie die Schaltfläche Aktualisieren in der oberen rechten Ecke der Projektseite, um die Projektdetails zu bearbeiten.
2. Ändern Sie den Namen des Projekts, indem Sie den Projektnamen bearbeiten.
3. Ändern Sie die Beschreibung des Projekts, indem Sie die Beschreibungsdetails bearbeiten.
4. Wählen Sie Aktualisieren aus, um Ihre Änderungen zu speichern.

The screenshot shows the 'Example-project' page in the AWS IoT SiteWise console. The page has a dark header with 'example-portal' and a user profile. A left sidebar contains links for 'Home', 'Projects', 'Documentation', and 'Contact'. The main content area shows the project details and a list of dashboards. The 'Update' button in the top right corner of the project details section is highlighted with a red rectangle.

Project details

Description	Creation date	Last update date
Example project	10/25/2024, 11:17:28 AM	10/25/2024, 11:17:28 AM

Dashboards

Filter dashboards

Name	Description	Creation date	Last update date
example dashboard	dashboard	10/25/2024, 11:17:52 AM	10/25/2024, 11:17:52 AM

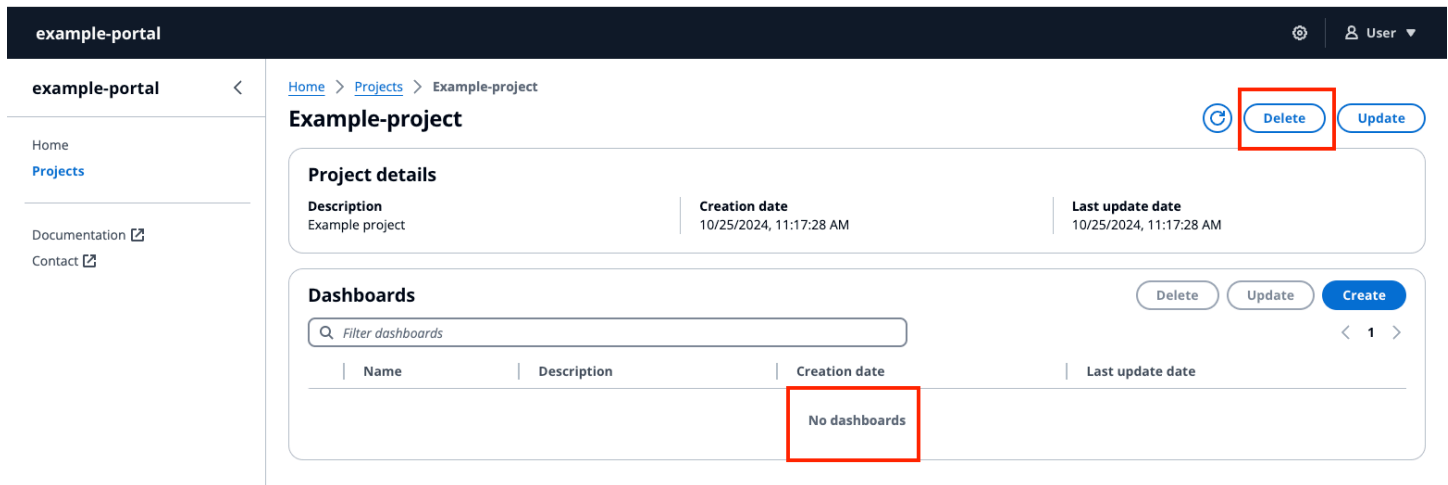
Projekt löschen

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Projekt löschen

1. Sie können das Projekt erst löschen, nachdem alle Dashboards im Projekt gelöscht wurden.
2. Wählen Sie oben rechts auf der Projektseite die Schaltfläche Löschen aus.
3. Bestätigen Sie erneut, dass Sie das Projekt löschen möchten.
4. Wählen Sie Löschen, um das Projekt zu löschen.



Erstellen eines Dashboards

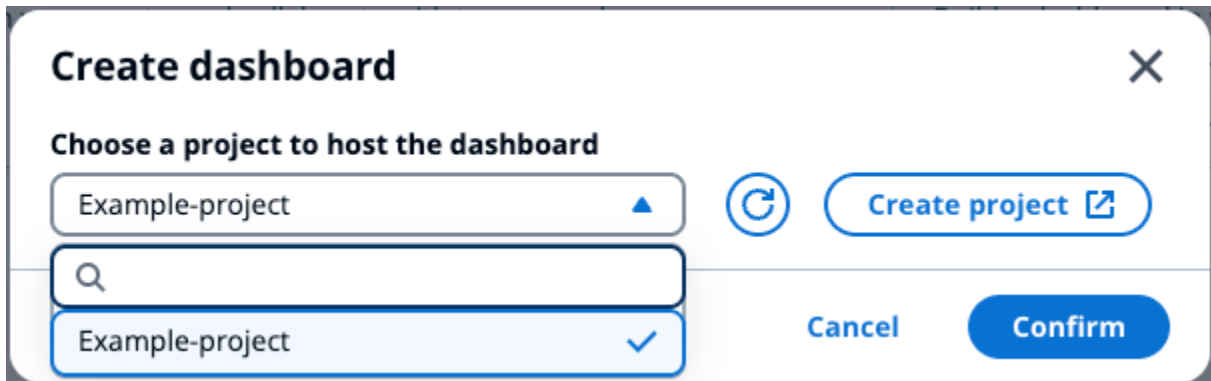
Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Erstellen eines Dashboards

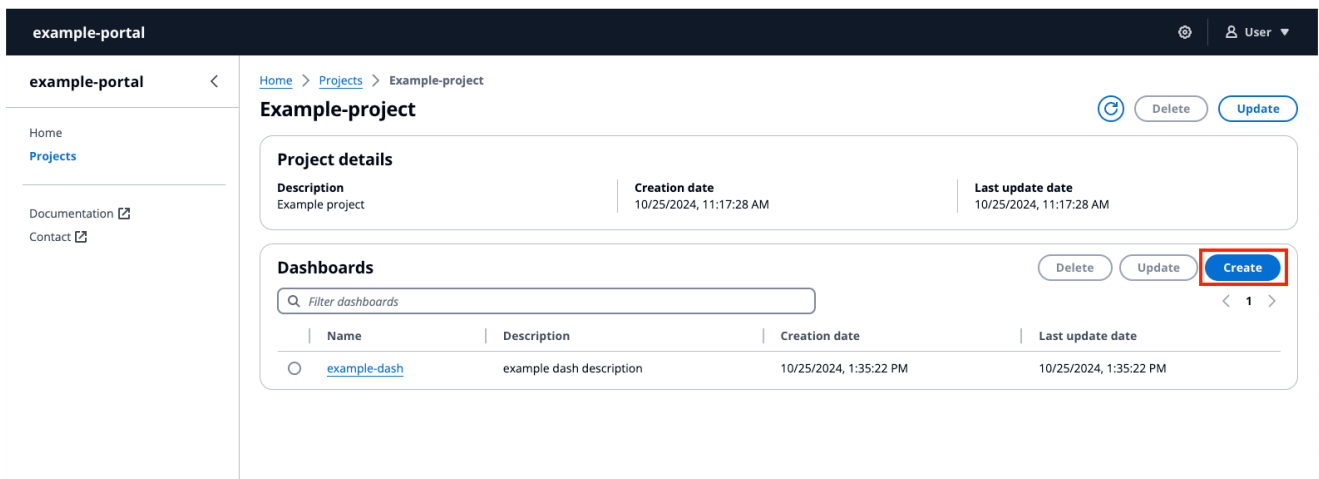
1. Erstellen Sie ein Dashboard auf zwei Arten:
 - a. Erstellen Sie auf der Startseite unter Dashboard erstellen ein Dashboard.
 - i. Um das Dashboard in einem vorhandenen Projekt zu erstellen, wählen Sie einen Projektnamen aus dem Drop-down-Menü unter Wählen Sie ein Projekt aus, das das Dashboard hosten soll.

- ii. Wenn Sie noch kein Projekt haben, wählen Sie Projekt erstellen und dann Bestätigen aus.



The image shows a 'Create dashboard' modal dialog. It has a title bar with a close button (X). The main heading is 'Create dashboard'. Below it is the instruction 'Choose a project to host the dashboard'. There is a dropdown menu showing 'Example-project' with a blue triangle icon. To the right of the dropdown is a circular refresh icon and a 'Create project' button with an external link icon. Below the dropdown is a search bar with a magnifying glass icon. Below the search bar is a list item 'Example-project' with a blue checkmark icon. At the bottom right are 'Cancel' and 'Confirm' buttons.

- b. Erstellen Sie im Bereich Projekte unter Dashboards ein Dashboard aus einem Projekt.



The image shows the 'Example-project' page in the 'example-portal'. The page has a dark header with 'example-portal' and a user profile. The left sidebar shows 'example-portal' with a back arrow, and a list of links: 'Home', 'Projects' (highlighted), 'Documentation', and 'Contact'. The main content area has a breadcrumb 'Home > Projects > Example-project'. The title is 'Example-project' with a refresh icon, 'Delete', and 'Update' buttons. Below the title is the 'Project details' section with a table:

Description	Creation date	Last update date
Example project	10/25/2024, 11:17:28 AM	10/25/2024, 11:17:28 AM

Below the project details is the 'Dashboards' section with a search bar 'Filter dashboards' and 'Delete', 'Update', and 'Create' buttons. The 'Create' button is highlighted with a red box. Below the buttons is a table:

Name	Description	Creation date	Last update date
example-dash	example dash description	10/25/2024, 1:35:22 PM	10/25/2024, 1:35:22 PM

2. Wählen Sie in der oberen rechten Ecke Erstellen aus.
3. Geben Sie einen Namen für das Dashboard und optional eine Beschreibung des Dashboards ein.
4. Wählen Sie Erstellen aus.

example-portal

example-portal

Home

Projects

Documentation

Contact

Home > Projects > Example-project > Create dashboard

Create dashboard

Create a dashboard within your project.

Dashboard name

example2-dash

The dashboard name must contain from 1 and 256 characters. Character count: 13/256.

Dashboard description - optional

dash description

The dashboard description must contain from 1 and 2048 characters. Character count: 16/2048.

Cancel

Create

5. Konfigurieren Sie Ihr neu erstelltes Dashboard.

Aktualisieren eines Dashboards

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Im Abschnitt Dashboards werden die Dashboards im Projekt aufgeführt. Wählen Sie ein Dashboard aus der Liste aus.

Aktualisieren eines Dashboards

1. Wählen Sie ein Dashboard aus, das aktualisiert werden soll.

example-portal

Home > Projects > Example-project

Example-project

Refresh

Delete

Update

Project details

Description

Example project

Creation date

10/25/2024, 11:17:28 AM

Last update date

10/25/2024, 11:17:28 AM

Dashboards

Filter dashboards

Name

Description

Creation date

Last update date

example-dash

example dash description

10/25/2024, 1:35:22 PM

10/25/2024, 1:35:22 PM

Delete

Update

Create

2. Aktualisieren Sie den Namen des Dashboards und optional die Beschreibung des Dashboards. Wählen Sie Aktualisieren aus, um die Änderungen zu speichern.

Home > Projects > example-dash > Update

Update dashboard: example-dash

Dashboard name

example-update-dash-name

The dashboard name must contain from 1 and 256 characters. Character count: 24/256.

Dashboard description - optional

update dash description

The dashboard description must contain from 1 and 2048 characters. Character count: 23/2048.

Cancel

Update

Löschen eines Dashboards

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Im Abschnitt Dashboards werden die Dashboards im Projekt aufgeführt. Wählen Sie ein Dashboard aus der Liste aus.

Löschen eines Dashboards

1. Wählen Sie ein zu löschendes Dashboard aus.

example-portal

User

Home > Projects > Example-project

Example-project

Delete

Update

Project details

Description

Example project

Creation date

10/25/2024, 11:17:28 AM

Last update date

10/25/2024, 11:17:28 AM

Dashboards

Filter dashboards

Delete

Update

Create

< 1 >

Name	Description	Creation date	Last update date
example-dash	example dash description	10/25/2024, 1:35:22 PM	10/25/2024, 1:35:22 PM

Löschen eines Dashboards

781

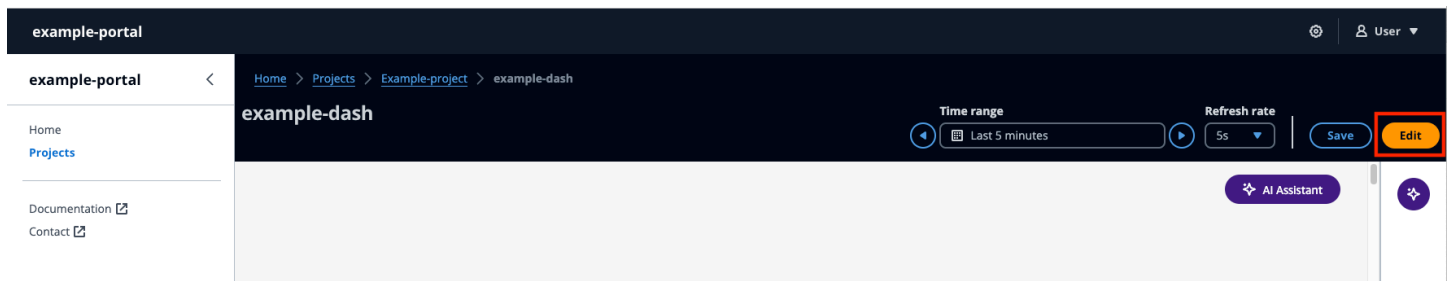
2. Wählen Sie Löschen aus, um das Dashboard zu löschen. Dies kann nicht rückgängig gemacht werden.

Dashboard konfigurieren

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Im Abschnitt Dashboards werden die Dashboards im Projekt aufgeführt. Wählen Sie ein Dashboard aus der Liste aus. Im Bearbeitungsmodus können Sie Ihr Dashboard konfigurieren, indem Sie Widgets hinzufügen und diese konfigurieren. Mit der Vorschau-Schaltfläche können Sie Ihre Änderungen visualisieren.



Schritte zur Konfiguration Ihres Dashboards:

- Ziehen Sie verschiedene Arten von Daten-Widgets zur Datenvisualisierung per Drag & Drop auf die Dashboard-Leinwand.
- Fügen Sie Daten über den Ressourcen-Explorer auf der linken Seite zu den gewünschten Widgets hinzu. Der Ressourcen-Explorer besteht aus den Abschnitten Modellerte, Unmodellerte und Dynamische Objekte. Suchen Sie nach dem Namen des Vermögenswerts oder des Eigenschaftsnamens. Wählen Sie die Eigenschaft aus, die Sie hinzufügen möchten, und klicken Sie auf Hinzufügen.
- Passen Sie das Layout und den Stil an, indem Sie die Konfigurationen der Widgets ändern. Konfigurieren Sie Komponenten wie Titel, Schwellenwerte und andere Konfigurationsspezifikationen.
- Konfigurieren Sie den Zeitraum, in dem Daten angezeigt werden.

- Wählen Sie den Zeitraum aus, in dem Daten angezeigt werden. Wählen Sie oben rechts einen Zeitraum und eine Aktualisierungsrate aus und personalisieren Sie den Bereich. Wählen Sie im Menü eine Rate aus, mit der die Daten aktualisiert werden sollen.
- Wählen Sie den Zeitraum in einem Widget aus, indem Sie das Scrollrad der Trackball-Maus verwenden oder mit der rechten Maustaste klicken. Dadurch wird der Anzeigezeitraum verschoben.
- Wählen Sie Speichern.

Themen

- [Ressourcen-Explorer](#)
- [Widgets](#)
- [Widgets konfigurieren](#)
- [Verwenden Sie Widgets](#)
- [Alarme in Widgets](#)
- [AWS IoT SiteWise Verwendung des Assistenten in Widgets](#)
- [Beispielfragen, die Sie dem Assistenten stellen sollten AWS IoT SiteWise](#)

Ressourcen-Explorer

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

In diesem Abschnitt werden modellierte, nicht modellierte und dynamische Assets beschrieben. Wählen Sie Assets aus einem der drei aus, fügen Sie sie Ihren Widgets hinzu und visualisieren Sie sie.

Themen

- [Modelliert](#)
- [Nicht modelliert](#)

- [Dynamische Vermögenswerte](#)

Modelliert

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

In diesem Abschnitt wird der Prozess der Auswahl und Visualisierung modellierter Anlagen beschrieben.

Auswahl von Vermögenswerten

Assets können wie folgt abgefragt werden:

- Suchen Sie nach einem Asset-Namen. Verwenden Sie einen Platzhalter. * Gibt beispielsweise Asset-Namen Wind* zurück, die mit dem Text Wind beginnen. Sie müssen [sich mit integrieren AWS IoT TwinMaker](#), um diese Funktion nutzen zu können.
- Alle Assets werden standardmäßig aufgelistet.

Filtern Sie aus den aufgelisteten Assets nach Name, Beschreibung, ID oder Asset-Modell-ID. Wählen Sie ein Asset aus, um dessen Eigenschaften (Datenströme) und Alarmer aufzulisten.

Auswahl des Datenstroms

Datenströme sind unter dem Menü Datenströme aufgeführt. Filtern Sie die Datenströme, die in der <https://docs.aws.amazon.com/iot-sitewise/neuesten> [Version nach Eigenschaftsmetadaten](#) aufgelistet/ APIReference. Wählen Sie je nach ausgewähltem Widget einen oder mehrere Datenströme aus.

- KPI und Gauge unterstützen nur einen einzigen Datenstrom.
- Die übrigen Widgets unterstützen mehrere Datenströme mit Mehrfachauswahl.

Auswahl des Alarms

AWS IoT SiteWise Alarmer sind unter dem Menü Alarm Data Streams aufgeführt. Filtern Sie die aufgelisteten Alarmdatenströme nach Alarmmetadaten. Name, Eingabeeigenschaft und ID des

zusammengesetzten Modells sind einige Metadaten, die zum Filtern verwendet werden. Wählen Sie je nach ausgewähltem Widget einen oder mehrere Datenströme aus.

- KPI und Gauge unterstützen nur einen einzigen Alarm.
- Die übrigen Widgets unterstützen mehrere Alarme mit Mehrfachauswahl.

Visualisierung modellierter Anlagen

1. Ziehen Sie das Widget auf die Leinwand. Wählen Sie die Eigenschaften für jedes Widget-Bedienfeld aus, um ein Dashboard zu erstellen.
2. Die Option Filter filtert die Assets, um das zu visualisierende Asset auszuwählen. Die Filterung erfolgt nach Text, Eigenschaft oder Wert. Die Filterung bezieht sich auf in den Browser geladene Elemente und nicht auf die Backend-Filterung.
3. Suchen Sie nach einem Asset, das Sie Ihrem Widget hinzufügen möchten.
4. Fügen Sie das Asset dem Widget auf der Arbeitsfläche hinzu.
5. Wählen Sie „Zurücksetzen“, um ein anderes Asset auszuwählen, oder nehmen Sie Änderungen an dem ausgewählten Asset vor.
6. Speichern Sie das Dashboard. Wählen Sie im Vorschaumodus verschiedene Assets aus dem Drop-down-Menü aus, um die Eigenschaften unter den einzelnen Assets zu überwachen, ohne die Datenfelder rekonstruieren zu müssen.

Note

Im Einstellungsrad auf der rechten Seite werden Einstellungen angezeigt, aus denen der Benutzer auswählen kann, z. B. Seitenformat, Erste Spalten im Sticky, Letzte Spalten behalten und Spalteneinstellungen. Passen Sie Ihre Einstellungen an und wählen Sie Bestätigen, um die Änderungen zu übernehmen.

Modeled

Unmodeled

Dynamic assets

Assets (1)

Browse through your asset hierarchy and select an asset to view its associated data streams.

Root

Search

Q

Search for resources

Search

Filter

Q

Filter assets by text, property, or value

< 1 > ⚙

Name

Description

●

Demo Wind Farm Asset

Asset properties (8)

Select a modeled datastream to add to a selected widget

Filter

Q

Filter asset properties by text, property, or value

< 1 > ⚙

☐	Name	Unit	Data ty...	Latest ...	Latest value time
☐	Total Aver...		DOUBLE	37478.2303	2024-10-01 09:50:00 p.m.
☐	Total Aver...		DOUBLE	6.0000	2022-10-26 03:42:43 p.m.
☐	Total Aver...		DOUBLE	555.0000	2022-10-13 11:59:49 p.m.
☐	Code		INTEGER	300.0000	2022-10-13 10:59:28 p.m.
☐	Reliability ...		STRING	Mary Major	2022-10-13 10:59:28 p.m.
☐	Location		STRING	Renton	2022-10-13 10:59:28 p.m.
☐	Total Over...		DOUBLE	900.0000	2024-10-01 09:50:00 p.m.
☐	recipient a...		STRING	54a88418-...	2022-10-26 03:42:43 p.m.

Nicht modelliert

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

In diesem Abschnitt wird beschrieben, wie Sie nach unmodellierten Datenströmen suchen und diese den zu visualisierenden Widgets hinzufügen.

Visualisierung unmodellierter Datenströme

1. Ziehen Sie das Widget auf die Leinwand. Wählen Sie die Eigenschaften für jedes Widget-Bedienfeld aus, um ein Dashboard zu erstellen.
2. Unmodellierte Datenströme sind im Abschnitt Zeitreihen aufgeführt. Sie haben Eigenschaften, die anpassbar sind.
3. Die Option Filter filtert die zu visualisierenden Datenströme. Die Filterung bezieht sich auf Datenströme, die in den Browser geladen wurden, und nicht auf die Back-End-Filterung.
4. Fügen Sie den Datenstream dem Widget auf der Arbeitsfläche hinzu.
5. Wählen Sie Zurücksetzen, um die Auswahl des Datenstroms aufzuheben.
6. Speichern Sie das Dashboard. Wählen Sie im Vorschaumodus verschiedene Assets aus dem Drop-down-Menü aus, um die Eigenschaften unter den einzelnen Assets zu überwachen, ohne die Datenfelder rekonstruieren zu müssen.

Note

Im Einstellungsrad auf der rechten Seite werden Einstellungen angezeigt, aus denen der Benutzer auswählen kann, z. B. Seitenformat, Erste Spalten im Sticky, Letzte Spalten behalten und Spalteneinstellungen. Passen Sie Ihre Einstellungen an und wählen Sie Bestätigen, um die Änderungen zu übernehmen.

Modeled
Unmodeled
Dynamic assets

Time series (1)

Select a unmodeled datastream to add to a selected widget

Filter < 1 > ⚙

☒	Alias	ID	Data type	Latest val...	Latest valu...
☒	DemoDisass...	33e78bb9-39...	DOUBLE	3.0283	2024-10-01 0...

Dynamische Vermögenswerte

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Mit dem neuen SiteWise Monitor können Kunden dynamisch zwischen Anlagen für ein ausgewähltes Anlagenmodell wechseln. Sie können Eigenschaften verschiedener Vermögenswerte visualisieren, indem Sie sie aus einem Drop-down-Menü auswählen.

Dynamische Visualisierung von Vermögenswerten

1. Wählen Sie im Ressourcen-Explorer die Registerkarte Dynamische Assets aus.
2. Wählen Sie im Drop-down-Menü ein Asset-Modell aus, für das Sie Assets auflisten möchten.
3. Wählen Sie das Standard-Asset aus dem Drop-down-Menü aus.
4. Wählen Sie Asset-Modell festlegen, um das Asset-Modell auszuwählen.
5. Speichern Sie das Dashboard. Wählen Sie im Vorschaumodus verschiedene Assets aus dem Drop-down-Menü aus, um die Eigenschaften unter jedem Asset zu überwachen, ohne die Datenfelder neu zu konstruieren.

Resource explorer

Modeled | **Unmodeled** | **Dynamic assets**

Dynamic asset visualizations allow you to build one visualization to represent any asset of a specified asset model. [Learn more](#)

Asset model
Select an asset model to add the associated properties into your dynamic display.

Demo Wind Farm Asset Model 2022-Oct-14 05:59:28

Q Filter asset models

Demo Turbine Asset Model 2022-Oct-14 05:59:12
This is an asset model used in the IoT SiteWise Demo for representing a turbine in a wind farm. It will be deleted at the end of the demo.

Demo Wind Farm A Demo Turbine Asset Model 2022-Oct-14 05:59:12
This is an asset model used in the IoT SiteWise Demo for representing a wind farm. It will be deleted at the end of the demo. ✓

Finished loading asset models.

Set asset model

Widgets

Note

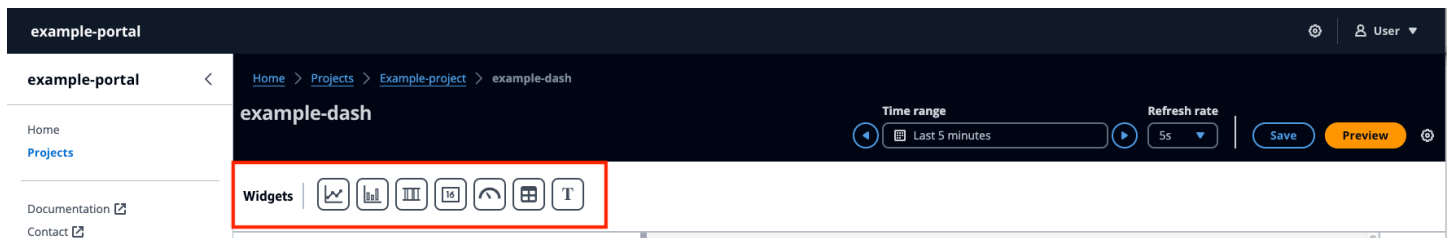
Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Widgets unterstützt eine Vielzahl von Funktionen, darunter Alarmer, leistungsstarkes Live-Streaming und eine reibungslose Synchronisation mit anderen IoT App Kit-Komponenten. Das Dashboard unterstützt die folgenden Widgets:

- Linie — Das Linien-Widget ist ein Visualisierungs-Widget, das Trends und Veränderungen im Laufe der Zeit anzeigt. Es besteht aus einer Reihe von Datenpunkten, die jeweils durch einen Punkt oder

eine Markierung dargestellt werden und durch gerade Liniensegmente miteinander verbunden sind, sodass ein Liniendiagramm entsteht. Es unterstützt eine Vielzahl von Funktionen, darunter Alarmer, Schwellenwerte, leistungsstarkes Live-Streaming und eine reibungslose Synchronisation mit anderen IoT App Kit-Komponenten. Dieses Widget ist anpassbar, um komplexe Daten klar und präzise zu kommunizieren.

- **Balkendiagramm** — Das Balkendiagramm ist ein leistungsstarkes Visualisierungstool, das Zeitreihendaten anzeigt. Es unterstützt eine Vielzahl von Funktionen, darunter Alarmer, leistungsstarkes Live-Streaming und eine reibungslose Synchronisation mit anderen IoT App Kit-Komponenten.
- **Zeitleiste** — Das Timeline-Widget bietet eine Möglichkeit, Zeitreihendaten aus Datenquellen zu visualisieren und darin zu navigieren. Es ist einzigartig für die Anzeige von Datenstromwerten in unterschiedlichen Farben auf der Zeitleiste. Es unterstützt eine Vielzahl von Funktionen, darunter Alarmer, leistungsstarkes Live-Streaming und reibungslose Synchronisation zwischen anderen IoT App Kit-Komponenten. Es eignet sich am besten für die Anzeige nichtnumerischer Datentypen/
- **KPI** — Die Komponente Key Performance Indicator (KPI) bietet eine kompakte Darstellung eines Überblicks über Ihre Anlageeigenschaften. Sie unterstützt Alarmer und Schwellenwerte. Diese Übersicht bietet wichtige Einblicke in die Gesamtleistung Ihrer Geräte, Anlagen und Prozesse. KPI unterstützt nur einen einzelnen Datenstrom oder Alarm und nicht mehrere Datenströme.
- **Gauge** — Die Gauge-Komponente bietet eine kompakte Darstellung eines Überblicks über Ihre Asset-Eigenschaften. Sie wird verwendet, um wichtige Einblicke in die Gesamtleistung Ihrer Geräte, Anlagen oder Prozesse zu visualisieren. Es ist funktionell identisch mit KPI, unterscheidet sich jedoch visuell. Gauge zeigt den Wert, den Schwellenwert und den Wertebereich des Datenstroms an. Mit Gauge können Sie mit AWS IoT Daten aus einer oder mehreren Datenquellen interagieren.
- **Tabelle** — Die Tabellenkomponente bietet ein kompaktes Formular für die Anzeige eines oder mehrerer Datenströme aus einer oder mehreren Zeitreihendatenquellen. Sie zeigt Vermögenswerte mit Eigenschaft, Neuestem Wert und Einheit in tabellarischer Form an. Unterstützt AWS IoT SiteWise Alarmer.
- **Text** — Das Text-Widget hilft beim Schreiben von Text mit verschiedenen Farben und Schriftarten. Sie können einen Link erstellen, indem Sie einen Text mit einer URL verknüpfen. Die Felder Eigenschaften und Schwellenwerte sind für dieses Widget nicht aktiviert.



Widgets konfigurieren

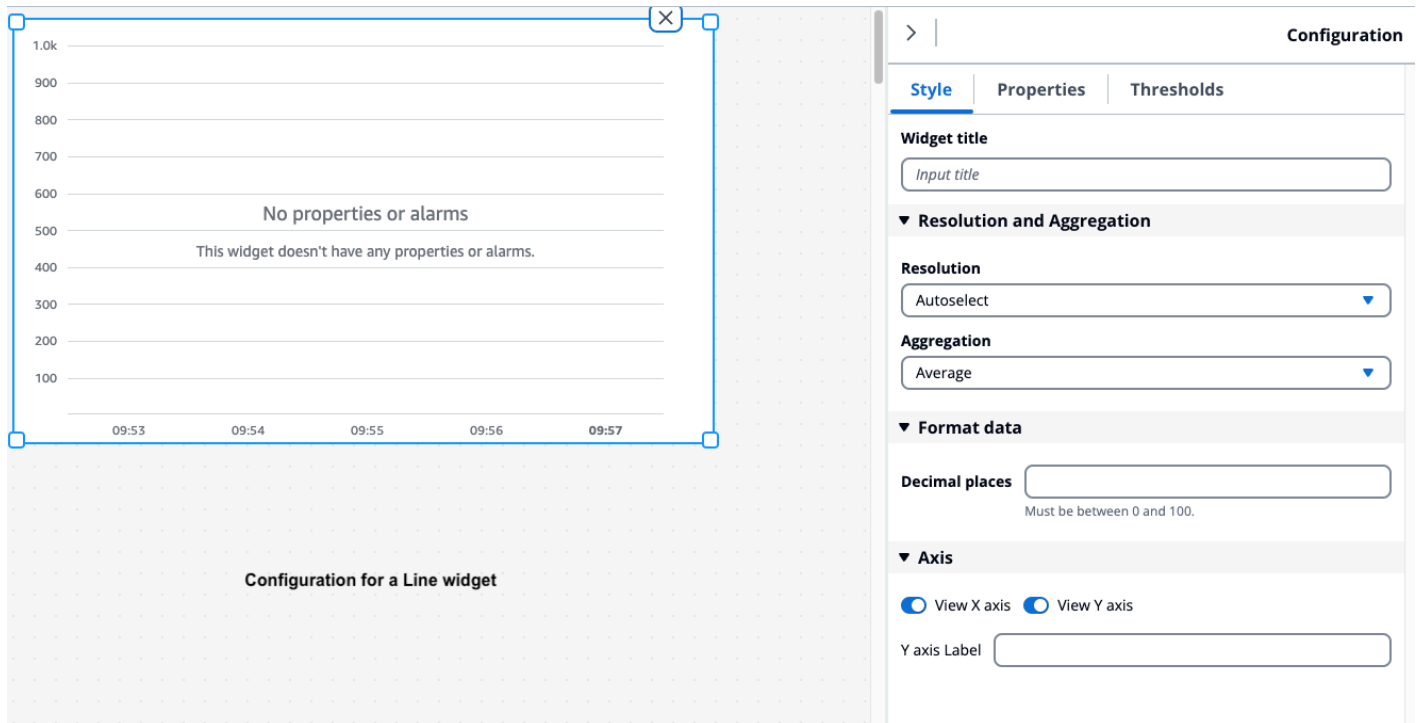
Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Sobald das Widget zum Dashboard hinzugefügt wurde, können Sie das Widget konfigurieren, indem Sie im rechten Bereich auf das Konfigurationssymbol klicken.

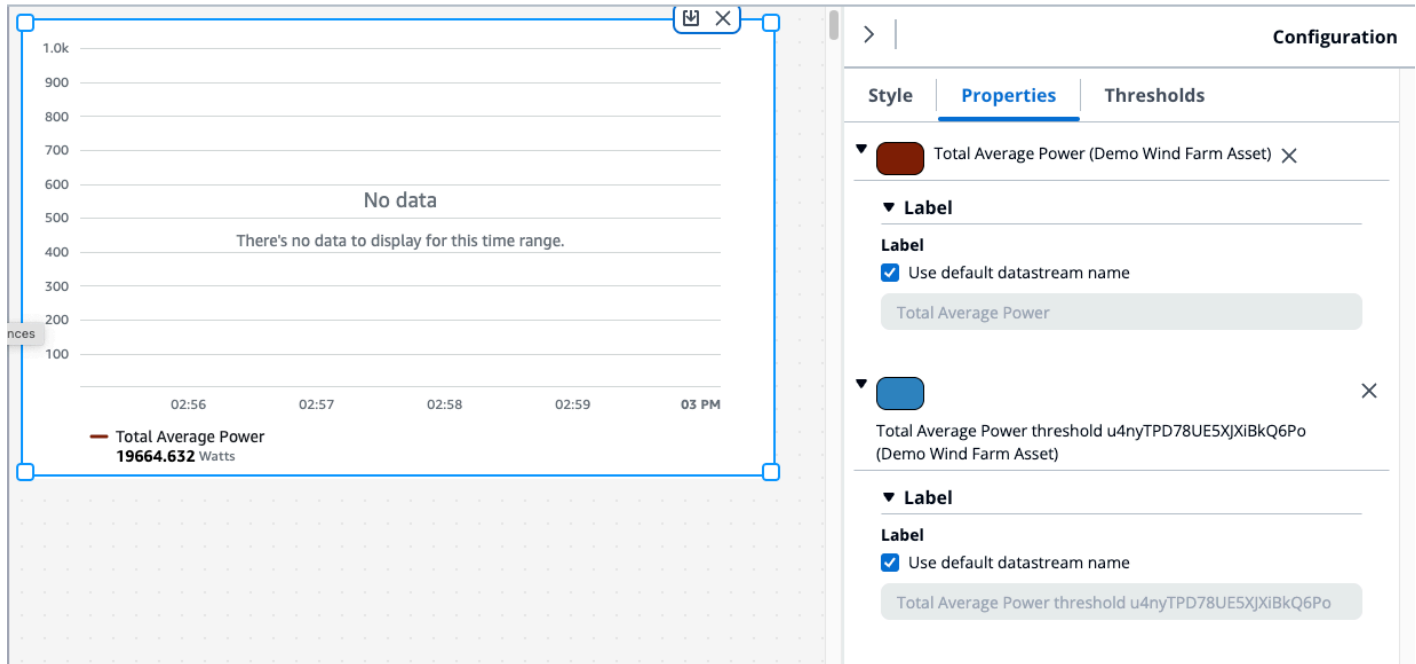
- **Stil** — Fügen Sie dem Widget-Titel einen Titel hinzu. Verschiedene Widgets haben unterschiedliche Konfigurationen. Nachfolgend sind einige Beispiele aufgeführt.
- **Balken-Widget:**
 - **Auflösung und Aggregation** — Legen Sie hier Werte für Auflösung und Aggregation fest.
 - **Daten formatieren** — Stellen Sie die Anzahl der anzuzeigenden Dezimalstellen als Dezimalstellen ein.
 - **Anzeigestil** — Wählen Sie die anzuzeigenden Werte aus.
 - **Achse** — Wählen Sie, ob die Achse angezeigt werden soll.
- **Linien-Widget:**
 - **Auflösung und Aggregation** — Legen Sie hier Werte für Auflösung und Aggregation fest.
 - **Daten formatieren** — Stellen Sie die Anzahl der anzuzeigenden Dezimalstellen als Dezimalstellen ein.
 - **Y-Achse** — Fügen Sie eine Bezeichnung sowie Min- und Max-Werte hinzu.
 - **Widget-Stil** — Wählen Sie die Werte Linientyp, Linienstil, Linienstärke und Datenpunktform aus.
 - **Legende** — Wählen Sie „Ausrichtung“ und „Anzeige“.
- **Messgerät-Widget:**

- **Auflösung und Aggregation** — Legen Sie hier Werte für Auflösung und Aggregation fest.
- **Daten formatieren** — Stellen Sie die Anzahl der anzuzeigenden Dezimalstellen als Dezimalstellen ein.
- **Anzeigestil** — Wählen Sie die anzuzeigenden Werte aus.
- **Y-Achse** — Fügen Sie eine Bezeichnung sowie Min - und Max-Werte hinzu.
- **Schriften** — Wählen Sie die Werte für Schriftgröße, Einheitsschriftgröße und Schriftgröße für Etiketten aus.



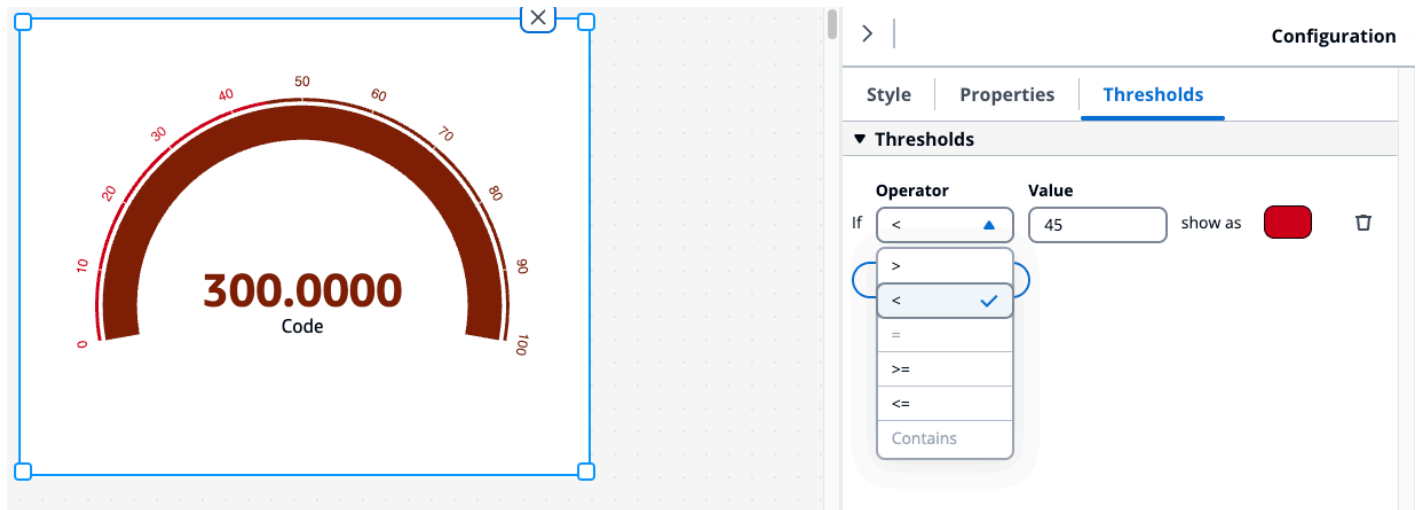
- **Eigenschaften** — Alle Eigenschaften von Widgets sind in diesem Abschnitt aufgeführt. Verschiedene Widgets haben unterschiedliche Eigenschaften. Nachfolgend sind einige Beispiele aufgeführt.
 - **Linien-Widget:**
 - **Bezeichnung** — Wählen Sie, ob Sie den Standard-Datenstromnamen verwenden oder einen neuen Namen angeben möchten.
 - **Stil** — Stellen Sie Linientyp und Linienstil auf die Anzahl der anzuzeigenden Dezimalstellen ein.
 - **Y-Achse** — Wählen Sie Werte für den Standardstil aus, zeigen Sie die Steuerelemente für die Y-Achse an und legen Sie die Min - und Max-Werte fest.
 - **Tabellen-Widget:**

- **Bezeichnung** — Wählen Sie, ob Sie den Standard-Datenstromnamen verwenden oder einen neuen Namen angeben möchten.
- **Tabellen-Widget:**
 - **Bezeichnung** — Wählen Sie, ob Sie den Standard-Datenstromnamen verwenden oder einen neuen Namen angeben möchten.



- **Schwellenwerte** — Fügen Sie einen Schwellenwert für ein Widget hinzu. Verschiedene Widgets haben unterschiedliche Konfigurationen. Nachfolgend sind einige Beispiele aufgeführt.
- **Balkendiagramm-Widget:**
 - Wählen Sie Schwellenwert hinzufügen aus, um ihn dem Widget hinzuzufügen.
 - Wählen Sie Operator und geben Sie einen Wert für den Schwellenwert ein. Passen Sie den Schwellenwert mit einer Farbe aus der Farbpalette an.
 - Sie können wählen, ob der Schwellenwert auf alle Daten angewendet werden soll.
- **Linien-Widget:**
 - Wählen Sie Schwellenwert hinzufügen aus, um ihn dem Widget hinzuzufügen.
 - Wählen Sie Operator und geben Sie einen Wert für den Schwellenwert ein. Passen Sie den Schwellenwert mit einer Farbe aus der Farbpalette an.
 - Wählen Sie aus dem Drop-down-Menü aus, wie Schwellenwerte angezeigt werden sollen.
- **Messgerät-Widget:**
 - Wählen Sie Schwellenwert hinzufügen aus, um ihn dem Widget hinzuzufügen.

- Wählen Sie Operator und geben Sie einen Wert für den Schwellenwert ein. Passen Sie den Schwellenwert mit einer Farbe aus der Farbpalette an.



Verwenden Sie Widgets

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Sie können Widgets im Dashboard einzeln oder durch Mehrfachauswahl verwenden.

Widgets im Dashboard bearbeiten

Wählen Sie ein einzelnes Widget und bearbeiten Sie es. Um mehrere Widgets im Dashboard zu bearbeiten, klicken Sie bei gedrückter Umschalttaste mit der linken Maustaste und wählen Sie alle Widgets im Dashboard aus. Nach der Auswahl können Benutzer neue Datenstreams hinzufügen und den Widget-Titel in den Style-Konfigurationseinstellungen ändern. Der Titel wird für alle Widgets im Dashboard geändert.

Klicken Sie mit der rechten Maustaste auf die Leinwand und gehen Sie wie folgt vor:

- Kopieren — Fügt eine Kopie des Widgets zur Arbeitsfläche hinzu.
- Löschen — Löscht das Widget.

- In den Vordergrund bringen — Bringt das ausgewählte Widget in den Vordergrund der Leinwand.
- In den Hintergrund senden — Sendet das ausgewählte Widget an die Rückseite der Leinwand.

Ändern Sie die Größe von Widgets

Ändern Sie die Größe von Widgets einzeln oder in einer Gruppe, indem Sie die Widgets im Dashboard mehrfach auswählen.

Um die Größe von Widgets zu ändern:

- Um die Größe eines einzelnen Widgets zu ändern, wählen Sie das Widget aus und ziehen Sie es an einer Ecke, um seine Größe zu ändern.
- Um die Größe mehrerer Widgets zu ändern, wählen Sie mehrere Widgets mit gedrückter Umschalttaste und ziehen Sie sie an einer Ecke, um ihre Größe zu ändern.

Löschen Sie Widgets im Dashboard

Löschen Sie Widgets einzeln oder in einer Gruppe, indem Sie die Widgets im Dashboard mehrfach auswählen.

Um Widgets zu löschen:

- Um ein einzelnes Widget zu löschen, wählen Sie das Widget aus, klicken Sie mit der rechten Maustaste und wählen Sie Löschen. Sie können das Widget auch auswählen und in der rechten oberen Ecke auf X klicken, um es zu löschen.
- Um mehrere Widgets zu löschen, wählen Sie mehrere Widgets aus, indem Sie bei gedrückter Umschalttaste mit der linken Maustaste klicken, dann mit der rechten Maustaste klicken und Löschen wählen.

Alarme in Widgets

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Alarme warnen Sie und Ihr Team, wenn Geräte oder Prozesse nicht optimal funktionieren. Optimale Leistung eines Geräts oder Prozesses bedeutet, dass die Werte für bestimmte Metriken innerhalb des Bereichs zwischen einem unteren und einem oberen Grenzwert liegen sollten. Wenn diese Messwerte außerhalb ihres Betriebsbereichs liegen, müssen die Anlagenbetreiber benachrichtigt werden, damit sie das Problem beheben können. Mithilfe von Alarmen können Sie Probleme schnell erkennen und die Bediener benachrichtigen, um die Leistung Ihrer Geräte und Prozesse zu maximieren.

Auf der Registerkarte Modelliert des Ressourcen-Explorers finden Sie einen Alarm, der einem Asset zugeordnet ist.

- Suchen Sie nach einem Asset und wählen Sie es aus.
- Scrollen Sie an der Tabelle Datenstreams vorbei nach unten zum Abschnitt Alarm Data Streams und erweitern Sie ihn.
- Wählen Sie in der Tabelle Alarme einen Alarm aus und klicken Sie auf Hinzufügen.

Themen

- [Alarme in verschiedenen Widgets](#)

Alarme in verschiedenen Widgets

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

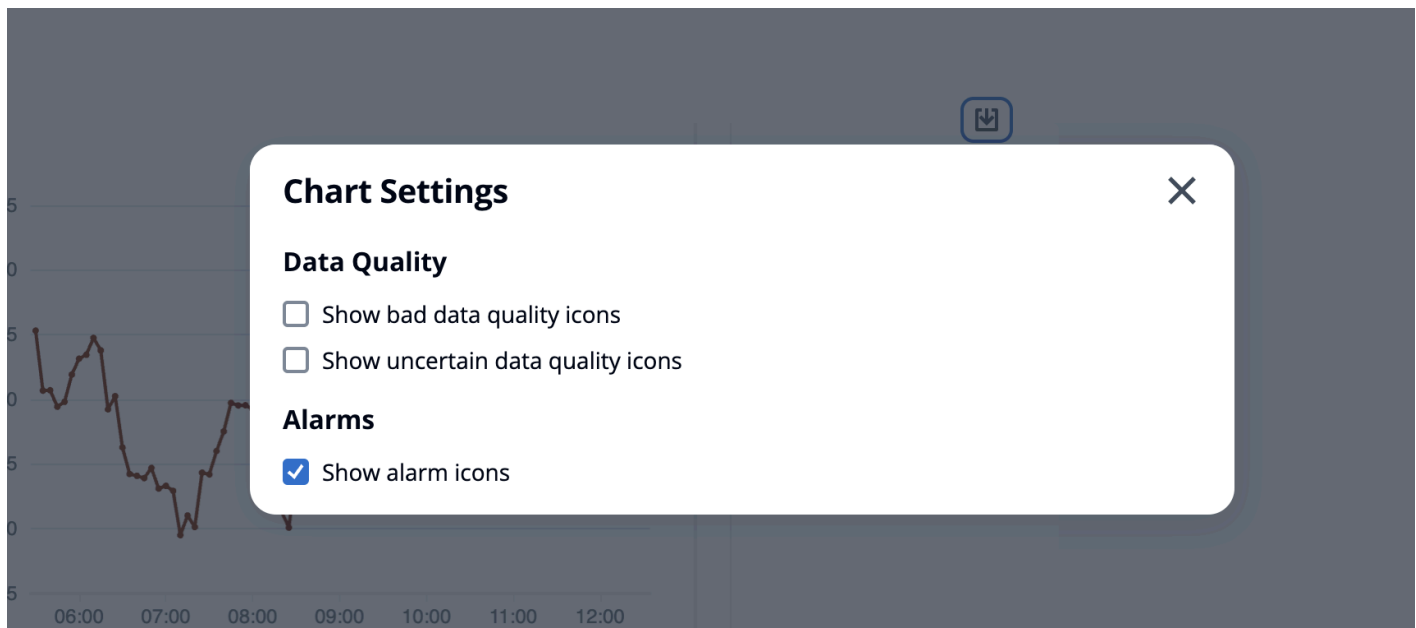
Für alle Widgets:

- Die Einstellungen für die Eigenschaften eines Datenstroms hängen davon ab, welcher Eigenschaftstyp einem Widget hinzugefügt wird. Datenstream-Eigenschaften bieten volle Unterstützung für Eigenschaftseinstellungen, wohingegen die Alarm-Eigenschaften derzeit keine Konfiguration von Eigenschaftseinstellungen zulassen.
- Wenn Sie einen Alarm-Datenstream hinzufügen, wird der zugehörige Datenstrom für Eingabeeigenschaften ebenfalls dem Diagramm hinzugefügt. Wenn Sie den Alarm-Datenstrom entfernen, wird auch seine Eingabeeigenschaft entfernt.

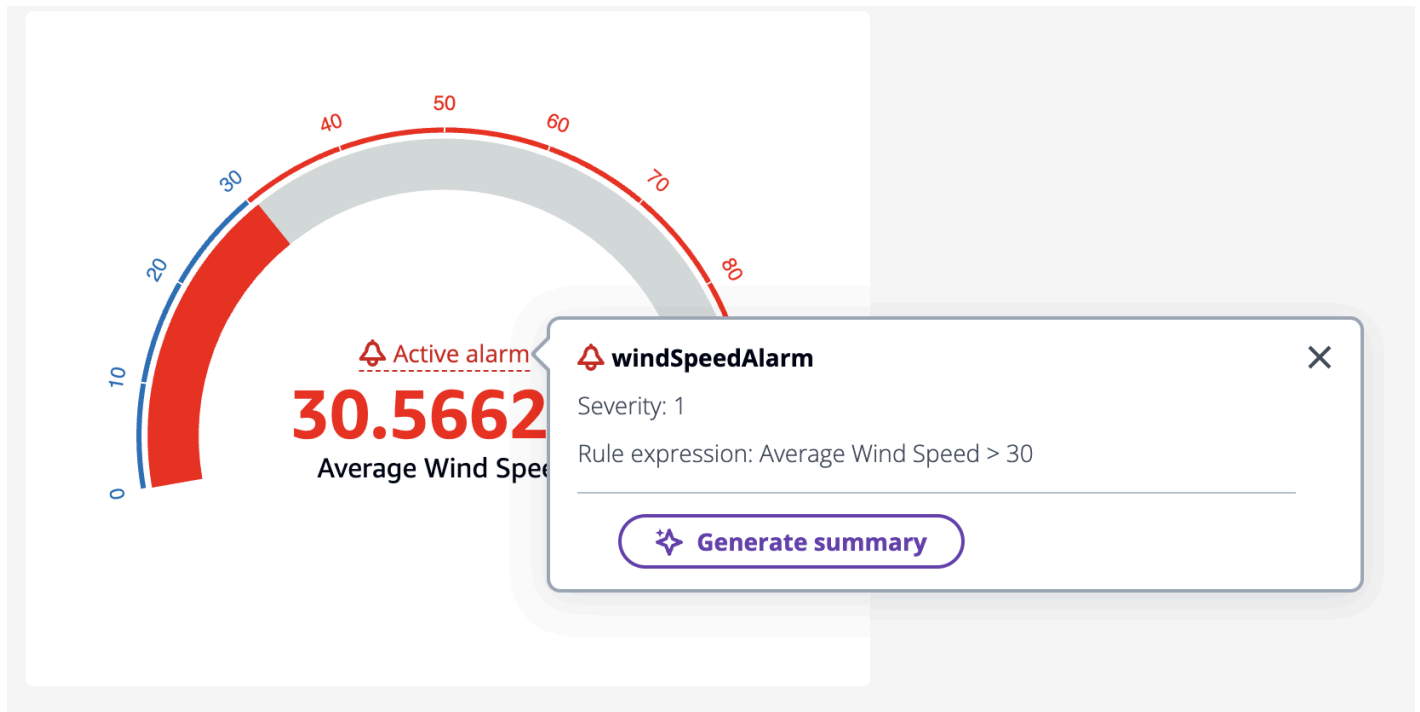
- Um den Datenstrom für Eingabeeigenschaften eines Alarms individuell zu steuern, müssen Sie beide separat hinzufügen.

Die folgenden Beispiele zeigen, wie einige Widgets Alarme verwenden.

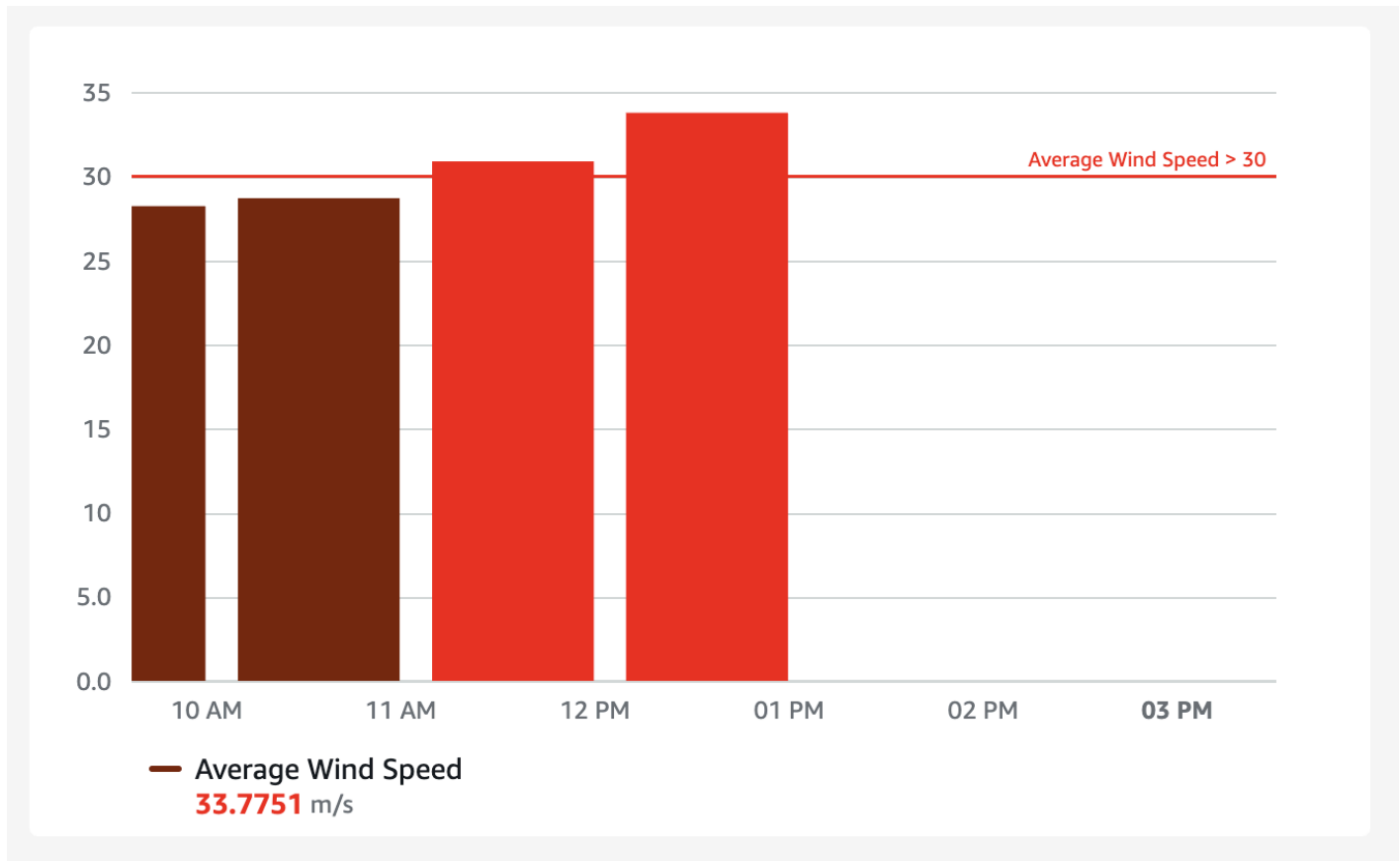
- Liniendiagramm
 - Der Alarm und der Datenstrom seiner Eingabeeigenschaft werden dem Diagramm hinzugefügt.
 - Sie können den Alarmstatus in der Diagrammlegende und als Symbole sehen, die über dem Datenstrom schweben, wenn der Alarm seinen Status ändert.
 - Sie können die Alarmsymbole in den Diagrammeinstellungen ausschalten.



- KPI und Gauge
 - Der Alarm und der Datenstrom seiner Eingabeeigenschaft werden dem ausgewählten Widget hinzugefügt.
 - Der Alarmschwellenwert wird dem Widget hinzugefügt, dessen Farbe sich je nach Konfiguration ändert.
 - Sie können den Alarmstatus im Widget auswählen, sich die Alarmdetails ansehen und auf Zusammenfassung generieren klicken, um das aufzurufen und eine Alarmzusammenfassung AWS IoT SiteWise zu erhalten.



- Tabelle
 - Der Alarm und seine Eingabeeigenschaft werden der Tabelle als Zeile hinzugefügt.
- Balkendiagramm
 - Der Alarm wird dem Diagramm als Schwellenwert hinzugefügt, wodurch die Farbe jedes Datenstroms geändert wird, der den Schwellenwert überschreitet.
 - Sie können alle zugehörigen Datenströme separat hinzufügen.
 - Sie können über das Widget nicht mit dem AWS IoT SiteWise Assistenten interagieren.



- Zeitleiste des Status
 - Der Alarm wird der Timeline als Schwellenwert hinzugefügt.
 - Das Hinzufügen des Alarmstatus und seiner eingegebenen Eigenschaftsdaten zur Timeline ist in Arbeit.
 - Sie können über das Widget nicht mit dem AWS IoT SiteWise Assistenten interagieren.

AWS IoT SiteWise Verwendung des Assistenten in Widgets

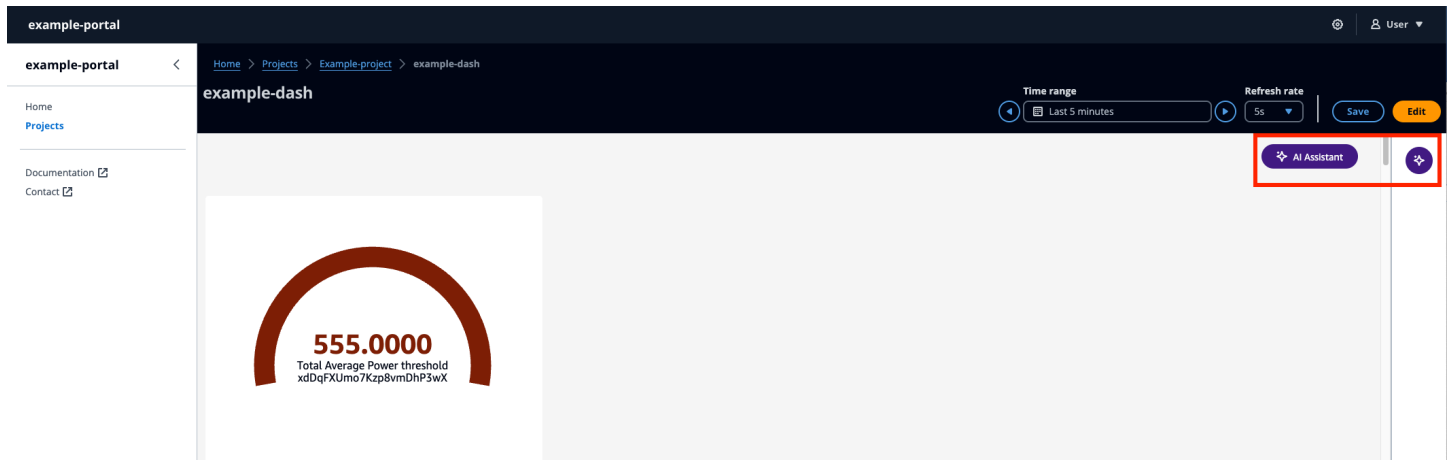
Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Der AWS IoT SiteWise Assistent ist ein generativer KI-gestützter Assistent. Er ermöglicht es Benutzern wie Werksleitern, Qualitätsingenieuren und Wartungstechnikern, direkt aus ihren

Betriebs- und Unternehmensdaten Erkenntnisse zu gewinnen, Probleme zu lösen und Maßnahmen zu ergreifen. Der AWS IoT SiteWise Assistent konsolidiert Informationen aus AWS IoT Daten, Anlagenmodellen, Handbüchern und Dokumentationen in verständlichen Zusammenfassungen kritischer Ereignisse. Darüber hinaus ermöglicht er interaktive, vertiefende Frage-und-Antwort-Sitzungen für einfache Diagnosen, Ursachenforschung und gezielte Empfehlungen.

Die AWS IoT SiteWise Assistent-Taste befindet sich in der oberen rechten Ecke des Dashboards. Klicken Sie darauf, um den Assistenten zu aktivieren. Kann nur mit dem Vorschaumodus des Dashboards verwendet werden.



Verwenden Sie den AWS IoT SiteWise Assistenten in den folgenden Szenarien:

Themen

- [Anwendungsfall — Alarm-Zusammenfassungen](#)
- [Anwendungsfall — Situationszusammenfassungen](#)
- [Anwendungsfall — Tiefgründige Zusammenfassungen](#)

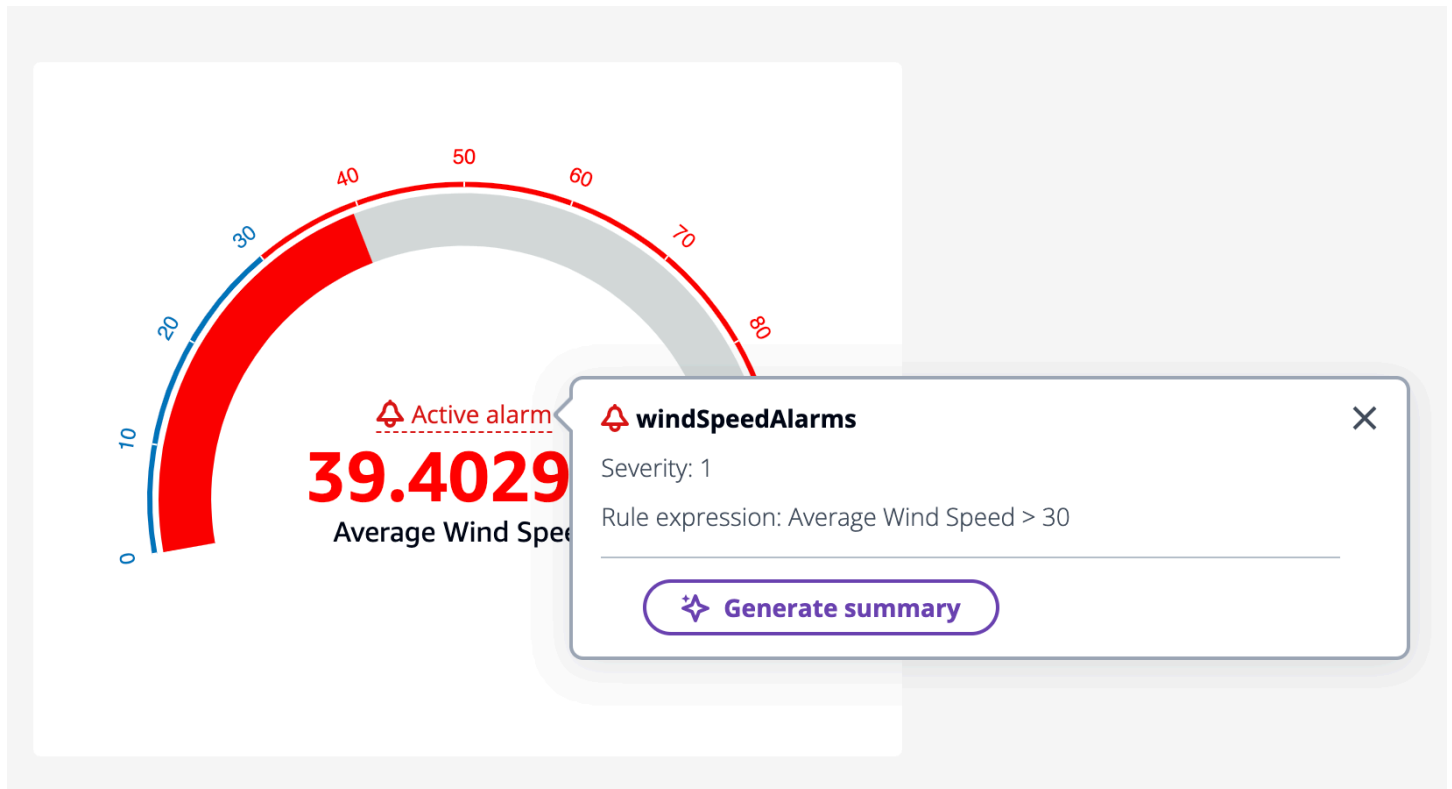
Anwendungsfall — Alarm-Zusammenfassungen

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Fasst den aktuellen Alarm für ein ausgewähltes Panel auf dem Dashboard zusammen. Alarme werden von den Widgets „Linie“, „KPI“, „Anzeige“ und „Tabelle“ unterstützt. Wählen Sie ein Widget mit einem Alarm und fassen Sie es zusammen.

- Wählen Sie Aktiver Alarm im Widget aus.
- Der Ausdruck Schweregrad und Regel wird für den Alarm angezeigt.
- Wählen Sie Zusammenfassung generieren, um eine Zusammenfassung zu generieren.



Anwendungsfall — Situationszusammenfassungen

Note

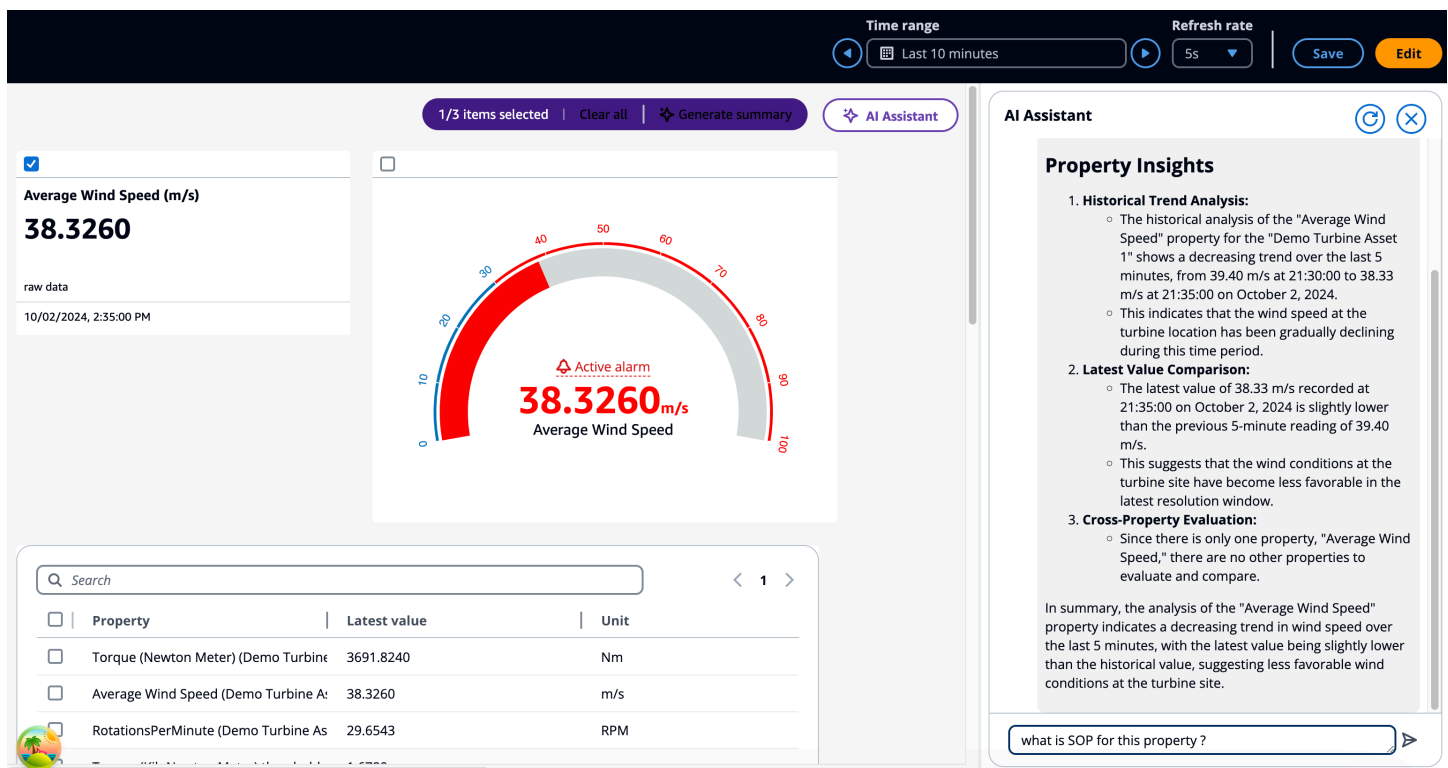
Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Wählen Sie bis zu drei Widgets für die Zusammenfassung aus. Sie können eine Kombination aus Widgets und Eigenschaften sein. Wenn mehr als drei ausgewählt sind, gibt der Assistent einen Fehler zurück.

Generieren Sie mit AWS IoT SiteWise Assistant eine Situationszusammenfassung

- Klicken Sie auf AI Assistant. Es wird ein Menü mit drei Optionen angezeigt.
 - Ausgewählte Elemente — Wählen Sie nur drei aus. Sie können nicht mehr als drei auswählen.
 - Alles löschen — Löscht Ihre Auswahl.
 - Zusammenfassung generieren — Generieren Sie eine Zusammenfassung der ausgewählten Elemente.
- Wählen Sie Zusammenfassung generieren, um eine Zusammenfassung der ausgewählten Elemente zu generieren.

In der Abbildung unten sind ein Widget und eine Zusammenfassung aus dem AWS IoT SiteWise Assistenten ausgewählt.



Anwendungsfall — Tiefgründige Zusammenfassungen

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Dies ist der Anwendungsfall, bei dem der Benutzer tief in die Materie eintauchen und auf Handbücher und Dokumentation SOPs (Standardverfahren) zugreifen und die nächsten Schritte abwägen kann. Für das Beispiel im vorherigen Abschnitt: Wenn der Benutzer mehr über die SOP für diese Eigenschaft erfahren möchte, fragen Sie den Assistenten nach der SOP für diese Eigenschaft. Dadurch werden dem Benutzer detaillierte Informationen zur SOP angezeigt.

Das folgende Beispiel zeigt die Antwort auf "Gibt es eine SOP für den Alarm windSpeedAlarm? "

The screenshot displays the AWS IoT SiteWise dashboard. At the top, there are controls for 'Time range' (set to 'Last 5 minutes') and 'Refresh rate' (set to '5s'), along with 'Save' and 'Edit' buttons. Below this, a toolbar shows '2/3 items selected', 'Clear all', 'Generate summary', and an 'AI Assistant' button. The main content area on the left features a gauge for 'Wind Direction' with a value of '36.9589 Degrees'. On the right, the 'AI Assistant' panel is open, displaying the question 'is there any SOP for the alarm windSpeedAlarm?'. The assistant's response provides a detailed 'SOP for "windSpeedAlarm"', including the title 'Standard Operational Procedures for Root Cause Analysis in Alarms when Wind Farm Engines have Wind Speed Higher than the Threshold:' and three numbered steps: 1. Alarm Identification, 2. Data Collection, and 3. Root Cause Analysis, each with specific instructions. At the bottom of the AI Assistant panel is a text input field with the placeholder 'Ask me anything about your IoT data' and a send button.

Beispielfragen, die Sie dem Assistenten stellen sollten AWS IoT SiteWise

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Note

- Der AWS IoT SiteWise Assistent muss einen Datensatz mit einem [Amazon Kendra Kendra-Index](#) verwenden, um Wissen und Beratung auf Unternehmensebene zu erhalten. Wenn Sie keinen Amazon Kendra Kendra-Index haben, finden Sie Informationen zur [Erstellung eines Indexes unter Index](#) erstellen. Das Hinzufügen eines [Datensatzes](#) verbessert die Qualität der Antwort des Assistenten. Weitere Informationen finden Sie unter [Datensatz erstellen](#).
- Einige Fragen erfordern eine AWS IoT TwinMaker Integration. Einzelheiten finden Sie unter [AWS IoT SiteWise Integrieren AWS IoT TwinMaker und](#).

Einige weitere Fragen, die Sie dem Assistenten stellen sollten, nachdem Sie im Dashboard eine Alarm-Zusammenfassung erhalten haben, als Teil derselben Konversation.

- Die Details des Assets aus der obigen Zusammenfassung anzeigen?
- Was ist der hierarchische Pfad vom Stamm zum genannten Asset?
- Was sind die abhängigen abgeleiteten Vermögenswerte des genannten Vermögenswerts?
- Bei welchen Anlagen handelt es sich um abhängige Anlagen der genannten Anlage, für die Alarmer aktiv sind?
- Finden Sie alle Anlagen, für die Alarmer aktiv sind.

Einige weitere Fragen, die Sie dem Assistenten stellen sollten, nachdem Sie im Dashboard eine Zusammenfassung der Eigenschaften angezeigt haben, als Teil derselben Konversation.

- Führen Sie dieselbe Analyse für die letzten 24 Stunden durch.
- Hier finden Sie die Dokumentation zu den oben genannten Eigenschaften.

- Geben Sie die Einzelheiten der Objekt-ID 1da67d28-14f8-4f71-a06a-386f0425a21d/Anlagenname Demo Turbine Asset 1 an.

AWS IoT SiteWise Rufen Sie den Assistenten über die API auf.

- Generieren Sie eine Alarmzusammenfassung für den Alarmnamen windSpeedAlarmin der Asset-ID. d591e153-e5cf-4206-96bb-ce3c119d9d2d
- Generieren Sie eine Alarmzusammenfassung für die letzten 12 Stunden/2 Tage/1 Woche für den Alarmnamen windSpeedAlarmin der Asset-ID. d591e153-e5cf-4206-96bb-ce3c119d9d2d
- Generieren Sie eine Zusammenfassung der Eigenschaften für die Objekt-ID in der Objekt-ID ab187fb7-d74b-44d9-bd9b-f2f19a9137cc d591e153-e5cf-4206-96bb-ce3c119d9d2d
- Generieren Sie eine Immobilienübersicht für die letzten 12 Stunden/2 Tage/1 Woche für die Immobilien-ID in der Objekt-ID ab187fb7-d74b-44d9-bd9b-f2f19a9137cc. d591e153-e5cf-4206-96bb-ce3c119d9d2d
- Finden Sie die Anlagen mit dem Vermögensnamen Turbine.
- Geben Sie mir die aktuellen Immobilienwerte der Immobilien-ID 5356168c-3390-456f-802c-9f6e047810d4 in der Asset-ID d591e153-e5cf-4206-96bb-ce3c119d9d2d, 3cbb084e-1ded-4b08-9f21-1b47b2fb86fd.
- Was ist die Beziehung zwischen Asset-ID d591e153-e5cf-4206-96bb-ce3c119d9d2d und Asset-ID 3cbb084e-1ded-4b08-9f21-1b47b2fb86fd.
- Hier finden Sie die Dokumentation zur Behebung des Problems mit niedrigen Drehzahlen bei Windkraftanlagen.
- Generieren Sie eine Eigenschaftsübersicht für den Eigenschaftsalias WindSpeed.
- Was sind laut meiner Wissensdatenbank die Prüfungen vor der Operation?

Daten abfragen von AWS IoT SiteWise

Sie können die AWS IoT SiteWise API-Operationen verwenden, um die aktuellen Werte, historischen Werte und Aggregate Ihrer Asset-Eigenschaften über bestimmte Zeitintervalle abzufragen. AWS IoT SiteWise bietet mehrere Abfrageschnittstellen, um unterschiedlichen Integrationsanforderungen gerecht zu werden:

- Direkte API-Operationen — Einfache, gezielte API-Aufrufe für spezifische Datenabrufanforderungen
- SQL-like Abfragesprache — Leistungsstarke, flexible Abfragen für komplexe Datenanalysen
- ODBC-Treiber — Integration mit Business Intelligence-Tools und -Anwendungen

Verwenden Sie diese Abfragefunktionen, um:

- Gewinnen Sie in Echtzeit Einblicke in Betriebsdaten
- Analysieren Sie historische Trends und Muster
- Berechnen Sie Leistungskennzahlen für Ihre Industrieanlagen
- Integrieren Sie IoT-Daten in Unternehmenssysteme und Dashboards
- Erstellen Sie maßgeschneiderte Anwendungen, die industrielle Daten nutzen

Sie können beispielsweise alle Anlagen mit bestimmten Eigenschaftswerten ermitteln, benutzerdefinierte Darstellungen Ihrer Daten erstellen oder Softwarelösungen entwickeln, die sich in die in Ihren AWS IoT SiteWise Anlagen gespeicherten Industriedaten integrieren lassen. Sie können Ihre Komponentendaten auch live in AWS IoT SiteWise Monitor untersuchen. Informationen zur Konfiguration von SiteWise Monitor finden Sie unter [Überwachen Sie Daten mit AWS IoT SiteWise Monitor](#).

Die in diesem Abschnitt beschriebenen Operationen geben Eigenschaftswertobjekte zurück, die Zeitstempel-, Qualitäts- und Wertstrukturen (TQV) enthalten:

- `timestamp` enthält die aktuelle Unix-Epoche in Sekunden mit Nanosekunden-Offset.
- `quality` enthält eine der folgenden Zeichenfolgen zur Angabe der Qualität des Datenpunkts:
 - `GOOD`— Die Daten sind von keinen Problemen betroffen.
 - `BAD`— Die Daten sind von einem Problem wie einem Sensorausfall betroffen.
 - `UNCERTAIN`— Die Daten sind von einem Problem wie einer Sensorungenauigkeit betroffen.

- value enthält abhängig vom Typ der Eigenschaft eines der folgenden Felder:
 - booleanValue
 - doubleValue
 - integerValue
 - stringValue
 - nullValue

Themen

- [Fragen Sie aktuelle Immobilienwerte ab inAWS IoT SiteWise](#)
- [Fragen Sie historische Werte von Vermögenswerten ab inAWS IoT SiteWise](#)
- [Abfragen von Asset-Eigenschaftenaggregaten inAWS IoT SiteWise](#)
- [AWS IoT SiteWiseAbfragesprache](#)
- [Optimierung der Abfrage](#)
- [ODBC](#)

Fragen Sie aktuelle Immobilienwerte ab inAWS IoT SiteWise

Dieses Tutorial zeigt zwei Möglichkeiten, den aktuellen Wert einer Anlageneigenschaft zu ermitteln. Sie können die AWS IoT SiteWise Konsole oder die API in der AWS Command Line Interface (AWS CLI) verwenden.

Themen

- [Fragen Sie den aktuellen Wert einer Asset-Eigenschaft ab \(Konsole\)](#)
- [Fragen Sie den aktuellen Wert einer Anlageeigenschaft ab \(AWS CLI\)](#)

Fragen Sie den aktuellen Wert einer Asset-Eigenschaft ab (Konsole)

Sie können die AWS IoT SiteWise Konsole verwenden, um den aktuellen Wert einer Anlageneigenschaft anzuzeigen.

So erhalten Sie den aktuellen Wert einer Komponenteneigenschaft (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im linken Navigationsbereich Komponenten aus.

3. Wählen Sie die Komponente mit der abzufragenden Eigenschaft aus.
4. Wählen Sie das Pfeilsymbol, um eine Asset-Hierarchie zu erweitern und Ihr Asset zu finden.
5. Wählen Sie die Registerkarte für den Eigenschaftstyp aus. Wählen Sie beispielsweise Messungen, um den aktuellen Wert einer Messungseigenschaft anzuzeigen.
6. Suchen Sie nach der anzuzeigenden Eigenschaft. Der aktuelle Wert wird in der Spalte Aktueller Wert angezeigt.

Fragen Sie den aktuellen Wert einer Anlageeigenschaft ab (AWS CLI)

Sie können das AWS Command Line Interface (AWS CLI) verwenden, um den aktuellen Wert einer Anlageeigenschaft abzufragen.

Verwenden Sie die [GetAssetPropertyValue](#) Operation, um den aktuellen Wert einer Anlageeigenschaft abzufragen.

Um eine Anlageeigenschaft zu identifizieren, geben Sie eine der folgenden Optionen an:

- Das `assetId` Ende `propertyId` der Anlageeigenschaft, an die Daten gesendet werden.
- The `propertyAlias`, bei dem es sich um einen Datenstream-Alias handelt (z. B. `/company/windfarm/3/turbine/7/temperature`). Um diese Option verwenden zu können, müssen Sie zuerst den Alias der Komponenteneigenschaft festlegen. Informationen zum Festlegen von Eigenschaftsaliasnamen finden Sie unter [Datenströme verwalten für AWS IoT SiteWise](#).

Um den aktuellen Wert einer Anlageeigenschaft zu ermitteln (AWS CLI)

- Führen Sie den folgenden Befehl aus, um den aktuellen Wert der Komponenteneigenschaft abzurufen. `asset-id` Ersetzen Sie ihn durch die ID des Vermögenswerts und `property-id` durch die ID der Immobilie.

```
aws iotsitewise get-asset-property-value \  
  --asset-id asset-id \  
  --property-id property-id
```

Die Operation gibt eine Antwort mit der aktuellen TQV der Eigenschaft im folgenden Format zurück.

```
{
```

```
"propertyValue": {
  "value": {
    "booleanValue": Boolean,
    "doubleValue": Number,
    "integerValue": Number,
    "stringValue": "String",
    "nullValue": {
      "valueType": "String"
    }
  },
  "timestamp": {
    "timeInSeconds": Number,
    "offsetInNanos": Number
  },
  "quality": "String"
}
```

Fragen Sie historische Werte von Vermögenswerten ab in AWS IoT SiteWise

Sie können den AWS IoT SiteWise [GetAssetPropertyValueHistory](#) API-Vorgang verwenden, um die historischen Werte einer Anlageneigenschaft abzufragen.

Um eine Anlageneigenschaft zu identifizieren, geben Sie eine der folgenden Optionen an:

- Das `assetId` Ende `propertyId` der Anlageneigenschaft, an die Daten gesendet werden.
- The `propertyAlias`, bei dem es sich um einen Datenstream-Alias handelt (z. B. `/company/windfarm/3/turbine/7/temperature`). Um diese Option verwenden zu können, müssen Sie zuerst den Alias der Komponenteneigenschaft festlegen. Informationen zum Festlegen von Eigenschaftsaliasnamen finden Sie unter [Datenströme verwalten für AWS IoT SiteWise](#).

Übergeben Sie die folgenden Parameter, um Ihre Ergebnisse zu verfeinern:

- `startDate`— Der ausschließliche Anfang des Bereichs, aus dem historische Daten abgefragt werden sollen, ausgedrückt in Sekunden in der Unix-Epoche.
- `endDate`— Das inklusive Ende des Bereichs, aus dem historische Daten abgefragt werden sollen, ausgedrückt in Sekunden in der Unix-Epochezeit.

- **maxResults**— Die maximale Anzahl von Ergebnissen, die in einer Anfrage zurückgegeben werden sollen. Standardmäßig werden 20 Ergebnisse verwendet.
- **nextToken**— Ein Paginierungstoken, das von einem vorherigen Aufruf dieser Operation zurückgegeben wurde.
- **timeOrdering**— Die Reihenfolge, die auf die zurückgegebenen Werte angewendet werden soll: ASCENDING oder DESCENDING.
- **qualities**— Die Qualität, nach der Ergebnisse gefiltert werden sollen nach: GOODBAD,, oder UNCERTAIN.

Um die Werthistorie einer Anlageeigenschaft abzufragen (AWS CLI)

1. Führen Sie den folgenden Befehl aus, um den Wertverlauf für die Komponenteneigenschaft abzurufen. Dieser Befehl fragt den Verlauf der Eigenschaft über ein bestimmtes 10-Minuten-Intervall ab. *asset-id* Ersetzen Sie durch die ID der Anlage und *property-id* durch die ID der Immobilie. Ersetzen Sie die Datumsparameter durch das abzufragende Intervall.

```
aws iotsitewise get-asset-property-value-history \
  --asset-id asset-id \
  --property-id property-id \
  --start-date 1575216000 \
  --end-date 1575216600
```

Die Operation gibt eine Antwort zurück, die die historischen TQVs der Immobilie im folgenden Format enthält:

```
{
  "assetPropertyValueHistory": [
    {
      "value": {
        "booleanValue": Boolean,
        "doubleValue": Number,
        "integerValue": Number,
        "stringValue": "String",
        "nullValue": {
          "valueType": "String"
        }
      },
      "timestamp": {
        "timeInSeconds": Number,
```

```
        "offsetInNanos": Number
      },
      "quality": "String"
    }
  ],
  "nextToken": "String"
}
```

2. Wenn mehr Werteinträge vorhanden sind, können Sie das Paginierungstoken aus dem `nextToken` Feld an einen nachfolgenden Aufruf der [GetAssetPropertyValueHistory](#) Operation übergeben.

Abfragen von Asset-Eigenschaftenaggregaten in AWS IoT SiteWise

AWS IoT SiteWise berechnet automatisch aggregierte Immobilienwerte, bei denen es sich um eine Reihe von Basiskennzahlen handelt, die über mehrere Zeitintervalle berechnet werden. AWS IoT SiteWise berechnet jede Minute, Stunde und Tag die folgenden Aggregate für Ihre Anlageeigenschaften:

- Durchschnitt — Der Durchschnitt (Mittelwert) der Werte einer Immobilie über ein Zeitintervall.
- Anzahl — Die Anzahl der Datenpunkte für eine Eigenschaft über ein Zeitintervall.
- Maximum — Das Maximum der Werte einer Eigenschaft über ein Zeitintervall.
- Minimum — Das Minimum der Werte einer Eigenschaft über ein Zeitintervall.
- Standardabweichung — Die Standardabweichung der Werte einer Eigenschaft über ein Zeitintervall.
- Summe — Die Summe der Werte einer Eigenschaft über ein Zeitintervall.

Für nicht numerische Eigenschaften, wie Zeichenketten und Boolesche Werte, wird nur das Aggregat für die AWS IoT SiteWise Anzahl berechnet.

Sie können für Ihre Komponentendaten auch benutzerdefinierte Metriken berechnen. Mit metrischen Eigenschaften definieren Sie Aggregationen, die für Ihren Vorgang spezifisch sind. Metrische Eigenschaften bieten zusätzliche Aggregationsfunktionen und Zeitintervalle, die für die API nicht im Voraus berechnet wurden. AWS IoT SiteWise Weitere Informationen finden Sie unter [Aggregieren Sie Daten aus Immobilien und anderen Vermögenswerten \(Metriken\)](#).

Themen

- [Aggregate für eine Anlageneigenschaft \(API\)](#)
- [Aggregate für eine Anlageeigenschaft \(AWS CLI\)](#)

Aggregate für eine Anlageneigenschaft (API)

Verwenden Sie die AWS IoT SiteWise API, um Aggregate für eine Anlageneigenschaft abzurufen.

Verwenden Sie die [GetAssetPropertyAggregates](#) Operation, um Aggregate einer Anlageneigenschaft abzufragen.

Um eine Anlageneigenschaft zu identifizieren, geben Sie eine der folgenden Optionen an:

- Das `assetId` Ende `propertyId` der Anlageneigenschaft, an die Daten gesendet werden.
- `ThepropertyAlias`, bei dem es sich um einen Datenstream-Alias handelt (z. B. `/company/windfarm/3/turbine/7/temperature`). Um diese Option verwenden zu können, müssen Sie zuerst den Alias der Komponenteneigenschaft festlegen. Informationen zum Festlegen von Eigenschaftsaliasnamen finden Sie unter [Datenströme verwalten für AWS IoT SiteWise](#).

Sie müssen die folgenden erforderlichen Parameter übergeben:

- `aggregateTypes`— Die Liste der abzurufenden Aggregate. Sie können AVERAGE, COUNT, MAXIMUM, MINIMUM, STANDARD_DEVIATION oder SUM angeben.
- `resolution`— Das Zeitintervall, für das die Metrik abgerufen werden soll: 1m (1 Minute), 15m (15 Minuten), 1h (1 Stunde) oder 1d (1 Tag).
- `startDate`— Der ausschließliche Anfang des Bereichs, aus dem historische Daten abgefragt werden sollen, ausgedrückt in Sekunden in Unix-Epochenzeit.
- `endDate`— Das inklusive Ende des Bereichs, aus dem historische Daten abgefragt werden sollen, ausgedrückt in Sekunden in der Unix-Epochenzeit.

Sie können auch einen der folgenden Parameter übergeben, um Ihre Ergebnisse zu verfeinern:

- `maxResults`— Die maximale Anzahl von Ergebnissen, die in einer Anfrage zurückgegeben werden sollen. Standardmäßig werden 20 Ergebnisse verwendet.
- `nextToken`— Ein Paginierungstoken, das von einem vorherigen Aufruf dieser Operation zurückgegeben wurde.

- `timeOrdering`— Die Reihenfolge, die auf die zurückgegebenen Werte angewendet werden soll: `ASCENDING` oder `DESCENDING`.
- `qualities`— Die Qualität, nach der Ergebnisse gefiltert werden sollen nach: `GOODBAD`,, oder `UNCERTAIN`.

Note

Die [GetAssetPropertyAggregates](#) Operation gibt ein TQV zurück, dessen Format sich von den anderen in diesem Abschnitt beschriebenen Operationen unterscheidet. Die `value`-Struktur enthält ein Feld für jeden der `aggregateTypes` in der Anforderung. Der `timestamp` enthält die Zeit in Sekunden in Unix-Epoche-Zeit, zu der die Aggregation stattfand.

Aggregate für eine Anlageeigenschaft (AWS CLI)

Um Aggregate für eine Anlageeigenschaft abzufragen (AWS CLI)

1. Führen Sie den folgenden Befehl aus, um Aggregate für die Komponenteneigenschaft abzurufen. Dieser Befehl fragt den Durchschnitt und die Summe mit einer Auflösung von 1 Stunde für ein bestimmtes Intervall von 1 Stunde ab. `asset-id` Ersetzen Sie durch die ID der Anlage und `property-id` durch die ID der Immobilie. Ersetzen Sie die Parameter durch die Aggregate und das abzufragende Intervall.

```
aws iotsitewise get-asset-property-aggregates \
  --asset-id asset-id \
  --property-id property-id \
  --start-date 1575216000 \
  --end-date 1575219600 \
  --aggregate-types AVERAGE SUM \
  --resolution 1h
```

Die Operation gibt eine Antwort mit den historischen TQVs der Eigenschaft im folgenden Format zurück. Die Antwort enthält nur die angeforderten Aggregate.

```
{
  "aggregatedValues": [
    {
      "timestamp": Number,
```



```
"quality": "String",
"value": {
  "average": Number,
  "count": Number,
  "maximum": Number,
  "minimum": Number,
  "standardDeviation": Number,
  "sum": Number
}
},
"nextToken": "String"
}
```

2. Wenn mehr Werteinträge vorhanden sind, können Sie das Paginierungstoken aus dem `nextToken` Feld an einen nachfolgenden Aufruf der [GetAssetPropertyAggregates](#) Operation übergeben.

Note

Wenn Ihr Abfragebereich den `null` Wert TQVs enthält, finden Sie weitere Informationen unter [AssetPropertyValueAPI](#). Alle Statistiken außer `count` führen zu einer `null` Antwort, ähnlich wie Statistiken für String-TQVs. Wenn Ihr Abfragebereich TQVs vom Typ `Double` enthält `Double.NaN`, führen alle Berechnungen außer `count` zu einem `Double.NaN`.

AWS IoT SiteWiseAbfragesprache

Mit dem [ExecuteQuery](#) API-Vorgang für den AWS IoT SiteWise Datenabruf können Sie Informationen zu deklarativen Strukturdefinitionen und den damit verbundenen Zeitreihendaten aus folgenden Quellen abrufen:

- -Modelle von
- Vermögenswerte
- Messungen
- Metriken
- Transformationen
- Aggregate

Dies kann mit SQL-ähnlichen Abfrageanweisungen in einer einzigen API-Anfrage erfolgen.

 Note

Diese Funktion ist in allen Regionen verfügbar, in denen sie verfügbar AWS IoT SiteWise ist AWS GovCloud (US-West), mit Ausnahme von Kanada (Zentral), China (Peking) und USA Ost (Ohio).

Themen

- [Sprachreferenz abfragen für AWS IoT SiteWise](#)

Sprachreferenz abfragen für AWS IoT SiteWise

AWS IoT SiteWise unterstützt eine umfangreiche Abfragesprache für die Arbeit mit Ihren Daten. Die verfügbaren Datentypen, Operatoren, Funktionen und Konstrukte werden in den folgenden Themen beschrieben.

Informationen [Beispielabfragen](#) zum Schreiben von Abfragen mit der AWS IoT SiteWise Abfragesprache finden Sie unter.

Themen

- [Referenzansichten abfragen](#)
- [Unterstützte Datentypen](#)
- [Unterstützte Klauseln](#)
- [Logische Operatoren](#)
- [Vergleichsoperatoren](#)
- [SQL-Funktionen](#)
- [Beispielabfragen](#)

Referenzansichten abfragen

In diesem Abschnitt finden Sie Informationen zum besseren Verständnis der Ansichten in AWS IoT SiteWise, z. B. Prozessmetadaten und Telemetriedaten.

Die folgenden Tabellen enthalten die Namen und Beschreibungen der Ansichten:

Datenmodell

Namen anzeigen	Ansichtsbeschreibung
Komponente	Enthält Informationen zur Anlage- und Modellableitung.
asset_property	Enthält Informationen über die Struktur der Anlageeigenschaft.
raw_time_series	Enthält die historischen Daten der Zeitreihe.
latest_value_time_series	Enthält den neuesten Wert der Zeitreihe.
precomputed_aggregates	Enthält die automatisch berechneten aggregierten Eigenschaftswerte von Vermögenswerten. Es handelt sich um eine Reihe von Basiskennzahlen, die über mehrere Zeitintervalle berechnet wurden.

In den folgenden Ansichten sind die Spaltennamen und Datentypen der einzelnen Ansichten aufgeführt.

View:asset

Name der Spalte	datatype
asset_id	Zeichenfolge
Name der Anlage	Zeichenfolge
Beschreibung der Anlage	Zeichenfolge
Asset_Modell-ID	Zeichenfolge
übergeordnete_Asset_ID	Zeichenfolge
externe_ID der Anlage	Zeichenfolge
externe_ID des Asset_Model-Modells	Zeichenfolge

Name der Spalte	datatype
hierarchische_id	Zeichenfolge

View:asset_Eigenschaft

Name der Spalte	datatype			
asset_id	Zeichenfolge			
Eigenschafts-ID	Zeichenfolge			
Eigenschaftsname	Zeichenfolge			
Eigenschaftsalias	Zeichenfolge			
Eigenschafts externe ID	Zeichenfolge			
asset_composite_model_id	Zeichenfolge			
Eigenschaftstyp	Zeichenfolge			
Eigenschaft_Datentyp	Zeichenfolge			
int_attribute_wert	Ganzzahl			
doppelter_Attributwert	double			
boolescher_Attributwert	boolesch			
Zeichenkettenattribute_wert	Zeichenfolge			

View:raw_Zeitreihe

Name der Spalte	datatype
asset_id	Zeichenfolge
Eigenschafts-ID	Zeichenfolge
Eigenschaftsalias	Zeichenfolge
event_timestamp	Zeitstempel
quality	Zeichenfolge
boolescher_Wert	boolesch
int_wert	Ganzzahl
doppelter_Wert	double
Zeichenfolge_Wert	Zeichenfolge

View:latest_Wert_Zeitreihe

Name der Spalte	datatype
asset_id	Zeichenfolge
Eigenschafts-ID	Zeichenfolge
Eigenschaftsalias	Zeichenfolge
event_timestamp	Zeitstempel
quality	Zeichenfolge
boolescher_Wert	boolesch
int_wert	Ganzzahl
doppelter_Wert	double

Name der Spalte	datatype
Zeichenfolge_Wert	Zeichenfolge

View:precomputed_Aggregate

Name der Spalte	datatype
asset_id	Zeichenfolge
Eigenschafts-ID	Zeichenfolge
Eigenschaftsalias	Zeichenfolge
event_timestamp	Zeitstempel
quality	Zeichenfolge
Auflösung	Zeichenfolge
Summe_Wert	double
Anzahl_Wert	Ganzzahl
Durchschnittswert	double
maximal_wert	double
Minimalwert	double
Standardwert	double

Unterstützte Datentypen

AWS IoT SiteWiseDie Abfragesprache unterstützt die folgenden Datentypen.

Skalarer Wert

Datentyp	Beschreibung
STRING	Eine Zeichenfolge mit einer maximalen Länge von 1024 Byte.
INTEGER	Eine 32-Bit-Ganzzahl mit Vorzeichen und einem Bereich von -2,147,483,648 to 2,147,483,647 .
DOUBLE	Eine Fließkommazahl mit einem Bereich von -10^{100} to 10^{100} oder Nan mit IEEE 754 doppelter Genauigkeit.
BOOLEAN	true oder false.
TIMESTAMP	ISO-8601 konforme Zeitstempel: 'yyyy-MM-dd HH:mm:ss[.SSS] ' - TIMESTAMP 'yyyy-MM-dd[\s T]HH:mm:ss[.SSS][+HH:mm 'Z'] ' 'yyyy-MM-dd'T'HH:mm:ss[.SSS][+HH:mm 'Z'] ' 'yyyy-MM-dd'T'HH:mm:ss+[hh:mm] '

Note

`Null`: Ein boolescher Wert, der `true` auf einen Mangel an definierten Daten hinweist.

Example

TIMESTAMP Beispiele für Werte:

```
TIMESTAMP '2025-12-21 23:59:59.999Z'
TIMESTAMP '2025-12-21 23:59:59+23:59'
'2025-12-21 23:59:59'
```

```
'2025-12-21T23:59:59.123+11:11'
```

Note

Die Daten mit doppelter Genauigkeit sind nicht exakt. Einige Werte werden nicht exakt konvertiert und stellen aufgrund der begrenzten Genauigkeit nicht alle reellen Zahlen dar. Floating-point Die Daten in der Abfrage haben möglicherweise nicht den gleichen Wert, der intern dargestellt wird. Der Wert wird gerundet, wenn die Genauigkeit einer eingegebenen Zahl zu hoch ist.

Unterstützte Klauseln

Die SELECT Anweisung wird verwendet, um Daten aus einer oder mehreren Ansichten abzurufen. AWS IoT SiteWise unterstützt die INNER JOIN Operationen JOIN und.

Ansichten werden mit einer expliziten JOIN Syntax oder mit durch Kommas getrennten Notationen in der Klausel verknüpft. FROM

Example

Eine allgemeine Aussage: SELECT

```
SELECT expression [, ...]
  [ FROM table_name AS alias [, ...] ]
  [ WHERE condition ]
  [ GROUP BY expression [, ...] ]
  [ HAVING condition ]
  [ ORDER BY expression [ ASC | DESC ] [ NULLS FIRST | NULLS LAST ] [, ...] ]
  [ LIMIT expression ]
```

Example

Eine SELECT-Anweisung mit den verschiedenen Klauseln:

```
SELECT
  a.asset_name,
  a.asset_id,
  p.property_type,
  p.property_data_type,
```



```
p.string_attribute_value,  
p.property_name  
FROM asset a, asset_property p  
WHERE a.asset_description LIKE '%description%'  
AND p.property_type IN ('attribute', 'metric')  
OR p.property_id IN (  
    SELECT property_id  
    FROM raw_time_series  
    WHERE event_timestamp BETWEEN TIMESTAMP '2025-01-01 00:00:00' AND TIMESTAMP  
'2025-01-02 00:00:00'  
    GROUP BY asset_id, property_id  
    HAVING COUNT(*) > 100  
)  
GROUP BY p.property_type  
HAVING COUNT(*) > 5  
ORDER BY a.asset_name ASC  
LIMIT 20;
```

Note

Eine implizite Methode JOIN kombiniert zwei oder mehr verschiedene Tabellen, ohne das auf AWS IoT SiteWise dem internen Schema basierende JOIN Schlüsselwort zu verwenden. Dies entspricht der Ausführung von A JOIN für die property_id Felder asset_id und zwischen Metadaten- und Rohdatentabellen. Dieses Muster SiteWise ermöglicht es, beliebige Metadatenfilter in der Abfrage beim Abrufen aus Rohdatentabellen so zu nutzen, dass insgesamt weniger Daten gescannt werden.

Example einer Abfrage:

```
SELECT a.asset_name, p.property_name, r.event_timestamp  
FROM asset a, asset_property p, raw_time_series r  
WHERE a.asset_name='my_asset' AND p.property_name='my_property'
```

Im obigen Beispiel werden nur Daten aus der Asset-Eigenschaft gescannt, die zu den angegebenen Metadatenamen gehören.

Example eines weniger optimierten Äquivalents der obigen Abfrage:

```
SELECT a.asset_name, p.property_name, r.event_timestamp  
FROM asset a  
JOIN asset_property p ON a.asset_id=p.asset_id
```

```
JOIN raw_time_series r ON p.asset_id=r.asset_id AND p.property_id=r.property_id
WHERE a.asset_name='my_asset' AND p.property_name='my_property'
```

Eine Erläuterung der einzelnen Klauseln und ihrer Beschreibung ist unten aufgeführt:

Klausel	Signature	Beschreibung
LIMIT	<pre>LIMIT { count }</pre>	Diese Klausel begrenzt die Ergebnismenge auf die angegebene Anzahl von Zeilen. Sie können LIMIT mit oder ohne ORDER BY OFFSET UND-Klauseln verwenden. LIMIT funktioniert nur mit nicht negativen Ganzzahlen ab [0,2147483647].
ORDER BY	<pre>ORDER BY expression [ASC DESC] [NULLS FIRST NULLS LAST]</pre>	Die ORDER BY Klausel sortiert die Ergebnismenge einer Abfrage.  Note Wenn Sie in der ORDER BY Klausel auf ausgewählte Spalten in einer Aggregation verweisen, verwenden Sie den Ordinalindex der Spalte und nicht den Namen oder Alias. <pre>SELECT AVG(t.double_value)</pre>

Klausel	Signature	Beschreibung
		<pre>FROM latest_value_time_series t GROUP BY t.asset_id ORDER BY 1</pre>
GROUP BY	<pre>GROUP BY expression [, ...]</pre>	Die GROUP BY Klausel identifiziert die Gruppierungsspalten für die Abfrage. Sie wird in Verbindung mit einem Aggregatausdruck verwendet.
HAVING	<pre>HAVING boolean-expression</pre>	Die HAVING Klausel filtert Gruppenzeilen, die durch die GROUP BY-Klausel erstellt wurden.
SUB SELECT	<pre>SELECT column1, column2 FROM table1 WHERE column3 IN (SELECT column4 FROM table2);</pre>	Eine SELECT Anweisung, die in eine andere SELECT Anweisung eingebettet ist.
JOIN	<pre>SELECT column1, column2 FROM table1 JOIN table2 ON table1.column1 = table2.column1;</pre>	
INNER JOIN	<pre>SELECT columns FROM table1 INNER JOIN table2 ON table1.column = table2.column;</pre>	An INNER JOIN gibt alle Zeilen aus beiden Tabellen zurück, die der Join-Bedingung entsprechen.

Klausel	Signature	Beschreibung
UNION	<pre>query { UNION [ALL] } another_query</pre>	Der UNION Operator berechnet die Mengenvereinigung seiner beiden Argumente und entfernt automatisch doppelte Datensätze aus der Ergebnismenge.

Logische Operatoren

AWS IoT SiteWise unterstützt die folgenden logischen Operatoren.

Operator	Signature	Beschreibung
AND	ein AND b	TRUE wenn beide Werte wahr sind
OR	a OR b	TRUE wenn ein Wert wahr ist
NOT	NOT Ausdruck	TRUE wenn ein Ausdruck falsch ist und FALSE wenn ein Ausdruck wahr ist
IN	INx-Ausdruck	TRUE wenn Wert im Ausdruck
BETWEEN	BETWEEN ein AND b	TRUE wenn der Wert zwischen Ober- und Untergrenze liegt und beide Grenzwerte einschließt
LIKE	LIKE Muster	TRUE wenn der Wert im Muster ist LIKE unterstützt Platzhalter. Im Folgenden finden Sie Beispiele:

Operator	Signature	Beschreibung
		• %ersetzt ein oder mehrere Zeichen in einer Zeichenfolge. • _ersetzt ein Zeichen in einer Zeichenfolge. • ESCAPEwird zusammen mit einem Zeichen verwendet, um ein Escape-Zeichen im LIKE-Muster zu bezeichnen.

Beispiele für alle logischen Operatoren:

Funktion	Beispiel
AND	<pre>SELECT a.asset_name FROM asset AS a, latest_value_time_series AS t WHERE t.int_value > 30 AND t.event_timestamp > TIMESTAMP '2025-05-15 00:00:01'</pre>
ODER	<pre>SELECT a.asset_name FROM asset AS a WHERE a.asset_name like 'abc' OR a.asset_name like 'pqr'</pre>
NOT	<pre>SELECT ma.asset_id AS a_id FROM asset AS ma WHERE (ma.asset_id NOT LIKE 'some %patterna%' escape 'a') AND ma.asset_ id='abc'</pre>
IN	<pre>SELECT a.asset_name FROM asset AS a</pre>

Funktion	Beispiel
	<pre>WHERE a.asset_name IN ('abc', 'pqr')</pre>
BETWEEN	<pre>SELECT asset_id, int_value, event_timestamp AS i_v FROM raw_time_series WHERE event_timestamp BETWEEN TIMESTAMP '2025-04-15 00:00:01' and TIMESTAMP '2025-05-15 00:00:01'</pre>
LIKE	• %Muster: <pre>SELECT POWER(rw.int_value, 5) AS raised_value FROM raw_time_series AS rw WHERE rw.asset_id LIKE 'some%pattern%' AND rw.int_value > 30</pre> • _Muster: <pre>SELECT asset_id, property_id FROM asset_property WHERE string_attribute_value LIKE 'Floor_'</pre> • ESCAPEMuster: <pre>SELECT asset_id FROM asset WHERE asset_name LIKE 'MyAsset/_%' ESCAPE '/'</pre>

Vergleichsoperatoren

AWS IoT SiteWise unterstützt die folgenden Vergleichsoperatoren. Alle Vergleichsoperationen sind für integrierte Datentypen verfügbar und werden als boolescher Wert ausgewertet.

Logische Operatoren

Operator	Beschreibung
<	Kleiner als
>	Größer als
<=	Kleiner als oder gleich
>=	Größer als oder gleich
=	Gleich
!=	Ungleich

Wahrheitstabelle für Vergleichsoperationen für nicht numerische Werte

Typ	Geben Sie >= x ein	Geben Sie <= x ein	Geben Sie > x ein	Geben Sie < x ein	Typ = x	Typ! = x
NULL	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE

Manche Prädikate verhalten sich wie Operatoren, haben aber eine spezielle Syntax. Siehe unten:

Vergleichsprädikate

Betreiber	Beschreibung
IS NULL	Testet, ob ein Wert istNULL.
IS NOT NULL	Testet, ob ein Wert dies nicht istNULL.

NaN-Operatoren

NaN, oder 'Keine Zahl', ist ein besonderer Wert in der Gleitkomma-Arithmetik. Hier ist eine Liste von Vergleichen und wie sie funktionieren. NaN

- NaNWerte müssen in einfache Anführungszeichen eingeschlossen werden. Zum Beispiel 'NaN'.

- NaN-Werte werden als einander gleichwertig betrachtet.
- NaN ist größer als andere numerische Werte.
- In Aggregatfunktionen wie AVG(), STDDEV(), und SUM(), falls es irgendwelche Werte gibt NaN, ist das Ergebnis NaN.
- In Aggregatfunktionen wie MAX() MIN() und werden NaN Werte in die Berechnungen einbezogen.

Vergleiche von NaN-Werten

Vergleich	Ergebnis	
'NaN' ≥ x	Wahr	
'NaN' ≤ x	Wahr, wenn x gleich NaN ist, andernfalls Falsch	
'NaN' > x	Falsch, wenn x gleich NaN ist, andernfalls Wahr	
'NaN' < x	Falsch	
'NaN' = x	Wahr, wenn x gleich NaN ist, andernfalls Falsch	
'NaN' != x	Falsch, wenn x gleich NaN ist, andernfalls Wahr	

SQL-Funktionen

Die unterstützten Funktionsgruppen sind:

Themen

- [Skalarfunktionen](#)
- [Aggregationsfunktionen](#)

Skalarfunktionen

Skalarfunktionen verwenden einen oder mehrere Eingabewerte und geben einen einzelnen Ausgabewert zurück. Sie werden in SQL (Structured Query Language) häufig für die Bearbeitung und den Abruf von Daten verwendet und verbessern so die Effizienz von Datenverarbeitungsaufgaben.

Themen

- [Nulldatenfunktionen](#)
- [Zeichenfolgenfunktionen](#)
- [Mathematische Funktionen](#)
- [Funktionen für Datum und Uhrzeit](#)
- [Funktionen zur Typkonvertierung](#)

Nulldatenfunktionen

Null-Datenfunktionen verarbeiten oder manipulieren NULL-Werte, die das Fehlen eines Werts darstellen. Die Funktionen ermöglichen es Ihnen, NULLs durch andere Werte zu ersetzen, zu überprüfen, ob ein Wert NULL ist, oder Operationen auszuführen, die NULLs auf eine bestimmte Weise behandeln.

Funktion	Signature	Beschreibung
COALESCE	COALESCE (Ausdruck1, Ausdruck2,..., AusdruckN)	Wenn alle Ausdrücke Null ergeben, gibt COALESCE Null zurück. Ausdrücke müssen vom gleichen Typ sein.

Example einer COALESCE-Funktion

```
SELECT COALESCE (1.double_value, 100) AS non_double_value FROM latest_value_time_series AS 1 LIMIT 1
```

Zeichenfolgenfunktionen

Zeichenkettenfunktionen sind integrierte Tools zur Manipulation und Verarbeitung von Textdaten. Sie ermöglichen Aufgaben wie Verkettung, Extraktion, Formatierung und Suche innerhalb von

Zeichenketten. Diese Funktionen sind für das Bereinigen, Transformieren und Analysieren von textbasierten Daten in einer Datenbank unerlässlich.

Zeichenfolgenfunktionen

Funktion	Signature	Beschreibung
LENGTH	LÄNGE (Zeichenfolge)	Gibt die Länge der Zeichenfolge zurück.
CONCAT	CONCAT (Zeichenfolge, Zeichenfolge)	Verkettet Argumente in einer Zeichenfolge.
SUBSTR	- SUBSTR (Zeichenfolge, Start) - SUBSTR (Zeichenfolge, Anfang, Länge) - SUBSTR (Zeichenfolge, Regexp)	Gibt einen der folgenden Werte zurück: - Gibt die Teilzeichenfolge der Eingabezeichenfolge zurück, die an der angegebenen Position beginnt und optional die angegebene Länge hat. - Gibt die erste Teilzeichenfolge der Eingabezeichenfolge zurück, die dem angegebenen regulären Ausdruck entspricht. Verwendet eine 1-basierte Indizierung für den Startparameter.
UPPER	UPPER (Zeichenfolge)	Konvertiert die Zeichen in der Eingabezeichenfolge in Großbuchstaben.

Funktion	Signature	Beschreibung
LOWER	LOWER (Zeichenfolge)	Konvertiert die Zeichen in der Eingabezeichenfolge in Kleinbuchstaben.
TRIM	TRIM (Zeichenfolge)	Entfernt alle Leerzeichen am Anfang, Ende oder auf beiden Seiten der Zeichenfolge.
LTRIM	LTRIM (Zeichenfolge)	Entfernt alle Leerzeichen am Anfang einer Zeichenfolge.
RTRIM	RTRIM (Zeichenfolge)	Entfernt alle Leerzeichen am Ende einer Zeichenfolge.
STR_REPLACE	STR_REPLACE (Zeichenfolge, von, bis)	Ersetzt alle Vorkommen der angegebenen Teilzeichenfolge durch eine andere angegebene Teilzeichenfolge.

Beispiele für alle Funktionen:

Funktion	Beispiel
LENGTH	<code>SELECT LENGTH(a.asset_id) AS asset_id_length FROM asset AS a</code>
CONCAT	<code>SELECT CONCAT(p.property_id, p.property_name) FROM asset_property AS p</code>
SUBSTR	<code>SELECT SUBSTR(a.asset_name, 1, 3) AS substr_val FROM asset AS a</code> <code>SELECT SUBSTR(p.property_name, 3) AS substr_val1 FROM asset_property AS p</code>

Funktion	Beispiel
	SELECT SUBSTR(p.property_name, '@[^\.]*') AS substr_val2 FROM asset_property AS p
UPPER	SELECT UPPER(d.string_value) AS up_string FROM raw_time_series AS d
LOWER	SELECT LOWER(d.string_value) AS low_string FROM raw_time_series AS d
TRIM	SELECT TRIM(d.string_value) AS tm_string FROM raw_time_series AS d
LTRIM	SELECT LTRIM(d.string_value) AS ltrim_string FROM raw_time_series AS d
RTRIM	SELECT RTRIM(d.string_value) AS rtrim_string FROM raw_time_series AS d
STR_REPLACE	SELECT STR_REPLACE(d.string_value, 'abc', 'def') AS replaced_string FROM raw_time_series AS d

Verkettungsoperator

Der Verkettungsoperator oder Pipe-Operator `||` verbindet zwei Zeichenketten miteinander. Er stellt eine Alternative zur CONCAT Funktion dar und ist besser lesbar, wenn mehrere Zeichenketten kombiniert werden.

Example des Verkettungsoperators

```
SELECT a.asset_name || ' - ' || p.property_name
       AS full_name
FROM   asset a, asset_property p
```

Mathematische Funktionen

Mathematische Funktionen sind vordefinierte mathematische Operationen, die in SQL-Abfragen verwendet werden, um Berechnungen mit numerischen Daten durchzuführen. Sie bieten Möglichkeiten, Daten zu manipulieren und zu transformieren, ohne sie aus der Datenbank extrahieren und separat verarbeiten zu müssen.

Mathematische Funktionen

Funktion	Signature	Beschreibung
POWER	LEISTUNG (int double, int double)	Gibt den Wert des ersten Arguments potenziert mit dem zweiten Argument zurück.
ROUND	- ROUND (int double, decimal_places_int) - RUND (int double)	Rundet auf die nächste Ganzzahl.
FLOOR	FLOOR (int double)	Gibt die größte Ganzzahl zurück, die nicht größer als der angegebene Wert ist.

Beispiele für alle Funktionen:

Funktion	Beispiel
POWER	- POWER (3, 77) - POWER (2.3, 3.9) - POWER (1.0, 4.2)
ROUND	ROUND (32.12435, 3)

Funktion	Beispiel
FLOOR	FLOOR (21.2)

Funktionen für Datum und Uhrzeit

Datums- und Uhrzeitfunktionen funktionieren mit Datums- und Uhrzeitangaben. Diese Funktionen ermöglichen die Extraktion bestimmter Komponenten eines Datums, die Durchführung von Berechnungen und die Bearbeitung von Datumswerten.

Die zulässigen Bezeichner in diesen Funktionen sind:

- JAHR
- MONAT
- TAG
- STUNDE
- MINUTE
- SECOND

Funktion	Signature	Beschreibung
NOW	JETZT ()	Gibt den aktuellen Zeitstempel mit Millisekundengenauigkeit zurück. Es gibt die genaue Uhrzeit zu dem Zeitpunkt an, zu dem es innerhalb einer Abfrage ausgeführt wird.
DATE_ADD	DATE_ADD (Bezeichner, Interval_Duration, Spalte)	Gibt die Summe aus a date/time und einer Zahl von oder aus einem days/hours Und-Intervall zurück. date/time date/time
DATE_SUB	DATE_SUB (Bezeichner, Interval_Duration, Spalte)	Gibt die Differenz zwischen einem date/time und einer

Funktion	Signature	Beschreibung
		Zahl von oder zwischen einem und einem days/hours Intervall zurück. date/time date/time
TIMESTAMP_ADD	TIMESTAMP_ADD (Bezeichner, Interval_Duration, Spalte)	Fügt einem Datetime-Ausdruck ein Zeitintervall in den angegebenen Zeiteinheiten hinzu.
TIMESTAMP_SUB	TIMESTAMP_SUB (Bezeichner, Interval_Duration, Spalte)	Subtrahiert ein Zeitintervall in den angegebenen Zeiteinheiten von einem Datetime-Ausdruck.
CAST	CAST (Ausdruck ALS TIMESTAMP-FORMAT-Muster)	Konvertiert einen Zeichenkettenausdruck unter Verwendung des angegebenen Formatmusters in einen Zeitstempel. Zu den gängigen Mustern gehört 'yyyy-MM-dd HH:mm:ss' das standardmäßige Datetime-Format. Beispiel: SELECT CAST('2023-12-25 14:30:00' AS TIMESTAMP) AS converted_timestamp

Example einer SQL-Abfrage unter Verwendung der aufgelisteten Funktionen:

```
SELECT r.asset_id, r.int_value,
       date_add(DAY, 7, r.event_timestamp) AS date_in_future,
       date_sub(YEAR, 2, r.event_timestamp) AS date_in_past,
       timestamp_add(DAY, 2, r.event_timestamp) AS timestamp_in_future,
       timestamp_sub(DAY, 2, r.event_timestamp) AS timestamp_in_past,
       now() AS time_now
```

```
FROM raw_time_series AS r
```

Funktionen zur Typkonvertierung

Typkonvertierungsfunktionen werden verwendet, um den Datentyp eines Werts von einem Wert in einen anderen zu ändern. Sie sind wichtig, um die Datenkompatibilität sicherzustellen und Operationen auszuführen, für die Daten in einem bestimmten Format erforderlich sind.

Funktion	Signature	Beschreibung
TO_DATE	- TO_DATE (Ganzzahl) - TO_DATE (Ausdruck, Format)	- Konvertiert eine Millisekunden-Ganzzahl aus einer Epoche in einen Datumswert. - Konvertiert eine Zeichenkettendarstellung in einen Datumswert.
TO_TIMESTAMP	- TO_TIMESTAMP (doppelt) - TO_TIMESTAMP (Zeichenfolge, Format)	- Konvertiert eine zweite Ganzzahl einer Epoche in einen Timestamp-Datentyp. - Konvertiert eine Zeichenkettendarstellung von Datum und Uhrzeit in einen Timestamp-Datentyp.
TO_TIME	- TO_TIME (int) - TO_TIME (Zeichenfolge, Format)	- Konvertiert eine Millisekunden-Ganzzahl aus einer Epoche in einen Zeitwert. - Konvertiert eine Zeichenkettendarstellung in einen Zeitwert.
CAST	BESETZUNG (<expression>AS<data type>)	Konvertiert eine Entität oder einen Ausdruck, der zu einem einzelnen Wert ausgewertet wird, von einem Typ in einen anderen.

Funktion	Signature	Beschreibung
		Unterstützte Datentypen sind: • BOOLEAN • INTEGER • INT • TIMESTAMP (ZEITSTEMPEL) • DATE • CHAR • CHARACTER • STRING

Example einer SQL-Abfrage unter Verwendung der aufgelisteten Funktionen:

```
SELECT TO_TIMESTAMP (100) AS timestamp_value,  
       TO_DATE(r.event_timestamp) AS date_value,  
       TO_TIME(r.event_timestamp) AS time_value  
FROM raw_time_series AS r
```

Aggregationsfunktionen

Aggregatfunktionen sind Datenbankoperationen, die Berechnungen über mehrere Datenzeilen hinweg durchführen, um ein einziges zusammengefasstes Ergebnis zu erhalten. Diese Funktionen analysieren Datensätze, um berechnete Werte wie Summen, Durchschnittswerte, Zahlen oder andere statistische Kennzahlen zurückzugeben.

Funktion	Signature	Beschreibung
AVG	AVG (Ausdruck)	Gibt den Durchschnitt eines numerischen Ausdrucks zurück.

Funktion	Signature	Beschreibung
COUNT	COUNT (Ausdruck)	Gibt die Anzahl der Zeilen zurück, die den angegebenen Kriterien entsprechen.
MAX	MAX (Ausdruck)	Gibt den größten Wert der ausgewählten Ausdrücke zurück.
MIN	MIN (Ausdruck)	Gibt den kleinsten Wert der ausgewählten Ausdrücke zurück.
SUM	SUM (Ausdruck)	Gibt die Summe eines numerischen Ausdrucks zurück.
STDDEV	STDDEV (Ausdruck)	Gibt die Standardabweichung der Stichprobe zurück.
GROUP BY	GROUP BY-Ausdruck	Gibt eine Zeile zurück, die durch die Gruppierung von Spalten erstellt wurde.
HAVING	HAT einen booleschen Ausdruck	Gibt nach Klausel gefilterte Gruppenzeilen zurück. GROUP BY

Beispiele für alle Funktionen:

Funktion	Beispiel
AVG	<pre>SELECT d.asset_id, d.property_id, AVG(d.int_value) FROM raw_time_ series AS d</pre>

Funktion	Beispiel
COUNT	<pre>SELECT COUNT(d.int_value) FROM raw_time_series AS d</pre>
MAX	<pre>SELECT MAX(d.int_value) FROM raw_time_ series AS d</pre>
MIN	<pre>SELECT MIN(d.int_value) FROM raw_time_ series AS d</pre>
SUM	<pre>SELECT SUM(d.int_value) FROM raw_time_ series AS d</pre>
STDDEV	<pre>SELECT STDDEV(d.int_value) FROM raw_time_series AS d</pre>
• GROUP BY • HAVING	<pre>SELECT MAX(d.int_value) AS max_int_v alue, d.asset_id FROM raw_time_series AS d GROUP BY d.asset_id HAVING MAX(d.int_value) > 5</pre>

Beispielabfragen

Metadatenfilterung

Das folgende Beispiel bezieht sich auf die Metadatenfilterung mit einer SELECT Anweisung in der AWS IoT SiteWise Abfragesprache:

```
SELECT a.asset_name, p.property_name
FROM asset a, asset_property p
WHERE a.asset_name LIKE 'Windmill%'
```

Filterung von Werten

Im Folgenden finden Sie ein Beispiel für die Wertfilterung mithilfe einer SELECT Anweisung in der AWS IoT SiteWise Abfragesprache:

```
SELECT a.asset_name, r.int_value
FROM asset a, raw_time_series r
WHERE r.int_value > 30
AND r.event_timestamp > TIMESTAMP '2022-01-05 12:15:00'
AND r.event_timestamp < TIMESTAMP '2022-01-05 12:20:00'
```

Optimierung der Abfrage

Metadatenfilter

Wenn Sie Metadaten oder Rohdaten abfragen, verwenden Sie die WHERE Klausel, um nach Metadatenfeldern zu filtern, um die Menge der gescannten Daten zu reduzieren. Verwenden Sie die folgenden Operatoren, um den Metadaten-Scan einzuschränken:

- Entspricht (=)
- Nicht gleich (!=)
- LIKE
- IN
- AND
- ODER

Verwenden Sie für Attributeigenschaften die folgenden Felder, um die Ergebnisse zu filtern. :

- double_attribute_value
- int_attribute_value
- boolean_attribute_value
- string_attribute_value

Diese Felder bieten eine bessere Leistung als die Tabelle latest_value_time_series für Asset-Eigenschaften des Attributtyps.

 Note

Verwenden Sie Literale auf der rechten Seite der Operatoren, um den Datenscan richtig einzuschränken. Die folgende Abfrage schneidet beispielsweise schlechter ab als die Verwendung eines strikten Zeichenkettenliterals:

```
SELECT property_id FROM asset_property WHERE property_name = CONCAT('my',  
'property')
```

Example für Metadatenfilter:

```
SELECT p.property_name FROM asset_property p  
WHERE p.property_type = 'attribute' AND p.string_attribute_value LIKE 'my-property-%'
```

Rohdatenfilter

Allen Rohdatentabellen (raw_time_series, latest_value_time_series, precomputed_aggregates) sind Zeitstempel mit ihren Zeilen verknüpft. Verwenden Sie zusätzlich zu den Metadatenfiltern auch Klauselfilter für das Feld, um die Menge der gescannten Daten zu reduzieren. WHERE event_timestamp Verwenden Sie die folgenden Operationen, um den Scan von Rohdaten einzuschränken:

- Entspricht (=)
- größer als (>)
- Kleiner als (<)
- Größer als oder gleich (>=)
- Kleiner als oder gleich (<=)
- BETWEEN
- AND

Beispiele für Filter:

- Geben Sie bei der Abfrage der Tabelle `precomputed_aggregates` immer einen Qualitätsfilter in der Klausel an. `WHERE` Dadurch wird die Datenmenge reduziert, die von der Abfrage gescannt wird, insbesondere wenn Sie nach OR-Daten suchen. `BAD UNCERTAIN`

Wir empfehlen außerdem dringend, bei der Abfrage der Tabelle `precomputed_aggregates` einen Auflösungsfilter (1m, 15m, 1h oder 1d) zu verwenden. Wenn Sie keinen Auflösungsfilter angeben, AWS IoT SiteWise wird standardmäßig ein vollständiger Tabellenscan für alle Auflösungen verwendet, was ineffizient ist.

- Bei der Abfrage von Rohdaten können in der `WHERE` Klausel auch Zeitstempelfunktionen verwendet werden, um die Menge der gescannten Daten zu filtern. Die folgende Abfrage scannt beispielsweise nur die Daten der letzten 30 Minuten aus der Tabelle `raw_time_series`:

```
SELECT r.event_timestamp, r.double_value
FROM raw_time_series r
WHERE r.event_timestamp > TIMESTAMP_SUB(MINUTE, 30, NOW())
```

Note

Entspricht nicht (`!=`) und `OR` Operatoren wenden normalerweise keine aussagekräftigen Filter auf den Rohdatenscan an. Filter für Rohdatenwerte (`string_value`, `double_value` usw.) schränken den Rohdatenscan ebenfalls nicht ein.

JOIN-Optimierung

AWS IoT SiteWiseSQL unterstützt das `JOIN` Schlüsselwort zum Zusammenführen von zwei Tabellen. Es werden nur `JOIN` s unterstützt, die aktiv nach einem Feld filtern (mithilfe des `ON` Schlüsselworts). Vollständige kartesische Verknüpfungen sind verboten.

AWS IoT SiteWiseunterstützt auch implizite `JOIN` s, ohne das Schlüsselwort zu verwenden. `JOIN` Diese sind zwischen verschiedenen Metadatentabellen sowie zwischen einer Metadatentabelle und einer Roh­tabelle zulässig. Beispielsweise würde diese Abfrage:

```
SELECT a.asset_name, p.property_name FROM asset a, asset_property p
```

Erbringt eine bessere Leistung als diese entsprechende Abfrage:

```
SELECT a.asset_name, p.property_name FROM asset a
```

```
JOIN asset_property p ON a.asset_id = p.asset_id
```

Die folgenden impliziten Verknüpfungen sind zulässig (O ist erlaubt, X ist verboten):

	Komponente	asset_property	Zeitreihe mit neuestem Wert	raw_time_series	vorberechnete_Aggregate	subquery
Komponente	X	O	O	O	O	X
Vermögensgegenstand	O	X	O	O	O	X
Zeitreihe mit neuestem Wert	O	O	X	X	X	X
raw_time_series	O	O	X	X	X	X
vorberechnete_Aggregate	O	O	X	X	X	X
subquery	X	X	X	X	X	X

Verwenden Sie nach Möglichkeit implizit. JOIN Wenn Sie das JOIN Schlüsselwort verwenden müssen, wenden Sie Filter auf die einzelnen ED-Tabellen an, um die JOIN Anzahl der gescannten Daten zu minimieren. Zum Beispiel anstelle dieser Abfrage:

```
SELECT level1.asset_id, level2.asset_id, level3.asset_id
FROM asset AS level1
JOIN asset AS level2 ON level2.parent_asset_id = level1.asset_id
JOIN asset AS level3 ON level3.parent_asset_id = level2.asset_id
WHERE level1.asset_name LIKE 'level1%'
```

```
AND level2.asset_name LIKE 'level2%'
AND level3.asset_name LIKE 'level3%'
```

Verwenden Sie diese effizientere Abfrage:

```
SELECT level1.asset_id, level2.asset_id, level3.asset_id
FROM asset AS level1
JOIN (SELECT asset_id, parent_asset_id FROM asset WHERE asset_name LIKE 'level2%') AS
    level2 ON level2.parent_asset_id = level1.asset_id
JOIN (SELECT asset_id, parent_asset_id FROM asset WHERE asset_name LIKE 'level3%') AS
    level3 ON level3.parent_asset_id = level2.asset_id
WHERE level1.asset_name LIKE 'level1%'
```

Indem Sie Metadatenfilter in Unterabfragen einfügen, stellen Sie sicher, dass einzelne Tabellen in den JOIN s während des Scanvorgangs gefiltert werden. Sie können das LIMIT Schlüsselwort auch in Unterabfragen verwenden, um den gleichen Effekt zu erzielen.

Große Abfragen

Für Abfragen, die mehr Zeilen als die Standardeinstellung erzeugen, legen Sie die Seitengröße der [ExecuteQuery](#) API auf den Maximalwert von 20000 fest. Dadurch wird die allgemeine Abfrageleistung verbessert.

Verwenden Sie die LIMIT Klausel, um die Menge der gescannten Daten für einige Abfragen zu reduzieren. Beachten Sie, dass für Aggregatfunktionen und bestimmte tabellenweite Klauseln (GROUP BY, ORDER BY, JOIN) ein vollständiger Scan erforderlich ist, bevor die Klausel angewendet wird.

LIMIT

Note

AWS IoT SiteWise kann selbst bei angewendeter LIMIT Klausel eine minimale Datenmenge scannen, insbesondere bei Rohdatenabfragen, bei denen mehrere Eigenschaften gescannt werden.

ODBC

Der [Open-Source-ODBC-Treiber](#) für AWS IoT SiteWise bietet eine SQL-relational Schnittstelle AWS IoT SiteWise für Entwickler und ermöglicht die Konnektivität von Business Intelligence (BI) -Tools

wie Power BI Desktop und Microsoft Excel. Der AWS IoT SiteWise ODBC-Treiber ist derzeit unter [Windows](#) verfügbar und unterstützt SSO mit Okta und Microsoft Azure Active Directory (AD).

Weitere Informationen finden Sie in der [AWS IoT SiteWiseODBC-Treiberdokumentation](#) unter. GitHub

Themen

- [Syntax und Optionen für die Verbindungszeichenfolge für den ODBC-Treiber](#)
- [Beispiele für Verbindungszeichenfolgen für AWS IoT SiteWiseODBC-Treiber](#)
- [Problembehandlung bei der Verbindung mit dem ODBC-Treiber](#)

Syntax und Optionen für die Verbindungszeichenfolge für den ODBC-Treiber

Die Syntax für die Angabe von Verbindungszeichenfolgenoptionen für den ODBC-Treiber lautet wie folgt:

```
Driver={AWS IoT SiteWiseODBC Driver};(option)=(value);
```

Die verfügbaren Optionen lauten wie folgt:

Verbindungsoptionen für den Treiber

- **Driver**(erforderlich) — Der Treiber, der mit ODBC verwendet wird.

Der Standardwert ist AWS IoT SiteWise.

- **DSN**— Der Datenquellenname (DSN), der für die Konfiguration der Verbindung verwendet werden soll.

Der Standardwert ist NONE.

- **Auth**— Der Authentifizierungsmodus. Dies muss einer der folgenden sein:
 - **AWS_PROFILE**— Verwenden Sie die standardmäßige Anmeldeinformationskette.
 - **IAM**— Verwenden Sie AWS IAM-Anmeldeinformationen.
 - **AAD**— Verwenden Sie den Azure Active Directory (AD) -Identitätsanbieter.
 - **OKTA**— Verwenden Sie den Okta-Identitätsanbieter.

Der Standardwert ist AWS_PROFILE.

Optionen für die Endpunktkonfiguration

- **EndpointOverride**— Die Endpunkt-Überschreibung für den AWS IoT SiteWise Dienst. Dies ist eine erweiterte Option, die die Region überschreibt. Beispiel:

```
iotsitewise.us-east-1.amazonaws.com
```

- **Region**— Die Signaturregion für den AWS IoT SiteWise Dienstendpunkt.

Der Standardwert ist `us-east-1`.

Option für den Anbieter von Anmeldeinformationen

- **ProfileName**— Der Profilname in der AWS Konfigurationsdatei.

Der Standardwert ist `NONE`.

AWSIAM-Authentifizierungsoptionen

- **UIDoder AccessKeyId**— Die ID des AWS Benutzerzugriffsschlüssels. Wenn UID sowohl als auch in der Verbindungszeichenfolge angegeben AccessKeyId werden, wird der UID Wert verwendet, sofern er nicht leer ist.

Der Standardwert ist `NONE`.

- **PWDoder SecretKey**— Der geheime AWS-Benutzerzugriffsschlüssel. Wenn PWD sowohl als auch in der Verbindungszeichenfolge angegeben SecretKey werden, wird der PWD Wert with verwendet, sofern er nicht leer ist.

Der Standardwert ist `NONE`.

- **SessionToken**— Das temporäre Sitzungstoken, das für den Zugriff auf eine Datenbank mit aktivierter Multi-Faktor-Authentifizierung (MFA) erforderlich ist. Fügen Sie kein abschließendes Zeichen `=` in die Eingabe ein.

Der Standardwert ist `NONE`.

SAML-based Authentifizierungsoptionen für Okta

- **IdPHost**— Der Hostname des angegebenen IdP.

Der Standardwert ist NONE.

- **UIDoder IdPUserName**— Der Benutzername für das angegebene IdP-Konto. Wenn beide UID und in der Verbindungszeichenfolge angegeben IdPUserName werden, wird der UID Wert verwendet, sofern er nicht leer ist.

Der Standardwert ist NONE.

- **PWDoder IdPPassword**— Das Passwort für das angegebene IdP-Konto. Wenn PWD sowohl als auch in der Verbindungszeichenfolge angegeben IdPPassword werden, wird der PWD Wert verwendet, sofern er nicht leer ist.

Der Standardwert ist NONE.

- **OktaApplicationID**— Die eindeutige Okta-provided ID, die der AWS IoT SiteWise Anwendung zugeordnet ist. Die Anwendungs-ID (AppId) finden Sie in dem entityID Feld, das in den Anwendungsmetadaten angegeben ist. Ein Beispiel ist:

```
entityID="http://www.okta.com//(IdPAppID)
```

Der Standardwert ist NONE.

- **RoleARN**— Der Amazon-Ressourcenname (ARN) der Rolle, die der Aufrufer annimmt.

Der Standardwert ist NONE.

- **IdPARN**— Der Amazon-Ressourcenname (ARN) des SAML-Anbieters in IAM, der den IdP beschreibt.

Der Standardwert ist NONE.

SAML-based Authentifizierungsoptionen für Azure Active Directory

- **UIDoder IdPUserName**— Der Benutzername für das angegebene IdP-Konto..

Der Standardwert ist NONE.

- **PWDoder IdPPassword**— Das Passwort für das angegebene IdP-Konto.

Der Standardwert ist NONE.

- **AADApplicationID**— Die eindeutige ID der registrierten Anwendung in Azure AD.

Der Standardwert ist NONE.

- **AADClientSecret**— Das Client-Geheimnis, das der registrierten Anwendung in Azure AD zugeordnet ist und zum Autorisieren des Abrufs von Token verwendet wird.

Der Standardwert ist NONE.

- **AADTenant**— Die Azure AD-Mandanten-ID.

Der Standardwert ist NONE.

- **RoleARN**— Der Amazon-Ressourcenname (ARN) der Rolle, die der Aufrufer annimmt.

Der Standardwert ist NONE.

- **IdPARN**— Der Amazon-Ressourcenname (ARN) des SAML-Anbieters in IAM, der den IdP beschreibt.

Der Standardwert ist NONE.

AWSSDK-Optionen (erweitert)

- **RequestTimeout**— Die Zeit in Millisekunden, die das AWS SDK auf eine Abfrageanforderung wartet, bevor es zu einem Timeout kommt. Jeder nicht positive Wert deaktiviert das Anforderungs-Timeout.

Der Standardwert ist 3000.

- **ConnectionTimeout**— Die Zeit in Millisekunden, in der das AWS SDK auf die Übertragung von Daten über eine offene Verbindung wartet, bevor es zu einem Timeout kommt. Ein Wert von 0 deaktiviert das Verbindungs-Timeout. Dieser Wert darf nicht negativ sein.

Der Standardwert ist 1000.

- **MaxRetryCountClient**— Die maximale Anzahl von Wiederholungsversuchen für Fehler, die wiederholt werden können, mit 5xx Fehlercodes im SDK. Der Wert darf nicht negativ sein.

Der Standardwert ist 0.

- **MaxConnections**— Die maximale Anzahl von erlaubten, gleichzeitig geöffneten HTTP-Verbindungen zu dem AWS IoT SiteWise Dienst. Der Wert muss positiv sein.

Der Standardwert ist 25.

Optionen für die ODBC-Treiberprotokollierung

- **LogLevel1**— Die Protokollebene für die Treiberprotokollierung. Zulässige Werte sind:
 - 0 (AUS).
 - 1 (FEHLER).
 - 2 (WARNUNG).
 - 3 (INFORMATION).
 - 4 (DEBUGGEN).

Die Standardeinstellung ist 1 (ERROR).

Warnung: Persönliche Informationen könnten vom Treiber protokolliert werden, wenn der DEBUG-Protokollierungsmodus verwendet wird.

- **LogOutput**— Ordner, in dem die Protokolldatei gespeichert werden soll.

Die Standardeinstellung ist:

- Windows: %USERPROFILE%, oder falls nicht verfügbar, %HOMEDRIVE%%HOMEPATH%.
- macOS und Linux: \$HOME oder falls nicht verfügbar, das Feld pw_dir aus dem getpwuid(getuid()) Rückgabewert der Funktion.

SDK-Protokollierungsoptionen

Die AWS SDK-Protokollebene ist von der Protokollebene des AWS IoT SiteWise ODBC-Treibers getrennt. Die Einstellung des einen hat keinen Einfluss auf das andere.

Die SDK-Protokollebene wird mithilfe der Umgebungsvariablen festgelegt `SW_AWS_LOG_LEVEL`. Folgende sind gültige Werte:

- OFF
- ERROR
- WARN
- INFO
- DEBUG
- TRACE
- FATAL

Wenn nicht festgelegt, `SW_AWS_LOG_LEVEL` wird die SDK-Protokollebene auf die Standardeinstellung gesetzt, d. `WARN` h.

Verbindung über einen Proxy herstellen

Der ODBC-Treiber unterstützt die Verbindung AWS IoT SiteWise über einen Proxy. Um diese Funktion zu verwenden, konfigurieren Sie die folgenden Umgebungsvariablen basierend auf Ihrer Proxyeinstellung:

- **`SW_PROXY_HOST`**— der Proxy-Host.
- **`SW_PROXY_PORT`**— Die Proxy-Portnummer.
- **`SW_PROXY_SCHEME`**— Das Proxyschema, entweder `http` oder `https`.
- **`SW_PROXY_USER`**— Der Benutzername für die Proxyauthentifizierung.
- **`SW_PROXY_PASSWORD`**— Das Benutzerkennwort für die Proxyauthentifizierung.
- **`SW_PROXY_SSL_CERT_PATH`**— Die SSL-Zertifikatsdatei, die für die Verbindung zu einem HTTPS-Proxy verwendet werden soll.
- **`SW_PROXY_SSL_CERT_TYPE`**— Der Typ des Proxy-Client-SSL-Zertifikats.
- **`SW_PROXY_SSL_KEY_PATH`**— Die private Schlüsseldatei, die für die Verbindung zu einem HTTPS-Proxy verwendet werden soll.
- **`SW_PROXY_SSL_KEY_TYPE`**— Der Typ der privaten Schlüsseldatei, die für die Verbindung zu einem HTTPS-Proxy verwendet wird.
- **`SW_PROXY_SSL_KEY_PASSWORD`**— Die Passphrase für die private Schlüsseldatei, die für die Verbindung zu einem HTTPS-Proxy verwendet wird.

Beispiele für Verbindungszeichenfolgen für AWS IoT SiteWise ODBC-Treiber

Beispiel für eine Verbindung zum ODBC-Treiber mit IAM-Anmeldeinformationen

```
Driver={AWS IoT SiteWiseODBC Driver};Auth=IAM;AccessKeyId=(your access key ID);SecretKey=(your secret key);SessionToken=(your session token);Region=us-east-1;
```

Beispiel für eine Verbindung zum ODBC-Treiber mit einem Profil

```
Driver={AWS IoT SiteWiseODBC Driver};ProfileName=(the profile name);region=us-east-1;
```

Der Treiber versucht, mit den in der Umgebungsvariablen angegebenen Anmeldeinformationen oder `~/.aws/credentials`, falls eine Datei in der Umgebungsvariablen angegeben ist `AWS_SHARED_CREDENTIALS_FILE`, mithilfe der Anmeldeinformationen in dieser Datei eine Verbindung herzustellen.

Beispiel für eine Verbindung zum ODBC-Treiber mit Okta

```
Driver={AWS IoT SiteWiseODBC Driver};Auth=OKTA;region=us-east-1;idPHost=(your host at Okta);idPUsername=(your user name);idPPassword=(your password);OktaApplicationID=(your Okta AppId);roleARN=(your role ARN);idPARN=(your Idp ARN);
```

Beispiel für eine Verbindung zum ODBC-Treiber mit Azure Active Directory (AAD)

```
Driver={AWS IoT SiteWiseODBC Driver};Auth=AAD;region=us-east-1;idPUsername=(your user name);idPPassword=(your password);aadApplicationID=(your AAD AppId);aadClientSecret=(your AAD client secret);aadTenant=(your AAD tenant);roleARN=(your role ARN);idPARN=(your idP ARN);
```

Beispiel für eine Verbindung zum ODBC-Treiber mit einem angegebenen Endpunkt und einer Protokollebene von 2 (WARNUNG)

```
Driver={AWS IoT SiteWiseODBC Driver};Auth=IAM;AccessKeyId=(your access key ID);SecretKey=(your secret key);EndpointOverride=iotsitewise.us-east-1.amazonaws.com;Region=us-east-1;LogLevel=2;
```

Problembehandlung bei der Verbindung mit dem ODBC-Treiber

Note

Wenn der Benutzername und das Passwort bereits im DSN angegeben sind, geben Sie sie nicht erneut an, wenn der ODBC-Treibermanager danach fragt.

Der Fehlercode `01S02` mit einer Meldung *Re-writing (connection string option) (have you specified it several times?)* tritt auf, wenn eine Verbindungszeichenfolgeoption in der Verbindungszeichenfolge mehr als einmal übergeben wird. Wenn Sie eine Option mehr als einmal angeben, wird ein Fehler ausgelöst. Wenn Sie eine Verbindung mit einem DSN und einer

Verbindungszeichenfolge herstellen und eine Verbindungsoption bereits im DSN angegeben ist, geben Sie sie nicht erneut in der Verbindungszeichenfolge an.

Interagiere mit anderen AWS Diensten

AWS IoT SiteWise kann Asset-Daten im Publish-Subscribe-Nachrichtenbroker von AWS IoT MQTT veröffentlichen, sodass Sie mit Ihren Asset-Daten aus anderen Diensten interagieren können. AWS IoT SiteWise weist jeder Asset-Eigenschaft ein eindeutiges MQTT-Thema zu, das Sie verwenden können, um Ihre Asset-Daten mithilfe von Core-Regeln an andere AWS Dienste weiterzuleiten. AWS IoT Sie können beispielsweise AWS IoT Core-Regeln für die folgenden Aufgaben konfigurieren:

- Ermittlung von Anlagenausfällen und Benachrichtigung der entsprechenden Mitarbeiter durch Senden von Daten an [AWS IoT Events](#).
- Historisieren Sie ausgewählte Asset-Daten zur Verwendung in externen Softwarelösungen, indem Sie Daten an [Amazon DynamoDB](#) senden.
- Generieren wöchentlicher Berichte durch Auslösen einer [AWS Lambda](#)-Funktion.

Sie können einem Tutorial folgen, das die Schritte beschreibt, die zum Einrichten einer Regel zum Speichern von Eigenschaftswerten in DynamoDB erforderlich sind. Weitere Informationen finden Sie unter [Aktualisierungen von Immobilienwerten in Amazon DynamoDB veröffentlichen](#).

Weitere Informationen zur Konfiguration einer Regel finden Sie unter [Regeln](#) im AWS IoT Entwicklerhandbuch.

Sie können auch Daten aus anderen AWS Diensten wieder in das System einlesen AWS IoT SiteWise. Informationen zur Aufnahme von Daten mithilfe der AWS IoT SiteWise Regelaktion finden Sie unter [Daten AWS IoT SiteWise mithilfe AWS IoT Core von Regeln aufnehmen](#).

Themen

- [Machen Sie sich mit den Eigenschaften von Assets in MQTT-Themen vertraut](#)
- [Aktivieren Sie Benachrichtigungen zu Vermögenswerten in AWS IoT SiteWise](#)
- [Benachrichtigungen über Vermögenseigenschaften abfragen in AWS IoT SiteWise](#)
- [Exportieren Sie Daten mit Benachrichtigungen über Vermögenseigenschaften nach Amazon S3](#)
- [Integrieren Sie AWS IoT SiteWise mit Grafana](#)
- [Integrieren AWS IoT SiteWise und AWS IoT TwinMaker](#)
- [Erkennen Sie Anomalien mit Lookout for Equipment](#)

Machen Sie sich mit den Eigenschaften von Assets in MQTT-Themen vertraut

Jede Komponenteneigenschaft verfügt über einen eindeutigen MQTT-Themenpfad im folgenden Format.

```
$aws/sitewise/asset-models/assetModelId/assets/assetId/properties/propertyId
```

Note

AWS IoT SiteWise unterstützt den Platzhalter für den Themenfilter # (mit mehreren Ebenen) in der AWS IoT Core Rules Engine nicht. Sie können den (einstufigen) Platzhalter + verwenden. So können Sie beispielsweise den folgenden Themenfilter verwenden, um alle Aktualisierungen für ein bestimmtes Komponentenmodell abzugleichen.

```
$aws/sitewise/asset-models/assetModelId/assets+/properties/+
```

Weitere Informationen zu Platzhaltern für Themenfilter finden Sie unter [Themen](#) im AWS IoT Core Developer Guide.

Aktivieren Sie Benachrichtigungen zu Vermögenswerten in AWS IoT SiteWise

Sie können Eigenschaftsbenachrichtigungen aktivieren AWS IoT Core, um Aktualisierungen der Objektdaten zu veröffentlichen und anschließend Abfragen für Ihre Daten durchzuführen. AWS IoT SiteWise Bietet mit Benachrichtigungen über Vermögenseigenschaften eine CloudFormation Vorlage, mit der Sie AWS IoT SiteWise Daten nach Amazon S3 exportieren können.

Note

Objektdaten werden bei AWS IoT Core jedem Empfang an gesendet AWS IoT SiteWise, unabhängig davon, ob sich der Wert geändert hat.

Themen

- [Aktivieren Sie Benachrichtigungen über Vermögenseigenschaften \(Konsole\)](#)
- [Aktivieren Sie Benachrichtigungen über Vermögenseigenschaften \(AWS CLI\)](#)

Aktivieren Sie Benachrichtigungen über Vermögenseigenschaften (Konsole)

Veröffentlicht standardmäßig AWS IoT SiteWise keine Aktualisierungen von Eigenschaftswerten. Sie können die AWS IoT SiteWise Konsole verwenden, um Benachrichtigungen für eine Objekteigenschaft zu aktivieren.

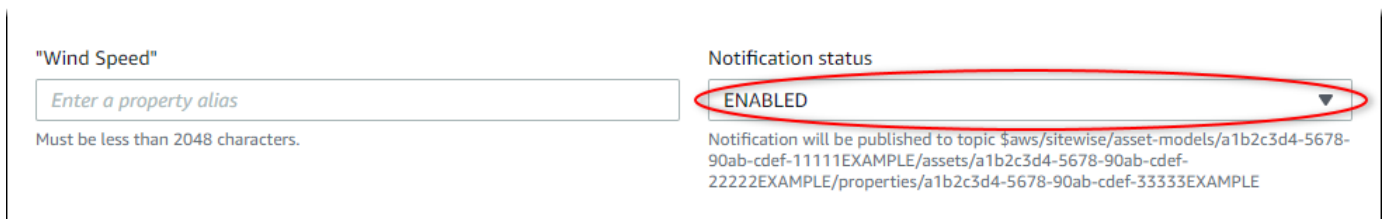
So aktivieren oder deaktivieren Sie Benachrichtigungen für eine Komponenteneigenschaft (Konsole)

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Komponenten aus.
3. Wählen Sie die Komponente aus, um die Benachrichtigungen einer Eigenschaft zu aktivieren.

Tip

Sie können eine Komponentenhierarchie mithilfe des Pfeilsymbols erweitern, um nach Ihrer Komponente zu suchen.

4. Wählen Sie Bearbeiten aus.
5. Wählen Sie für den Benachrichtigungsstatus der Komponenteneigenschaft AKTIVIERT aus.



The screenshot shows the AWS IoT SiteWise console interface. On the left, under the heading '"Wind Speed"', there is a text input field with the placeholder text 'Enter a property alias' and a note below it stating 'Must be less than 2048 characters.' To the right of this input field is a 'Notification status' dropdown menu. The dropdown is currently set to 'ENABLED' and is circled in red. Below the dropdown, a notification message is displayed: 'Notification will be published to topic \$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE'.

Sie können auch DEAKTIVIERT wählen, um Benachrichtigungen für die Komponenteneigenschaft zu deaktivieren.

6. Wählen Sie Speichern.

Aktivieren Sie Benachrichtigungen über Vermögenseigenschaften (AWS CLI)

Veröffentlicht standardmäßig AWS IoT SiteWise keine Aktualisierungen von Eigenschaftswerten. Sie können das AWS Command Line Interface (AWS CLI) verwenden, um Benachrichtigungen für eine Asset-Eigenschaft zu aktivieren oder zu deaktivieren.

Um dieses Verfahren abzuschließen, müssen Sie die `assetId` Ihrer Komponenten und die `propertyId` Ihrer Eigenschaft kennen. Sie können auch die externe ID verwenden. Wenn Sie ein Asset erstellt haben und es nicht kennen `assetId`, verwenden Sie die [ListAssets](#) API, um alle Assets für ein bestimmtes Modell aufzulisten. Verwenden Sie den [DescribeAsset](#) Vorgang, um die Eigenschaften Ihres Assets einschließlich der Immobilien anzuzeigen IDs.

Verwenden Sie den [UpdateAssetProperty](#) Vorgang, um Benachrichtigungen für eine Vermögenseigenschaft zu aktivieren oder zu deaktivieren. Geben Sie die folgenden Parameter an:

- `assetId`— Die ID des Vermögenswerts.
- `propertyId`— Die ID des Vermögenswerts.
- `propertyNotificationState`— Status der Benachrichtigung über den Immobilienwert: `ENABLED` oder `DISABLED`.
- `propertyAlias`— Der Alias der Immobilie. Geben Sie den vorhandenen Alias der Eigenschaft an, wenn Sie den Benachrichtigungsstatus aktualisieren. Wenn Sie diesen Parameter auslassen, wird der vorhandene Alias der Eigenschaft entfernt.

So aktivieren oder deaktivieren Sie Benachrichtigungen für eine Komponenteneigenschaft (CLI)

1. Führen Sie den folgenden Befehl aus, um den Alias der Komponenteneigenschaft abzurufen. `asset-id` Ersetzen Sie es durch die ID des Assets und `property-id` durch die ID der Immobilie.

```
aws iotsitewise describe-asset-property \  
  --asset-id asset-id \  
  --property-id property-id
```

Die Operation gibt eine Antwort zurück, die Informationen zur Komponenteneigenschaft im folgenden Format enthält. Der Eigenschaftensalias befindet sich in `assetProperty.alias` im JSON-Objekt.

```
{
  "assetId": "a1b2c3d4-5678-90ab-cdef-2222EXAMPLE",
  "assetName": "Wind Turbine 7",
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-1111EXAMPLE",
  "assetProperty": {
    "id": "a1b2c3d4-5678-90ab-cdef-3333EXAMPLE",
    "name": "Wind Speed",
    "alias": "/company/windfarm/3/turbine/7/windspeed",
    "notification": {
      "topic": "$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-1111EXAMPLE/
assets/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE/properties/a1b2c3d4-5678-90ab-
cdef-3333EXAMPLE",
      "state": "DISABLED"
    },
    "dataType": "DOUBLE",
    "unit": "m/s",
    "type": {
      "measurement": {}
    }
  }
}
```

2. Führen Sie den folgenden Befehl aus, um Benachrichtigungen für die Komponenteneigenschaft zu aktivieren. *property-alias* Ersetzen Sie es durch den Eigenschaftsalias aus der Antwort des vorherigen Befehls, oder lassen Sie es aus, die Eigenschaft ohne Alias `--property-alias` zu aktualisieren.

```
aws iotsitewise update-asset-property \
  --asset-id asset-id \
  --property-id property-id \
  --property-notification-state ENABLED \
  --property-alias property-alias
```

Sie können auch `--property-notification-state DISABLED` übergeben, um Benachrichtigungen für die Komponenteneigenschaft zu deaktivieren.

Benachrichtigungen über Vermögenseigenschaften abfragen in AWS IoT SiteWise

Um Benachrichtigungen über Vermögenseigenschaften abzufragen, erstellen Sie AWS IoT Core Regeln, die aus SQL-Anweisungen bestehen.

AWS IoT SiteWise veröffentlicht Aktualisierungen von Asset-Eigenschaftsdaten in AWS IoT Core im folgenden Format.

```
{
  "type": "PropertyValueUpdate",
  "payload": {
    "assetId": "String",
    "propertyId": "String",
    "values": [
      {
        "timestamp": {
          "timeInSeconds": Number,
          "offsetInNanos": Number
        },
        "quality": "String",
        "value": {
          "booleanValue": Boolean,
          "doubleValue": Number,
          "integerValue": Number,
          "stringValue": "String",
          "nullValue": {
            "valueType": "String"
          }
        }
      }
    ]
  }
}
```

Jede Struktur in der values Liste ist eine timestamp-quality-value (TQV-) Struktur.

- `timestamp` enthält die aktuelle Unix-Epoche in Sekunden mit Nanosekunden-Offset.
- `quality` enthält eine der folgenden Zeichenfolgen zur Angabe der Qualität des Datenpunkts:
 - `GOOD`— Die Daten sind von keinen Problemen betroffen.

- BAD— Die Daten sind von einem Problem wie einem Sensorausfall betroffen.
- UNCERTAIN— Die Daten sind von einem Problem wie einer Sensorungenauigkeit betroffen.
- value enthält abhängig vom Typ der Eigenschaft eines der folgenden Felder:
 - booleanValue
 - doubleValue
 - integerValue
 - stringValue
 - nullValue

nullValue— Eine Struktur mit dem folgenden Feld, das den Typ des Eigenschaftswerts mit dem Wert Null und der Qualität oder angibt. BAD UNCERTAIN

- valueType— Aufzählung von {"B", "D", "S", "I"}

Um Werte aus dem values-Array zu analysieren, müssen Sie in den SQL-Anweisungen Ihrer Regeln komplexe verschachtelte Objektabfragen verwenden. Weitere Informationen finden Sie unter [Abfragen verschachtelter Objekte](#) im AWS IoT Entwicklerhandbuch oder in der Anleitung finden Sie ein konkretes Beispiel für das [Aktualisierungen von Immobilienwerten in Amazon DynamoDB veröffentlichen](#) Analysieren von Benachrichtigungen über Objekteigenschaften.

Example Beispielabfrage zum Extrahieren des Werte-Arrays

Die folgende Anweisung veranschaulicht, wie das Array aktueller Eigenschaftswerte für eine bestimmte Eigenschaft vom doppelten Typ für alle Komponenten mit dieser Eigenschaft abgefragt wird.

```
SELECT
  (SELECT VALUE (value.doubleValue) FROM payload.values) AS windspeed
FROM
  '$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/+/
properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE'
WHERE
  type = 'PropertyValueUpdate'
```

Die vorherige Anweisung zur Regelabfrage gibt Daten im folgenden Format aus.

```
{
```

```
"windspeed": [  
  26.32020195042838,  
  26.282584572975477,  
  26.352566977372508,  
  26.283084346171442,  
  26.571883739599322,  
  26.60684140743005,  
  26.628738636715045,  
  26.273486932802125,  
  26.436379105473964,  
  26.600590095377303  
]  
}
```

Example Beispielabfrage zum Extrahieren eines einzelnen Wertes

Die folgende Anweisung veranschaulicht, wie der erste Wert aus dem Array von Eigenschaftswerten für eine bestimmte Eigenschaft vom doppelten Typ für alle Komponenten mit dieser Eigenschaft abgefragt wird.

```
SELECT  
  get((SELECT VALUE (value.doubleValue) FROM payload.values), 0) AS windspeed  
FROM  
  '$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/+/properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE'  
WHERE  
  type = 'PropertyValueUpdate'
```

Die vorherige Anweisung zur Regelabfrage gibt Daten im folgenden Format aus.

```
{  
  "windspeed": 26.32020195042838  
}
```

Important

Diese Regelabfrageanweisung ignoriert Wertaktualisierungen, abgesehen vom ersten in jedem Stapel. Jeder Stapel kann bis zu 10 Werte enthalten. Wenn Sie die verbleibenden Werte einschließen müssen, ist es erforderlich, eine komplexere Lösung einzurichten, um Eigenschaftswerte der Komponenten an andere Services auszugeben. Sie können beispielsweise eine Regel mit einer AWS Lambda Aktion einrichten, um jeden Wert im Array

erneut in einem anderen Thema zu veröffentlichen, und eine weitere Regel einrichten, um dieses Thema abzufragen und jeden Wert in der gewünschten Regelaktion zu veröffentlichen.

Exportieren Sie Daten mit Benachrichtigungen über Vermögenseigenschaften nach Amazon S3

Sie können eingehende Daten aus AWS IoT SiteWise einem Amazon S3 S3-Bucket in Ihrem Konto exportieren. Sie können Ihre Daten in einem Format sichern, das Sie verwenden können, um historische Berichte zu erstellen oder Ihre Daten mit komplexen Methoden zu analysieren.

Um Zeitreihendaten aus zu exportieren AWS IoT SiteWise, aktivieren Sie die Cold-Tier-Funktion, damit die Daten in einem Amazon S3 S3-Bucket gespeichert werden. Einzelheiten finden [Sie unter Datenspeicher verwalten in AWS IoT SiteWise](#).

Verwenden Sie zum Exportieren von Asset-Modell- und Asset-Metadaten die Funktion für Massenoperationen AWS IoT SiteWise, um Metadaten in einen Amazon S3 S3-Bucket zu exportieren. Einzelheiten finden Sie unter [Massenoperationen mit Assets und Modellen](#).

Integrieren Sie AWS IoT SiteWise mit Grafana

Grafana ist eine Datenvisualisierungsplattform zur Visualisierung und Überwachung von Daten in Dashboards. Verwenden Sie in Grafana-Version 10.4.0 und höher das AWS IoT SiteWise Plugin, um Ihre AWS IoT SiteWise Asset-Daten in Grafana-Dashboards zu visualisieren. Benutzer können Daten aus mehreren AWS Quellen (wie AWS IoT SiteWise Amazon Timestream und Amazon CloudWatch) und anderen Datenquellen mit einem einzigen Grafana-Dashboard visualisieren.

Sie haben zwei Möglichkeiten, das Plugin zu verwenden: AWS IoT SiteWise

- Lokale Grafana-Server

Sie können das AWS IoT SiteWise Plugin auf einem Grafana-Server einrichten, den Sie verwalten. Weitere Informationen zum Hinzufügen und Verwenden des Plugins finden Sie in der [AWS IoT SiteWise Datasource-README-Datei](#) auf der Website. GitHub

- AWS Managed Service for Grafana

Sie können das AWS IoT SiteWise Plugin im AWS Managed Service for Grafana (AMG) verwenden. AMG verwaltet Grafana-Server für Sie, sodass Sie Ihre Daten visualisieren können,

ohne Hardware oder andere Grafana-Infrastruktur erstellen, paketieren oder bereitstellen zu müssen. Weitere Informationen finden Sie in den folgenden Themen im AWS Managed Service for Grafana Grafana-Benutzerhandbuch:

- [Was ist Amazon Managed Service for Grafana \(AMG\)?](#)
- [Verwenden der AWS IoT SiteWise Datenquelle](#)

Example Beispiel für ein Grafana-Dashboard

Das folgende Grafana-Dashboard visualisiert den [Demo-Windpark](#). Sie können auf dieses Demo-Dashboard auf der [Grafana Play-Website](#) zugreifen.



Integrieren AWS IoT SiteWise und AWS IoT TwinMaker

Durch die Integration AWS IoT TwinMaker mit erhalten Sie Zugriff auf robuste Funktionen wie die AWS IoT SiteWise ExecuteQuery Datenabruf-API und die erweiterte Asset-Suche in der AWS IoT SiteWise Konsole. AWS IoT SiteWise Um die Dienste zu integrieren und diese Funktionen zu nutzen, müssen Sie zuerst die Integration aktivieren.

Themen

- [Aktivierung der Integration](#)
- [Integrieren und AWS IoT SiteWise AWS IoT TwinMaker](#)

Aktivierung der Integration

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann. Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Weitere Informationen zu AWS IoT SiteWise unterstützten Aktionen finden Sie unter [Actions defined by AWS IoT SiteWise](#) in der Service Authorization Reference.

Weitere Informationen zu AWS IoT TwinMaker dienstverknüpften Rollen finden Sie unter [Dienstbezogene Rollen für AWS IoT TwinMaker](#) im AWS IoT TwinMaker Benutzerhandbuch.

Bevor Sie AWS IoT SiteWise und integrieren können AWS IoT TwinMaker, müssen Sie die folgenden Berechtigungen erteilen, die die Integration in einen AWS IoT TwinMaker verknüpften Workspace ermöglichen AWS IoT SiteWise :

- `iotsitewise:EnableSiteWiseIntegration`— Ermöglicht AWS IoT SiteWise die Integration in einen verknüpften AWS IoT TwinMaker Workspace. Diese Integration ermöglicht AWS IoT TwinMaker das Einlesen all Ihrer Modellierungsinformationen AWS IoT SiteWise über eine AWS IoT TwinMaker serviceverknüpfte Rolle. Um diese Berechtigung zu aktivieren, fügen Sie Ihrer IAM-Rolle die folgende Richtlinie hinzu:

JSON

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

```
    "Effect": "Allow",
    "Action": [
        "iotsitewise:EnableSiteWiseIntegration"
    ],
    "Resource": "*"
  }
]
```

Integrieren und AWS IoT SiteWise AWS IoT TwinMaker

Um AWS IoT SiteWise und zu integrieren AWS IoT TwinMaker, benötigen Sie Folgendes:

- AWS IoT SiteWise In Ihrem Konto ist eine dienstbezogene Rolle eingerichtet
- AWS IoT TwinMaker In Ihrem Konto wurde eine dienstbezogene Rolle eingerichtet
- AWS IoT TwinMaker Workspace mit ID `IoTSiteWiseDefaultWorkspace` in deinem Konto in der Region.

Zur Integration mithilfe der AWS IoT SiteWise Konsole

Wenn Sie das AWS IoT TwinMaker Banner „Integration mit“ in der Konsole sehen, wählen Sie Grant Permission aus. Die Voraussetzungen werden in Ihrem Konto erstellt.

Zur Integration mit dem AWS CLI

Geben Sie zur Integration AWS IoT SiteWise und AWS IoT TwinMaker mithilfe von die folgenden Befehle ein: AWS CLI

1. Rufen Sie `CreateServiceLinkedRole` mit einem `AWSServiceName` von `aniotsitewise.amazonaws.com`.

```
aws iam create-service-linked-role --aws-service-name iotsitewise.amazonaws.com
```

2. Rufen Sie `CreateServiceLinkedRole` mit einem `AWSServiceName` von `iottwinmaker.amazonaws.com`.

```
aws iam create-service-linked-role --aws-service-name iottwinmaker.amazonaws.com
```

3. Rufen Sie `CreateWorkspace` mit einem ID von `anIoTSiteWiseDefaultWorkspace`.

```
aws iottwinmaker create-workspace --workspace-id IoTSiteWiseDefaultWorkspace
```

Erkennen Sie Anomalien mit Lookout for Equipment

Note

Die Erkennung von Anomalien ist nur in den Regionen verfügbar, in denen Amazon Lookout for Equipment verfügbar ist.

Sie können Amazon Lookout for Equipment integrieren AWS IoT SiteWise , um mithilfe von Anomalieerkennung und vorausschauender Wartung von Industrieanlagen Einblicke in Ihre Industrieanlagen zu gewinnen. Lookout for Equipment ist ein Service für maschinelles Lernen (ML) zur Überwachung von Industrieanlagen, der abnormales Geräteverhalten erkennt und potenzielle Ausfälle identifiziert. Mit Lookout for Equipment können Sie prädiktive Wartungsprogramme implementieren und suboptimale Geräteprozesse identifizieren. Weitere Informationen zu Lookout for Equipment finden Sie unter [Was ist Amazon Lookout for Equipment?](#) im Amazon Lookout for Equipment Equipment-Benutzerhandbuch.

Wenn Sie eine Prognose erstellen, um ein ML-Modell zu trainieren, um anomales Geräteverhalten zu erkennen, AWS IoT SiteWise sendet es die Werte der Anlageneigenschaften an Lookout for Equipment, um ein ML-Modell zur Erkennung von anomalem Geräteverhalten zu trainieren. Um eine Prognosedefinition für ein Asset-Modell zu definieren, geben Sie die IAM-Rollen an, die Lookout for Equipment benötigt, um auf Ihre Daten zuzugreifen, und die Eigenschaften, die an Lookout for Equipment gesendet und verarbeitete Daten an Amazon S3 gesendet werden sollen. Weitere Informationen finden Sie unter [Erstellen Sie Asset-Modelle in AWS IoT SiteWise](#).

Um Lookout for Equipment zu integrieren AWS IoT SiteWise , führen Sie die folgenden allgemeinen Schritte aus:

- Fügen Sie einem Asset-Modell eine Prognosedefinition hinzu, die beschreibt, welche Eigenschaften Sie verfolgen möchten. Die Vorhersagedefinition ist eine wiederverwendbare Sammlung von Messungen, Transformationen und Metriken, die verwendet wird, um Vorhersagen für die Anlagen zu erstellen, die auf diesem Anlagenmodell basieren.
- Trainieren Sie die Vorhersage auf der Grundlage der von Ihnen bereitgestellten historischen Daten.

- Planen Sie Inferenz, die angibt, AWS IoT SiteWise wie oft eine bestimmte Vorhersage ausgeführt werden soll.

Sobald die Inferenz geplant ist, überwacht das Modell Lookout for Equipment die Daten, die es von Ihren Geräten empfängt, und sucht nach Anomalien im Geräteverhalten. Sie können die Ergebnisse in SiteWise Monitor mithilfe der AWS IoT SiteWise GET-API-Operationen oder der Lookout for Equipment Equipment-Konsole anzeigen und analysieren. Sie können auch Alarme mithilfe von Alarmmeldern aus dem Anlagenmodell erstellen, um Sie über abnormales Geräteverhalten zu informieren.

Themen

- [Fügen Sie eine Vorhersagedefinition hinzu \(Konsole\)](#)
- [Trainieren Sie eine Vorhersage \(Konsole\)](#)
- [Starten oder beenden Sie die Inferenz für eine Vorhersage \(Konsole\)](#)
- [Eine Vorhersagedefinition hinzufügen \(CLI\)](#)
- [Trainieren Sie eine Vorhersage und starten Sie die Inferenz \(CLI\)](#)
- [Eine Vorhersage trainieren \(CLI\)](#)
- [Inferenz auf eine Vorhersage starten oder beenden \(CLI\)](#)

Fügen Sie eine Vorhersagedefinition hinzu (Konsole)

Um mit dem Senden der von gesammelten Daten AWS IoT SiteWise an Lookout for Equipment zu beginnen, müssen Sie einem Anlagenmodell eine AWS IoT SiteWise Prognosedefinition hinzufügen.

Um einem AWS IoT SiteWise Anlagenmodell eine Vorhersagedefinition hinzuzufügen

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Modelle und dann das Assetmodell aus, dem Sie die Vorhersagedefinition hinzufügen möchten.
3. Wählen Sie Prognosen aus.
4. Wählen Sie Vorhersagedefinition hinzufügen.
5. Definieren Sie Details zur Vorhersagedefinition.

- a. Geben Sie einen eindeutigen Namen und eine Beschreibung für Ihre Prognosedefinition ein. Wählen Sie den Namen sorgfältig aus, da Sie den Namen der Vorhersagedefinition nicht mehr ändern können, nachdem Sie sie erstellt haben.
- b. Erstellen oder wählen Sie eine IAM-Berechtigungsrolle aus, die es AWS IoT SiteWise ermöglicht, Ihre Asset-Daten mit Amazon Lookout for Equipment zu teilen. Die Rolle sollte die folgenden IAM- und Vertrauensrichtlinien haben. Hilfe zum Erstellen der Rolle finden Sie unter [Erstellen einer Rolle mithilfe benutzerdefinierter Vertrauensrichtlinien \(Konsole\)](#).

IAM-Richtlinie

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "L4EPermissions",
      "Effect": "Allow",
      "Action": [
        "lookoutequipment:CreateDataset",
        "lookoutequipment:CreateModel",
        "lookoutequipment:CreateInferenceScheduler",
        "lookoutequipment:DescribeDataset",
        "lookoutequipment:DescribeModel",
        "lookoutequipment:DescribeInferenceScheduler",
        "lookoutequipment:ListInferenceExecutions",
        "lookoutequipment:StartDataIngestionJob",
        "lookoutequipment:StartInferenceScheduler",
        "lookoutequipment:UpdateInferenceScheduler",
        "lookoutequipment:StopInferenceScheduler"
      ],
      "Resource": [
        "arn:aws:lookoutequipment:us-east-1:123456789012:inference-scheduler/IoTSiteWise_*",
        "arn:aws:lookoutequipment:us-east-1:123456789012:model/IoTSiteWise_*",
        "arn:aws:lookoutequipment:us-east-1:123456789012:dataset/IoTSiteWise_*"
      ]
    }
  ]
}
```

```

    {
      "Sid": "L4EPermissions2",
      "Effect": "Allow",
      "Action": [
        "lookoutequipment:DescribeDataIngestionJob"
      ],
      "Resource": "*"
    },
    {
      "Sid": "S3Permissions",
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:ListBucket",
        "s3:PutObject",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::iotsitewise-*"
      ]
    },
    {
      "Sid": "IAMPermissions",
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::111122223333:role/Role_name"
    }
  ]
}

```

Vertrauensrichtlinie

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {

```



```

        "Effect": "Allow",
        "Principal": {
            "Service": "iotsitewise.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "123456789012"
            },
            "ArnEquals": {
                "aws:SourceArn": "arn:aws:iotsitewise:us-
east-1:123456789012:asset/*"
            }
        }
    },
    {
        "Effect": "Allow",
        "Principal": {
            "Service": "lookoutequipment.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "123456789012"
            },
            "ArnEquals": {
                "aws:SourceArn": "arn:aws:lookoutequipment:us-
east-1:123456789012:*"
            }
        }
    }
}
]
}

```

- c. Wählen Sie Weiter aus.
6. Wählen Sie Datenattribute (Messungen, Transformationen und Metriken) aus, die Sie an Lookout for Equipment senden möchten.
 - a. (Optional) Wählen Sie Messungen aus.
 - b. (Optional) Wählen Sie Transformationen aus.
 - c. (Optional) Wählen Sie Metriken aus.
 - d. Wählen Sie Weiter aus.

7. Überprüfen Sie Ihre Auswahl. Um die Prognosedefinition zum Asset-Modell hinzuzufügen, wählen Sie auf der Übersichtsseite die Option Vorhersagedefinition hinzufügen aus.

Sie können auch eine bestehende Vorhersagedefinition bearbeiten oder löschen, der aktive Vorhersagen angehängt sind.

Trainieren Sie eine Vorhersage (Konsole)

Nachdem Sie einem Anlagenmodell eine Prognosedefinition hinzugefügt haben, können Sie die Vorhersagen trainieren, die sich auf Ihre Anlagen beziehen.

Um eine Vorhersage zu trainieren in AWS IoT SiteWise

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Assets und dann das Asset aus, das Sie überwachen möchten.
3. Wählen Sie Prognosen aus.
4. Wählen Sie die Vorhersagen aus, die Sie trainieren möchten.
5. Wählen Sie unter Aktionen die Option Training starten aus und gehen Sie wie folgt vor:
 - a. Wählen Sie unter Prognosedetails eine IAM-Berechtigungsrolle aus, mit der Sie Ihre Asset-Daten mit Lookout for Equipment teilen können AWS IoT SiteWise . Wenn Sie eine neue Rolle erstellen müssen, wählen Sie Neue Rolle erstellen aus.
 - b. Geben Sie unter Einstellungen für Trainingsdaten einen Zeitraum für Trainingsdaten ein, um auszuwählen, welche Daten zum Trainieren der Vorhersage verwendet werden sollen.
 - c. (Optional) Wählen Sie die Samplerate für die Daten nach der Nachverarbeitung aus.
 - d. (Optional) Geben Sie für Datenlabels einen Amazon S3 S3-Bucket und ein Präfix an, das Ihre Kennzeichnungsdaten enthält. Weitere Informationen zur Kennzeichnung von Daten finden Sie unter [Kennzeichnen Ihrer Daten](#) im Amazon Lookout for Equipment Equipment-Benutzerhandbuch.
 - e. Wählen Sie Weiter aus.
6. (Optional) Wenn Sie möchten, dass die Vorhersage aktiv ist, sobald das Training abgeschlossen ist, wählen Sie unter Erweiterte Einstellungen die Option Vorhersage nach dem Training automatisch aktivieren aus, und gehen Sie dann wie folgt vor:

- a. Definieren Sie unter Eingabedaten für Häufigkeit des Daten-Uploads, wie oft Daten hochgeladen werden, und definieren Sie für Offset-Verzögerungszeit, wie viel Puffer verwendet werden soll.
 - b. Wählen Sie Weiter aus.
7. Überprüfen Sie die Details der Prognose und wählen Sie Speichern und starten aus.

Starten oder beenden Sie die Inferenz für eine Vorhersage (Konsole)

Note

Die Gebühren von Lookout for Equipment fallen für geplante Inferenzen mit den Daten an, die zwischen AWS IoT SiteWise und Lookout for Equipment übertragen werden. Weitere Informationen finden Sie unter [Amazon Lookout for Equipment Pricing](#).

Wenn Sie die Vorhersage hinzugefügt `lookoutequipment:CreateDataset`, sie aber nach dem Training nicht aktiviert haben, müssen Sie sie aktivieren, um mit der Überwachung Ihrer Ressourcen zu beginnen.

Um die Inferenz für eine Vorhersage zu starten

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Assets und dann das Asset aus, zu dem die Prognose hinzugefügt werden soll.
3. Wählen Sie Prognosen aus.
4. Wählen Sie die Prognosen aus, die Sie aktivieren möchten.
5. Wählen Sie unter Aktionen die Option Inferenz starten aus und gehen Sie wie folgt vor:
 - a. Definieren Sie unter Eingabedaten für Häufigkeit des Daten-Uploads, wie oft Daten hochgeladen werden, und definieren Sie für Offset-Verzögerungszeit, wie viel Puffer verwendet werden soll.
 - b. Wählen Sie Speichern und starten.

Um die Inferenz für eine Vorhersage zu beenden

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Assets und dann das Asset aus, zu dem die Prognose hinzugefügt werden soll.
3. Wählen Sie Prognosen aus.
4. Wählen Sie die Vorhersagen aus, die Sie beenden möchten.
5. Wählen Sie unter Aktionen die Option Inferenz beenden aus.

Eine Vorhersagedefinition hinzufügen (CLI)

Um eine Vorhersagedefinition für ein neues oder vorhandenes Asset-Modell zu definieren, können Sie die AWS Command Line Interface (AWS CLI) verwenden. Nachdem Sie die Prognosedefinition für das Anlagenmodell definiert haben, trainieren Sie eine Vorhersage für eine Anlage und planen die Inferenz für diese, AWS IoT SiteWise um mit Lookout for Equipment eine Anomalieerkennung durchzuführen.

Voraussetzungen

Um diese Schritte ausführen zu können, müssen Sie ein Anlagenmodell und mindestens eine Anlage erstellt haben. Weitere Informationen erhalten Sie unter [Erstellen Sie ein Asset-Modell \(AWS CLI\)](#) und [Erstellen Sie ein Asset \(AWS CLI\)](#).

Wenn Sie noch nicht damit vertraut sind AWS IoT SiteWise, müssen Sie den `CreateBulkImportJob` API-Vorgang aufrufen, in AWS IoT SiteWise den die Eigenschaftswerte der Anlage importiert werden. Dieser Vorgang wird dann zum Trainieren des Modells verwendet. Weitere Informationen finden Sie unter [Erstellen Sie einen AWS IoT SiteWise Massenimportauftrag \(\)AWS CLI](#).

Um eine Vorhersagedefinition hinzuzufügen

1. Erstellen Sie eine Datei mit dem Namen `asset-model-payload.json`. Folgen Sie den Schritten in diesen anderen Abschnitten, um der Datei die Details Ihres Asset-Modells hinzuzufügen, reichen Sie aber nicht die Anfrage zur Erstellung oder Aktualisierung des Asset-Modells ein.
 - Weitere Informationen zum Erstellen eines Vermögensmodells finden Sie unter [Erstellen Sie ein Asset-Modell \(AWS CLI\)](#)

- Weitere Informationen zum Aktualisieren eines vorhandenen Asset-Modells finden Sie unter [Aktualisieren Sie ein Objektmodell, ein Komponentenmodell oder eine Schnittstelle \(AWS CLI\)](#)
2. Fügen Sie dem Asset-Modell ein Verbundmodell von Lookout for Equipment (`assetModelCompositeModels`) hinzu, indem Sie den folgenden Code hinzufügen.
- **Property** Ersetzen Sie es durch die ID der Eigenschaften, die Sie einbeziehen möchten. Um diese zu bekommen IDs, rufen Sie an [DescribeAssetModel](#).
 - **RoleARN** Ersetzen Sie es durch den ARN einer IAM-Rolle, die Lookout for Equipment den Zugriff auf Ihre AWS IoT SiteWise Daten ermöglicht.

```
{
  ...
  "assetModelCompositeModels": [
    {
      "name": "L4Epredictiondefinition",
      "type": "AWS/L4E_ANOMALY",
      "properties": [
        {
          "name": "AWS/L4E_ANOMALY_RESULT",
          "dataType": "STRUCT",
          "dataTypeSpec": "AWS/L4E_ANOMALY_RESULT",
          "unit": "none",
          "type": {
            "measurement": {}
          }
        },
        {
          "name": "AWS/L4E_ANOMALY_INPUT",
          "dataType": "STRUCT",
          "dataTypeSpec": "AWS/L4E_ANOMALY_INPUT",
          "type": {
            "attribute": {
              "defaultValue": "{\"properties\": [\"Property1\", \"Property2\"]}"
            }
          }
        },
        {
          "name": "AWS/L4E_ANOMALY_PERMISSIONS",
          "dataType": "STRUCT",
          "dataTypeSpec": "AWS/L4E_ANOMALY_PERMISSIONS",
          "type": {
```

```

        "attribute": {
            "defaultValue": "{\"roleArn\": \"RoleARN\"}"
        }
    },
    {
        "name": "AWS/L4E_ANOMALY_DATASET",
        "dataType": "STRUCT",
        "dataTypeSpec": "AWS/L4E_ANOMALY_DATASET",
        "type": {
            "attribute": {}
        }
    },
    {
        "name": "AWS/L4E_ANOMALY_MODEL",
        "dataType": "STRUCT",
        "dataTypeSpec": "AWS/L4E_ANOMALY_MODEL",
        "type": {
            "attribute": {}
        }
    },
    {
        "name": "AWS/L4E_ANOMALY_INFERENCE",
        "dataType": "STRUCT",
        "dataTypeSpec": "AWS/L4E_ANOMALY_INFERENCE",
        "type": {
            "attribute": {}
        }
    },
    {
        "name": "AWS/L4E_ANOMALY_TRAINING_STATUS",
        "dataType": "STRUCT",
        "dataTypeSpec": "AWS/L4E_ANOMALY_TRAINING_STATUS",
        "type": {
            "attribute": {
                "defaultValue": "{}"
            }
        }
    },
    {
        "name": "AWS/L4E_ANOMALY_INFERENCE_STATUS",
        "dataType": "STRUCT",
        "dataTypeSpec": "AWS/L4E_ANOMALY_INFERENCE_STATUS",
        "type": {

```

```

        "attribute": {
            "defaultValue": "{}"
        }
    }
}
]
}

```

3. Erstellen Sie das Asset-Modell oder aktualisieren Sie das bestehende Asset-Modell. Führen Sie eine der folgenden Aktionen aus:

- Führen Sie den folgenden Befehl aus, um das Asset-Modell zu erstellen:

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

- Führen Sie den folgenden Befehl aus, um das bestehende Asset-Modell zu aktualisieren. *asset-model-id* Ersetzen Sie es durch die ID des Asset-Modells, das Sie aktualisieren möchten.

```
aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --cli-input-json file://asset-model-payload.json
```

Nachdem Sie den Befehl ausgeführt haben, notieren Sie sich das `assetModelId` in der Antwort.

Trainieren Sie eine Vorhersage und starten Sie die Inferenz (CLI)

Nachdem die Definition der Vorhersage nun definiert ist, können Sie darauf basierende Ressourcen trainieren und mit der Inferenz beginnen. Wenn Sie Ihre Vorhersage trainieren, aber keine Inferenz starten möchten, fahren Sie mit fort. [Eine Vorhersage trainieren \(CLI\)](#) Um die Vorhersage zu trainieren und die Inferenz für das Asset zu starten, benötigen Sie die `assetId` der Zielressource.

Um die Vorhersage zu trainieren und mit der Inferenz zu beginnen

- Führen Sie den folgenden Befehl aus, um das `assetModelCompositeModelId` `assetModelCompositeModelSummaries` Under zu finden. *asset-model-id* Ersetzen Sie es durch die ID des Asset-Modells, in dem Sie es erstellt haben [Aktualisieren Sie ein Objektmodell, ein Komponentenmodell oder eine Schnittstelle \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \
  --asset-model-id asset-model-id \
```

2. Führen Sie den folgenden Befehl aus, um `actionDefinitionId` die `TrainingWithInference` Aktion zu finden. *asset-model-id* Ersetzen Sie durch die im vorherigen Schritt verwendete ID und *asset-model-composite-model-id* ersetzen Sie sie durch die im vorherigen Schritt zurückgegebene ID.

```
aws iotsitewise describe-asset-model-composite-model \
  --asset-model-id asset-model-id \
  --asset-model-composite-model-id asset-model-composite-model-id \
```

3. Erstellen Sie eine Datei mit dem Namen `train-start-inference-prediction.json` und fügen Sie den folgenden Code hinzu, der den folgenden ersetzt:

- *asset-id* mit der ID des Ziel-Assets
- *action-definition-id* mit der ID der `TrainingWithInference` Aktion
- *StartTime* mit dem Beginn der Trainingsdaten, angegeben in Epochensekunden
- *EndTime* mit dem Ende der Trainingsdaten, angegeben in Epochensekunden
- *TargetSamplingRate* mit der Abtastrate der Daten nach der Nachbearbeitung durch Lookout for Equipment. Zulässige Werte sind: PT1S | PT5S | PT10S | PT15S | PT30S | PT1M | PT5M | PT10M | PT15M | PT30M | PT1H.

```
{
  "targetResource": {
    "assetId": "asset-id"
  },
  "actionDefinitionId": "action-definition-Id",
  "actionPayload": {
    "stringValue": "{\"l4ETrainingWithInference\":{\"trainingWithInferenceMode\": \"START\", \"trainingPayload\": {\"exportDataStartTime\": StartTime, \"exportDataEndTime\": EndTime}, \"targetSamplingRate\": \"TargetSamplingRate\"}, \"inferencePayload\": {\"dataDelayOffsetInMinutes\": 0, \"dataUploadFrequency\": \"PT5M\"}}}"
  }
}
```

4. Führen Sie den folgenden Befehl aus, um das Training und die Inferenz zu starten:


```
aws iotsitewise execute-action --cli-input-json file://train-start-inference-prediction.json
```

Eine Vorhersage trainieren (CLI)

Da die Prognosedefinition nun definiert ist, können Sie darauf aufbauend Anlagen trainieren. Um die Vorhersage auf der Anlage zu trainieren, benötigen Sie die `assetId` der Zielressource.

Um die Vorhersage zu trainieren

1. Führen Sie den folgenden Befehl aus, um das `assetModelCompositeModelId` Under zu finden `assetModelCompositeModelSummaries`. `asset-model-id` Ersetzen Sie es durch die ID des Asset-Modells, in dem Sie es erstellt haben [Aktualisieren Sie ein Objektmodell, ein Komponentenmodell oder eine Schnittstelle \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \  
  --asset-model-id asset-model-id \  
  --asset-model-composite-model-id asset-model-composite-model-id \  
  --action-definition-id action-definition-id \  
  --start-time start-time \  
  --end-time end-time \  
  --bucket-name bucket-name \  
  --prefix prefix
```

2. Führen Sie den folgenden Befehl aus, um `actionDefinitionId` die Training Aktion zu finden. `asset-model-id` Ersetzen Sie durch die im vorherigen Schritt verwendete ID und `asset-model-composite-model-id` ersetzen Sie sie durch die im vorherigen Schritt zurückgegebene ID.

```
aws iotsitewise describe-asset-model-composite-model \  
  --asset-model-id asset-model-id \  
  --asset-model-composite-model-id asset-model-composite-model-id \  
  --action-definition-id action-definition-id \  
  --start-time start-time \  
  --end-time end-time \  
  --bucket-name bucket-name \  
  --prefix prefix
```

3. Erstellen Sie eine Datei mit dem Namen `train-prediction.json` und fügen Sie den folgenden Code hinzu, der den folgenden ersetzt:
 - `asset-id` mit der ID des Ziel-Assets
 - `action-definition-id` mit der ID der Trainingsaktion
 - `StartTime` mit dem Beginn der Trainingsdaten, angegeben in Epochensekunden
 - `EndTime` mit dem Ende der Trainingsdaten, angegeben in Epochensekunden
 - (Optional) `BucketName` mit dem Namen des Amazon S3 S3-Buckets, der Ihre Etikettendaten enthält
 - (Optional) `Prefix` mit dem Präfix, das dem Amazon S3 S3-Bucket zugeordnet ist.

- *TargetSamplingRate* mit der Abtastrate der Daten nach der Nachbearbeitung durch Lookout for Equipment. Zulässige Werte sind: PT1S | PT5S | PT10S | PT15S | PT30S | PT1M | PT5M | PT10M | PT15M | PT30M | PT1H.

 Note

Geben Sie sowohl den Bucket-Namen als auch das Präfix oder keines von beiden an.

```
{
  "targetResource": {
    "assetId": "asset-id"
  },
  "actionDefinitionId": "action-definition-Id",
  "actionPayload": { "stringValue": "{\"l4ETraining\": {\"trainingMode\": \"START\", \"exportDataStartTime\": StartTime, \"exportDataEndTime\": EndTime, \"targetSamplingRate\": \"TargetSamplingRate\", \"labelInputConfiguration\": {\"bucketName\": \"BucketName\", \"prefix\": \"Prefix\"}}}"
  }
}
```

4. Führen Sie den folgenden Befehl aus, um das Training zu starten:

```
aws iotsitewise execute-action --cli-input-json file://train-prediction.json
```

Bevor Sie mit der Inferenz beginnen können, muss das Training abgeschlossen sein. Gehen Sie wie folgt vor, um den Status der Schulung zu überprüfen:

- Navigieren Sie in der Konsole zu dem Asset, für das sich die Prognose bezieht.
- Rufen Sie von der AWS CLI aus `BatchGetAssetPropertyValue` über `propertyId` die `trainingStatus` Eigenschaft auf.

Inferenz auf eine Vorhersage starten oder beenden (CLI)

Sobald die Vorhersage trainiert ist, können Sie mit der Inferenz beginnen und Lookout for Equipment anweisen, mit der Überwachung Ihrer Anlagen zu beginnen. Um die Inferenz zu starten oder zu beenden, benötigen Sie die Daten `assetId` der Zielressource.

Um die Inferenz zu starten

1. Führen Sie den folgenden Befehl aus, um das `assetModelCompositeModelId` unter `assetModelCompositeModelSummaries` zu finden. *asset-model-id* Ersetzen Sie es durch die ID des Asset-Modells, in dem Sie es erstellt haben [Aktualisieren Sie ein Objektmodell, ein Komponentenmodell oder eine Schnittstelle \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \
  --asset-model-id asset-model-id \
```

2. Führen Sie den folgenden Befehl aus, um `actionDefinitionId` die Inference Aktion zu finden. *asset-model-id* Ersetzen Sie durch die im vorherigen Schritt verwendete ID und *asset-model-composite-model-id* ersetzen Sie sie durch die im vorherigen Schritt zurückgegebene ID.

```
aws iotsitewise describe-asset-model-composite-model \
  --asset-model-id asset-model-id \
  --asset-model-composite-model-id asset-model-composite-model-id \
```

3. Erstellen Sie eine Datei mit dem Namen `start-inference.json` und fügen Sie den folgenden Code hinzu, der den folgenden ersetzt:

- *asset-id* mit der ID des Ziel-Assets
- *action-definition-id* mit der ID der Start-Inferenzaktion
- *Offset* mit der Menge des zu verwendenden Puffers
- *Frequency* mit wie oft Daten hochgeladen werden

```
{
  "targetResource": {
    "assetId": "asset-id"
  },
  "actionDefinitionId": "action-definition-Id",
  "actionPayload": { "stringValue": "{\\"l4EInference\\": {\\"inferenceMode\\":\\"START\\",\\"dataDelayOffsetInMinutes\\": Offset, \\"dataUploadFrequency\\": \\"Frequency\\"}\\""
  }
}
```

4. Führen Sie den folgenden Befehl aus, um die Inferenz zu starten:

```
aws iotsitewise execute-action --cli-input-json file://start-inference.json
```

Um die Inferenz zu beenden

1. Führen Sie den folgenden Befehl aus, um das `assetModelCompositeModelId` `assetModelCompositeModelSummaries` Under zu finden. `asset-model-id` Ersetzen Sie es durch die ID des Asset-Modells, in dem Sie es erstellt haben [Aktualisieren Sie ein Objektmodell, ein Komponentenmodell oder eine Schnittstelle \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \  
  --asset-model-id asset-model-id \  
  --asset-model-composite-model-id asset-model-composite-model-id \  
  --action-definition-id action-definition-id \  
  --cli-input-json file://stop-inference.json
```

2. Führen Sie den folgenden Befehl aus, um `actionDefinitionId` die Inference Aktion zu finden. `asset-model-id` Ersetzen Sie durch die im vorherigen Schritt verwendete ID und `asset-model-composite-model-id` ersetzen Sie sie durch die im vorherigen Schritt zurückgegebene ID.

```
aws iotsitewise describe-asset-model-composite-model \  
  --asset-model-id asset-model-id \  
  --asset-model-composite-model-id asset-model-composite-model-id \  
  --action-definition-id action-definition-id \  
  --cli-input-json file://stop-inference.json
```

3. Erstellen Sie eine Datei mit dem Namen `stop-inference.json` und fügen Sie den folgenden Code hinzu, der den folgenden ersetzt:

- `asset-id` mit der ID des Ziel-Assets
- `action-definition-id` mit der ID der Start-Inferenzaktion

```
{  
  "targetResource": {  
    "assetId": "asset-id"  
  },  
  "actionDefinitionId": "action-definition-id",  
  "actionPayload": { "stringValue": "{\"l4EInference\":{\"inferenceMode\":\"STOP\"}}"}  
}
```

4. Führen Sie den folgenden Befehl aus, um die Inferenz zu beenden:

```
aws iotsitewise execute-action --cli-input-json file://stop-inference.json
```

Native Erkennung von Anomalien

AWS IoT SiteWise Bei der systemeigenen Anomalieerkennung handelt es sich um eine Funktion des maschinellen Lernens (ML) zur Überwachung von Industrieanlagen, die abnormales Geräteverhalten erkennt und potenzielle Ausfälle identifiziert. Mit der systemeigenen Anomalieerkennung können Sie prädiktive Wartungsprogramme implementieren und suboptimale Geräteprozesse identifizieren.

AWS IoT SiteWise Für die native Erkennung von Anomalien sind keine umfangreichen ML-Kenntnisse oder Erfahrungen erforderlich. Sie wählen einfach die Eigenschaften aus, um ein benutzerdefiniertes ML-Modell zu trainieren, das potenzielle Fehler für Sie findet. **AWS IoT SiteWise** Die native Anomalieerkennung erstellt automatisch das am besten geeignete Modell, um die normalen Betriebsbedingungen Ihrer Geräte zu ermitteln. Das Modell ist optimiert, um abnormales Geräteverhalten zu ermitteln, das in den historischen Daten aufgetreten ist. Mithilfe der AWS IoT SiteWise Konsole oder des SDK führen Sie das Modell aus, um neue Zeitreihendaten gemäß Ihrem gewünschten Zeitplan zu verarbeiten.

Um die AWS IoT SiteWise native Anomalieerkennung zu verwenden, gehen Sie wie folgt vor:

- Wählen Sie die Eigenschaften und den Zeitraum aus, für den Sie trainieren möchten.
- Fügen Sie die Perioden historischer Fehler hinzu, die in den Daten (Labeldaten) angezeigt wurden, falls vorhanden.
- Trainieren Sie Ihr ML-Modell mithilfe der AWS IoT SiteWise systemeigenen Anomalieerkennung. Optional können Sie ein automatisches Retraining konfigurieren, um Ihr Modell im Laufe der Zeit auf dem neuesten Stand zu halten.
- Richten Sie Ihren Inferenzplan ein, um Ihre Live-Datenströme mit Ihrem trainierten Modell zu testen.

Die native Anomalieerkennung überwacht stationäre und stationäre Industrieanlagen, deren Betriebsbedingungen nur begrenzt variabel sind. Zu den unterstützten Geräten gehören rotierende Maschinen wie Pumpen, Kompressoren, Motoren, CNC-Maschinen und Turbinen. Zu den Anwendungen in der Prozessindustrie gehören Wärmetauscher, Kessel und Wechselrichter. Die native Erkennung von Anomalien ist ein Back-End-Analysedienst, der in bestehende Wartungssysteme integriert ist AWS IoT SiteWise und diese ergänzt.

Themen

- [Funktionen zur systemeigenen Erkennung von Anomalien](#)

- [Voraussetzungen](#)
- [Aktivieren Sie die Erkennung von Anomalien auf den Sensoren einer Anlage](#)
- [Aktivieren Sie die Erkennung von Anomalien auf Sensoren in allen Anlagen](#)
- [Fortgeschrittene Trainingskonfigurationen](#)
- [Erweiterte Inferenzkonfigurationen](#)
- [Überprüfen Sie die Inferenzergebnisse](#)
- [Löst bei anomalem Verhalten benutzerdefinierte Aktionen aus \(\)AWS-Managementkonsole](#)
- [Best Practices](#)

Funktionen zur systemeigenen Erkennung von Anomalien

Die AWS IoT SiteWise Plattform bietet eine Reihe leistungsstarker Funktionen, die Ihre industriellen Abläufe verändern können:

- **Vorausschauende Wartung:** Erkennen Sie Ausfälle frühzeitig und lassen sich in Arbeitsauftragssysteme integrieren, um auf der Grundlage von Erkenntnissen zu handeln.
- **Enge Integration mit AWS IoT SiteWise:** Nutzen Sie die enge Integration in Ihre bestehende Infrastruktur, sodass Sie Ihre Daten nicht an einen externen Service übertragen müssen.
- **Automatische Modellauswahl und Schulung:** Profitieren Sie von dieser Unterstützung AWS IoT SiteWise, ohne dass Sie Fachwissen im Bereich maschinelles Lernen benötigen.
- **Passen Sie Ihre Inferenzplanung an:** Planen Sie Inferenzaufträge so, dass sie Ihren betrieblichen Anforderungen und Schichtzeiten entsprechen.
- **Gekennzeichnete Daten:** Verbessern Sie die Genauigkeit mit bekannten Ausfallintervallen während des Modelltrainings.
- **Modellevaluierung mit punktwiser Diagnose:** Evaluieren Sie die Modellleistung auf Ereignisebene.
- **Anlagenübergreifend skalierbar:** Erstellen Sie Berechnungsmodelle für einzelne Anlagen oder für mehrere Anlagen, um Skalierbarkeit zu erreichen.
- **Einblicke auf Sensorebene:** Erhalten Sie detaillierte Diagnosen, mit denen die spezifischen Sensoren identifiziert werden können, die zu einer Anomalie beigetragen haben.
- **Schnellere Wertschöpfung:** Der Übergang von der Sensormodellierung zur Erkennung von Anomalien dauert nur wenige Stunden, nicht Wochen.

Voraussetzungen

Um diese Schritte ausführen zu können, müssen Sie ein Asset-Modell haben und mindestens ein Asset erstellt haben. Weitere Informationen finden Sie unter [Erstellen eines Asset-Modells \(AWS CLI\)](#) und [Erstellen eines Assets \(AWS CLI\)](#). Derzeit bieten wir keine Unterstützung für externe Geräte IDs an.

Wenn Sie noch nicht mit historischen Daten vertraut sind AWS IoT SiteWise (und Sie noch keine historischen Daten haben), müssen Sie die [CreateBulkImportJob](#)API aufrufen, in die Sie die Eigenschaftswerte importieren möchten AWS IoT SiteWise. Dies wird verwendet, um das Modell zu trainieren. Weitere Informationen finden Sie unter [Erstellen eines AWS IoT SiteWise Massenimportauftrags \(AWS CLI\)](#).

Einrichtung AWS CLI für das Berechnungsmodell APIs

Gehen Sie wie folgt vor, um Ihre AWS CLI Konfiguration zu aktualisieren und auf das Berechnungsmodell zuzugreifen: APIs

- Installieren Sie die neueste awscli-Version. `aws-cli`
- Überprüfen Sie die Installation, indem Sie nach dem neuen suchen: APIs

```
aws iotsitewise help
```

In der Befehlsausgabe wird die vollständige Liste der Operationen angezeigt AWS IoT SiteWise APIs, einschließlich der neu hinzugefügten Berechnungsmodelloperationen.

Anforderungen an die Immobilie

Um die Anomalieerkennung einzurichten, müssen Sie die folgenden Anforderungen erfüllen und die [UpdateAssetModel \(AWS CLI\)](#):

- Mindestens eine Eingabeeigenschaft, die entweder vom INTEGER Datentyp DOUBLE oder ist. Sie ist entweder eine Mess- oder eine Transformationseigenschaft und wird zum Trainieren des Modells verwendet.
- Eine Ergebniseigenschaft vom STRING Datentyp. Es muss sich um eine Messeigenschaft handeln und speichert die Ergebnisse der Anomalieerkennung.

Voraussetzungen für die Kennzeichnung

- Laden Sie Ihre Datenbeschriftungen in einen Amazon S3 S3-Bucket hoch.
- Aktualisieren Sie die Bucket-Richtlinie für diesen Bucket, AWS IoT SiteWise damit Ihre Labels gelesen werden können.

Gehen Sie auf der Konsole zu Berechtigungen -> Bucket-Richtlinie. Ersetzen Sie den Bucket ARN durch den ARN Ihres Buckets.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SiteWiseS3ReadAccess",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucketbucket-name",
        "arn:aws:s3:::amzn-s3-demo-bucketbucket-name/*"
      ]
    }
  ]
}
```

Voraussetzungen für die Modellevaluierung

- Die Modellevaluierung generiert punktweise Modelldiagnosen an dem von Ihnen angegebenen Amazon S3 S3-Bucket-Speicherort.
- Damit die punktuellen Diagnoseergebnisse in Ihren Amazon S3 S3-Bucket geschrieben werden können, aktualisieren Sie die Bucket-Richtlinie dieses Buckets, sodass AWS IoT SiteWise die Ergebnisse geschrieben werden können.

Gehen Sie auf der Konsole zu. Permissions -> Bucket policy Ersetzen Sie den Bucket ARN durch den ARN Ihres Buckets.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SiteWiseS3Access",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ]
    }
  ]
}
```

Voraussetzungen für Modellmetriken

- Modellmetriken bieten umfassende Einblicke in die Leistung, einschließlich der Bewertung der Modellqualität, Statistiken zur Ereigniserkennung und Vergleichsmetriken zwischen Modellversionen am angegebenen Amazon S3 S3-Bucket-Speicherort.
- Damit die JSON-Datei für Modellmetriken in Ihren Amazon S3 S3-Bucket geschrieben werden kann, aktualisieren Sie die Bucket-Richtlinie, AWS IoT SiteWise damit die Ergebnisse geschrieben werden können.

Gehen Sie auf der Konsole zu **Permissions** -> **Bucket policy**. Ersetzen Sie den Bucket ARN durch den ARN Ihres Buckets.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "SiteWiseS3Access",
```

```

        "Effect": "Allow",
        "Action": [
            "s3:GetObject",
            "s3:ListBucket",
            "s3:PutObject"
        ],
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-bucket",
            "arn:aws:s3:::amzn-s3-demo-bucket-s3-bucket;/*"
        ]
    }
}

```

Aktivieren Sie die Erkennung von Anomalien auf den Sensoren einer Anlage

Erstellen Sie ein Berechnungsmodell ()AWS CLI

Verwenden Sie die AWS Befehlszeilenschnittstelle ()AWS CLI, um ein Berechnungsmodell zu erstellen. Nachdem Sie das Berechnungsmodell definiert haben, trainieren Sie das Modell und planen Sie die Inferenz für die Erkennung von Anomalien an einer Anlage in. AWS IoT SiteWise

- Erstellen Sie eine Datei `anomaly-detection-computation-model-payload.json` mit dem folgenden Inhalt:

```

{
  "computationModelName": "anomaly-detection-computation-model-name",
  "computationModelConfiguration": {
    "anomalyDetection": {
      "inputProperties": "${input_properties}",
      "resultProperty": "${result_property}"
    }
  },
  "computationModelDataBinding": {
    "input_properties": {
      "list": [{
        "assetModelProperty": {
          "assetModelId": "asset-model-id",
          "propertyId": "input-property-id-1"
        }
      ]
    }
  }
}

```

```

    }
  },
  {
    "assetModelProperty": {
      "assetModelId": "asset-model-id",
      "propertyId": "input-property-id-2"
    }
  }
],
"result_property": {
  "assetModelProperty": {
    "assetModelId": "asset-model-id",
    "propertyId": "results-property-id"
  }
}
}
}

```

- Führen Sie den folgenden Befehl aus, um ein Berechnungsmodell zu erstellen:

```

aws iotsitewise create-computation-model \
  --cli-input-json file://anomaly-detection-computation-model-payload.json

```

ExecuteAction Vorbereitung der API-Nutzlast

Die nächsten Schritte zur Durchführung von Training und Inferenz werden mit der [ExecuteAction](#) API durchgeführt. Sowohl Training als auch Inferenz werden mit einer JSON-Aktionsnutzlastkonfiguration konfiguriert. Beim Aufrufen der [ExecuteAction](#) API muss die Aktionsnutzlast als Wert zusammen mit einer Nutzlast bereitgestellt werden. stringValue

Die Payload muss sich strikt an die API-Anforderungen halten. Insbesondere muss der Wert eine flache Zeichenfolge ohne Steuerzeichen sein (z. B. Zeilenumbrüche, Tabulatoren oder Zeilenumbrüche).

Die folgenden Optionen bieten zwei zuverlässige Methoden zur Bereitstellung einer gültigen Aktions-Payload:

Option 1: Verwenden Sie eine saubere Payload-Datei

Das folgende Verfahren beschreibt die Schritte für eine saubere Payload-Datei:

1. Säubern Sie die Datei, um Steuerzeichen zu entfernen.

```
tr -d '\n\r\t' < original-action-payload.json > training-or-inference-action-payload.json
```

2. Führen Sie die Aktion mit der Datei `stringValue@=file:///...`

```
aws iotsitewise execute-action \
  --target-resource computationModelId=<MODEL_ID> \
  --action-definition-id <ACTION_DEFINITION_ID> \
  --resolve-to assetId=<ASSET_ID> \
  --action-payload stringValue@=file:///training-or-inference-action-payload.json
```

Option 2: Inline-Zeichenfolge mit maskierten Anführungszeichen

In den folgenden Schritten werden die Schritte beschrieben, um die Nutzdaten inline bereitzustellen und Zwischendateien zu vermeiden:

- Verwenden Sie maskierte Anführungszeichen (`\"`) innerhalb der JSON-Zeichenfolge.
- Setzen Sie den gesamten `StringValue=..` Ausdruck in doppelte Anführungszeichen.

Example einer Payload einer Aktion, der entgangen ist:

```
aws iotsitewise execute-action \
  --target-resource computationModelId=<MODEL_ID> \
  --action-definition-id <ACTION_DEFINITION_ID> \
  --resolve-to assetId=<ASSET_ID> \
  --action-payload "stringValue={\"exportDataStartTime\":1717225200,
  \"exportDataEndTime\":1722789360,\"targetSamplingRate\": \"PT1M\"}"
```

Trainiere das Modell (AWS CLI)

Mit einem erstellten Berechnungsmodell können Sie ein Modell anhand der Ressourcen trainieren. Gehen Sie wie folgt vor, um ein Modell für eine Anlage zu trainieren:

1. Führen Sie den folgenden Befehl aus, um `actionDefinitionId` die AWS/ANOMALY_DETECTION_TRAINING Aktion zu ermitteln. `computation-model-id` Ersetzen Sie es durch die im vorherigen Schritt zurückgegebene ID.

```
aws iotsitewise describe-computation-model \  
  --computation-model-id computation-model-id
```

2. Erstellen Sie eine Datei mit dem Namen `anomaly-detection-training-payload.json` und fügen Sie die folgenden Werte hinzu:

 Note

Die Nutzlast muss den Anforderungen entsprechen. [Option 1: Verwenden Sie eine saubere Payload-Datei](#)

- a. `StartTime` mit dem Beginn der Trainingsdaten, angegeben in Epochensekunden.
- b. `EndTime` mit dem Ende der Trainingsdaten, angegeben in Epochensekunden.
- c. Sie können optional konfigurieren [Fortgeschrittene Trainingskonfigurationen](#), um die Modellleistung zu verbessern.
 - i. (Optional) `TargetSamplingRate` mit der Abtastrate der Daten.
 - ii. (Optional) `LabelInputConfiguration` zur Angabe von Zeiträumen, in denen anomales Verhalten auftrat, um das Modelltraining zu verbessern.
 - iii. (Optional) `ModelEvaluationConfiguration` um die Leistung des Modells zu bewerten, indem nach Abschluss des Trainings Inferenzen für einen bestimmten Zeitraum ausgeführt werden.
 - iv. (Optional) `ModelMetricsDestination` zur Erfassung umfassender Leistungsdaten (Präzision, Recall, Area Under the Curve).

```
{  
  "trainingMode": "TRAIN_MODEL",  
  "exportDataStartTime": StartTime,  
  "exportDataEndTime": EndTime  
}
```

Example eines Beispiels für eine Trainingsnutzlast:

```
{  
  "trainingMode": "TRAIN_MODEL",
```

```
"exportDataStartTime": 1717225200,
"exportDataEndTime": 1722789360
}
```

3. Führen Sie den folgenden Befehl aus, um das Training zu starten. Ersetzen Sie die folgenden Parameter im Befehl:
 - a. `computation-model-id` mit der ID des Zielberechnungsmodells.
 - b. `asset-id` mit der ID des Assets, anhand dessen Sie das Modell trainieren werden.
 - c. `training-action-definition-id` mit der ID der AWS/ANOMALY_DETECTION_TRAINING Aktion aus Schritt 1.

```
aws iotsitewise execute-action \
  --target-resource computationModelId=computation-model-id \
  --resolve-to assetId=asset-id \
  --action-definition-id training-action-definition-id \
  --action-payload stringValue@=file://anomaly-detection-training-payload.json
```

Example einer Ausführungsaktion:

```
aws iotsitewise execute-action --target-resource computationModelId=27cb824c-
fd84-45b0-946b-0a5b0466d890 --resolve-to assetId=cefd4b68-481b-4735-
b466-6a4220cd19ee --action-definition-id e54cea94-5d1c-4230-a59e-4f54dcbc972d --
action-payload stringValue@=file://anomaly-detection-training-payload.json
```

4. Führen Sie den folgenden Befehl aus, um den Status des Modelltrainingsprozesses zu überprüfen. Die letzte Ausführungsübersicht zeigt den Ausführungsstatus (RUNNING/COMPLETED/FAILED).

```
aws iotsitewise list-executions \
  --target-resource-type COMPUTATION_MODEL \
  --target-resource-id computation-model-id \
  --resolve-to-resource-type ASSET \
  --resolve-to-resource-id asset-id
```

5. Führen Sie den folgenden Befehl aus, um die Konfiguration des zuletzt trainierten Modells zu überprüfen. Dieser Befehl erzeugt nur dann eine Ausgabe, wenn mindestens ein Modell erfolgreich trainiert wurde.

```
aws iotsitewise describe-computation-model-execution-summary \  
  --computation-model-id computation-model-id \  
  --resolve-to-resource-type ASSET \  
  --resolve-to-resource-id asset-id
```

6. Wenn a verwendet ComputationModel wird AssetModelProperty, verwenden Sie die [ListComputationModelResolveToResources](#)API, um die Assets zu identifizieren, für die Aktionen ausgeführt wurden.

```
aws iotsitewise list-computation-model-resolve-to-resources \  
  --computation-model-id computation-model-id
```

Startet und beendet das erneute Training des Modells ()AWS CLI

Nach dem ersten Modelltraining können Sie automatisches Retraining konfigurieren, um Datenabweichungen entgegenzuwirken und die Modellgenauigkeit im Laufe der Zeit aufrechtzuerhalten. Mit dem Retrainingplaner können Sie regelmäßige Modellaktualisierungen mit konfigurierbaren Promotion-Modi einrichten.

Starten Sie den Retrainingplaner

1. Führen Sie den folgenden Befehl aus, um die Aktion `actionDefinitionId` zu ermitteln. `AWS/ANOMALY_DETECTION_TRAINING` `computation-model-id` Ersetzen Sie es durch die ID, die Sie bei der Erstellung Ihres Berechnungsmodells zurückgegeben haben.

```
aws iotsitewise describe-computation-model \  
  --computation-model-id computation-model-id
```

2. Erstellen Sie eine Datei mit dem Namen `anomaly-detection-start-retraining-payload.json` und fügen Sie den folgenden Code hinzu. Ersetzen Sie die Parameter wie beschrieben durch Werte.

Note

Die Nutzlast muss den Anforderungen entsprechen. [Option 1: Verwenden Sie eine saubere Payload-Datei](#)

- a. `lookbackWindow` mit dem historischen Datenfenster, das für die Umschulung verwendet werden soll (P180D//P360D/P540D). P720D
- b. `retrainingFrequency` mit der Häufigkeit, mit der das Modell neu trainiert werden soll (MinimumP30D, MaximumP1Y).
- c. (Optional) `promotion` mit dem Model-Promotion-Modus (SERVICE_MANAGED oder CUSTOMER_MANAGED). Der Standardwert ist SERVICE_MANAGED.
- d. (Optional) `retrainingStartDate` mit dem Startdatum für den Umschulungsplan, angegeben in Epochensekunden. Kürzt die Zeit auf den nächsten UTC-Tag. Optional, standardmäßig das aktuelle Datum.
- e. Sie können optional konfigurieren [Fortgeschrittene Trainingskonfigurationen](#), um die Modellleistung zu verbessern.
 - (Optional) `ModelMetricsDestination`, um umfassende Leistungsdaten (Präzision, Recall, Area Under the Curve) zu erhalten.

```
{
  "trainingMode": "START_RETRAINING_SCHEDULER",
  "retrainingConfiguration": {
    "lookbackWindow": "P180D",
    "promotion": "SERVICE_MANAGED",
    "retrainingFrequency": "P30D",
    "retrainingStartDate": "StartDate"
  }
}
```

3. Führen Sie den folgenden Befehl aus, um den Retrainingplaner zu starten. Ersetzen Sie die folgenden Parameter im Befehl:
 - a. `computation-model-id` mit der ID des Zielberechnungsmodells.
 - b. `asset-id` mit der ID des Assets, anhand dessen Sie das Modell trainieren werden.
 - c. `training-action-definition-id` mit der ID der AWS/ANOMALY_DETECTION_TRAINING Aktion aus Schritt 1.

```
aws iotsitewise execute-action \
  --target-resource computationModelId=computation-model-id \
```



```
--resolve-to assetId=asset-id \
--action-definition-id training-action-definition-id \
--action-payload stringValue@=file://anomaly-detection-start-retraining-
payload.json
```

Example Beispiel für den Befehl „Aktion ausführen“

```
aws iotsitewise execute-action --target-resource computationModelId=27cb824c-
fd84-45b0-946b-0a5b0466d890 --resolve-to assetId=cefd4b68-481b-4735-
b466-6a4220cd19ee --action-definition-id e54cea94-5d1c-4230-a59e-4f54dcbc972d --
action-payload stringValue@=file://anomaly-detection-start-retraining-payload.json
```

Beenden Sie das erneute Training des Schedulers

1. Führen Sie den folgenden Befehl aus, um die Aktion `actionDefinitionId` zu ermitteln. `AWS/ANOMALY_DETECTION_TRAINING` `computation-model-id` Ersetzen Sie es durch die tatsächliche ID des zuvor erstellten Berechnungsmodells.

```
aws iotsitewise describe-computation-model \
--computation-model-id computation-model-id
```

2. Erstellen Sie eine Datei `anomaly-detection-stop-retraining-payload.json` und fügen Sie Folgendes hinzu:

Note

Die Nutzlast muss den Anforderungen entsprechen. [Option 1: Verwenden Sie eine saubere Payload-Datei](#)

```
{
  "trainingMode": "STOP_RETRAINING_SCHEDULER"
}
```

3. Führen Sie den folgenden Befehl aus, um den Retrainingplaner zu beenden. Ersetzen Sie die folgenden Parameter im Befehl:
 - a. `computation-model-id` mit der ID des Zielberechnungsmodells.
 - b. `asset-id` mit der ID des Assets, anhand dessen Sie das Modell trainieren werden.

- c. `training-action-definition-id` mit der ID der AWS/ANOMALY_DETECTION_TRAINING Aktion aus Schritt 1.

```
aws iotsitewise execute-action \
  --target-resource computationModelId=computation-model-id \
  --resolve-to assetId=asset-id \
  --action-definition-id training-action-definition-id \
  --action-payload stringValue@=file://anomaly-detection-stop-retraining-
payload.json
```

Inferenz starten und beenden ()AWS CLI

Starten Sie nach dem Training des Modells die Inferenz. Dies weist Sie an AWS IoT SiteWise, Ihre Industrieanlagen aktiv auf Anomalien zu überwachen.

Inferenz starten

1. Führen Sie den folgenden Befehl aus, um die AWS/ANOMALY_DETECTION_INFERENCE Aktion `actionDefinitionId` zu ermitteln. `computation-model-id` Ersetzen Sie es durch die tatsächliche ID des zuvor erstellten Berechnungsmodells.

```
aws iotsitewise describe-computation-model \
  --computation-model-id computation-model-id
```

2. Erstellen Sie eine Datei `anomaly-detection-start-inference-payload.json` und fügen Sie die folgenden Werte hinzu:

Note

Die Nutzlast muss den Anforderungen entsprechen. [Option 1: Verwenden Sie eine saubere Payload-Datei](#)

```
"inferenceMode": "START",
"dataUploadFrequency": "DataUploadFrequency"
```

- a. `DataUploadFrequency`: Konfigurieren Sie die Häufigkeit, mit der der Inferenzplan ausgeführt wird, um Anomalien zu erkennen. Zulässige Werte sind: PT5M, PT10M, PT15M, PT30M, PT1H, PT2H..PT12H, PT1D
 - b. (Optional) `DataDelayOffsetInMinutes` mit dem Verzögerungsoffset in Minuten. Stellen Sie diesen Wert zwischen 0 und 60 Minuten ein.
 - c. (Optional) `TargetModelVersion` mit der zu aktivierenden Modellversion.
 - d. (Optional) Konfigurieren Sie das `weeklyOperatingWindow` mit einer Schichtkonfiguration.
 - e. Sie können optional konfigurieren [Erweiterte Inferenzkonfigurationen](#).
 - i. [Hochfrequente Inferenzierung \(5 Minuten — 1 Stunde\)](#).
 - ii. [Niederfrequenz-Inferenzierung \(2 Stunden — 1 Tag\)](#).
 - iii. [Flexible Planung](#).
3. Führen Sie den folgenden Befehl aus, um die Inferenz zu starten. Ersetzen Sie die folgenden Parameter in der Payload-Datei.
- a. `computation-model-id` mit der ID des Zielberechnungsmodells.
 - b. `asset-id` mit der ID des Assets, für das das Modell trainiert wurde.
 - c. `inference-action-definition-id` mit der ID der AWS/ANOMALY_DETECTION_INFERENCE Aktion aus Schritt 1.

```
aws iotsitewise execute-action \
  --target-resource computationModelId=computation-model-id \
  --resolve-to assetId=asset-id \
  --action-definition-id inference-action-definition-id \
  --action-payload stringValue@=file://anomaly-detection-inference-payload.json
```

4. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob Inference noch läuft. Das `inferenceTimerActive` Feld ist auf „TRUE“ Wenn Inferenz aktiv“ gesetzt ist.

```
aws iotsitewise describe-computation-model-execution-summary \
  --computation-model-id computation-model-id \
  --resolve-to-resource-type ASSET \
  --resolve-to-resource-id asset-id
```

5. Der folgende Befehl listet alle Inferenzausführungen auf:

```
aws iotsitewise list-executions \  
  --target-resource-type COMPUTATION_MODEL \  
  --target-resource-id computation-model-id \  
  --resolve-to-resource-type ASSET \  
  --resolve-to-resource-id asset-id
```

6. Führen Sie den folgenden Befehl aus, um eine einzelne Ausführung zu beschreiben.
`execution-id` Ersetzen Sie es durch die ID aus dem vorherigen Schritt 5.

```
aws iotsitewise describe-execution \  
  --execution-id execution-id
```

Beenden Sie die Inferenz

1. Führen Sie den folgenden Befehl aus, um die AWS/ANOMALY_DETECTION_INFERENCE Aktion `actionDefinitionId` zu ermitteln. `computation-model-id` Ersetzen Sie es durch die tatsächliche ID des zuvor erstellten Berechnungsmodells.

```
aws iotsitewise describe-computation-model \  
  --computation-model-id computation-model-id
```

2. Erstellen Sie eine Datei `anomaly-detection-stop-inference-payload.json` und fügen Sie den folgenden Code hinzu.

```
{  
  "inferenceMode": "STOP"  
}
```

Note

Die Nutzlast muss den Anforderungen entsprechen. [Option 1: Verwenden Sie eine saubere Payload-Datei](#)

3. Führen Sie den folgenden Befehl aus, um die Inferenz zu beenden. Ersetzen Sie den folgenden Parameter in der Payload-Datei:
 - a. `computation-model-id` mit der ID des Zielberechnungsmodells.
 - b. `asset-id` mit der ID des Assets, für das das Modell trainiert wurde.

- c. `inference-action-definition-id` mit der ID der AWS/ANOMALY_DETECTION_INFERENCE Aktion aus Schritt 1.

Example des Befehls Stop Inference:

```
aws iotsitewise execute-action \
  --target-resource computationModelId=computation-model-id \
  --resolve-to assetId=asset-id \
  --action-definition-id inference-action-definition-id \
  --action-payload stringValue@=file://anomaly-detection-stop-inference-
  payload.json
```

Findet Berechnungsmodelle, die eine bestimmte Ressource bei der Datenbindung verwenden

Um Berechnungsmodelle aufzulisten, die an eine bestimmte Ressource gebunden sind:

- Asset-Modell (ruft alle Berechnungsmodelle ab, an die Eigenschaften dieses Asset-Modells gebunden sind).
- Asset (ruft alle Berechnungsmodelle ab, bei denen die Eigenschaften dieses Assets gebunden sind)
- Eigenschaft des Asset-Modells (ruft alle Berechnungsmodelle ab, an die diese Eigenschaft gebunden ist)
- Anlageneigenschaft (ruft alle Berechnungsmodelle ab, an die diese Eigenschaft gebunden ist). Dies kann zu Informationszwecken dienen oder erforderlich sein, wenn der Benutzer versucht, diese Eigenschaft an ein anderes Berechnungsmodell zu binden, sie aber bereits an eine andere Stelle gebunden ist)

Verwenden Sie die [ListComputationModelDataBindingUsages](#) API, um eine Liste von `ComputationModelId`s abzurufen, die das Asset (Property) oder das Asset-Modell (Property) als Datenbindung verwenden.

Bereiten Sie eine `request.json` mit den folgenden Informationen vor:

```
{
  "dataBindingValueFilter": {
```

```

    "asset": {
      "assetId": "<string>"
    }
    // OR
    "assetModel": {
      "assetModelId": "<string>"
    }
    // OR
    "assetProperty": {
      "assetId": "<string>",
      "propertyId": "<string>"
    }
    // OR
    "assetModelProperty": {
      "assetModelId": "<string>",
      "propertyId": "<string>"
    }
  },
  "nextToken": "<string>",
  "maxResults": "<number>"
}

```

Verwenden Sie den `list-computation-model-data-binding-usages` Befehl, um die Modelle mit Assets oder Asset-Modellen als Datenbindungen abzurufen.

```

aws iotsitewise list-computation-model-data-binding-usages \
--cli-input-json file://request.json

```

Aktivieren Sie die Erkennung von Anomalien auf Sensoren in allen Anlagen

Erstellen Sie ein Berechnungsmodell (AWS CLI)

Verwenden Sie die AWS Befehlszeilenschnittstelle (AWS CLI), um ein Berechnungsmodell zu erstellen. Nachdem Sie das Berechnungsmodell definiert haben, trainieren Sie das Modell und planen Sie die Inferenz so ein, dass die Anomalieerkennung in allen Anlagen erfolgt. AWS IoT SiteWise

In den folgenden Schritten wird dieser Prozess erklärt:

1. Um die Erkennung von Anomalien einzurichten, verwenden Sie die [UpdateAssetModel\(AWS CLI\)](#) und erfüllen Sie die folgenden Anforderungen:

- a. Mindestens eine Eingabeeigenschaft, die entweder vom INTEGER Datentyp DOUBLE oder ist. Sie ist entweder eine Mess- oder eine Transformationseigenschaft und wird zum Trainieren des Modells verwendet.
 - b. Eine Ergebniseigenschaft vom STRING Datentyp. Es muss sich um eine Messeigenschaft handeln und speichert die Ergebnisse der Anomalieerkennung.
2. Erstellen Sie eine Datei `anomaly-detection-computation-model-payload.json` mit dem folgenden Inhalt:

 Note

Erstellen Sie ein Berechnungsmodell, indem Sie es direkt `assetProperty` als Datenquelle angeben.

```
{
  "computationModelName": "name of ComputationModel",
  "computationModelConfiguration": {
    "anomalyDetection": {
      "inputProperties": "${properties}",
      "resultProperty": "${p3}"
    }
  },
  "computationModelDataBinding": {
    "properties": {
      "list": [
        {
          "assetProperty": {
            "assetId": "asset-id",
            "propertyId": "input-property-id-1"
          }
        },
        {
          "assetProperty": {
            "assetId": "asset-id",
            "propertyId": "input-property-id-2"
          }
        }
      ]
    },
    "p3": {
```

```

        "assetProperty": {
            "assetId": "asset-id",
            "propertyId": "results-property-id"
        }
    }
}

```

3. Führen Sie den folgenden Befehl aus, um ein Berechnungsmodell zu erstellen:

```

aws iotsitewise create-computation-model \
    --cli-input-json file://anomaly-detection-computation-model-payload.json

```

ExecuteAction Vorbereitung der API-Nutzlast

Die nächsten Schritte zur Durchführung von Training und Inferenz werden mit der [ExecuteAction](#) API durchgeführt. Sowohl Training als auch Inferenz werden mit einer JSON-Aktionsnutzlastkonfiguration konfiguriert. Beim Aufrufen der [ExecuteAction](#) API muss die Aktionsnutzlast als Wert zusammen mit einer Nutzlast bereitgestellt werden. stringValue

Die Payload muss sich strikt an die API-Anforderungen halten. Insbesondere muss der Wert eine flache Zeichenfolge ohne Steuerzeichen sein (z. B. Zeilenumbrüche, Tabulatoren oder Zeilenumbrüche). Die folgenden Optionen bieten zwei zuverlässige Methoden, um eine gültige Aktions-Payload bereitzustellen.

Option 1: Verwenden Sie eine saubere Payload-Datei

Das folgende Verfahren beschreibt die Schritte für eine saubere Payload-Datei:

1. Säubern Sie die Datei, um Steuerzeichen zu entfernen.

```

tr -d '\n\r\t' < original-action-payload.json > training-or-inference-action-payload.json

```

2. Führen Sie die Aktion mit der Datei `stringValue@=file:///...`

```

aws iotsitewise execute-action \
    --target-resource computationModelId=<MODEL_ID> \
    --action-definition-id <ACTION_DEFINITION_ID> \
    --action-payload stringValue@=file:///training-or-inference-action-payload.json

```


Option 2: Inline-Zeichenfolge mit maskierten Anführungszeichen

In den folgenden Schritten werden die Schritte beschrieben, um die Nutzdaten inline bereitzustellen und Zwischendateien zu vermeiden:

- Verwenden Sie maskierte Anführungszeichen (\") innerhalb der JSON-Zeichenfolge.
- Setzen Sie den gesamten StringValue=. . Ausdruck in doppelte Anführungszeichen.

Example einer Payload einer Aktion, der entgangen ist:

```
aws iotsitewise execute-action \  
  --target-resource computationModelId=<MODEL_ID> \  
  --action-definition-id <ACTION_DEFINITION_ID> \  
  --action-payload "stringValue={\"exportDataStartTime\":1717225200,  
\"exportDataEndTime\":1722789360,\"targetSamplingRate\": \"PT1M\"}"
```

Trainiere das Modell (AWS CLI)

1. Führen Sie den folgenden Befehl aus, um die AWS/ANOMALY_DETECTION_TRAINING Aktion actionDefinitionId zu ermitteln. computation-model-id Ersetzen Sie es durch die im vorherigen Schritt zurückgegebene ID.

```
aws iotsitewise describe-computation-model \  
  --computation-model-id computation-model-id
```

2. Erstellen Sie eine Datei mit dem Namen anomaly-detection-training-payload.json und fügen Sie die folgenden Werte hinzu:

Note

Die Nutzlast muss den Anforderungen entsprechen. [Option 1: Verwenden Sie eine saubere Payload-Datei](#)

- a. StartTime mit dem Beginn der Trainingsdaten, angegeben in Epochensekunden.
- b. EndTime mit dem Ende der Trainingsdaten, angegeben in Epochensekunden.
- c. Sie können optional konfigurieren [Erweiterte Inferenzkonfigurationen](#).

- i. (Optional) `TargetSamplingRate` mit der Abtastrate der Daten.
- ii. (Optional) `LabelInputConfiguration` zur Angabe von Zeiträumen, in denen anomales Verhalten auftrat, um das Modelltraining zu verbessern.
- iii. (Optional) `ModelEvaluationConfiguration` um die Leistung des Modells zu bewerten, indem nach Abschluss des Trainings Inferenzen für einen bestimmten Zeitraum ausgeführt werden.

```
{
  "exportDataStartTime": StartTime,
  "exportDataEndTime": EndTime
}
```

Example eines Beispiels für eine Trainingsnutzlast:

```
{
  "exportDataStartTime": 1717225200,
  "exportDataEndTime": 1722789360
}
```

3. Führen Sie den folgenden Befehl aus, um das Training zu starten (ohne die Ressource als Zielressource bereitzustellen). Ersetzen Sie die folgenden Parameter im Befehl:

```
aws iotsitewise execute-action \
  --target-resource computationModelId=computation-model-id \
  --action-definition-id training-action-definition-id \
  --action-payload stringValue@=file://anomaly-detection-training-payload.json
```

4. Führen Sie den folgenden Befehl aus, um den Status des Modelltrainingsprozesses zu überprüfen. Die letzte Ausführungsübersicht zeigt den Ausführungsstatus (RUNNING/COMPLETED/FAILED).

```
aws iotsitewise list-executions \
  --target-resource-type COMPUTATION_MODEL \
  --target-resource-id computation-model-id
```

5. Führen Sie den folgenden Befehl aus, um die Konfiguration des zuletzt trainierten Modells zu überprüfen. Dieser Befehl erzeugt nur dann eine Ausgabe, wenn mindestens ein Modell das Training erfolgreich abgeschlossen hat.

```
aws iotsitewise describe-computation-model-execution-summary \  
  --computation-model-id computation-model-id
```

Startet und beendet das erneute Trainieren des Modells (AWS CLI)

Nach dem ersten Modelltraining können Sie automatisches Retraining konfigurieren, um Datenabweichungen entgegenzuwirken und die Modellgenauigkeit im Laufe der Zeit aufrechtzuerhalten. Mit dem Retrainingplaner können Sie regelmäßige Modellaktualisierungen mit konfigurierbaren Promotion-Modi einrichten.

Starten Sie den Retrainingplaner

1. Bereiten Sie dieselbe Nutzlast wie unter beschrieben vor. [Starten Sie den Retrainingplaner](#)
2. Führen Sie die Trainingsaktion aus (ohne die Ressource als Zielressource anzugeben). Ersetzen Sie die folgenden Parameter im Befehl:
 - a. `computation-model-id` mit der ID des Zielberechnungsmodells.
 - b. `training-action-definition-id` mit der ID der AWS/ANOMALY_DETECTION_TRAINING Aktion.

```
aws iotsitewise execute-action \  
  --target-resource computationModelId=computation-model-id \  
  --action-definition-id training-action-definition-id \  
  --action-payload stringValue@=file://anomaly-detection-start-retraining-  
  payload.json
```

3. Führen Sie den folgenden Befehl aus, um den Status des Prozesses Start Retraining Scheduler zu überprüfen. Die letzte Ausführungsübersicht zeigt den Ausführungsstatus (RUNNING/COMPLETED/FAILED).

```
aws iotsitewise list-executions \  
  --target-resource-type COMPUTATION_MODEL \  
  --target-resource-id computation-model-id
```

Beenden Sie das erneute Training des Schedulers

1. Bereiten Sie dieselbe Nutzlast wie unter beschrieben vor. [Beenden Sie das erneute Training des Schedulers](#)
2. Führen Sie die Trainingsaktion aus (ohne die Ressource als Zielressource anzugeben). Ersetzen Sie die folgenden Parameter im Befehl:
 - a. `computation-model-id` mit der ID des Zielberechnungsmodells.
 - b. `training-action-definition-id` mit der ID der AWS/ANOMALY_DETECTION_TRAINING Aktion.

```
aws iotsitewise execute-action \  
  --target-resource computationModelId=computation-model-id \  
  --action-definition-id training-action-definition-id \  
  --action-payload stringValue@=file://anomaly-detection-stop-retraining-  
payload.json
```

3. Führen Sie den folgenden Befehl aus, um den Status des Scheduler-Prozesses zum Beenden des Retrainings zu überprüfen. In der Zusammenfassung der letzten Ausführung wird der Ausführungsstatus (RUNNING/COMPLETED/FAILED) angezeigt.

```
aws iotsitewise list-executions \  
  --target-resource-type COMPUTATION_MODEL \  
  --target-resource-id computation-model-id
```

Inferenz starten und beenden ()AWS CLI

Nachdem Sie das Modell trainiert haben, starten Sie die Inferenz, die Sie anweist, mit der Überwachung Ihrer Industrieanlagen auf Anomalien AWS IoT SiteWise zu beginnen.

Starten Sie die Inferenz

1. Führen Sie den folgenden Befehl aus, um die AWS/ANOMALY_DETECTION_INFERENCE Aktion `actionDefinitionId` zu ermitteln. `computation-model-id` Ersetzen Sie es durch die tatsächliche ID des zuvor erstellten Berechnungsmodells.

```
aws iotsitewise describe-computation-model \  
  --computation-model-id computation-model-id
```

```
--computation-model-id computation-model-id
```

2. Erstellen Sie eine Datei `anomaly-detection-start-inference-payload.json` und fügen Sie den folgenden Code hinzu. Ersetzen Sie die folgenden Parameter wie beschrieben:

 Note

Die Nutzlast muss den Anforderungen entsprechen. [Option 1: Verwenden Sie eine saubere Payload-Datei](#)

- a. `DataUploadFrequency`: Konfigurieren Sie die Häufigkeit, mit der der Inferenzplan ausgeführt wird, um Anomalien zu erkennen. Zulässige Werte sind: PT5M, PT10M, PT15M, PT30M, PT1H, PT2H..PT12H, PT1D

```
"inferenceMode": "START",
"dataUploadFrequency": "DataUploadFrequency"
```

- b. (Optional) `DataDelayOffsetInMinutes` mit dem Verzögerungsoffset in Minuten. Stellen Sie diesen Wert zwischen 0 und 60 Minuten ein.
 - c. (Optional) `TargetModelVersion` mit der zu aktivierenden Modellversion.
 - d. (Optional) Konfigurieren Sie das `weeklyOperatingWindow` mit einer Schichtkonfiguration.
 - e. Sie können optional konfigurieren [Erweiterte Inferenzkonfigurationen](#).
 - i. [Hochfrequente Inferenzierung \(5 Minuten — 1 Stunde\)](#).
 - ii. [Niederfrequenz-Inferenzierung \(2 Stunden — 1 Tag\)](#).
 - iii. [Flexible Planung](#).
3. Führen Sie den folgenden Befehl aus, um die Inferenz zu starten. Ersetzen Sie die folgenden Parameter in der Payload-Datei.
 - a. `computation-model-id` mit der ID des Zielberechnungsmodells.
 - b. `inference-action-definition-id` mit der ID der AWS/ANOMALY_DETECTION_INFERENCE Aktion aus Schritt 1.

```
aws iotsitewise execute-action \
  --target-resource computationModelId=computation-model-id \
  --action-definition-id inference-action-definition-id \
```

```
--action-payload stringValue@=file://anomaly-detection-inference-payload.json
```

4. Führen Sie den folgenden Befehl aus, um zu überprüfen, ob Inference noch läuft. Das `inferenceTimerActive` Feld ist auf „TRUE“ gesetzt, wenn Inferenz aktiv ist.

```
aws iotsitewise describe-computation-model-execution-summary \  
  --computation-model-id computation-model-id
```

5. Der folgende Befehl listet alle Inferenzausführungen auf:

```
aws iotsitewise list-executions \  
  --target-resource-type COMPUTATION_MODEL \  
  --target-resource-id computation-model-id
```

6. Führen Sie den folgenden Befehl aus, um eine einzelne Ausführung zu beschreiben. `execution-id` Ersetzen Sie es durch die ID aus dem vorherigen Schritt 5.

```
aws iotsitewise describe-execution \  
  --execution-id execution-id
```

Beenden Sie die Inferenz

1. Führen Sie den folgenden Befehl aus, um die AWS/ANOMALY_DETECTION_INFERENCE Aktion `actionDefinitionId` zu ermitteln. `computation-model-id` Ersetzen Sie es durch die tatsächliche ID des zuvor erstellten Berechnungsmodells.

```
aws iotsitewise describe-computation-model \  
  --computation-model-id computation-model-id
```

2. Erstellen Sie eine Datei `anomaly-detection-stop-inference-payload.json` und fügen Sie den folgenden Code hinzu.

```
{  
  "inferenceMode": "STOP"  
}
```

Note

Die Nutzlast muss den Anforderungen entsprechen. [Option 1: Verwenden Sie eine saubere Payload-Datei](#)

3. Führen Sie den folgenden Befehl aus, um die Inferenz zu beenden. Ersetzen Sie den folgenden Parameter in der Payload-Datei:
 - a. `computation-model-id` mit der ID des Zielberechnungsmodells.
 - b. `inference-action-definition-id` mit der ID der AWS/ANOMALY_DETECTION_INFERENCE Aktion aus Schritt 1.

Example des Befehls Stop Inference:

```
aws iotsitewise execute-action \  
--target-resource computationModelId=computation-model-id \  
--action-definition-id inference-action-definition-id \  
--action-payload stringValue@=file://anomaly-detection-stop-inference-payload.json
```

Fortgeschrittene Trainingskonfigurationen

Konfiguration der Abtastrate

Die Abtastrate definiert, wie häufig Sensormesswerte aufgezeichnet werden (z. B. einmal pro Sekunde oder einmal pro Minute). Diese Einstellung wirkt sich direkt auf die Granularität der Trainingsdaten aus und beeinflusst die Fähigkeit des Modells, kurzfristige Schwankungen des Sensorverhaltens zu erfassen.

Besuchen Sie uns [Stichprobenerhebung für hochfrequente Daten und Konsistenz zwischen Training und Inferenz](#), um mehr über bewährte Verfahren zu erfahren.

Konfigurieren Sie die Ziel-Abtastrate

Sie können `TargetSamplingRate` in Ihrer Trainingskonfiguration optional angeben, um die Häufigkeit zu steuern, mit der Daten abgetastet werden. Unterstützte Werte sind:

PT1S | PT5S | PT10S | PT15S | PT30S | PT1M | PT5M | PT10M | PT15M | PT30M | PT1H

Dies sind ISO 8601-Dauerformate, die die folgenden Zeitformate darstellen:

- PT1S= 1 Sekunde
- PT1M= 1 Minute
- PT1H= 1 Stunde

Wählen Sie eine Abtastrate, die das richtige Gleichgewicht zwischen Datenauflösung und Trainingseffizienz bietet. Die folgenden Raten sind verfügbar:

- Höhere Abtastraten (PT1S) bieten feinere Details, können jedoch das Datenvolumen und die Trainingszeit erhöhen.
- Niedrigere Abtastraten (PT10M,PT1H) reduzieren die Datengröße und die Kosten, können jedoch kurzlebige Anomalien übersehen.

Behandlung von Zeitstempelfehlern

AWS IoT SiteWise kompensiert automatisch die falsche Ausrichtung von Zeitstempeln in mehreren Datenströmen während des Trainings. Dies gewährleistet ein konsistentes Modellverhalten, auch wenn die Eingangssignale zeitlich nicht perfekt aufeinander abgestimmt sind.

Besuchen Sie uns [Stichprobenerhebung für hochfrequente Daten und Konsistenz zwischen Training und Inferenz](#), um mehr über bewährte Verfahren zu erfahren.

Sampling aktivieren

Fügen Sie den folgenden Code zu `hinzuanomaly-detection-training-payload.json`.

Konfigurieren Sie das Sampling, indem `TargetSamplingRate` Sie die Payload für die Trainingsaktion mit der Sampling-Rate der Daten hinzufügen. Die zulässigen Werte sind: PT1S | PT5S | PT10S | PT15S | PT30S | PT1M | PT5M | PT10M | PT15M | PT30M | PT1H.

```
{
  "exportDataStartTime": StartTime,
  "exportDataEndTime": EndTime,
  "targetSamplingRate": "TargetSamplingRate"
}
```


Example einer Konfiguration mit Samplerate:

```
{
  "exportDataStartTime": 1717225200,
  "exportDataEndTime": 1722789360,
  "targetSamplingRate": "PT1M"
}
```

Kennzeichnen Sie Ihre Daten

Bei der Kennzeichnung Ihrer Daten müssen Sie Zeitintervalle definieren, die Perioden mit abnormalem Geräteverhalten darstellen. Diese Kennzeichnungsinformationen werden als CSV Datei bereitgestellt, wobei jede Zeile einen Zeitraum angibt, in dem die Geräte nicht ordnungsgemäß funktionierten.

Jede Zeile enthält zwei Zeitstempel:

- Die Startzeit, die angibt, wann vermutlich abnormales Verhalten begonnen hat.
- Die Endzeit, die angibt, wann der Fehler oder das Problem zum ersten Mal beobachtet wurde.

Diese CSV-Datei wird in einem Amazon S3 S3-Bucket gespeichert und wird beim Modelltraining verwendet, um dem System zu helfen, aus bekannten Beispielen für abnormales Verhalten zu lernen. Das folgende Beispiel zeigt, wie Ihre Etikettendaten als .csv Datei angezeigt werden sollten. Die Datei hat keinen Header.

Example einer CSV-Datei:

```
2024-06-21T00:00:00.000000,2024-06-21T12:00:00.000000
2024-07-11T00:00:00.000000,2024-07-11T12:00:00.000000
2024-07-31T00:00:00.000000,2024-07-31T12:00:00.000000
```

Zeile 1 steht für ein Wartungsereignis am 21. Juni 2024 mit einem 12-Stunden-Fenster (von bis2024-06-21T12:00:00.000000Z), in dem 2024-06-21T00:00:00.000000Z nach abnormalem Verhalten gesucht werden AWS IoT SiteWise kann.

Zeile 2 steht für ein Wartungsereignis am 11. Juli 2024 mit einem 12-Stunden-Fenster (von bis2024-07-11T12:00:00.000000Z), in dem 2024-07-11T00:00:00.000000Z nach abnormalem AWS IoT SiteWise Verhalten gesucht werden kann.

Zeile 3 steht für ein Wartungsereignis am 31. Juli 2024 mit einem 12-Stunden-Fenster (von bis2024-07-31T12:00:00.000000Z), in dem 2024-07-31T00:00:00.000000Z nach abnormalem AWS IoT SiteWise Verhalten gesucht werden kann.

AWS IoT SiteWise verwendet all diese Zeitfenster, um Modelle zu trainieren und zu bewerten, mit denen abnormales Verhalten im Zusammenhang mit diesen Ereignissen erkannt werden kann. Beachten Sie, dass nicht alle Ereignisse nachweisbar sind und die Ergebnisse stark von der Qualität und den Eigenschaften der zugrunde liegenden Daten abhängen.

Einzelheiten zu bewährten Methoden für die Probenahme finden Sie unter [Best Practices](#).

Schritte zur Datenkennzeichnung

- Konfigurieren Sie Ihren Amazon S3 S3-Bucket gemäß den Kennzeichnungsvoraussetzungen unter Voraussetzungen [für Kennzeichnungsdaten](#).
- Laden Sie die Datei in Ihren Labeling-Bucket hoch.
- Fügen Sie Folgendes hinzu `anomaly-detection-training-payload.json`.
 - Geben Sie die Speicherorte im `labelInputConfiguration` Abschnitt der Datei an. `labels-bucket` Ersetzen Sie es durch den Bucket-Namen und `files-prefix` durch den Pfad der Datei (en) oder einen beliebigen Teil des Präfixes. Alle Dateien am Speicherort werden analysiert und (bei Erfolg) als Labeldateien verwendet.

```
{
  "exportDataStartTime": StartTime,
  "exportDataEndTime": EndTime,
  "labelInputConfiguration":
    {
      "bucketName": "label-bucket",
      "prefix": "files-prefix"
    }
}
```

Example einer Labelkonfiguration:

```
{
  "exportDataStartTime": 1717225200,
  "exportDataEndTime": 1722789360,
  "labelInputConfiguration": {
    "bucketName": "anomaly-detection-customer-data-278129555252-iad",
```

```

    "prefix": "Labels/model=b2d8ab3e-73af-48d8-9b8f-a290bef931b4/
asset[d3347728-4796-4c5c-afdb-ea2f551ffe7a]/Lables.csv"
  }
}

```

Bewerten Ihres Modells

Die punktuelle Modelldiagnostik für ein AWS IoT SiteWise Trainingsmodell ist eine Bewertung der Modellleistung bei den einzelnen Ereignissen. AWS IoT SiteWise Generiert während des Trainings einen Anomalie-Score und eine Diagnose des Sensorbeitrags für jede Zeile im Eingabedatensatz. Ein höherer Anomaliewert weist auf eine höhere Wahrscheinlichkeit eines abnormalen Ereignisses hin.

Punktgenaue Diagnosen sind verfügbar, wenn Sie ein Modell mit [ExecuteAction](#) API und AWS/ANOMALY_DETECTION_TRAINING Aktionstyp trainieren.

Um die Modellevaluierung zu konfigurieren,

- Konfigurieren Sie Ihren Amazon S3 S3-Bucket gemäß den Kennzeichnungsvoraussetzungen unter Voraussetzungen [für Kennzeichnungsdaten](#).
- Fügen Sie Folgendes hinzu `anomaly-detection-training-payload.json`.
 - Geben Sie die `evaluationStartTime` Werte und `evaluationEndTime` (beide in Epochensekunden) für die Daten in dem Fenster an, das zur Bewertung der Leistung des Modells verwendet wird.
 - Geben Sie den Amazon S3 S3-Bucket-Standort (`resultDestination`) an, an den die Bewertungsdiagnose geschrieben werden soll.

Note

Das Modellevaluierungsintervall (`dataStartTime` bis `dataEndTime`) muss sich entweder mit dem Trainingsintervall überschneiden oder an dieses angrenzen. Lücken sind nicht zulässig.

```

{
  "exportDataStartTime": StartTime,
  "exportDataEndTime": EndTime,
  "modelEvaluationConfiguration": {
    "dataStartTime": evaluationStartTime,
    "dataEndTime": evaluationEndTime
  }
}

```

```
"resultDestination": {
  "bucketName": "s3BucketName",
  "prefix": "bucketPrefix"
}
}
```

Example einer Konfiguration zur Modellevaluierung:

```
{
  "exportDataStartTime": 1717225200,
  "exportDataEndTime": 1722789360,
  "modelEvaluationConfiguration": {
    "dataStartTime": 1722789360,
    "dataEndTime": 1725174000,
    "resultDestination": {
      "bucketName": "anomaly-detection-customer-data-278129555252-iad",
      "prefix": "Evaluation/asset[d3347728-4796-4c5c-afdb-ea2f551ffe7a]/1747681026-
evaluation_results.jsonl"
    }
  }
}
```

Generieren Sie Modellmetriken

Modellmetriken bieten umfassende Einblicke in die Leistung und Qualität Ihrer trainierten Modelle zur Erkennung von Anomalien. Der Schulungsprozess generiert diese Metriken automatisch und veröffentlicht sie in Ihrem angegebenen Amazon S3 S3-Bucket, sodass sie für Analysen, Modellvergleiche und Beförderungsentscheidungen in Umschulungsworkflows leicht zugänglich sind.

Modellmetriken verstehen

Der Trainingsprozess generiert automatisch Modellmetriken und bietet detaillierte Informationen zu:

- **Modellleistung:** Quantitative Messgrößen wie Präzision, Erinnerungsvermögen und AUC, sofern beschriftete Daten verfügbar sind
- **Datenqualität:** Informationen über die verwendeten Trainingsdaten und die erfassten Zeiträume
- **Erkennung von Ereignissen:** Statistiken über identifizierte Anomalien und markierte Ereignisse

- Modellvergleich: Vergleichsmetriken zwischen verschiedenen Modellversionen während der Umschulung

Konfigurieren Sie das Ziel für Modellmetriken

Um die Generierung von Modellmetriken zu aktivieren, konfigurieren Sie ein Amazon S3 S3-Ziel, in dem die Metriken veröffentlicht werden.

1. Konfigurieren Sie Ihren Amazon S3 S3-Bucket gemäß der [Voraussetzungen für die Modellevaluierung](#).
2. Fügen Sie Ihrer Payload für Trainingsaktionen Folgendes hinzu, um anzugeben, wo Modellmetriken gespeichert werden sollen:

```
{
  "trainingMode": "TRAIN_MODEL",
  "exportDataStartTime": StartTime,
  "exportDataEndTime": EndTime,
  "modelMetricsDestination": {
    "bucketName": "bucket-name",
    "prefix": "prefix"
  }
}
```

Example Beispiel für die Konfiguration von Modellmetriken

```
{
  "exportDataStartTime": 1717225200,
  "exportDataEndTime": 1722789360,
  "modelMetricsDestination": {
    "bucketName": "anomaly-detection-metrics-bucket-123456789012-iad",
    "prefix": "ModelMetrics/computation-model-id/asset-id/training-  
metrics.json"
  }
}
```

Konfigurieren Sie Modellmetriken für die Umschulung

Wenn Sie Umschulungspläne einrichten, müssen Sie das Ziel der Modellmetriken angeben, um eine umfassende Nachverfolgung und einen Vergleich der Modellleistung zu ermöglichen:

```
{
  "trainingMode": "START_RETRAINING_SCHEDULER",
  "modelMetricsDestination": {
    "bucketName": "bucket-name",
    "prefix": "prefix"
  },
  "retrainingConfiguration": {
    "lookbackWindow": "P180D",
    "promotion": "SERVICE_MANAGED",
    "retrainingFrequency": "P30D",
    "retrainingStartDate": "StartDate"
  }
}
```

Parameters

bucketName

Amazon S3 S3-Bucket, in dem Modellmetriken gespeichert werden

prefix

Amazon S3 prefix/path zum Organisieren von Modellmetrikdateien

Struktur der Modellmetriken

Modellmetriken werden als JSON-Dateien in Ihrem Amazon S3 S3-Bucket in der folgenden Struktur gespeichert:

```
{
  "labeled_ranges": [],
  "labeled_event_metrics": {
    "num_labeled": 0,
    "num_identified": 0,
    "total_warning_time_in_seconds": 0
  },
  "predicted_ranges": [],
  "unknown_event_metrics": {
    "num_identified": 0,
    "total_duration_in_seconds": 0
  },
  "data_start_time": "2023-11-01",
```

```
"data_end_time": "2023-12-31",
"labels_present": false,
"model_version_metrics": {
  "precision": 1.0,
  "recall": 1.0,
  "mean_fractional_lead_time": 0.7760964912280702,
  "auc": 0.5971207364893062
}
```

Die wichtigsten Metriken

labeled_ranges

Zeitbereiche, in denen während der Schulung markierte Anomalien festgestellt wurden

labeled_event_metrics

Statistiken darüber, wie gut das Modell bekannte markierte Ereignisse identifizierte

num_labeled

Gesamtzahl der markierten Ereignisse in den Trainingsdaten

num_identified

Anzahl der markierten Ereignisse, die das Modell korrekt identifiziert hat

total_warning_time_in_seconds

Gesamtzeit, die das Modell für markierte Ereignisse im Warnstatus verbracht hat

predicted_ranges

Zeitbereiche, in denen das Modell während der Evaluierung Anomalien vorhergesagt hat

unknown_event_metrics

Statistiken über Anomalien, die in unmarkierten Daten festgestellt wurden

data_start_time / data_end_time

Zeitfenster, das von den Trainingsdaten abgedeckt wird

labels_present

Boolescher Wert, der angibt, ob während des Trainings beschriftete Daten verwendet wurden

model_version_metrics

Zusätzliche versionsspezifische Metriken für den Modellvergleich

Erweiterte Metriken für beschriftete Modelle

Wenn Sie während des Trainings beschriftete Daten angeben, sind zusätzliche Leistungskennzahlen in den Amazon S3 S3-Dateien enthalten:

- **Erinnerung:** Der Anteil der Ereignisse, die AWS IoT SiteWise korrekt identifiziert wurden, an den Ereignissen, die Sie im selben Zeitraum gekennzeichnet haben. Beispielsweise haben Sie möglicherweise 10 Ereignisse gekennzeichnet, aber AWS IoT SiteWise nur 9 davon identifiziert. In diesem Fall liegt der Rückruf bei 90%.
- **Präzision:** Der Anteil der wirklich positiven Ergebnisse an der Gesamtzahl der identifizierten Ereignisse. Wenn beispielsweise 10 Ereignisse AWS IoT SiteWise identifiziert werden, aber nur 7 dieser Ereignisse den von Ihnen beschrifteten Ereignissen entsprechen, liegt die Genauigkeit bei 70%.
- **MeanFractionalLeadTime:** Ein Maß dafür, wie schnell (im Verhältnis zur Dauer des Ereignisses) jedes Ereignis im Durchschnitt AWS IoT SiteWise erkannt wird. Ein typisches Ereignis in Ihrer Einrichtung kann beispielsweise 10 Stunden dauern. Im Durchschnitt kann es 3 Stunden dauern, bis das Modell das Ereignis identifiziert hat. In diesem Fall beträgt die durchschnittliche fraktionierte Vorlaufzeit 0,7.
- **AUC:** Area Under the Curve (AUC) misst die Fähigkeit eines Modells für maschinelles Lernen, für positive Beispiele eine höhere Punktzahl vorherzusagen als für negative Beispiele. Ein Wert zwischen 0 und 1, der angibt, wie gut Ihr Modell die Kategorien in Ihrem Datensatz trennen kann. Ein Wert von 1 gibt an, dass die Kategorien perfekt getrennt werden konnten.

Werbung für Modelle und Metriken

Bei Umschulungsabläufen ermöglichen die in Amazon S3 gespeicherten Kennzahlen fundierte Entscheidungen zur Modellförderung:

Verwalteter Modus (Automatische Beförderung)

- Das System vergleicht anhand der in Amazon S3 gespeicherten Daten automatisch Metriken zwischen alten und neuen Modellversionen
- Modelle werden auf der Grundlage verbesserter Leistungsindikatoren beworben

- Entscheidungen zur Beförderung beinhalten spezifische Ursachencodes, die zusammen mit den Kennzahlen gespeichert werden:
 - `AUTO_PROMOTION_SUCCESSFUL`: Neue Modellmetriken sind besser als die aktuelle Version
 - `MODEL_METRICS_DIDNT_IMPROVE`: Die Leistung des neuen Modells hat sich nicht verbessert
 - `POOR_MODEL_QUALITY_DETECTED`: Das neue Modell hat eine schlechte Qualitätsbewertung

Manueller Modus (vom Kunden gesteuerte Werbung)

- Sie können detaillierte Messwerte von Amazon S3 herunterladen und analysieren, um Werbeentscheidungen zu treffen.
- Alle historischen Modellversionen und ihre Metriken bleiben in Amazon S3 zugänglich
- Sie können mithilfe der in Amazon S3 gespeicherten Metriken benutzerdefinierte Dashboards und Analysetools erstellen.

Erweiterte Inferenzkonfigurationen

AWS IoT SiteWise ermöglicht es Kunden, Zeitpläne für Modellinferenzen zu konfigurieren, die auf ihre betrieblichen Anforderungen zugeschnitten sind.

Die Planung von Inferenzen ist grob in drei Modi unterteilt:

- [Hochfrequente Inferenzierung \(5 Minuten — 1 Stunde\)](#)
- [Niederfrequenz-Inferenzierung \(2 Stunden — 1 Tag\)](#)
- [Flexible Planung](#)

Hochfrequente Inferenzierung (5 Minuten — 1 Stunde)

Dieser Modus ist ideal für Prozesse, die kontinuierlich laufen oder bei denen sich die Sensorwerte stark ändern. In dieser Konfiguration wird die Inferenz häufig bis zu alle 5 Minuten ausgeführt.

Anwendungsfälle:

- Es wird zur Überwachung schnell wechselnder Geräte wie Kompressoren oder Förderanlagen eingesetzt.
- Es ist hilfreich bei der Erkennung kurzlebiger Anomalien, die eine sofortige Reaktion erfordern.
- Es ist ein ständig verfügbarer Betrieb, bei dem Daten konsistent fließen.

Bedingte Offset-Unterstützung:

Sie können einen bedingten Offset (0-60 Minuten) definieren, um die Inferenz nach der Datenaufnahme zu verzögern. Dadurch wird sichergestellt, dass verspätet eintreffende Daten weiterhin im Analysefenster enthalten sind.

So konfigurieren Sie Hochfrequenz-Inferenzen:

- Konfigurieren Sie den `AWS/ANOMALY_DETECTION_INFERENCE` Aktionsnutzdatenwert mit `DataUploadFrequency` with values: `PT5M`, `PT10M`, `PT15M`, `PT30M`, `PT1H` beim Starten der Inferenz.
- (Optional) Konfigurieren Sie `DataDelayOffsetInMinutes` mit dem Verzögerungsoffset in Minuten. Stellen Sie diesen Wert zwischen 0 und 60 Minuten ein.

```
{
  "inferenceMode": "START",
  "dataDelayOffsetInMinutes": "DataDelayOffsetInMinutes",
  "dataUploadFrequency": "DataUploadFrequency"
}
```

Example der Konfiguration mit Hochfrequenz-Inferenz:

```
{
  "inferenceMode": "START",
  "dataDelayOffsetInMinutes": "2",
  "dataUploadFrequency": "PT5M"
}
```

Niederfrequenz-Inferenzierung (2 Stunden — 1 Tag)

Dieser Modus eignet sich für langsamer ablaufende Prozesse oder Anwendungsfälle, bei denen tägliche Auswertungen ausreichend sind. Kunden konfigurieren die Inferenz so, dass sie stündlich oder einmal täglich ausgeführt wird.

Unterstützung zur Startzeit für ein eintägiges Intervall:

Für tägliche Inferenzen können Sie optional einen Wert **startTime**(jeden Tag um 8 Uhr) zusammen mit der Angabe der Zeitzone angeben.

Unterstützung für Zeitzonen:

Wenn eine bereitgestellt `startTime` wird, AWS IoT SiteWise verwendet die [Zeitzonendatenbank](#), die von der Internet Assigned Numbers Authority (IANA) verwaltet wird. Dadurch wird sichergestellt, dass Ihre Schlussfolgerungen auch in allen Regionen mit den lokalen Arbeitszeiten übereinstimmen.

Bedingte Offset-Unterstützung:

Wie bei anderen Modi ist ein bedingter Offset von 0 — 60 Minuten konfiguriert.

Anwendungsfälle:

- Tägliche Gesundheitschecks für Batch-Prozesse oder Schichtoperationen.
- Vermeidet Rückschlüsse bei Wartungsarbeiten oder Ausfallzeiten.
- Dies ist hilfreich in Umgebungen mit begrenzten Ressourcen, in denen die Rechenauslastung minimiert werden muss.

So konfigurieren Sie Niederfrequenz-Inferenzen:

- Konfigurieren Sie den Payload-Wert für die `AWS/ANOMALY_DETECTION_INFERENCE` Aktion `DataUploadFrequency` mit folgenden Werten: `PT2H..PT12H`
 - Im Fall von 1 Tag `DataUploadFrequency` ist `P1D`.
- (Optional) Konfigurieren Sie `DataDelayOffsetInMinutes` mit dem Verzögerungsoffset in Minuten. Stellen Sie diesen Wert zwischen 0 und 60 Minuten ein.

Example der Konfiguration mit Niederfrequenz-Inferenz:

```
{
  "inferenceMode": "START",
  "dataUploadFrequency": "P1D",
  "inferenceStartTime": "13:00",
  "inferenceTimeZone": "America/Chicago"
}
```

Flexible Planung

Die flexible Planung ermöglicht es Kunden, bestimmte Tage und Zeitbereiche zu definieren, in denen Inferenzen durchgeführt werden. Auf diese Weise haben Kunden die vollständige Kontrolle über die Planung auf der Grundlage von Produktionsstunden, Schichtzeiten und geplanten Ausfallzeiten.

Das `weeklyOperatingWindow` hilft, wenn:

- Die Ausrüstung läuft nur zu bestimmten Zeiten (8 Uhr — 16 Uhr).
- Am Wochenende findet keine Produktion statt.
- Die tägliche Wartung ist während bekannter Zeitblöcke geplant.

Unterstützung für Zeitzonen:

Wenn eine bereitgestellt `startTime` wird, AWS IoT SiteWise verwendet die [Zeitzonendatenbank](#), die von der Internet Assigned Numbers Authority (IANA) verwaltet wird. Dadurch wird sichergestellt, dass die Inferenz auch in allen Regionen mit den lokalen Arbeitszeiten übereinstimmt.

Bedingte Offset-Unterstützung:

Wie bei anderen Modi kann ein bedingter Offset von 0 bis 60 Minuten konfiguriert werden.

Vorteile von `weeklyOperatingWindow`:

- Dadurch werden Rückschlüsse während Leerlauf- oder Wartungsphasen vermieden und Fehlalarme reduziert.
- Es stimmt die Erkennung von Anomalien mit betrieblichen Prioritäten und schichtbasierten Arbeitsabläufen ab.

So konfigurieren Sie die flexible Planung:

- Konfigurieren Sie den Payload-Wert für die `AWS/ANOMALY_DETECTION_INFERENCE` Aktion mit `DataUploadFrequency`.
- (Optional) `DataDelayOffsetInMinutes` mit dem Verzögerungsoffset in Minuten. Stellen Sie diesen Wert zwischen 0 und 60 Minuten ein.
- `weeklyOperatingWindow` mit einer Schichtkonfiguration konfigurieren:
 - Die Schlüssel für die `weeklyOperatingWindow` sind `Wochentage:monday | tuesday | wednesday | thursday | friday | saturday | sunday`.
 - Jeder Zeitbereich muss im 24-Stunden-Format "`HH:MM-HH:MM`" ("`08:00-16:00`") angegeben werden.
 - Pro Tag können mehrere Bereiche angegeben werden.

Example mit flexibler Planungskonfiguration:

```
{
```

```
"inferenceMode": "START",
"dataUploadFrequency": "PT5M",
"weeklyOperatingWindow": {
  "tuesday": ["11:00-13:00"],
  "monday": ["10:00-11:00", "13:00-15:00"]
}
}
```

Aktivierung der Modellversion

Wenn Sie die Inferenz starten, können Sie optional eine bestimmte Modellversion aktivieren, um sie für die Erkennung von Anomalien zu verwenden. Mit dieser Funktion können Sie eine bestimmte trainierte Modellversion auswählen, zu früheren Versionen zurückkehren oder automatische Entscheidungen zur Modellförderung außer Kraft setzen.

Anwendungsfälle:

- Rollback für die Produktion: Schnelles Zurücksetzen auf eine stabile Modellversion, wenn die aktuelle Version Leistungseinbußen oder ein unerwartetes Verhalten aufweist.
- A/B-Tests: Vergleichen Sie die Leistung zwischen verschiedenen Modellversionen, indem Sie während bestimmter Zeiträume zwischen ihnen wechseln.
- Manuelle Modellauswahl: Überschreiben Sie automatische Werbeentscheidungen und wählen Sie Ihre bevorzugte Modellversion manuell auf der Grundlage der Geschäftsanforderungen aus.
- Schrittweise Bereitstellung: Testen Sie neuere Modellversionen in unkritischen Zeitfenstern, bevor Sie sie für den vollen Produktionseinsatz hochstufen.
- Leistungsoptimierung: Wählen Sie Modellversionen aus, die für bestimmte Betriebsbedingungen oder saisonale Muster besser funktionieren.
- Rollback während der Wartung: Verwenden Sie ältere, gut getestete Modellversionen während der Systemwartung oder bei Upgrades, um die Stabilität sicherzustellen.

Verhalten bei der Auswahl der Modellversion

Wann `targetModelVersion` ist angegeben:

- Das System aktiviert die angeforderte Modellversion für die Inferenz.
- Überprüft, ob die angegebene Modellversion existiert.
- Setzt alle Einstellungen für automatische Werbeaktionen außer Kraft.

Wann `targetModelVersion` ist nicht angegeben:

- Aktiviert die letzte aktive Modellversion, wenn die Inferenz zuvor gestartet wurde.
- Wenn die Inferenz nie aktiviert wurde, wird die neueste trainierte Modellversion verwendet.

Um eine bestimmte Modellversion zu aktivieren:

- Konfigurieren Sie die Payload für die Inferenzaktion und `targetModelVersion` stellen Sie sie auf die gewünschte Modellversionsnummer ein.
- Die angegebene Modellversion wird validiert und aktiviert, falls sie existiert.

Example der Aktivierung der Modellversion:

```
{
  "inferenceMode": "START",
  "dataUploadFrequency": "PT15M",
  "targetModelVersion": 2
}
```

Überprüfung der Modellversionen

Um die derzeit aktive Modellversion zu überprüfen:

- Verwenden Sie die [DescribeComputationModelExecutionSummary](#)API, die die aktive Modellversion in der Antwort enthält.

Um alle verfügbaren Modellversionen anzuzeigen:

- Verwenden Sie die [ListExecutions](#)API, um eine vollständige Liste der historischen Modellversionen abzurufen.
- Verwenden Sie die [DescribeExecution](#)API, um Informationen zum trainierten Modell abzurufen, einschließlich des Zeitbereichs der Exportdaten, der Version des Berechnungsmodells und der fakturierbaren Dauer in Minuten.

Eigenschaften der Modellversion

- Modellversionsnummern werden sequentiell ab 1 vergeben.

- Sie können alle zuvor trainierten Modellversionen aktivieren.
- Die aktivierte Modellversion bleibt bestehen, bis sie explizit geändert wird.
- Die Aktivierung der Modellversion funktioniert mit allen Inferenzplanungsmodi (Hochfrequenz, Niederfrequenz und flexibel).
- Wenn die angegebene Modellversion nicht existiert, schlägt die Inferenzaktion mit einem Fehler fehl.

Überprüfen Sie die Inferenzergebnisse

Inferenzergebnisse abrufen

Aktuelle Inferenzergebnisse

Führen Sie den folgenden Befehl aus, um das neueste Inferenzergebnis für eine Anlageneigenschaft abzurufen. Weitere Informationen finden Sie [get-asset-property-value](#) im AWS CLI Befehlsreferenzhandbuch.

```
aws iotsitewise get-asset-property-value \  
  --asset-id asset-id \  
  --property-id result-property-id
```

Verlauf der Inferenzergebnisse

Führen Sie den folgenden Befehl aus, um den Verlauf der Inferenzergebnisse für ein bestimmtes Zeitfenster abzurufen. Weitere Informationen finden Sie unter [get-asset-property-value-history](#) im AWS CLI Befehlsreferenzhandbuch.

```
aws iotsitewise get-asset-property-value-history \  
  --asset-id asset-id \  
  --property-id result-property-id \  
  --start-date start-time \  
  --end-date end-time
```

Beispielantwort

Example einer Antwort auf ein Inferenzergebnis:

```
{
```

```

"value": {
  "stringValue": "{\"timestamp\": \"2025-02-10T22:42:00.000000\", \"prediction\": 0,
  \"prediction_reason\": \"NO_ANOMALY_DETECTED\", \"diagnostics\": [{\"name\": \"asset-id\\\\\\\\property-id\", \"value\": 0.53528}]}"
},
"timestamp": {
  "timeInSeconds": 1739227320,
  "offsetInNanos": 0
},
"quality": "GOOD"
}

```

Antwortfelder

- **value.stringValue** — Eine JSON-Zeichenfolge, die das Inferenzergebnis mit den folgenden Feldern enthält:
 - **timestamp** — Der Zeitstempel des TQV, anhand dessen die Inferenz durchgeführt wird.
 - **Prognose** — Das Prognoseergebnis (0 für keine Anomalie, 1 für festgestellte Anomalie).
 - **prediction_reason** — Der Grund für die Vorhersage (oder). NO_ANOMALY_DETECTED
ANOMALY_DETECTED
 - **Diagnosen** — Eine Reihe von Diagnoseinformationen, die dazu beitragende Faktoren aufzeigen.
- **Zeitstempel** — Der Zeitstempel, in dem das Ergebnis aufgezeichnet wird. AWS IoT SiteWise
- **Qualität** — Die Qualität des Datenpunkts (normalerweise GOOD).

Verstehen Sie die Inferenzergebnisse

Ein bei der AWS IoT SiteWise Anomalieerkennung zurückgegebenes Inferenzergebnis enthält wichtige Informationen über die Vorhersage des Modells zu einem bestimmten Zeitpunkt, einschließlich der Frage, ob eine Anomalie erkannt wurde und welche Sensoren zu der Anomalie beigetragen haben.

Example eines detaillierten Inferenzergebnisses:

```

{
  "timestamp": "2021-03-11T22:25:00.000000",
  "prediction": 1,
  "prediction_reason": "ANOMALY_DETECTED",
  "anomaly_score": 0.72385,

```



```
"diagnostics": [
  { "name": "asset_id_1\\\\\\\\property_id_1", "value": 0.02346 },
  { "name": "asset_id_2\\\\\\\\property_id_2", "value": 0.10011 },
  { "name": "asset_id_3\\\\\\\\property_id_3", "value": 0.11162 }
]
```

Das `diagnostics` Feld ist nützlich, um zu interpretieren, warum das Modell eine bestimmte Vorhersage trifft. Jeder Eintrag in der Liste enthält:

- `name`: Der Sensor, der zur Vorhersage beigetragen hat (Format: `asset_id\\\\\\\\property_id`).
- `value`: Eine Gleitkommazahl zwischen 0 und 1, die das relative Gewicht oder die Bedeutung dieses Sensors zu diesem Zeitpunkt darstellt.

Vorteile für Benutzer:

- Finden Sie heraus, welche Sensoren den stärksten Einfluss auf eine Anomalie hatten.
- Korrelieren Sie Sensoren mit hohem Gewicht und dem Verhalten der physikalischen Ausrüstung.
- Informieren Sie sich über die Ursachenanalyse.

Note

Selbst wenn `prediction = 0` (normales Verhalten), wird die Diagnoseliste zurückgegeben. Dies hilft zu beurteilen, welche Sensoren derzeit die Entscheidungen des Modells beeinflussen, selbst in gesunden Zuständen.

Löst bei anomalem Verhalten benutzerdefinierte Aktionen aus (AWS-Managementkonsole)

Sie können benutzerdefinierte Aktionen als Reaktion auf anomales Verhalten aktivieren, indem Sie AWS IoT SiteWise MQTT-Benachrichtigungen in Kombination mit verwenden. AWS IoT Core

Gehen Sie wie folgt vor, um MQTT-Benachrichtigungen zu konfigurieren und AWS IoT Core basierend auf den Inferenzergebnissen eine benutzerdefinierte Aktion auszulösen: AWS IoT SiteWise

- Suchen Sie das Asset AWS IoT SiteWise, in dem die Inferenz ausgeführt wird.

- Identifizieren Sie die Eigenschaft, die Sie `resultProperty` bei der Erstellung des Berechnungsmodells verwendet haben. Aktivieren Sie die MQTT-Benachrichtigung für diese Eigenschaft.
- Sobald Sie die MQTT-Benachrichtigung aktiviert haben, kopieren Sie das generierte Benachrichtigungsthema. AWS IoT SiteWise
- Navigieren Sie zu. AWS IoT Core Abonnieren Sie unter MQTT-Testclient das kopierte Benachrichtigungsthema, um die eingehenden Nachrichten zu überwachen.
- Erstellen Sie eine AWS IoT Core Regel zur Verarbeitung von Inferenzergebnissen. (Dadurch wird sichergestellt, dass Aktionen nur ausgelöst werden, wenn das System eine Anomalie erkennt). `notification-topic` Ersetzen Sie es durch das AWS IoT SiteWise generierte Benachrichtigungsthema.

```
SELECT * FROM "notification-topic"
WHERE indexof(get(get(payload.values, 0).value, 'stringValue'),
"NO_ANOMALY_DETECTED") < 0
```

Konfigurieren Sie die Regel so, dass sie jede der AWS IoT Core unterstützten Aktionen auslöst. Erfahren Sie mehr über die Aktionen, die von unterstützt werden [AWS IoT Core](#).

Best Practices

Machen Sie sich mit dem Mindestzeitraum vertraut

Verwenden Sie für die Dauer der Trainingsdaten mindestens 14 Tage. In vielen Fällen empfehlen wir jedoch, einen längeren Datenzeitraum einzubeziehen.

Stellen Sie sicher, dass Ihr Trainingsdatensatz einen Zeitraum umfasst, in dem das Gerät in all seinen normalen Betriebsmodi betrieben wurde. Dieser Ansatz hilft dabei, AWS IoT SiteWise genau zwischen erwartetem Verhalten und echten Anomalien zu unterscheiden.

Wenn Ihre Trainingsdaten nicht alle typischen Betriebsmodi abbilden, werden ungewohnte, aber normale Muster AWS IoT SiteWise möglicherweise fälschlicherweise als Anomalien gekennzeichnet, was zu einer Zunahme von Fehlalarmen führt.

Stichprobenerhebung für hochfrequente Daten und Konsistenz zwischen Training und Inferenz

Wenn Ihre Sensoren Daten mit einer Frequenz von mehr als 1 Hz (mehr als eine Messung pro Sekunde) erzeugen, führen Sie während des Trainings Stichproben durch. Durch die Probennahme wird das Datenvolumen reduziert und gleichzeitig wichtige Trends beibehalten. Dies ermöglicht eine effiziente Verarbeitung und verbessert die Modellgeneralisierung, indem die Auswirkungen von Rauschen oder vorübergehenden Schwankungen minimiert werden.

AWS IoT SiteWise Die native Erkennung von Anomalien unterstützt derzeit keine Daten, die mit Raten unter 1 Hz aufgenommen werden. Stellen Sie sicher, dass Ihre Daten diese Mindestanforderung an die Frequenz erfüllen, bevor Sie die Anomalieerkennung konfigurieren.

AWS IoT SiteWise Verwendet außerdem die Samplingrate, die Sie während des Trainings konfigurieren, auch für Inferenz. Um die Konsistenz zu wahren und genaue Ergebnisse bei der Erkennung von Anomalien sicherzustellen, wählen Sie eine Samplerate, die sowohl Ihren betrieblichen Anforderungen als auch dem Verhalten Ihrer Sensordaten entspricht.

Weitere Informationen zur Einstellung der Probenahmerate finden Sie unter [Konfiguration der Abtastrate](#)

Empfehlungen zur Kennzeichnung

Eine genaue und konsistente Kennzeichnung von Anomalien ist für eine effektive Modellevaluierung und kontinuierliche Verbesserung unerlässlich. Beachten Sie bei der Kennzeichnung von Anomalien die folgenden bewährten Methoden:

- **Zusammenhängende Anomalien konsolidieren:** Kennzeichnen Sie häufig auftretende Anomalien nicht als separate Ereignisse, wenn sie Teil desselben zugrundeliegenden Problems sind. Wenn Anomalien beispielsweise innerhalb von 1–2 Tagen auftreten und sie von derselben Grundursache verursacht werden, behandeln Sie sie als ein einziges Anomaliefenster. Dieser Ansatz hilft dem Modell, das Muster abnormalen Verhaltens besser zu erkennen, und reduziert das Rauschen in Ihren Bewertungsdaten.
- **Kennzeichnen Sie Anomaliefenster, nicht nur Punkte:** Anstatt einzelne Datenpunkte als anomal zu kennzeichnen, kennzeichnen Sie das gesamte Fenster, das abnormales Verhalten vom Beginn der Abweichung bis zur Wiederherstellung widerspiegelt. Dieser Ansatz bietet klarere Grenzen und verbessert die Abstimmung des Modells auf tatsächliche betriebliche Probleme.

- **Unsichere Zeiträume ausschließen:** Wenn Sie sich nicht sicher sind, ob ein Zeitraum ungewöhnlich ist, lassen Sie ihn unbeschriftet. Mehrdeutige Beschriftungen können das Modell verwirren und seine Genauigkeit im Laufe der Zeit verschlechtern.

Weitere Informationen zum Hinzufügen von Beschriftungen finden Sie unter [Kennzeichnen Sie Ihre Daten](#)

Datenspeicher verwalten in AWS IoT SiteWise

Sie können so konfigurieren AWS IoT SiteWise , dass Ihre Daten in den folgenden Speicherstufen gespeichert werden:

Heiße Stufe

Bei der Hot-Storage-Tier handelt es sich um einen AWS IoT SiteWise verwalteten Zeitreihenspeicher. Hot Tier ist am effektivsten für Daten, auf die häufig zugegriffen wird, und hat eine geringe write-to-read Latenz. Im Hot-Tier gespeicherte Daten werden von Industrieanwendungen verwendet, die schnellen Zugriff auf die neuesten Messwerte in Ihren Geräten benötigen. Dazu gehören Anwendungen, die Echtzeit-Metriken mit einem interaktiven Dashboard visualisieren, oder Anwendungen, die den Betrieb überwachen und Alarmer auslösen, um Leistungsprobleme zu identifizieren.

Standardmäßig werden die AWS IoT SiteWise aufgenommenen Daten im Hot-Tier gespeichert. Sie können einen Aufbewahrungszeitraum für das Hot-Tier definieren. Danach werden die Daten auf dem Hot-Tier je nach Konfiguration entweder in den Warm- oder Cold-Tier-Speicher AWS IoT SiteWise verschoben. Um optimale Leistung und Kosteneffizienz zu erzielen, sollten Sie die Aufbewahrungsdauer für die Hot-Tier-Stufe so festlegen, dass sie länger ist als die Zeit, die häufig für das Abrufen von Daten benötigt wird. Dies wird für Echtzeit-Metriken, Alarmer und Überwachungsszenarien verwendet. Wenn kein Aufbewahrungszeitraum festgelegt ist, werden Ihre Daten auf unbestimmte Zeit im Hot-Tier gespeichert.

Warme Stufe

Bei der Warm-Storage-Tier handelt AWS IoT SiteWise es sich um eine verwaltete Stufe, die sich für die kosteneffiziente Speicherung historischer Daten eignet. Sie eignet sich am besten zum Abrufen großer Datenmengen mit mittleren write-to-read Latenzeigenschaften. Verwenden Sie die warme Ebene, um historische Daten zu speichern, die für große Workloads benötigt werden. Es wird beispielsweise für den Datenabruf für Analysen, Business Intelligence-Anwendungen (BI), Berichtstools und das Training von Modellen für maschinelles Lernen (ML) verwendet. Wenn Sie die Cold-Storage-Stufe aktivieren, können Sie eine Aufbewahrungsfrist für die warme Stufe definieren. AWS IoT SiteWise Löscht nach Ablauf der Aufbewahrungsfrist Daten aus der warmen Stufe.

Kalte Stufe

Die Kühlspeicherebene verwendet einen Amazon S3 S3-Bucket zum Speichern von Daten, die selten verwendet werden. Bei aktiviertem Cold Tier werden die Zeitreihen,

einschließlich Messungen, Metriken, Transformationen und Aggregaten sowie Definitionen von Anlagenmodellen, alle 6 Stunden AWS IoT SiteWise repliziert. Cold Tier wird verwendet, um Daten zu speichern, die eine hohe Leselatenz für historische Berichte und Backups tolerieren.

Themen

- [Konfigurieren Sie die Speichereinstellungen in AWS IoT SiteWise](#)
- [Problembehandlung bei den Speichereinstellungen für AWS IoT SiteWise](#)
- [Dateipfade und Schemas von Daten, die auf der kalten Ebene gespeichert wurden](#)

Konfigurieren Sie die Speichereinstellungen in AWS IoT SiteWise

Sie können Speichereinstellungen so konfigurieren, dass Sie sich für die Wartung von verwaltetem Speicher auf der warmen Ebene entscheiden und Daten auch auf das kalte Tier replizieren. Weitere Informationen zur Aufbewahrungsdauer für die Warm- und Hot-Tarife finden Sie unter [Auswirkungen auf die Datenspeicherung](#). Gehen Sie bei der Konfiguration der Speichereinstellungen wie folgt vor:

- **Aufbewahrung auf hoher Ebene** — Legen Sie einen Aufbewahrungszeitraum fest, in dem Ihre Daten auf der heißen Ebene gespeichert werden, bevor sie gelöscht und je nach Ihren Speichereinstellungen in den vom Service verwalteten Speicher auf der warmen oder kalten Ebene verschoben werden. AWS IoT SiteWise löscht alle Daten in der Hot-Tier, die vor Ablauf der Aufbewahrungsfrist vorhanden waren. Wenn Sie keinen Aufbewahrungszeitraum festlegen, werden Ihre Daten auf unbestimmte Zeit im Hot-Tier gespeichert.
- **Aufbewahrung auf warmer Ebene** — Legen Sie einen Aufbewahrungszeitraum fest, in dem Ihre Daten auf der Warm-Tier-Ebene gespeichert werden, bevor sie aus dem AWS IoT SiteWise Speicher gelöscht und in den vom Kunden verwalteten Cold-Tier-Speicher verschoben werden. AWS IoT SiteWise löscht alle Daten aus der Warm-Tier, die vor Ablauf der Aufbewahrungsfrist vorhanden waren. Wenn kein Aufbewahrungszeitraum festgelegt ist, werden Ihre Daten auf unbestimmte Zeit in der Warm-Tier gespeichert.

Note

Um die Abfrageleistung zu verbessern, legen Sie mit Warm-Tier-Speicher einen Hot-Tier-Aufbewahrungszeitraum fest.

Auswirkungen der Datenspeicherung auf Speicher der heißen und warmen Speicherebene

- Wenn Sie die Aufbewahrungsdauer des Hot-Tier-Speichers verkürzen, werden Daten dauerhaft vom Hot-Tier in das Warm- oder Cold-Tier verschoben. Wenn Sie die Aufbewahrungsdauer der warmen Schicht verkürzen, werden Daten in die kalte Schicht verschoben und aus der warmen Schicht dauerhaft gelöscht.
- Wenn Sie die Aufbewahrungsdauer des Speichers der heißen oder warmen Ebene verlängern, wirkt sich die Änderung auf Daten aus, an die AWS IoT SiteWise ab diesem Zeitpunkt gesendet werden. AWS IoT SiteWise ruft keine Daten aus dem warmen oder kalten Speicher ab, um den heißen Speicher zu füllen. Wenn beispielsweise die Aufbewahrungsdauer des Hot-Tier-Speichers zunächst auf 30 Tage festgelegt und dann auf 60 Tage erhöht wird, dauert es 30 Tage, bis der Hot-Tier-Speicher Daten im Wert von 60 Tagen enthält.

Themen

- [Konfigurieren Sie die Speichereinstellungen für das warme Tier \(Konsole\)](#)
- [Konfigurieren Sie die Speichereinstellungen für die Warmstufe \(AWS CLI\)](#)
- [Konfigurieren Sie die Speichereinstellungen für das Cold-Tier \(Konsole\)](#)
- [Konfigurieren Sie die Speichereinstellungen für Cold Tier \(AWS CLI\)](#)

Konfigurieren Sie die Speichereinstellungen für das warme Tier (Konsole)

Das folgende Verfahren zeigt Ihnen, wie Sie die Speichereinstellungen für die Replikation von Daten auf das warme Tier in der AWS IoT SiteWise Konsole konfigurieren.

So konfigurieren Sie die Speichereinstellungen in der Konsole

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich unter Einstellungen die Option Speicher aus.
3. Wählen Sie rechts oben die Option Edit (Bearbeiten) aus.
4. Gehen Sie auf der Seite Speicher bearbeiten wie folgt vor:
5. Gehen Sie für Hot-Tier-Einstellungen wie folgt vor:

- Wenn Sie einen Aufbewahrungszeitraum für die Dauer festlegen möchten, für die Ihre Daten auf dem Hot-Tier gespeichert werden, bevor sie gelöscht und in den vom Service verwalteten Warm-Tier-Speicher verschoben werden, wählen Sie Aufbewahrungszeitraum aktivieren.
- Um einen Aufbewahrungszeitraum zu konfigurieren, geben Sie eine ganze Zahl ein und wählen Sie eine Einheit aus. Die Aufbewahrungsfrist muss mindestens 30 Tage betragen.

AWS IoT SiteWise löscht alle Daten im Hot-Tier, die älter als die Aufbewahrungsfrist sind.

Wenn Sie keinen Aufbewahrungszeitraum festlegen, werden Ihre Daten auf unbestimmte Zeit gespeichert.

6. (Empfohlen) Gehen Sie für die Warm-Tier-Einstellungen wie folgt vor:

- Um sich für den Warm-Tier-Speicher zu entscheiden, wählen Sie Ich bestätige die Option Ich bestätige die Option Warm-Tier-Speicher, um sich für den Warm-Tier-Speicher zu entscheiden.
- (Optional) Um einen Aufbewahrungszeitraum zu konfigurieren, geben Sie eine ganze Zahl ein und wählen Sie eine Einheit aus. Die Aufbewahrungsdauer muss mindestens 365 Tage betragen.

AWS IoT SiteWise löscht Daten in der Warm-Tier, die vor dem Aufbewahrungszeitraum existierten. Wenn Sie keinen Aufbewahrungszeitraum festlegen, werden Ihre Daten auf unbestimmte Zeit gespeichert.

 Note

- Wenn Sie sich für die Warm-Stufe entscheiden, wird die Konfiguration nur einmal angezeigt.
- Um die Aufbewahrung auf der heißen Ebene festzulegen, müssen Sie entweder über einen warmen oder einen kalten Speicher verfügen. Aus Gründen der Kosteneffizienz und des Abrufs historischer Daten AWS IoT SiteWise empfiehlt es sich, Langzeitdaten im Warm-Tier zu speichern.
- Um die Aufbewahrung auf der Warm-Tier-Ebene festzulegen, müssen Sie über einen Cold-Tier-Speicher verfügen.

7. Wählen Sie Speichern, um Ihre Speichereinstellungen zu speichern.

Im AWS IoT SiteWise Speicherbereich befindet sich der Warm Tier-Speicher in einem der folgenden Zustände:

- Aktiviert — Wenn Ihre Daten bereits vor dem Aufbewahrungszeitraum für das heiße Tier vorhanden waren, werden die Daten auf das Warm-Tier AWS IoT SiteWise verschoben.“
- Deaktiviert — Der Warm-Tier-Speicher ist deaktiviert.

Konfigurieren Sie die Speichereinstellungen für die Warmstufe (AWS CLI)

Sie können Speichereinstellungen so konfigurieren, dass Daten auf die warme Ebene verschoben werden, indem Sie die AWS CLI und die folgenden Befehle verwenden.

Um zu verhindern, dass die bestehende Konfiguration überschrieben wird, rufen Sie die aktuellen Speicherkonfigurationsinformationen ab, indem Sie den folgenden Befehl ausführen:

```
aws iotsitewise describe-storage-configuration
```

Example Antwort ohne bestehende Cold-Tier-Konfiguration

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "disassociatedDataStorage": "ENABLED",
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2021-10-14T15:53:35-07:00",
  "warmTier": "DISABLED"
}
```

Example Antwort mit vorhandener Cold-Tier-Konfiguration

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/",
      "roleArn": "arn:aws:iam::aws-account-id:role/role-name"
    }
  },
  "disassociatedDataStorage": "ENABLED",
}
```

```
{
  "retentionPeriod": {
    "numberOfDays": retention-in-days
  },
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2023-10-25T15:59:46-07:00",
  "warmTier": "DISABLED"
}
```

Konfigurieren Sie die Speichereinstellungen für die warme Stufe mit AWS CLI

Führen Sie den folgenden Befehl aus, um die Speichereinstellungen zu konfigurieren. `file-name` Ersetzen Sie es durch den Namen der Datei, die die AWS IoT SiteWise Speicherkonfiguration enthält.

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

Example AWS IoT SiteWise Konfiguration mit heißer und warmer Stufe

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "disassociatedDataStorage": "ENABLED",
  "warmTier": "ENABLED",
  "retentionPeriod": {
    "numberOfDays": hot-tier-retention-in-days
  }
}
```

`hot-tier-retention-in-days` muss eine ganze Zahl größer oder gleich 30 Tagen sein.

Example response

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "configurationStatus": {
    "state": "UPDATE_IN_PROGRESS"
  }
}
```

Wenn Sie Cold-Tier-Speicher aktiviert haben, finden Sie weitere Informationen unter [Konfigurieren Sie Speichereinstellungen mit einem AWS CLI vorhandenen Cold-Tier](#).

Konfigurieren Sie Speichereinstellungen mit einem AWS CLI vorhandenen Cold-Tier

Konfigurieren Sie die Speichereinstellungen AWS CLI mithilfe des vorhandenen Cold-Tier-Speichers

- Führen Sie den folgenden Befehl aus, um die Speichereinstellungen zu konfigurieren.
file-name Ersetzen Sie es durch den Namen der Datei, die die AWS IoT SiteWise Speicherkonfiguration enthält.

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

Example AWS IoT SiteWise Speicherkonfiguration

- amzn-s3-demo-bucket* Ersetzen Sie es durch Ihren Amazon S3 S3-Bucket-Namen.
- Ersetzen Sie es *prefix* durch Ihr Amazon S3 S3-Präfix.
- aws-account-id* Ersetzen Sie es durch Ihre AWS Konto-ID.
- role-name* Ersetzen Sie es durch den Namen der Amazon S3-Zugriffsrolle, die das Senden von Daten AWS IoT SiteWise an Amazon S3 ermöglicht.
- hot-tier-retention-in-days* Ersetzen Sie es durch eine ganze Zahl, die größer oder gleich 30 Tagen ist.
- Ersetze es *warm-tier-retention-in-days* durch eine ganze Zahl, die größer oder gleich 365 Tagen ist.

Note

AWS IoT SiteWise löscht alle Daten in der warmen Stufe, die älter sind als die Aufbewahrungsfrist der kalten Stufe. Wenn Sie keinen Aufbewahrungszeitraum festlegen, werden Ihre Daten auf unbestimmte Zeit gespeichert.

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/",

```

```

        "roleArn": "arn:aws:iam::aws-account-id:role/role-name"
      },
    },
    "disassociatedDataStorage": "ENABLED",
    "retentionPeriod": {
      "numberOfDays": hot-tier-retention-in-days
    },
    "warmTier": "ENABLED",
    "warmTierRetentionPeriod": {
      "numberOfDays": warm-tier-retention-in-days
    }
  }
}

```

Example response

```

{
  "storageType": "MULTI_LAYER_STORAGE",
  "configurationStatus": {
    "state": "UPDATE_IN_PROGRESS"
  }
}

```

Konfigurieren Sie die Speichereinstellungen für das Cold-Tier (Konsole)

Das folgende Verfahren zeigt Ihnen, wie Sie die Speichereinstellungen so konfigurieren, dass Daten auf das Cold-Tier in der AWS IoT SiteWise Konsole repliziert werden.

So konfigurieren Sie die Speichereinstellungen in der Konsole

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich unter Einstellungen die Option Speicher aus.
3. Wählen Sie rechts oben die Option Edit (Bearbeiten) aus.
4. Gehen Sie auf der Seite Speicher bearbeiten wie folgt vor:
 - a. Wählen Sie unter Speichereinstellungen die Option Cold-Tier-Speicher aktivieren aus. Der Cold-Tier-Speicher ist standardmäßig deaktiviert.
 - b. Geben Sie für S3-Bucket-Standort den Namen eines vorhandenen Amazon S3 S3-Buckets und ein Präfix ein.

 Note

- Amazon S3 verwendet das Präfix als Ordernamen im Amazon S3 S3-Bucket. Das Präfix muss 1—255 Zeichen lang sein und mit einem Schrägstrich (/) enden. Ihre AWS IoT SiteWise Daten werden in diesem Ordner gespeichert.
- Wenn Sie keinen Amazon S3 S3-Bucket haben, wählen Sie View und erstellen Sie dann einen in der Amazon S3 S3-Konsole. Weitere Informationen finden Sie unter [Erstellen Sie Ihren ersten S3-Bucket](#) im Amazon S3 S3-Benutzerhandbuch.

c. Gehen Sie für die S3-Zugriffsrolle wie folgt vor:

- Wählen Sie Create a role from an AWS managed template. Dadurch AWS wird automatisch eine IAM-Rolle erstellt, die das Senden von Daten AWS IoT SiteWise an Amazon S3 ermöglicht.
- Wählen Sie Bestehende Rolle verwenden und wählen Sie dann die Rolle, die Sie erstellt haben, aus der Liste aus.

 Note

- Sie müssen denselben Amazon S3 S3-Bucket-Namen für den S3-Bucket-Speicherort verwenden, den Sie im vorherigen Schritt und in Ihrer IAM-Richtlinie verwendet haben.
- Stellen Sie sicher, dass Ihre Rolle über die im folgenden Beispiel gezeigten Berechtigungen verfügt.

Example Berechtigungsrichtlinie:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
```

```

        "s3:GetBucketLocation",
        "s3:ListBucket"
    ],
    "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ]
}
]
}

```

Ersetzen Sie amzn-s3-demo-bucket durch den Namen Ihres Amazon S3 S3-Buckets.

- Wenn der Amazon S3 S3-Bucket mit einem vom Kunden verwalteten KMS-Schlüssel verschlüsselt ist, muss der KMS-Schlüssel über eine Zugriffsrichtlinie mit einer IAM-Rolle für kms:Decrypt und kms:GenerateDataKey - Operationen verfügen.

- Informationen zur Einrichtung von Hot Tier finden Sie in [Konfigurieren Sie die Speichereinstellungen für das warme Tier \(Konsole\)](#) Schritt 5 unter.
- (Optional) Gehen Sie zur AWS IoT Analytics Integration wie folgt vor.

Note

Hinweis zum Ende des Supports: Am 15. Dezember 2025 AWS endet der Support für AWS IoT Analytics. Nach dem 15. Dezember 2025 können Sie nicht mehr auf die AWS IoT Analytics Konsole oder die AWS IoT Analytics Ressourcen zugreifen. Weitere Informationen finden Sie unter [AWS IoT Analytics Ende des Supports](#).

- Wenn Sie Ihre Daten abfragen AWS IoT Analytics möchten, wählen Sie Enabled AWS IoT Analytics data store aus.
- AWS IoT SiteWise generiert einen Namen für Ihren Datenspeicher, oder Sie können einen anderen Namen eingeben.

AWS IoT SiteWise erstellt automatisch einen Datenspeicher AWS IoT Analytics zum Speichern Ihrer Daten. Um die Daten abzufragen, können Sie sie verwenden, AWS IoT

Analytics um Datensätze zu erstellen. Weitere Informationen finden Sie im AWS IoT Analytics Benutzerhandbuch unter [Arbeiten mit AWS IoT SiteWise Daten](#).

- f. Wählen Sie Speichern.

Im Bereich AWS IoT SiteWise Speicher kann der Cold-Tier-Speicher einen der folgenden Werte annehmen:

- Aktiviert — AWS IoT SiteWise repliziert Ihre Daten in den angegebenen Amazon S3 S3-Bucket.
- Aktiviert — AWS IoT SiteWise verarbeitet Ihre Anfrage zur Aktivierung des Cold-Tier-Speichers. Dieser Vorgang kann mehrere Minuten in Anspruch nehmen.
- Enable_Failed — Ihre Anfrage zur Aktivierung des Cold-Tier-Speichers AWS IoT SiteWise konnte nicht verarbeitet werden. Wenn Sie AWS IoT SiteWise das Senden von Protokollen an Amazon CloudWatch Logs aktiviert haben, können Sie diese Protokolle zur Behebung von Problemen verwenden. Weitere Informationen finden Sie unter [Mit Amazon CloudWatch Logs überwachen](#).
- Deaktiviert — Der Cold-Tier-Speicher ist deaktiviert.

Konfigurieren Sie die Speichereinstellungen für Cold Tier (AWS CLI)

Das folgende Verfahren zeigt Ihnen, wie Sie die Speichereinstellungen für die Replikation von Daten auf das Cold-Tier mithilfe von AWS CLI konfigurieren.

Um Speichereinstellungen zu konfigurieren mit AWS CLI

1. Um Daten in einen Amazon S3 S3-Bucket in Ihrem Konto zu exportieren, führen Sie den folgenden Befehl aus, um die Speichereinstellungen zu konfigurieren. *file-name* Ersetzen Sie es durch den Namen der Datei, die die AWS IoT SiteWise Speicherkonfiguration enthält.

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

Example AWS IoT SiteWise Speicherkonfiguration

- *amzn-s3-demo-bucket* Ersetzen Sie es durch Ihren Amazon S3 S3-Bucket-Namen.
- Ersetzen Sie es *prefix* durch Ihr Amazon S3 S3-Präfix.
- *aws-account-id* Ersetzen Sie es durch Ihre AWS Konto-ID.
- *role-name* Ersetzen Sie es durch den Namen der Amazon S3-Zugriffsrolle, die das Senden von Daten AWS IoT SiteWise an Amazon S3 ermöglicht.

- *retention-in-days* Ersetzen Sie es durch eine ganze Zahl, die größer oder gleich 30 Tagen ist.

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3::amzn-s3-demo-bucket/prefix",
      "roleArn": "arn:aws:iam::aws-account-id:role/role-name"
    }
  },
  "retentionPeriod": {
    "numberOfDays": retention-in-days,
    "unlimited": false
  }
}
```

Note

- Sie müssen denselben Amazon S3 S3-Bucket-Namen in der AWS IoT SiteWise Speicherkonfiguration und in der IAM-Richtlinie verwenden.
- Stellen Sie sicher, dass Ihre Rolle über die im folgenden Beispiel gezeigten Berechtigungen verfügt.

Example Berechtigungsrichtlinie:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:GetBucketLocation",
        "s3:ListBucket"
      ]
    }
  ],
}
```



```

      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
      ]
    }
  ]
}

```

Ersetzen Sie `amzn-s3-demo-bucket` durch den Namen Ihres Amazon S3 S3-Buckets.

- Wenn der Amazon S3 S3-Bucket mit einem vom Kunden verwalteten KMS-Schlüssel verschlüsselt ist, muss der KMS-Schlüssel über eine Zugriffsrichtlinie mit einer IAM-Rolle für `kms:Decrypt` und `kms:GenerateDataKey` -Operationen verfügen.

Example response

```

{
  "storageType": "MULTI_LAYER_STORAGE",
  "retentionPeriod": {
    "numberOfDays": 100,
    "unlimited": false
  },
  "configurationStatus": {
    "state": "UPDATE_IN_PROGRESS"
  }
}

```

Note

Die Aktualisierung der Speicherkonfiguration kann einige Minuten AWS IoT SiteWise dauern.

2. Führen Sie den folgenden Befehl aus, um die Informationen zur Speicherkonfiguration abzurufen.

```
aws iotsitewise describe-storage-configuration
```

Example response

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3:::amzn-s3-demo-bucket/torque/",
      "roleArn": "arn:aws:iam::123456789012:role/SWAccessS3Role"
    }
  },
  "retentionPeriod": {
    "numberOfDays": 100,
    "unlimited": false
  },
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2021-03-30T15:54:14-07:00"
}
```

3. Um den Export von Daten in den Amazon S3 S3-Bucket zu beenden, führen Sie den folgenden Befehl aus, um die Speichereinstellungen zu konfigurieren.

```
aws iotsitewise put-storage-configuration --storage-type SITEWISE_DEFAULT_STORAGE
```

Note

Standardmäßig werden Ihre Daten nur im Hot-Tier von gespeichert AWS IoT SiteWise.

Example response

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "configurationStatus": {
    "state": "UPDATE_IN_PROGRESS"
  }
}
```

4. Führen Sie den folgenden Befehl aus, um die Informationen zur Speicherkonfiguration abzurufen.

```
aws iotsitewise describe-storage-configuration
```

Example response

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2021-03-30T15:57:14-07:00"
}
```

(Optional) Erstellen Sie einen AWS IoT Analytics Datenspeicher (AWS CLI)

 Note

Hinweis zum Ende des Supports: Am 15. Dezember 2025 AWS endet der Support für AWS IoT Analytics. Nach dem 15. Dezember 2025 können Sie nicht mehr auf die AWS IoT Analytics Konsole oder die AWS IoT Analytics Ressourcen zugreifen. Weitere Informationen finden Sie unter [AWS IoT Analytics Ende des Supports](#).

Ein AWS IoT Analytics Datenspeicher ist ein skalierbares und abfragbares Repository, das Daten empfängt und speichert. Sie können die AWS IoT SiteWise Konsole verwenden oder einen AWS IoT Analytics Datenspeicher AWS IoT Analytics APIs zum Speichern Ihrer AWS IoT SiteWise Daten erstellen. Um die Daten abzufragen, erstellen Sie Datensätze mithilfe AWS IoT Analytics von. Weitere Informationen finden Sie im AWS IoT Analytics Benutzerhandbuch unter [Arbeiten mit AWS IoT SiteWise Daten](#).

Die folgenden Schritte dienen AWS CLI zum Erstellen eines Datenspeichers in AWS IoT Analytics.

Führen Sie den folgenden Befehl aus, um einen Datenspeicher zu erstellen. *file-name* Ersetzen Sie ihn durch den Namen der Datei, die die Datenspeicherkonfiguration enthält.

```
aws iotanalytics create-datastore --cli-input-json file://file-name.json
```

 Note

- Sie müssen den Namen eines vorhandenen Amazon S3 S3-Buckets angeben. Wenn Sie keinen Amazon S3 S3-Bucket haben, erstellen Sie zuerst einen. Weitere Informationen finden Sie unter [Erstellen Sie Ihren ersten S3-Bucket](#) im Amazon S3 S3-Benutzerhandbuch.
- Sie müssen denselben Amazon S3 S3-Bucket-Namen in der AWS IoT SiteWise Speicherkonfiguration, der IAM-Richtlinie und der AWS IoT Analytics Datenspeicherkonfiguration verwenden.

Example AWS IoT Analytics Datenspeicher-Konfiguration

Ersetzen Sie *data-store-name* und durch *amzn-s3-demo-bucket* den Namen Ihres AWS IoT Analytics Datenspeichers und den Namen Ihres Amazon S3 S3-Buckets.

```
{
  "datastoreName": "data-store-name",
  "datastoreStorage": {
    "iotSiteWiseMultiLayerStorage": {
      "customerManagedS3Storage": {
        "bucket": "amzn-s3-demo-bucket"
      }
    }
  },
  "retentionPeriod": {
    "numberOfDays": 90
  }
}
```

Example response

```
{
  "datastoreName": "datastore_IoTSiteWise_demo",
  "datastoreArn": "arn:aws:iotanalytics:us-west-2:123456789012:datastore/
datastore_IoTSiteWise_demo",
  "retentionPeriod": {
    "numberOfDays": 90,
    "unlimited": false
  }
}
```

```
}
```

Problembehandlung bei den Speichereinstellungen für AWS IoT SiteWise

Verwenden Sie die folgenden Informationen, um Probleme mit der Speicherkonfiguration zu beheben und zu lösen.

Problembereiche

- [Fehler: Bucket ist nicht vorhanden](#)
- [Fehler: Zugriff auf den Amazon S3-Pfad verweigert](#)
- [Fehler: Rollen-ARN kann nicht übernommen werden](#)
- [Fehler: Auf den regionsübergreifenden Amazon S3 S3-Bucket konnte nicht zugegriffen werden](#)

Fehler: Bucket ist nicht vorhanden

Lösung: Ihr Amazon S3 S3-Bucket AWS IoT SiteWise konnte nicht gefunden werden. Stellen Sie sicher, dass Sie den Namen eines vorhandenen Amazon S3 S3-Buckets in der aktuellen Region eingeben.

Fehler: Zugriff auf den Amazon S3-Pfad verweigert

Lösung: AWS IoT SiteWise Ich konnte nicht auf Ihren Amazon S3 S3-Bucket zugreifen. Gehen Sie wie folgt vor:

- Stellen Sie sicher, dass Sie denselben Amazon S3 S3-Bucket verwenden, den Sie in der IAM-Richtlinie angegeben haben.
- Stellen Sie sicher, dass Ihre Rolle über die im folgenden Beispiel gezeigten Berechtigungen verfügt.

Example Berechtigungsrichtlinie

JSON

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "s3:*",  
            "Resource": "arn:aws:s3:::my-bucket/*"        }  
    ]  
}
```

```
{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject",
    "s3:GetObject",
    "s3:DeleteObject",
    "s3:GetBucketLocation",
    "s3:ListBucket"
  ],
  "Resource": [
    "arn:aws:s3:::amzn-s3-demo-bucket",
    "arn:aws:s3:::amzn-s3-demo-bucket/*"
  ]
}
```

Ersetzen Sie amzn-s3-demo-bucket durch den Namen Ihres Amazon S3 S3-Buckets.

Fehler: Rollen-ARN kann nicht übernommen werden

Lösung: Die IAM-Rolle AWS IoT SiteWise konnte nicht in Ihrem Namen übernommen werden. Stellen Sie sicher, dass Ihre Rolle dem folgenden Dienst vertraut: `ioticsitewise.amazonaws.com` Weitere Informationen finden Sie unter [Ich kann keine Rolle annehmen](#) im IAM-Benutzerhandbuch.

Fehler: Auf den regionsübergreifenden Amazon S3 S3-Bucket konnte nicht zugegriffen werden

Lösung: Der Amazon S3 S3-Bucket, den Sie angegeben haben, befindet sich in einer anderen AWS Region. Stellen Sie sicher, dass sich Ihr Amazon S3 S3-Bucket und Ihre AWS IoT SiteWise Assets in derselben Region befinden.

Dateipfade und Schemas von Daten, die auf der kalten Ebene gespeichert wurden

AWS IoT SiteWise speichert Ihre Daten auf der kalten Ebene, indem Zeitreihen repliziert werden, einschließlich Messungen, Metriken, Transformationen und Aggregaten sowie Definitionen von Anlagen und Anlagenmodellen. Im Folgenden werden die Dateipfade und Schemas der Daten beschrieben, die an die Cold-Tier gesendet werden.

Themen

- [Gerätedaten \(Messungen\)](#)
- [Metriken, Transformationen und Aggregationen](#)
- [Asset-Metadaten](#)
- [Metadaten der Asset-Hierarchie](#)
- [Speicherdaten, Indexdateien](#)

Gerätedaten (Messungen)

AWS IoT SiteWise exportiert alle sechs Stunden Gerätedaten (Messungen) in die Kühlzelle. Rohdaten werden im Cold-Tier im [Apache AVRO](#) (.avro) -Format gespeichert.

Dateipfad

AWS IoT SiteWise speichert Gerätedaten (Messungen) im Cold-Tier unter Verwendung der folgenden Vorlage.

```
{keyPrefix}/raw/startYear={startYear}/startMonth={startMonth}/startDay={startDay}/  
seriesBucket={seriesBucket}/raw_{timeseriesId}_{startTimestamp}_{quality}.avro
```

Jeder Dateipfad zu Rohdaten in Amazon S3 enthält die folgenden Komponenten.

Pfadkomponente	Description
keyPrefix	Das Amazon S3 S3-Präfix, das Sie in der AWS IoT SiteWise Speicherkonfiguration angegeben haben. Amazon S3 verwendet das Präfix als Ordernamen im Bucket.
raw	Der Ordner, in dem Zeitreihendaten von Geräten (Messungen) gespeichert werden. Der raw Ordner wird im Präfixordner gespeichert.
seriesBucket	Eine Hexadezimalzahl zwischen 00 und ff. Diese Zahl ist abgeleitet von. timeSeriesId Diese Partition wird verwendet, um den Durchsatz bei AWS IoT SiteWise Schreibvo

Pfadkomponente	Description
	rgängen auf das Cold-Tier zu erhöhen. Wenn Sie Amazon Athena zum Ausführen von Abfragen verwenden, können Sie die Partition für eine detaillierte Partitionierung verwenden, um die Abfrageleistung zu verbessern. <code>seriesBucket</code> und <code>timeSeriesBucket</code> in den Asset-Metadaten steht dieselbe Zahl.
<code>startYear</code>	Das Jahr der exklusiven Startzeit, die den Zeitreihendaten zugeordnet ist.
<code>startMonth</code>	Der Monat der exklusiven Startzeit, die den Zeitreihendaten zugeordnet ist.
<code>startDay</code>	Der Tag des Monats, an dem die exklusive Startzeit den Zeitreihendaten zugeordnet ist.
<code>fileName</code>	Der Dateiname verwendet den Unterstrich (_) als Trennzeichen, um Folgendes zu trennen: • Das Präfix. <code>raw</code> • Der <code>timeSeriesId</code> Wert. • Der Epochenzeitstempel der exklusiven Startzeit, die den Zeitreihendaten zugeordnet ist. • Die Qualität der Daten. Gültige Werte: GOOD, BAD und UNCERTAIN Weitere Informationen finden Sie unter AssetPropertyValue in der AWS IoT SiteWise -API-Referenz. Die Datei wird mithilfe der Snappy-Komprimierung in dem <code>.avro</code> Format gespeichert.

Example Dateipfad zu den Rohdaten in der kalten Ebene

```
keyPrefix/raw/startYear=2021/startMonth=1/startDay=2/seriesBucket=a2/
raw_7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_95e63da7-d34e-43e1-
bc6f-1b490154b07a_1609577700_G00D.avro
```

Felder

Das Schema der Rohdaten, die in die Cold-Tier exportiert werden, enthält die folgenden Felder.

AWS IoT SiteWise rät Kunden, die Unterstützung für die Schemaentwicklung auf Systemen zu implementieren, die Rohdaten aus dem Cold-Tier lesen, da in future möglicherweise weitere Felder eingeführt werden.

Nulldaten werden so dargestellt, dass alle Wertfelder Null sind. Kunden erhalten jedoch weiterhin den richtigen Datentyp, wenn sie eine Anfrage mit AWS IoT SiteWise APIs stellen.

Feldname	Unterstützte -Typen	-Standardtyp	Description
seriesId	string	–	Die ID, die die Zeitreihendaten von Geräten identifiziert (Messungen). Sie können dieses Feld verwenden, um Rohdaten und Asset-Metadaten in Abfragen zu verknüpfen.
timeInSeconds	long	–	Das Zeitstempeldatum in Sekunden im Unix-Epochenformat. Daten in Bruchteilen von Nanosekunden werden bereitgestellt von. offsetInNanos

Feldname	Unterstützte -Typen	-Standardtyp	Description
offsetInNanos	long	–	Der Nanosekunden-Offset von. timeInSeconds
quality	string	–	Die Qualität des Zeitreihenwerts.
doubleValue	double oder null	null	Zeitreihendaten vom Typ Double (Fließkommazahl).
stringValue	string oder null	null	Zeitreihendaten vom Typ Zeichenfolge (Zeichenfolge).
integerValue	int oder null	null	Zeitreihendaten vom Typ Integer (ganze Zahl).
booleanValue	boolean oder null	null	Zeitreihendaten vom Typ Boolean (wahr oder falsch).
jsonValue	string oder null	null	Zeitreihendaten des Typs JSON (komplexe Datentypen, die als Zeichenfolge gespeichert werden).

Feldname	Unterstützte -Typen	-Standardtyp	Description
recordVersion	long oder null	null	Die Versionsnummer für den Datensatz. Sie können die Versionsnummer verwenden, um den neuesten Datensatz auszuwählen. Neuere Datensätze haben größere Versionsnummern.

Example Rohdaten in der kalten Stufe

```
{
  "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "timeInSeconds": 1625675887,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "doubleValue": 0.75,
  "stringValue": null,
  "integerValue": null,
  "booleanValue": null,
  "jsonValue": null,
  "recordVersion": 1
}, {
  "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "timeInSeconds": 1625675889,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "doubleValue": 0.69,
  "stringValue": null,
  "integerValue": null,
  "booleanValue": null,
  "jsonValue": null,
  "recordVersion": 2
}, {
  "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "timeInSeconds": 1625675890,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "doubleValue": 0.66,
  "stringValue": null,
  "integerValue": null,
  "booleanValue": null,
  "jsonValue": null,
  "recordVersion": 3
}, {
  "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "timeInSeconds": 1625675891,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "doubleValue": 0.92,
  "stringValue": null,
  "integerValue": null,
  "booleanValue": null,
  "jsonValue": null,
  "recordVersion": 4
}, {
  "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "timeInSeconds": 1625675892,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "doubleValue": 0.73,
  "stringValue": null,
  "integerValue": null,
  "booleanValue": null,
  "jsonValue": null,
  "recordVersion": 5
}
```

Metriken, Transformationen und Aggregationen

AWS IoT SiteWise exportiert alle sechs Stunden Metriken, Transformationen und Aggregationen in das Cold-Tier. Metriken, Transformationen und Aggregate werden im Cold-Tier im [Apache AVRO](#) () -Format gespeichert. `.avro`

Dateipfad

AWS IoT SiteWise speichert Metriken, Transformationen und Aggregate im Cold-Tier mithilfe der folgenden Vorlage.

```
{keyPrefix}/agg/startYear={startYear}/startMonth={startMonth}/startDay={startDay}/
seriesBucket={seriesBucket}/agg_{timeseriesId}_{startTimestamp}_{quality}.avro
```

Jeder Dateipfad zu Metriken, Transformationen und Aggregationen in Amazon S3 enthält die folgenden Komponenten.

Pfadkomponente	Description
keyPrefix	Das Amazon S3 S3-Präfix, das Sie in der AWS IoT SiteWise Speicherkonfiguration angegeben haben. Amazon S3 verwendet das Präfix als Ordernamen im Bucket.
agg	Der Ordner, in dem Zeitreihendaten aus Metriken gespeichert werden. Der agg Ordner wird im Präfixordner gespeichert.
seriesBucket	Eine Hexadezimalzahl zwischen 00 und ff. Diese Zahl ist abgeleitet von. <code>timeSeriesId</code> Diese Partition wird verwendet, um den Durchsatz bei AWS IoT SiteWise Schreibvorgängen auf das Cold-Tier zu erhöhen. Wenn Sie Amazon Athena zum Ausführen von Abfragen verwenden, können Sie die Partition für eine detaillierte Partitionierung verwenden, um die Abfrageleistung zu verbessern. <code>seriesBucket</code> und <code>timeSeriesBucket</code> in den Asset-Metadaten steht dieselbe Zahl.
startYear	Das Jahr der exklusiven Startzeit, die den Zeitreihendaten zugeordnet ist.

Pfadkomponente	Description
startMonth	Der Monat der exklusiven Startzeit, die den Zeitreihendaten zugeordnet ist.
startDay	Der Tag des Monats, an dem die exklusive Startzeit den Zeitreihendaten zugeordnet ist.
fileName	Der Dateiname verwendet den Unterstrich (_) als Trennzeichen, um Folgendes zu trennen: • Das Präfix. raw • Der <code>timeSeriesId</code> Wert. • Der Epochenzeitstempel der exklusiven Startzeit, die den Zeitreihendaten zugeordnet ist. • Die Qualität der Daten. Gültige Werte: GOOD, BAD und UNCERTAIN Weitere Informationen finden Sie unter AssetPropertyValue in der AWS IoT SiteWise -API-Referenz. Die Datei wird mithilfe der Snappy-Komprimierung in dem .avro Format gespeichert.

Example Dateipfad zu den Messwerten in der kalten Stufe

```
keyPrefix/agg/startYear=2021/startMonth=1/startDay=2/seriesBucket=a2/agg_7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_95e63da7-d34e-43e1-bc6f-1b490154b07a_1609577700_GOOD.avro
```

Felder

Das Schema der Metriken, Transformationen und Aggregate, die in das Cold-Tier exportiert werden, enthält die folgenden Felder.

Feldname	Unterstützte -Typen	-Standardtyp	Description
seriesId	string	–	Die ID, die die Zeitreihendaten von Geräten, Metriken oder Transformationen identifiziert. Sie können dieses Feld verwenden, um Rohdaten und Asset-Metadaten in Abfragen zu verknüpfen.
timeInSeconds	long	–	Das Zeitstempeldatum in Sekunden im Unix-Epochenformat. Daten in Bruchteilen von Nanosekunden werden bereitgestellt von. offsetInNanos
offsetInNanos	long	–	Der Nanosekunden-Offset von. timeInSeconds
quality	string	–	Die Qualität, nach der Anlagendaten gefiltert werden sollen.
resolution	string	–	Das Zeitintervall, über das Daten aggregiert werden sollen.
count	double oder null	null	Die Gesamtzahl der Datenpunkte für

Feldname	Unterstützte -Typen	-Standardtyp	Description
			die angegebenen Variablen im aktuellen Zeitintervall.
average	double oder null	null	Der Mittelwert der Werte der angegebenen Variablen im aktuellen Zeitintervall.
min	double oder null	null	Das Minimum der Werte der angegebenen Variablen im aktuellen Zeitintervall.
max	boolean oder null	null	Das Maximum der Werte der angegebenen Variablen im aktuellen Zeitintervall.
sum	string oder null	null	Die Summe der Werte der angegebenen Variablen im aktuellen Zeitintervall.
recordVersion	long oder null	null	Die Versionsnummer für den Datensatz. Sie können die Versionsnummer verwenden , um den neuesten Datensatz auszuwählen. Neuere Datensätze haben größere Versionsnummern.

Example Metrische Daten in der kalten Stufe

```
{
  "seriesId": "f74c2828-5317-4df3-ba16-6d41b5bcb531",
  "timeInSeconds": 1637334060,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "resolution": "16.0",
  "min": {
    "double": 1.0
  },
  "max": {
    "double": 31.0
  },
  "sum": {
    "double": 496.0
  },
  "recordVersion": null
}
{"seriesId": "f74c2828-5317-4df3-ba16-6d41b5bcb531",
  "timeInSeconds": 1637334120,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "resolution": "16.0",
  "min": {
    "double": 32.0
  },
  "max": {
    "double": 60.0
  },
  "sum": {
    "double": 1334.0
  },
  "recordVersion": null
}
{"seriesId": "f74c2828-5317-4df3-ba16-6d41b5bcb531",
  "timeInSeconds": 1637334540,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "resolution": "16.0",
  "min": {
    "double": 1.0
  },
  "max": {
    "double": 31.0
  },
  "sum": {
    "double": 496.0
  },
  "recordVersion": null
}
{"seriesId": "f74c2828-5317-4df3-ba16-6d41b5bcb531",
  "timeInSeconds": 1637334600,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "resolution": "16.0",
  "min": {
    "double": 32.0
  },
  "max": {
    "double": 60.0
  },
  "sum": {
    "double": 1334.0
  },
  "recordVersion": null
}
{"seriesId": "f74c2828-5317-4df3-ba16-6d41b5bcb531",
  "timeInSeconds": 1637335020,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "resolution": "16.0",
  "min": {
    "double": 1.0
  },
  "max": {
    "double": 31.0
  },
  "sum": {
    "double": 496.0
  },
  "recordVersion": null
}
```

Asset-Metadaten

Wenn Sie AWS IoT SiteWise zum ersten Mal den Export von Daten in die kalte Ebene aktivieren, werden Asset-Metadaten in die kalte Ebene exportiert. AWS IoT SiteWise Exportiert nach der Erstkonfiguration Asset-Metadaten nur dann in die Ebene, wenn Sie Asset-Modelldefinitionen oder Asset-Definitionen ändern. Asset-Metadaten werden in der kalten Ebene im durch Zeilenumbrüche getrennten JSON (`.ndjson`) -Format gespeichert.

Dateipfad

AWS IoT SiteWise speichert Asset-Metadaten unter Verwendung der folgenden Vorlage in der kalten Ebene.

```
{keyPrefix}/asset_metadata/asset_{assetId}.ndjson
```

Jeder Dateipfad zu Asset-Metadaten in der kalten Ebene enthält die folgenden Komponenten.

Pfadkomponente	Description
keyPrefix	Das Amazon S3 S3-Präfix, das Sie in der Speicherkonfiguration AWS IoT SiteWise s angegeben haben. Amazon S3 verwendet das Präfix als Ordernamen im Bucket.
asset_metadata	Der Ordner, der Asset-Metadaten speichert . Der asset_metadata Ordner wird im Präfixordner gespeichert.
fileName	Der Dateiname verwendet den Unterstrich (_) als Trennzeichen, um Folgendes zu trennen: • Das Präfix. asset • Der assetId Wert. Die Datei wird im .ndjson Format gespeichert.

Example Dateipfad zu den Asset-Metadaten in der kälteren Ebene

keyPrefix/asset_metadata/asset_35901915-d476-4dca-8637-d9ed4df939ed.ndjson

Felder

Das Schema der Asset-Metadaten, das in die kalte Ebene exportiert wird, enthält die folgenden Felder.

Feldname	Description
assetId	Die ID der -Komponente.
assetName	Der Name des Assets.
assetExternalId	Die externe ID des Assets.
assetModelId	Die ID des Asset-Modells, das zur Erstellung dieses Assets verwendet wurde.

Feldname	Description
assetModelName	Der Name des Asset-Modells.
assetModelExternalId	Die externe ID des Asset-Modells.
assetPropertyId	Die ID der Asset-Eigenschaft.
assetPropertyName	Der Name der Anlageeigenschaft.
assetPropertyExternalId	Die externe ID der Anlageeigenschaft.
assetPropertyDataType	Der Datentyp der Anlageneigenschaft.
assetPropertyUnit	Die Einheit der Anlageeigenschaft (z. B. Newtons undRPM).
assetPropertyAlias	Der Alias, der die Asset-Eigenschaft identifiziert, z. B. einen Datenstream-Pfad eines OPC UA-Servers (z. B./company/windfarm/3/turbine/7/temperature).
timeSeriesId	Die ID, die die Zeitreihendaten von Geräten, Metriken oder Transformationen identifiziert. Sie können dieses Feld verwenden, um Rohdaten und Asset-Metadaten in Abfragen zu verknüpfen.

Feldname	Description
<code>timeSeriesBucket</code>	Eine Hexadezimalzahl zwischen 00 und ff. Diese Zahl ist abgeleitet von <code>timeSeriesId</code>. Diese Partition wird verwendet, um den Durchsatz bei AWS IoT SiteWise Schreibvorgängen auf das Cold-Tier zu erhöhen. Wenn Sie Amazon Athena zum Ausführen von Abfragen verwenden, können Sie die Partition für eine detaillierte Partitionierung verwenden, um die Abfrageleistung zu verbessern. <code>timeSeriesBucket</code> und <code>seriesBucket</code> im Dateipfad zu den Rohdaten stehen dieselben Zahlen.
<code>assetCompositeModelId</code>	Die ID des zusammengesetzten Modells.
<code>assetCompositeModelExternalId</code>	Die externe ID des zusammengesetzten Modells.
<code>assetCompositeModelDescription</code>	Die Beschreibung des zusammengesetzten Modells.
<code>assetCompositeModelName</code>	Der Name des zusammengesetzten Modells.
<code>assetCompositeModelType</code>	Der Typ des zusammengesetzten Modells. Bei zusammengesetzten Alarmmodellen ist dieser Typ <code>AWS/ALARM</code> .
<code>assetCreationDate</code>	Das Datum, an dem das Asset erstellt wurde, in Unix-Epochezeit.
<code>assetLastUpdateDate</code>	Das Datum, an dem das Asset zuletzt aktualisiert wurde, in Unix-Epochezeit.
<code>assetStatusErrorCode</code>	Der Fehlercode.
<code>assetStatusErrorMessage</code>	Die Fehlermeldung.

Feldname	Description
assetStatusState	Der aktuelle Status des Assets.

Example Asset-Metadaten auf der kalten Ebene

```
{
  "assetId": "7020c8e2-e6db-40fa-9845-ed0ddddd4c77d",
  "assetExternalId": null,
  "assetName": "Wind Turbine Asset 2",
  "assetModelId": "ec1d924f-f07d-444f-b072-e2994c165d35",
  "assetModelExternalId": null,
  "assetModelName": "Wind Turbine Asset Model",
  "assetPropertyId": "95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "assetPropertyExternalId": null,
  "assetPropertyName": "Temperature",
  "assetPropertyData": {
    "value": 100,
    "unit": "Celsius",
    "timeSeriesId": "7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_95e63da7-d34e-43e1-bc6f-1b490154b07a",
    "timeSeriesBucket": "f6",
    "assetArn": null,
    "assetCompositeModelDescription": null
  }
},
{
  "assetId": "7020c8e2-e6db-40fa-9845-ed0ddddd4c77d",
  "assetExternalId": null,
  "assetName": "Wind Turbine Asset 2",
  "assetModelId": "ec1d924f-f07d-444f-b072-e2994c165d35",
  "assetModelExternalId": null,
  "assetModelName": "Wind Turbine Asset Model",
  "assetPropertyId": "c706d54d-4c11-42dc-9a01-63662fc697b4",
  "assetPropertyExternalId": null,
  "assetPropertyName": "Pressure",
  "assetPropertyData": {
    "value": 100,
    "unit": "Psi",
    "timeSeriesId": "7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_c706d54d-4c11-42dc-9a01-63662fc697b4",
    "timeSeriesBucket": "1e",
    "assetArn": null,
    "assetCompositeModelDescription": null
  }
},
{
  "assetId": "7020c8e2-e6db-40fa-9845-ed0ddddd4c77d",
  "assetExternalId": null,
  "assetName": "Wind Turbine Asset 2",
  "assetModelId": "ec1d924f-f07d-444f-b072-e2994c165d35",
  "assetModelExternalId": null,
  "assetModelName": "Wind Turbine Asset Model",
  "assetPropertyId": "8cf1162f-dead-4fbe-b468-c8e24cde9f50",
  "assetPropertyExternalId": null,
  "assetPropertyName": "Max Temperature",
  "assetPropertyDataType": "DOUBLE",
  "assetPropertyUnit": null,
  "assetPropertyAlias": null,
  "assetPropertyData": {
    "value": 100,
    "unit": "Celsius",
    "timeSeriesId": "7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_8cf1162f-dead-4fbe-b468-c8e24cde9f50",
    "timeSeriesBucket": "d7",
    "assetArn": null,
    "assetCompositeModelDescription": null,
    "assetStatusState": "ONLINE"
  }
},
{
  "assetId": "3a5f2a22-3b37-4332-9c1c-404ea1d73fab",
  "assetExternalId": null,
  "assetName": "BatchAssessment",
  "assetModelExternalId": null,
  "assetModelName": "FlashTestAssetModelDouble",
  "assetPropertyId": "ebc75e75e827-b410-ab401a9176ed",
  "assetPropertyExternalId": null,
  "assetPropertyName": "measurementProperty",
  "assetPropertyData": {
    "value": 100,
    "unit": "Celsius",
    "timeSeriesId": "3a5f2a22-3b37-4332-9c1c-404ea1d73fab_ebc75e75e827-b410-ab401a9176ed",
    "timeSeriesBucket": "af",
    "assetArn": null,
    "assetCompositeModelDescription": null,
    "assetStatusState": "ONLINE"
  }
}
```

Metadaten der Asset-Hierarchie

Wenn Sie das Speichern von Daten AWS IoT SiteWise auf der kalten Ebene zum ersten Mal aktivieren, werden Metadaten der Asset-Hierarchie in die kalte Ebene exportiert. AWS IoT SiteWise Exportiert nach der Erstkonfiguration Metadaten der Asset-Hierarchie nur dann in die Cold-Tier, wenn Sie Änderungen am Asset-Modell oder an den Asset-Definitionen vornehmen. Metadaten der Asset-Hierarchie werden in der kalten Ebene im durch Zeilenumbrüche getrennten JSON () .ndjson - Format gespeichert.

Eine externe Kennung für die Hierarchie, das Ziel-Asset oder das Quell-Asset wird durch Aufrufen der API abgerufen. [DescribeAsset](#)

Dateipfad

AWS IoT SiteWise speichert Metadaten der Asset-Hierarchie auf der kalten Ebene mithilfe der folgenden Vorlage.

```
{keyPrefix}/asset_hierarchy_metadata/{parentAssetId}_{hierarchyId}.ndjson
```

Jeder Dateipfad zu den Metadaten der Asset-Hierarchie in der kalten Ebene enthält die folgenden Komponenten.

Pfadkomponente	Description
keyPrefix	Das Amazon S3 S3-Präfix, das Sie in der AWS IoT SiteWise Speicherkonfiguration angegeben haben. Amazon S3 verwendet das Präfix als Ordernamen im Bucket.
asset_hierarchy_metadata	Der Ordner, der Metadaten der Asset-Hierarchie speichert. Der asset_hierarchy_metadata Ordner wird im Präfixordner gespeichert.
fileName	Der Dateiname verwendet den Unterstrich (_) als Trennzeichen, um Folgendes zu trennen: Der Wert. parentAssetId

Pfadkomponente	Description
	Der hierarchyId Wert. Die Datei wird im .ndjson Format gespeichert.

Example Dateipfad zu den Metadaten der Asset-Hierarchie in der kalten Ebene

keyPrefix/asset_hierarchy_metadata/35901915-d476-4dca-8637-d9ed4df939ed_c5b3ced8-589a-48c7-9998-cdcccfc9747a0.ndjson

Felder

Das Schema der Metadaten der Asset-Hierarchie, das in die kalte Ebene exportiert wird, enthält die folgenden Felder.

Feldname	Description
sourceAssetId	Die ID des Quell-Assets in dieser Asset-Beziehung.
targetAssetId	Die ID der Zielanlage in dieser Vermögensbeziehung.
hierarchyId	Die ID der Hierarchie.
associationType	Der Zuordnungstyp dieser Vermögensbeziehung. Der Wert muss seinCHILD. Die Zielanlage ist eine untergeordnete Anlage der Quellanlage.

Example Metadaten der Asset-Hierarchie auf der kalten Ebene

```
{
  "sourceAssetId": "80388e72-2284-44fb-9c89-bfbaf0dfedd2",
  "targetAssetId": "2b866c25-0c74-4750-bdf5-b73683c8a2a2",
  "hierarchyId": "bbed9f59-0412-4585-a61d-6044db526aee",
  "associationType": "CHILD"
}
```

```
{"sourceAssetId":"80388e72-2284-44fb-9c89-bfbaf0dfedd2","targetAssetId":"6b51246e-984d-460d-bc0b-470ea47d1e31","hierarchyId":"bbed9f59-0412-4585-a61d-6044db526aee","associationType":"CHILD"}
```

Um Ihre Daten auf der kalten Ebene anzuzeigen

1. Navigieren Sie zur [Amazon S3 S3-Konsole](#).
2. Wählen Sie im Navigationsbereich Buckets und dann Ihren Amazon S3 S3-Bucket aus.
3. Navigieren Sie zu dem Ordner, der die Rohdaten, Asset-Metadaten oder Asset-Hierarchie-Metadaten enthält.
4. Wählen Sie die Dateien aus, und klicken Sie dann unter Aktionen auf Herunterladen.

Speicherdaten, Indexdateien

AWS IoT SiteWise verwendet diese Dateien, um die Leistung von Datenabfragen zu optimieren. Sie werden in Ihrem Amazon S3 S3-Bucket angezeigt, aber Sie müssen sie nicht verwenden.

Dateipfad

AWS IoT SiteWise speichert Datenindexdateien mithilfe der folgenden Vorlage im Cold-Tier.

```
keyPrefix/index/series=timeseriesId/startYear=startYear/startMonth=startMonth/  
startDay=startDay/index_timeseriesId_startTimestamp_quality
```

Example Dateipfad zur Datenspeicher-Indexdatei

```
keyPrefix/index/series=7020c8e2-e6db-40fa-9845-ed0dddd4c77d_95e63da7-  
d34e-43e1-bc6f-1b490154b07a/startYear=2022/startMonth=02/startDay=03/  
index_7020c8e2-e6db-40fa-9845-ed0dddd4c77d_95e63da7-d34e-43e1-  
bc6f-1b490154b07a_1643846400_G00D
```

Codebeispiele für die AWS IoT SiteWise Verwendung AWS SDKs

Die folgenden Codebeispiele zeigen, wie die Verwendung AWS IoT SiteWise mit einem AWS Software Development Kit (SDK) funktioniert.

Bei Grundlagen handelt es sich um Codebeispiele, die Ihnen zeigen, wie Sie die wesentlichen Vorgänge innerhalb eines Services ausführen.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Während Aktionen Ihnen zeigen, wie Sie einzelne Service-Funktionen aufrufen, können Sie Aktionen im Kontext der zugehörigen Szenarien anzeigen.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Service mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Codebeispiele

- [Grundlegende Beispiele für die AWS IoT SiteWise Verwendung AWS SDKs](#)
 - [Hallo AWS IoT SiteWise](#)
 - [Lernen Sie die Grundlagen AWS IoT SiteWise mit einem AWS SDK kennen](#)
 - [Aktionen zur AWS IoT SiteWise Verwendung AWS SDKs](#)
 - [Verwendung BatchPutAssetPropertyValue mit einem AWS SDK oder CLI](#)
 - [Verwendung CreateAsset mit einem AWS SDK oder CLI](#)
 - [Verwendung CreateAssetModel mit einem AWS SDK oder CLI](#)
 - [Verwendung CreateGateway mit einem AWS SDK oder CLI](#)
 - [Verwendung DeleteAsset mit einem AWS SDK oder CLI](#)
 - [Verwendung DeleteAssetModel mit einem AWS SDK oder CLI](#)
 - [Verwendung DeleteGateway mit einem AWS SDK oder CLI](#)
 - [Verwendung DescribeAssetModel mit einem AWS SDK oder CLI](#)
 - [Verwendung DescribeGateway mit einem AWS SDK oder CLI](#)
 - [Verwendung GetAssetPropertyValue mit einem AWS SDK oder CLI](#)
 - [ListAssetModelPropertiesMit einem AWS SDK verwenden](#)

- [Verwendung ListAssetModels mit einem AWS SDK oder CLI](#)

Grundlegende Beispiele für die AWS IoT SiteWise Verwendung AWS SDKs

Die folgenden Codebeispiele zeigen, wie die Grundlagen von AWS IoT SiteWise with verwendet AWS SDKs werden.

Beispiele

- [Hallo AWS IoT SiteWise](#)
- [Lernen Sie die Grundlagen AWS IoT SiteWise mit einem AWS SDK kennen](#)
- [Aktionen zur AWS IoT SiteWise Verwendung AWS SDKs](#)
 - [Verwendung BatchPutAssetPropertyValue mit einem AWS SDK oder CLI](#)
 - [Verwendung CreateAsset mit einem AWS SDK oder CLI](#)
 - [Verwendung CreateAssetModel mit einem AWS SDK oder CLI](#)
 - [Verwendung CreateGateway mit einem AWS SDK oder CLI](#)
 - [Verwendung DeleteAsset mit einem AWS SDK oder CLI](#)
 - [Verwendung DeleteAssetModel mit einem AWS SDK oder CLI](#)
 - [Verwendung DeleteGateway mit einem AWS SDK oder CLI](#)
 - [Verwendung DescribeAssetModel mit einem AWS SDK oder CLI](#)
 - [Verwendung DescribeGateway mit einem AWS SDK oder CLI](#)
 - [Verwendung GetAssetPropertyValue mit einem AWS SDK oder CLI](#)
 - [ListAssetModelPropertiesMit einem AWS SDK verwenden](#)
 - [Verwendung ListAssetModels mit einem AWS SDK oder CLI](#)

Hallo AWS IoT SiteWise

Die folgenden Codebeispiele zeigen, wie Sie mit der Verwendung von AWS IoT SiteWise beginnen.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
public class HelloSitewise {
    private static final Logger logger =
        LoggerFactory.getLogger(HelloSitewise.class);
    public static void main(String[] args) {
        fetchAssetModels();
    }

    /**
     * Fetches asset models using the provided {@link IoTSiteWiseAsyncClient}.
     */
    public static void fetchAssetModels() {
        IoTSiteWiseAsyncClient siteWiseAsyncClient =
            IoTSiteWiseAsyncClient.create();
        ListAssetModelsRequest assetModelsRequest =
            ListAssetModelsRequest.builder()
                .assetModelTypes(AssetModelType.ASSET_MODEL)
                .build();

        // Asynchronous paginator - process paginated results.
        ListAssetModelsPublisher listModelsPaginator =
            siteWiseAsyncClient.listAssetModelsPaginator(assetModelsRequest);
        CompletableFuture<Void> future = listModelsPaginator.subscribe(response -
> {
            response.assetModelSummaries().forEach(assetSummary ->
                logger.info("Asset Model Name: {} ", assetSummary.name())
            );
        });

        // Wait for the asynchronous operation to complete
        future.join();
    }
}
```

- Einzelheiten zur API finden Sie [ListAssetModels](#) in der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
import {
  paginateListAssetModels,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";

// Call ListDocuments and display the result.
export const main = async () => {
  const client = new IoTSiteWiseClient();
  const listAssetModelsPaginated = [];
  console.log(
    "Hello, AWS Systems Manager! Let's list some of your documents:\n",
  );
  try {
    // The paginate function is a wrapper around the base command.
    const paginator = paginateListAssetModels({ client }, { maxResults: 5 });
    for await (const page of paginator) {
      listAssetModelsPaginated.push(...page.assetModelSummaries);
    }
  } catch (caught) {
    console.error(`There was a problem saying hello: ${caught.message}`);
    throw caught;
  }
  for (const { name, creationDate } of listAssetModelsPaginated) {
    console.log(`${name} - ${creationDate}`);
  }
};

// Call function if run directly.
```

```
import { fileURLToPath } from "node:url";
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main();
}
```

- Einzelheiten zur API finden Sie [ListAssetModels](#) in der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
import boto3

def hello_iot_sitewise(iot_sitewise_client):
    """
    Use the AWS SDK for Python (Boto3) to create an AWS IoT SiteWise
    client and list the asset models in your account.
    This example uses the default settings specified in your shared credentials
    and config files.

    :param iot_sitewise_client: A Boto3 AWS IoT SiteWise Client object. This
    object wraps
                                the low-level AWS IoT SiteWise service API.
    """
    print("Hello, AWS IoT SiteWise! Let's list some of your asset models:\n")
    paginator = iot_sitewise_client.get_paginator("list_asset_models")
    page_iterator = paginator.paginate(PaginationConfig={"MaxItems": 10})

    asset_model_names: [str] = []
    for page in page_iterator:
        for asset_model in page["assetModelSummaries"]:
            asset_model_names.append(asset_model["name"])
```

```

print(f"{len(asset_model_names)} asset model(s) retrieved.")
for asset_model_name in asset_model_names:
    print(f"\t{asset_model_name}")

if __name__ == "__main__":
    hello_iot_sitewise(boto3.client("iotsitewise"))

```

- Einzelheiten zur API finden Sie [ListAssetModels](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

TRY.
    oo_result = lo_ios->listassetmodels( ). " oo_result is returned for
testing purposes. "
    DATA(lt_asset_models) = oo_result->get_assetmodelsummaries( ).
    MESSAGE 'Retrieved list of asset models.' TYPE 'I'.
CATCH /aws1/cx_rt_generic.
    MESSAGE 'Unable to list asset models.' TYPE 'E'.
ENDTRY.

```

- Einzelheiten zur API finden Sie [ListAssetModels](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Service mit einem AWSSDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Lernen Sie die Grundlagen AWS IoT SiteWise mit einem AWS SDK kennen

Die folgenden Code-Beispiele veranschaulichen Folgendes:

- Erstellen Sie ein AWS IoT SiteWise Asset-Modell.
- Erstellen Sie ein AWS IoT SiteWise Asset.
- Rufen Sie die Eigenschafts-ID-Werte ab.
- Daten an ein AWS IoT SiteWise Asset senden.
- Ruft den Wert der Eigenschaft AWS IoT SiteWise Asset ab.
- Erstellen Sie ein AWS IoT SiteWise Portal.
- Erstellen Sie ein AWS IoT SiteWise Gateway.
- Beschreiben Sie das AWS IoT SiteWise Gateway.
- Löschen Sie die AWS IoT SiteWise Assets.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

Führen Sie ein interaktives Szenario durch, in dem AWS IoT SiteWise Funktionen demonstriert werden.

```
public class SitewiseScenario {

    public static final String DASHES = new String(new char[80]).replace("\0",
    "-");

    private static final Logger logger =
    LoggerFactory.getLogger(SitewiseScenario.class);
    static Scanner scanner = new Scanner(System.in);

    private static final String ROLES_STACK = "RoleSitewise";
```

```
static SitewiseActions sitewiseActions = new SitewiseActions();

public static void main(String[] args) throws Throwable {
    Scanner scanner = new Scanner(System.in);
    String contactEmail = "user@mydomain.com"; // Change email address.
    String assetModelName = "MyAssetModel1";
    String assetName = "MyAsset1" ;
    String portalName = "MyPortal1" ;
    String gatewayName = "MyGateway1" ;
    String myThing = "MyThing1" ;

    logger.info("""
        AWS IoT SiteWise is a fully managed software-as-a-service (SaaS)
that
        makes it easy to collect, store, organize, and monitor data from
industrial equipment and processes.
        It is designed to help industrial and manufacturing organizations
collect data from their equipment and
        processes, and use that data to make informed decisions about their
operations.

        One of the key features of AWS IoT SiteWise is its ability to connect
to a wide range of industrial
        equipment and systems, including programmable logic controllers
(PLCs), sensors, and other
        industrial devices. It can collect data from these devices and
organize it into a unified data model,
        making it easier to analyze and gain insights from the data. AWS IoT
SiteWise also provides tools for
        visualizing the data, setting up alarms and alerts, and generating
reports.

        Another key feature of AWS IoT SiteWise is its ability to scale to
handle large volumes of data.
        It can collect and store data from thousands of devices and process
millions of data points per second,
        making it suitable for large-scale industrial operations.
Additionally, AWS IoT SiteWise is designed
        to be secure and compliant, with features like role-based access
controls, data encryption,
        and integration with other AWS services for additional security and
compliance features.

        Let's get started...
```

```

        """);

        waitForInputToContinue(scanner);
        logger.info(DASHES);

        try {
            runScenario(assetModelName, assetName, portalName, contactEmail,
gatewayName, myThing);
        } catch (RuntimeException e) {
            logger.info(e.getMessage());
        }
    }

    public static void runScenario(String assetModelName, String assetName,
String portalName, String contactEmail, String gatewayName, String myThing)
throws Throwable {
        logger.info("Use AWS CloudFormation to create an IAM role that is
required for this scenario.");
        CloudFormationHelper.deployCloudFormationStack(ROLES_STACK);
        Map<String, String> stackOutputs =
CloudFormationHelper.getStackOutputsAsync(ROLES_STACK).join();
        String iamRole = stackOutputs.get("SitewiseRoleArn");
        logger.info("The ARN of the IAM role is {}", iamRole);
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("1. Create an AWS SiteWise Asset Model");
        logger.info("""
            An AWS IoT SiteWise Asset Model is a way to represent the physical
assets, such as equipment,
            processes, and systems, that exist in an industrial environment.
This model provides a structured and
            hierarchical representation of these assets, allowing users to
define the relationships and properties
            of each asset.

            This scenario creates two asset model properties: temperature and
humidity.
        """);
        waitForInputToContinue(scanner);
        String assetModelId = null;
        try {
            CreateAssetModelResponse response =
sitewiseActions.createAssetModelAsync(assetModelName).join();

```



```

        assetModelId = response.assetModelId();
        logger.info("Asset Model successfully created. Asset Model ID: {}. ",
assetModelId);
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceAlreadyExistsException) {
            try {
                assetModelId =
sitewiseActions.getAssetModelIdAsync(assetModelName).join();
                logger.info("The Asset Model {} already exists. The id of the
existing model is {}. Moving on...", assetModelName, assetModelId);
            } catch (CompletionException cex) {
                logger.error("Exception thrown acquiring the asset model id:
{}", cex.getCause().getCause(), cex);
                return;
            }
        } else {
            logger.info("An unexpected error occurred: " +
cause.getMessage(), cause);
            return;
        }
    }
    waitForInputToContinue(scanner);

    logger.info(DASHES);
    logger.info("2. Create an AWS IoT SiteWise Asset");
    logger.info("""
        The IoT SiteWise model that we just created defines the structure
and metadata for your physical assets.
        Now we create an asset from the asset model.

        """);
    logger.info("Let's wait 30 seconds for the asset to be ready.");
    countdown(30);
    waitForInputToContinue(scanner);
    String assetId;
    try {
        CreateAssetResponse response =
sitewiseActions.createAssetAsync(assetName, assetModelId).join();
        assetId = response.assetId();
        logger.info("Asset created with ID: {}", assetId);
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceNotFoundException) {

```

```

        logger.info("The asset model id was not found: {}",
cause.getMessage(), cause);
    } else {
        logger.info("An unexpected error occurred: {}",
cause.getMessage(), cause);
    }
    return;
}
waitForInputToContinue(scanner);
logger.info(DASHES);

logger.info(DASHES);
logger.info("3. Retrieve the property ID values");
logger.info(""""
    To send data to an asset, we need to get the property ID values. In
this scenario, we access the
    temperature and humidity property ID values.
    """);
waitForInputToContinue(scanner);
Map<String, String> propertyIds = null;
try {
    propertyIds = sitewiseActions.getPropertyIds(assetModelId).join();
} catch (CompletionException ce) {
    Throwable cause = ce.getCause();
    if (cause instanceof IoTSiteWiseException) {
        logger.error("IoTSiteWiseException occurred: {}",
cause.getMessage(), ce);
    } else {
        logger.error("An unexpected error occurred: {}",
cause.getMessage(), ce);
    }
    return;
}
String humPropId = propertyIds.get("Humidity");
logger.info("The Humidity property Id is {}", humPropId);
String tempPropId = propertyIds.get("Temperature");
logger.info("The Temperature property Id is {}", tempPropId);

waitForInputToContinue(scanner);
logger.info(DASHES);

logger.info(DASHES);
logger.info("4. Send data to an AWS IoT SiteWise Asset");
logger.info(""""

```

By sending data to an IoT SiteWise Asset, you can aggregate data from multiple sources, normalize the data into a standard format, and store it in a centralized location. This makes it easier to analyze and gain insights from the data.

In this example, we generate sample temperature and humidity data and send it to the AWS IoT SiteWise asset.

```

        """);
        waitForInputToContinue(scanner);
        try {
            sitewiseActions.sendDataToSiteWiseAsync(assetId, tempPropId,
            humPropId).join();
            logger.info("Data sent successfully.");
        } catch (CompletionException ce) {
            Throwable cause = ce.getCause();
            if (cause instanceof ResourceNotFoundException) {
                logger.error("The AWS resource was not found: {}",
                cause.getMessage(), cause);
            } else {
                logger.error("An unexpected error occurred: {}",
                cause.getMessage(), cause);
            }
            return;
        }
        waitForInputToContinue(scanner);
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("5. Retrieve the value of the IoT SiteWise Asset property");
        logger.info("""
            IoT SiteWise is an AWS service that allows you to collect, process,
            and analyze industrial data
            from connected equipment and sensors. One of the key benefits of
            reading an IoT SiteWise property
            is the ability to gain valuable insights from your industrial data.

            """);
        waitForInputToContinue(scanner);
        try {
            Double assetVal = sitewiseActions.getAssetPropValueAsync(tempPropId,
            assetId).join();

```

```

        logger.info("The property name is: {}", "Temperature");
        logger.info("The value of this property is: {}", assetVal);

        waitForInputToContinue(scanner);

        assetVal = sitewiseActions.getAssetPropValueAsync(humPropId,
assetId).join();
        logger.info("The property name is: {}", "Humidity");
        logger.info("The value of this property is: {}", assetVal);
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceNotFoundException) {
            logger.info("The AWS resource was not found: {}",
cause.getMessage(), cause);
        } else {
            logger.info("An unexpected error occurred: {}",
cause.getMessage(), cause);
        }
        return;
    }
    waitForInputToContinue(scanner);
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("6. Create an IoT SiteWise Gateway");
    logger.info(
        """"
            IoT SiteWise Gateway serves as the bridge between industrial
equipment, sensors, and the
            cloud-based IoT SiteWise service. It is responsible for securely
collecting, processing, and
            transmitting data from various industrial assets to the IoT
SiteWise platform,
            enabling real-time monitoring, analysis, and optimization of
industrial operations.

            """);
    waitForInputToContinue(scanner);
    String gatewayId = "";
    try {
        gatewayId = sitewiseActions.createGatewayAsync(gatewayName,
myThing).join();
        logger.info("Gateway creation completed successfully. id is {}",
gatewayId );
    }

```

```

        } catch (CompletionException ce) {
            Throwable cause = ce.getCause();
            if (cause instanceof IoTSiteWiseException siteWiseEx) {
                logger.error("IoT SiteWise error occurred: Error message: {},
Error code {}",
                    siteWiseEx.getMessage(),
siteWiseEx.awsErrorDetails().errorCode(), siteWiseEx);
            } else {
                logger.error("An unexpected error occurred: {}",
cause.getMessage());
            }
            return;
        }
        logger.info(DASHES);
        logger.info(DASHES);

        logger.info("7. Describe the IoT SiteWise Gateway");
        waitForInputToContinue(scanner);
        try {
            sitewiseActions.describeGatewayAsync(gatewayId)
                .thenAccept(response -> {
                    logger.info("Gateway Name: {}", response.gatewayName());
                    logger.info("Gateway ARN: {}", response.gatewayArn());
                    logger.info("Gateway Platform: {}",
response.gatewayPlatform());
                    logger.info("Gateway Creation Date: {}",
response.creationDate());
                }).join();
        } catch (CompletionException ce) {
            Throwable cause = ce.getCause();
            if (cause instanceof ResourceNotFoundException notFoundException) {
                logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                    notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
            } else {
                logger.error("An unexpected error occurred: {}",
cause.getMessage(), cause);
            }
            return;
        }
        logger.info(DASHES);

        logger.info(DASHES);

```

```

        logger.info("8. Delete the AWS IoT SiteWise Assets");
        logger.info(
            ""
            Before you can delete the Asset Model, you must delete the assets.

            "");
        logger.info("Would you like to delete the IoT SiteWise Assets? (y/n)");
        String delAns = scanner.nextLine().trim();
        if (delAns.equalsIgnoreCase("y")) {
            logger.info("You selected to delete the SiteWise assets.");

            try {
                sitewiseActions.deleteGatewayAsync(gatewayId).join();
                logger.info("Gateway {} was deleted successfully.", gatewayId);
            } catch (CompletionException ce) {
                Throwable cause = ce.getCause();
                if (cause instanceof ResourceNotFoundException notFoundException)
            {
                logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                    notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
            } else {
                logger.error("An unexpected error occurred: {}",
cause.getMessage());
            }
        }

        try {
            sitewiseActions.deleteAssetAsync(assetId).join();
            logger.info("Request to delete asset {} sent successfully",
assetId);
        } catch (CompletionException ce) {
            Throwable cause = ce.getCause();
            if (cause instanceof ResourceNotFoundException notFoundException)
        {
            logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
            } else {
                logger.error("An unexpected error occurred: {}",
cause.getMessage());
            }
        }
    }

```

```

    }
    logger.info("Let's wait 1 minute for the asset to be deleted.");
    countdown(60);
    waitForInputToContinue(scanner);
    logger.info("Delete the AWS IoT SiteWise Asset Model");
    try {
        sitewiseActions.deleteAssetModelAsync(assetModelId).join();
        logger.info("Asset model deleted successfully.");
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceNotFoundException notFoundException)
        {
            logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                        notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
        } else {
            logger.error("An unexpected error occurred: {}",
cause.getMessage());
        }
    }
    waitForInputToContinue(scanner);

} else {
    logger.info("The resources will not be deleted.");
}
logger.info(DASHES);

logger.info(DASHES);
CloudFormationHelper.destroyCloudFormationStack(ROLES_STACK);
logger.info("This concludes the AWS IoT SiteWise Scenario");
logger.info(DASHES);
}

private static void waitForInputToContinue(Scanner scanner) {
    while (true) {
        logger.info("");
        logger.info("Enter 'c' followed by <ENTER> to continue:");
        String input = scanner.nextLine();

        if (input.trim().equalsIgnoreCase("c")) {
            logger.info("Continuing with the program...");
            logger.info("");
            break;
        }
    }
}

```

```

        } else {
            logger.info("Invalid input. Please try again.");
        }
    }
}

public static void countdown(int totalSeconds) throws InterruptedException {
    for (int i = totalSeconds; i >= 0; i--) {
        int displayMinutes = i / 60;
        int displaySeconds = i % 60;
        System.out.printf("\r%02d:%02d", displayMinutes, displaySeconds);
        Thread.sleep(1000); // Wait for 1 second
    }
    System.out.println(); // Move to the next line after countdown
    logger.info("Countdown complete!");
}
}

```

Eine Wrapper-Klasse für AWS IoT SiteWise SDK-Methoden.

```

public class SitewiseActions {

    private static final Logger logger =
        LoggerFactory.getLogger(SitewiseActions.class);

    private static IoTSiteWiseAsyncClient ioTSiteWiseAsyncClient;

    private static IoTSiteWiseAsyncClient getAsyncClient() {
        if (ioTSiteWiseAsyncClient == null) {
            SdkAsyncHttpClient httpClient = NettyNioAsyncHttpClient.builder()
                .maxConcurrency(100)
                .connectionTimeout(Duration.ofSeconds(60))
                .readTimeout(Duration.ofSeconds(60))
                .writeTimeout(Duration.ofSeconds(60))
                .build();

            ClientOverrideConfiguration overrideConfig =
                ClientOverrideConfiguration.builder()
                    .apiCallTimeout(Duration.ofMinutes(2))
                    .apiCallAttemptTimeout(Duration.ofSeconds(90))
                    .retryStrategy(RetryMode.STANDARD)
                    .build();

```



```

        ioTSiteWiseAsyncClient = IoTSiteWiseAsyncClient.builder()
            .httpClient(httpClient)
            .overrideConfiguration(overrideConfig)
            .build();
    }
    return ioTSiteWiseAsyncClient;
}

/**
 * Creates an asset model.
 *
 * @param name the name of the asset model to create.
 * @return a {@link CompletableFuture} that represents a {@link
CreateAssetModelResponse} result. The calling code
 *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *
 * <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
 *         available to the calling code as a {@link CompletionException}. By
calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<CreateAssetModelResponse>
createAssetModelAsync(String name) {
    PropertyType humidity = PropertyType.builder()
        .measurement(Measurement.builder().build())
        .build();

    PropertyType temperaturePropertyType = PropertyType.builder()
        .measurement(Measurement.builder().build())
        .build();

    AssetModelPropertyDefinition temperatureProperty =
AssetModelPropertyDefinition.builder()
        .name("Temperature")
        .dataType(PropertyDataType.DOUBLE)
        .type(temperaturePropertyType)
        .build();

```

```

        AssetModelPropertyDefinition humidityProperty =
AssetModelPropertyDefinition.builder()
    .name("Humidity")
    .dataType(PropertyDataType.DOUBLE)
    .type(humidity)
    .build();

        CreateAssetModelRequest createAssetModelRequest =
CreateAssetModelRequest.builder()
    .assetModelName(name)
    .assetModelDescription("This is my asset model")
    .assetModelProperties(temperatureProperty, humidityProperty)
    .build();

        return getAsyncClient().createAssetModel(createAssetModelRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("Failed to create asset model: {} ",
exception.getCause().getMessage());
                }
            });
    }

}

/**
 * Creates an asset with the specified name and asset model Id.
 *
 * @param assetName the name of the asset to create.
 * @param assetModelId the Id of the asset model to associate with the asset.
 * @return a {@link CompletableFuture} that represents a {@link
CreateAssetResponse} result. The calling code can
 *      attach callbacks, then handle the result or exception by calling
{@link CompletableFuture#join()} or
 *      {@link CompletableFuture#get()}.
 *      <p>
 *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
 *      available to the calling code as a {@link CompletionException}. By
calling
 *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<CreateAssetResponse> createAssetAsync(String
assetName, String assetModelId) {

```

```

        CreateAssetRequest createAssetRequest = CreateAssetRequest.builder()
            .assetModelId(assetModelId)
            .assetDescription("Created using the AWS SDK for Java")
            .assetName(assetName)
            .build();

        return getAsyncClient().createAsset(createAssetRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("Failed to create asset: {}",
exception.getCause().getMessage());
                }
            });
    }

    /**
     * Sends data to the SiteWise service.
     *
     * @param assetId          the ID of the asset to which the data will be sent.
     * @param tempPropertyId the ID of the temperature property.
     * @param humidityPropId the ID of the humidity property.
     * @return a {@link CompletableFuture} that represents a {@link
BatchPutAssetPropertyValueResponse} result. The
     *         calling code can attach callbacks, then handle the result or
exception by calling
     *         {@link CompletableFuture#join()} or {@link
CompletableFuture#get()}.
     *
     * <p>
     *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
     *         available to the calling code as a {@link CompletionException}. By
calling
     *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
     */
    public CompletableFuture<BatchPutAssetPropertyValueResponse>
sendDataToSiteWiseAsync(String assetId, String tempPropertyId, String
humidityPropId) {
        Map<String, Double> sampleData = generateSampleData();
        long timestamp = Instant.now().toEpochMilli();

        TimeInNanos time = TimeInNanos.builder()
            .timeInSeconds(timestamp / 1000)
            .offsetInNanos((int) ((timestamp % 1000) * 1000000))

```

```

        .build();

        BatchPutAssetPropertyValueRequest request =
        BatchPutAssetPropertyValueRequest.builder()
            .entries(Arrays.asList(
                PutAssetPropertyValueEntry.builder()
                    .entryId("entry-3")
                    .assetId(assetId)
                    .propertyId(tempPropertyId)
                    .propertyValues(Arrays.asList(
                        AssetPropertyValue.builder()
                            .value(Variant.builder()
                                .doubleValue(sampleData.get("Temperature"))
                                .build())
                            .timestamp(time)
                            .build()
                    ))
                    .build(),
                PutAssetPropertyValueEntry.builder()
                    .entryId("entry-4")
                    .assetId(assetId)
                    .propertyId(humidityPropId)
                    .propertyValues(Arrays.asList(
                        AssetPropertyValue.builder()
                            .value(Variant.builder()
                                .doubleValue(sampleData.get("Humidity"))
                                .build())
                            .timestamp(time)
                            .build()
                    ))
                    .build()
            ))
            .build();

        return getAsyncClient().batchPutAssetPropertyValue(request)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("An exception occurred: {}",
exception.getCause().getMessage());
                }
            });
    }

    /**

```

```

    * Fetches the value of an asset property.
    *
    * @param propId the ID of the asset property to fetch.
    * @param assetId the ID of the asset to fetch the property value for.
    * @return a {@link CompletableFuture} that represents a {@link Double}
result. The calling code can attach
    *     callbacks, then handle the result or exception by calling {@link
CompletableFuture#join()} or
    *     {@link CompletableFuture#get()}.
    *     <p>
    *     If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
    *     it available to the calling code as a {@link CompletionException}.
By calling
    *     {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<Double> getAssetPropValueAsync(String propId, String
assetId) {
        GetAssetPropertyValueRequest assetPropertyValueRequest =
GetAssetPropertyValueRequest.builder()
            .propertyId(propId)
            .assetId(assetId)
            .build();

        return getAsyncClient().getAssetPropertyValue(assetPropertyValueRequest)
            .handle((response, exception) -> {
                if (exception != null) {
                    logger.error("Error occurred while fetching property
value: {}.", exception.getCause().getMessage());
                    throw (CompletionException) exception;
                }
                return response.propertyValue().value().doubleValue();
            });
    }

    /**
    * Retrieves the property IDs associated with a specific asset model.
    *
    * @param assetModelId the ID of the asset model that defines the properties.
    * @return a {@link CompletableFuture} that represents a {@link Map} result
that associates the property name to the
    *     propert ID. The calling code can attach callbacks, then handle the
result or exception by calling

```

```

    *      {@link CompletableFuture#join()} or {@link
CompletableFuture#get()}.
    *      <p>
    *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
    *      it available to the calling code as a {@link CompletionException}.
By calling
    *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<Map<String, String>> getPropertyIds(String
assetModelId) {
        ListAssetModelPropertiesRequest modelPropertiesRequest =
ListAssetModelPropertiesRequest.builder().assetModelId(assetModelId).build();
        return getAsyncClient().listAssetModelProperties(modelPropertiesRequest)
            .handle((response, throwable) -> {
                if (response != null) {
                    return response.assetModelPropertySummaries().stream()
                        .collect(Collectors
                            .toMap(AssetModelPropertySummary::name,
AssetModelPropertySummary::id));
                } else {
                    logger.error("Error occurred while fetching property IDs:
{}. ", throwable.getCause().getMessage());
                    throw (CompletionException) throwable;
                }
            });
    }

    /**
    * Deletes an asset.
    *
    * @param assetId the ID of the asset to be deleted.
    * @return a {@link CompletableFuture} that represents a {@link
DeleteAssetResponse} result. The calling code can
    *      attach callbacks, then handle the result or exception by calling
{@link CompletableFuture#join()} or
    *      {@link CompletableFuture#get()}.
    *      <p>
    *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
    *      it available to the calling code as a {@link CompletionException}.
By calling

```

```

        *      {@link CompletionException#getCause()}, the calling code can
        access the original exception.
        */
        public CompletableFuture<DeleteAssetResponse> deleteAssetAsync(String
        assetId) {
            DeleteAssetRequest deleteAssetRequest = DeleteAssetRequest.builder()
                .assetId(assetId)
                .build();

            return getAsyncClient().deleteAsset(deleteAssetRequest)
                .whenComplete((response, exception) -> {
                    if (exception != null) {
                        logger.error("An error occurred deleting asset with id: {}",
assetId);
                    }
                });
        }

    /**
     * Deletes an Asset Model with the specified ID.
     *
     * @param assetModelId the ID of the Asset Model to delete.
     * @return a {@link CompletableFuture} that represents a {@link
DeleteAssetModelResponse} result. The calling code
     *      can attach callbacks, then handle the result or exception by
     *      calling {@link CompletableFuture#join()} or
     *      {@link CompletableFuture#get()}.
     *
     *      <p>
     *      If any completion stage in this method throws an exception, the
     *      method logs the exception cause and keeps
     *      it available to the calling code as a {@link CompletionException}.
     *      By calling
     *      {@link CompletionException#getCause()}, the calling code can
     *      access the original exception.
     *      */
    public CompletableFuture<DeleteAssetModelResponse>
deleteAssetModelAsync(String assetModelId) {
        DeleteAssetModelRequest deleteAssetModelRequest =
DeleteAssetModelRequest.builder()
            .assetModelId(assetModelId)
            .build();

        return getAsyncClient().deleteAssetModel(deleteAssetModelRequest)
            .whenComplete((response, exception) -> {

```

```

        if (exception != null) {
            logger.error("Failed to delete asset model with ID:{}.",
exception.getMessage());
        }
    });
}

/**
 * Retrieves the asset model ID for the given asset model name.
 *
 * @param assetModelName the name of the asset model for the ID.
 * @return a {@link CompletableFuture} that represents a {@link String}
result of the asset model ID or null if the
 *      asset model cannot be found. The calling code can attach
callbacks, then handle the result or exception
 *      by calling {@link CompletableFuture#join()} or {@link
CompletableFuture#get()}.
 *      <p>
 *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *      it available to the calling code as a {@link CompletionException}.
By calling
 *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<String> getAssetModelIdAsync(String assetModelName)
{
    ListAssetModelsRequest listAssetModelsRequest =
ListAssetModelsRequest.builder().build();
    return getAsyncClient().listAssetModels(listAssetModelsRequest)
        .handle((listAssetModelsResponse, exception) -> {
            if (exception != null) {
                logger.error("Failed to retrieve Asset Model ID: {}",
exception.getCause().getMessage());
                throw (CompletionException) exception;
            }
            for (AssetModelSummary assetModelSummary :
listAssetModelsResponse.assetModelSummaries()) {
                if (assetModelSummary.name().equals(assetModelName)) {
                    return assetModelSummary.id();
                }
            }
            return null;
        });
}

```



```

    }

    /**
     * Creates a new IoT Sitewise gateway.
     *
     * @param gatewayName The name of the gateway to create.
     * @param myThing      The name of the core device thing to associate with the
     gateway.
     * @return a {@link CompletableFuture} that represents a {@link String}
     result of the gateways ID. The calling code
     *         can attach callbacks, then handle the result or exception by
     calling {@link CompletableFuture#join()} or
     *         {@link CompletableFuture#get()}.
     *         <p>
     *         If any completion stage in this method throws an exception, the
     method logs the exception cause and keeps
     *         it available to the calling code as a {@link CompletionException}.
     By calling
     *         {@link CompletionException#getCause()}, the calling code can
     access the original exception.
     */
    public CompletableFuture<String> createGatewayAsync(String gatewayName,
String myThing) {
        GreengrassV2 gg = GreengrassV2.builder()
            .coreDeviceThingName(myThing)
            .build();

        GatewayPlatform platform = GatewayPlatform.builder()
            .greengrassV2(gg)
            .build();

        Map<String, String> tag = new HashMap<>();
        tag.put("Environment", "Production");

        CreateGatewayRequest createGatewayRequest =
CreateGatewayRequest.builder()
            .gatewayName(gatewayName)
            .gatewayPlatform(platform)
            .tags(tag)
            .build();

        return getAsyncClient().createGateway(createGatewayRequest)
            .handle((response, exception) -> {

```

```

        if (exception != null) {
            logger.error("Error creating the gateway.");
            throw (CompletionException) exception;
        }
        logger.info("The ARN of the gateway is {}" ,
response.gatewayArn());
        return response.gatewayId();
    });
}

/**
 * Deletes the specified gateway.
 *
 * @param gatewayId the ID of the gateway to delete.
 * @return a {@link CompletableFuture} that represents a {@link
DeleteGatewayResponse} result.. The calling code
 *      can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *      {@link CompletableFuture#get()}.
 *      <p>
 *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *      it available to the calling code as a {@link CompletionException}.
By calling
 *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<DeleteGatewayResponse> deleteGatewayAsync(String
gatewayId) {
    DeleteGatewayRequest deleteGatewayRequest =
DeleteGatewayRequest.builder()
        .gatewayId(gatewayId)
        .build();

    return getAsyncClient().deleteGateway(deleteGatewayRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to delete gateway: {}",
exception.getCause().getMessage());
            }
        });
}

/**

```

```

    * Describes the specified gateway.
    *
    * @param gatewayId the ID of the gateway to describe.
    * @return a {@link CompletableFuture} that represents a {@link
DescribeGatewayResponse} result. The calling code
    *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
    *         {@link CompletableFuture#get()}.
    *         <p>
    *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
    *         it available to the calling code as a {@link CompletionException}.
By calling
    *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<DescribeGatewayResponse> describeGatewayAsync(String
gatewayId) {
        DescribeGatewayRequest request = DescribeGatewayRequest.builder()
            .gatewayId(gatewayId)
            .build();

        return getAsyncClient().describeGateway(request)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("An error occurred during the describeGateway
method: {}", exception.getCause().getMessage());
                }
            });
    }

    private static Map<String, Double> generateSampleData() {
        Map<String, Double> data = new HashMap<>();
        data.put("Temperature", 23.5);
        data.put("Humidity", 65.0);
        return data;
    }
}

```

- Weitere API-Informationen finden Sie in den folgenden Themen der AWS SDK for Java 2.x - API-Referenz.
- [BatchPutAssetPropertyValue](#)

- [CreateAsset](#)
- [CreateAssetModel](#)
- [CreateGateway](#)
- [DeleteAsset](#)
- [DeleteAssetModel](#)
- [DeleteGateway](#)
- [DescribeAssetModel](#)
- [DescribeGateway](#)
- [GetAssetPropertyValue](#)
- [ListAssetModelProperties](#)
- [ListAssetModels](#)

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
import {
  Scenario,
  ScenarioAction,
  ScenarioInput,
  ScenarioOutput,
  //} from "@aws-doc-sdk-examples/lib/scenario/index.js";
} from "../../libs/scenario/index.js";
import {
  IoTSiteWiseClient,
  CreateAssetModelCommand,
  CreateAssetCommand,
  ListAssetModelPropertiesCommand,
  BatchPutAssetPropertyValueCommand,
  GetAssetPropertyValueCommand,
  CreatePortalCommand,
```

```

    DescribePortalCommand,
    CreateGatewayCommand,
    DescribeGatewayCommand,
    DeletePortalCommand,
    DeleteGatewayCommand,
    DeleteAssetCommand,
    DeleteAssetModelCommand,
    DescribeAssetModelCommand,
} from "@aws-sdk/client-iotsitewise";
import {
    CloudFormationClient,
    CreateStackCommand,
    DeleteStackCommand,
    DescribeStacksCommand,
    waitUntilStackExists,
    waitUntilStackCreateComplete,
    waitUntilStackDeleteComplete,
} from "@aws-sdk/client-cloudformation";
import { wait } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";
import { parseArgs } from "node:util";
import { readFileSync } from "node:fs";
import { fileURLToPath } from "node:url";
import { dirname } from "node:path";

const __filename = fileURLToPath(import.meta.url);
const __dirname = dirname(__filename);
const stackName = "SiteWiseBasicsStack";

/**
 * @typedef {
 *   iotSiteWiseClient: import('@aws-sdk/client-iotsitewise').IotSiteWiseClient,
 *   cloudFormationClient: import('@aws-sdk/client-
cloudformation').CloudFormationClient,
 *   stackName,
 *   stack,
 *   askToDeleteResources: true,
 *   asset: {assetName: "MyAsset1"},
 *   assetModel: {assetModelName: "MyAssetModel1"},
 *   portal: {portalName: "MyPortal1"},
 *   gateway: {gatewayName: "MyGateway1"},
 *   propertyIds: [],
 *   contactEmail: "user@mydomain.com",
 *   thing: "MyThing1",
 *   sampleData: { temperature: 23.5, humidity: 65.0}

```

```

* }} State
*/

/**
 * Used repeatedly to have the user press enter.
 * @type {ScenarioInput}
 */
const pressEnter = new ScenarioInput("continue", "Press Enter to continue", {
  type: "confirm",
});

const greet = new ScenarioOutput(
  "greet",
  `AWS IoT SiteWise is a fully managed industrial software-as-a-service (SaaS)
  that makes it easy to collect, store, organize, and monitor data from industrial
  equipment and processes. It is designed to help industrial and manufacturing
  organizations collect data from their equipment and processes, and use that data
  to make informed decisions about their operations.
  One of the key features of AWS IoT SiteWise is its ability to connect to a
  wide range of industrial equipment and systems, including programmable logic
  controllers (PLCs), sensors, and other industrial devices. It can collect data
  from these devices and organize it into a unified data model, making it easier
  to analyze and gain insights from the data. AWS IoT SiteWise also provides tools
  for visualizing the data, setting up alarms and alerts, and generating reports.
  Another key feature of AWS IoT SiteWise is its ability to scale to handle large
  volumes of data. It can collect and store data from thousands of devices and
  process millions of data points per second, making it suitable for large-scale
  industrial operations. Additionally, AWS IoT SiteWise is designed to be secure
  and compliant, with features like role-based access controls, data encryption,
  and integration with other AWS services for additional security and compliance
  features.

  Let's get started...`,
  { header: true },
);

const displayBuildCloudFormationStack = new ScenarioOutput(
  "displayBuildCloudFormationStack",
  "This scenario uses AWS CloudFormation to create an IAM role that is required
  for this scenario. The stack will now be deployed.",
);

const sdkBuildCloudFormationStack = new ScenarioAction(
  "sdkBuildCloudFormationStack",

```

```

    async (** @type {State} */ state) => {
      try {
        const data = readFileSync(
          `${__dirname}/../../../../../resources/cfn/iotsitewise_basics/SitewiseRoles-
            template.yml`,
          "utf8",
        );
        await state.cloudFormationClient.send(
          new CreateStackCommand({
            StackName: stackName,
            TemplateBody: data,
            Capabilities: ["CAPABILITY_IAM"],
          }),
        );
        await waitUntilStackExists(
          { client: state.cloudFormationClient },
          { StackName: stackName },
        );
        await waitUntilStackCreateComplete(
          { client: state.cloudFormationClient },
          { StackName: stackName },
        );
        const stack = await state.cloudFormationClient.send(
          new DescribeStacksCommand({
            StackName: stackName,
          }),
        );
        state.stack = stack.Stacks[0].Outputs[0];
        console.log(`The ARN of the IAM role is ${state.stack.OutputValue}`);
      } catch (caught) {
        console.error(caught.message);
        throw caught;
      }
    },
  );

  const displayCreateAWSSiteWiseAssetModel = new ScenarioOutput(
    "displayCreateAWSSiteWiseAssetModel",
    `1. Create an AWS SiteWise Asset Model

```

An AWS IoT SiteWise Asset Model is a way to represent the physical assets, such as equipment, processes, and systems, that exist in an industrial environment. This model provides a structured and hierarchical representation of these assets, allowing users to define the relationships and properties of each asset.

```

This scenario creates two asset model properties: temperature and humidity.`
);

const sdkCreateAWSSiteWiseAssetModel = new ScenarioAction(
  "sdkCreateAWSSiteWiseAssetModel",
  async (/** @type {State} */ state) => {
    let assetModelResponse;
    try {
      assetModelResponse = await state.iotSiteWiseClient.send(
        new CreateAssetModelCommand({
          assetModelName: state.assetModel.assetModelName,
          assetModelProperties: [
            {
              name: "Temperature",
              dataType: "DOUBLE",
              type: {
                measurement: {},
              },
            },
            {
              name: "Humidity",
              dataType: "DOUBLE",
              type: {
                measurement: {},
              },
            },
          ],
        })
      );
      state.assetModel.assetModelId = assetModelResponse.assetModelId;
      console.log(
        `Asset Model successfully created. Asset Model ID:
        ${state.assetModel.assetModelId}`,
      );
    } catch (caught) {
      if (caught.name === "ResourceAlreadyExistsException") {
        console.log(
          `The Asset Model ${state.assetModel.assetModelName} already exists.`
        );
        throw caught;
      }
      console.error(`${caught.message}`);
      throw caught;
    }
  }
}

```



```

    },
  );

const displayCreateAWSIoTSiteWiseAssetModel = new ScenarioOutput(
  "displayCreateAWSIoTSiteWiseAssetModel",
  `2. Create an AWS IoT SiteWise Asset
The IoT SiteWise model that we just created defines the structure and metadata
for your physical assets. Now we create an asset from the asset model.

Let's wait 30 seconds for the asset to be ready.`
);

const waitThirtySeconds = new ScenarioAction("waitThirtySeconds", async () => {
  await wait(30); // wait 30 seconds
  console.log("Time's up! Let's check the asset's status.");
});

const sdkCreateAWSIoTSiteWiseAssetModel = new ScenarioAction(
  "sdkCreateAWSIoTSiteWiseAssetModel",
  async (** @type {State} */ state) => {
    try {
      const assetResponse = await state.iotSiteWiseClient.send(
        new CreateAssetCommand({
          assetModelId: state.assetModel.assetModelId,
          assetName: state.asset.assetName,
        })
      );
      state.asset.assetId = assetResponse.assetId;
      console.log(`Asset created with ID: ${state.asset.assetId}`);
    } catch (caught) {
      if (caught.name === "ResourceNotFoundException") {
        console.log(
          `The Asset ${state.assetModel.assetModelName} was not found.`
        );
        throw caught;
      }
      console.error(`${caught.message}`);
      throw caught;
    }
  },
);

const displayRetrievePropertyId = new ScenarioOutput(
  "displayRetrievePropertyId",

```

`3. Retrieve the property ID values

To send data to an asset, we need to get the property ID values. In this scenario, we access the temperature and humidity property ID values.`
);

```
const sdkRetrievePropertyId = new ScenarioAction(
  "sdkRetrievePropertyId",
  async (state) => {
    try {
      const retrieveResponse = await state.iotSiteWiseClient.send(
        new ListAssetModelPropertySummariesCommand({
          assetModelId: state.assetModel.assetModelId,
        }),
      );
      for (const retrieveResponseKey in
retrieveResponse.assetModelPropertySummaries) {
        if (
          retrieveResponse.assetModelPropertySummaries[retrieveResponseKey]
            .name === "Humidity"
        ) {
          state.propertyIds.Humidity =
            retrieveResponse.assetModelPropertySummaries[
              retrieveResponseKey
            ].id;
        }
        if (
          retrieveResponse.assetModelPropertySummaries[retrieveResponseKey]
            .name === "Temperature"
        ) {
          state.propertyIds.Temperature =
            retrieveResponse.assetModelPropertySummaries[
              retrieveResponseKey
            ].id;
        }
      }
      console.log(`The Humidity propertyId is ${state.propertyIds.Humidity}`);
      console.log(
        `The Temperature propertyId is ${state.propertyIds.Temperature}`,
      );
    } catch (caught) {
      if (caught.name === "IoTSiteWiseException") {
        console.log(
          `There was a problem retrieving the properties: ${caught.message}`,
        );
      }
    }
  }
);
```

```

        );
        throw caught;
    }
    console.error(`${caught.message}`);
    throw caught;
}
},
);

```

```

const displaySendDataToIoTSiteWiseAsset = new ScenarioOutput(
  "displaySendDataToIoTSiteWiseAsset",
  `4. Send data to an AWS IoT SiteWise Asset

```

By sending data to an IoT SiteWise Asset, you can aggregate data from multiple sources, normalize the data into a standard format, and store it in a centralized location. This makes it easier to analyze and gain insights from the data.

In this example, we generate sample temperature and humidity data and send it to the AWS IoT SiteWise asset.`,

```

);

```

```

const sdkSendDataToIoTSiteWiseAsset = new ScenarioAction(
  "sdkSendDataToIoTSiteWiseAsset",
  async (state) => {
    try {
      const sendResponse = await state.iotSiteWiseClient.send(
        new BatchPutAssetPropertyValueCommand({
          entries: [
            {
              entryId: "entry-3",
              assetId: state.asset.assetId,
              propertyId: state.propertyIds.Humidity,
              propertyValues: [
                {
                  value: {
                    doubleValue: state.sampleData.humidity,
                  },
                  timestamp: {
                    timeInSeconds: Math.floor(Date.now() / 1000),
                  },
                },
              ],
            },
          ],
        })
      );
    }
  },
);

```

```

        {
            entryId: "entry-4",
            assetId: state.asset.assetId,
            propertyId: state.propertyIds.Temperature,
            propertyValues: [
                {
                    value: {
                        doubleValue: state.sampleData.temperature,
                    },
                    timestamp: {
                        timeInSeconds: Math.floor(Date.now() / 1000),
                    },
                },
            ],
        },
    ],
    )),
);
console.log("The data was sent successfully.");
} catch (caught) {
    if (caught.name === "ResourceNotFoundException") {
        console.log(`The Asset ${state.asset.assetName} was not found.`);
        throw caught;
    }
    console.error(`${caught.message}`);
    throw caught;
}
},
);

const displayRetrieveValueOfIoTSiteWiseAsset = new ScenarioOutput(
    "displayRetrieveValueOfIoTSiteWiseAsset",
    `5. Retrieve the value of the IoT SiteWise Asset property

IoT SiteWise is an AWS service that allows you to collect, process, and analyze
industrial data from connected equipment and sensors. One of the key benefits of
reading an IoT SiteWise property is the ability to gain valuable insights from
your industrial data.`,
);

const sdkRetrieveValueOfIoTSiteWiseAsset = new ScenarioAction(
    "sdkRetrieveValueOfIoTSiteWiseAsset",
    async (/** @type {State} */ state) => {
        try {

```

```

    const temperatureResponse = await state.iotSiteWiseClient.send(
      new GetAssetPropertyValueCommand({
        assetId: state.asset.assetId,
        propertyId: state.propertyIds.Temperature,
      }),
    );
    const humidityResponse = await state.iotSiteWiseClient.send(
      new GetAssetPropertyValueCommand({
        assetId: state.asset.assetId,
        propertyId: state.propertyIds.Humidity,
      }),
    );
    console.log(
      `The property value for Temperature is
    ${temperatureResponse.propertyValue.value.doubleValue}`,
    );
    console.log(
      `The property value for Humidity is
    ${humidityResponse.propertyValue.value.doubleValue}`,
    );
  } catch (caught) {
    if (caught.name === "ResourceNotFoundException") {
      console.log(`The Asset ${state.asset.assetName} was not found.`);
      throw caught;
    }
    console.error(`${caught.message}`);
    throw caught;
  }
},
);

const displayCreateIoTSiteWisePortal = new ScenarioOutput(
  "displayCreateIoTSiteWisePortal",
  `6. Create an IoT SiteWise Portal

An IoT SiteWise Portal allows you to aggregate data from multiple industrial
sources, such as sensors, equipment, and control systems, into a centralized
platform.`,
);

const sdkCreateIoTSiteWisePortal = new ScenarioAction(
  "sdkCreateIoTSiteWisePortal",
  async (/** @type {State} */ state) => {
    try {

```

```

    const createPortalResponse = await state.iotSiteWiseClient.send(
      new CreatePortalCommand({
        portalName: state.portal.portalName,
        portalContactEmail: state.contactEmail,
        roleArn: state.stack.OutputValue,
      }),
    );
    state.portal = { ...state.portal, ...createPortalResponse };
    await wait(5); // Allow the portal to properly propagate.
    console.log(
      `Portal created successfully. Portal ID
    ${createPortalResponse.portalId}`,
    );
  } catch (caught) {
    if (caught.name === "IoTSiteWiseException") {
      console.log(
        `There was a problem creating the Portal: ${caught.message}.`,
      );
      throw caught;
    }
    console.error(`${caught.message}`);
    throw caught;
  }
},
);

const displayDescribePortal = new ScenarioOutput(
  "displayDescribePortal",
  `7. Describe the Portal

In this step, we get a description of the portal and display the portal URL.`,
);

const sdkDescribePortal = new ScenarioAction(
  "sdkDescribePortal",
  async (/** @type {State} */ state) => {
    try {
      const describePortalResponse = await state.iotSiteWiseClient.send(
        new DescribePortalCommand({
          portalId: state.portal.portalId,
        }),
      );
      console.log(`Portal URL: ${describePortalResponse.portalStartUrl}`);
    } catch (caught) {

```

```

        if (caught.name === "ResourceNotFoundException") {
            console.log(`The Portal ${state.portal.portalName} was not found.`);
            throw caught;
        }
        console.error(`${caught.message}`);
        throw caught;
    }
},
);

```

```

const displayCreateIoTSiteWiseGateway = new ScenarioOutput(
    "displayCreateIoTSiteWiseGateway",
    `8. Create an IoT SiteWise Gateway

```

IoT SiteWise Gateway serves as the bridge between industrial equipment, sensors, and the cloud-based IoT SiteWise service. It is responsible for securely collecting, processing, and transmitting data from various industrial assets to the IoT SiteWise platform, enabling real-time monitoring, analysis, and optimization of industrial operations.`,

```

);

```

```

const sdkCreateIoTSiteWiseGateway = new ScenarioAction(
    "sdkCreateIoTSiteWiseGateway",
    async (** @type {State} */ state) => {
        try {
            const createGatewayResponse = await state.iotSiteWiseClient.send(
                new CreateGatewayCommand({
                    gatewayName: state.gateway.gatewayName,
                    gatewayPlatform: {
                        greengrassV2: {
                            coreDeviceThingName: state.thing,
                        },
                    },
                }),
            );
            console.log(
                `Gateway creation completed successfully. ID is
                ${createGatewayResponse.gatewayId}`,
            );
            state.gateway.gatewayId = createGatewayResponse.gatewayId;
        } catch (caught) {
            if (caught.name === "IoTSiteWiseException") {
                console.log(
                    `There was a problem creating the gateway: ${caught.message}.`,
                );
            }
        }
    },
);

```

```

        );
        throw caught;
    }
    console.error(`${caught.message}`);
    throw caught;
}
},
);

const displayDescribeIoTSiteWiseGateway = new ScenarioOutput(
    "displayDescribeIoTSiteWiseGateway",
    "9. Describe the IoT SiteWise Gateway",
);

const sdkDescribeIoTSiteWiseGateway = new ScenarioAction(
    "sdkDescribeIoTSiteWiseGateway",
    async (** @type {State} */ state) => {
        try {
            const describeGatewayResponse = await state.iotSiteWiseClient.send(
                new DescribeGatewayCommand({
                    gatewayId: state.gateway.gatewayId,
                }),
            );
            console.log("Gateway creation completed successfully.");
            console.log(`Gateway Name: ${describeGatewayResponse.gatewayName}`);
            console.log(`Gateway ARN: ${describeGatewayResponse.gatewayArn}`);
            console.log(
                `Gateway Platform:
${Object.keys(describeGatewayResponse.gatewayPlatform)}`,
            );
            console.log(
                `Gateway Creation Date: ${describeGatewayResponse.creationDate}`,
            );
        } catch (caught) {
            if (caught.name === "ResourceNotFoundException") {
                console.log(`The Gateway ${state.gateway.gatewayId} was not found.`);
                throw caught;
            }
            console.error(`${caught.message}`);
            throw caught;
        }
    },
);

```



```
const askToDeleteResources = new ScenarioInput(
  "askToDeleteResources",
  `10. Delete the AWS IoT SiteWise Assets

Before you can delete the Asset Model, you must delete the assets.`
  { type: "confirm" },
);

const displayConfirmDeleteResources = new ScenarioAction(
  "displayConfirmDeleteResources",
  async (** @type {State} */ state) => {
    if (state.askToDeleteResources) {
      return "You selected to delete the SiteWise assets.";
    }
    return "The resources will not be deleted. Please delete them manually to
avoid charges.";
  },
);

const sdkDeleteResources = new ScenarioAction(
  "sdkDeleteResources",
  async (** @type {State} */ state) => {
    await wait(10); // Give the portal status time to catch up.
    try {
      await state.iotSiteWiseClient.send(
        new DeletePortalCommand({
          portalId: state.portal.portalId,
        }),
      );
      console.log(
        `Portal ${state.portal.portalName} was deleted successfully.`
      );
    } catch (caught) {
      if (caught.name === "ResourceNotFoundException") {
        console.log(`The Portal ${state.portal.portalName} was not found.`);
      } else {
        console.log(`When trying to delete the portal: ${caught.message}`);
      }
    }

    try {
      await state.iotSiteWiseClient.send(
        new DeleteGatewayCommand({
          gatewayId: state.gateway.gatewayId,
```

```
    }},
  );
  console.log(
    `Gateway ${state.gateway.gatewayName} was deleted successfully.`,
  );
} catch (caught) {
  if (caught.name === "ResourceNotFoundException") {
    console.log(`The Gateway ${state.gateway.gatewayId} was not found.`);
  } else {
    console.log(`When trying to delete the gateway: ${caught.message}`);
  }
}

try {
  await state.iotSiteWiseClient.send(
    new DeleteAssetCommand({
      assetId: state.asset.assetId,
    }),
  );
  await wait(5); // Allow the delete to finish.
  console.log(`Asset ${state.asset.assetName} was deleted successfully.`);
} catch (caught) {
  if (caught.name === "ResourceNotFoundException") {
    console.log(`The Asset ${state.asset.assetName} was not found.`);
  } else {
    console.log(`When deleting the asset: ${caught.message}`);
  }
}

await wait(30); // Allow asset deletion to finish.
try {
  await state.iotSiteWiseClient.send(
    new DeleteAssetModelCommand({
      assetModelId: state.assetModel.assetModelId,
    }),
  );
  console.log(
    `Asset Model ${state.assetModel.assetModelName} was deleted successfully.`,
  );
} catch (caught) {
  if (caught.name === "ResourceNotFoundException") {
    console.log(
      `The Asset Model ${state.assetModel.assetModelName} was not found.`,
    );
  }
}
```

```
    );
  } else {
    console.log(`When deleting the asset model: ${caught.message}`);
  }
}

try {
  await state.cloudFormationClient.send(
    new DeleteStackCommand({
      StackName: stackName,
    }),
  );
  await waitUntilStackDeleteComplete(
    { client: state.cloudFormationClient },
    { StackName: stackName },
  );
  console.log("The stack was deleted successfully.");
} catch (caught) {
  console.log(
    `${caught.message}. The stack was NOT deleted. Please clean up the resources manually.`
  );
}
},
{ skipWhen: (/** @type {} */ state) => !state.askToDeleteResources },
);

const goodbye = new ScenarioOutput(
  "goodbye",
  "This concludes the IoT SiteWise Basics scenario for the AWS Javascript SDK v3. Thank you!",
);

const myScenario = new Scenario(
  "IoTSiteWise Basics",
  [
    greet,
    pressEnter,
    displayBuildCloudFormationStack,
    sdkBuildCloudFormationStack,
    pressEnter,
    displayCreateAWSSiteWiseAssetModel,
    sdkCreateAWSSiteWiseAssetModel,
    displayCreateAWSIoTSiteWiseAssetModel,
```

```

    pressEnter,
    waitThirtySeconds,
    sdkCreateAWSIoTSiteWiseAssetModel,
    pressEnter,
    displayRetrievePropertyId,
    sdkRetrievePropertyId,
    pressEnter,
    displaySendDataToIoTSiteWiseAsset,
    sdkSendDataToIoTSiteWiseAsset,
    pressEnter,
    displayRetrieveValueOfIoTSiteWiseAsset,
    sdkRetrieveValueOfIoTSiteWiseAsset,
    pressEnter,
    displayCreateIoTSiteWisePortal,
    sdkCreateIoTSiteWisePortal,
    pressEnter,
    displayDescribePortal,
    sdkDescribePortal,
    pressEnter,
    displayCreateIoTSiteWiseGateway,
    sdkCreateIoTSiteWiseGateway,
    pressEnter,
    displayDescribeIoTSiteWiseGateway,
    sdkDescribeIoTSiteWiseGateway,
    pressEnter,
    askToDeleteResources,
    displayConfirmDeleteResources,
    sdkDeleteResources,
    goodbye,
  ],
  {
    iotSiteWiseClient: new IoTSiteWiseClient({}),
    cloudFormationClient: new CloudFormationClient({}),
    asset: { assetName: "MyAsset1" },
    assetModel: { assetModelName: "MyAssetModel1" },
    portal: { portalName: "MyPortal1" },
    gateway: { gatewayName: "MyGateway1" },
    propertyIds: [],
    contactEmail: "user@mydomain.com",
    thing: "MyThing1",
    sampleData: { temperature: 23.5, humidity: 65.0 },
  },
);

```

```
/** @type {{ stepHandlerOptions: StepHandlerOptions }} */
export const main = async (stepHandlerOptions) => {
  await myScenario.run(stepHandlerOptions);
};

// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  const { values } = parseArgs({
    options: {
      yes: {
        type: "boolean",
        short: "y",
      },
    },
  });
  main({ confirmAll: values.yes });
}
```

- Weitere API-Informationen finden Sie in den folgenden Themen der AWS SDK für JavaScript -API-Referenz.
 - [BatchPutAssetPropertyValue](#)
 - [CreateAsset](#)
 - [CreateAssetModel](#)
 - [CreateGateway](#)
 - [DeleteAsset](#)
 - [DeleteAssetModel](#)
 - [DeleteGateway](#)
 - [DescribeAssetModel](#)
 - [DescribeGateway](#)
 - [GetAssetPropertyValue](#)
 - [ListAssetModelProperties](#)
 - [ListAssetModels](#)

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

Führen Sie ein interaktives Szenario an einem Prompt aus.

```
class IoTSiteWiseGettingStarted:
    """
    A scenario that demonstrates how to use Boto3 to manage IoT physical assets
    using
    the AWS IoT SiteWise.
    """

    def __init__(
        self,
        iot_siteWise_wrapper: IoTSiteWiseWrapper,
        cloud_formation_resource: ServiceResource,
    ):
        self.iot_siteWise_wrapper = iot_siteWise_wrapper
        self.cloud_formation_resource = cloud_formation_resource
        self.stack = None
        self.asset_model_id = None
        self.asset_id = None
        self.portal_id = None
        self.gateway_id = None

    def run(self) -> None:
        """
        Runs the scenario.
        """
        print(
            """
AWS IoT SiteWise is a fully managed software-as-a-service (SaaS) that
makes it easy to collect, store, organize, and monitor data from industrial
equipment and processes.
It is designed to help industrial and manufacturing organizations collect data
from their equipment and
```

processes, and use that data to make informed decisions about their operations.

One of the key features of AWS IoT SiteWise is its ability to connect to a wide range of industrial equipment and systems, including programmable logic controllers (PLCs), sensors, and other industrial devices. It can collect data from these devices and organize it into a unified data model, making it easier to analyze and gain insights from the data. AWS IoT SiteWise also provides tools for visualizing the data, setting up alarms and alerts, and generating reports.

Another key feature of AWS IoT SiteWise is its ability to scale to handle large volumes of data.

It can collect and store data from thousands of devices and process millions of data points per second, making it suitable for large-scale industrial operations. Additionally, AWS IoT SiteWise is designed to be secure and compliant, with features like role-based access controls, data encryption, and integration with other AWS services for additional security and compliance features.

Let's get started...

```
"""
)
press_enter_to_continue()
print_dashes()
print(f"")
print(
    f"Use AWS CloudFormation to create an IAM role that is required for
this scenario."
)
template_file = IoTSitewiseGettingStarted.get_template_as_string()

self.stack = self.deploy_cloudformation_stack(
    "python-iot-sitewise-basics", template_file
)
outputs = self.stack.outputs
iam_role = None

for output in outputs:
    if output.get("OutputKey") == "SitewiseRoleArn":
        iam_role = output.get("OutputValue")
```

```

        if iam_role is None:
            error_string = f"Failed to retrieve iam_role from CloudFormation
stack."
            logger.error(error_string)
            raise ValueError(error_string)

        print(f"The ARN of the IAM role is {iam_role}")
        print_dashes()
        print_dashes()
        print(f"1. Create an AWS SiteWise Asset Model")
        print(
            """

```

An AWS IoT SiteWise Asset Model is a way to represent the physical assets, such as equipment, processes, and systems, that exist in an industrial environment. This model provides a structured and hierarchical representation of these assets, allowing users to define the relationships and values of each asset.

This scenario creates two asset model values: temperature and humidity.

```

        """
    )
    press_enter_to_continue()
    asset_model_name = "MyAssetModel1"
    temperature_property_name = "temperature"
    humidity_property_name = "humidity"
    try:
        properties = [
            {
                "name": temperature_property_name,
                "dataType": "DOUBLE",
                "type": {
                    "measurement": {}},
            },
            {
                "name": humidity_property_name,
                "dataType": "DOUBLE",
                "type": {
                    "measurement": {}},
            },
        ]
    
```



```

        ]
        self.asset_model_id = self.iot_sitewise_wrapper.create_asset_model(
            asset_model_name, properties
        )
        print(
            f"Asset Model successfully created. Asset Model ID:
{self.asset_model_id}. "
        )
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
            self.asset_model_id =
self.get_model_id_for_model_name(asset_model_name)
            print(
                f"Asset Model {asset_model_name} already exists. Asset Model
ID: {self.asset_model_id}. "
            )
        else:
            raise

    press_enter_to_continue()
    print_dashes()
    print(f"2. Create an AWS IoT SiteWise Asset")
    print(
        """

```

The IoT SiteWise model that we just created defines the structure and metadata for your physical assets.

Now we create an asset from the asset model.

```

        """
    )
    press_enter_to_continue()

    self.asset_id = self.iot_sitewise_wrapper.create_asset(
        "MyAsset1", self.asset_model_id
    )

    print(f"Asset created with ID: {self.asset_id}")
    press_enter_to_continue()
    print_dashes()
    print_dashes()
    print(f"3. Retrieve the property ID values")
    print(
        """

```

To send data to an asset, we need to get the property ID values. In this scenario, we access the temperature and humidity property ID values.

```

"""
)
press_enter_to_continue()
property_ids = self.iot_sitewise_wrapper.list_asset_model_properties(
    self.asset_model_id
)
humidity_property_id = None
temperature_property_id = None
for property_id in property_ids:
    if property_id.get("name") == humidity_property_name:
        humidity_property_id = property_id.get("id")
    elif property_id.get("name") == temperature_property_name:
        temperature_property_id = property_id.get("id")
if humidity_property_id is None or temperature_property_id is None:
    error_string = f"Failed to retrieve property IDs from Asset Model."
    logger.error(error_string)
    raise ValueError(error_string)

print(f"The Humidity property Id is {humidity_property_id}")
print(f"The Temperature property Id is {temperature_property_id}")
press_enter_to_continue()
print_dashes()
print_dashes()

print(f"4. Send data to an AWS IoT SiteWise Asset")
print(
    """

```

By sending data to an IoT SiteWise Asset, you can aggregate data from multiple sources, normalize the data into a standard format, and store it in a centralized location. This makes it easier to analyze and gain insights from the data.

In this example, we generate sample temperature and humidity data and send it to the AWS IoT SiteWise asset.

```

"""
)
press_enter_to_continue()

values = [
    {

```

```

        "propertyId": humidity_property_id,
        "valueType": "doubleValue",
        "value": 65.0,
    },
    {
        "propertyId": temperature_property_id,
        "valueType": "doubleValue",
        "value": 23.5,
    },
]
self.iot_sitewise_wrapper.batch_put_asset_property_value(self.asset_id,
values)
print(f"Data sent successfully.")

press_enter_to_continue()
print_dashes()
print_dashes()

print(f"5. Retrieve the value of the IoT SiteWise Asset property")
print(
    """
IoT SiteWise is an AWS service that allows you to collect, process, and analyze
industrial data
from connected equipment and sensors. One of the key benefits of reading an IoT
SiteWise property
is the ability to gain valuable insights from your industrial data.

    """
)
press_enter_to_continue()

property_value = self.iot_sitewise_wrapper.get_asset_property_value(
    self.asset_id, temperature_property_id
)
print(f"The property name is '{temperature_property_name}'.")

print(
    f"The value of this property is: {property_value['value']}
['doubleValue']}"
)
press_enter_to_continue()

property_value = self.iot_sitewise_wrapper.get_asset_property_value(
    self.asset_id, humidity_property_id

```

```

    )
    print(f"The property name is '{humidity_property_name}'.")
    print(
        f"The value of this property is: {property_value['value']
['doubleValue']}"
    )
    press_enter_to_continue()
    print_dashes()
    print_dashes()

    print(f"6. Create an IoT SiteWise Portal")
    print(
        """
An IoT SiteWise Portal allows you to aggregate data from multiple industrial
sources,
such as sensors, equipment, and control systems, into a centralized platform.
        """
    )

    press_enter_to_continue()
    contact_email = q.ask("Enter a contact email for the portal:",
q.non_empty)
    print("Creating the portal. The portal may take a while to become
active.")
    self.portal_id = self.iot_sitewise_wrapper.create_portal(
        "MyPortal1", iam_role, contact_email
    )
    print(f"Portal created successfully. Portal ID {self.portal_id}")
    press_enter_to_continue()
    print_dashes()
    print_dashes()

    print(f"7. Describe the Portal")
    print(
        """
In this step, we get a description of the portal and display the portal URL.
        """
    )
    press_enter_to_continue()
    portal_description =
self.iot_sitewise_wrapper.describe_portal(self.portal_id)
    print(f"Portal URL: {portal_description['portalStartUrl']}")
    press_enter_to_continue()
    print_dashes()

```

```

        print_dashes()

        print(f"8. Create an IoT SiteWise Gateway")
        press_enter_to_continue()
        self.gateway_id = self.iot_sitewise_wrapper.create_gateway(
            "MyGateway1", "MyThing1"
        )
        print(f"Gateway creation completed successfully. id is
{self.gateway_id}")
        print_dashes()
        print_dashes()
        print(f"9. Describe the IoT SiteWise Gateway")
        press_enter_to_continue()

        gateway_description = self.iot_sitewise_wrapper.describe_gateway(
            self.gateway_id
        )
        print(f"Gateway Name: {gateway_description['gatewayName']}")
        print(f"Gateway ARN: {gateway_description['gatewayArn']}")
        print(f"Gateway Platform:\n{gateway_description['gatewayPlatform']}")
        print(f"Gateway Creation Date: {gateway_description['gatewayArn']}")
        print_dashes()
        print_dashes()

        print(f"10. Delete the AWS IoT SiteWise Assets")
        if q.ask("Would you like to delete the IoT SiteWise Assets? (y/n)",
q.is_yesno):
            self.cleanup()
        else:
            print(f"The resources will not be deleted.")
        print_dashes()
        print_dashes()
        print(f"This concludes the AWS IoT SiteWise Scenario")

def cleanup(self) -> None:
    """
    Deletes the CloudFormation stack and the resources created for the demo.
    """

    if self.gateway_id is not None:
        self.iot_sitewise_wrapper.delete_gateway(self.gateway_id)
        print(f"Deleted gateway with id {self.gateway_id}.")
        self.gateway_id = None
    if self.portal_id is not None:

```

```

        self.iot_sitewise_wrapper.delete_portal(self.portal_id)
        print(f"Deleted portal with id {self.portal_id}.")
        self.portal_id = None
    if self.asset_id is not None:
        self.iot_sitewise_wrapper.delete_asset(self.asset_id)
        print(f"Deleted asset with id {self.asset_id}.")
        self.iot_sitewise_wrapper.wait_asset_deleted(self.asset_id)
        self.asset_id = None
    if self.asset_model_id is not None:
        self.iot_sitewise_wrapper.delete_asset_model(self.asset_model_id)
        print(f"Deleted asset model with id {self.asset_model_id}.")
        self.asset_model_id = None
    if self.stack is not None:
        stack = self.stack
        self.stack = None
        self.destroy_cloudformation_stack(stack)

def deploy_cloudformation_stack(
    self, stack_name: str, cfn_template: str
) -> ServiceResource:
    """
    Deploys prerequisite resources used by the scenario. The resources are
    defined in the associated `SitewiseRoles-template.yaml` AWS
CloudFormation script and are deployed
    as a CloudFormation stack, so they can be easily managed and destroyed.

    :param stack_name: The name of the CloudFormation stack.
    :param cfn_template: The CloudFormation template as a string.
    :return: The CloudFormation stack resource.
    """
    print(f"Deploying CloudFormation stack: {stack_name}.")
    stack = self.cloud_formation_resource.create_stack(
        StackName=stack_name,
        TemplateBody=cfn_template,
        Capabilities=["CAPABILITY_NAMED_IAM"],
    )
    print(f"CloudFormation stack creation started: {stack_name}")
    print("Waiting for CloudFormation stack creation to complete...")
    waiter = self.cloud_formation_resource.meta.client.get_waiter(
        "stack_create_complete"
    )
    waiter.wait(StackName=stack.name)
    stack.load()
    print("CloudFormation stack creation complete.")

```

```

        return stack

    def destroy_cloudformation_stack(self, stack: ServiceResource) -> None:
        """
        Destroys the resources managed by the CloudFormation stack, and the
        CloudFormation
        stack itself.

        :param stack: The CloudFormation stack that manages the example
        resources.
        """
        print(
            f"CloudFormation stack '{stack.name}' is being deleted. This may take
a few minutes."
        )
        stack.delete()
        waiter = self.cloud_formation_resource.meta.client.get_waiter(
            "stack_delete_complete"
        )
        waiter.wait(StackName=stack.name)
        print(f"CloudFormation stack '{stack.name}' has been deleted.")

    @staticmethod
    def get_template_as_string() -> str:
        """
        Returns a string containing this scenario's CloudFormation template.
        """
        template_file_path = os.path.join(script_dir, "SitewiseRoles-
template.yaml")
        file = open(template_file_path, "r")
        return file.read()

    def get_model_id_for_model_name(self, model_name: str) -> str:
        """
        Returns the model ID for the given model name.

        :param model_name: The name of the model.
        :return: The model ID.
        """
        model_id = None
        asset_models = self.iot_sitewise_wrapper.list_asset_models()
        for asset_model in asset_models:
            if asset_model["name"] == model_name:

```

```

        model_id = asset_model["id"]
        break
    return model_id

```

IoTSitewise Wrapper-Klasse, die Aktionen umschließt AWS IoT SiteWise .

```

class IoTSitewiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def create_asset_model(
        self, asset_model_name: str, properties: List[Dict[str, Any]]
    ) -> str:
        """
        Creates an AWS IoT SiteWise Asset Model.

```



```

:param asset_model_name: The name of the asset model to create.
:param properties: The property definitions of the asset model.
:return: The ID of the created asset model.
"""
try:
    response = self.iotsitewise_client.create_asset_model(
        assetModelName=asset_model_name,
        assetModelDescription="This is a sample asset model
description.",
        assetModelProperties=properties,
    )
    asset_model_id = response["assetModelId"]
    waiter = self.iotsitewise_client.get_waiter("asset_model_active")
    waiter.wait(assetModelId=asset_model_id)
    return asset_model_id
except ClientError as err:
    if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
        logger.error("Asset model %s already exists.", asset_model_name)
    else:
        logger.error(
            "Error creating asset model %s. Here's why %s",
            asset_model_name,
            err.response["Error"]["Message"],
        )
    raise

def create_asset(self, asset_name: str, asset_model_id: str) -> str:
    """
    Creates an AWS IoT SiteWise Asset.

    :param asset_name: The name of the asset to create.
    :param asset_model_id: The ID of the asset model to associate with the
asset.
    :return: The ID of the created asset.
    """
    try:
        response = self.iotsitewise_client.create_asset(
            assetName=asset_name, assetModelId=asset_model_id
        )
        asset_id = response["assetId"]
        waiter = self.iotsitewise_client.get_waiter("asset_active")
        waiter.wait(assetId=asset_id)
        return asset_id

```

```

    except ClientError as err:
        if err.response["Error"] == "ResourceNotFoundException":
            logger.error("Asset model %s does not exist.", asset_model_id)
        else:
            logger.error(
                "Error creating asset %s. Here's why %s",
                asset_name,
                err.response["Error"]["Message"],
            )
            raise

def list_asset_models(self) -> List[Dict[str, Any]]:
    """
    Lists all AWS IoT SiteWise Asset Models.

    :return: A list of dictionaries containing information about each asset
    model.

    """
    try:
        asset_models = []
        paginator =
self.iotsitewise_client.get_paginator("list_asset_models")
        pages = paginator.paginate()
        for page in pages:
            asset_models.extend(page["assetModelSummaries"])
        return asset_models
    except ClientError as err:
        logger.error(
            "Error listing asset models. Here's why %s",
            err.response["Error"]["Message"],
        )
        raise

def list_asset_model_properties(self, asset_model_id: str) -> List[Dict[str,
Any]]:
    """
    Lists all AWS IoT SiteWise Asset Model Properties.

    :param asset_model_id: The ID of the asset model to list values for.
    :return: A list of dictionaries containing information about each asset
    model property.

```

```

"""
try:
    asset_model_properties = []
    paginator = self.iotsitewise_client.get_paginator(
        "list_asset_model_properties"
    )
    pages = paginator.paginate(assetModelId=asset_model_id)
    for page in pages:
        asset_model_properties.extend(page["assetModelPropertySummaries"])
    return asset_model_properties
except ClientError as err:
    logger.error(
        "Error listing asset model values. Here's why %s",
        err.response["Error"]["Message"],
    )
    raise

def batch_put_asset_property_value(
    self, asset_id: str, values: List[Dict[str, str]]
) -> None:
    """
    Sends data to an AWS IoT SiteWise Asset.

    :param asset_id: The asset ID.
    :param values: A list of dictionaries containing the values in the form
        {propertyId : property_id,
         valueType : [stringValue|integerValue|doubleValue|
booleanValue],
         value : the_value}.
    """
    try:
        entries = self.properties_to_values(asset_id, values)

self.iotsitewise_client.batch_put_asset_property_value(entries=entries)
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Asset %s does not exist.", asset_id)
        else:
            logger.error(
                "Error sending data to asset. Here's why %s",
                err.response["Error"]["Message"],
            )

```

```

        raise

    def properties_to_values(
        self, asset_id: str, values: list[dict[str, Any]]
    ) -> list[dict[str, Any]]:
        """
        Utility function to convert a values list to the entries parameter for
        batch_put_asset_property_value.
        :param asset_id : The asset ID.
        :param values : A list of dictionaries containing the values in the form
                        {propertyId : property_id,
                         valueType : [stringValue|integerValue|doubleValue|
booleanValue],
                                value : the_value}.
        :return: An entries list to pass as the 'entries' parameter to
        batch_put_asset_property_value.
        """
        entries = []
        for value in values:
            epoch_ns = time.time_ns()
            self.entry_id += 1
            if value["valueType"] == "stringValue":
                property_value = {"stringValue": value["value"]}
            elif value["valueType"] == "integerValue":
                property_value = {"integerValue": value["value"]}
            elif value["valueType"] == "booleanValue":
                property_value = {"booleanValue": value["value"]}
            elif value["valueType"] == "doubleValue":
                property_value = {"doubleValue": value["value"]}
            else:
                raise ValueError("Invalid valueType: %s", value["valueType"])
            entry = {
                "entryId": f"{self.entry_id}",
                "assetId": asset_id,
                "propertyId": value["propertyId"],
                "propertyValues": [
                    {
                        "value": property_value,
                        "timestamp": {
                            "timeInSeconds": int(epoch_ns / 1000000000),
                            "offsetInNanos": epoch_ns % 1000000000,
                        },
                    },
                ],
            }

```

```

        ],
    }
    entries.append(entry)
    return entries

def get_asset_property_value(
    self, asset_id: str, property_id: str
) -> Dict[str, Any]:
    """
    Gets the value of an AWS IoT SiteWise Asset Property.

    :param asset_id: The ID of the asset.
    :param property_id: The ID of the property.
    :return: A dictionary containing the value of the property.
    """
    try:
        response = self.iotsitewise_client.get_asset_property_value(
            assetId=asset_id, propertyId=property_id
        )
        return response["propertyValue"]
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error(
                "Asset %s or property %s does not exist.", asset_id,
property_id
            )
        else:
            logger.error(
                "Error getting asset property value. Here's why %s",
                err.response["Error"]["Message"],
            )
        raise

def create_portal(
    self, portal_name: str, iam_role_arn: str, portal_contact_email: str
) -> str:
    """
    Creates an AWS IoT SiteWise Portal.

    :param portal_name: The name of the portal to create.
    :param iam_role_arn: The ARN of an IAM role.
    :param portal_contact_email: The contact email of the portal.

```

```

        :return: The ID of the created portal.
        """
        try:
            response = self.iotsitewise_client.create_portal(
                portalName=portal_name,
                roleArn=iam_role_arn,
                portalContactEmail=portal_contact_email,
            )
            portal_id = response["portalId"]
            waiter = self.iotsitewise_client.get_waiter("portal_active")
            waiter.wait(portalId=portal_id, WaiterConfig={"MaxAttempts": 40})
            return portal_id
        except ClientError as err:
            if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
                logger.error("Portal %s already exists.", portal_name)
            else:
                logger.error(
                    "Error creating portal %s. Here's why %s",
                    portal_name,
                    err.response["Error"]["Message"],
                )
            raise

def describe_portal(self, portal_id: str) -> Dict[str, Any]:
    """
    Describes an AWS IoT SiteWise Portal.

    :param portal_id: The ID of the portal to describe.
    :return: A dictionary containing information about the portal.
    """
    try:
        response =
self.iotsitewise_client.describe_portal(portalId=portal_id)
        return response
    except ClientError as err:
        logger.error(
            "Error describing portal %s. Here's why %s",
            portal_id,
            err.response["Error"]["Message"],
        )
        raise

```

```

def create_gateway(self, gateway_name: str, my_thing: str) -> str:
    """
    Creates an AWS IoT SiteWise Gateway.

    :param gateway_name: The name of the gateway to create.
    :param my_thing: The core device thing name.
    :return: The ID of the created gateway.
    """
    try:
        response = self.iotsitewise_client.create_gateway(
            gatewayName=gateway_name,
            gatewayPlatform={
                "greengrassV2": {"coreDeviceThingName": my_thing},
            },
            tags={"Environment": "Production"},
        )
        gateway_id = response["gatewayId"]
        return gateway_id
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
            logger.error("Gateway %s already exists.", gateway_name)
        else:
            logger.error(
                "Error creating gateway %s. Here's why %s",
                gateway_name,
                err.response["Error"]["Message"],
            )
        raise

def describe_gateway(self, gateway_id: str) -> Dict[str, Any]:
    """
    Describes an AWS IoT SiteWise Gateway.

    :param gateway_id: The ID of the gateway to describe.
    :return: A dictionary containing information about the gateway.
    """
    try:
        response =
self.iotsitewise_client.describe_gateway(gatewayId=gateway_id)
        return response
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Gateway %s does not exist.", gateway_id)

```

```
        else:
            logger.error(
                "Error describing gateway %s. Here's why %s",
                gateway_id,
                err.response["Error"]["Message"],
            )
        raise

def delete_gateway(self, gateway_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Gateway.

    :param gateway_id: The ID of the gateway to delete.
    """
    try:
        self.iotsitewise_client.delete_gateway(gatewayId=gateway_id)
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Gateway %s does not exist.", gateway_id)
        else:
            logger.error(
                "Error deleting gateway %s. Here's why %s",
                gateway_id,
                err.response["Error"]["Message"],
            )
        raise

def delete_portal(self, portal_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Portal.

    :param portal_id: The ID of the portal to delete.
    """
    try:
        self.iotsitewise_client.delete_portal(portalId=portal_id)
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Portal %s does not exist.", portal_id)
        else:
            logger.error(
                "Error deleting portal %s. Here's why %s",
                portal_id,
```



```

        err.response["Error"]["Message"],
    )
    raise

def delete_asset(self, asset_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Asset.

    :param asset_id: The ID of the asset to delete.
    """
    try:
        self.iotsitewise_client.delete_asset(assetId=asset_id)
    except ClientError as err:
        logger.error(
            "Error deleting asset %s. Here's why %s",
            asset_id,
            err.response["Error"]["Message"],
        )
        raise

def delete_asset_model(self, asset_model_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Asset Model.

    :param asset_model_id: The ID of the asset model to delete.
    """
    try:
        self.iotsitewise_client.delete_asset_model(assetModelId=asset_model_id)
    except ClientError as err:
        logger.error(
            "Error deleting asset model %s. Here's why %s",
            asset_model_id,
            err.response["Error"]["Message"],
        )
        raise

def wait_asset_deleted(self, asset_id: str) -> None:
    """
    Waits for an AWS IoT SiteWise Asset to be deleted.

```

```
:param asset_id: The ID of the asset to wait for.
"""
try:
    waiter = self.iotsitewise_client.get_waiter("asset_not_exists")
    waiter.wait(assetId=asset_id)
except ClientError as err:
    logger.error(
        "Error waiting for asset %s to be deleted. Here's why %s",
        asset_id,
        err.response["Error"]["Message"],
    )
    raise
```

- Weitere API-Informationen finden Sie in den folgenden Themen der API-Referenz zum AWS SDK für Python (Boto3).
 - [BatchPutAssetPropertyValue](#)
 - [CreateAsset](#)
 - [CreateAssetModel](#)
 - [CreateGateway](#)
 - [DeleteAsset](#)
 - [DeleteAssetModel](#)
 - [DeleteGateway](#)
 - [DescribeAssetModel](#)
 - [DescribeGateway](#)
 - [GetAssetPropertyValue](#)
 - [ListAssetModelProperties](#)
 - [ListAssetModels](#)

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. [Verwenden Sie diesen Service mit einem AWS SDK](#) Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Aktionen zur AWS IoT SiteWise Verwendung AWS SDKs

Die folgenden Codebeispiele zeigen, wie Sie einzelne AWS IoT SiteWise Aktionen mit ausführen können AWS SDKs. Jedes Beispiel enthält einen Link zu GitHub, wo Sie Anweisungen zum Einrichten und Ausführen des Codes finden.

Die folgenden Beispiele enthalten nur die am häufigsten verwendeten Aktionen. Eine vollständige Liste finden Sie in der [AWS IoT SiteWise -API-Referenz](#).

Beispiele

- [Verwendung BatchPutAssetPropertyValue mit einem AWS SDK oder CLI](#)
- [Verwendung CreateAsset mit einem AWS SDK oder CLI](#)
- [Verwendung CreateAssetModel mit einem AWS SDK oder CLI](#)
- [Verwendung CreateGateway mit einem AWS SDK oder CLI](#)
- [Verwendung DeleteAsset mit einem AWS SDK oder CLI](#)
- [Verwendung DeleteAssetModel mit einem AWS SDK oder CLI](#)
- [Verwendung DeleteGateway mit einem AWS SDK oder CLI](#)
- [Verwendung DescribeAssetModel mit einem AWS SDK oder CLI](#)
- [Verwendung DescribeGateway mit einem AWS SDK oder CLI](#)
- [Verwendung GetAssetPropertyValue mit einem AWS SDK oder CLI](#)
- [ListAssetModelProperties mit einem AWS SDK verwenden](#)
- [Verwendung ListAssetModels mit einem AWS SDK oder CLI](#)

Verwendung **BatchPutAssetPropertyValue** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie BatchPutAssetPropertyValue verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Kennenlernen der Grundlagen](#)

CLI

AWS CLI

So senden Sie Daten an Komponenteneigenschaften

Im folgenden Beispiel für `batch-put-asset-property-value` werden Leistungs- und Temperaturdaten an die Eigenschaften der Komponente gesendet, die durch Eigenschaftsalias identifiziert sind.

```
aws iotsitewise batch-put-asset-property-value \  
  --cli-input-json file://batch-put-asset-property-value.json
```

Inhalt von `batch-put-asset-property-value.json`:

```
{  
  "entries": [  
    {  
      "entryId": "1575691200-company-windfarm-3-turbine-7-power",  
      "propertyAlias": "company-windfarm-3-turbine-7-power",  
      "propertyValues": [  
        {  
          "value": {  
            "doubleValue": 4.92  
          },  
          "timestamp": {  
            "timeInSeconds": 1575691200  
          },  
          "quality": "GOOD"  
        }  
      ]  
    },  
    {  
      "entryId": "1575691200-company-windfarm-3-turbine-7-temperature",  
      "propertyAlias": "company-windfarm-3-turbine-7-temperature",  
      "propertyValues": [  
        {  
          "value": {  
            "integerValue": 38  
          },  
          "timestamp": {  
            "timeInSeconds": 1575691200  
          }  
        }  
      ]  
    }  
  ]  
}
```

```

    }
  ]
}

```

Ausgabe:

```

{
  "errorEntries": []
}

```

Weitere Informationen finden Sie unter [Daten mithilfe der AWS SiteWise IoT-API aufnehmen](#) im AWS SiteWise IoT-Benutzerhandbuch.

- Einzelheiten zur API finden Sie [BatchPutAssetPropertyValue](#) in der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

/**
 * Sends data to the SiteWise service.
 *
 * @param assetId      the ID of the asset to which the data will be sent.
 * @param tempPropertyId the ID of the temperature property.
 * @param humidityPropId the ID of the humidity property.
 * @return a {@link CompletableFuture} that represents a {@link
BatchPutAssetPropertyValueResponse} result. The
 *         calling code can attach callbacks, then handle the result or
exception by calling
 *         {@link CompletableFuture#join()} or {@link
CompletableFuture#get()}.
 *
 * <p>

```

```

    *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
    *      available to the calling code as a {@link CompletionException}. By
calling
    *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<BatchPutAssetPropertyValueResponse>
sendDataToSiteWiseAsync(String assetId, String tempPropertyId, String
humidityPropId) {
    Map<String, Double> sampleData = generateSampleData();
    long timestamp = Instant.now().toEpochMilli();

    TimeInNanos time = TimeInNanos.builder()
        .timeInSeconds(timestamp / 1000)
        .offsetInNanos((int) ((timestamp % 1000) * 1000000))
        .build();

    BatchPutAssetPropertyValueRequest request =
BatchPutAssetPropertyValueRequest.builder()
        .entries(Arrays.asList(
            PutAssetPropertyValueEntry.builder()
                .entryId("entry-3")
                .assetId(assetId)
                .propertyId(tempPropertyId)
                .propertyValues(Arrays.asList(
                    AssetPropertyValue.builder()
                        .value(Variant.builder()
                            .doubleValue(sampleData.get("Temperature"))
                            .build())
                        .timestamp(time)
                        .build()
                ))
                .build(),
            PutAssetPropertyValueEntry.builder()
                .entryId("entry-4")
                .assetId(assetId)
                .propertyId(humidityPropId)
                .propertyValues(Arrays.asList(
                    AssetPropertyValue.builder()
                        .value(Variant.builder()
                            .doubleValue(sampleData.get("Humidity"))
                            .build()
                        )
                        .timestamp(time)
                ))
                .build()
        ))
        .build()
    }

```

```

        .build()
    ))
    .build()
))
.build();

return getAsyncClient().batchPutAssetPropertyValue(request)
    .whenComplete((response, exception) -> {
        if (exception != null) {
            logger.error("An exception occurred: {}",
exception.getCause().getMessage());
        }
    });
}

```

- Einzelheiten zur API finden Sie [BatchPutAssetPropertyValue](#) in der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

import {
    BatchPutAssetPropertyValueCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Batch put asset property values.
 * @param {{ entries : array }}
 */
export const main = async ({ entries }) => {
    const client = new IoTSiteWiseClient({});

```

```

try {
  const result = await client.send(
    new BatchPutAssetPropertyValueCommand({
      entries: entries,
    }),
  );
  console.log("Asset properties batch put successfully.");
  return result;
} catch (caught) {
  if (caught instanceof Error && caught.name === "ResourceNotFound") {
    console.warn(`${caught.message}. A resource could not be found.`);
  } else {
    throw caught;
  }
}
};

```

- Einzelheiten zur API finden Sie [BatchPutAssetPropertyValue](#) in der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.

```



```

        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def batch_put_asset_property_value(
        self, asset_id: str, values: List[Dict[str, str]]
    ) -> None:
        """
        Sends data to an AWS IoT SiteWise Asset.

        :param asset_id: The asset ID.
        :param values: A list of dictionaries containing the values in the form
                        {propertyId : property_id,
                        valueType : [stringValue|integerValue|doubleValue|
booleanValue],
                        value : the_value}.
        """
        try:
            entries = self.properties_to_values(asset_id, values)

        self.iotsitewise_client.batch_put_asset_property_value(entries=entries)
        except ClientError as err:
            if err.response["Error"]["Code"] == "ResourceNotFoundException":
                logger.error("Asset %s does not exist.", asset_id)
            else:
                logger.error(
                    "Error sending data to asset. Here's why %s",
                    err.response["Error"]["Message"],
                )
            raise

```

Eine Hilfsfunktion zum Generieren des Parameters „entries“ aus einer Werteliste.

```
def properties_to_values(
    self, asset_id: str, values: list[dict[str, Any]]
) -> list[dict[str, Any]]:
    """
    Utility function to convert a values list to the entries parameter for
    batch_put_asset_property_value.
    :param asset_id : The asset ID.
    :param values : A list of dictionaries containing the values in the form
                    {propertyId : property_id,
                     valueType : [stringValue|integerValue|doubleValue|
booleanValue],
                               value : the_value}.
    :return: An entries list to pass as the 'entries' parameter to
    batch_put_asset_property_value.
    """
    entries = []
    for value in values:
        epoch_ns = time.time_ns()
        self.entry_id += 1
        if value["valueType"] == "stringValue":
            property_value = {"stringValue": value["value"]}
        elif value["valueType"] == "integerValue":
            property_value = {"integerValue": value["value"]}
        elif value["valueType"] == "booleanValue":
            property_value = {"booleanValue": value["value"]}
        elif value["valueType"] == "doubleValue":
            property_value = {"doubleValue": value["value"]}
        else:
            raise ValueError("Invalid valueType: %s", value["valueType"])
        entry = {
            "entryId": f"{self.entry_id}",
            "assetId": asset_id,
            "propertyId": value["propertyId"],
            "propertyValues": [
                {
                    "value": property_value,
                    "timestamp": {
                        "timeInSeconds": int(epoch_ns / 1000000000),
```

```

        "offsetInNanos": epoch_ns % 1000000000,
    },
    }
],
}
entries.append(entry)
return entries

```

Hier ist ein Beispiel für eine Werteliste, die an die Hilfsfunktion übergeben werden kann.

```

values = [
    {
        "propertyId": humidity_property_id,
        "valueType": "doubleValue",
        "value": 65.0,
    },
    {
        "propertyId": temperature_property_id,
        "valueType": "doubleValue",
        "value": 23.5,
    },
]

```

- Einzelheiten zur API finden Sie [BatchPutAssetPropertyValue](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

TRY.
    lo_ios->batchputassetpropertyvalue(

```

```

        it_entries = it_entries
    ).
    MESSAGE 'Data sent to IoT SiteWise asset successfully.' TYPE 'I'.
CATCH /aws1/cx_iosourcenotfoundex.
    MESSAGE 'Asset does not exist.' TYPE 'E'.
ENDTRY.

```

- Einzelheiten zur API finden Sie [BatchPutAssetPropertyValue](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Service mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **CreateAsset** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie CreateAsset verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Kennenlernen der Grundlagen](#)

CLI

AWS CLI

So erstellen Sie eine Komponente

Im folgenden Beispiel für create-asset wird aus dem Komponentenmodell einer Windturbine eine Windturbinenkomponente erstellt.

```

aws iotsitewise create-asset \
  --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
  --asset-name "Wind Turbine 1"

```

Ausgabe:

```
{
```

```

    "assetId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
    "assetArn": "arn:aws:iot:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
    "assetStatus": {
        "state": "CREATING"
    }
}

```

Weitere Informationen finden Sie im AWS SiteWise IoT-Benutzerhandbuch unter [Assets erstellen](#).

- Einzelheiten zur API finden Sie [CreateAsset](#) in der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

/**
 * Creates an asset with the specified name and asset model Id.
 *
 * @param assetName    the name of the asset to create.
 * @param assetModelId the Id of the asset model to associate with the asset.
 * @return a {@link CompletableFuture} that represents a {@link
CreateAssetResponse} result. The calling code can
 *         attach callbacks, then handle the result or exception by calling
{@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
 *         available to the calling code as a {@link CompletionException}. By
calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */

```

```

public CompletableFuture<CreateAssetResponse> createAssetAsync(String
assetName, String assetModelId) {
    CreateAssetRequest createAssetRequest = CreateAssetRequest.builder()
        .assetModelId(assetModelId)
        .assetDescription("Created using the AWS SDK for Java")
        .assetName(assetName)
        .build();

    return getAsyncClient().createAsset(createAssetRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to create asset: {}",
exception.getCause().getMessage());
            }
        });
}

```

- Einzelheiten zur API finden Sie [CreateAsset](#) in der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

import {
    CreateAssetCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create an Asset.
 * @param {{ assetName : string, assetModelId: string }}
 */
export const main = async ({ assetName, assetModelId }) => {
    const client = new IoTSiteWiseClient({});

```

```

try {
  const result = await client.send(
    new CreateAssetCommand({
      assetName: assetName, // The name to give the Asset.
      assetModelId: assetModelId, // The ID of the asset model from which to
create the asset.
    }),
  );
  console.log("Asset created successfully.");
  return result;
} catch (caught) {
  if (caught instanceof Error && caught.name === "ResourceNotFound") {
    console.warn(
      `${caught.message}. The asset model could not be found. Please check the
asset model id.`,
    );
  } else {
    throw caught;
  }
}
};

```

- Einzelheiten zur API finden Sie [CreateAsset](#) in der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

```

```

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.

        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def create_asset(self, asset_name: str, asset_model_id: str) -> str:
        """
        Creates an AWS IoT SiteWise Asset.

        :param asset_name: The name of the asset to create.
        :param asset_model_id: The ID of the asset model to associate with the
        asset.

        :return: The ID of the created asset.
        """
        try:
            response = self.iotsitewise_client.create_asset(
                assetName=asset_name, assetModelId=asset_model_id
            )
            asset_id = response["assetId"]
            waiter = self.iotsitewise_client.get_waiter("asset_active")
            waiter.wait(assetId=asset_id)
            return asset_id
        except ClientError as err:
            if err.response["Error"] == "ResourceNotFoundException":
                logger.error("Asset model %s does not exist.", asset_model_id)
            else:
                logger.error(
                    "Error creating asset %s. Here's why %s",

```



```

        asset_name,
        err.response["Error"]["Message"],
    )
    raise

```

- Einzelheiten zur API finden Sie [CreateAsset](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

TRY.
    oo_result = lo_ios->createasset(
        iv_assetname = iv_asset_name
        iv_assetmodelid = iv_asset_model_id
    ). " oo_result is returned for testing purposes. "
    MESSAGE 'IoT SiteWise asset created' TYPE 'I'.
CATCH /aws1/cx_iosresourcenotfoundex.
    MESSAGE 'Asset model does not exist.' TYPE 'E'.
ENDTRY.

```

- Einzelheiten zur API finden Sie [CreateAsset](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Service mit einem AWSSDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **CreateAssetModel** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `CreateAssetModel` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Kennenlernen der Grundlagen](#)

CLI

AWS CLI

So erstellen Sie ein Komponentenmodell

Im folgenden Beispiel für `create-asset-model` wird ein Komponentenmodell erstellt, das eine Windturbine mit den folgenden Eigenschaften definiert:

Seriennummer — Die Seriennummer einer Windkraftanlage — Der erzeugte Strom — Der erzeugte Strom aus einer Windturbine
Temperatur C — Der Temperaturdatenstrom einer Windkraftanlage in Celsius
Temperature F — Die abgebildeten Temperaturdatenpunkte von Celsius bis Fahrenheit

```
aws iotsitewise create-asset-model \
  --cli-input-json file://create-wind-turbine-model.json
```

Inhalt von `create-wind-turbine-model.json`:

```
{
  "assetModelName": "Wind Turbine Model",
  "assetModelDescription": "Represents a wind turbine",
  "assetModelProperties": [
    {
      "name": "Serial Number",
      "dataType": "STRING",
      "type": {
        "attribute": {}
      }
    },
    {
      "name": "Generated Power",
      "dataType": "DOUBLE",
      "unit": "kW",
      "type": {
        "measurement": {}
      }
    }
  ]
}
```

```

    },
    {
      "name": "Temperature C",
      "dataType": "DOUBLE",
      "unit": "Celsius",
      "type": {
        "measurement": {}
      }
    },
    {
      "name": "Temperature F",
      "dataType": "DOUBLE",
      "unit": "Fahrenheit",
      "type": {
        "transform": {
          "expression": "temp_c * 9 / 5 + 32",
          "variables": [
            {
              "name": "temp_c",
              "value": {
                "propertyId": "Temperature C"
              }
            }
          ]
        }
      }
    },
    {
      "name": "Total Generated Power",
      "dataType": "DOUBLE",
      "unit": "kW",
      "type": {
        "metric": {
          "expression": "sum(power)",
          "variables": [
            {
              "name": "power",
              "value": {
                "propertyId": "Generated Power"
              }
            }
          ],
          "window": {
            "tumbling": {

```

```

    "interval": "1h"
  }
}
}
}
}
]
}

```

Ausgabe:

```

{
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "assetModelStatus": {
    "state": "CREATING"
  }
}

```

Weitere Informationen finden Sie unter [Definieren von Asset-Modellen](#) im AWS SiteWise IoT-Benutzerhandbuch.

- Einzelheiten zur API finden Sie [CreateAssetModel](#) in der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

/**
 * Creates an asset model.
 *
 * @param name the name of the asset model to create.

```

```

    * @return a {@link CompletableFuture} that represents a {@link
CreateAssetModelResponse} result. The calling code
    *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
    *         {@link CompletableFuture#get()}.
    *         <p>
    *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
    *         available to the calling code as a {@link CompletionException}. By
calling
    *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<CreateAssetModelResponse>
createAssetModelAsync(String name) {
        PropertyType humidity = PropertyType.builder()
            .measurement(Measurement.builder().build())
            .build();

        PropertyType temperaturePropertyType = PropertyType.builder()
            .measurement(Measurement.builder().build())
            .build();

        AssetModelPropertyDefinition temperatureProperty =
AssetModelPropertyDefinition.builder()
            .name("Temperature")
            .dataType(PropertyDataType.DOUBLE)
            .type(temperaturePropertyType)
            .build();

        AssetModelPropertyDefinition humidityProperty =
AssetModelPropertyDefinition.builder()
            .name("Humidity")
            .dataType(PropertyDataType.DOUBLE)
            .type(humidity)
            .build();

        CreateAssetModelRequest createAssetModelRequest =
CreateAssetModelRequest.builder()
            .assetModelName(name)
            .assetModelDescription("This is my asset model")
            .assetModelProperties(temperatureProperty, humidityProperty)
            .build();

```

```

        return getAsyncClient().createAssetModel(createAssetModelRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("Failed to create asset model: {} ",
exception.getCause().getMessage());
                }
            });
    }
}

```

- Einzelheiten zur API finden Sie [CreateAssetModel](#) in der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

import {
    CreateAssetModelCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create an Asset Model.
 * @param {{ assetName : string, assetModelId: string }}
 */
export const main = async ({ assetModelName, assetModelId }) => {
    const client = new IoTSiteWiseClient({});
    try {
        const result = await client.send(
            new CreateAssetModelCommand({
                assetModelName: assetModelName, // The name to give the Asset Model.
            }),
        );
    }
};

```

```

    console.log("Asset model created successfully.");
    return result;
} catch (caught) {
    if (caught instanceof Error && caught.name === "IoTSiteWiseError") {
        console.warn(
            `${caught.message}. There was a problem creating the asset model.`
        );
    } else {
        throw caught;
    }
}
};

```

- Einzelheiten zur API finden Sie [CreateAssetModel](#) in der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

class IoTSitewiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

```

```

@classmethod
def from_client(cls) -> "IoTSitewiseWrapper":
    """
    Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
    client.

    :return: An instance of IoTSitewiseWrapper initialized with the default
    AWS IoT SiteWise client.
    """
    iotsitewise_client = boto3.client("iotsitewise")
    return cls(iotsitewise_client)

def create_asset_model(
    self, asset_model_name: str, properties: List[Dict[str, Any]]
) -> str:
    """
    Creates an AWS IoT SiteWise Asset Model.

    :param asset_model_name: The name of the asset model to create.
    :param properties: The property definitions of the asset model.
    :return: The ID of the created asset model.
    """
    try:
        response = self.iotsitewise_client.create_asset_model(
            assetModelName=asset_model_name,
            assetModelDescription="This is a sample asset model
description.",
            assetModelProperties=properties,
        )
        asset_model_id = response["assetModelId"]
        waiter = self.iotsitewise_client.get_waiter("asset_model_active")
        waiter.wait(assetModelId=asset_model_id)
        return asset_model_id
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
            logger.error("Asset model %s already exists.", asset_model_name)
        else:
            logger.error(
                "Error creating asset model %s. Here's why %s",
                asset_model_name,
                err.response["Error"]["Message"],
            )

```



```
raise
```

Hier finden Sie ein Beispiel für eine Eigenschaftenliste, die an die Funktion übergeben wird.

```
properties = [  
    {  
        "name": temperature_property_name,  
        "dataType": "DOUBLE",  
        "type": {  
            "measurement": {},  
        },  
    },  
    {  
        "name": humidity_property_name,  
        "dataType": "DOUBLE",  
        "type": {  
            "measurement": {},  
        },  
    },  
]
```

- Einzelheiten zur API finden Sie [CreateAssetModel](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

TRY.

```
oo_result = lo_ios->createassetmodel(  
    iv_assetmodelname = iv_asset_model_name  
    iv_assetmodeldescription = 'This is a sample asset model description.'
```

```

        it_assetmodelproperties = it_properties
    ). " oo_result is returned for testing purposes. "
    MESSAGE 'IoT SiteWise asset model created' TYPE 'I'.
CATCH /aws1/cx_iosresrcalrdyexistsex.
    MESSAGE 'Asset model already exists.' TYPE 'E'.
ENDTRY.

```

- Einzelheiten zur API finden Sie [CreateAssetModel](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Service mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **CreateGateway** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie CreateGateway verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Kennenlernen der Grundlagen](#)

CLI

AWS CLI

Um ein Gateway zu erstellen

Das folgende create-gateway Beispiel erstellt ein Gateway, das auf AWS IoT Greengrass läuft.

```

aws iotsitewise create-gateway \
  --gateway-name ExampleCorpGateway \
  --gateway-platform greengrass={groupArn=arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/a1b2c3d4-5678-90ab-cdef-1b1b1EXAMPLE}

```

Ausgabe:

```
{
```

```

    "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
    "gatewayArn": "arn:aws:iot:us-west-2:123456789012:gateway/a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
  }

```

Weitere Informationen finden Sie unter [Konfiguration eines Gateways](#) im AWS SiteWise IoT-Benutzerhandbuch.

- Einzelheiten zur API finden Sie [CreateGateway](#) in der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

/**
 * Creates a new IoT Sitewise gateway.
 *
 * @param gatewayName The name of the gateway to create.
 * @param myThing      The name of the core device thing to associate with the
gateway.
 * @return a {@link CompletableFuture} that represents a {@link String}
result of the gateways ID. The calling code
 *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
By calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<String> createGatewayAsync(String gatewayName,
String myThing) {

```

```

GreengrassV2 gg = GreengrassV2.builder()
    .coreDeviceThingName(myThing)
    .build();

GatewayPlatform platform = GatewayPlatform.builder()
    .greengrassV2(gg)
    .build();

Map<String, String> tag = new HashMap<>();
tag.put("Environment", "Production");

CreateGatewayRequest createGatewayRequest =
CreateGatewayRequest.builder()
    .gatewayName(gatewayName)
    .gatewayPlatform(platform)
    .tags(tag)
    .build();

return getAsyncClient().createGateway(createGatewayRequest)
    .handle((response, exception) -> {
        if (exception != null) {
            logger.error("Error creating the gateway.");
            throw (CompletionException) exception;
        }
        logger.info("The ARN of the gateway is {}" ,
response.gatewayArn());
        return response.gatewayId();
    });
}

```

- Einzelheiten zur API finden Sie [CreateGateway](#) in der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
import {
  CreateGatewayCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create a Gateway.
 * @param {{ }}
 */
export const main = async ({ gatewayName }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new CreateGatewayCommand({
        gatewayName: gatewayName, // The name to give the created Gateway.
      }),
    );
    console.log("Gateway created successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "IoTSiteWiseError") {
      console.warn(
        `${caught.message}. There was a problem creating the Gateway.`,
      );
    } else {
      throw caught;
    }
  }
};
```

- Einzelheiten zur API finden Sie [CreateGateway](#) in der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
class IoTSitewiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def create_gateway(self, gateway_name: str, my_thing: str) -> str:
        """
        Creates an AWS IoT SiteWise Gateway.
```

```
:param gateway_name: The name of the gateway to create.
:param my_thing: The core device thing name.
:return: The ID of the created gateway.
"""
try:
    response = self.iotsitewise_client.create_gateway(
        gatewayName=gateway_name,
        gatewayPlatform={
            "greengrassV2": {"coreDeviceThingName": my_thing},
        },
        tags={"Environment": "Production"},
    )
    gateway_id = response["gatewayId"]
    return gateway_id
except ClientError as err:
    if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
        logger.error("Gateway %s already exists.", gateway_name)
    else:
        logger.error(
            "Error creating gateway %s. Here's why %s",
            gateway_name,
            err.response["Error"]["Message"],
        )
    raise
```

- Einzelheiten zur API finden Sie [CreateGateway](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel](#) einrichten und ausführen.

TRY.

```

oo_result = lo_ios->creategateway(
  iv_gatewayname = iv_gateway_name
  io_gatewayplatform = NEW /aws1/cl_iosgatewayplatform(
    io_greengrassv2 = NEW /aws1/cl_iosgreengrassv2(
      iv_coredevicethingname = iv_core_device_thing_name
    )
  )
  it_tags = VALUE /aws1/cl_iostagmap_w=>tt_tagmap(
    (
      VALUE /aws1/cl_iostagmap_w=>ts_tagmap_maprow(
        key = 'Environment'
        value = NEW /aws1/cl_iostagmap_w( 'Production' )
      )
    )
  )
). " oo_result is returned for testing purposes. "
MESSAGE 'IoT SiteWise gateway created' TYPE 'I'.
CATCH /aws1/cx_iosresrcalrdyexistsex.
MESSAGE 'Gateway already exists.' TYPE 'E'.
ENDTRY.

```

- Einzelheiten zur API finden Sie [CreateGateway](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Service mit einem AWSSDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **DeleteAsset** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `DeleteAsset` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Kennenlernen der Grundlagen](#)

CLI

AWS CLI

So löschen Sie ein Asset

Im folgenden Beispiel für `delete-asset` wird eine Windturbinen-Komponente gelöscht.

```
aws iotsitewise delete-asset \  
  --asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE
```

Ausgabe:

```
{  
  "assetStatus": {  
    "state": "DELETING"  
  }  
}
```

Weitere Informationen finden Sie unter [Löschen von Assets](#) im AWS SiteWise IoT-Benutzerhandbuch.

- Einzelheiten zur API finden Sie [DeleteAsset](#) in der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
/**  
 * Deletes an asset.  
 *  
 * @param assetId the ID of the asset to be deleted.  
 * @return a {@link CompletableFuture} that represents a {@link  
 DeleteAssetResponse} result. The calling code can
```

```

    *      attach callbacks, then handle the result or exception by calling
    {@link CompletableFuture#join()} or
    *      {@link CompletableFuture#get()}.
    *      <p>
    *      If any completion stage in this method throws an exception, the
    method logs the exception cause and keeps
    *      it available to the calling code as a {@link CompletionException}.
    By calling
    *      {@link CompletionException#getCause()}, the calling code can
    access the original exception.
    */
    public CompletableFuture<DeleteAssetResponse> deleteAssetAsync(String
    assetId) {
        DeleteAssetRequest deleteAssetRequest = DeleteAssetRequest.builder()
            .assetId(assetId)
            .build();

        return getAsyncClient().deleteAsset(deleteAssetRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("An error occurred deleting asset with id: {}",
    assetId);
                }
            });
    }
}

```

- Einzelheiten zur API finden Sie [DeleteAsset](#) in der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

import {
    DeleteAssetCommand,
    IoTSiteWiseClient,

```

```

} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Delete an asset.
 * @param {{ assetId : string }}
 */
export const main = async ({ assetId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    await client.send(
      new DeleteAssetCommand({
        assetId: assetId, // The model id to delete.
      }),
    );
    console.log("Asset deleted successfully.");
    return { assetDeleted: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. There was a problem deleting the asset.`
      );
    } else {
      throw caught;
    }
  }
};

```

- Einzelheiten zur API finden Sie [DeleteAsset](#) in der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
class IoTSiteWiseWrapper:
```

```

"""Encapsulates AWS IoT SiteWise actions using the client interface."""

def __init__(self, iotsitewise_client: client) -> None:
    """
    Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

    :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
    provides low-level
                                access to AWS IoT SiteWise services.
    """
    self.iotsitewise_client = iotsitewise_client
    self.entry_id = 0 # Incremented to generate unique entry IDs for
    batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

def delete_asset(self, asset_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Asset.

    :param asset_id: The ID of the asset to delete.
    """
    try:
        self.iotsitewise_client.delete_asset(assetId=asset_id)
    except ClientError as err:
        logger.error(
            "Error deleting asset %s. Here's why %s",
            asset_id,
            err.response["Error"]["Message"],
        )
        raise

```

- Einzelheiten zur API finden Sie [DeleteAsset](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
TRY.  
    lo_ios->deleteasset(  
        iv_assetid = iv_asset_id  
    ).  
    MESSAGE 'IoT SiteWise asset deleted.' TYPE 'I'.  
CATCH /aws1/cx_rt_generic.  
    MESSAGE 'Unable to delete asset.' TYPE 'E'.  
ENDTRY.
```

- Einzelheiten zur API finden Sie [DeleteAsset](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Service mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **DeleteAssetModel** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `DeleteAssetModel` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Kennenlernen der Grundlagen](#)

CLI

AWS CLI

So löschen Sie ein Komponentenmodell

Im folgenden Beispiel für `delete-asset-model` wird ein Windturbinen-Komponentenmodell gelöscht.

```
aws iotsitewise delete-asset-model \
  --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Ausgabe:

```
{
  "assetModelStatus": {
    "state": "DELETING"
  }
}
```

Weitere Informationen finden Sie unter [Löschen von Asset-Modellen](#) im AWS SiteWise IoT-Benutzerhandbuch.

- Einzelheiten zur API finden Sie [DeleteAssetModel](#) in der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
/**
 * Deletes an Asset Model with the specified ID.
 *
 * @param assetModelId the ID of the Asset Model to delete.
 * @return a {@link CompletableFuture} that represents a {@link
 * DeleteAssetModelResponse} result. The calling code
```

```

    *      can attach callbacks, then handle the result or exception by
    calling {@link CompletableFuture#join()} or
    *      {@link CompletableFuture#get()}.
    *      <p>
    *      If any completion stage in this method throws an exception, the
    method logs the exception cause and keeps
    *      it available to the calling code as a {@link CompletionException}.
    By calling
    *      {@link CompletionException#getCause()}, the calling code can
    access the original exception.
    */
    public CompletableFuture<DeleteAssetModelResponse>
    deleteAssetModelAsync(String assetModelId) {
        DeleteAssetModelRequest deleteAssetModelRequest =
        DeleteAssetModelRequest.builder()
            .assetModelId(assetModelId)
            .build();

        return getAsyncClient().deleteAssetModel(deleteAssetModelRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("Failed to delete asset model with ID:{}",
exception.getMessage());
                }
            });
    }
}

```

- Einzelheiten zur API finden Sie [DeleteAssetModel](#) in der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
import {
  DeleteAssetModelCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Delete an asset model.
 * @param {{ assetModelId : string }}
 */
export const main = async ({ assetModelId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    await client.send(
      new DeleteAssetModelCommand({
        assetModelId: assetModelId, // The model id to delete.
      }),
    );
    console.log("Asset model deleted successfully.");
    return { assetModelDeleted: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. There was a problem deleting the asset model.`
      );
    } else {
      throw caught;
    }
  }
};
```

- Einzelheiten zur API finden Sie [DeleteAssetModel](#) in der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSiteWiseWrapper":
        """
        Creates an IoTSiteWiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSiteWiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def delete_asset_model(self, asset_model_id: str) -> None:
        """
        Deletes an AWS IoT SiteWise Asset Model.
```

```

        :param asset_model_id: The ID of the asset model to delete.
        """
        try:

self.iotsitewise_client.delete_asset_model(assetModelId=asset_model_id)
        except ClientError as err:
            logger.error(
                "Error deleting asset model %s. Here's why %s",
                asset_model_id,
                err.response["Error"]["Message"],
            )
            raise

```

- Einzelheiten zur API finden Sie [DeleteAssetModel](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

TRY.
    lo_ios->deleteassetmodel(
        iv_assetmodelid = iv_asset_model_id
    ).
    MESSAGE 'IoT SiteWise asset model deleted.' TYPE 'I'.
CATCH /aws1/cx_rt_generic.
    MESSAGE 'Unable to delete asset model.' TYPE 'E'.
ENDTRY.

```

- Einzelheiten zur API finden Sie [DeleteAssetModel](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Service mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **DeleteGateway** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie DeleteGateway verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Kennenlernen der Grundlagen](#)

CLI

AWS CLI

Um ein Gateway zu löschen

Im folgenden Beispiel für delete-gateway wird ein Gateway gelöscht.

```
aws iotsitewise delete-gateway \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Mit diesem Befehl wird keine Ausgabe zurückgegeben.

Weitere Informationen finden Sie unter [Daten mithilfe eines Gateways](#) aufnehmen im AWS SiteWise IoT-Benutzerhandbuch.

- Einzelheiten zur API finden Sie [DeleteGateway](#) in der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

/**
 * Deletes the specified gateway.
 *
 * @param gatewayId the ID of the gateway to delete.
 * @return a {@link CompletableFuture} that represents a {@link
DeleteGatewayResponse} result.. The calling code
 *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *
 * <p>
 *
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
By calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<DeleteGatewayResponse> deleteGatewayAsync(String
gatewayId) {
    DeleteGatewayRequest deleteGatewayRequest =
DeleteGatewayRequest.builder()
        .gatewayId(gatewayId)
        .build();

    return getAsyncClient().deleteGateway(deleteGatewayRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to delete gateway: {}",
exception.getCause().getMessage());
            }
        });
}

```

- Einzelheiten zur API finden Sie [DeleteGateway](#) in der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
import {
  DeleteGatewayCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create an SSM document.
 * @param {{ content: string, name: string, documentType?: DocumentType }}
 */
export const main = async ({ gatewayId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    await client.send(
      new DeleteGatewayCommand({
        gatewayId: gatewayId, // The ID of the Gateway to describe.
      }),
    );
    console.log("Gateway deleted successfully.");
    return { gatewayDeleted: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. The Gateway could not be found. Please check the Gateway Id.`,
      );
    } else {
      throw caught;
    }
  }
};
```

- Einzelheiten zur API finden Sie [DeleteGateway](#) in der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
class IoTSitewiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)
```

```
def delete_gateway(self, gateway_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Gateway.

    :param gateway_id: The ID of the gateway to delete.
    """
    try:
        self.iotsitewise_client.delete_gateway(gatewayId=gateway_id)
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Gateway %s does not exist.", gateway_id)
        else:
            logger.error(
                "Error deleting gateway %s. Here's why %s",
                gateway_id,
                err.response["Error"]["Message"],
            )
        raise
```

- Einzelheiten zur API finden Sie [DeleteGateway](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
TRY.
    lo_ios->deletgateway(
        iv_gatewayid = iv_gateway_id
    ).
    MESSAGE 'IoT SiteWise gateway deleted.' TYPE 'I'.
CATCH /aws1/cx_iosresourcenotfoundex.
    MESSAGE 'Gateway does not exist.' TYPE 'E'.
```

```
ENDTRY.
```

- Einzelheiten zur API finden Sie [DeleteGateway](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Service mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **DescribeAssetModel** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeAssetModel` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Kennenlernen der Grundlagen](#)

CLI

AWS CLI

So beschreiben Sie ein Komponentenmodell

Im folgenden Beispiel für `describe-asset-model` wird das Windpark-Komponentenmodell beschrieben.

```
aws iotsitewise describe-asset-model \
  --asset-model-id a1b2c3d4-5678-90ab-cdef-2222EXAMPLE
```

Ausgabe:

```
{
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-2222EXAMPLE",
  "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE",
  "assetModelName": "Wind Farm Model",
  "assetModelDescription": "Represents a wind farm that comprises many wind turbines",
  "assetModelProperties": [
```



```

    {
      "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
      "name": "Total Generated Power",
      "dataType": "DOUBLE",
      "unit": "kW",
      "type": {
        "metric": {
          "expression": "sum(power)",
          "variables": [
            {
              "name": "power",
              "value": {
                "propertyId": "a1b2c3d4-5678-90ab-
cdef-66666EXAMPLE",
                "hierarchyId": "a1b2c3d4-5678-90ab-
cdef-77777EXAMPLE"
              }
            }
          ]
        },
        "window": {
          "tumbling": {
            "interval": "1h"
          }
        }
      }
    },
    {
      "id": "a1b2c3d4-5678-90ab-cdef-88888EXAMPLE",
      "name": "Region",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": " "
        }
      }
    }
  ],
  "assetModelHierarchies": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
      "name": "Wind Turbines",
      "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
    }
  ]
}

```

```

    ],
    "assetModelCreationDate": 1575671284.0,
    "assetModelLastUpdateDate": 1575671988.0,
    "assetModelStatus": {
        "state": "ACTIVE"
    }
}

```

Weitere Informationen finden Sie unter [Beschreibung eines bestimmten Asset-Modells](#) im AWS SiteWise IoT-Benutzerhandbuch.

- Einzelheiten zur API finden Sie [DescribeAssetModel](#) in der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

/**
 * Retrieves the property IDs associated with a specific asset model.
 *
 * @param assetModelId the ID of the asset model that defines the properties.
 * @return a {@link CompletableFuture} that represents a {@link Map} result
 * that associates the property name to the
 *         property ID. The calling code can attach callbacks, then handle the
 * result or exception by calling
 *         {@link CompletableFuture#join()} or {@link
 * CompletableFuture#get()}.
 *
 * <p>
 *         If any completion stage in this method throws an exception, the
 * method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
 * By calling
 *         {@link CompletionException#getCause()}, the calling code can
 * access the original exception.
 */

```

```

    public CompletableFuture<Map<String, String>> getPropertyIds(String
assetModelId) {
        ListAssetModelPropertiesRequest modelPropertiesRequest =
ListAssetModelPropertiesRequest.builder().assetModelId(assetModelId).build();
        return getAsyncClient().listAssetModelProperties(modelPropertiesRequest)
            .handle((response, throwable) -> {
                if (response != null) {
                    return response.assetModelPropertySummaries().stream()
                        .collect(Collectors
                            .toMap(AssetModelPropertySummary::name,
AssetModelPropertySummary::id));
                } else {
                    logger.error("Error occurred while fetching property IDs:
{}.", throwable.getCause().getMessage());
                    throw (CompletionException) throwable;
                }
            });
    }
}

```

- Einzelheiten zur API finden Sie [DescribeAssetModel](#) in der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

import {
    DescribeAssetModelCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Describe an asset model.

```

```

* @param {{ assetModelId : string }}
*/
export const main = async ({ assetModelId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const { assetModelDescription } = await client.send(
      new DescribeAssetModelCommand({
        assetModelId: assetModelId, // The ID of the Gateway to describe.
      }),
    );
    console.log("Asset model information retrieved successfully.");
    return { assetModelDescription: assetModelDescription };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. The asset model could not be found. Please check the asset model id.`,
      );
    } else {
      throw caught;
    }
  }
};

```

- Einzelheiten zur API finden Sie [DescribeAssetModel](#) in der AWS SDK für JavaScript API-Referenz.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Service mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **DescribeGateway** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `DescribeGateway` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Kennenlernen der Grundlagen](#)

CLI

AWS CLI

So beschreiben Sie ein Gateway

Im folgenden Beispiel für `describe-gateway` wird ein Gateway beschrieben.

```
aws iotsitewise describe-gateway \  
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Ausgabe:

```
{  
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",  
  "gatewayName": "ExampleCorpGateway",  
  "gatewayArn": "arn:aws:iotsitewise:us-west-2:123456789012:gateway/  
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",  
  "gatewayPlatform": {  
    "greengrass": {  
      "groupArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/  
groups/a1b2c3d4-5678-90ab-cdef-1b1b1EXAMPLE"  
    }  
  },  
  "gatewayCapabilitySummaries": [  
    {  
      "capabilityNamespace": "iotsitewise:opcuacollector:1",  
      "capabilitySyncStatus": "IN_SYNC"  
    }  
  ],  
  "creationDate": 1588369971.457,  
  "lastUpdateDate": 1588369971.457  
}
```

Weitere Informationen finden Sie unter [Daten mithilfe eines Gateways](#) aufnehmen im AWS SiteWise IoT-Benutzerhandbuch.

- Einzelheiten zur API finden Sie [DescribeGateway](#) in der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
/**
 * Describes the specified gateway.
 *
 * @param gatewayId the ID of the gateway to describe.
 * @return a {@link CompletableFuture} that represents a {@link
DescribeGatewayResponse} result. The calling code
 *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
By calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<DescribeGatewayResponse> describeGatewayAsync(String
gatewayId) {
    DescribeGatewayRequest request = DescribeGatewayRequest.builder()
        .gatewayId(gatewayId)
        .build();

    return getAsyncClient().describeGateway(request)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("An error occurred during the describeGateway
method: {}", exception.getCause().getMessage());
            }
        });
}
```

- Einzelheiten zur API finden Sie [DescribeGateway](#) in der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
import {
  DescribeGatewayCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create an SSM document.
 * @param {{ content: string, name: string, documentType?: DocumentType }}
 */
export const main = async ({ gatewayId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const { gatewayDescription } = await client.send(
      new DescribeGatewayCommand({
        gatewayId: gatewayId, // The ID of the Gateway to describe.
      }),
    );
    console.log("Gateway information retrieved successfully.");
    return { gatewayDescription: gatewayDescription };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. The Gateway could not be found. Please check the Gateway Id.`,
      );
    } else {
      throw caught;
    }
  }
};
```

```

    }
  }
};

```

- Einzelheiten zur API finden Sie [DescribeGateway](#) in der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSiteWiseWrapper":
        """
        Creates an IoTSiteWiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSiteWiseWrapper initialized with the default
        AWS IoT SiteWise client.

```



```

"""
    iotsitewise_client = boto3.client("iotsitewise")
    return cls(iotsitewise_client)

def describe_gateway(self, gateway_id: str) -> Dict[str, Any]:
    """
    Describes an AWS IoT SiteWise Gateway.

    :param gateway_id: The ID of the gateway to describe.
    :return: A dictionary containing information about the gateway.
    """
    try:
        response =
self.iotsitewise_client.describe_gateway(gatewayId=gateway_id)
        return response
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Gateway %s does not exist.", gateway_id)
        else:
            logger.error(
                "Error describing gateway %s. Here's why %s",
                gateway_id,
                err.response["Error"]["Message"],
            )
        raise

```

- Einzelheiten zur API finden Sie [DescribeGateway](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

TRY.
  oo_result = lo_ios->describegateway(
    iv_gatewayid = iv_gateway_id
  ). " oo_result is returned for testing purposes. "
  MESSAGE 'Retrieved gateway description.' TYPE 'I'.
CATCH /aws1/cx_iosresourcenotfoundex.
  MESSAGE 'Gateway does not exist.' TYPE 'E'.
ENDTRY.

```

- Einzelheiten zur API finden Sie [DescribeGateway](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Service mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **GetAssetPropertyValue** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie **GetAssetPropertyValue** verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Kennenlernen der Grundlagen](#)

CLI

AWS CLI

So rufen Sie den aktuellen Wert einer Komponenteneigenschaft ab

Im folgenden Beispiel für `get-asset-property-value` wird die aktuelle Gesamtleistung einer Windturbinenkomponente abgerufen.

```

aws iotsitewise get-asset-property-value \
  --asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE \
  --property-id a1b2c3d4-5678-90ab-cdef-6666EXAMPLE

```

Ausgabe:

```
{
  "propertyValue": {
    "value": {
      "doubleValue": 6890.8677520453875
    },
    "timestamp": {
      "timeInSeconds": 1580853000,
      "offsetInNanos": 0
    },
    "quality": "GOOD"
  }
}
```

Weitere Informationen finden Sie unter [Abfragen aktueller Objekteigenschaftswerte](#) im AWS SiteWise IoT-Benutzerhandbuch.

- Einzelheiten zur API finden Sie unter [GetAssetPropertyValue AWS CLI](#) Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
/**
 * Fetches the value of an asset property.
 *
 * @param propId the ID of the asset property to fetch.
 * @param assetId the ID of the asset to fetch the property value for.
 * @return a {@link CompletableFuture} that represents a {@link Double}
 * result. The calling code can attach
 *         callbacks, then handle the result or exception by calling {@link
 * CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *
 * <p>
 *         If any completion stage in this method throws an exception, the
 * method logs the exception cause and keeps
```

```

    *           it available to the calling code as a {@link CompletionException}.
    By calling
    *           {@link CompletionException#getCause()}, the calling code can
    access the original exception.
    */
    public CompletableFuture<Double> getAssetPropValueAsync(String propId, String
    assetId) {
        GetAssetPropertyValueRequest assetPropertyValueRequest =
        GetAssetPropertyValueRequest.builder()
            .propertyId(propId)
            .assetId(assetId)
            .build();

        return getAsyncClient().getAssetPropertyValue(assetPropertyValueRequest)
            .handle((response, exception) -> {
                if (exception != null) {
                    logger.error("Error occurred while fetching property
    value: {}.", exception.getCause().getMessage());
                    throw (CompletionException) exception;
                }
                return response.propertyValue().value().doubleValue();
            });
    }

```

- Einzelheiten zur API finden Sie [GetAssetPropertyValue](#) in der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

import {
    GetAssetPropertyValueCommand,
    IoTSiteWiseClient,

```

```
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Describe an asset property value.
 * @param {{ entryId : string }}
 */
export const main = async ({ entryId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new GetAssetPropertyValueCommand({
        entryId: entryId, // The ID of the Gateway to describe.
      }),
    );
    console.log("Asset property information retrieved successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. The asset property entry could not be found. Please check the entry id.`,
      );
    } else {
      throw caught;
    }
  }
};
```

- Einzelheiten zur API finden Sie [GetAssetPropertyValue](#) in der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSiteWiseWrapper":
        """
        Creates an IoTSiteWiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSiteWiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def get_asset_property_value(
        self, asset_id: str, property_id: str
    ) -> Dict[str, Any]:
        """
        Gets the value of an AWS IoT SiteWise Asset Property.

        :param asset_id: The ID of the asset.
        :param property_id: The ID of the property.
        :return: A dictionary containing the value of the property.
        """
        try:
            response = self.iotsitewise_client.get_asset_property_value(
                assetId=asset_id, propertyId=property_id
            )
            return response["propertyValue"]

```

```

except ClientError as err:
    if err.response["Error"]["Code"] == "ResourceNotFoundException":
        logger.error(
            "Asset %s or property %s does not exist.", asset_id,
property_id
        )
    else:
        logger.error(
            "Error getting asset property value. Here's why %s",
            err.response["Error"]["Message"],
        )
    raise

```

- Einzelheiten zur API finden Sie [GetAssetPropertyValue](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel](#) einrichten und ausführen.

```

TRY.
    oo_result = lo_ios->getassetpropertyvalue(
        iv_assetid = iv_asset_id
        iv_propertyid = iv_property_id
    ). " oo_result is returned for testing purposes. "
    MESSAGE 'Retrieved asset property value.' TYPE 'I'.
CATCH /aws1/cx_iosresourcenotfoundex.
    MESSAGE 'Asset or property does not exist.' TYPE 'E'.
ENDTRY.

```

- Einzelheiten zur API finden Sie [GetAssetPropertyValue](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Service mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

ListAssetModelProperties Mit einem AWS SDK verwenden

Das folgende Codebeispiel zeigt, wie es verwendet wird `ListAssetModelProperties`.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Kennenlernen der Grundlagen](#)

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
TRY.  
    oo_result = lo_ios->listassetmodelproperties(  
        iv_assetmodelid = iv_asset_model_id  
    ). " oo_result is returned for testing purposes. "  
    DATA(lt_properties) = oo_result->get_assetmodelpropertysums( ).  
    MESSAGE 'Retrieved list of asset model properties.' TYPE 'I'.  
CATCH /aws1/cx_rt_generic.  
    MESSAGE 'Unable to list asset model properties.' TYPE 'E'.  
ENDTRY.
```

- Einzelheiten zur API finden Sie [ListAssetModelProperties](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Service mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwendung **ListAssetModels** mit einem AWS SDK oder CLI

Die folgenden Code-Beispiele zeigen, wie `ListAssetModels` verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Kennenlernen der Grundlagen](#)

CLI

AWS CLI

So listen Sie alle Komponentenmodelle auf

Das folgende `list-asset-models` Beispiel listet alle Vermögensmodelle auf, die in Ihrem AWS Konto in der aktuellen Region definiert sind.

```
aws iotsitewise list-asset-models
```

Ausgabe:

```
{
  "assetModelSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "name": "Wind Farm Model",
      "description": "Represents a wind farm that comprises many wind turbines",
      "creationDate": 1575671284.0,
      "lastUpdateDate": 1575671988.0,
      "status": {
        "state": "ACTIVE"
      }
    },
    {
      "id": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",

```

```

        "name": "Wind Turbine Model",
        "description": "Represents a wind turbine manufactured by Example
Corp",
        "creationDate": 1575671207.0,
        "lastUpdateDate": 1575686273.0,
        "status": {
            "state": "ACTIVE"
        }
    }
]
}

```

Weitere Informationen finden Sie unter [Auflisten aller Asset-Modelle](#) im AWS SiteWise IoT-Benutzerhandbuch.

- Einzelheiten zur API finden Sie [ListAssetModels](#) in der AWS CLI Befehlsreferenz.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

/**
 * Retrieves the asset model ID for the given asset model name.
 *
 * @param assetModelName the name of the asset model for the ID.
 * @return a {@link CompletableFuture} that represents a {@link String}
result of the asset model ID or null if the
 *         asset model cannot be found. The calling code can attach
callbacks, then handle the result or exception
 *         by calling {@link CompletableFuture#join()} or {@link
CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
By calling

```

```

    *      {@link CompletionException#getCause()}, the calling code can
    access the original exception.
    */
    public CompletableFuture<String> getAssetModelIdAsync(String assetModelName)
    {
        ListAssetModelsRequest listAssetModelsRequest =
        ListAssetModelsRequest.builder().build();
        return getAsyncClient().listAssetModels(listAssetModelsRequest)
            .handle((listAssetModelsResponse, exception) -> {
                if (exception != null) {
                    logger.error("Failed to retrieve Asset Model ID: {}",
exception.getCause().getMessage());
                    throw (CompletionException) exception;
                }
                for (AssetModelSummary assetModelSummary :
listAssetModelsResponse.assetModelSummaries()) {
                    if (assetModelSummary.name().equals(assetModelName)) {
                        return assetModelSummary.id();
                    }
                }
                return null;
            });
    }

```

- Einzelheiten zur API finden Sie [ListAssetModels](#) in der AWS SDK for Java 2.x API-Referenz.

JavaScript

SDK für JavaScript (v3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```

import {
    ListAssetModelsCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

```

```
/**
 * List asset models.
 * @param {{ assetModelTypes : array }}
 */
export const main = async ({ assetModelTypes = [] }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new ListAssetModelsCommand({
        assetModelTypes: assetModelTypes, // The model types to list
      }),
    );
    console.log("Asset model types retrieved successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "IoTSiteWiseError") {
      console.warn(
        `${caught.message}. There was a problem listing the asset model types.`,
      );
    } else {
      throw caught;
    }
  }
};
```

- Einzelheiten zur API finden Sie [ListAssetModels](#) in der AWS SDK für JavaScript API-Referenz.

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
class IoTSitewiseWrapper:
```

```

    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def list_asset_models(self) -> List[Dict[str, Any]]:
        """
        Lists all AWS IoT SiteWise Asset Models.

        :return: A list of dictionaries containing information about each asset
        model.

        """
        try:
            asset_models = []
            paginator =
self.iotsitewise_client.get_paginator("list_asset_models")
            pages = paginator.paginate()
            for page in pages:
                asset_models.extend(page["assetModelSummaries"])
            return asset_models
        except ClientError as err:

```

```
logger.error(  
    "Error listing asset models. Here's why %s",  
    err.response["Error"]["Message"],  
)  
raise
```

- Einzelheiten zur API finden Sie [ListAssetModels](#) in AWS SDK for Python (Boto3) API Reference.

SAP ABAP

SDK für SAP ABAP

Note

Es gibt noch mehr dazu. GitHub Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS -Code-Beispiel-](#) einrichten und ausführen.

```
TRY.  
    oo_result = lo_ios->listassetmodels( ). " oo_result is returned for  
testing purposes. "  
    DATA(lt_asset_models) = oo_result->get_assetmodelsummaries( ).  
    MESSAGE 'Retrieved list of asset models.' TYPE 'I'.  
CATCH /aws1/cx_rt_generic.  
    MESSAGE 'Unable to list asset models.' TYPE 'E'.  
ENDTRY.
```

- Einzelheiten zur API finden Sie [ListAssetModels](#) in der API-Referenz zum AWS SDK für SAP ABAP.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden Sie diesen Service mit einem AWSSDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Sicherheit bei AWS IoT SiteWise

Cloud-Sicherheit hat AWS höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und 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, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Third-party Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der [AWS](#) und . Weitere Informationen zu den Compliance-Programmen, die für gelten AWS IoT SiteWise, finden Sie unter [AWS Leistungen im Umfang nach Compliance-Programmen AWS](#) .
- **Sicherheit in der Cloud** — Ihre Verantwortung richtet sich nach dem AWS Dienst, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung anwenden können AWS IoT SiteWise. In den folgenden Themen erfahren Sie, wie Sie die Konfiguration vornehmen AWS IoT SiteWise, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, die Sie bei der Überwachung und Sicherung Ihrer AWS IoT SiteWise Ressourcen unterstützen.

Topics

- [Datenschutz in AWS IoT SiteWise](#)
- [Datenverschlüsselung in AWS IoT SiteWise](#)
- [Identitäts- und Zugriffsmanagement für AWS IoT SiteWise](#)
- [Konformitätsprüfung für AWS IoT SiteWise](#)
- [Resilienz in AWS IoT SiteWise](#)
- [Sicherheit der Infrastruktur in AWS IoT SiteWise](#)
- [Konfiguration und Schwachstellenanalyse in AWS IoT SiteWise](#)
- [VPC-Endpunkte für AWS IoT SiteWise](#)

- [Bewährte Sicherheitsmethoden für AWS IoT SiteWise](#)

Datenschutz in AWS IoT SiteWise

Für den Datenschutz in AWS IoT SiteWise gilt das [Modell der geteilten Verantwortung](#) das von AWS. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der AWS Cloud alle Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter [Häufig gestellte Fragen zum Datenschutz](#) . Weitere Informationen zum Datenschutz in Europa finden Sie im [Zentrum für die Datenschutz-Grundverordnung \(DSGVO\)](#).

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter [Arbeiten mit CloudTrail Pfaden](#) im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-3](#).

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit

der Konsole, der API AWS IoT SiteWise oder den SDKs arbeiten oder diese anderweitig AWS-Services verwenden. AWS CLI AWS Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Themen

- [Datenschutz für den NetzwerkverkehrAWS IoT SiteWise](#)
- [AWS IoT SiteWiseAssistentin bei der Verbesserung des Business Service](#)

Datenschutz für den NetzwerkverkehrAWS IoT SiteWise

Verbindungen zwischen AWS IoT SiteWise und lokalen Anwendungen, wie SiteWise Edge-Gateways, werden über TLS-Verbindungen (Transport Layer Security) gesichert. Weitere Informationen finden Sie unter [Datenverschlüsselung bei der Übertragung fürAWS IoT SiteWise](#).

AWS IoT SiteWiseunterstützt keine Verbindungen zwischen Availability Zones innerhalb einer AWS Region oder Verbindungen zwischen AWS Konten.

Sie können IAM Identity Center jeweils nur in einer Region konfigurieren. SiteWise Monitor stellt eine Verbindung zu der Region her, die Sie für IAM Identity Center konfiguriert haben. Das bedeutet, dass Sie eine Region für den Zugriff auf das IAM Identity Center verwenden, aber Sie können Portale in jeder Region erstellen.

AWS IoT SiteWiseAssistentin bei der Verbesserung des Business Service

AWS IoT SiteWiseAssistant verwendet Kundendaten nicht zur Verbesserung des Services oder zur Verbesserung der zugrunde liegenden LLMs.

Datenverschlüsselung inAWS IoT SiteWise

Datenverschlüsselung bezieht sich auf den Schutz von Daten während der Übertragung (bei der Übertragung zu und von AWS IoT SiteWise und zwischen SiteWise Edge-Gateways und -Servern) und im Ruhezustand (während sie auf lokalen Geräten oder in AWS Diensten gespeichert werden). Sie können Daten während der Übertragung mit TLS (Transport Layer Security) oder im Ruhezustand mit clientseitiger Verschlüsselung schützen.

 Note

AWS IoT SiteWiseDie Edge-Verarbeitung macht APIs verfügbar, die auf SiteWise Edge-Gateways gehostet werden und auf die über das lokale Netzwerk zugegriffen werden kann. Diese APIs werden über eine TLS-Verbindung verfügbar gemacht, die durch ein Serverzertifikat gestützt wird, das dem AWS IoT SiteWise Edge-Connector gehört. Für die Client-Authentifizierung verwenden diese APIs ein Passwort für die Zugriffskontrolle. Der private Schlüssel des Serverzertifikats und das Passwort für die Zugriffskontrolle werden beide auf der Festplatte gespeichert. AWS IoT SiteWiseDie Edge-Verarbeitung stützt sich auf die Dateisystemverschlüsselung, um die Sicherheit dieser Anmeldeinformationen im Ruhezustand zu gewährleisten.

Weitere Informationen zur serverseitigen Verschlüsselung und zur clientseitigen Verschlüsselung finden Sie in den unten aufgeführten Themen.

Themen

- [Verschlüsselung im Ruhezustand inAWS IoT SiteWise](#)
- [Datenverschlüsselung bei der Übertragung fürAWS IoT SiteWise](#)
- [Schlüsselverwaltung inAWS IoT SiteWise](#)

Verschlüsselung im Ruhezustand inAWS IoT SiteWise

AWS IoT SiteWisespeichert Ihre Daten in der AWS Cloud und auf AWS IoT SiteWise Edge-Gateways.

Daten im Ruhezustand imAWS Cloud

AWS IoT SiteWisespeichert Daten in anderen AWS Diensten, die Daten im Ruhezustand standardmäßig verschlüsseln. Encryption at Rest ist in AWS Key Management Service (AWS KMS) integriert, um den Verschlüsselungsschlüssel zu verwalten, der zum Verschlüsseln Ihrer Objektwerte und Aggregatwerte in verwendet wird. AWS IoT SiteWise Sie können sich dafür entscheiden, einen vom Kunden verwalteten Schlüssel zur Verschlüsselung von Vermögenswerten und Aggregatwerten in zu verwenden. AWS IoT SiteWise Sie können Ihren Verschlüsselungsschlüssel über AWS KMS erstellen, verwalten und einsehen.

Sie können einen auswählen, AWS-eigener Schlüssel um Ihre Daten zu verschlüsseln, oder einen vom Kunden verwalteten Schlüssel wählen, um Ihre Immobilienwerte und aggregierten Werte zu verschlüsseln:

Funktionsweise

Encryption at Rest ist in die Verwaltung des Verschlüsselungsschlüssels integriert, der zur Verschlüsselung Ihrer Daten verwendet wird. AWS KMS

- **AWS-eigener Schlüssel**— Standard-Verschlüsselungsschlüssel. AWS IoT SiteWise besitzt diesen Schlüssel. Sie können diesen Schlüssel nicht in Ihrem AWS Konto einsehen. Sie können auch keine Operationen mit dem Schlüssel in den AWS CloudTrail Protokollen sehen. Sie können diesen Schlüssel ohne zusätzliche Kosten verwenden.
- **Vom Kunden verwalteter Schlüssel** — Der Schlüssel wird in Ihrem Konto gespeichert, das Sie erstellen, besitzen und verwalten. Sie haben die volle Kontrolle über den KMS-Schlüssel. Es fallen AWS KMS zusätzliche Gebühren an.

AWS-eigene Schlüssel

AWS-eigene Schlüssel sind nicht in Ihrem Konto gespeichert. Sie sind Teil einer Sammlung von KMS-Schlüsseln, die AWS Eigentümer sind und für die Verwendung in mehreren AWS Konten verwaltet werden. AWS-Dienste, die Sie AWS-eigene Schlüssel zum Schutz Ihrer Daten verwenden können.

Sie können ihre Verwendung nicht einsehen, verwalten AWS-eigene Schlüssel, verwenden oder überprüfen. Sie müssen jedoch keine Arbeit verrichten oder Programme ändern, um die Schlüssel zu schützen, mit denen Ihre Daten verschlüsselt werden.

Für die Nutzung wird Ihnen keine monatliche Gebühr oder Nutzungsgebühr berechnet AWS-eigene Schlüssel, und sie werden auch nicht auf die AWS KMS Kontingente für Ihr Konto angerechnet.

Kundenseitig verwaltete Schlüssel

Kundenverwaltete Schlüssel sind KMS-Schlüssel in Ihrem , die Sie erstellen, besitzen und verwalten. Sie haben die volle Kontrolle über diese KMS-Schlüssel, z. B. über die folgenden:

- Festlegung und Pflege ihrer wichtigsten Richtlinien, IAM-Richtlinien und Zuschüsse
- Sie aktivieren und deaktivieren
- Rotation ihres kryptographischen Materials

- Hinzufügen von -Tags
- Aliase erstellen, die auf sie verweisen
- Sie für das Löschen planen

Sie können auch Amazon CloudTrail CloudWatch Logs verwenden, um die Anfragen zu verfolgen, die in Ihrem Namen AWS IoT SiteWise AWS KMS an gesendet werden.

Wenn Sie vom Kunden verwaltete Schlüssel verwenden, müssen Sie AWS IoT SiteWise Zugriff auf den in Ihrem Konto gespeicherten KMS-Schlüssel gewähren. AWS IoT SiteWise verwendet Umschlagverschlüsselung und Schlüsselhierarchie, um Daten zu verschlüsseln. Ihr AWS KMS Verschlüsselungsschlüssel wird verwendet, um den Stammschlüssel dieser Schlüsselhierarchie zu verschlüsseln. Weitere Informationen zur [Envelope-Verschlüsselung](#) finden Sie im AWS Key Management Service-Entwicklerhandbuch.

Die folgende Beispielrichtlinie gewährt einem Benutzer die AWS IoT SiteWise Erlaubnis, in Ihrem Namen einen vom Kunden verwalteten Schlüssel zu erstellen. Wenn Sie Ihren Schlüssel erstellen, müssen Sie die `kms:DescribeKey` Aktionen `kms:CreateGrant` und zulassen.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1603902045292",
      "Action": [
        "kms:CreateGrant",
        "kms:DescribeKey"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Der Verschlüsselungskontext für Ihren erstellten Zuschuss verwendet Ihre Konto-ID `aws:iotsitewise:subscriberId` und Ihre Konto-ID.

Daten im Ruhezustand auf SiteWise Edge-Gateways

AWS IoT SiteWiseGateways speichern die folgenden Daten im lokalen Dateisystem:

- Informationen zur OPC UA-Quellkonfiguration
- Der Satz von OPC UA-Datenstream-Pfaden aus verbundenen OPC UA-Quellen
- Industriedaten werden zwischengespeichert, wenn das SiteWise Edge-Gateway die Verbindung zum Internet verliert

SiteWise Edge-Gateways laufen auf AWS IoT Greengrass. AWS IoT Greengrass stützt sich auf Unix-Dateiberechtigungen und vollständige Festplattenverschlüsselung (falls aktiviert), um Daten zu schützen, die sich auf dem Kern befinden. Es liegt in Ihrer Verantwortung, das Dateisystem und das Gerät zu sichern.

Verschlüsselt AWS IoT Greengrass jedoch lokale Kopien Ihrer vom Secrets Manager abgerufenen OPC UA-Servergeheimnisse. Weitere Informationen finden Sie unter [Secrets-Verschlüsselung](#) im AWS IoT Greengrass Version 1 Entwicklerhandbuch.

Weitere Informationen zur Verschlüsselung ruhender AWS IoT Greengrass Kerne finden Sie unter [Verschlüsselung im Ruhezustand](#) im AWS IoT Greengrass Version 1 Entwicklerhandbuch.

Datenverschlüsselung bei der Übertragung für AWS IoT SiteWise

AWS IoT SiteWise verwendet Verschlüsselung bei der Übertragung, um die zwischen Ihren Geräten, Gateways und der AWS Cloud übertragenen Daten zu sichern. Die Kommunikation mit AWS IoT SiteWise wird mit HTTPS und TLS 1.2 verschlüsselt, wodurch sichergestellt wird, dass Ihre Daten vertraulich bleiben und vor unbefugtem Zugriff oder Abfangen geschützt sind.

Es gibt drei Kommunikationsarten, bei denen Daten übertragen werden:

- [Über das Internet](#) — Die Kommunikation zwischen lokalen Geräten (einschließlich SiteWise Edge-Gateways) und AWS IoT SiteWise erfolgt verschlüsselt.
- [Über das lokale Netzwerk](#) — Die Kommunikation zwischen unseren OpsHub SiteWise Anwendungen und SiteWise Edge-Gateways ist immer verschlüsselt. Die Kommunikation zwischen der SiteWise Monitor-Anwendung, die in Ihrem Browser ausgeführt wird, und den SiteWise Edge-Gateways ist immer verschlüsselt. Die Kommunikation zwischen SiteWise Edge-Gateways und OPC UA-Quellen kann verschlüsselt werden.

- [Zwischen Komponenten auf SiteWise Edge-Gateways](#) — Die Kommunikation zwischen AWS IoT Greengrass Komponenten auf SiteWise Edge-Gateways ist nicht verschlüsselt.

Themen

- [Daten in Übertragung über das Internet](#)
- [Daten in Übertragung über das lokale Netzwerk](#)
- [Daten werden zwischen lokalen Komponenten auf SiteWise Edge übertragen](#)

Daten in Übertragung über das Internet

AWS IoT SiteWise verwendet Transport Layer Security (TLS), um die gesamte Kommunikation über das Internet zu verschlüsseln. Alle an die AWS Cloud gesendeten Daten werden über eine TLS-Verbindung unter Verwendung der Protokolle MQTT oder HTTPS gesendet, sodass sie standardmäßig sicher sind. SiteWise Edge-Gateways, die laufen AWS IoT Greengrass, und Benachrichtigungen über Eigenschaftswerte verwenden das AWS IoT Transportsicherheitsmodell. Weitere Informationen finden Sie unter [Transportsicherheit](#) im AWS IoT-Entwicklerhandbuch.

Daten in Übertragung über das lokale Netzwerk

SiteWise Edge-Gateways folgen den OPC UA-Spezifikationen für die Kommunikation mit lokalen OPC UA-Quellen. Sie sind dafür verantwortlich, Ihre Quellen für die Verwendung eines Nachrichtensicherheitsmodus zu konfigurieren, der Daten während der Übertragung verschlüsselt.

Wenn Sie einen Sicherheitsmodus für Signnachrichten wählen, werden Daten, die zwischen SiteWise Edge-Gateways und Quellen übertragen werden, signiert, aber nicht verschlüsselt. Wenn Sie einen Sicherheitsmodus zum Signieren und Verschlüsseln von Nachrichten wählen, werden die Daten, die zwischen SiteWise Edge-Gateways und Quellen übertragen werden, signiert und verschlüsselt. Weitere Informationen zur Konfiguration von Quellen finden Sie unter [Fügen Sie Datenquellen zu Ihrem hinzu AWS IoT SiteWise Edge-Gateway](#).

Die Kommunikation zwischen der Edge-Konsolenanwendung und den SiteWise Edge-Gateways wird immer mit TLS verschlüsselt. Der SiteWise Edge-Connector auf dem SiteWise Edge-Gateway generiert und speichert ein selbstsigniertes Zertifikat, um eine TLS-Verbindung mit der Edge-Konsole für AWS IoT SiteWise die Anwendung herstellen zu können. Sie müssen dieses Zertifikat für die AWS IoT SiteWise Anwendung von Ihrem SiteWise Edge-Gateway auf die Edge-Konsole kopieren, bevor Sie die Anwendung mit dem SiteWise Edge-Gateway verbinden. Dadurch wird sichergestellt, dass

die Edge-Konsole für die AWS IoT SiteWise Anwendung überprüfen kann, ob sie eine Verbindung zu Ihrem vertrauenswürdigen SiteWise Edge-Gateway hergestellt hat.

Zusätzlich zu TLS für Geheimhaltung und Serverauthentizität verwendet SiteWise Edge das SigV4-Protokoll, um die Authentizität der Edge-Konsolenanwendung festzustellen. Der SiteWise Edge-Connector auf dem SiteWise Edge-Gateway akzeptiert und speichert ein Passwort, um eingehende Verbindungen von der Edge-Konsolenanwendung, der SiteWise Monitor-Anwendung, die in Browsern ausgeführt wird, und anderen Clients, die auf dem SDK basieren, verifizieren zu können.

AWS IoT SiteWise

Weitere Informationen zum Generieren des Kennworts und des Serverzertifikats finden Sie unter [the section called “Gateways verwalten”](#).

Daten werden zwischen lokalen Komponenten auf SiteWise Edge übertragen

SiteWise Edge-Gateways laufen auf AWS IoT Greengrass, wodurch Daten, die lokal auf dem AWS IoT Greengrass Core ausgetauscht werden, nicht verschlüsselt werden, da die Daten das Gerät nicht verlassen. Dazu gehört auch die Kommunikation zwischen AWS IoT Greengrass Komponenten wie dem AWS IoT SiteWise Connector. Weitere Informationen finden Sie unter [Daten auf dem Kerngerät](#) im AWS IoT Greengrass Version 1 Entwicklerhandbuch.

Schlüsselverwaltung in AWS IoT SiteWise

AWS IoT SiteWise Verwaltung von Cloud-Schlüsseln

Wird standardmäßig Von AWS verwaltete Schlüssel zum Schutz Ihrer Daten in der AWS Cloud AWS IoT SiteWise verwendet. Sie können Ihre Einstellungen aktualisieren, um einige Daten mit einem vom Kunden verwalteten Schlüssel zu verschlüsseln. AWS IoT SiteWise Sie können Ihren Verschlüsselungsschlüssel über AWS Key Management Service (AWS KMS) erstellen, verwalten und einsehen.

AWS IoT SiteWise unterstützt serverseitige Verschlüsselung mit vom Kunden verwalteten Schlüsseln AWS KMS, um die folgenden Daten zu verschlüsseln:

- Eigenschaftswerte von Vermögenswerten
- Werte aggregieren

 Note

Andere Daten und Ressourcen werden mit der Standardverschlüsselung mit Schlüsseln verschlüsselt, die von verwaltet werdenAWS IoT SiteWise. Dieser Schlüssel wird im AWS IoT SiteWise Konto gespeichert.

Weitere Informationen finden Sie unter [Was istAWS Key Management Service?](#) im AWS Key Management ServiceEntwicklerhandbuch.

Aktivieren Sie die Verschlüsselung mit vom Kunden verwalteten Schlüsseln

Um vom Kunden verwaltete Schlüssel mit verwenden zu könnenAWS IoT SiteWise, müssen Sie Ihre AWS IoT SiteWise Einstellungen aktualisieren.

Um die Verschlüsselung mit KMS-Schlüsseln zu aktivieren

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie Kontoeinstellungen und dann Bearbeiten, um die Seite Kontoeinstellungen bearbeiten zu öffnen.
3. Wählen Sie als Typ des Verschlüsselungsschlüssels die Option Anderen AWS KMS Schlüssel auswählen aus. Dies ermöglicht die Verschlüsselung mit vom Kunden verwalteten Schlüsseln, die in gespeichert sindAWS KMS.

 Note

Derzeit können Sie die vom Kunden verwaltete Schlüsselverschlüsselung nur für Immobilienwerte und aggregierte Werte verwenden.

4. Wählen Sie Ihren KMS-Schlüssel mit einer der folgenden Optionen:
 - Um einen vorhandenen KMS-Schlüssel zu verwenden — Wählen Sie Ihren KMS-Schlüsselalias aus der Liste aus.
 - Um einen neuen KMS-Schlüssel zu erstellen — Wählen Sie Create an AWS KMS key.

 Note

Dadurch wird das AWS KMS-Dashboard geöffnet. Weitere Informationen zum Erstellen eines KMS-Schlüssels finden Sie unter [Creating Keys](#) im AWS Key Management Service Developer Guide.

5. Wählen Sie Speichern, um Ihre Einstellungen zu aktualisieren.

SiteWise Schlüsselverwaltung für das Edge-Gateway

SiteWise Edge-Gateways laufen auf und AWS IoT Greengrass Kerengeräte verwenden öffentliche und private Schlüssel AWS IoT Greengrass, um sich bei der AWS Cloud zu authentifizieren und lokale Geheimnisse wie OPC UA-Authentifizierungsgeheimnisse zu verschlüsseln. Weitere Informationen finden Sie unter [Schlüsselverwaltung](#) im Entwicklerhandbuch. AWS IoT Greengrass Version 1

Identitäts- und Zugriffsmanagement für AWS IoT SiteWise

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service, den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Ressourcen zu verwenden. AWS IoT SiteWise IAM ist ein Programm AWS-Service, das Sie ohne zusätzliche Kosten nutzen können.

Themen

- [Publikum für AWS IoT SiteWise Sicherheit](#)
- [Authentifizieren Sie sich mit Identitäten in AWS IoT SiteWise](#)
- [Wie AWS IoT SiteWise funktioniert mit IAM](#)
- [AWS verwaltete Richtlinien für AWS IoT SiteWise](#)
- [Verwenden Sie serviceverknüpfte Rollen für AWS IoT SiteWise](#)
- [Richten Sie Berechtigungen für Ereignisalarme ein in AWS IoT SiteWise](#)
- [Cross-service verwirrter Stellvertreter, Prävention in AWS IoT SiteWise](#)
- [Fehlerbehebung AWS IoT SiteWise Identität und Zugriff](#)

Publikum für AWS IoT SiteWise Sicherheit

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von Ihrer Rolle ab:

- Servicebenutzer – Fordern Sie von Ihrem Administrator Berechtigungen an, wenn Sie nicht auf Features zugreifen können (siehe [Fehlerbehebung AWS IoT SiteWise Identität und Zugriff](#)).
- Serviceadministrator – Bestimmen Sie den Benutzerzugriff und stellen Sie Berechtigungsanfragen (siehe [Wie AWS IoT SiteWise funktioniert mit IAM](#)).
- IAM-Administrator – Schreiben Sie Richtlinien zur Zugriffsverwaltung (siehe [AWS IoT SiteWise Beispiele für identitätsbasierte Richtlinien](#)).

Authentifizieren Sie sich mit Identitäten in AWS IoT SiteWise

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen sich als IAM-Benutzer authentifizieren oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich als föderierte Identität anmelden, indem Sie Anmeldeinformationen aus einer Identitätsquelle wie AWS IAM Identity Center (IAM Identity Center), Single Sign-On-Authentifizierung oder Anmeldeinformationen verwenden. Google/Facebook Weitere Informationen zum Anmelden finden Sie unter [So melden Sie sich bei Ihrem AWS-Konto an](#) im Benutzerhandbuch für AWS-Anmeldung.

AWS bietet für den programmatischen Zugriff ein SDK und eine CLI zum kryptografischen Signieren von Anfragen. Weitere Informationen finden Sie unter [AWS Signature Version 4 for API requests](#) im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie ein AWS-Konto erstellen, beginnen Sie mit einer Anmeldeidentität, dem sogenannten AWS-Konto Root-Benutzer, der vollständigen Zugriff auf alle AWS-Services Ressourcen hat. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Tasks that require root user credentials](#) im IAM-Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wir empfehlen die Verwendung temporärer Anmeldeinformationen anstelle von IAM-

Benutzern mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter [Erfordern, dass menschliche Benutzer für den Zugriff AWS mithilfe temporärer Anmeldeinformationen einen Verbund mit einem Identitätsanbieter](#) verwenden müssen.

Eine [IAM-Gruppe](#) spezifiziert eine Sammlung von IAM-Benutzern und erleichtert die Verwaltung von Berechtigungen für große Gruppen von Benutzern. Weitere Informationen finden Sie unter [Anwendungsfälle für IAM-Benutzer](#) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität mit spezifischen Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie [von einer Benutzer- zu einer IAM-Rolle \(Konsole\) wechseln](#) AWS CLI oder einen AWS API-Vorgang aufrufen. Weitere Informationen finden Sie unter [Methoden, um eine Rolle zu übernehmen](#) im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für den Verbundbenutzer-Zugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, serviceübergreifenden Zugriff und Anwendungen, die auf Amazon EC2 laufen. Weitere Informationen finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

WieAWS IoT SiteWisefunktioniert mit IAM

Bevor Sie AWS Identity and Access Management (IAM) zur Verwaltung des Zugriffs auf verwendenAWS IoT SiteWise, sollten Sie wissen, mit welchen IAM-Funktionen Sie arbeiten können.

IAM-Feature	Unterstützt von? AWS IoT SiteWise
Identity-based Richtlinien mit Berechtigungen auf Ressourcenebene	Ja
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja

IAM-Feature	Unterstützt von? AWS IoT SiteWise
Bedingungsschlüssel für die Richtlinie	Ja
Resource-based Richtlinien	Nein
Zugriffssteuerungslisten (ACLs)	Nein
Tags-based Autorisierung (ABAC)	Ja
Temporäre Anmeldeinformationen	Ja
Zugriffssitzungen weiterleiten (FAS)	Ja
Service-linked Rollen	Ja
Servicerollen	Ja

Einen allgemeinen Überblick darüber, wie AWS IoT SiteWise und andere AWS Dienste mit IAM funktionieren, finden Sie im [IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren](#).

Inhalt

- [AWS IoT SiteWise IAM-Rollen](#)
 - [Verwenden Sie temporäre Anmeldeinformationen mit AWS IoT SiteWise](#)
 - [Zugriffssitzungen \(FAS\) weiterleiten für AWS IoT SiteWise](#)
 - [Service-linked Rollen](#)
 - [Servicerollen](#)
 - [Wählen Sie eine IAM-Rolle in AWS IoT SiteWise](#)
- [Autorisierung basiert auf AWS IoT SiteWise Tags](#)
- [AWS IoT SiteWise Identitätsbasierte Richtlinien](#)
 - [Richtlinienaktionen](#)

- [BatchPutAssetPropertyValue Autorisierung](#)
- [Richtlinienressourcen](#)
- [Bedingungsschlüssel für die Richtlinie](#)
- [Beispiele](#)
- [AWS IoT SiteWiseBeispiele für identitätsbasierte Richtlinien](#)
 - [Best Practices für Richtlinien](#)
 - [Verwenden Sie denAWS IoT SiteWiseKonsole](#)
 - [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)
 - [Erlaubt Benutzern, Daten in Assets in einer Hierarchie aufzunehmen](#)
 - [AnzeigenAWS IoT SiteWiseauf Tags basierende Vermögenswerte](#)
- [Verwalten Sie den Zugriff mithilfe von Richtlinien inAWS IoT SiteWise](#)
 - [Identity-based Richtlinien](#)
 - [Resource-based Richtlinien](#)
 - [Zugriffssteuerungslisten \(ACLs\)](#)
 - [Weitere Richtlinientypen](#)
 - [Mehrere Richtlinientypen](#)

AWS IoT SiteWiseIAM-Rollen

Eine [IAM-Rolle](#) ist eine Entität in Ihrem AWS-Konto mit spezifischen Berechtigungen.

Verwenden Sie temporäre Anmeldeinformationen mitAWS IoT SiteWise

Sie können temporäre Anmeldeinformationen verwenden, um sich über einen Verbund anzumelden, eine IAM-Rolle anzunehmen oder eine kontenübergreifende Rolle anzunehmen. Sie erhalten temporäre Sicherheitsanmeldedaten, indem Sie AWS STS API-Operationen wie [AssumeRole](#)oder aufrufen [GetFederationToken](#).

AWS IoT SiteWiseunterstützt die Verwendung temporärer Anmeldeinformationen.

SiteWise Monitor unterstützt Verbundbenutzer beim Zugriff auf Portale. Portalbenutzer authentifizieren sich mit ihren IAM Identity Center- oder IAM-Anmeldeinformationen.

 Important

Benutzer oder Rollen müssen über die `iotsitewise:DescribePortal` Berechtigung verfügen, sich beim Portal anzumelden.

Wenn sich ein Benutzer bei einem Portal anmeldet, generiert SiteWise Monitor eine Sitzungsrichtlinie, die die folgenden Berechtigungen bietet:

- Read-only Zugriff auf die Assets und Asset-Daten AWS IoT SiteWise in Ihrem Konto, auf die die Rolle dieses Portals Zugriff gewährt.
- Zugriff auf Projekte in diesem Portal, auf die der Benutzer Administratorzugriff (Projektbesitzer) oder schreibgeschützten Zugriff (Projektanzeiger) hat.

Weitere Informationen zu föderierten Portalbenutzerberechtigungen finden Sie unter [Verwenden Sie Servicerollen fürAWS IoT SiteWise Monitor](#).

Zugriffssitzungen (FAS) weiterleiten fürAWS IoT SiteWise

Unterstützt Forward Access Sessions (FAS): Ja

Forward Access Sessions (FAS) verwenden die Berechtigungen des Prinzipals, der einen aufruftAWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. Einzelheiten zu den Richtlinien für FAS-Anforderungen finden Sie unter [Zugriffssitzungen weiterleiten](#).

Service-linked Rollen

[Service-linked Rollen](#) ermöglichen es AWS Diensten, auf Ressourcen in anderen Diensten zuzugreifen, um eine Aktion in Ihrem Namen auszuführen. Mit Diensten verknüpfte Rollen werden in Ihrem AWS Konto angezeigt und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für serviceverknüpfte Rollen anzeigen, aber nicht bearbeiten.

AWS IoT SiteWiseunterstützt dienstbezogene Rollen. Details zum Erstellen oder Verwalten von serviceverknüpften AWS IoT SiteWise-Rollen finden Sie unter [Verwenden Sie serviceverknüpfte Rollen fürAWS IoT SiteWise](#).

Servicerollen

Dieses Feature ermöglicht einem Service das Annehmen einer [Servicerolle](#) in Ihrem Namen. Diese Rolle gewährt dem Service Zugriff auf Ressourcen in anderen Diensten, um eine Aktion in Ihrem Namen auszuführen. Servicerollen werden in Ihrem AWS Konto angezeigt und gehören dem Konto. Dies bedeutet, dass ein IAM-Administrator die Berechtigungen für diese Rolle ändern kann. Dies kann jedoch die Funktionalität des Dienstes beeinträchtigen.

AWS IoT SiteWise verwendet eine Servicerolle, um SiteWise Monitor-Portalbenutzern den Zugriff auf einige Ihrer AWS IoT SiteWise Ressourcen in Ihrem Namen zu ermöglichen. Weitere Informationen finden Sie unter [Verwenden Sie Servicerollen für AWS IoT SiteWise Monitor](#).

Sie müssen über die erforderlichen Berechtigungen verfügen, bevor Sie AWS IoT Events Alarmmodelle in erstellen können AWS IoT SiteWise. Weitere Informationen finden Sie unter [Richten Sie Berechtigungen für Ereignisalarme ein in AWS IoT SiteWise](#).

Wählen Sie eine IAM-Rolle in AWS IoT SiteWise

Wenn Sie eine portal Ressource in erstellen AWS IoT SiteWise, müssen Sie eine Rolle auswählen, auf die die Verbundbenutzer Ihres SiteWise Monitor-Portals in Ihrem Namen zugreifen AWS IoT SiteWise können. Wenn Sie zuvor eine Servicerolle erstellt haben, AWS IoT SiteWise erhalten Sie eine Liste mit Rollen, aus denen Sie wählen können. Andernfalls können Sie beim Erstellen eines Portals eine Rolle mit den erforderlichen Berechtigungen erstellen. Es ist wichtig, eine Rolle auszuwählen, die den Zugriff auf Ihre Komponenten und Komponentendaten ermöglicht. Weitere Informationen finden Sie unter [Verwenden Sie Servicerollen für AWS IoT SiteWise Monitor](#).

Autorisierung basiert auf AWS IoT SiteWise Tags

Sie können Tags an AWS IoT SiteWise Ressourcen anhängen oder Tags in einer Anfrage an übergeben AWS IoT SiteWise. Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungelement einer [Richtlinie Tag-Informationen](#) an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden. Weitere Informationen über das Markieren von AWS IoT SiteWise-Ressourcen mit Tags finden Sie unter [Kennzeichnen Sie Ihre AWS IoT SiteWise Ressourcen](#).

Ein Beispiel für eine identitätsbasierte Richtlinie zur Einschränkung des Zugriffs auf eine Ressource auf der Grundlage der Markierungen dieser Ressource finden Sie unter [Anzeigen AWS IoT SiteWise auf Tags basierende Vermögenswerte](#).

AWS IoT SiteWise Identitätsbasierte Richtlinien

Mit IAM-Richtlinien können Sie kontrollieren, wer was tun kann. AWS IoT SiteWise Sie können entscheiden, welche Aktionen zulässig sind oder nicht, und spezifische Bedingungen für diese Aktionen festlegen. Sie können beispielsweise Regeln dafür festlegen, wer Informationen sehen oder ändern kann AWS IoT SiteWise. AWS IoT SiteWise unterstützt bestimmte Aktionen, Ressourcen und Bedingungsschlüssel. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

Richtlinienaktionen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Nehmen Sie Aktionen in eine Richtlinie auf, um Berechtigungen zur Ausführung des zugehörigen Vorgangs zu erteilen.

Bei Richtlinienaktionen wird vor der Aktion das folgende Präfix AWS IoT SiteWise verwendet: `iotsitewise:`. Um beispielsweise jemandem die Erlaubnis zu erteilen, Objektdaten im Rahmen des `BatchPutAssetPropertyValue` API-Vorgangs hochzuladen, nehmen Sie die `iotsitewise:BatchPutAssetPropertyValue` Aktion in seine Richtlinie auf. AWS IoT SiteWise Richtlinienerklärungen müssen `Action` entweder ein `NotAction` Oder-Element enthalten. AWS IoT SiteWise definiert einen eigenen Satz von Aktionen, die Aufgaben beschreiben, die Sie mit diesem Dienst ausführen können.

Um mehrere -Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie folgendermaßen durch Kommas.

```
"Action": [  
  "iotsitewise:action1",  
  "iotsitewise:action2"  
]
```

Sie können auch Platzhalter (*) verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

```
"Action": "iotsitewise:Describe*"
```


Eine Liste der AWS IoT SiteWise [Aktionen finden Sie AWS IoT SiteWise im IAM-Benutzerhandbuch unter Definierte Aktionen von](#).

BatchPutAssetPropertyValue Autorisierung

AWS IoT SiteWise autorisiert den Zugriff auf die [BatchPutAssetPropertyValue](#) Aktion auf ungewöhnliche Weise. Wenn Sie bei den meisten Aktionen den Zugriff zulassen oder verweigern, gibt diese Aktion einen Fehler zurück, wenn keine Berechtigungen erteilt wurden. Mit BatchPutAssetPropertyValue können Sie in einer einzigen API-Anfrage mehrere Dateneinträge an verschiedene Assets und Asset-Eigenschaften senden. AWS IoT SiteWise autorisiert jede Dateneingabe unabhängig. Fügt für jeden einzelnen Eintrag, bei dem die Autorisierung in der Anfrage fehlschlägt, AWS IoT SiteWise, eine Fehlerliste `AccessDeniedException` in die zurückgegebene Liste ein. AWS IoT SiteWise empfängt die Daten für jeden Eintrag, der autorisiert wurde und erfolgreich ist, auch wenn ein anderer Eintrag in derselben Anfrage fehlschlägt.

Important

Gehen Sie wie folgt vor, bevor Sie Daten in einen Datenstream aufnehmen:

- Autorisieren Sie die `time-series` Ressource, wenn Sie einen Eigenschaftsalias verwenden, um den Datenstrom zu identifizieren.
- Autorisieren Sie die `asset` Ressource, wenn Sie eine Asset-ID verwenden, um das Asset zu identifizieren, das die zugehörige Asset-Eigenschaft enthält.

Richtlinienressourcen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Als Best Practice geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen \(ARN\)](#) an. Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*" 
```

Jede IAM-Richtlinienanweisung gilt für die Ressourcen, die Sie mithilfe ihrer ARNs angegeben haben. Ein ARN weist die folgende allgemeine Syntax auf:

```
arn:${Partition}:${Service}:${Region}:${Account}:${ResourceType}/${ResourcePath}
```

Weitere Informationen zum Format von ARNs finden Sie unter [Identifizieren von AWS Ressourcen mit Amazon-Ressourcennamen \(ARNs\)](#).

Um beispielsweise die Komponente mit der ID a1b2c3d4-5678-90ab-cdef-2222EXAMPLE in Ihrer Anweisung anzugeben, verwenden Sie den folgenden ARN.

```
"Resource": "arn:aws:iotsitewise:region:123456789012:asset/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE"
```

Um alle Datenströme anzugeben, die zu einem bestimmten Konto gehören, verwenden Sie den Platzhalter (*):

```
"Resource": "arn:aws:iotsitewise:region:123456789012:time-series/*"
```

Um alle Komponenten anzugeben, die zu einem bestimmten Konto gehören, verwenden Sie den Platzhalter (*).

```
"Resource": "arn:aws:iotsitewise:region:123456789012:asset/*"
```

Einige AWS IoT SiteWise Aktionen, z. B. die zum Erstellen von Ressourcen, können für eine bestimmte Ressource nicht ausgeführt werden. In diesen Fällen müssen Sie den Platzhalter (*) verwenden.

```
"Resource": "*"
```

Um mehrere Ressourcen in einer einzigen Anweisung anzugeben, trennen Sie die ARNs durch Kommata voneinander.

```
"Resource": [  
  "resource1",  
  "resource2"
```

]

Eine Liste der AWS IoT SiteWise Ressourcentypen und ihrer ARNs finden Sie AWS IoT SiteWise im IAM-Benutzerhandbuch unter [Ressourcentypen definiert von](#). Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter [Von AWS IoT SiteWise definierte Aktionen](#).

Bedingungsschlüssel für die Richtlinie

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Condition` gibt an, wann Anweisungen auf der Grundlage definierter Kriterien ausgeführt werden. 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. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontextschlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

Important

Einige API-Aktionen verwenden mehrere Ressourcen. Viele Bedingungsschlüssel sind jedoch ressourcenspezifisch. Wenn Sie eine Richtlinienanweisung mit einem Bedingungsschlüssel schreiben, legen Sie über das `Resource`-Element der Anweisung fest, für welche Ressource der Bedingungsschlüssel gültig ist. Andernfalls verhindert die Richtlinie möglicherweise, dass Benutzer die Aktion überhaupt ausführen können, da die Bedingungsprüfung für die Ressourcen fehlschlägt, für die der Bedingungsschlüssel nicht gilt. Wenn Sie keine Ressource angeben möchten oder über das `Action`-Element Ihrer Richtlinie mehrere API-Aktionen hinzugefügt haben, müssen Sie mit dem `...IfExists`-Bedingungstyp sicherstellen, dass der Bedingungsschlüssel für die Ressourcen, die ihn nicht verwenden, ignoriert wird. [Weitere Informationen finden Sie unter... IfExists](#) Bedingungen im IAM-Benutzerhandbuch.

AWS IoT SiteWise definiert seinen eigenen Satz von Bedingungsschlüsseln und unterstützt auch die Verwendung einiger globaler Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontext-Schlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

AWS IoT SiteWiseBedingungsschlüssel

Bedingungsschlüssel	Description	Typen
<code>iotsitewise:isAssociatedWithAssetProperty</code>	Ob Datenströme mit einer Anlageneigenschaft verknüpft sind. Verwenden Sie diesen Bedingungsschlüssel, um Berechtigungen zu definieren, die auf dem Vorhandensein einer zugehörigen Asset-Eigenschaft für Datenstreams basieren. Beispielwert: <code>true</code>	Zeichenfolge
<code>iotsitewise:assetHierarchyPath</code>	Der Hierarchiepfad der Komponente, bei dem es sich um eine Zeichenfolge von Komponenten-IDs handelt, die jeweils durch einen Schrägstrich getrennt sind. Verwenden Sie diesen Bedingungsschlüssel, um Berechtigungen basierend auf einer Teilmenge Ihrer Hierarchie aller Komponenten in Ihrem Konto zu definieren. Beispielwert: <code>/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE/a1b2c3d4-5678-90ab-cdef-6666EXAMPLE</code>	Zeichenfolge
<code>iotsitewise:propertyId</code>	Die ID einer Komponenteneigenschaft. Verwenden Sie diesen Bedingungsschlüssel,	Zeichenfolge

Bedingungsschlüssel	Description	Typen
	um Berechtigungen basierend auf der angegebenen Eigenschaft eines Komponentenmodells zu definieren. Dieser Bedingungsschlüssel gilt für alle Komponenten dieses Modells. Beispielwert: a1b2c3d4-5678-90ab-cdef-3333EXAMPLE	
iotsitewise:childAssetId	ID einer Komponente, die als untergeordnetes Element mit einer anderen Komponente verknüpft ist. Verwenden Sie diesen Bedingungsschlüssel, um Berechtigungen basierend auf untergeordneten Komponenten zu definieren. Um Berechtigungen basierend auf übergeordneten Komponenten zu definieren, verwenden Sie den Ressourcensegment einer Richtlinienganweisung. Beispielwert: a1b2c3d4-5678-90ab-cdef-6666EXAMPLE	Zeichenfolge

Bedingungsschlüssel	Description	Typen
<code>iotsitewise:iam</code>	Der ARN einer IAM-Identität beim Auflisten von Zugriffsrichtlinien. Verwenden Sie diesen Bedingungsschlüssel, um Zugriffsrichtlinienberechtigungen für eine IAM-Identität zu definieren. Beispielwert: <code>arn:aws:iam::123456789012:user/JohnDoe</code>	Zeichenfolge, Null
<code>iotsitewise:propertyAlias</code>	Der Alias, der eine Asset-Eigenschaft oder einen Datenstrom identifiziert. Verwenden Sie diesen Bedingungsschlüssel, um Berechtigungen auf der Grundlage des Alias zu definieren.	Zeichenfolge
<code>iotsitewise:user</code>	Die ID eines IAM Identity Center-Benutzers beim Auflisten von Zugriffsrichtlinien. Verwenden Sie diesen Bedingungsschlüssel, um Zugriffsrichtlinienberechtigungen für einen IAM Identity Center-Benutzer zu definieren. Beispielwert: <code>a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE</code>	Zeichenfolge, Null

Bedingungsschlüssel	Description	Typen
<code>iotsitewise:group</code>	Die ID einer IAM Identity Center-Gruppe bei der Auflistung der Zugriffsrichtlinien. Verwenden Sie diesen Bedingungsschlüssel, um Zugriffsrichtlinienberechtigungen für eine IAM Identity Center-Gruppe zu definieren. Beispielwert: a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbbEXAMPLE	Zeichenfolge, Null
<code>iotsitewise:portal</code>	Die ID eines Portals in einer Zugriffsrichtlinie. Verwenden Sie diesen Bedingungsschlüssel, um Zugriffsrichtlinienberechtigungen basierend auf einem Portal zu definieren. Beispielwert: a1b2c3d4-5678-90ab-cdef-7777EXAMPLE	Zeichenfolge, Null

Bedingungsschlüssel	Description	Typen
<code>iotsitewise:project</code>	Die ID eines Projekts in einer Zugriffsrichtlinie oder die ID eines Projekts für ein Dashboard. Verwenden Sie diesen Bedingungsschlüssel, um Dashboard- oder Zugriffsrichtlinienberechtigungen basierend auf einem Projekt zu definieren. Beispielwert: <code>a1b2c3d4-5678-90ab-cdef-8888EXAMPLE</code>	Zeichenfolge, Null

Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Aktionen definiert von AWS IoT SiteWise](#).

Beispiele

Beispiele für AWS IoT SiteWise identitätsbasierte Richtlinien finden Sie unter [AWS IoT SiteWiseBeispiele für identitätsbasierte Richtlinien](#)

AWS IoT SiteWiseBeispiele für identitätsbasierte Richtlinien

Standardmäßig sind Entitäten (Benutzer und Rollen) nicht berechtigt, AWS IoT SiteWise Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben mit der AWS-Managementkonsole, AWS Command Line Interface (AWS CLI) oder AWS API ausführen. Um Berechtigungen anzupassen, muss ein AWS Identity and Access Management (IAM-) Administrator wie folgt vorgehen:

1. Erstellen Sie IAM-Richtlinien, die Benutzern und Rollen die Berechtigung gewähren, bestimmte API-Operationen für Ressourcen auszuführen, die sie benötigen.
2. Ordnen Sie diese Richtlinien den Benutzern oder Gruppen zu, für die diese Berechtigungen erforderlich sind.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von Richtlinien auf der JSON-Registerkarte](#) im IAM-Benutzerhandbuch.

Themen

- [Best Practices für Richtlinien](#)
- [Verwenden Sie den AWS IoT SiteWise-Konsole](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)
- [Erlaubt Benutzern, Daten in Assets in einer Hierarchie aufzunehmen](#)
- [Anzeigen AWS IoT SiteWise auf Tags basierende Vermögenswerte](#)

Best Practices für Richtlinien

Identity-based Richtlinien legen fest, ob jemand AWS IoT SiteWise Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Beachten Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Richtlinien und Empfehlungen:

- Erste Schritte mit AWS verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS-verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads 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 unter [Von AWS verwaltete Richtlinien](#) oder [Von AWS verwaltete Richtlinien für Auftragsfunktionen](#) im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung 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 von IAM zum Anwenden von Berechtigungen finden Sie unter [Richtlinien und Berechtigungen in IAM](#) im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben,

um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. CloudFormation B. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.

- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter [Richtlinienvvalidierung mit IAM Access Analyzer](#) im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter [Sicherer API-Zugriff mit MFA](#) im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter [Best Practices für die Sicherheit in IAM](#) im IAM-Benutzerhandbuch.

Verwenden Sie den AWS IoT SiteWise Konsole

Um auf die AWS IoT SiteWise Konsole zugreifen zu können, benötigen Sie grundlegende Berechtigungen. Mit diesen Berechtigungen können Sie Details zu den AWS IoT SiteWise Ressourcen in Ihrem AWS Konto einsehen und verwalten.

Wenn Sie eine zu restriktive Richtlinie festlegen, funktioniert die Konsole für Benutzer oder Rollen (Entitäten), für die diese Richtlinie gilt, möglicherweise nicht wie erwartet. Um sicherzustellen, dass diese Entitäten die AWS IoT SiteWise Konsole weiterhin verwenden können, fügen Sie ihnen die [AWSIoTSiteWiseConsoleFullAccess](#) verwaltete Richtlinie bei oder definieren Sie entsprechende Berechtigungen für diese Entitäten. Weitere Informationen finden Sie unter [Hinzufügen von Berechtigungen zu einem Benutzer](#) im IAM-Benutzerhandbuch.

Wenn Entitäten nur die AWS Command Line Interface (CLI) oder die AWS IoT SiteWise API und nicht die Konsole verwenden, benötigen sie diese Mindestberechtigungen nicht. Geben Sie ihnen in diesem Fall einfach Zugriff auf die spezifischen Aktionen, die sie für ihre API-Aufgaben benötigen.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie beinhaltet Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API AWS CLI oder AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsForUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

Erlaubt Benutzern, Daten in Assets in einer Hierarchie aufzunehmen

In diesem Beispiel möchten Sie einem Benutzer in Ihrem AWS Konto Zugriff darauf gewähren, Daten für alle Asset-Eigenschaften in einer bestimmten Asset-Hierarchie zu schreiben, beginnend mit dem Stammobjekt. `a1b2c3d4-5678-90ab-cdef-2222EXAMPLE`. Die Richtlinie erteilt dem Benutzer die `iotsitewise:BatchPutAssetPropertyValue`-Berechtigung. Diese Richtlinie verwendet den `iotsitewise:assetHierarchyPath`-Bedingungsschlüssel, um den Zugriff auf Komponenten einzuschränken, deren Hierarchiepfad mit der Komponenten oder ihren abhängigen Elementen übereinstimmt.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PutAssetPropertyValuesForHierarchy",
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "arn:aws:iotsitewise:*:*:asset/*",
      "Condition": {
        "StringLike": {
          "iotsitewise:assetHierarchyPath": [
            "/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE",
            "/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE/*"
          ]
        }
      }
    }
  ]
}
```

Anzeigen AWS IoT SiteWise auf Tags basierende Vermögenswerte

Verwenden Sie Bedingungen in Ihrer identitätsbasierten Richtlinie, um den Zugriff auf AWS IoT SiteWise Ressourcen auf der Grundlage von Tags zu steuern. Dieses Beispiel zeigt, wie Sie eine Richtlinie erstellen, die das Anzeigen von Assets ermöglicht. Die Berechtigung wird jedoch nur erteilt, wenn das Tag der Komponente `Owner` den Wert des Benutzernamens dieses Benutzers hat. Diese Richtlinie gewährt auch die Erlaubnis, diese Aktion auf der Konsole abzuschließen.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListAllAssets",
      "Effect": "Allow",
      "Action": [
        "iotsitewise:ListAssets",
        "iotsitewise:ListAssociatedAssets"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DescribeAssetIfOwner",
      "Effect": "Allow",
      "Action": "iotsitewise:DescribeAsset",
      "Resource": "arn:aws:iotsitewise:*:*:asset/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Owner": "${aws:username}"
        }
      }
    }
  ]
}
```

Hängen Sie diese Richtlinie den Benutzern in Ihrem Konto an. Wenn ein benannter Benutzer `richard-roe` versucht, ein AWS IoT SiteWise Asset aufzurufen, muss das Asset mit `Owner=richard-roe` oder markiert werden `owner=richard-roe`. Andernfalls wird Richard der Zugriff verweigert. Bei den Schlüsselnamen der Bedingungs-tags wird nicht zwischen Groß- und Kleinschreibung unterschieden. `Owner` entspricht also sowohl als `Owner` auch `owner`. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.

Verwalten Sie den Zugriff mithilfe von Richtlinien in AWS IoT SiteWise

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie definiert Berechtigungen, wenn sie mit einer Identität oder

Ressource verknüpft sind. AWS bewertet diese Richtlinien, wenn ein Principal eine Anfrage stellt. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Mit Hilfe von Richtlinien legen Administratoren fest, wer Zugriff auf was hat, indem sie definieren, welches Prinzipal welche Aktionen auf welchen Ressourcen und unter welchen Bedingungen durchführen darf.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie zu Rollen hinzu, die die Benutzer dann übernehmen können. IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identity-based Richtlinien

Identity-based Richtlinien sind Richtliniendokumente für JSON-Berechtigungen, die Sie an eine Identität (Benutzer, Gruppe oder Rolle) anhängen. Diese Richtlinien steuern, welche Aktionen Identitäten für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

Identity-based Richtlinien können Inline-Richtlinien (direkt in eine einzelne Identität eingebettet) oder verwaltete Richtlinien (eigenständige Richtlinien, die mehreren Identitäten zugeordnet sind) sein. Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter [Choose between managed policies and inline policies](#) im IAM-Benutzerhandbuch.

Resource-based Richtlinien

Resource-based Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele hierfür sind Vertrauensrichtlinien für IAM-Rollen und Amazon S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#).

Resource-based Richtlinien sind Inline-Richtlinien, die sich in diesem Dienst befinden. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Services, die ACLs unterstützen. AWS WAF Weitere Informationen“ zu ACLs finden Sie unter [Zugriffskontrollliste \(ACL\) – Übersicht](#) (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

Weitere Richtlinientypen

AWS unterstützt zusätzliche Richtlinientypen, mit denen die maximalen Berechtigungen festgelegt werden können, die durch gängigere Richtlinientypen gewährt werden:

- **Berechtigungsgrenzen** – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im IAM-Benutzerhandbuch.
- **Service-Kontrollrichtlinien (SCPs)** – SCPs legen die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in AWS Organizations fest. Weitere Informationen finden Sie unter [Service-Kontrollrichtlinien](#) im AWS Organizations-Benutzerhandbuch.
- **Ressourcen-Kontrollrichtlinien (RCPs)** – RCPs definieren die maximale Anzahl an Berechtigungen, die Ressourcen in Ihren Konten zur Verfügung stehen. Weitere Informationen finden Sie unter [Ressourcen-Kontrollrichtlinien](#) im AWS Organizations-Benutzerhandbuch.
- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere Arten von Richtlinien für eine Anfrage gelten, sind die sich daraus ergebenden Berechtigungen schwieriger zu verstehen. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie unter [Bewertungslogik für Richtlinien](#) im IAM-Benutzerhandbuch.

AWSverwaltete Richtlinien für AWS IoT SiteWise

Vereinfachen Sie das Hinzufügen von Berechtigungen für Benutzer, Gruppen und Rollen mithilfe AWS verwalteter Richtlinien, anstatt Richtlinien selbst zu schreiben. Es erfordert Zeit und

Fachwissen, um vom [Kunden verwaltete IAM-Richtlinien zu erstellen](#), die Ihrem Team präzise Berechtigungen gewähren. Für eine schnellere Einrichtung sollten Sie in Erwägung ziehen, unsere AWS verwalteten Richtlinien für allgemeine Anwendungsfälle zu verwenden. Suchen Sie in Ihrem AWS Konto nach AWS verwalteten Richtlinien. Weitere Informationen zu verwalteten AWS-Richtlinien finden Sie unter [Verwaltete AWS-Richtlinien](#) im IAM-Leitfaden.

AWS Die Dienste kümmern sich um die Aktualisierung und Pflege der AWS verwalteten Richtlinien, sodass Sie die Berechtigungen dieser Richtlinien nicht ändern können. Gelegentlich AWS IoT SiteWise können Berechtigungen hinzugefügt werden, um neuen Funktionen gerecht zu werden, was sich auf alle Identitäten auswirkt, an die die Richtlinie angehängt ist. Solche Aktualisierungen treten häufig bei der Einführung neuer Dienste oder Funktionen auf. Berechtigungen werden jedoch niemals entfernt, um sicherzustellen, dass Ihre Einstellungen intakt bleiben.

AWS Unterstützt außerdem verwaltete Richtlinien für Jobfunktionen, die sich über mehrere Dienste erstrecken. Die `ReadOnlyAccess` AWS verwaltete Richtlinie bietet beispielsweise schreibgeschützten Zugriff auf alle AWS Dienste und Ressourcen. Wenn ein Dienst eine neue Funktion startet, werden nur Leseberechtigungen für neue Operationen und Ressourcen AWS hinzugefügt. Eine Liste mit Beschreibungen der Richtlinien für Jobfunktionen finden Sie im IAM-Benutzerhandbuch unter [AWS Verwaltete Richtlinien für Jobfunktionen](#).

AWS verwaltete Richtlinie: `AWSIoTSiteWiseReadOnlyAccess`

Verwenden Sie die `AWSIoTSiteWiseReadOnlyAccess` AWS verwaltete Richtlinie, um schreibgeschützten Zugriff auf zu gewähren. AWS IoT SiteWise

Sie können die `AWSIoTSiteWiseReadOnlyAccess`-Richtlinie an Ihre IAM-Identitäten anfügen.

Service-level permissions

Diese Richtlinie bietet schreibgeschützten Zugriff auf AWS IoT SiteWise, einschließlich der Berechtigungen zur Ausführung schreibgeschützter SQL-Abfragen. In dieser Richtlinie sind keine anderen Dienstberechtigungen enthalten.

JSON

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



```
"Action": [  
    "iotsitewise:BatchGetAssetPropertyAggregates",  
    "iotsitewise:BatchGetAssetPropertyValue",  
    "iotsitewise:BatchGetAssetPropertyValueHistory",  
    "iotsitewise:DescribeAccessPolicy",  
    "iotsitewise:DescribeAction",  
    "iotsitewise:DescribeAsset",  
    "iotsitewise:DescribeAssetCompositeModel",  
    "iotsitewise:DescribeAssetModel",  
    "iotsitewise:DescribeAssetModelCompositeModel",  
    "iotsitewise:DescribeAssetModelInterfaceRelationship",  
    "iotsitewise:DescribeAssetProperty",  
    "iotsitewise:DescribeBulkImportJob",  
    "iotsitewise:DescribeComputationModel",  
    "iotsitewise:DescribeComputationModelExecutionSummary",  
    "iotsitewise:DescribeDashboard",  
    "iotsitewise:DescribeDataset",  
    "iotsitewise:DescribeDefaultEncryptionConfiguration",  
    "iotsitewise:DescribeExecution",  
    "iotsitewise:DescribeGateway",  
    "iotsitewise:DescribeGatewayCapabilityConfiguration",  
    "iotsitewise:DescribeLoggingOptions",  
    "iotsitewise:DescribePortal",  
    "iotsitewise:DescribeProject",  
    "iotsitewise:DescribeStorageConfiguration",  
    "iotsitewise:DescribeTimeSeries",  
    "iotsitewise:GetAssetPropertyAggregates",  
    "iotsitewise:GetAssetPropertyValue",  
    "iotsitewise:GetAssetPropertyValueHistory",  
    "iotsitewise:GetInterpolatedAssetPropertyValues",  
    "iotsitewise:ListAccessPolicies",  
    "iotsitewise:ListActions",  
    "iotsitewise:ListAssetModelCompositeModels",  
    "iotsitewise:ListAssetModelProperties",  
    "iotsitewise:ListAssetModels",  
    "iotsitewise:ListAssetProperties",  
    "iotsitewise:ListAssetRelationships",  
    "iotsitewise:ListAssets",  
    "iotsitewise:ListAssociatedAssets",  
    "iotsitewise:ListBulkImportJobs",  
    "iotsitewise:ListCompositionRelationships",  
    "iotsitewise:ListComputationModelDataBindingUsages",  
    "iotsitewise:ListComputationModelResolveToResources",  
    "iotsitewise:ListComputationModels",
```

```

        "iotsitewise:ListDashboards",
        "iotsitewise:ListDatasets",
        "iotsitewise:ListExecutions",
        "iotsitewise:ListGateways",
        "iotsitewise:ListInterfaceRelationships",
        "iotsitewise:ListPortals",
        "iotsitewise:ListProjectAssets",
        "iotsitewise:ListProjects",
        "iotsitewise:ListTagsForResource",
        "iotsitewise:ListTimeSeries"
    ],
    "Resource": "*"
}
]
}

```

AWSverwaltete Richtlinie: AWSServiceRoleForIoTSiteWise

Die `AWSServiceRoleForIoTSiteWise` Rolle verwendet die `AWSServiceRoleForIoTSiteWise` Richtlinie mit den folgenden Berechtigungen. Diese Richtlinie:

- Ermöglicht AWS IoT SiteWise die Bereitstellung von SiteWise Edge-Gateways (die auf ausgeführt werden AWS IoT Greengrass).
- Ermöglicht AWS IoT SiteWise die Protokollierung.
- Ermöglicht AWS IoT SiteWise die Ausführung einer Metadaten-Suchabfrage in der AWS IoT TwinMaker Datenbank.

[Wenn Sie AWS IoT SiteWise mit einem einzelnen Benutzerkonto arbeiten, erstellt die `AWSServiceRoleForIoTSiteWise` Rolle die `AWSServiceRoleForIoTSiteWise` Richtlinie in Ihrem IAM-Konto und fügt sie den `AWSServiceRoleForIoTSiteWise` Service-linked Rollen für hinzu. \[AWS IoT SiteWise\]\(#\)](#)

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {

```

```

    "Sid": "AllowSiteWiseReadGreenGrass",
    "Effect": "Allow",
    "Action": [
        "greengrass:GetAssociatedRole",
        "greengrass:GetCoreDefinition",
        "greengrass:GetCoreDefinitionVersion",
        "greengrass:GetGroup",
        "greengrass:GetGroupVersion"
    ],
    "Resource": "*"
},
{
    "Sid": "AllowSiteWiseAccessLogGroup",
    "Effect": "Allow",
    "Action": [
        "logs:CreateLogGroup",
        "logs:DescribeLogGroups"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*"
},
{
    "Sid": "AllowSiteWiseAccessLog",
    "Effect": "Allow",
    "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*:log-stream:*"
},
{
    "Sid": "AllowSiteWiseAccessSiteWiseManagedWorkspaceInTwinMaker",
    "Effect": "Allow",
    "Action": [
        "iottwinmaker:GetWorkspace",
        "iottwinmaker:ExecuteQuery"
    ],
    "Resource": "arn:aws:iottwinmaker:*:*:workspace/*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "iottwinmaker:linkedServices": [
                "IOTSITewise"
            ]
        }
    }
}

```

```

    }
  }
]
}

```

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSiteWiseReadGreenGrass",
      "Effect": "Allow",
      "Action": [
        "greengrass:GetAssociatedRole",
        "greengrass:GetCoreDefinition",
        "greengrass:GetCoreDefinitionVersion",
        "greengrass:GetGroup",
        "greengrass:GetGroupVersion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowSiteWiseAccessLogGroup",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:DescribeLogGroups"
      ],
      "Resource": "arn:aws-us-gov:logs:*:*:log-group:/aws/iotsitewise*"
    },
    {
      "Sid": "AllowSiteWiseAccessLog",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws-us-gov:logs:*:*:log-group:/aws/iotsitewise*:log-stream:*"
    }
  ]
}

```

```

    },
    {
      "Sid": "AllowSiteWiseAccessSiteWiseManagedWorkspaceInTwinMaker",
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:GetWorkspace",
        "iottwinmaker:ExecuteQuery"
      ],
      "Resource": "arn:aws-us-gov:iottwinmaker:*:*:workspace/*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "iottwinmaker:linkedServices": [
            "IOTSITWISE"
          ]
        }
      }
    }
  ]
}

```

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSiteWiseReadGreenGrass",
      "Effect": "Allow",
      "Action": [
        "greengrass:GetAssociatedRole",
        "greengrass:GetCoreDefinition",
        "greengrass:GetCoreDefinitionVersion",
        "greengrass:GetGroup",
        "greengrass:GetGroupVersion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowSiteWiseAccessLogGroup",
      "Effect": "Allow",

```

```

    "Action": [
      "logs:CreateLogGroup",
      "logs:DescribeLogGroups"
    ],
    "Resource": "arn:aws-cn:logs:*:*:log-group:/aws/iotsitewise*"
  },
  {
    "Sid": "AllowSiteWiseAccessLog",
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogStream",
      "logs:DescribeLogStreams",
      "logs:PutLogEvents"
    ],
    "Resource": "arn:aws-cn:logs:*:*:log-group:/aws/iotsitewise*:log-stream:*"
  },
  {
    "Sid": "AllowSiteWiseAccessSiteWiseManagedWorkspaceInTwinMaker",
    "Effect": "Allow",
    "Action": [
      "iottwinmaker:GetWorkspace",
      "iottwinmaker:ExecuteQuery"
    ],
    "Resource": "arn:aws-cn:iottwinmaker:*:*:workspace/*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "iottwinmaker:linkedServices": [
          "IOTSITWISE"
        ]
      }
    }
  }
]
}

```

AWS IoT SiteWiseAktualisierungen fürAWSVerwaltete Richtlinien

Sie können sich Details zu Aktualisierungen AWS verwalteter Richtlinien anzeigen lassen, und zwar ab dem ZeitpunktAWS IoT SiteWise, zu dem dieser Dienst mit der Nachverfolgung der Änderungen begann. Abonnieren Sie den RSS-Feed auf der Seite AWS IoT SiteWise Dokumentenverlauf, um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten.

Änderungen	Beschreibung	Datum
AWSServiceRoleForIoTSiteWise – Aktualisierung auf eine bestehende Richtlinie	AWS IoT SiteWise kann jetzt eine Metadaten-Suchabfrage für die AWS IoT TwinMaker Datenbank ausführen.	6. November 2023
AWSIoTSiteWiseReadOnlyAccess – Aktualisierung auf eine bestehende Richtlinie	AWS IoT SiteWise hat ein neues Richtlinienpräfix hinzugefügt <code>BatchGet*</code> , das es Ihnen ermöglicht, Batch-Lesevorgänge durchzuführen.	16. September 2022
AWSIoTSiteWiseReadOnlyAccess – Neue Richtlinie	AWS IoT SiteWise hat eine neue Richtlinie hinzugefügt, auf die nur Lesezugriff gewährt werden kann. AWS IoT SiteWise	24. November 2021
AWS IoT SiteWise hat begonnen, Änderungen zu verfolgen	AWS IoT SiteWise hat begonnen, Änderungen für die AWS verwalteten Richtlinien zu verfolgen.	24. November 2021

Verwenden Sie serviceverknüpfte Rollen für AWS IoT SiteWise

AWS IoT SiteWise verwendet [dienstverknüpfte](#) Rollen AWS Identity and Access Management (IAM). Eine dienstverknüpfte Rolle ist ein einzigartiger Typ von IAM-Rolle, mit der direkt verknüpft ist AWS IoT SiteWise. Mit dem Dienst verknüpfte Rollen sind vordefiniert AWS IoT SiteWise und enthalten alle Berechtigungen, die der Dienst benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen.

Service-linked Rollen vereinfachen die Konfiguration von, AWS IoT SiteWise indem sie automatisch alle erforderlichen Berechtigungen enthalten. AWS IoT SiteWise definiert die Berechtigungen seiner dienstbezogenen Rollen und AWS IoT SiteWise kann, sofern nicht anders definiert, nur seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Und diese Berechtigungsrichtlinie kann keiner anderen IAM-Entität zugeordnet werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dadurch werden Ihre AWS IoT SiteWise Ressourcen geschützt, da Sie die Zugriffsberechtigung für die Ressourcen nicht versehentlich entfernen können.

Informationen zu anderen Diensten, die dienstbezogene Rollen unterstützen, finden Sie unter [AWS Dienste, die mit IAM funktionieren](#). Suchen Sie in der Spalte Rolle nach den Diensten, für die Ja steht. Service-linked Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

Themen

- [Service-linked Rollenberechtigungen für AWS IoT SiteWise](#)
- [Erstellen Sie eine serviceverknüpfte Rolle für AWS IoT SiteWise](#)
- [Aktualisieren Sie eine serviceverknüpfte Rolle für AWS IoT SiteWise](#)
- [Löschen Sie eine dienstverknüpfte Rolle für AWS IoT SiteWise](#)
- [Unterstützte Regionen für AWS IoT SiteWise Serviceverknüpfte Rollen](#)
- [Verwenden Sie Servicerollen für AWS IoT SiteWise Monitor](#)

Service-linked Rollenberechtigungen für AWS IoT SiteWise

AWS IoT SiteWise verwendet die benannte `AWSServiceRoleForIoTSiteWise` dienstbezogene Rolle. AWS IoT SiteWise verwendet diese dienstverknüpfte Rolle, um SiteWise Edge-Gateways (die auf laufen AWS IoT Greengrass) bereitzustellen und die Protokollierung durchzuführen.

Die `AWSServiceRoleForIoTSiteWise` dienstverknüpfte Rolle verwendet die `AWSServiceRoleForIoTSiteWise` Richtlinie mit den folgenden Berechtigungen. Diese Richtlinie:

- Ermöglicht AWS IoT SiteWise die Bereitstellung von SiteWise Edge-Gateways (die auf ausgeführt werden AWS IoT Greengrass).
- Ermöglicht AWS IoT SiteWise die Protokollierung.
- Ermöglicht AWS IoT SiteWise die Ausführung einer Metadaten-Suchabfrage in der AWS IoT TwinMaker Datenbank.

Weitere Informationen zu den zulässigen Aktionen in finden Sie

`AWSServiceRoleForIoTSiteWise` unter [AWS Verwaltete Richtlinien für AWS IoT SiteWise](#).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSiteWiseReadGreenGrass",
      "Effect": "Allow",
      "Action": [
        "greengrass:GetAssociatedRole",
        "greengrass:GetCoreDefinition",
        "greengrass:GetCoreDefinitionVersion",
        "greengrass:GetGroup",
        "greengrass:GetGroupVersion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowSiteWiseAccessLogGroup",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:DescribeLogGroups"
      ],
      "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*"
    },
    {
      "Sid": "AllowSiteWiseAccessLog",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*:log-stream:*"
    },
    {
      "Sid": "AllowSiteWiseAccessSiteWiseManagedWorkspaceInTwinMaker",
      "Effect": "Allow",
      "Action": [
        "iottwinmaker:GetWorkspace",
        "iottwinmaker:ExecuteQuery"
      ],
    },
  ]
}
```

```

    "Resource": "arn:aws:iottwinmaker:*:*:workspace/*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "iottwinmaker:linkedServices": [
          "IOTSITWISE"
        ]
      }
    }
  ]
}

```

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSiteWiseReadGreenGrass",
      "Effect": "Allow",
      "Action": [
        "greengrass:GetAssociatedRole",
        "greengrass:GetCoreDefinition",
        "greengrass:GetCoreDefinitionVersion",
        "greengrass:GetGroup",
        "greengrass:GetGroupVersion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowSiteWiseAccessLogGroup",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:DescribeLogGroups"
      ],
      "Resource": "arn:aws-us-gov:logs:*:*:log-group:/aws/iotsitewise*"
    },
    {
      "Sid": "AllowSiteWiseAccessLog",

```

```

    "Effect": "Allow",
    "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents"
    ],
    "Resource": "arn:aws-us-gov:logs:*:*:log-group:/aws/iotsitewise*:log-stream:*"
},
{
    "Sid": "AllowSiteWiseAccessSiteWiseManagedWorkspaceInTwinMaker",
    "Effect": "Allow",
    "Action": [
        "iottwinmaker:GetWorkspace",
        "iottwinmaker:ExecuteQuery"
    ],
    "Resource": "arn:aws-us-gov:iottwinmaker:*:*:workspace/*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "iottwinmaker:linkedServices": [
                "IOTSITWISE"
            ]
        }
    }
}
]
}

```

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSiteWiseReadGreenGrass",
            "Effect": "Allow",
            "Action": [
                "greengrass:GetAssociatedRole",
                "greengrass:GetCoreDefinition",
                "greengrass:GetCoreDefinitionVersion",
                "greengrass:GetGroup",
            ]
        }
    ]
}

```

```

    "greengrass:GetGroupVersion"
  ],
  "Resource": "*"
},
{
  "Sid": "AllowSiteWiseAccessLogGroup",
  "Effect": "Allow",
  "Action": [
    "logs:CreateLogGroup",
    "logs:DescribeLogGroups"
  ],
  "Resource": "arn:aws-cn:logs:*:*:log-group:/aws/iotsitewise*"
},
{
  "Sid": "AllowSiteWiseAccessLog",
  "Effect": "Allow",
  "Action": [
    "logs:CreateLogStream",
    "logs:DescribeLogStreams",
    "logs:PutLogEvents"
  ],
  "Resource": "arn:aws-cn:logs:*:*:log-group:/aws/iotsitewise*:log-stream:*"
},
{
  "Sid": "AllowSiteWiseAccessSiteWiseManagedWorkspaceInTwinMaker",
  "Effect": "Allow",
  "Action": [
    "iottwinmaker:GetWorkspace",
    "iottwinmaker:ExecuteQuery"
  ],
  "Resource": "arn:aws-cn:iottwinmaker:*:*:workspace/*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "iottwinmaker:linkedServices": [
        "IOTSITewise"
      ]
    }
  }
}
]
}

```

Sie können die Protokolle verwenden, um Ihre SiteWise Edge-Gateways zu überwachen und Fehler zu beheben. Weitere Informationen finden Sie unter [SiteWise Edge-Gateway-Protokolle überwachen](#).

Damit eine IAM-Entität (z. B. ein Benutzer, eine Gruppe oder eine Rolle) eine dienstbezogene Rolle erstellen, bearbeiten oder löschen kann, müssen Sie zunächst die Berechtigungen konfigurieren. Weitere Informationen finden Sie unter [Service-linked Rollenberechtigungen](#) im IAM-Benutzerhandbuch.

Erstellen Sie eine serviceverknüpfte Rolle für AWS IoT SiteWise

AWS IoT SiteWise erfordert eine dienstbezogene Rolle, um in Ihrem Namen bestimmte Aktionen auszuführen und auf Ressourcen zuzugreifen. Eine dienstverknüpfte Rolle ist eine einzigartige Art von AWS Identity and Access Management (IAM) -Rolle, mit der direkt verknüpft ist.

AWS IoT SiteWise Durch die Erstellung dieser Rolle gewähren Sie AWS IoT SiteWise die erforderlichen Berechtigungen für den Zugriff auf andere AWS Dienste und Ressourcen, die für ihren Betrieb erforderlich sind, z. B. Amazon S3 für die Datenspeicherung oder AWS IoT für die Gerätekommunikation.

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie die folgenden Operationen in der AWS IoT SiteWise Konsole ausführen, AWS IoT SiteWise wird die dienstbezogene Rolle für Sie erstellt.

- Erstellen Sie ein Greengrass V1-Gateway.
- Konfigurieren Sie die Protokollierungsoption.
- Wählen Sie die Opt-in-Schaltfläche im Banner „Abfrage ausführen“.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie einen Vorgang in der AWS IoT SiteWise Konsole ausführen, AWS IoT SiteWise wird die mit dem Dienst verknüpfte Rolle erneut für Sie erstellt.

Sie können auch die IAM-Konsole oder API verwenden, um eine serviceverknüpfte Rolle für zu erstellen. AWS IoT SiteWise

- Erstellen Sie dazu in der IAM-Konsole eine Rolle mit der `AWSServiceRoleForIoTSiteWise` Richtlinie und einer Vertrauensbeziehung mit `iotsitewise.amazonaws.com`

- Erstellen Sie dazu mithilfe der AWS CLI oder der IAM-API eine Rolle mit der `arn:aws:iam::aws:policy/aws-service-role/AWSServiceRoleForIoTSiteWise` Richtlinie und eine Vertrauensbeziehung mit `iotsitewise.amazonaws.com`

Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter [Erstellen einer serviceverknüpften Rolle](#).

Wenn Sie diese serviceverknüpfte Rolle löschen, können Sie mit demselben Verfahren die Rolle erneut erstellen.

Aktualisieren Sie eine serviceverknüpfte Rolle fürAWS IoT SiteWise

AWS IoT SiteWiseerlaubt es Ihnen nicht, die mit dem `AWSServiceRoleForIoTSiteWise` Dienst verknüpfte Rolle zu bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollename nach der Erstellung einer serviceverknüpften Rolle nicht bearbeitet werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter [Aktualisieren einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Löschen Sie eine dienstverknüpfte Rolle fürAWS IoT SiteWise

Wenn eine Funktion oder ein Dienst, für den eine serviceverknüpfte Rolle erforderlich ist, nicht mehr verwendet wird, empfiehlt es sich, die zugehörige Rolle zu löschen. Auf diese Weise soll vermieden werden, dass eine inaktive Entität nicht überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpfte Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

Note

Wenn der AWS IoT SiteWise Dienst die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies der Fall ist, warten Sie einige Minuten und versuchen Sie es erneut.

Um zu löschenAWS IoT SiteWiseRessourcen, die verwendet werden von `AWSServiceRoleForIoTSiteWise`

1. Deaktivieren Sie die Protokollierung fürAWS IoT SiteWise. Weitere Informationen finden Sie unter [Ändern Sie Ihre Protokollierungsstufe](#).

2. Löschen Sie alle aktiven SiteWise Edge-Gateways.

So löschen Sie die serviceverknüpfte Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die AWS API/AWS CLI, um die `AWSServiceRoleForIoTSiteWise` serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie im [IAM-Benutzerhandbuch unter Löschen von Rollen oder Instanzprofilen](#).

Unterstützte Regionen für AWS IoT SiteWise serviceverknüpfte Rollen

AWS IoT SiteWise unterstützt die Verwendung von dienstbezogenen Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter [AWS IoT SiteWise-Endpunkte und -Kontingente](#).

Verwenden Sie Servicerollen für AWS IoT SiteWise Monitor

Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.

Damit Verbundbenutzer des SiteWise Monitor-Portals auf Ihre AWS IoT SiteWise AWS IAM Identity Center Ressourcen zugreifen können, müssen Sie jedem Portal, das Sie erstellen, eine Servicerolle zuordnen. In der Servicerolle muss SiteWise Monitor als vertrauenswürdige Entität angegeben und die [AWSIoTSiteWiseMonitorPortalAccess](#) verwaltete Richtlinie enthalten oder [entsprechende Berechtigungen](#) definiert werden. Diese Richtlinie wird von den Berechtigungen verwaltet AWS und definiert die Berechtigungen, die SiteWise Monitor für den Zugriff auf Ihre AWS IoT SiteWise und IAM Identity Center-Ressourcen verwendet.

Wenn Sie ein SiteWise Monitor-Portal erstellen, müssen Sie eine Rolle auswählen, die Benutzern dieses Portals den Zugriff auf Ihre Ressourcen AWS IoT SiteWise und die Ressourcen von IAM Identity Center ermöglicht. Die AWS IoT SiteWise Konsole kann die Rolle für Sie erstellen und konfigurieren. Sie können die Rolle später in IAM bearbeiten. Ihre Portalbenutzer werden Probleme bei der Verwendung ihrer SiteWise Monitor-Portale haben, wenn Sie die erforderlichen Berechtigungen aus der Rolle entfernen oder die Rolle löschen.

Note

Portale, die vor dem 29. April 2020 erstellt wurden, haben keine Servicerollen benötigt. Wenn Sie vor diesem Datum Portale erstellt haben, müssen Sie diesen Servicerollen anfügen, um

sie weiter verwenden zu können. Navigieren Sie dazu in der [AWS IoT SiteWiseKonsole](#) zur Seite Portale und wählen Sie dann Alle Portale migrieren, um IAM-Rollen zu verwenden.

In den folgenden Abschnitten wird beschrieben, wie Sie die SiteWise Monitor-Servicerolle in der AWS-Managementkonsole oder der AWS Command Line Interface erstellen und verwalten.

Inhalt

- [Berechtigungen für Servicerollen für SiteWise Monitor \(Classic\)](#)
- [Berechtigungen für Servicerollen für SiteWise Monitor \(AI-aware\)](#)
- [Die SiteWise Monitor-Servicerolle verwalten \(Konsole\)](#)
 - [Finden Sie die Servicerolle \(Konsole\) eines Portals](#)
 - [Erstellen Sie eine SiteWise Monitor-Servicerolle \(AWS IoT SiteWiseKonsole\)](#)
 - [Erstellen Sie eine SiteWise Monitor-Servicerolle \(IAM-Konsole\)](#)
 - [Ändern Sie die Servicerolle eines Portals \(Konsole\)](#)
- [Die SiteWise Monitor-Servicerolle \(CLI\) verwalten](#)
 - [Finden Sie die Servicerolle \(CLI\) eines Portals](#)
 - [Erstellen Sie die SiteWise Monitor-Servicerolle \(CLI\)](#)
- [SiteWise Überwachen Sie Aktualisierungen für AWSIoTSiteWiseMonitorServiceRole](#)

Berechtigungen für Servicerollen für SiteWise Monitor (Classic)

Wenn Sie ein Portal erstellen, AWS IoT SiteWise können Sie damit eine Rolle erstellen, deren Name mit beginnt `AWSIoTSiteWiseMonitorServiceRole`. Diese Rolle ermöglicht Benutzern von Federated SiteWise Monitor den Zugriff auf Ihre Portalkonfiguration, Ihre Assets, Asset-Daten sowie die IAM Identity Center-Konfiguration .

Die Rolle vertraut darauf, dass der folgende Service diese Rolle annimmt:

- `monitor.iotsitewise.amazonaws.com`

Die Rolle verwendet die folgende Berechtigungsrichtlinie, die mit beginnt `AWSIoTSiteWiseMonitorServicePortalPolicy`, damit SiteWise Monitor-Benutzer Aktionen an Ressourcen in Ihrem Konto ausführen können. Die von [AWSIoTSiteWiseMonitorPortalAccess](#) verwaltete Richtlinie definiert gleichwertige Berechtigungen.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal",
        "iotsitewise:CreateProject",
        "iotsitewise:DescribeProject",
        "iotsitewise:UpdateProject",
        "iotsitewise>DeleteProject",
        "iotsitewise:ListProjects",
        "iotsitewise:BatchAssociateProjectAssets",
        "iotsitewise:BatchDisassociateProjectAssets",
        "iotsitewise:ListProjectAssets",
        "iotsitewise:CreateDashboard",
        "iotsitewise:DescribeDashboard",
        "iotsitewise:UpdateDashboard",
        "iotsitewise>DeleteDashboard",
        "iotsitewise:ListDashboards",
        "iotsitewise:CreateAccessPolicy",
        "iotsitewise:DescribeAccessPolicy",
        "iotsitewise:UpdateAccessPolicy",
        "iotsitewise>DeleteAccessPolicy",
        "iotsitewise:ListAccessPolicies",
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssets",
        "iotsitewise:ListAssociatedAssets",
        "iotsitewise:DescribeAssetProperty",
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetAssetPropertyValueHistory",
        "iotsitewise:GetAssetPropertyAggregates",
        "iotsitewise:BatchPutAssetPropertyValue",
        "iotsitewise:ListAssetRelationships",
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:ListAssetModels",
        "iotsitewise:UpdateAssetModel",
        "iotsitewise:UpdateAssetModelPropertyRouting",
        "sso-directory:DescribeUsers",
        "sso-directory:DescribeUser",
        "iotevents:DescribeAlarmModel",

```

```

        "iotevents:ListTagsForResource"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "iotevents:BatchAcknowledgeAlarm",
        "iotevents:BatchSnoozeAlarm",
        "iotevents:BatchEnableAlarm",
        "iotevents:BatchDisableAlarm"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "iotevents:keyValue": "false"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "iotevents>CreateAlarmModel",
        "iotevents:TagResource"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "aws:RequestTag/iotsitewisemonitor": "false"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "iotevents:UpdateAlarmModel",
        "iotevents>DeleteAlarmModel"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "aws:ResourceTag/iotsitewisemonitor": "false"
        }
    }
}

```

```

    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "iotevents.amazonaws.com"
          ]
        }
      }
    }
  ]
}

```

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:CreateProject",
        "iotsitewise:DescribeProject",
        "iotsitewise:UpdateProject",
        "iotsitewise>DeleteProject",
        "iotsitewise:ListProjects",
        "iotsitewise:BatchAssociateProjectAssets",
        "iotsitewise:BatchDisassociateProjectAssets",
        "iotsitewise:ListProjectAssets",
        "iotsitewise:CreateDashboard",
        "iotsitewise:DescribeDashboard",
        "iotsitewise:UpdateDashboard",
        "iotsitewise>DeleteDashboard",
        "iotsitewise:ListDashboards",
        "iotsitewise:CreateAccessPolicy",
        "iotsitewise:DescribeAccessPolicy",
        "iotsitewise:UpdateAccessPolicy",

```

```

        "iotsitewise:DeleteAccessPolicy",
        "iotsitewise:ListAccessPolicies",
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssets",
        "iotsitewise:ListAssociatedAssets",
        "iotsitewise:DescribeAssetProperty",
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetAssetPropertyValueHistory",
        "iotsitewise:GetAssetPropertyAggregates"
    ],
    "Resource": "*"
}
]
}

```

Weitere Informationen zu den erforderlichen Berechtigungen für Alarmer finden Sie unter [Richten Sie Berechtigungen für Ereignisalarmer ein in AWS IoT SiteWise](#).

Wenn sich ein Portalbenutzer anmeldet, erstellt SiteWise Monitor eine [Sitzungsrichtlinie](#), die auf der Schnittmenge zwischen der Servicerolle und den Zugriffsrichtlinien dieses Benutzers basiert. Zugriffsrichtlinien definieren die Zugriffsstufe von -Identitäten auf Ihre Portale und Projekte. Weitere Informationen zu Portalberechtigungen und Zugriffsrichtlinien finden Sie unter [Verwalten Sie Ihre SiteWise Monitor-Portale](#) und [CreateAccessPolicy](#).

Berechtigungen für Servicerollen für SiteWise Monitor (AI-aware)

Wenn Sie ein Portal erstellen, AWS IoT SiteWise können Sie damit eine Rolle erstellen, deren Name mit `IoTSiteWisePortalRole` beginnt. Diese Rolle ermöglicht Benutzern von Federated SiteWise Monitor den Zugriff auf Ihre Portalkonfiguration, Ihre Assets, Asset-Daten sowie die IAM Identity Center-Konfiguration.

Warning

Die Rollen Projekteinhaber und Projektbetrachter werden für SiteWise Monitor () AI-aware nicht unterstützt.

Die Rolle vertraut darauf, dass der folgende Service diese Rolle annimmt:

- `monitor.iotsitewise.amazonaws.com`

Die Rolle verwendet die folgende Berechtigungsrichtlinie, die mit beginnt `IoTSiteWiseAPIPortalAccessPolicy`, damit SiteWise Monitor-Benutzer Aktionen an Ressourcen in Ihrem Konto ausführen können.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:CreateProject",
        "iotsitewise:DescribePortal",
        "iotsitewise:ListProjects",
        "iotsitewise:DescribeProject",
        "iotsitewise:UpdateProject",
        "iotsitewise>DeleteProject",
        "iotsitewise:CreateDashboard",
        "iotsitewise:DescribeDashboard",
        "iotsitewise:UpdateDashboard",
        "iotsitewise>DeleteDashboard",
        "iotsitewise:ListDashboards",
        "iotsitewise:ListAssets",
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssociatedAssets",
        "iotsitewise:ListAssetProperties",
        "iotsitewise:DescribeAssetProperty",
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetAssetPropertyValueHistory",
        "iotsitewise:GetAssetPropertyAggregates",
        "iotsitewise:GetInterpolatedAssetPropertyValues",
        "iotsitewise:BatchGetAssetPropertyAggregates",
        "iotsitewise:BatchGetAssetPropertyValue",
        "iotsitewise:BatchGetAssetPropertyValueHistory",
        "iotsitewise:ListAssetRelationships",
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:ListAssetModels",
        "iotsitewise:DescribeAssetCompositeModel",
        "iotsitewise:DescribeAssetModelCompositeModel",
        "iotsitewise:ListAssetModelProperties",
        "iotsitewise:ExecuteQuery",

```

```

        "iotsitewise:ListTimeSeries",
        "iotsitewise:DescribeTimeSeries",
        "iotsitewise:InvokeAssistant",
        "iotsitewise:DescribeDataset",
        "iotsitewise:ListDatasets",
        "iotevents:DescribeAlarmModel",
        "iotevents:ListTagsForResource",
        "iottwinmaker:ListWorkspaces",
        "iottwinmaker:ExecuteQuery",
        "iottwinmaker:GetWorkspace",
        "identitystore:DescribeUser"
    ],
    "Resource": "*"
}
]
}

```

Wenn sich ein Portalbenutzer anmeldet, erstellt SiteWise Monitor eine [Sitzungsrichtlinie](#), die auf der Schnittmenge zwischen der Servicerolle und den Zugriffsrichtlinien dieses Benutzers basiert.

Die SiteWise Monitor-Servicerolle verwalten (Konsole)

Das AWS-IoT-SiteWise-Konsole erleichtert die Verwaltung der SiteWise Monitor-Dienstrolle für Portale. Beim Erstellen eines Portals sucht die Konsole nach vorhandenen Rollen, die für eine Zuordnung geeignet sind. Wenn keine verfügbar sind, kann die Konsole eine Servicerolle für Sie erstellen und konfigurieren. Weitere Informationen finden Sie unter [Erstellen Sie ein Portal in SiteWise Monitor](#).

Themen

- [Finden Sie die Servicerolle \(Konsole\) eines Portals](#)
- [Erstellen Sie eine SiteWise Monitor-Servicerolle \(AWS IoT SiteWiseKonsole\)](#)
- [Erstellen Sie eine SiteWise Monitor-Servicerolle \(IAM-Konsole\)](#)
- [Ändern Sie die Servicerolle eines Portals \(Konsole\)](#)

Finden Sie die Servicerolle (Konsole) eines Portals

Gehen Sie wie folgt vor, um die einem SiteWise Monitor-Portal zugeordnete Servicerolle zu finden.

So finden Sie die Servicerolle eines Portals

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im linken Navigationsbereich die Option Portale aus.
3. Wählen Sie das Portal aus, dessen Servicerolle Sie finden möchten.

Die dem Portal angefügte Rolle wird unter Permissions (Berechtigungen), Service role (Servicerolle) angezeigt.

Erstellen Sie eine SiteWise Monitor-Servicerolle (AWS IoT SiteWiseKonsole)

Wenn Sie ein SiteWise Monitor-Portal erstellen, können Sie eine Servicerolle für Ihr Portal erstellen. Weitere Informationen finden Sie unter [Erstellen Sie ein Portal in SiteWise Monitor](#).

Sie können auch eine Servicerolle für ein vorhandenes Portal in der AWS IoT SiteWise Konsole erstellen. Dies ersetzt die bestehende Servicerolle des Portals.

So erstellen Sie eine Servicerolle für ein vorhandenes Portal

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im linken Navigationsbereich die Option Portale aus.
3. Wählen Sie das Portal aus, für das Sie eine neue Servicerolle erstellen möchten.
4. Wählen Sie unter Portal details (Portaldetails) die Option Edit (Bearbeiten).
5. Wählen Sie unter Permissions (Berechtigungen) die Option Create and use a new service role (Eine neue Servicerolle erstellen und verwenden) aus der Liste aus.
6. Geben Sie einen Namen für die neue Rolle ein.
7. Wählen Sie Speichern.

Erstellen Sie eine SiteWise Monitor-Servicerolle (IAM-Konsole)

Sie können eine Servicerolle anhand der Servicerollenvorlage in der IAM-Konsole erstellen. Diese Rollenvorlage enthält die [AWSIoTSiteWiseMonitorPortalAccess](#) verwaltete Richtlinie und gibt SiteWise Monitor als vertrauenswürdige Entität an.

Um eine Servicerolle aus der Servicerollenvorlage des Portals zu erstellen

1. Navigieren Sie zur [IAM-Konsole](#).

2. Wählen Sie im Navigationsbereich Rollen aus.
3. Wählen Sie Create role (Rolle erstellen) aus.
4. Wählen Sie unter Wählen Sie einen Anwendungsfall die Option IoT aus SiteWise.
5. Wählen Sie unter Wählen Sie Ihren Anwendungsfall aus die Option IoT SiteWise Monitor - Portal.
6. Wählen Sie Next: Permissions aus.
7. Wählen Sie Next: Tags (Weiter: Tags) aus.
8. Klicken Sie auf Weiter: Prüfen.
9. Geben Sie einen Rollennamen für die neue Servicerolle ein.
10. Wählen Sie Rolle erstellen aus.

Ändern Sie die Servicerolle eines Portals (Konsole)

Gehen Sie wie folgt vor, um eine andere SiteWise Monitor-Servicerolle für ein Portal auszuwählen.

So ändern Sie die Servicerolle eines Portals

1. Navigieren Sie zur [AWS IoT SiteWise-Konsole](#).
2. Wählen Sie im linken Navigationsbereich die Option Portale aus.
3. Wählen Sie das Portal aus, dessen Servicerolle Sie ändern möchten.
4. Wählen Sie unter Portal details (Portaldetails) die Option Edit (Bearbeiten).
5. Wählen Sie unter Permissions (Berechtigungen) die Option Use an existing role (Vorhandene Rolle verwenden) aus.
6. Wählen Sie eine vorhandene Rolle aus, die diesem Portal angefügt werden soll.
7. Wählen Sie Speichern.

Die SiteWise Monitor-Servicerolle (CLI) verwalten

Sie können den AWS CLI für die folgenden Aufgaben zur Verwaltung der Portaldienstrollen verwenden:

Themen

- [Finden Sie die Servicerolle \(CLI\) eines Portals](#)
- [Erstellen Sie die SiteWise Monitor-Servicerolle \(CLI\)](#)

Finden Sie die Servicerolle (CLI) eines Portals

Um die einem SiteWise Monitor-Portal zugeordnete Servicerolle zu finden, führen Sie den folgenden Befehl aus, um alle Ihre Portale in der aktuellen Region aufzulisten.

```
aws iotsitewise list-portals
```

Die Operation gibt eine Antwort mit einer Portalzusammenfassung im folgenden Format zurück.

```
{
  "portalSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
      "name": "WindFarmPortal",
      "description": "A portal that contains wind farm projects for Example Corp.",
      "roleArn": "arn:aws:iam::123456789012:role/service-role/role-name",
      "startUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",
      "creationDate": "2020-02-04T23:01:52.90248068Z",
      "lastUpdateDate": "2020-02-04T23:01:52.90248078Z"
    }
  ]
}
```

Sie können den [DescribePortal](#)Vorgang auch verwenden, um die Rolle Ihres Portals zu ermitteln, wenn Sie die ID Ihres Portals kennen.

Erstellen Sie die SiteWise Monitor-Servicerolle (CLI)

Gehen Sie wie folgt vor, um eine neue SiteWise Monitor-Dienstrolle zu erstellen.

So erstellen Sie eine SiteWise Monitor-Dienstrolle

1. Erstellen Sie eine Rolle mit einer Vertrauensrichtlinie, die es SiteWise Monitor ermöglicht, die Rolle zu übernehmen. In diesem Beispiel wird eine Rolle mit dem Namen **MySiteWiseMonitorPortalRole** aus einer Vertrauensrichtlinie erstellt, die in einer JSON-Zeichenfolge gespeichert ist.

Linux, macOS, or Unix

```
aws iam create-role --role-name MySiteWiseMonitorPortalRole --assume-role-
policy-document '{
  "Version": "2012-10-17",
```

```

    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": "monitor.iotsitewise.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
      }
    ]
  }
}'

```

Windows command prompt

```

aws iam create-role --role-name MySiteWiseMonitorPortalRole --assume-role-
policy-document "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"Service\":\"monitor.iotsitewise.amazonaws.com\"},\"Action\":
\"sts:AssumeRole\"}]}"

```

2. Kopieren Sie den Rollen-ARN aus den Rollenmetadaten in der Ausgabe. Wenn Sie ein Portal erstellen, verwenden Sie diesen ARN, um die Rolle Ihrem Portal zuzuordnen. Weitere Informationen zum Erstellen eines Portals finden Sie [CreatePortal](#) in der AWS IoT SiteWiseAPI-Referenz.
3. a. Für den SiteWise Monitor (Classic) — Hängen Sie die `AWSIoTSiteWiseMonitorPortalAccess` Richtlinie an die Rolle an, oder fügen Sie eine Richtlinie hinzu, die entsprechende Berechtigungen definiert.

```

aws iam attach-role-policy --role-name MySiteWiseMonitorPortalRole --policy-arn
arn:aws:iam::aws:policy/service-role/AWSIoTSiteWiseMonitorPortalAccess

```

- b. Für den SiteWise Monitor (AI-aware) — Hängen Sie die `IoTSiteWiseAIPortalAccessPolicy` Richtlinie an die Rolle an, oder fügen Sie eine Richtlinie hinzu, die entsprechende Berechtigungen definiert. Erstellen Sie beispielsweise eine Richtlinie mit Portalzugriffsberechtigungen. Im folgenden Beispiel wird eine Richtlinie mit dem Namen `MySiteWiseMonitorPortalAccess` erstellt.

```

aws iam create-policy \
  --policy-name MySiteWiseMonitorPortalAccess \
  --policy-document '{
    "Version": "2012-10-17",
    "Statement": [

```

```

{
  "Effect": "Allow",
  "Action": [
    "iotsitewise:CreateProject",
    "iotsitewise:DescribePortal",
    "iotsitewise:ListProjects",
    "iotsitewise:DescribeProject",
    "iotsitewise:UpdateProject",
    "iotsitewise>DeleteProject",
    "iotsitewise:CreateDashboard",
    "iotsitewise:DescribeDashboard",
    "iotsitewise:UpdateDashboard",
    "iotsitewise>DeleteDashboard",
    "iotsitewise:ListDashboards",
    "iotsitewise:ListAssets",
    "iotsitewise:DescribeAsset",
    "iotsitewise:ListAssociatedAssets",
    "iotsitewise:ListAssetProperties",
    "iotsitewise:DescribeAssetProperty",
    "iotsitewise:GetAssetPropertyValue",
    "iotsitewise:GetAssetPropertyValueHistory",
    "iotsitewise:GetAssetPropertyAggregates",
    "iotsitewise:GetInterpolatedAssetPropertyValues",
    "iotsitewise:BatchGetAssetPropertyAggregates",
    "iotsitewise:BatchGetAssetPropertyValue",
    "iotsitewise:BatchGetAssetPropertyValueHistory",
    "iotsitewise:ListAssetRelationships",
    "iotsitewise:DescribeAssetModel",
    "iotsitewise:ListAssetModels",
    "iotsitewise:DescribeAssetCompositeModel",
    "iotsitewise:DescribeAssetModelCompositeModel",
    "iotsitewise:ListAssetModelProperties",
    "iotsitewise:ExecuteQuery",
    "iotsitewise:ListTimeSeries",
    "iotsitewise:DescribeTimeSeries",
    "iotsitewise:InvokeAssistant",
    "iotsitewise:DescribeDataset",
    "iotsitewise:ListDatasets",
    "iotevents:DescribeAlarmModel",
    "iotevents:ListTagsForResource",
    "iottwinmaker:ListWorkspaces",
    "iottwinmaker:ExecuteQuery",
    "iottwinmaker:GetWorkspace",
    "identitystore:DescribeUser"
  ]
}

```

```

    ],
    "Resource": "*"
  }
]
}'

```

So fügen Sie einem vorhandenen Portal eine Servicerolle an

1. Führen Sie den folgenden Befehl aus, um die vorhandenen Details des Portals abzurufen.
portal-id Ersetzen Sie durch die ID des Portals.

```
aws iotsitewise describe-portal --portal-id portal-id
```

Die Operation gibt eine Antwort zurück, die die Details des Portals im folgenden Format enthält.

```

{
  "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalArn": "arn:aws:iotsitewise:region:account-id:portal/a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalName": "WindFarmPortal",
  "portalDescription": "A portal that contains wind farm projects for Example Corp.",
  "portalClientId": "E-1a2b3c4d5e6f_sn6tbqHVzLWVEXAMPLE",
  "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",
  "portalContactEmail": "support@example.com",
  "portalStatus": {
    "state": "ACTIVE"
  },
  "portalCreationDate": "2020-04-29T23:01:52.90248068Z",
  "portalLastUpdateDate": "2020-04-29T00:28:26.103548287Z",
  "roleArn": "arn:aws:iam::123456789012:role/service-role/AWSIoTSiteWiseMonitorServiceRole_1aEXAMPLE"
}

```

2. Führen Sie den folgenden Befehl aus, um einem Portal eine Servicerolle anzufügen. *role-arn* Ersetzen Sie durch die Servicerolle ARN und ersetzen Sie die verbleibenden Parameter durch die vorhandenen Werte des Portals.

```
aws iotsitewise update-portal \
```

```
--portal-id portal-id \
--role-arn role-arn \
--portal-name portal-name \
--portal-description portal-description \
--portal-contact-email portal-contact-email
```

SiteWise Überwachen Sie Aktualisierungen für AWSIoTSiteWiseMonitorServiceRole

Sie können sich Details zu Updates AWSIoTSiteWiseMonitorServiceRole für SiteWise Monitor anzeigen lassen, und zwar ab dem Zeitpunkt, zu dem dieser Dienst mit der Nachverfolgung der Änderungen begann. Abonnieren Sie den RSS-Feed auf der Seite AWS IoT SiteWise Dokumentenverlauf, um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten.

Änderungen	Beschreibung	Date
AWSIoTSiteWiseMonitorPortalAccess – Richtlinie aktualisiert	AWS IoT SiteWise AWSIoTSiteWiseMonitorPortalAccess hat die verwaltete Richtlinie für die Alarmfunktion aktualisiert.	27. Mai 2021
AWS IoT SiteWise hat begonnen, Änderungen zu verfolgen	AWS IoT SiteWise hat begonnen, Änderungen für seine Servicerolle zu verfolgen.	15. Dezember 2020

Richten Sie Berechtigungen für Ereignisalarme ein in AWS IoT SiteWise

Wenn Sie ein AWS IoT Events Alarmmodell zur Überwachung einer AWS IoT SiteWise Anlageneigenschaft verwenden, benötigen Sie die folgenden IAM-Berechtigungen:

- Eine AWS IoT Events Servicerolle, AWS IoT Events an die Daten gesendet werden können. AWS IoT SiteWise Weitere Informationen finden Sie unter [Identitäts- und Zugriffsmanagement für AWS IoT Events](#) im AWS IoT Events Entwicklerhandbuch.
- Sie benötigen die folgenden AWS IoT SiteWise Aktionsberechtigungen:
`ioticsitewise:DescribeAssetModel`

`undiotssitewise:UpdateAssetModelPropertyRouting`. Diese Berechtigungen ermöglichen AWS IoT SiteWise das Senden von Objekteigenschaftswerten an AWS IoT Events Alarmmodelle.

Weitere Informationen finden Sie unter [Resource-based Richtlinien](#) im IAM-Benutzerhandbuch.

Erforderliche Aktionsberechtigungen

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann. Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können.

Bevor Sie ein AWS IoT Events Alarmmodell definieren, müssen Sie die folgenden Berechtigungen erteilen, die es ermöglichen, Asset-Eigenschaftswerte AWS IoT SiteWise an das Alarmmodell zu senden.

- `iotsitewise:DescribeAssetModel`, `iotsitewise:ListAssetModels` — Ermöglicht AWS IoT Events die Überprüfung, ob eine Anlageneigenschaft existiert.
- `iotsitewise:UpdateAssetModelPropertyRouting` — Ermöglicht AWS IoT SiteWise das automatische Erstellen von Abonnements, AWS IoT SiteWise an die Daten gesendet werden können AWS IoT Events.

Weitere Informationen zu AWS IoT SiteWise unterstützten Aktionen finden Sie unter [Aktionen definiert von AWS IoT SiteWise](#) in der Service Authorization Reference.

Example Beispiel Berechtigungsrichtlinie 1

Die folgende Richtlinie ermöglicht AWS IoT SiteWise das Senden von Objekteigenschaftswerten an beliebige AWS IoT Events Alarmmodelle.

JSON

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

```

        "iotevents:UpdateAlarmModel"
    ],
    "Resource": "arn:aws:iotevents:us-east-1:123456789012:alarmModel/*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iotsitewise:DescribeAssetModel",
      "iotsitewise:ListAssetModels",
      "iotsitewise:UpdateAssetModelPropertyRouting"
    ],
    "Resource": "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/*"
  }
]
}

```

Example Beispiel Berechtigungsrichtlinie 2

Die folgende Richtlinie ermöglicht AWS IoT SiteWise das Senden von Werten einer bestimmten Anlageneigenschaft an ein bestimmtes AWS IoT Events Alarmmodell.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotevents:CreateAlarmModel",
        "iotevents:UpdateAlarmModel"
      ],
      "Resource": "arn:aws:iotevents:us-east-1:123456789012:alarmModel/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:ListAssetModels"
      ],

```

```

    "Resource": "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/
  *
  },
  {
    "Effect": "Allow",
    "Action": [
      "iotsitewise:UpdateAssetModelPropertyRouting"
    ],
    "Resource": [
      "arn:aws:iotsitewise:us-east-1:123456789012:asset-
model/12345678-90ab-cdef-1234-567890abcdef"
    ],
    "Condition": {
      "StringLike": {
        "iotsitewise:propertyId": "abcdef12-3456-7890-abcd-
ef1234567890",
        "aws:ResourceTag/AlarmModel": "arn:aws:iotevents:us-
east-1:123456789012:alarmModel/MyAlarmModel"
      }
    }
  }
]
}

```

(Fakultativ) ListInputRoutings Erlaubnis

Wenn Sie ein Asset-Modell aktualisieren oder löschen, AWS IoT SiteWise kann überprüft werden, ob ein Alarmmodell eine mit diesem Asset-Modell verknüpfte Anlageneigenschaft überwacht. AWS IoT Events Dadurch wird verhindert, dass Sie eine Anlageneigenschaft löschen, die derzeit von einem AWS IoT Events Alarm verwendet wird. Um diese Funktion in zu aktivieren AWS IoT SiteWise, benötigen Sie die `iotevents:ListInputRoutings` entsprechende Genehmigung. Diese Berechtigung AWS IoT SiteWise ermöglicht Aufrufe des [ListInputRoutings](#) API-Vorgangs, der von unterstützt wird AWS IoT Events.

Note

Wir empfehlen dringend, dass Sie die `ListInputRoutings` Erlaubnis hinzufügen.

Example Beispiel für eine Berechtigungsrichtlinie

Die folgende Richtlinie ermöglicht es Ihnen, Asset-Modelle zu aktualisieren und zu löschen und die ListInputRoutings API in zu verwendenAWS IoT SiteWise.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:UpdateAssetModel",
        "iotsitewise:DeleteAssetModel",
        "iotevents:ListInputRoutings"
      ],
      "Resource": "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/*"
    }
  ]
}
```

Erforderliche Berechtigungen für SiteWise Monitor

Wenn Sie die Alarmfunktion in SiteWise Monitor-Portalen verwenden möchten, müssen Sie die [SiteWise Monitor-Servicerolle](#) mit der folgenden Richtlinie aktualisieren:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal",
        "iotsitewise:CreateProject",
        "iotsitewise:DescribeProject",
        "iotsitewise:UpdateProject",

```

```

        "iotsitewise:DeleteProject",
        "iotsitewise:ListProjects",
        "iotsitewise:BatchAssociateProjectAssets",
        "iotsitewise:BatchDisassociateProjectAssets",
        "iotsitewise:ListProjectAssets",
        "iotsitewise:CreateDashboard",
        "iotsitewise:DescribeDashboard",
        "iotsitewise:UpdateDashboard",
        "iotsitewise:DeleteDashboard",
        "iotsitewise:ListDashboards",
        "iotsitewise:CreateAccessPolicy",
        "iotsitewise:DescribeAccessPolicy",
        "iotsitewise:UpdateAccessPolicy",
        "iotsitewise:DeleteAccessPolicy",
        "iotsitewise:ListAccessPolicies",
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssets",
        "iotsitewise:ListAssociatedAssets",
        "iotsitewise:DescribeAssetProperty",
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetAssetPropertyValueHistory",
        "iotsitewise:GetAssetPropertyAggregates",
        "iotsitewise:BatchPutAssetPropertyValue",
        "iotsitewise:ListAssetRelationships",
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:ListAssetModels",
        "iotsitewise:UpdateAssetModel",
        "iotsitewise:UpdateAssetModelPropertyRouting",
        "sso-directory:DescribeUsers",
        "sso-directory:DescribeUser",
        "iotevents:DescribeAlarmModel",
        "iotevents:ListTagsForResource"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "iotevents:BatchAcknowledgeAlarm",
        "iotevents:BatchSnoozeAlarm",
        "iotevents:BatchEnableAlarm",
        "iotevents:BatchDisableAlarm"
    ],
    "Resource": "*",

```

```

        "Condition": {
            "Null": {
                "iotevents:keyValue": "false"
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iotevents:CreateAlarmModel",
                "iotevents:TagResource"
            ],
            "Resource": "*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/iotsitewisemonitor": "false"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iotevents:UpdateAlarmModel",
                "iotevents>DeleteAlarmModel"
            ],
            "Resource": "*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/iotsitewisemonitor": "false"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "iotevents.amazonaws.com"
                    ]
                }
            }
        }
    ]
}

```

```
}  
  }  
] }  
}
```

Cross-service verwirrter Stellvertreter, Prävention in AWS IoT SiteWise

Das Problem des verwirrten Stellvertreters ist ein Sicherheitsproblem, bei dem eine Entität, die keine Berechtigung zur Durchführung einer Aktion hat, eine privilegiere Entität zur Durchführung der Aktion zwingen kann. In AWS kann ein dienstübergreifendes Identitätswechsels zum Problem des verwirrten Stellvertreters führen. Cross-service Ein Identitätswechsel kann auftreten, wenn ein Dienst (der anrufende Dienst) einen anderen Dienst (den angerufenen Dienst) aufruft. Der Anruf-Service kann so manipuliert werden, dass er seine Berechtigungen verwendet, um auf die Ressourcen eines anderen Kunden zu reagieren, auf die er sonst nicht zugreifen dürfte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Wir empfehlen, die Kontextschlüssel [aws:SourceArn](#) und die [aws:SourceAccount](#) globalen Bedingungsschlüssel in Ressourcenrichtlinien zu verwenden, um die Berechtigungen einzuschränken, die AWS IoT SiteWise der Ressource einen anderen Dienst gewähren. Wenn der `aws:SourceArn`-Wert nicht die Konto-ID enthält, z. B. den Amazon-Ressourcennamen (ARN) eines Amazon-S3-Buckets, müssen Sie beide globale Bedingungskontext-Schlüssel verwenden, um Berechtigungen einzuschränken. Wenn Sie beide globale Bedingungskontextschlüssel verwenden und der `aws:SourceArn`-Wert die Konto-ID enthält, müssen der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert dieselbe Konto-ID verwenden, wenn sie in der gleichen Richtlinienanweisung verwendet wird.

- Verwenden Sie `aws:SourceArn`, wenn Sie nur eine Ressource mit dem betriebsübergreifenden Zugriff verknüpfen möchten.
- Verwenden Sie `aws:SourceAccount`, wenn Sie zulassen möchten, dass Ressourcen in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft werden.

Der Wert von `aws:SourceArn` muss die AWS IoT SiteWise Kundenressource sein, die der `sts:AssumeRole` Anfrage zugeordnet ist.

Der effektivste Weg, um sich vor dem Confused-Deputy-Problem zu schützen, ist die Verwendung des globalen Bedingungskontext-Schlüssels `aws:SourceArn` mit dem vollständigen

ARN der Ressource. Wenn Sie den vollständigen ARN der Ressource nicht kennen oder wenn Sie mehrere Ressourcen angeben, verwenden Sie den globalen Bedingungskontext-Schlüssel `aws:SourceArn` mit Platzhaltern (*) für die unbekannten Teile des ARN. Beispiel, `arn:aws:servicename:*:123456789012:*`.

Example— Verwirrter Stellvertreter, Prävention

Das folgende Beispiel zeigt, wie Sie die Kontexttasten `aws:SourceArn` und die `aws:SourceAccount` globale Bedingung verwenden können, AWS IoT SiteWise um das Problem des verwirrten Stellvertreters zu verhindern.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "iotsitewise.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:iotsitewise:*:123456789012:*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

FehlerbehebungAWS IoT SiteWiselIdentität und Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit AWS IoT SiteWise und AWS Identity and Access Management (IAM) auftreten können.

Themen

- [Ich bin nicht berechtigt, eine Aktion durchzuführen in AWS IoT SiteWise](#)
- [Ich bin nicht berechtigt, IAM auszuführen: PassRole](#)
- [Ich möchte Personen außerhalb meines AWS-Kontos, um auf mein zuzugreifen AWS IoT SiteWise-Ressourcen](#)

Ich bin nicht berechtigt, eine Aktion durchzuführen in AWS IoT SiteWise

Wenn Ihnen AWS-Managementkonsole mitgeteilt wird, dass Sie nicht berechtigt sind, eine Aktion durchzuführen, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator ist die Person, die Ihnen Ihren Benutzernamen und Ihr Passwort zur Verfügung gestellt hat.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson` IAM-Benutzer versucht, die Konsole zu verwenden, um Details zu einem Asset anzuzeigen, aber nicht über die `iot:DescribeAsset` entsprechenden Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
iot:DescribeAsset on resource: a1b2c3d4-5678-90ab-cdef-2222EXAMPLE
```

In diesem Fall bittet Mateo den Administrator, die Richtlinien zu aktualisieren, um ihm den Zugriff auf die Ressource mit der ID `a1b2c3d4-5678-90ab-cdef-2222EXAMPLE` über die Aktion `iot:DescribeAsset` zu ermöglichen.

Ich bin nicht berechtigt, **IAM** auszuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Durchführen der `iam:PassRole`-Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an AWS IoT SiteWise übergeben zu können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte 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 AWS IoT SiteWise 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 meinesAWSKonto, um auf mein zuzugreifenAWS IoT SiteWiseRessourcen

Sie können eine Rolle erstellen, mit der Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation auf Ihre Ressourcen zugreifen können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob diese Funktionen AWS IoT SiteWise unterstützt werden, finden Sie unter [WieAWS IoT SiteWisefunktioniert 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 IAM-Benutzer in einem anderenAWS-Konto, den Sie besitzen](#).
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren könnenAWS-Konten, finden Sie [AWS-Kontenim IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte](#).
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter [Gewähren von Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\)](#) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Konformitätsprüfung fürAWS IoT SiteWise

AWS IoT SiteWisefällt nicht in den Geltungsbereich AWS irgendwelcher Compliance-Programme.

Eine Liste der AWS Services im Rahmen bestimmter Compliance-Programme finden Sie unter [AWS Services im Umfang nach Compliance-Programmen AWS](#) . Allgemeine Informationen finden Sie unter [AWS Compliance-Programme AWS](#) .

Sie können Prüfberichte von Drittanbietern unter [herunterladen AWS Artifact](#) . Weitere Informationen finden Sie unter [Berichte herunterladen unter](#) .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS IoT SiteWise hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- Schnellstartanleitungen zu [Sicherheit und Compliance Schnellstartanleitungen](#) zu — In diesen Bereitstellungshandbüchern werden architektonische Überlegungen erörtert und Schritte für die Implementierung von sicherheits- und Compliance-orientierten Basisumgebungen beschrieben. AWS
- Whitepaper „[Architecting for HIPAA Security and Compliance](#)“ — In diesem Whitepaper wird beschrieben, wie Unternehmen Anwendungen entwickeln können. AWS HIPAA-compliant
- [AWS Compliance-Ressourcen](#) — Diese Sammlung von Arbeitsmappen und Leitfäden kann auf Ihre Branche und Ihren Standort zutreffen.
- [Bewertung von Ressourcen anhand von Regeln](#) im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- [AWS Security Hub CSPM](#) — Dieser AWS Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus, sodass Sie überprüfen können AWS, ob Sie die Sicherheitsstandards und Best Practices der Branche einhalten.
- [Zehn goldene Sicherheitsregeln für industrielle IoT-Lösungen](#) — In diesem Blogbeitrag werden zehn goldene Regeln vorgestellt, mit denen Sie Ihre industriellen Steuerungssysteme (ICS), das industrielle Internet der Dinge (IIoT) und Cloud-Umgebungen schützen können.
- [Bewährte Sicherheitspraktiken für OT-Systeme in der Fertigung](#) — In diesem Whitepaper werden bewährte Sicherheitsmethoden für die Entwicklung, Bereitstellung und Architektur dieser hybriden Fertigungs-Workloads vor Ort für die Cloud beschrieben. AWS

Resilienz in AWS IoT SiteWise

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones. AWS Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

AWS IoT SiteWise wird vollständig verwaltet und nutzt hochverfügbare und langlebige AWS Dienste wie Amazon S3 und Amazon EC2. Um die Verfügbarkeit im Falle einer Unterbrechung der Availability Zone sicherzustellen, AWS IoT SiteWise arbeitet in mehreren Verfügbarkeitszonen.

Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#).

Zusätzlich zur AWS globalen Infrastruktur AWS IoT SiteWise bietet es mehrere Funktionen, die Sie bei Ihren Anforderungen an Datenstabilität und Datensicherung unterstützen:

- Sie können Aktualisierungen von Eigenschaftswerten AWS IoT Core über MQTT-Nachrichten veröffentlichen und dann Regeln konfigurieren, um auf diese Daten zu reagieren. Mit dieser Funktion können Sie Daten in anderen AWS Diensten wie Amazon S3 und Amazon DynamoDB sichern. Weitere Informationen erhalten Sie unter [Interagiere mit anderen AWS Diensten](#) und [Exportieren Sie Daten mit Benachrichtigungen über Vermögenseigenschaften nach Amazon S3](#).
- Sie können die AWS IoT SiteWise Get * APIs verwenden, um historische Vermögensdaten abzurufen und zu sichern. Weitere Informationen finden Sie unter [Fragen Sie historische Werte von Vermögenswerten ab in AWS IoT SiteWise](#).
- Sie können die AWS IoT SiteWise Describe * APIs verwenden, um die Definitionen für Ihre Ressourcen wie Assets und Modelle abzurufen. Sie können diese Definitionen sichern und später verwenden, um Ihre Ressourcen neu zu erstellen. Weitere Informationen finden Sie in der [AWS IoT SiteWise-API-Referenz](#).

Sicherheit der Infrastruktur in AWS IoT SiteWise

Als verwalteter Dienst AWS IoT SiteWise ist er durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter

[AWSCloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe für den Zugriff AWS IoT SiteWise über das Netzwerk. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Cipher-Suites mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Diffie-Hellman Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

SiteWise Edge-Gateways, die auf der Cloud ausgeführt werden, verwenden Zertifikate und kryptografische Schlüssel, um sich mit der Cloud zu verbinden und zu authentifizieren AWS IoT Greengrass. X.509 AWS Weitere Informationen finden Sie unter [Geräteauthentifizierung und Autorisierung für AWS IoT Greengrass im AWS IoT Greengrass Version 1 Entwicklerhandbuch](#).

Konfiguration und Schwachstellenanalyse in AWS IoT SiteWise

IoT-Flotten können aus einer großen Anzahl von Geräten mit unterschiedlichsten Funktionen bestehen, sind langlebig und geografisch verteilt. Aufgrund dieser Merkmale ist die Flotteneinrichtung komplex und fehleranfällig. Da Geräte in der Regel nur über begrenzte Rechenleistung, Arbeitsspeicher und Speicherplatz verfügen, können sie Verschlüsselung und andere Sicherheitsmaßnahmen nicht immer unterstützen. Außerdem verwenden Geräte häufig Software mit bekannten Schwachstellen. Diese Faktoren machen IoT-Flotten zu einem attraktiven Ziel für Hacker und erschweren die kontinuierliche Sicherung Ihrer Geräteflotte.

AWS IoT Device Defender begegnet diesen Herausforderungen durch die Bereitstellung von Tools zur Identifizierung von Sicherheitsproblemen und Abweichungen von bewährten Verfahren. Wird verwendet, AWS IoT Device Defender um verbundene Geräte zu analysieren, zu prüfen und zu überwachen, um ungewöhnliches Verhalten zu erkennen und Sicherheitsrisiken zu minimieren. AWS IoT Device Defender kann Geräteflotten überprüfen, um sicherzustellen, dass sie sich an bewährte Sicherheitsmethoden halten, und um abnormales Verhalten auf Geräten zu erkennen. Auf diese Weise können Sie einheitliche Sicherheitsrichtlinien für Ihre gesamte AWS IoT Geräteflotte durchsetzen und schnell reagieren, wenn Geräte kompromittiert werden. Weitere Informationen finden Sie unter [Was ist AWS IoT Device Defender](#) im AWS IoT Device Defender-Entwicklerhandbuch.

Wenn Sie SiteWise Edge-Gateways verwenden, um Daten in den Dienst aufzunehmen, liegt es in Ihrer Verantwortung, die Umgebung Ihres SiteWise Edge-Gateways zu konfigurieren und zu warten. Diese Verantwortung umfasst die Aktualisierung auf die neuesten Versionen der Systemsoftware, AWS IoT Greengrass Software und des Connectors des SiteWise AWS IoT SiteWise Edge-Gateways. Weitere Informationen finden [Sie unter Konfiguration des AWS IoT Greengrass Kerns](#) im AWS IoT Greengrass Version 1Entwicklerhandbuch und [SiteWise Edge-Gateways verwalten](#).

VPC-Endpunkte fürAWS IoT SiteWise

Ein Schnittstellen-VPC-Endpunkt stellt eine private Verbindung zwischen Ihrer Virtual Private Cloud (VPC) und her. AWS IoT SiteWise [AWS PrivateLink](#)versorgt Schnittstellenendpunkte und ermöglicht so den privaten Zugriff auf AWS IoT SiteWise API-Operationen. AWS IoT SiteWiseunterstützt über seine Schnittstellenendpunkte sowohl IPv4 als auch IPv6 (Dual-Stack). Sie können die Notwendigkeit eines Internet-Gateways, eines NAT-Geräts, einer VPN-Verbindung oder umgehen. AWS Direct Connect Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um mit AWS IoT SiteWise API-Vorgängen zu kommunizieren. Datenverkehr zwischen Ihrer VPC und verlässt das AWS Netzwerk AWS IoT SiteWise nicht.

Jeder Schnittstellenendpunkt wird durch eine oder mehrere [Elastic Network-Schnittstellen](#) in Ihren Subnetzen dargestellt.

Bevor Sie einen VPC-Schnittstellen-Endpunkt für einrichtenAWS IoT SiteWise, lesen Sie den Abschnitt [Zugriff auf einen AWS Dienst mithilfe eines VPC-Schnittstellen-Endpunkts](#) im AWS PrivateLink Handbuch.

API-Operationen für VPC-Endpunkte inAWS IoT SiteWise

AWS IoT SiteWiseunterstützt Aufrufe der folgenden AWS IoT SiteWise API-Operationen von Ihrer VPC aus:

- Verwenden Sie für alle API-Operationen auf der Datenebene den folgenden Endpunkt:
region Ersetzen Sie durch AWS-Region

```
data.iotsitewise.region.amazonaws.com
```

Die API-Operationen auf der Datenebene umfassen Folgendes:

- [BatchGetAssetPropertyValue](#)
- [BatchGetAssetPropertyValueHistory](#)

- [BatchPutAssetPropertyValue](#)
 - [GetAssetPropertyAggregates](#)
 - [GetAssetPropertyValue](#)
 - [GetAssetPropertyValueHistory](#)
 - [GetInterpolatedAssetPropertyValues](#)
- Verwenden Sie für die API-Operationen auf der Steuerungsebene, mit denen Sie Asset-Modelle, Assets, SiteWise Edge-Gateways, Tags und Kontokonfigurationen verwalten, den folgenden Endpunkt. Ersetze *region* durch deine AWS-Region.

```
api.iotsitewise.region.amazonaws.com
```

Zu den unterstützten API-Vorgängen auf der Kontrollebene gehören:

- [AssociateAssets](#)
- [CreateAsset](#)
- [CreateAssetModel](#)
- [DeleteAsset](#)
- [DeleteAssetModel](#)
- [DeleteDashboard](#)
- [DescribeAsset](#)
- [DescribeAssetModel](#)
- [DescribeAssetProperty](#)
- [DescribeDashboard](#)
- [DescribeLoggingOptions](#)
- [DisassociateAssets](#)
- [ListAssetModels](#)
- [ListAssetRelationships](#)
- [ListAssets](#)
- [ListAssociatedAssets](#)
- [PutLoggingOptions](#)
- [UpdateAsset](#)
- [UpdateAssetModel](#)

- [UpdateAssetProperty](#)
- [CreateGateway](#)
- [DeleteGateway](#)
- [DescribeDefaultEncryptionConfiguration](#)
- [DescribeGateway](#)
- [DescribeGatewayCapabilityConfiguration](#)
- [DescribeStorageConfiguration](#)
- [ListGateways](#)
- [ListTagsForResource](#)
- [UpdateGateway](#)
- [UpdateGatewayCapabilityConfiguration](#)
- [PutDefaultEncryptionConfiguration](#)
- [PutStorageConfiguration](#)
- [TagResource](#)
- [UntagResource](#)

 Note

Der VPC-Schnittstellen-Endpunkt für die API-Operationen der Kontrollebene unterstützt derzeit keine Aufrufe der folgenden SiteWise Monitor-API-Operationen:

- [BatchAssociateProjectAssets](#)
- [BatchDisassociateProjectAssets](#)
- [CreateAccessPolicy](#)
- [CreateDashboard](#)
- [CreatePortal](#)
- [CreateProject](#)
- [DeleteAccessPolicy](#)
- [DeletePortal](#)
- [DeleteProject](#)
- [DescribeAccessPolicy](#)

- [DescribeProject](#)
- [ListAccessPolicies](#)
- [ListDashboards](#)
- [ListPortals](#)
- [ListProjects](#)
- [ListProjectAssets](#)
- [UpdateAccessPolicy](#)
- [UpdateDashboard](#)
- [UpdatePortal](#)
- [UpdateProject](#)

Erstellen Sie einen Schnittstellen-VPC-Endpunkt für AWS IoT SiteWise

Um einen VPC-Endpunkt für den AWS IoT SiteWise Service zu erstellen, verwenden Sie entweder die Amazon VPC-Konsole oder die AWS Command Line Interface (AWS CLI). Weitere Informationen finden Sie im AWS PrivateLinkHandbuch unter [Zugreifen auf einen AWS Dienst über einen Schnittstellen-VPC-Endpunkt](#).

Erstellen Sie einen VPC-Endpunkt für, AWS IoT SiteWise indem Sie einen der folgenden Dienstnamen verwenden:

- Verwenden Sie für die API-Operationen auf Datenebene den folgenden Dienstnamen:

```
com.amazonaws.region.iotsitewise.data
```

- Verwenden Sie für die API-Operationen auf der Steuerungsebene den folgenden Dienstnamen:

```
com.amazonaws.region.iotsitewise.api
```

Zugriff AWS IoT SiteWise über einen VPC-Endpunkt mit Schnittstelle

Wenn Sie einen Schnittstellenendpunkt erstellen, generieren wir endpunktspezifische DNS-Hostnamen, mit denen Sie kommunizieren können. AWS IoT SiteWise Die private DNS-Option ist

standardmäßig aktiviert. Weitere Informationen finden Sie unter [Verwenden von privat gehosteten Zonen](#) im Amazon VPC-Benutzerhandbuch.

Wenn Sie privates DNS für den Endpunkt aktivieren, können Sie API-Anfragen an einen AWS IoT SiteWise der folgenden VPC-Endpunkte stellen.

- Verwenden Sie für die API-Operationen auf der Datenebene den folgenden Endpunkt:
region Ersetzen Sie durch Ihren AWS-Region

```
data.iotsitewise.region.amazonaws.com
```

- Verwenden Sie für die API-Operationen auf der Steuerungsebene den folgenden Endpunkt:
region Ersetzen Sie durch Ihre AWS-Region.

```
api.iotsitewise.region.amazonaws.com
```

Wenn Sie privates DNS für den Endpunkt deaktivieren, müssen Sie für den Zugriff AWS IoT SiteWise über den Endpunkt wie folgt vorgehen:

1. Geben Sie die VPC-Endpunkt-URL in API-Anfragen an.

- Verwenden Sie für die API-Operationen auf der Datenebene die folgende Endpunkt-URL.
Ersetzen Sie *vpc-endpoint-id* und *region* durch Ihre VPC-Endpunkt-ID und Region.

```
vpc-endpoint-id.data.iotsitewise.region.vpce.amazonaws.com
```

- Verwenden Sie für die API-Operationen auf der Kontrollebene die folgende Endpunkt-URL.
Ersetzen Sie *vpc-endpoint-id* und *region* durch Ihre VPC-Endpunkt-ID und Region.

```
vpc-endpoint-id.api.iotsitewise.region.vpce.amazonaws.com
```

2. Deaktivieren Sie die Host-Präfix-Injektion. Die AWS SDKs AWS CLI und stellen dem Service-Endpunkt verschiedene Host-Präfixe voran, wenn Sie die einzelnen API-Operationen aufrufen. Diese Funktion veranlasst die AWS SDKs AWS CLI und, URLs zu erzeugen, die nicht gültig sind, AWS IoT SiteWise wenn Sie einen VPC-Endpunkt angeben.

Important

Sie können die Hostpräfixinjektion in der AWS CLI oder der nicht deaktivieren. AWS -Tools für PowerShell Das heißt, wenn Sie privates DNS deaktivieren, können Sie diese Tools

nicht für den Zugriff AWS IoT SiteWise über den VPC-Endpunkt verwenden. Aktivieren Sie privates DNS, um das AWS CLI oder das für den AWS -Tools für PowerShell Zugriff AWS IoT SiteWise über den Endpunkt zu verwenden.

Weitere Informationen zur Deaktivierung der Hostpräfixinjektion in den AWS SDKs finden Sie in den folgenden Dokumentationsabschnitten für jedes SDK:

- [AWS SDK für C++](#)
- [AWS SDK für Go](#)
- [AWS SDK für Go v2](#)
- [AWS SDK für Java](#)
- [AWS SDK for Java 2.x](#)
- [AWS SDK für JavaScript](#)
- [AWS SDK für .NET](#)
- [AWS SDK für PHP](#)
- [AWS SDK für Python \(Boto3\)](#)
- [AWS SDK für Ruby](#)

Weitere Informationen finden Sie im AWS PrivateLinkHandbuch unter [Zugreifen auf einen AWS Dienst über einen Schnittstellen-VPC-Endpunkt](#).

Erstellen Sie eine VPC-Endpunktrichtlinie fürAWS IoT SiteWise

Sie können eine Endpunktrichtlinie an Ihren VPC-Endpunkt anhängen, der den Zugriff auf AWS IoT SiteWise steuert. Die Richtlinie gibt die folgenden Informationen an:

- Der Principal, der Operationen ausführen kann.
- Die Operationen, die ausgeführt werden können.
- Die Ressourcen, auf denen Operationen ausgeführt werden können.

Weitere Informationen finden Sie unter [Steuern des Zugriffs auf VPC-Endpunkte mithilfe von Endpunktrichtlinien](#) im Amazon-VPC-Benutzerhandbuch.

Beispiel: VPC-Endpunktrichtlinie fürAWS IoT SiteWiseactions

Erstellen einer VPC-Endpunktrichtlinie

Im Folgenden finden Sie ein Beispiel für eine Endpunktrichtlinie für AWS IoT SiteWise. Wenn diese Richtlinie an einen Endpunkt angehängt ist, gewährt sie dem Benutzer *iotsitewiseadmin* Zugriff *123456789012* auf die aufgelisteten AWS IoT SiteWise Aktionen für das angegebene Asset. AWS

```
{
  "Statement": [
    {
      "Action": [
        "iotsitewise:CreateAsset",
        "iotsitewise:ListGateways",
        "iotsitewise:ListTagsForResource"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:iotsitewise:us-west-2:123456789012:asset/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "Principal": {
        "AWS": [
          "123456789012:user/iotsitewiseadmin"
        ]
      }
    ]
  ]
}
```

Bewährte Sicherheitsmethoden für AWS IoT SiteWise

Dieses Thema enthält bewährte Sicherheitsmethoden für AWS IoT SiteWise.

Verwenden Sie Authentifizierungsdaten auf Ihren OPC UA-Servern

Erfordern Sie Authentifizierungsdaten, um eine Verbindung zu Ihren OPC UA-Servern herzustellen. Weitere Informationen hierzu finden Sie in der Dokumentation für Ihren Server. Damit Ihr SiteWise Edge-Gateway dann eine Verbindung zu Ihren OPC UA-Servern herstellen kann, fügen Sie Ihrem SiteWise Edge-Gateway Serverauthentifizierungsgeheimnisse hinzu. Weitere Informationen finden Sie unter [Konfigurieren Sie die Datenquellenauthentifizierung für SiteWise Edge](#).

Verwenden Sie verschlüsselte Kommunikationsmodi für Ihre OPC UA-Server

Wählen Sie einen nicht veralteten Sicherheitsmodus für verschlüsselte Nachrichten, wenn Sie Ihre OPC UA-Quellen für Ihr Edge-Gateway konfigurieren. SiteWise Dies trägt zum Schutz Ihrer Industriedaten bei der Übertragung von Ihren OPC UA-Servern zum Edge-Gateway bei. SiteWise Weitere Informationen erhalten Sie unter [Daten in Übertragung über das lokale Netzwerk](#) und [Richten Sie eine OPC UA-Quelle in SiteWise Edge ein](#).

Halten Sie Ihre Komponenten auf dem neuesten Stand

Wenn Sie SiteWise Edge-Gateways verwenden, um Daten in den Dienst aufzunehmen, liegt es in Ihrer Verantwortung, die Umgebung Ihres Edge-Gateways zu konfigurieren und zu warten. SiteWise Diese Verantwortung umfasst die Aktualisierung auf die neuesten Versionen der Systemsoftware, AWS IoT Greengrass Software und Konnektoren des Gateways.

Note

Der AWS IoT SiteWise Edge-Connector speichert Geheimnisse in Ihrem Dateisystem. Diese Geheimnisse steuern, wer die in Ihrem SiteWise Edge-Gateway zwischengespeicherten Daten einsehen kann. Es wird dringend empfohlen, die Festplatten- oder Dateisystemverschlüsselung für das System zu aktivieren, auf dem Ihr SiteWise Edge-Gateway ausgeführt wird.

Informationen zum Aktualisieren von Komponenten in der AWS IoT SiteWise Konsole finden Sie unter [Ändern Sie die Version der SiteWise Edge Gateway-Komponentenpakete](#)

Verschlüsseln Sie das SiteWise Dateisystem Ihres Edge-Gateways

Verschlüsseln und sichern Sie Ihr SiteWise Edge-Gateway, sodass Ihre Industriedaten sicher sind, wenn sie das SiteWise Edge-Gateway passieren. Wenn Ihr SiteWise Edge-Gateway über ein Hardware-Sicherheitsmodul verfügt, können Sie es so konfigurieren, AWS IoT Greengrass dass Ihr SiteWise Edge-Gateway gesichert wird. Weitere Informationen finden Sie unter [Hardware-sicherheitsintegration](#) im AWS IoT Greengrass Version 1Entwicklerhandbuch. Andernfalls finden Sie in der Dokumentation Ihres Betriebssystems Informationen zum Verschlüsseln und Sichern des Dateisystems.

Sicherer Zugriff auf Ihre Edge-Konfiguration

Geben Sie weder Ihr Passwort für die Edge-Console-Anwendung noch das Passwort Ihrer SiteWise Monitor-Anwendung weiter. Geben Sie dieses Passwort nicht an Orten ab, an denen es für jedermann sichtbar ist. Implementieren Sie eine Richtlinie zur korrekten Passwortrotation, indem Sie ein geeignetes Ablaufdatum für Ihr Passwort konfigurieren.

Daten sichern auf Industrielles Edge-Management von Siemens

Die Gerätedaten, die Sie mit AWS IoT SiteWise Edge teilen möchten, werden in Ihren Siemens IEM Databus Konfigurationsthemen festgelegt. Wenn Sie Themen auswählen, die Sie mit SiteWise Edge teilen möchten, geben Sie Daten auf Themenebene weiter. AWS IoT SiteWise Das Siemens Industrial Edge Marketplace ist ein unabhängiger Marktplatz, unabhängig von. AWS Um Ihre gemeinsam genutzten Daten zu schützen, kann die SiteWise Edge-Anwendung nur ausgeführt werden, wenn Sie sie verwenden Siemens Secured Storage. Weitere Informationen finden Sie in der Siemens Dokumentation unter [Sicherer Speicher](#).

Gewähren SiteWise Sie Monitor-Benutzern die geringstmöglichen Berechtigungen

Folgen Sie dem Prinzip der geringsten Rechte, indem Sie die Mindestanzahl an Zugriffsrichtlinienberechtigungen für Ihre Portalbenutzer verwenden.

- Definieren Sie bei der Erstellung eines Portals eine Rolle, die die Mindestanzahl der für dieses Portal erforderlichen Komponenten zulässt. Weitere Informationen finden Sie unter [Verwenden Sie Servicerollen für AWS IoT SiteWise Monitor](#).
- Wenn Sie und Ihre Portaladministratoren Projekte erstellen und freigeben, verwenden Sie die Mindestanzahl an Komponenten, die für dieses Projekt erforderlich sind.
- Wenn eine Identität keinen Zugriff mehr auf ein Portal oder Projekt benötigt, entfernen Sie sie aus dieser Ressource. Wenn diese Identität für Ihre Organisation nicht mehr gilt, löschen Sie diese Identität aus Ihrem Identitätsspeicher.

Die bewährte Methode nach dem Prinzip „Least Principles“ gilt auch für IAM-Rollen. Weitere Informationen finden Sie unter [Best Practices für Richtlinien](#).

Legen Sie vertrauliche Informationen nicht offen

Sie sollten verhindern, dass Anmeldeinformationen und andere vertrauliche Informationen wie beispielsweise personenbezogene Daten protokolliert werden. Es wird empfohlen, die folgenden Sicherheitsvorkehrungen zu implementieren, auch wenn für den Zugriff auf lokale Protokolle auf einem SiteWise Edge-Gateway Root-Rechte und für den Zugriff auf CloudWatch Protokolle IAM-Berechtigungen erforderlich sind.

- Verwenden Sie keine vertraulichen Informationen in Namen, Beschreibungen oder Eigenschaften Ihrer Komponenten oder Modelle.
- Verwenden Sie keine vertraulichen Informationen in SiteWise Edge-Gateways oder Quellnamen.
- Verwenden Sie keine vertraulichen Informationen in Namen oder Beschreibungen Ihrer Portale, Projekte oder Dashboards.

FollowAWS IoT GreengrassBewährte Methoden für die Sicherheit

Befolgen Sie die bewährten AWS IoT Greengrass Sicherheitsmethoden für Ihr SiteWise Edge-Gateway. Weitere Informationen finden Sie unter [Bewährte Sicherheitsmethoden](#) im AWS IoT Greengrass Version 1Entwicklerhandbuch.

Weitere Informationen finden Sie auch unter

- [Bewährte Sicherheitsmethoden](#) im AWS IoTEntwicklerhandbuch
- [Zehn goldene Sicherheitsregeln für industrielle IoT-Lösungen](#)

Einloggen und einloggen AWS IoT SiteWise

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer AWS IoT SiteWise anderen AWS Lösungen. AWS IoT SiteWise unterstützt die folgenden Überwachungstools, um den Service zu überwachen, zu melden, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen zu ergreifen:

- Amazon CloudWatch überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie laufen, AWS in Echtzeit. Erfassen und verfolgen Sie Kennzahlen, erstellen Sie maßgeschneiderte Dashboards und richten Sie Alarmer ein, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen bestimmten Schwellenwert erreicht. Sie können beispielsweise die CPU-Auslastung oder andere Kennzahlen Ihrer EC2 Amazon-Instances CloudWatch verfolgen und bei Bedarf automatisch neue Instances starten. Weitere Informationen finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).
- Amazon CloudWatch Logs überwacht, speichert und greift auf Ihre Protokolldateien von SiteWise Edge-Gateways und anderen CloudTrail Quellen zu. CloudWatch Logs kann Informationen in den Protokolldateien überwachen und Sie benachrichtigen, wenn bestimmte Schwellenwerte erreicht werden. Sie können Ihre Protokolldaten auch in einem sehr robusten Speicher archivieren. Weitere Informationen finden Sie im [Amazon CloudWatch Logs-Benutzerhandbuch](#).
- AWS CloudTrail erfasst API-Aufrufe und zugehörige Ereignisse, die von oder im Namen Ihres AWS Kontos getätigt wurden. Anschließend werden die Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket übermittelt. Sie können feststellen, welche Benutzer und Konten angerufen wurden AWS, von welcher Quell-IP-Adresse aus die Anrufe getätigt wurden und wann die Anrufe erfolgten. Weitere Informationen finden Sie im [AWS CloudTrail - Benutzerhandbuch](#).

Themen

- [Mit Amazon CloudWatch Logs überwachen](#)
- [SiteWise Edge-Gateway-Protokolle überwachen](#)
- [Überwachen Sie AWS IoT SiteWise mit CloudWatch Amazon-Metriken](#)
- [AWS IoT SiteWise API-Aufrufe protokollieren mit AWS CloudTrail](#)

Mit Amazon CloudWatch Logs überwachen

Stellen Sie AWS IoT SiteWise die Konfiguration so ein, dass Informationen in CloudWatch Logs protokolliert werden, um den Dienst zu überwachen und Fehler zu beheben.

Wenn Sie die AWS IoT SiteWise Konsole verwenden, AWS IoT SiteWise wird eine dienstbezogene Rolle erstellt, die es dem Dienst ermöglicht, Informationen in Ihrem Namen zu protokollieren. Wenn Sie die AWS IoT SiteWise Konsole nicht verwenden, müssen Sie manuell eine dienstbezogene Rolle erstellen, um Protokolle zu empfangen. Weitere Informationen finden Sie unter [Erstellen Sie eine serviceverknüpfte Rolle für AWS IoT SiteWise](#).

Sie benötigen eine Ressourcenrichtlinie, die es ermöglicht, Protokollereignisse in CloudWatch Streams AWS IoT SiteWise zu speichern. Führen Sie den folgenden Befehl aus, um eine Ressourcenrichtlinie für CloudWatch Logs zu erstellen und zu aktualisieren. *logging-policy-name* Ersetzen Sie ihn durch den Namen der zu erstellenden Richtlinie.

```
aws logs put-resource-policy --policy-name logging-policy-name --policy-document "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Sid\": \"IoTSiteWiseToCloudWatchLogs\", \"Effect\": \"Allow\", \"Principal\": { \"Service\": [ \"iotsitewise.amazonaws.com\" ] }, \"Action\": \"logs:PutLogEvents\", \"Resource\": \"*\" } ] }"
```

CloudWatch Logs unterstützt auch die Kontextschlüssel [aws: SourceArn](#) und [aws: SourceAccount](#) condition. Diese Bedingungskontextschlüssel sind optional.

Um eine Ressourcenrichtlinie zu erstellen oder AWS IoT SiteWise zu aktualisieren, die es erlaubt, nur Protokolle, die mit der angegebenen AWS IoT SiteWise Ressource verknüpft sind, in CloudWatch Streams zu speichern, führen Sie den Befehl aus und gehen Sie wie folgt vor:

- *logging-policy-name* Ersetzen Sie ihn durch den Namen der zu erstellenden Richtlinie.
- *source-ARN* Ersetzen Sie es durch den ARN Ihrer AWS IoT SiteWise Ressource, z. B. eines Asset-Modells oder eines Assets. Den ARN für jeden AWS IoT SiteWise Ressourcentyp finden Sie unter [Ressourcentypen definiert von AWS IoT SiteWise](#) in der Service Authorization Reference.
- *account-ID* Ersetzen Sie es durch die AWS Konto-ID, die der angegebenen AWS IoT SiteWise Ressource zugeordnet ist.

```
aws logs put-resource-policy --policy-name logging-policy-name --policy-document "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Sid\":
```

```
\ "IoTSiteWiseToCloudWatchLogs\", \ "Effect\": \ "Allow\", \ "Principal\": { \ "Service\": [ \ "iotsitewise.amazonaws.com\" ] }, \ "Action\": \ "logs:PutLogEvents\", \ "Resource\": \ "*\", \ "Condition\": { \ "StringLike\": { \ "aws:SourceArn\": [ \ "source-ARN\" ], \ "aws:SourceAccount\": [ \ "account-ID\" ] } } } }
```

Standardmäßig werden AWS IoT SiteWise keine Informationen in CloudWatch Logs protokolliert. Um die Protokollierung zu aktivieren, wählen Sie eine andere Protokollierungsebene als Deaktiviert (OFF). AWS IoT SiteWise unterstützt die folgenden Protokollierungsebenen:

- OFF— Die Protokollierung ist ausgeschaltet.
- ERROR— Fehler werden protokolliert.
- INFO— Fehler und Informationsmeldungen werden protokolliert.

Sie können SiteWise Edge-Gateways so konfigurieren, dass sie Informationen in CloudWatch Logs protokollieren. AWS IoT Greengrass Weitere Informationen finden Sie unter [SiteWise Edge-Gateway-Protokolle überwachen](#).

Sie können auch so konfigurieren AWS IoT Core , dass Informationen in CloudWatch Protokollen protokolliert werden, wenn Sie eine AWS IoT SiteWise Regelaktion beheben. Weitere Informationen finden Sie unter [Problembehandlung bei einer AWS IoT SiteWise Regelaktion](#).

Inhalt

- [Die Anmeldung verwalten AWS IoT SiteWise](#)
 - [Finden Sie Ihre Protokollierungsstufe](#)
 - [Ändern Sie Ihre Protokollierungsstufe](#)
- [Beispiel: Einträge in der AWS IoT SiteWise Protokolldatei](#)

Die Anmeldung verwalten AWS IoT SiteWise

Verwenden Sie die AWS IoT SiteWise Konsole oder AWS CLI für die folgenden Aufgaben zur Konfiguration der Protokollierung.

Finden Sie Ihre Protokollierungsstufe

Console

Gehen Sie wie folgt vor, um die aktuelle Protokollierungsstufe in der AWS IoT SiteWise -Konsole zu finden.

Um Ihre aktuelle AWS IoT SiteWise Protokollierungsstufe zu ermitteln

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Logging options (Protokollierungsoptionen) aus.

Der aktuelle Protokollierungsstatus wird unter Logging status (Protokollierungsstatus) angezeigt. Wenn die Protokollierung aktiviert ist, wird die aktuelle Protokollierungsstufe unter Ausführlichkeitsstufe angezeigt.

AWS CLI

Führen Sie den folgenden Befehl aus, um Ihre aktuelle AWS IoT SiteWise Protokollierungsstufe mit dem zu ermitteln. AWS CLI

```
aws iotsitewise describe-logging-options
```

Die Operation gibt eine Antwort mit Ihrer Protokollierungsstufe im folgenden Format zurück.

```
{
  "loggingOptions": {
    "level": "String"
  }
}
```

Ändern Sie Ihre Protokollierungsstufe

Gehen Sie wie folgt vor, um Ihre Protokollierungsstufe in der AWS IoT SiteWise Konsole oder mithilfe von zu ändern AWS CLI.

Console

Um Ihre AWS IoT SiteWise Protokollierungsstufe zu ändern

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Logging options (Protokollierungsoptionen) aus.
3. Wählen Sie Edit (Bearbeiten) aus.
4. Wählen Sie den Grad der Ausführlichkeit, den Sie aktivieren möchten.

5. Wählen Sie Save (Speichern) aus.

AWS CLI

Führen Sie den folgenden AWS CLI Befehl aus, um Ihre AWS IoT SiteWise Protokollierungsstufe zu ändern. `logging-level` Ersetzen Sie es durch die gewünschte Protokollierungsebene.

```
aws iotsitewise put-logging-options --logging-options level=logging-level
```

Beispiel: Einträge in der AWS IoT SiteWise Protokolldatei

Jeder AWS IoT SiteWise Protokolleintrag enthält Ereignisinformationen und relevante Ressourcen für dieses Ereignis, sodass Sie die Protokolldaten verstehen und analysieren können.

Das folgende Beispiel zeigt einen CloudWatch Logs-Eintrag, der AWS IoT SiteWise protokolliert, wann Sie ein Asset-Modell erfolgreich erstellt haben.

```
{
  "eventTime": "2020-05-05T00:10:22.902Z",
  "logLevel": "INFO",
  "eventType": "AssetModelCreationSuccess",
  "message": "Successfully created asset model.",
  "resources": {
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
  }
}
```

SiteWise Edge-Gateway-Protokolle überwachen

Sie können Ihr AWS IoT SiteWise Edge-Gateway so konfigurieren, dass Informationen in Amazon CloudWatch Logs oder im lokalen Dateisystem protokolliert werden.

Themen

- [Verwenden Sie Amazon CloudWatch Logs](#)
- [Loggen Sie sich in den Dienst ein AWS IoT SiteWise](#)

Verwenden Sie Amazon CloudWatch Logs

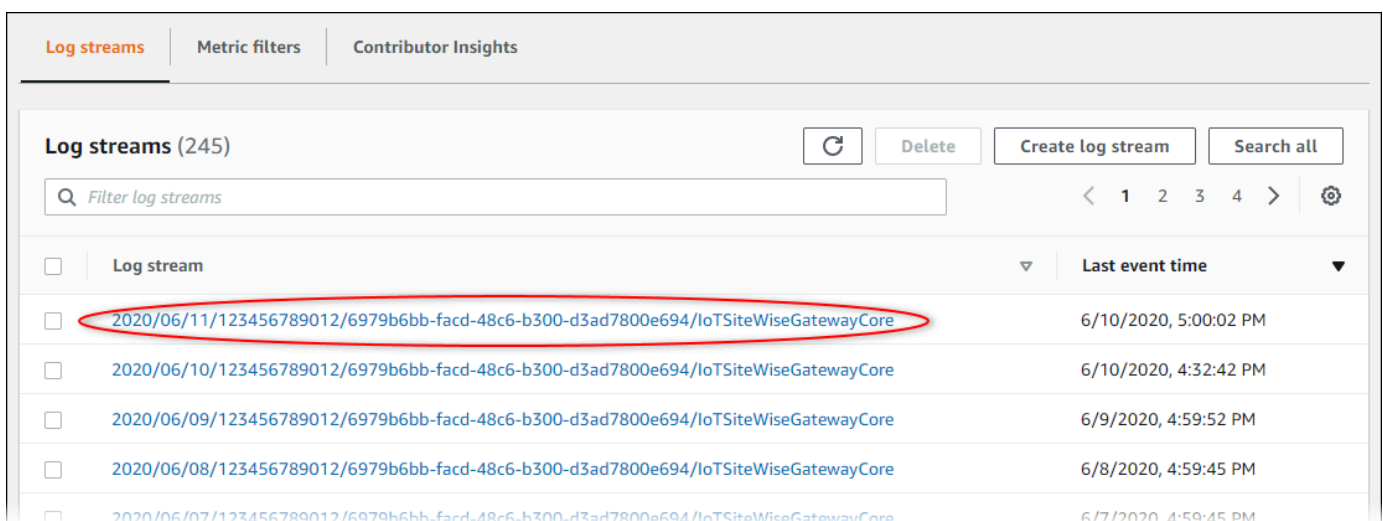
Sie können Ihr SiteWise Edge-Gateway so konfigurieren, dass CloudWatch Protokolle an Logs gesendet werden. Weitere Informationen finden Sie unter [Aktivieren der Protokollierung für CloudWatch Protokolle](#) im AWS IoT Greengrass Version 2 Entwicklerhandbuch.

So konfigurieren Sie CloudWatch Protokolle und greifen auf sie zu (Konsole)

1. Navigieren Sie zur [CloudWatch-Konsole](#).
2. Wählen Sie im Navigationsbereich Protokollgruppen aus.
3. Sie finden die AWS IoT SiteWise Komponentenprotokolle in den folgenden Protokollgruppen:
 - `/aws/greengrass/UserComponent/region/aws.iot.SiteWiseEdgeCollectorOpcua`— Die Protokolle für die Komponente des SiteWise Edge-Gateways, die Daten aus den OPC-UA-Quellen des SiteWise Edge-Gateways sammelt.
 - `/aws/greengrass/UserComponent/region/aws.iot.SiteWiseEdgePublisher`— Die Protokolle für die Komponente des SiteWise Edge-Gateways, für die OPC UA-Datenströme veröffentlicht werden. AWS IoT SiteWise

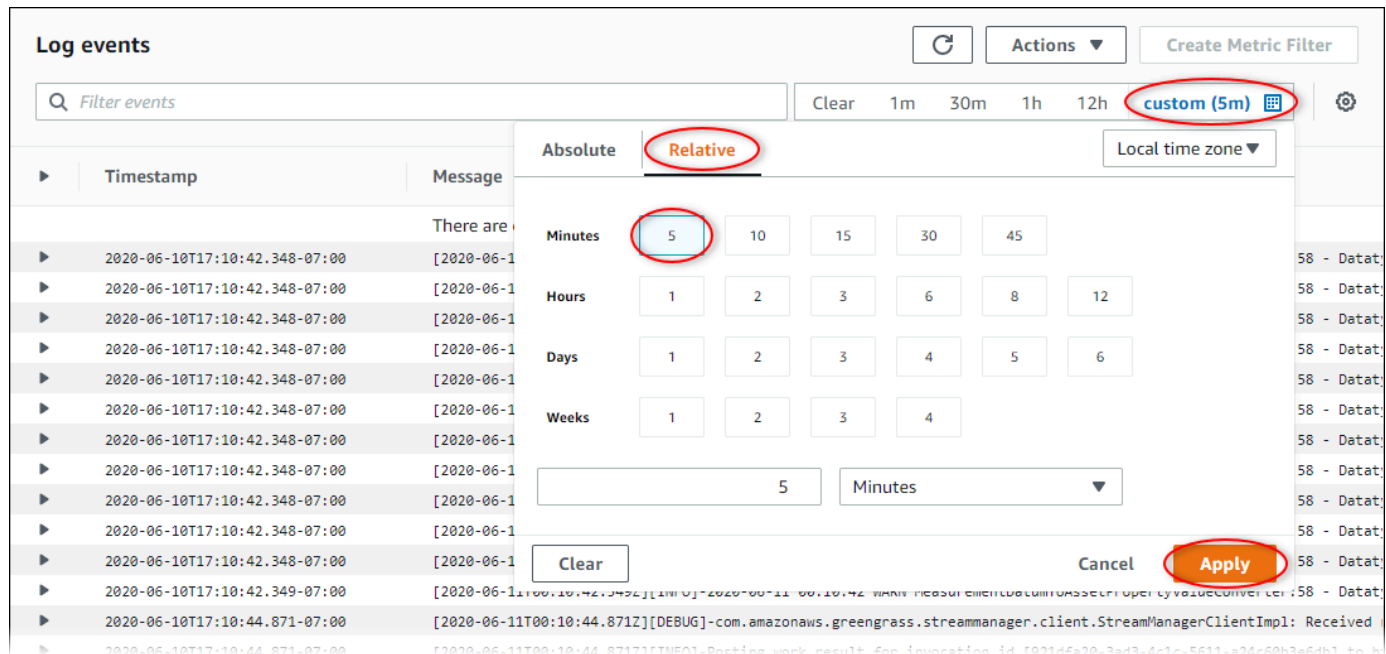
Wählen Sie die Protokollgruppe für die Funktion aus, die debuggt werden soll.

4. Wählen Sie einen Protokollstream aus, dessen Name mit dem Namen Ihrer AWS IoT Greengrass Gruppe endet. CloudWatch Zeigt standardmäßig den neuesten Log-Stream zuerst an.



5. Führen Sie die folgenden Schritte aus, um Protokolle der letzten 5 Minuten anzuzeigen:

- Wählen Sie in der oberen rechten Ecke Custom (Benutzerdefiniert).
- Wählen Sie Relative (Relativ).
- Wählen Sie 5 Minuten.
- Wählen Sie Anwenden aus.



- (Optional) Um weniger Protokolle anzuzeigen, können Sie rechts oben 1m auswählen.
- Scrollen Sie zum Ende der Protokolleinträge, um die neuesten Protokolle anzuzeigen.

Loggen Sie sich in den Dienst ein AWS IoT SiteWise

SiteWise Edge-Gateway-Geräte enthalten Dienstprotokolldateien, die beim Debuggen von Problemen helfen. Die folgenden Abschnitte helfen Ihnen dabei, die Dienstprotokolldateien für die Komponenten AWS IoT SiteWise OPC UA Collector und AWS IoT SiteWise Publisher zu finden und zu verwenden.

AWS IoT SiteWise OPC UA Collector-Serviceprotokolldatei

Die AWS IoT SiteWise OPC UA Collector-Komponente verwendet die folgende Protokolldatei.

Linux

```
/greengrass/v2/logs/aws.iot.SiteWiseEdgeCollectorOpcua.log
```

Windows

```
C:\greengrass\v2\logs\aws.iot.SiteWiseEdgeCollectorOpcua.log
```

Um die Logs dieser Komponente einzusehen

- Führen Sie den folgenden Befehl auf dem Kerngerät aus, um die Protokolldatei dieser Komponente in Echtzeit anzuzeigen. Ersetzen Sie */greengrass/v2* oder *C:\greengrass\v2* durch den Pfad zum AWS IoT Greengrass Stammordner.

Linux

```
sudo tail -f /greengrass/v2/logs/aws.iot.SiteWiseEdgeCollectorOpcua.log
```

Windows (PowerShell)

```
Get-Content C:\greengrass\v2\logs\aws.iot.SiteWiseEdgeCollectorOpcua.log -Tail  
10 -Wait
```

AWS IoT SiteWise Publisher-Dienstprotokolldatei

Die AWS IoT SiteWise Publisher-Komponente verwendet die folgende Protokolldatei.

Linux

```
/greengrass/v2/logs/aws.iot.SiteWiseEdgePublisher.log
```

Windows

```
C:\greengrass\v2\logs\aws.iot.SiteWiseEdgePublisher.log
```

Um die Protokolle dieser Komponente anzuzeigen

- Führen Sie den folgenden Befehl auf dem Kerngerät aus, um die Protokolldatei dieser Komponente in Echtzeit anzuzeigen. Ersetzen Sie */greengrass/v2* oder *C:\greengrass\v2* durch den Pfad zum AWS IoT Greengrass Stammordner.

Linux

```
sudo tail -f /greengrass/v2/logs/aws.iot.SiteWiseEdgePublisher.log
```

Windows (PowerShell)

```
Get-Content C:\greengrass\v2\logs\aws.iot.SiteWiseEdgePublisher.log -Tail 10 -  
Wait
```

Überwachen Sie AWS IoT SiteWise mit CloudWatch Amazon-Metriken

Sie können die AWS IoT SiteWise Nutzung überwachen CloudWatch, wobei Rohdaten gesammelt und zu lesbaren Metriken verarbeitet werden, die nahezu in Echtzeit verfügbar sind. Diese Statistiken werden 15 Monate gespeichert, damit Sie auf Verlaufsinformationen zugreifen können und einen besseren Überblick darüber erhalten, wie Ihre Webanwendung oder der Service ausgeführt werden. Sie können auch Alarme einrichten, die auf bestimmte Grenzwerte achten und Benachrichtigungen senden oder Aktivitäten auslösen, wenn diese Grenzwerte erreicht werden. Weitere Informationen finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).

AWS IoT SiteWise veröffentlicht die in den folgenden Abschnitten aufgeführten Metriken und Dimensionen im AWS/IoTSiteWise Namespace.

Tip

AWS IoT SiteWise veröffentlicht Metriken in einem Intervall von einer Minute. Wenn Sie diese Metriken in der CloudWatch Konsole grafisch anzeigen, empfehlen wir Ihnen, einen Zeitraum von 1 Minute zu wählen. So können Sie die Metrikdaten in der höchsten verfügbaren Auflösung anzeigen.

Themen

- [AWS IoT Greengrass Version 2 Gateway-Metriken](#)

AWS IoT Greengrass Version 2 Gateway-Metriken

AWS IoT SiteWise veröffentlicht Gateway-Metriken für Classic-Streams, V2-Gateways und MQTT-fähige V3-Gateways. Sofern nicht anders angegeben, gilt jede Metrik für beide selbst gehosteten Gateway-Versionen. Alle SiteWise Edge-Gateway-Metriken werden in einem Intervall von einer Minute veröffentlicht.

SiteWise Edge-Gateway-Metriken

Metrik	Beschreibung
<code>Gateway.AvailableMemory</code>	Der verfügbare Speicher eines SiteWise Edge-Gateways. Einheit: Byte Dimension: Keine
<code>Gateway.AvailableDiskSpace</code>	Der verfügbare Festplattenspeicher eines SiteWise Edge-Gateways. Einheit: Byte Dimension: Keine
<code>Gateway.CloudConnectivity</code>	Der Cloud-Konnektivitätsstatus eines SiteWise Edge-Gateways. Einheit: keine Dimension: GatewayId
<code>Gateway.CpuUsage</code>	Die CPU-Auslastung eines SiteWise Edge-Gateways. Einheit: Prozentsatz Dimension: Keine
<code>Gateway.TotalDiskSpace</code>	Der gesamte Festplattenspeicher eines SiteWise Edge-Gateways.

Metrik	Beschreibung
	Einheit: Byte Dimension: Keine
Gateway.TotalMemory	Der gesamte Speicher eines SiteWise Edge-Gateways. Einheit: Byte Dimension: Keine
Gateway.UsedDiskSpace	Der verwendete Festplattenspeicher eines SiteWise Edge-Gateways. Einheit: Byte Dimension: Keine
Gateway.UsedMemory	Der verwendete Speicher eines SiteWise Edge-Gateways. Einheit: Byte Dimension: Keine
Gateway.UsedPercentageDiskSpace	Der verwendete Prozentsatz des Festplattenspeichers eines SiteWise Edge-Gateways. Einheit: Byte Dimension: Keine
Gateway.UsedPercentageMemory	Der prozentuale Anteil des verwendeten Speichers eines SiteWise Edge-Gateways. Einheit: Byte Dimension: Keine

AWS IoT SiteWise Metriken für Herausgeber

Metrik	Beschreibung
<code>IoTSiteWisePublisher.ComponentBuildVersion</code>	Diese Metrik gibt die Build-Version der SiteWise IoT-Publisher-Komponente an, die auf dem Gateway ausgeführt wird. Ein Wert von 1 bedeutet, dass auf dem Gateway eine Version des Publishers ausgeführt wird, die der <code>ComponentBuildVersion</code> Dimension entspricht. Einheit: 1 Abmessungen: GatewayId, ComponentBuildVersion
<code>IoTSiteWisePublisher.DroppedCount</code>	Die Anzahl der Datenpunkte, die von einem SiteWise Edge-Gateway (GatewayId) gelöscht und nicht in der Cloud veröffentlicht werden. Sie werden jede Minute generiert. Einheit: Anzahl Abmessungen: GatewayId
<code>IoTSiteWisePublisher.Heartbeat</code>	Wird jede Minute vom Publisher im SiteWise Edge-Gateway generiert. Einheit: 1 (1 steht dafür, dass der Publisher läuft und der Datenpunkt fehlt, was bedeutet, dass der Publisher nicht läuft.) Abmessungen: GatewayId
<code>IoTSiteWisePublisher.IsConnectedToMqttBroker</code>	Wird jede Minute vom Publisher im SiteWise Edge-Gateway generiert. Einheit: 1 (1 steht für den Herausgeber, der mit einem MQTT-Broker verbunden ist.)

Metrik	Beschreibung
	Abmessungen: GatewayId
<code>IoTSiteWisePublisher.MessageCheckpointPersistenceErrorCount</code>	Die Metrik gibt an, dass das Gateway ein Problem mit der Checkpoint-Datei erkannt hat, die zur Nachverfolgung der vom Herausgeber verarbeiteten Daten verwendet wird. Der Wert von 1 bedeutet, dass ein Fehler aufgetreten ist. Einheit: keine Abmessungen: AccountId, GatewayId
<code>IoTSiteWisePublisher.MqttMessageReceivedSuccessCount</code>	Die Anzahl der Nachrichten, die der Publisher erfolgreich vom MQTT-Broker empfangen hat und die jede Minute generiert wurden. Einheit: Anzahl Abmessungen: GatewayId
<code>IoTSiteWisePublisher.MqttReceivedSuccessBytes</code>	Die Anzahl der Byte an Nachrichtendaten, die der Herausgeber erfolgreich vom MQTT-Broker empfangen hat und die jede Minute generiert wurden. Einheit: Anzahl Abmessungen: GatewayId
<code>IoTSiteWisePublisher.NumberOfSubscriptionsToMqttBroker</code>	Die Anzahl der Themen, die der Herausgeber für den MQTT-Broker abonniert hat, generiert pro Minute. Ein mehrstufiges Wildcard-Thema wird als 1 gezählt. Einheit: Anzahl Abmessungen: GatewayId

Metrik	Beschreibung
<code>IoTSiteWisePublisher.NumberOfUniqueMqttTopicsReceived</code>	Die Anzahl der eindeutigen Themen, die der Publisher vom MQTT-Broker erhält und pro Minute generiert wird. Einheit: Anzahl Abmessungen: GatewayId
<code>IoTSiteWisePublisher.PublisherFailureCount</code>	Die Anzahl der pro Minute generierten Datenpunkte, die ein SiteWise Edge-Gateway (GatewayId) nicht veröffentlichen konnte. Einheit: Anzahl Abmessungen: GatewayId
<code>IoTSiteWisePublisher.PublisherRejectedCount</code>	Die Anzahl der pro Minute generierten Datenpunkte, die ein SiteWise Edge-Gateway (GatewayId) von der Cloud-Seite zurückgewiesen hat. Einheit: Anzahl Abmessungen: GatewayId
<code>IoTSiteWisePublisher.PublisherSuccessCount</code>	Die Anzahl der Datenpunkte, die ein SiteWise Edge-Gateway (GatewayId) erfolgreich in der Cloud veröffentlicht hat und die jede Minute generiert wurden. Einheit: Anzahl Abmessungen: GatewayId

Metrik	Beschreibung
<code>IoTSiteWisePublisher.PublisherToS3FailureCount</code>	Die Anzahl der Datenpunkte, die ein Gateway (GatewayId) nicht in einem Amazon S3 S3-Bucket veröffentlichen konnte. Einheit: Anzahl Abmessungen: GatewayId
<code>IoTSiteWisePublisher.PublisherToS3SuccessCount</code>	Die Anzahl der Datenpunkte, die ein Gateway (GatewayId) erfolgreich in einem Amazon S3 S3-Bucket veröffentlicht hat. Einheit: Anzahl Abmessungen: GatewayId

OPC UA-Collector-Metriken

Metrik	Beschreibung
<code>OpcUaCollector.ActiveDataStreamCount</code>	Die Anzahl der Datenströme, die ein SiteWise Edge-Gateway (gatewayId) für eine OPC UA-Quelle () abonniert hat. sourceName Einheit: Anzahl Abmessungen: GatewayId,, SourceName PropertyGroup
<code>OpcUaCollector.ComponentBuildVersion</code> (nicht verfügbar für Classic-Streams, V2-Gateways)	Diese Metrik gibt die Build-Version der IoT SiteWise OPC UA-Collector-Komponente an, die auf dem Gateway ausgeführt wird. Ein Wert von 1 bedeutet, dass auf dem Gateway eine Version des Collectors ausgeführt wird, die der Dimension entspricht. ComponentBuildVersion

Metrik	Beschreibung
	Einheit: 1 Abmessungen: GatewayId, Component BuildVersion
<code>OpcUaCollector.ConversionErrors</code>	Die Anzahl der Datenpunkte, die ein SiteWise Edge-Gateway (gatewayId) für eine OPC UA-Quelle (sourceName) empfangen hat, was zu Konvertierungsfehlern beim Senden der Daten führte. AWS IoT SiteWise Diese Datenpunkte werden nicht von OPC UA Collector aufgenommen. Einheit: Anzahl Abmessungen: GatewayId SourceName
<code>OpcUaCollector.Heartbeat</code>	Wird jede Minute für jede OPC UA-Quelle (sourceName) generiert, die mit einem SiteWise Edge-Gateway (gatewayId) verbunden ist. Einheit: Anzahl (1 steht für die Verbindung der Quelle und 0 für die Unterbrechung der Quelle.) Abmessungen: GatewayId, SourceName
<code>OpcUaCollector.IncomingValuesCount</code>	Die Anzahl der pro Minute generierten Datenpunkte, die ein SiteWise Edge-Gateway (gatewayId) für eine OPC UA-Quelle (sourceName) empfangen hat. Einheit: Anzahl Abmessungen: GatewayId,, SourceName PropertyGroup

Metrik	Beschreibung
<code>OpcUaCollector.IncomingValueErrors</code>	Die Anzahl der Datenpunkte, die ein SiteWise Edge-Gateway (gatewayId) von einer OPC UA-Quelle (sourceName) empfängt und bei denen es sich nicht um gültige Werte handelt. Diese Datenpunkte werden nicht vom OPC UA Collector aufgenommen, der jede Minute generiert wird. Einheit: Anzahl Abmessungen: GatewayId,, SourceName PropertyGroup
<code>OpcUaCollector.IsConnectedToMqttBroker</code> (nicht verfügbar für Classic-Streams, V2-Gateways)	Wird jede Minute von der SiteWise IoT-OPC-UA-Kollektorkomponente im SiteWise Edge-Gateway generiert. Einheit: 1 (1 steht für die SiteWise IoT-OPC-UA-Kollektorkomponente, ist mit einem MQTT-Broker verbunden) Abmessungen: GatewayId
<code>OpcUaCollector.MqttMessagesDroppedCount</code> (nicht verfügbar für Classic-Streams, V2-Gateways)	Die Anzahl der MQTT-Nachrichten, die von der SiteWise IoT-OPC-UA-Kollektorkomponente gelöscht wurden. Einheit: Anzahl Abmessungen:, GatewayId SourceName

Metrik	Beschreibung
<code>OpcUaCollector.MqttMessagesPublishedBytes</code> (nicht verfügbar für Classic-Streams, V2-Gateways)	Die Anzahl der Byte von MQTT-Nachrichtendaten, die erfolgreich von der IoT SiteWise OPC UA-Collector-Komponente an den MQTT-Broker veröffentlicht wurden. Einheit: Anzahl Abmessungen: GatewayId SourceName
<code>OpcUaCollector.MqttMessagesPublishedCount</code> (nicht verfügbar für Classic-Streams, V2-Gateways)	Die Anzahl der MQTT-Nachrichten, die erfolgreich von der IoT SiteWise OPC UA-Collector-Komponente an den MQTT-Broker veröffentlicht wurden. Einheit: Anzahl Abmessungen: GatewayId SourceName
<code>OpcUaCollector.NullValueCount</code> (nicht verfügbar für Classic-Streams, V2-Gateways)	Die Anzahl der Nullwerte, die von der IoT SiteWise OPC UA-Kollektorkomponente vom OPC UA-Server empfangen wurden. Einheit: Anzahl Abmessungen: GatewayId, SourceName PropertyGroup
<code>OpcUaCollector.NumberOfUniqueMqttTopicsPublished</code> (nicht verfügbar für Classic-Streams, V2-Gateways)	Die Anzahl der eindeutigen MQTT-Themen, die vom IoT SiteWise OPC UA-Collector an den MQTT-Broker veröffentlicht wurden. Einheit: Anzahl Abmessungen: GatewayId SourceName

AWS IoT SiteWise Prozessor-Metriken

Metrik	Beschreibung
<code>Gateway.DataProcessor.IngestionThrottled</code> (nicht verfügbar auf MQTT-fähigen V3-Gateways)	Die Anzahl der pro Minute generierten Datenpunkte, die gedrosselt wurden. Einheit: Anzahl Abmessungen: ThrottledAt
<code>Gateway.DataProcessor.MeasurementRejected</code> (nicht verfügbar auf MQTT-fähigen V3-Gateways)	Die Anzahl der verworfenen Messungen, generiert pro Minute. Einheit: Anzahl Abmessungen: Grund
<code>Gateway.DataProcessor.MessagesRemaining</code> (nicht verfügbar auf MQTT-fähigen V3-Gateways)	Die Anzahl der in einem Stream verbleibenden Nachrichten, die jede Minute generiert werden. Einheit: Anzahl Abmessungen: StreamName
<code>Gateway.DataProcessor.ProcessingError</code> (nicht verfügbar auf MQTT-fähigen V3-Gateways)	Die Anzahl der Verarbeitungsfehler, die jede Minute generiert werden. Einheit: Anzahl Dimensionen: Grund

AWS IoT SiteWise API-Aufrufe protokollieren mit AWS CloudTrail

AWS IoT SiteWise ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in ausgeführt wurden AWS IoT SiteWise. CloudTrail erfasst API-Aufrufe AWS IoT SiteWise als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der AWS IoT SiteWise Konsole und Codeaufrufen für die AWS IoT SiteWise API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Übermittlung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich

Ereignissen für AWS IoT SiteWise. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie die Anfrage ermitteln CloudTrail, an die die Anfrage gestellt wurde AWS IoT SiteWise, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details.

Weitere Informationen zu CloudTrail finden Sie im [AWS CloudTrail Benutzerhandbuch](#).

AWS IoT SiteWise Informationen in CloudTrail

CloudTrail ist auf Ihrem AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn unterstützte Ereignisaktivitäten in auftreten AWS IoT SiteWise, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen im CloudTrail Ereignisverlauf in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem AWS Konto ansehen, suchen und herunterladen. Weitere Informationen finden Sie unter [Ereignisse mit CloudTrail Ereignisverlauf anzeigen](#).

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS Konto, einschließlich der Ereignisse für AWS IoT SiteWise, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS -Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie hier:

- [Übersicht zum Erstellen eines Trails](#)
- [CloudTrail unterstützte Dienste und Integrationen](#)
- [Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail Protokolldateien aus mehreren Regionen](#) und [Empfangen von CloudTrail Protokolldateien von mehreren Konten](#)

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 Root- oder AWS Identity and Access Management (IAM-) Benutzeranmeldedaten gestellt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.

- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie unter [CloudTrail -Element userIdentity](#).

AWS IoT SiteWise 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 AWS IoT SiteWise Ressourcentypen mithilfe der CloudTrail Konsole oder CloudTrail API-Operationen protokollieren. AWS CLI Die [Tabelle](#) in diesem Abschnitt zeigt die verfügbaren Ressourcentypen für AWS IoT SiteWise.

- Um Datenereignisse mithilfe der CloudTrail Konsole zu protokollieren, erstellen Sie einen [Trail](#) - oder [Ereignisdatenspeicher, um Datenereignisse](#) zu protokollieren, oder [aktualisieren Sie einen vorhandenen Trail- oder Ereignisdatenspeicher, um Datenereignisse](#) zu protokollieren.
 1. Wählen Sie Datenereignisse aus, um Datenereignisse zu protokollieren.
 2. Wählen Sie aus der Liste Datenereignistyp den Ressourcentyp aus, für den Sie Datenereignisse protokollieren möchten.
 3. Wählen Sie die Protokollauswahlvorlage aus, die Sie verwenden möchten. Sie können alle Datenereignisse für den Ressourcentyp protokollieren, alle `readOnly` Ereignisse protokollieren, alle `writeOnly` Ereignisse protokollieren oder eine benutzerdefinierte Protokollauswahlvorlage erstellen, um nach den Feldern `readOnlyeventName`, und `resources.ARN` zu filtern.
- Um Datenereignisse mithilfe von zu protokollieren AWS CLI, konfigurieren Sie den `--advanced-event-selectors` Parameter so, dass das `eventCategory` Feld dem Ressourcentypwert entspricht Data und das `resources.type` Feld dem Ressourcentypwert entspricht (siehe [Tabelle](#)). Sie können Bedingungen hinzufügen, um nach den Werten der `resources.ARN` Felder `readOnlyeventName`, und zu filtern.

- Führen Sie den [AWS CloudTrail put-event-selectors](#) Befehl aus, um einen Trail zum Protokollieren von Datenereignissen zu konfigurieren. Weitere Informationen finden Sie unter [Protokollieren von Datenereignissen für Trails mit dem AWS CLI](#).
- Um einen Ereignisdatenspeicher für die Protokollierung von Datenereignissen zu konfigurieren, führen Sie den [AWS CloudTrail create-event-data-store](#) Befehl aus, um einen neuen Ereignisdatenspeicher zum Protokollieren von Datenereignissen zu erstellen, oder führen Sie den [AWS CloudTrail update-event-data-store](#) Befehl aus, um einen vorhandenen Ereignisdatenspeicher zu aktualisieren. Weitere Informationen finden Sie unter [Protokollieren von Datenereignissen für Ereignisdatenspeicher mit dem AWS CLI](#).

In der folgenden Tabelle sind die AWS IoT SiteWise Ressourcentypen aufgeführt. In der Spalte Datenereignistyp (Konsole) wird der Wert angezeigt, der aus der Liste Datenereignistyp auf der CloudTrail Konsole ausgewählt werden kann. 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 APIs Protokolierte 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 auf*
AWS IoT SiteWise Komponente	AWS::IoTSiteWise::Asset	• BatchPutAssetPropertyValues • GetAssetPropertyValue • GetAssetPropertyValueHistory • GetAssetPropertyAggregates • GetInterpolatedAssetPropertyValues • BatchGetAssetPropertyValues • BatchGetAssetPropertyValuesHistory

Typ des Datenereignisses (Konsole)	resources.type-Wert	Daten, die APIs protokolliert wurden CloudTrail auf*
		• BatchGetAssetPropertyAggregates
AWS IoT SiteWise Zeitreihen	AWS::IoTSiteWise::TimeSeries	• BatchPutAssetPropertyValue • GetAssetPropertyValue • GetAssetPropertyValueHistory • GetAssetPropertyAggregates • GetInterpolatedAssetPropertyValues • BatchGetAssetPropertyValue • BatchGetAssetPropertyValueHistory • BatchGetAssetPropertyAggregates
AWS IoT SiteWise Assistentin	AWS::SitewiseAssistant::Conversation	• InvokeAssistant

 Note

Der im Cloudtrail-Ereignis protokollierte resources.type hängt von der in der API-Anfrage verwendeten Kennung ab. Wenn in der Anfrage eine Asset-ID angegeben ist, wird der Asset resources.type protokolliert, andernfalls wird der resources.type protokolliert. TimeSeries

*Sie können erweiterte Event-Selektoren so konfigurieren, dass sie nach den resources.ARN Feldern, und filtern eventNameReadOnly, sodass nur die Ereignisse protokolliert werden, die für Sie wichtig sind. Weitere Informationen zu diesen Feldern finden Sie unter [AdvancedFieldSelector](#).

AWS IoT SiteWise Verwaltungsereignisse in CloudTrail

Die [Protokollierung von Verwaltungsereignissen](#) liefert Informationen über Verwaltungsvorgänge, die mit Ressourcen in Ihrem AWS Konto ausgeführt werden. Sie werden auch als Vorgänge auf Steuerebene bezeichnet. In der Standardeinstellung werden Verwaltungsereignisse CloudTrail protokolliert.

AWS IoT SiteWise protokolliert alle Operationen auf der AWS IoT SiteWise Steuerungsebene als Verwaltungsereignisse. Eine Liste der Vorgänge auf der AWS IoT SiteWise Steuerungsebene, bei denen die AWS IoT SiteWise Anmeldung erfolgt CloudTrail, finden Sie in der [AWS IoT SiteWise API-Referenz](#).

Beispiel: Einträge in AWS IoT SiteWise Protokolldateien

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über den angeforderten Vorgang, Datum und Uhrzeit des Vorgangs, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der den CreateAsset Vorgang demonstriert.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/Administrator",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Administrator",
    "sessionContext": {
      "sessionIssuer": {},
      "webIdFederationData": {},
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2020-03-11T17:26:40Z"
      }
    }
  },
```

```
    "invokedBy": "signin.amazonaws.com"
  },
  "eventTime": "2020-03-11T18:01:22Z",
  "eventSource": "iotsitewise.amazonaws.com",
  "eventName": "CreateAsset",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "signin.amazonaws.com",
  "requestParameters": {
    "assetName": "Wind Turbine 1",
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-00000EXAMPLE"
  },
  "responseElements": {
    "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetArn": "arn:aws:iotsitewise:us-east-1:123456789012:asset/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetStatus": {
      "state": "CREATING"
    }
  },
  "requestID": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "eventID": "a1b2c3d4-5678-90ab-cdef-bbbbbbEXAMPLE",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```

Kennzeichnen Sie Ihre AWS IoT SiteWise Ressourcen

Das Markieren Ihrer AWS IoT SiteWise Ressourcen bietet eine leistungsstarke Möglichkeit, Unternehmensressourcen effizient zu kategorisieren, zu verwalten und abzurufen. Durch die Zuweisung von Tags, die aus Schlüssel-Wert-Paaren bestehen, können Sie Ihren Ressourcen beschreibende Metadaten hinzufügen. Die Metadaten aus Tags können zur Optimierung von Vorgängen verwendet werden. In einem Windpark-Szenario ermöglichen es Ihnen beispielsweise Tags, Turbinen mit bestimmten Attributen wie Standort, Kapazität und Betriebsstatus zu kennzeichnen, was eine schnelle Identifizierung und Verwaltung innerhalb AWS IoT SiteWise der Anlage ermöglicht.

Die Integration von Tags in AWS Identity and Access Management (IAM-) Richtlinien verbessert die Sicherheit und die Betriebskontrolle, indem Regeln für den bedingten Zugriff definiert werden. Das bedeutet, dass Sie angeben können, dass nur Benutzer mit bestimmten Tags angemeldet sind. Beispielsweise können nur Personen, die mit einer bestimmten Rolle oder Abteilung gekennzeichnet sind, auf bestimmte Ressourcen zugreifen oder diese ändern.

Verwenden Sie Tags in AWS IoT SiteWise

Verwenden Sie Tags, um Ihre AWS IoT SiteWise Ressourcen nach Zweck, Eigentümer, Umgebung oder einer anderen Klassifizierung für Ihren Anwendungsfall zu kategorisieren. Wenn es viele Ressourcen desselben Typs gibt, können Sie eine bestimmte Ressourcen schnell basierend auf ihren Tags identifizieren.

Jedes Tag besteht aus einem Schlüssel und einem optionalen Wert, den Sie angeben. Sie können beispielsweise eine Reihe von Tags für Ihre Anlagenmodelle einrichten, um sie entsprechend den industriellen Prozessen, die sie unterstützen, nachzuverfolgen. Es wird empfohlen, für jeden von Ihnen verwalteten Ressourcentyp einen maßgeschneiderten Satz von Tag-Schlüsseln zu entwickeln. Die Verwendung eines konsistenten Satzes von Tag-Schlüsseln kann die Verwaltung von Ressourcen erleichtern.

Tag mit dem AWS-Managementkonsole

Der Tag-Editor im AWS-Managementkonsole bietet Ihnen eine zentrale, einheitliche Möglichkeit, Ihre Tags für Ressourcen aus allen AWS Diensten zu erstellen und zu verwalten. Weitere Informationen finden Sie unter [Erste Schritte mit dem Tag-Editor](#) im Benutzerhandbuch Tagging AWS Resources and Tag Editor.

Taggen Sie mit der API AWS IoT SiteWise

Die AWS IoT SiteWise API verwendet auch Tags. Beachten Sie vor dem Erstellen von Tags Beschränkungen für Tags. Weitere Informationen finden Sie unter [Konventionen für Benennung und Nutzung von Tags](#) in der Allgemeine AWS-Referenz.

- Wenn Sie bei der Erstellung einer Ressource Tags hinzufügen möchten, definieren Sie diese in der Eigenschaft `tags` der Ressource.
- Verwenden Sie den [TagResource](#)Vorgang, um einer vorhandenen Ressource Tags hinzuzufügen oder Tag-Werte zu aktualisieren.
- Verwenden Sie den [UntagResource](#)Vorgang, um Tags aus einer Ressource zu entfernen.
- Um die mit einer Ressource verknüpften Tags abzurufen, verwenden Sie die [ListTagsForResource](#)Operation oder beschreiben Sie die Ressource und überprüfen Sie ihre `tags` Eigenschaften.

In der folgenden Tabelle sind Ressourcen aufgeführt, die Sie mithilfe der AWS IoT SiteWise API taggen können, sowie die entsprechenden Create Describe AND-Operationen.

Taggierbare Ressourcen AWS IoT SiteWise

Ressource	Operation erstellen	Operation beschreiben
Anlagenmodell oder Komponentenmodell	CreateAssetModel	DescribeAssetModel
Komponente	CreateAsset	DescribeAsset
SiteWise Edge-Gateway	CreateGateway	DescribeGateway
Portal	CreatePortal	DescribePortal
Projekt	CreateProject	DescribeProject
Dashboard	CreateDashboard	DescribeDashboard
Zugriffsrichtlinie	CreateAccessPolicy	DescribeAccessPolicy
Zeitreihen	BatchPutAssetPropertyValue	DescribeTimeSeries

Denn Sie können Ihre Datenquellen so konfigurieren [BatchPutAssetPropertyValue](#), dass Industriedaten an AWS IoT SiteWise gesendet werden, bevor Sie Anlagenmodelle und Anlagen erstellen. AWS IoT SiteWise erstellt automatisch Datenströme, um Rohdatenströme von Ihren Geräten zu empfangen. Weitere Informationen finden Sie unter [Verwaltung der Datenaufnahme](#).

Mit den folgenden Operationen können Sie Tags für Ressourcen anzeigen und verwalten, die die Markierung mit Tags unterstützen:

- [TagResource](#)— Fügt einer Ressource Tags hinzu oder aktualisiert den Wert eines vorhandenen Tags.
- [ListTagsForResource](#)— Listet die Tags für eine Ressource auf.
- [UntagResource](#)— Entfernt Tags aus einer Ressource.

Sie können jederzeit Tags zu einer Ressource hinzufügen oder daraus entfernen. Um den Wert eines vorhandenen Tag-Schlüssels zu aktualisieren, fügen Sie der Ressource ein neues Tag mit demselben Schlüssel und dem gewünschten neuen Wert hinzu. Diese Aktion ersetzt den alten Wert durch den neuen. Es ist zwar möglich, eine leere Zeichenfolge als Tag-Wert zuzuweisen, aber Sie können keinen Nullwert zuweisen.

Durch das Löschen einer Ressource werden auch alle damit verknüpften Tags entfernt.

Verwenden Sie Tags mit IAM-Richtlinien

Verwenden Sie Ressourcen-Tags in Ihren IAM-Richtlinien, um den Benutzerzugriff und die Benutzerberechtigungen zu kontrollieren. Richtlinien können es Benutzern beispielsweise ermöglichen, nur Ressourcen zu erstellen, denen ein bestimmtes Tag angehängt ist. Richtlinien können auch verhindern, dass Benutzer Ressourcen mit bestimmten Tags erstellen oder ändern.

Note

Wenn Sie Tags verwenden, um den Zugriff von Benutzern auf Ressourcen zuzulassen oder abzulehnen, sollten Sie Benutzern nicht die Möglichkeit geben, diese Tags diesen Ressourcen hinzuzufügen oder aus diesen Ressourcen zu entfernen. Andernfalls könnte ein Benutzer Ihre Einschränkungen umgehen und Zugriff auf eine Ressource erhalten, indem er deren Tags ändert.

Sie können im Element Condition (auch als Condition-Block bezeichnet) einer Richtlinienanweisung die folgenden Bedingungskontextschlüssel und -werte verwenden.

`aws:ResourceTag/tag-key: tag-value`

Sie können mithilfe bestimmter Tags Aktionen für Ressourcen zulassen oder ablehnen.

`aws:RequestTag/tag-key: tag-value`

Erfordert, dass beim Erstellen oder Ändern einer markierbaren Ressource ein bestimmtes Tag verwendet (oder nicht verwendet) wird.

`aws:TagKeys: [tag-key, ...]`

Erfordert, dass beim Erstellen oder Ändern einer markierbaren Ressource ein bestimmter Satz von Tag-Schlüsseln verwendet (oder nicht verwendet) wird.

Note

Die Bedingungskontextschlüssel und -werte in einer IAM-Richtlinie gelten nur für Aktionen, für die eine Ressource mit Tags als erforderlichem Parameter angegeben werden kann. Sie können beispielsweise den tagbasierten bedingten Zugriff für einrichten. [ListAssets](#) Sie können den tagbasierten bedingten Zugriff nicht aktivieren, [PutLoggingOptions](#) da in der Anfrage auf keine markierbare Ressource verwiesen wird.

Weitere Informationen finden Sie unter [Steuern des Zugriffs auf AWS Ressourcen mithilfe von Ressourcen-Tags](#) und [IAM-JSON-Richtlinienreferenz](#) im IAM-Benutzerhandbuch.

Beispiel für IAM-Richtlinien mit Tags

- [AnzeigenAWS IoT SiteWiseauf Tags basierende Vermögenswerte](#)

Problembehebung AWS IoT SiteWise

Verwenden Sie die folgenden Informationen, um Probleme mit zu beheben AWS IoT SiteWise.

Themen

- [Fehlerbehebung bei einem SiteWise Edge-Gateway](#)
- [Fehler bei einem AWS IoT SiteWise Portal beheben](#)
- [Problembehandlung bei einer AWS IoT SiteWise Regelaktion](#)
- [Problembehandlung bei Massenimport- und -exportvorgängen](#)

Fehlerbehebung bei einem SiteWise Edge-Gateway

Beheben Sie häufig auftretende Probleme mit dem AWS IoT SiteWise Edge-Gateway, indem Sie sich mit den entsprechenden Themen befassen.

Sie können sich auch CloudWatch Messwerte ansehen, die von Ihren SiteWise Edge-Gateways gemeldet wurden, um Probleme mit Konnektivität oder Datenströmen zu beheben. Weitere Informationen finden Sie unter [Überwachen Sie AWS IoT SiteWise mit CloudWatch Amazon-Metriken](#).

Themen

- [SiteWise Edge-Gateway-Protokolle konfigurieren und darauf zugreifen](#)
- [Behebung von Problemen mit dem SiteWise Edge-Gateway](#)
- [Fehlerbehebung bei der AWS IoT SiteWise Edge-Anwendung auf Siemens Industrial Edge](#)
- [Fehlerbehebung bei Open-Source-Integrationen am Edge](#)
- [Probleme beheben AWS IoT Greengrass](#)

SiteWise Edge-Gateway-Protokolle konfigurieren und darauf zugreifen

Bevor Sie SiteWise Edge-Gateway-Protokolle anzeigen können, müssen Sie Ihr SiteWise Edge-Gateway so konfigurieren, dass es CloudWatch Protokolle an Amazon Logs sendet oder Protokolle im lokalen Dateisystem speichert.

- Verwenden Sie CloudWatch Logs, wenn Sie das verwenden möchten AWS-Managementkonsole , um die Protokolldateien Ihres SiteWise Edge-Gateways einzusehen. Weitere Informationen finden Sie unter [Verwenden Sie Amazon CloudWatch Logs](#).
- Verwenden Sie lokale Dateisystemprotokolle, wenn Sie die Befehlszeile oder lokale Software verwenden möchten, um die Protokolldateien Ihres SiteWise Edge-Gateways anzuzeigen. Weitere Informationen finden Sie unter [Loggen Sie sich in den Dienst ein AWS IoT SiteWise](#).

Behebung von Problemen mit dem SiteWise Edge-Gateway

Verwenden Sie die folgenden Informationen, um Probleme mit dem SiteWise Edge-Gateway zu beheben.

Problembereiche

- [Pakete können nicht für SiteWise Edge-Gateways bereitgestellt werden](#)
- [AWS IoT SiteWise empfängt keine Daten von OPC UA-Servern](#)
- [Im Dashboard werden keine Daten angezeigt](#)
- [„Hauptklasse konnte nicht gefunden oder geladen werden“ wird in aws.iot angezeigt. SiteWiseEdgePublisher meldet sich bei einem Fehler von//an greengrass/v2/logs](#)
- [Ich sehe 'SESSION_TAKEN_OVER' oder 'com.aws.greengrass.mqttclient. MqttClient: Die Nachricht konnte nicht über Spooler veröffentlicht werden und es wird erneut versucht. ' in den Protokollen](#)
- [Ich sehe „com.aws.greengrass.deployment“. lotJobsHelper: Es wurde kein Bereitstellungsjob gefunden. ' oder 'Das Bereitstellungsergebnis wurde bereits gemeldet. ' in den Protokollen](#)
- [Beim Versuch, die Zeitstempелеinstellung in einer Eigenschaftsgruppe einer OPC UA-Datenquelle zu konfigurieren, wird der Status „SYNC_FAILED“ angezeigt](#)
- [Konvertierte Datentypen sind nicht enthalten](#)
- [Probleme mit dem Trust Store](#)
- [Probleme bei der Installation mit einem Proxy](#)

Pakete können nicht für SiteWise Edge-Gateways bereitgestellt werden

Wenn die AWS IoT Greengrass Nucleus-Komponente (`aws.greengrass.Nucleus`) veraltet ist, können Sie möglicherweise keine Packs auf Ihrem SiteWise Edge-Gateway bereitstellen. Sie können

die AWS IoT Greengrass V2 Konsole verwenden, um die AWS IoT Greengrass Nucleus-Komponente zu aktualisieren.

Um die AWS IoT Greengrass Nucleus-Komponente (Konsole) zu aktualisieren

1. Navigieren Sie zur [AWS IoT Greengrass -Konsole](#).
2. Wählen Sie im Navigationsbereich unter AWS IoT GreengrassDeployments aus.
3. Wählen Sie in der Liste Bereitstellungen die Bereitstellung aus, die Sie überarbeiten möchten.
4. Wählen Sie Überarbeiten aus.
5. Wählen Sie auf der Seite „Ziel angeben“ die Option Weiter.
6. Geben Sie auf der Seite Komponenten auswählen unter Öffentliche Komponenten in das Suchfeld **aws.greengrass.Nucleus** AWS.Greengrass.Nucleus ein und wählen Sie dann aus.
7. Wählen Sie Weiter aus.
8. Wählen Sie auf der Seite Komponenten konfigurieren die Option Weiter aus.
9. Wählen Sie auf der Seite Erweiterte Einstellungen konfigurieren die Option Weiter aus.
10. Wählen Sie auf der Seite Review (Prüfen) die Option Deploy (Bereitstellen) aus.

AWS IoT SiteWise empfängt keine Daten von OPC UA-Servern

Wenn Ihre Geräte AWS IoT SiteWise keine von Ihren OPC UA-Servern gesendeten Daten empfangen, können Sie die Protokolle Ihres SiteWise Edge-Gateways durchsuchen, um Probleme zu beheben. Suchen Sie nach `swPublisher` Protokollen auf Informationsebene, die die folgende Meldung enthalten.

```
Emitting diagnostic name=PublishError.SomeException
```

Verwenden Sie je nach Typ *SomeException* im Protokoll die folgenden Ausnahmetypen und die entsprechenden Probleme, um Ihr SiteWise Edge-Gateway zu beheben:

- `ResourceNotFoundException`— Ihre OPC UA-Server senden Daten, die keinem Eigenschaftsalias für ein Asset entsprechen. Diese Ausnahme kann in zwei Fällen auftreten:
 - Ihre Eigenschaftsalias stimmen nicht genau mit Ihren OPC UA-Variablen überein, einschließlich der von Ihnen definierten Quellpräfixe. Überprüfen Sie, ob Ihre Eigenschaftsalias und Quellpräfixe korrekt sind.

- Sie haben Ihre OPC UA-Variablen nicht den Eigenschaften von Vermögenswerten zugeordnet. Weitere Informationen finden Sie unter [Datenströme verwalten für AWS IoT SiteWise](#).

Wenn Sie bereits alle gewünschten OPC-UA-Variablen zugeordnet haben, können Sie filtern AWS IoT SiteWise, welche OPC-UA-Variablen das Edge-Gateway sendet. SiteWise Weitere Informationen finden Sie unter [Verwenden Sie OPC UA-Knotenfilter in Edge SiteWise](#).

- `InvalidRequestException`— Die Datentypen Ihrer OPC UA-Variablen stimmen nicht mit den Datentypen Ihrer Anlageneigenschaft überein. Wenn eine OPC UA-Variable beispielsweise einen Integer-Datentyp hat, muss Ihre entsprechende Asset-Eigenschaft vom Datentyp Integer sein. Eine Asset-Eigenschaft vom Typ Double kann keine ganzzahligen OPC UA-Werte empfangen. Um dieses Problem zu beheben, definieren Sie neue Eigenschaften mit den richtigen Datentypen.
- `TimestampOutOfRangeException`— Ihr SiteWise Edge-Gateway sendet Daten, die außerhalb des zulässigen Bereichs liegen. AWS IoT SiteWise lehnt alle Datenpunkte ab, deren Zeitstempel vor 7 Tagen in der Vergangenheit oder weniger als 5 Minuten in der future liegen. Wenn Ihr SiteWise Edge-Gateway die Stromversorgung oder die Verbindung zur AWS Cloud verloren hat, müssen Sie möglicherweise den Cache Ihres SiteWise Edge-Gateways leeren.
- `ThrottlingException` oder `LimitExceededException`— Ihre Anfrage hat ein AWS IoT SiteWise Servicekontingent überschritten, z. B. die Rate der aufgenommenen Datenpunkte oder die Anforderungsrate für API-Operationen mit Objektdaten. Überprüfen Sie, dass Ihre Konfiguration [AWS IoT SiteWise Kontingente](#) nicht überschreitet.

Im Dashboard werden keine Daten angezeigt

Wenn in Ihrem Dashboard keine Daten angezeigt werden, sind die Publisher-Konfiguration und die Datenquelle des SiteWise Edge-Gateways möglicherweise nicht synchron. Wenn sie nicht synchron sind, kann die Aktualisierung des Namens der Datenquelle die Synchronisierung von der Cloud zum Edge beschleunigen und so den Fehler „Nicht synchron“ beheben.

Um den Namen einer Datenquelle zu aktualisieren

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie das SiteWise Edge-Gateway aus, das mit dem Dashboard verbunden ist.
4. Wählen Sie unter Datenquellen die Option Bearbeiten aus.
5. Wählen Sie einen neuen Quellennamen und klicken Sie auf Speichern, um Ihre Änderung zu bestätigen.

- Überprüfen Sie Ihre Änderungen, indem Sie in der Tabelle Datenquellen überprüfen, ob der Datenquellenname aktualisiert wurde.

„Hauptklasse konnte nicht gefunden oder geladen werden“ wird in `aws.iot` angezeigt. `SiteWiseEdgePublisher` meldet sich bei einem Fehler von `//an greengrass/v2/logs`

Wenn Sie diesen Fehler sehen, müssen Sie möglicherweise die Java-Version Ihres SiteWise Edge-Gateways aktualisieren.

- Führen Sie von einem Terminal folgenden Befehl aus:

```
java -version
```

Die Version von Java, mit der Ihr SiteWise Edge-Gateway ausgeführt wird, wird unter `OpenJDK Runtime Environment` angezeigt. Sie werden eine Antwort wie die folgende sehen:

```
openjdk version "11.0.20" 2023-07-18 LTS
OpenJDK Runtime Environment Corretto011.0.20.8.1 (build 11.0.20+8-LTS
OpenJDK 64-Bit Server VM Corretto-11.0.20.8.1 (build 11.0.20+8-LTS, mixed node)
```

Wenn Sie die Java-Version 11.0.20.8.1 ausführen, müssen Sie das IoT SiteWise Publisher-Paket auf Version 2.4.1 oder neuer aktualisieren. Nur die Java-Version 11.0.20.8.1 ist betroffen. Umgebungen mit anderen Java-Versionen können weiterhin ältere Versionen der IoT SiteWise Publisher-Komponente verwenden. Weitere Informationen zum Aktualisieren eines Komponentenpakets finden Sie unter [Ändern Sie die Version der SiteWise Edge Gateway-Komponentenpakete](#)

Ich sehe 'SESSION_TAKEN_OVER' oder 'com.aws.greengrass.mqttclient.MqttClient: Die Nachricht konnte nicht über Spooler veröffentlicht werden und es wird erneut versucht.' in den Protokollen

Wenn Sie unter eine Warnung `SESSION_TAKEN_OVER` oder einen Fehler `com.aws.greengrass.mqttclient.MqttClient: Failed to publish the message via Spooler and will retry.` in Ihren Protokollen sehen, versuchen Sie möglicherweise `//greengrass/v2/logs/greengrass.log`, dieselbe Konfigurationsdatei für mehrere SiteWise Edge-Gateways auf mehreren Geräten zu verwenden. Jedes SiteWise Edge-Gateway benötigt eine eigene Konfigurationsdatei, um eine Verbindung zu Ihrem AWS Konto herzustellen.

Ich sehe „com.aws.greengrass.deployment“. IotJobsHelper: Es wurde kein Bereitstellungsjob gefunden. ' oder 'Das Bereitstellungsergebnis wurde bereits gemeldet. ' in den Protokollen

Wenn Sie `com.aws.greengrass.deployment.IotJobsHelper: No deployment job found.` oder `Deployment result already reported.` in Ihren Protokollen unter `sehen/greengrass/v2/logs/greengrass.log`, versuchen Sie möglicherweise, dieselbe Konfigurationsdatei wiederzuverwenden.

Es gibt mehrere Lösungen:

- Wenn Sie die Konfigurationsdatei wiederverwenden möchten, gehen Sie wie folgt vor:
 1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
 2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
 3. Wählen Sie das SiteWise Edge-Gateway aus, das Sie wiederverwenden möchten.
 4. Wählen Sie die Registerkarte Updates.
 5. Wählen Sie eine andere Publisher-Version und anschließend Bereitstellen aus.

Folgen Sie den Schritten unter [Erstellen Sie ein Gateway für Siemens Industrial Edge](#), um eine neue Konfigurationsdatei zu erstellen.

Beim Versuch, die Zeitstempleinstellung in einer Eigenschaftsgruppe einer OPC UA-Datenquelle zu konfigurieren, wird der Status „SYNC_FAILED“ angezeigt

Bei der AWS IoT SiteWise Aktualisierung der OPC UA-Collector-Komponente für AWS IoT Greengrass in Version 2.5.0 haben wir eine neue Zeitstempel-Konfigurationsoption eingeführt. Sie können den Zeitstempel entweder von Ihrem Gerät oder den Zeitstempel vom Server verwenden. Ältere Versionen der OPC UA-Collector-Komponente unterstützen diese Option nicht und können nicht synchronisiert werden.

Es gibt zwei Möglichkeiten, den Status einer fehlgeschlagenen Datenquellensynchronisierung zu beheben. Es wird empfohlen, die IoT SiteWise OPC UA-Collector-Komponente auf Version 2.5.0 oder höher zu aktualisieren. Alternativ können Sie weiterhin die ältere Version der OPC UA-Collector-Komponente verwenden, wenn Sie den Zeitstempel auf setzen. Source Informationen zum Upgrade der SiteWise IoT-OPC-UA-Kollektorkomponente finden Sie unter [Aktualisieren Sie die Version einer](#)

[Komponente AWS IoT SiteWise](#). Wir empfehlen, die neuesten Versionen aller Komponenten zu verwenden.

 Note

Es gibt keine Datenunterbrechung, wenn der Synchronisierungsstatus einer Datenquelle fehlschlägt. Die Quelldaten fließen weiterhin in AWS IoT SiteWise. Die Konfiguration wird einfach nicht mit der IoT SiteWise OPC UA-Collector-Komponente in Ihrer AWS IoT Greengrass V2 Bereitstellung synchronisiert.

Um die Zeitstempelkonfiguration für eine Eigenschaftsgruppe zu ändern

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.
3. Wählen Sie das Gateway aus, das bearbeitet werden soll.
4. Wählen Sie im Abschnitt Datenquellen die Datenquelle aus, deren Synchronisierungsstatus fehlgeschlagen ist, und klicken Sie auf Bearbeiten.
5. Erweitern Sie Erweiterte Konfiguration und dann Gruppeneinstellungen.
6. Wählen Sie unter Timestamp die Option Quelle aus. Wenn Sie Source auswählen, wird die `timestampToReturn` Eigenschaft aus der Konfiguration entfernt. Diese Einstellung ermöglicht standardmäßig die Erfassung des Datenquellen-Zeitstempels von Ihrem Gerät, sodass die Datenquelle mit der IoT SiteWise OPC UA-Collector-Komponente synchronisiert werden kann.
7. Wählen Sie Speichern.

Konvertierte Datentypen sind nicht enthalten

Wenn bei der Konvertierung von nicht unterstützten OPC UA-Datentypen in Zeichenketten ein Fehler auftritt AWS IoT SiteWise, kann das mehrere Gründe haben:

- Der Datentyp, den Sie konvertieren möchten, ist ein komplexer Datentyp. Komplexe Datentypen werden nicht unterstützt.
- Wenn Destinations as AWS IoT SiteWise Buffered mit Amazon S3 verwendet wird, wird der vollständige Zeichenkettenwert in Dateien beibehalten, die in einen Amazon S3 S3-Bucket übertragen werden. Wenn Sie später Daten aufnehmen AWS IoT SiteWise, werden vollständige Zeichenkettenwerte, die länger als 1024 Byte sind, zurückgewiesen.

Probleme mit dem Trust Store

Wenn Sie in SiteWise Edge auf Probleme im Zusammenhang mit Trust Stores stoßen, sollten Sie die folgenden Schritte zur Fehlerbehebung in Betracht ziehen:

- Stellen Sie sicher, dass das AWS IoT Greengrass Root-CA-Zertifikat in den entsprechenden Trust Stores vorhanden und korrekt formatiert ist
- Stellen Sie sicher, dass das KeyStore Java-Passwort korrekt festgelegt ist und SiteWise Edge-Komponenten darauf zugreifen können
- Vergewissern Sie sich, dass alle benutzerdefinierten Zertifikate (z. B. für HTTPS-Proxys) das richtige Format (normalerweise PEM) haben und ordnungsgemäß in die Trust Stores importiert wurden
- Stellen Sie sicher, dass die Trust Stores über die richtigen Dateiberechtigungen verfügen und für die Edge-Prozesse zugänglich sind SiteWise
- Überprüfen Sie die SiteWise Edge-Protokolle auf SSL/TLS damit verbundene Fehler, die auf Probleme mit dem Trust Store hinweisen können
- Testen Sie SSL/TLS Verbindungen unabhängig voneinander mithilfe von Tools wie `openssl` der Überprüfung der Trust Store-Funktionalität

Probleme bei der Installation mit einem Proxy

Wenn bei der Proxykonfiguration Probleme auftreten, sollten Sie die folgenden Schritte zur Fehlerbehebung in Betracht ziehen:

- Stellen Sie sicher, dass die Proxy-URL korrekt formatiert ist und das richtige Schema (`http://` oder `https://`) enthält
- Stellen Sie sicher, dass alle Proxy-Anmeldeinformationen URL-codiert sind, wenn sie Sonderzeichen enthalten
- Vergewissern Sie sich, dass die Liste ohne Proxy alle erforderlichen lokalen Adressen und Dienstendpunkte enthält AWS
- Stellen Sie bei HTTPS-Proxys sicher, dass das bereitgestellte CA-Zertifikat im PEM-Format vorliegt
- Suchen Sie in den Installationsprotokollen nach spezifischen Fehlermeldungen, die möglicherweise auf die Ursache des Problems hinweisen
- Testen Sie die Proxyverbindung unabhängig voneinander, um sicherzustellen, dass sie ordnungsgemäß funktioniert

Fehlerbehebung bei der AWS IoT SiteWise Edge-Anwendung auf Siemens Industrial Edge

Um Probleme mit der AWS IoT SiteWise Edge-Anwendung auf Ihrem Siemens Industrial Edge Gerät zu beheben, können Sie über die Portale Siemens Industrial Edge Management oder Siemens Industrial Edge Device (IED) auf die Protokolle für die Anwendung zugreifen. Weitere Informationen finden Sie in der Siemens-Dokumentation unter [Protokolle herunterladen](#).

Meine Daten werden nicht angezeigt in AWS IoT SiteWise

- Stellen Sie sicher, dass es keine Probleme mit Ihren Databus Benutzern gibt und dass das Häkchensymbol für Databus_Configuration grün statt grau ist.
- Möglicherweise verwenden Sie keine Version, Siemens Industrial Edge Management die enthält. Secure Storage Aktualisieren Sie Ihre Version von Siemens OS. Weitere Informationen finden Sie unter [Siemens Secure Storage und die AWS IoT SiteWise Edge-Anwendung](#).

Ich sehe „Konfigurationsdatei fehlt AWS_REGION“ in den Protokollen

Wenn Sie `Config file missing AWS_REGION` in den Siemens-Protokollen sehen, dass die JSON-Datei der Konfigurationsdatei beschädigt wurde. Sie müssen eine neue Konfigurationsdatei erstellen. Folgen Sie den Schritten unter [Erstellen Sie ein Gateway für Siemens Industrial Edge](#), um eine neue Konfigurationsdatei zu erstellen.

In der Edge-Gateway-Konfiguration wird die Fehlermeldung „Nicht synchron“ angezeigt

Wenn Sie nach Abschluss der Bereitstellung eine `Out of sync` Fehlermeldung auf Ihrem Siemens Industrial Edge Gateway sehen, bedeutet dies, dass die SiteWise IoT-Publisher-Komponente nicht mit Ihrem Gateway synchronisiert ist. Die SiteWise IoT-Publisher-Komponente arbeitet im Hintergrund auf Siemens Industrial Edge Gateways, um MQTT-Themenfunktionen bereitzustellen. Wir haben Siemens Industrial Edge Gateways so aktualisiert, dass sie den `iotsitewise:publisher:3` Capability-Namespace verwenden und nicht. `iotsitewise:publisher:2` Sie können auf die neueste Version des Herausgebers aktualisieren, um dieses Problem zu beheben.

Um auf die neueste Version des SiteWise IoT-Publishers zu aktualisieren

1. Navigieren Sie zur [AWS IoT SiteWise -Konsole](#).
2. Wählen Sie im Navigationsbereich Edge-Gateways aus.

3. Wählen Sie das Siemens Industrial Edge Gateway aus, das bearbeitet werden soll.
4. Wählen Sie im Abschnitt Edge-Funktionen die Option Softwareversionen anzeigen aus.
5. Wählen Sie im Dropdownmenü SiteWise Publisher die neueste Version des IoT-Publishers aus.
6. Wählen Sie Fertig aus.

Fehlerbehebung bei Open-Source-Integrationen am Edge

Dieser Abschnitt enthält Lösungen für häufig auftretende Probleme, die bei der Integration von Open-Source-Tools in Edge auftreten können. SiteWise

Note

Node-RED®, InfluxDB® und Grafana® sind keine Anbieter oder Lieferanten für Edge. SiteWise

Verbindungsprobleme

Node-RED kann keine Verbindung zum MQTT-Broker herstellen

Stellen Sie sicher, dass der MQTT-Broker auf dem angegebenen Port läuft und darauf zugegriffen werden kann. Überprüfen Sie Ihre Netzwerkkonfiguration und stellen Sie sicher, dass die Broker-Adresse korrekt ist.

Um den Status des MQTT-Brokers zu überprüfen, führen Sie Folgendes aus:

```
docker ps | grep emqx
```

InfluxDB-Verbindungsfehler

Stellen Sie sicher, dass Ihr Authentifizierungstoken gültig ist und dass Sie die richtigen Organisations- und Bucket-Namen angegeben haben. Überprüfen Sie, ob InfluxDB läuft und darauf zugegriffen werden kann.

Um den InfluxDB-Status zu überprüfen, führen Sie Folgendes aus:

```
curl -I http://localhost:8086
```

Grafana kann keine Verbindung zu InfluxDB herstellen

Stellen Sie sicher, dass die InfluxDB-Datenquellenkonfiguration in Grafana korrekt ist, einschließlich URL, Authentifizierungstoken, Organisation und Bucket.

Probleme mit dem Datenfluss

Es erscheinen keine Daten in AWS IoT SiteWise

Vergewissern Sie sich, dass Ihr Eigenschaftsalias im Node-RED-Flow dem erwarteten Format entspricht. Stellen Sie sicher, dass die MQTT-Themenstruktur korrekt ist und dass das SiteWise Edge-Gateway ordnungsgemäß konfiguriert ist, um Daten vom MQTT-Broker zu empfangen.

In InfluxDB sind keine SiteWise Edge-Daten gespeichert

Stellen Sie sicher, dass der Node-RED-Retentionsfluss korrekt konfiguriert ist und dass der InfluxDB-Writer-Knoten über die richtigen Bucket- und Messeinstellungen verfügt. Überprüfen Sie die Node-RED-Debug-Ausgabe auf Fehler.

Datenformatfehler

Stellen Sie sicher, dass Ihre Datentransformationsfunktionen Daten korrekt zwischen Formaten konvertieren. Verwenden Sie die Node-RED-Debug-Knoten, um die Daten in jeder Phase des Datenflusses zu überprüfen.

Leistungsprobleme

Hohe CPU- oder Speicherauslastung

Überwachen Sie die Ressourcennutzung und passen Sie die Konfiguration Ihrer Komponenten nach Bedarf an. Erwägen Sie, die Häufigkeit der Datenerfassung zu reduzieren oder Datenfilterung zu implementieren, um die Verarbeitungslast zu reduzieren.

Um die Ressourcennutzung zu überwachen, führen Sie folgenden Befehl aus:

```
docker stats
```

Langsames Laden des Grafana-Dashboards

Optimieren Sie Ihre InfluxDB-Abfragen und erwägen Sie, Ihren Dashboard-Panels Zeitbeschränkungen hinzuzufügen. Reduzieren Sie die Anzahl der angezeigten Datenpunkte, indem Sie geeignete Aggregationsfunktionen verwenden.

Protokollierung und Diagnose

Um Probleme zu beheben, überprüfen Sie die Protokolle für jede Komponente:

Node-RED-Protokolle

Zeigen Sie die Protokolle in der Node-RED-Konsole an oder führen Sie Folgendes aus:

```
docker logs node-red
```

InfluxDB-Protokolle

Greifen Sie auf Protokolle zu, indem Sie Folgendes ausführen

```
docker logs influxdb
```

Grafana-Protokolle

Logs anzeigen, indem Sie Folgendes ausführen:

```
docker logs grafana
```

SiteWise Edge-Protokolle

Überprüfen Sie die SiteWise Edge-Gateway-Protokolle auf Probleme mit der MQTT-Verbindung und Datenverarbeitung. Weitere Informationen finden Sie unter [Fehlerbehebung bei einem SiteWise Edge-Gateway](#).

Probleme beheben AWS IoT Greengrass

Lösungen für viele Probleme bei der Konfiguration oder Bereitstellung Ihres SiteWise Edge-Gateways finden Sie AWS IoT Greengrass im AWS IoT Greengrass Entwicklerhandbuch unter [Problembehandlung](#). AWS IoT Greengrass

Fehler bei einem AWS IoT SiteWise Portal beheben

Beheben Sie häufig auftretende Probleme mit Ihren AWS IoT SiteWise Portalen.

Benutzer und Administratoren können nicht auf das AWS IoT SiteWise Portal zugreifen

Wenn Benutzer oder Administratoren nicht auf Ihr AWS IoT SiteWise Portal zugreifen können, verfügen Sie möglicherweise über eingeschränkte Berechtigungen in angehängten AWS Identity and Access Management (IAM-) Richtlinien, die erfolgreiche Anmeldungen verhindern.

Sehen Sie sich die folgenden Beispiele für IAM-Richtlinien an, die zu einem Anmeldefehler führen können:

Note

Alle angehängten IAM-Richtlinien, die ein "Condition" Element enthalten, führen zu einem Anmeldefehler.

Beispiel 1: Die Bedingung hier ist eine eingeschränkte IP, und dies führt zu einem Anmeldefehler.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal"
      ],
      "Resource": "*",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "203.0.113.0/24"
          ]
        }
      }
    }
  ]
}
```

```

    }
  ]
}

```

Beispiel 2: Die Bedingung hier ist ein eingeschlossenes Tag, und dies führt zu einem Anmeldefehler.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/project": "*"
        }
      }
    }
  ]
}

```

Vermeiden Sie beim Hinzufügen von Benutzern oder Administratoren zum Portal die Erstellung von IAM-Richtlinien, die Benutzerberechtigungen einschränken, z. B. eingeschränkte IP-Adressen. Alle angehängten Richtlinien mit eingeschränkten Berechtigungen können keine Verbindung zum AWS IoT SiteWise Portal herstellen.

Problembehandlung bei einer AWS IoT SiteWise Regelaktion

Um Probleme mit Ihrer AWS IoT SiteWise Regelaktion in zu beheben AWS IoT Core, können Sie eines der folgenden Verfahren ausführen:

- Amazon CloudWatch Logs konfigurieren
- Konfigurieren einer Fehler-Aktion für die erneute Veröffentlichung für Ihre Regel

Vergleichen Sie anschließend die Fehlermeldungen mit den Fehlern in diesem Thema, um Ihr Problem zu beheben.

Themen

- [AWS IoT Core Protokolle konfigurieren](#)
- [Konfigurieren Sie eine Aktion zum erneuten Veröffentlichen von Fehlern](#)
- [Beheben Sie Regelprobleme](#)
- [Problembehandlung bei einer Regel \(\)AWS IoT SiteWise](#)
- [Problembehandlung bei einer Regel \(DynamoDB\)](#)

AWS IoT Core Protokolle konfigurieren

Sie können so konfigurieren AWS IoT , dass verschiedene Informationsebenen in CloudWatch Logs protokolliert werden.

Um CloudWatch Protokolle zu konfigurieren und darauf zuzugreifen

1. Informationen zur Konfiguration der Protokollierung finden Sie unter [Monitoring with CloudWatch Logs](#) im AWS IoT Developer Guide. AWS IoT Core
2. Navigieren Sie zur [CloudWatch -Konsole](#).
3. Wählen Sie im Navigationsbereich Protokollgruppen aus.
4. Wählen Sie die Gruppe AWSIoTLogs aus.
5. Wählen Sie einen aktuellen Protokolldatenstrom aus. CloudWatch Zeigt standardmäßig den neuesten Protokollstream zuerst an.
6. Wählen Sie einen Protokolleintrag, um die Protokollmeldung zu erweitern. Ihr Protokolleintrag könnte wie der folgende Screenshot aussehen.

CloudWatch > Log Groups > AWSIoTLogs > 9ca6614a-00fc-4f9e-8100-5c2a34918e90_123456789012_0

Expand all ☒ Row ☐ Text   

Filter events all 2020-02-10 (19:36:11) ▾

Time (UTC +00:00)	Message
2020-02-11	No older events found at the moment. Retry .
▼ 19:36:11	2020-02-11 19:36:11.823 TRACEID:d4cd3bd0-ac41-cd4a-4f59-74a242ec70e6 PRINCIPALID:AIDAZ2YMUHYHIEDEL3VA3 [ERROR] EVENT:IotSiteWise
	2020-02-11 19:36:11.823 TRACEID:d4cd3bd0-ac41-cd4a-4f59-74a242ec70e6 PRINCIPALID:AIDAZ2YMUHYHIEDEL3VA3 [ERROR] EVENT:IotSiteWiseActionFailure TOPICNAME:/tutorial/device/SiteWiseTutorialDevice1/cpu CLIENTID:iotconsole-1581444173801-0 MESSAGE:Failed to send message data to IoT SiteWise asset properties. [Code: InvalidRequestException, Message: Property value does not match data type DOUBLE]. Message arrived on: /tutorial/device/SiteWiseTutorialDevice1/cpu, Action: iotSiteWise
	No newer events found at the moment. Retry .

7. Vergleichen Sie die Fehlermeldungen mit den Fehlern in diesem Thema, um Ihr Problem zu beheben.

Konfigurieren Sie eine Aktion zum erneuten Veröffentlichen von Fehlern

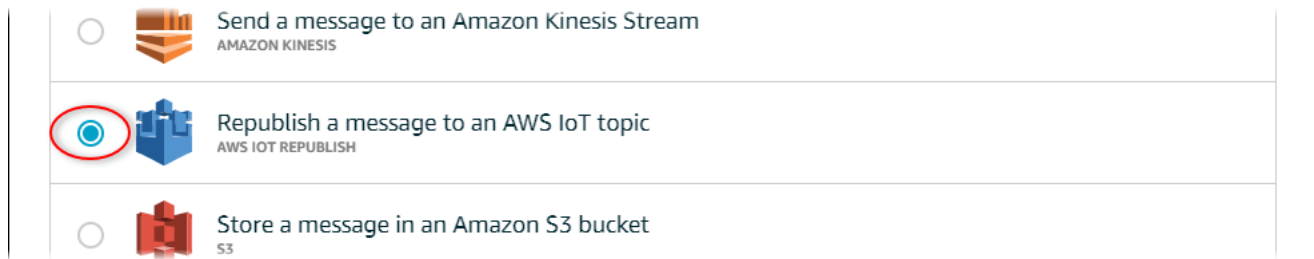
Sie können eine Fehleraktion für Ihre Regel konfigurieren, um Fehlermeldungen zu verarbeiten. In diesem Verfahren konfigurieren Sie die Aktion zur Wiederveröffentlichung der Regel, um Fehlermeldungen im MQTT-Testclient anzuzeigen.

Note

Die Aktion zum erneuten Veröffentlichen eines Fehlers gibt nur das Äquivalent der ERROR-Ebenenprotokolle aus. Wenn Sie ausführlichere Protokolle wünschen, müssen Sie Logs [konfigurieren CloudWatch](#).

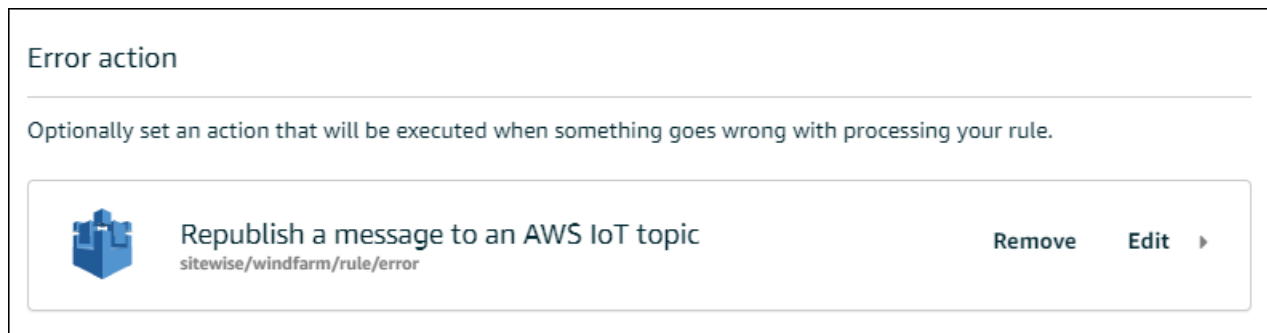
So fügen Sie einer Regel eine Aktion zur Wiederveröffentlichung eines Fehlers hinzu

1. Navigieren Sie zur [AWS IoT -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Act (Agieren) und dann Rules (Regeln) aus.
3. Wählen Sie Ihre Regel aus.
4. Wählen Sie unter Error action (Fehleraktion) die Option Add action (Aktion hinzufügen) aus.
5. Wählen Sie Nachricht zu einem AWS IoT Thema erneut veröffentlichen.



6. Klicken Sie unten auf der Seite auf Configure action (Aktion konfigurieren).
7. Geben Sie im Feld Thema ein eindeutiges Thema ein (z. B. **sitewise/windfarm/rule/error**). AWS IoT Core veröffentlicht Fehlermeldungen zu diesem Thema erneut.
8. Wählen Sie „Auswählen“, um AWS IoT Core Zugriff zur Ausführung der Fehleraktion zu gewähren.
9. Wählen Sie neben der Rolle, die Sie für die Regel erstellt haben, Select (Auswählen).
10. Wählen Sie Update Role (Rolle aktualisieren) aus, um der Rolle die zusätzlichen Berechtigungen hinzuzufügen.
11. Wählen Sie Aktion hinzufügen aus.

Die Fehleraktion Ihrer Regel sollte dem folgenden Screenshot ähnlich aussehen.



12. Klicken Sie oben links auf der Konsole auf den Zurück-Pfeil, um zur Startseite der AWS IoT Konsole zurückzukehren.

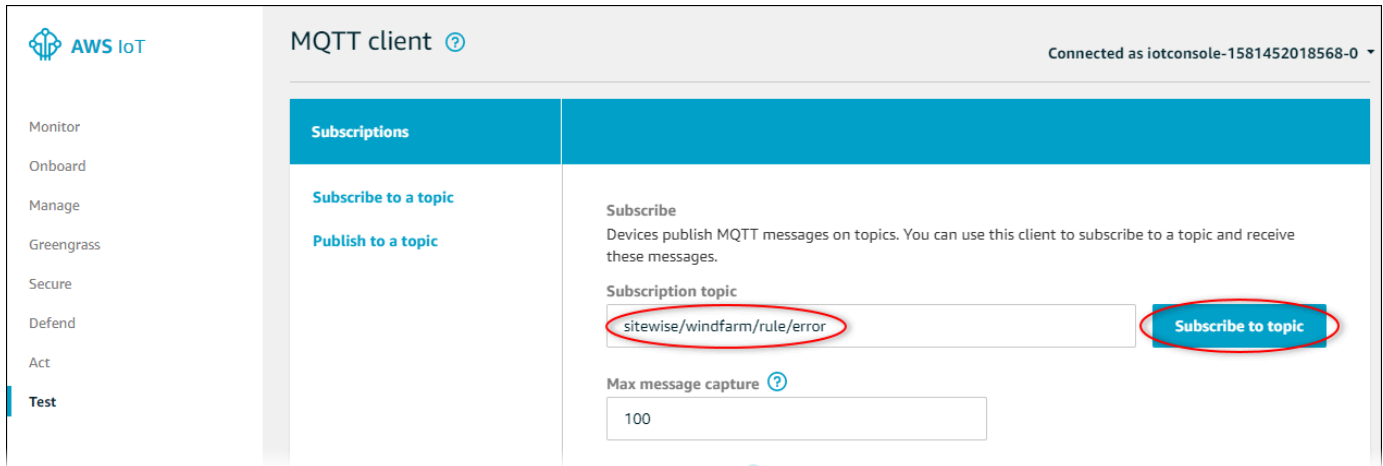
Nachdem Sie die Aktion „Republish error (Fehler wiederveröffentlichen)“ eingerichtet haben, können Sie die Fehlermeldungen in AWS IoT Core im MQTT-Testclient anzeigen.

Im folgenden Verfahren abonnieren Sie das Fehlerthema im MQTT-Testclient. Im MQTT-Testclient können Sie die Fehlermeldungen Ihrer Regel erhalten, um das Problem zu beheben.

So abonnieren Sie das Fehleraktionsthema.

1. Navigieren Sie zur [AWS IoT -Konsole](#).

2. Wählen Sie auf der linken Navigationsseite Test, um den MQTT-Testclient zu öffnen.
3. Geben Sie im Feld Subscription topic (Abonnementsthema) das zuvor konfigurierte Fehlerthema ein (z. B. **sitewise/windfarm/rule/error**), und wählen Sie Subscribe to topic (Thema abonnieren).



4. Achten Sie auf die angezeigten Fehlermeldungen, und erweitern Sie dann das failures-Array in einer beliebigen Fehlermeldung.

Vergleichen Sie anschließend die Fehlermeldungen mit den Fehlern in diesem Thema, um Ihr Problem zu beheben.

Beheben Sie Regelprobleme

Verwenden Sie die folgenden Informationen, um Regelprobleme zu beheben.

Problembereiche

- [Fehler: Das Mitglied muss sich innerhalb von 604.800 Sekunden vor und 300 Sekunden nach dem aktuellen Zeitstempel befinden](#)
- [Fehler: Eigenschaftswert stimmt nicht mit dem Datentyp <type> überein](#)
- [Fehler: Benutzer: <role-arn> ist nicht berechtigt, Folgendes auszuführen: iotsitewise: on resource BatchPutAssetPropertyValue](#)
- [Fehler: iot.amazonaws.com kann Folgendes nicht ausführen: sts: auf der Ressource: AssumeRole <role-arn>](#)
- [Information: Es wurden keine Anfragen gesendet. PutAssetPropertyValueEntries war nach der Ausführung von Ersatzvorlagen leer.](#)

Fehler: Das Mitglied muss sich innerhalb von 604.800 Sekunden vor und 300 Sekunden nach dem aktuellen Zeitstempel befinden

Ihr Zeitstempel ist älter als 7 Tage oder neuer als 5 Minuten, verglichen mit der aktuellen Unix-Epoche. Gehen Sie wie folgt vor:

- Überprüfen Sie, ob Ihr Zeitstempel in Unix-Epoche (UTC) Zeit angegeben wird. Wenn Sie einen Zeitstempel mit einer anderen Zeitzone angeben, erhalten Sie diesen Fehler.
- Vergewissern Sie sich, dass Ihr Zeitstempel in Sekunden angegeben ist. AWS IoT SiteWise erwartet, dass Zeitstempel in Zeit in Sekunden (in der Unix-Epochezeit) und Offset in Nanosekunden aufgeteilt sind.
- Vergewissern Sie sich, dass Sie Daten hochladen, die nicht später als 7 Tage in der Vergangenheit mit einem Zeitstempel versehen sind.

Fehler: Eigenschaftswert stimmt nicht mit dem Datentyp <type> überein

Ein Eintrag in der Regelaktion hat einen anderen Datentyp als die Zielkomponenteneigenschaft. Beispielsweise ist Ihre Zielkomponenteneigenschaft `DOUBLE` und Ihr ausgewählter Datentyp ist `Integer` (Ganzzahl) oder Sie haben den Wert `integerValue` übergeben. Gehen Sie wie folgt vor:

- Wenn Sie die Regel von der AWS IoT Konsole aus konfigurieren, überprüfen Sie, ob Sie für jeden Eintrag den richtigen Datentyp ausgewählt haben.
- Wenn Sie die Regel über die API oder AWS Command Line Interface (AWS CLI) konfigurieren, überprüfen Sie, ob Ihr `value` Objekt das richtige Typfeld verwendet (z. B. `doubleValue` für eine `DOUBLE` Eigenschaft).

Fehler: Benutzer: <role-arn>ist nicht berechtigt, Folgendes auszuführen: `iotsitewise: on resource BatchPutAssetPropertyValue`

Ihre Regel ist nicht berechtigt, auf die Zielkomponenteneigenschaft zuzugreifen, oder die Zielkomponenteneigenschaft ist nicht vorhanden. Gehen Sie wie folgt vor:

- Überprüfen Sie, ob Ihr Eigenschaftensalias korrekt ist, und ob Sie eine Komponenteneigenschaft mit dem angegebenen Eigenschaftensalias haben. Weitere Informationen finden Sie unter [Datenströme verwalten für AWS IoT SiteWise](#).
- Überprüfen Sie, ob Ihre Regel über eine Rolle verfügt, und ob die Rolle die `iotsitewise:BatchPutAssetPropertyValue`-Berechtigung für die

Zielkomponenteneigenschaft zulässt, z. B. über die Hierarchie der Zielkomponente. Weitere Informationen finden Sie unter [Gewähren AWS IoT Sie den erforderlichen Zugriff](#).

Fehler: `iot.amazonaws.com` kann Folgendes nicht ausführen: `sts: auf der Ressource: AssumeRole <role-arn>`

Ihr Benutzer ist nicht berechtigt, die Rolle in Ihrer Regel in (IAM) zu übernehmen. AWS Identity and Access Management

Vergewissern Sie sich, dass Ihr Benutzer `iam:PassRole` Zugriff auf die Rolle in Ihrer Regel hat. Weitere Informationen finden Sie im AWS IoT Entwicklerhandbuch unter [Rollenberechtigungen weitergeben](#).

Information: Es wurden keine Anfragen gesendet. `PutAssetPropertyValueEntries` war nach der Ausführung von Ersatzvorlagen leer.

 Note

Diese Nachricht ist ein INFO-Ebenenprotokoll.

Ihre Anforderung muss mindestens einen Eintrag mit allen erforderlichen Parametern aufweisen.

Überprüfen Sie, ob die Parameter Ihrer Regel, einschließlich der Substitutionsvorlagen, zu nicht-leeren Werten führen. Substitutionsvorlagen können nicht auf Werte zugreifen, die in AS-Klauseln in Ihrer Regelabfrageanweisung definiert sind. Weitere Informationen finden Sie unter [Substitutionsvorlagen](#) im AWS IoT Entwicklerhandbuch.

Problembehandlung bei einer Regel ()AWS IoT SiteWise

Folgen Sie den Schritten in diesem Verfahren, um Fehler in Ihrer Regel zu beheben, falls die Daten zur CPU- und Speicherauslastung nicht AWS IoT SiteWise wie erwartet angezeigt werden. In diesem Verfahren konfigurieren Sie die Aktion zur Wiederveröffentlichung der Regel, um Fehlermeldungen im MQTT-Testclient anzuzeigen. Zur Fehlerbehebung können Sie auch die CloudWatch Protokollierung in Logs konfigurieren. Weitere Informationen finden Sie unter [Problembehandlung bei einer AWS IoT SiteWise Regelaktion](#).

So fügen Sie einer Regel eine Aktion zur Wiederveröffentlichung eines Fehlers hinzu

1. Navigieren Sie zur [AWS IoT -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Nachrichtenweiterleitung und dann Regeln aus.
3. Wählen Sie die Regel aus, die Sie zuvor erstellt haben, und klicken Sie auf Bearbeiten.
4. Wählen Sie unter Fehleraktion — optional die Option Fehleraktion hinzufügen aus.
5. Wählen Sie Nachricht zu einem AWS IoT Thema erneut veröffentlichen aus.
6. Geben Sie im Feld Thema den Pfad zu Ihrem Fehler ein (z. B. **sitewise/rule/tutorial/error**). AWS IoT Core veröffentlicht die Fehlermeldungen zu diesem Thema erneut.
7. Wählen Sie die Rolle aus, die Sie zuvor erstellt haben (z. B. SiteWiseTutorialDeviceRuleRole).
8. Wählen Sie Aktualisieren aus.

Nachdem Sie die Aktion „Republish error (Fehler wiederveröffentlichen)“ eingerichtet haben, können Sie die Fehlermeldungen in AWS IoT Core im MQTT-Testclient anzeigen.

Im folgenden Verfahren abonnieren Sie das Fehlerthema im MQTT-Testclient.

So abonnieren Sie das Fehleraktionsthema.

1. Navigieren Sie zur [AWS IoT -Konsole](#).
2. Wählen Sie auf der linken Navigationsseite MQTT-Testclient aus, um den MQTT-Testclient zu öffnen.
3. Geben Sie im Feld Themenfilter den Text Abonnieren ein **sitewise/rule/tutorial/error** und wählen Sie Abonnieren.

Wenn Fehlermeldungen angezeigt werden, zeigen Sie das `failures`-Array in einer beliebigen Fehlermeldung an, um Probleme zu diagnostizieren. Weitere Informationen zu möglichen Problemen und deren Behebung finden Sie unter [Problembehandlung bei einer AWS IoT SiteWise Regelaktion](#).

Wenn keine Fehler angezeigt werden, überprüfen Sie, ob Ihre Regel aktiviert ist und ob Sie das in der Aktion „Fehler wiederveröffentlichen“ konfigurierte Thema abonniert haben. Wenn nach dem Vorgehen weiterhin keine Fehler auftreten, überprüfen Sie, ob das Geräteskript ausgeführt wird und den Schatten des Geräts erfolgreich aktualisiert.

Note

Sie können auch das Shadow-Update-Thema Ihres Geräts abonnieren, um die Payload zu sehen, die Ihre AWS IoT SiteWise Aktion analysiert. Abonnieren Sie dazu das folgende Thema.

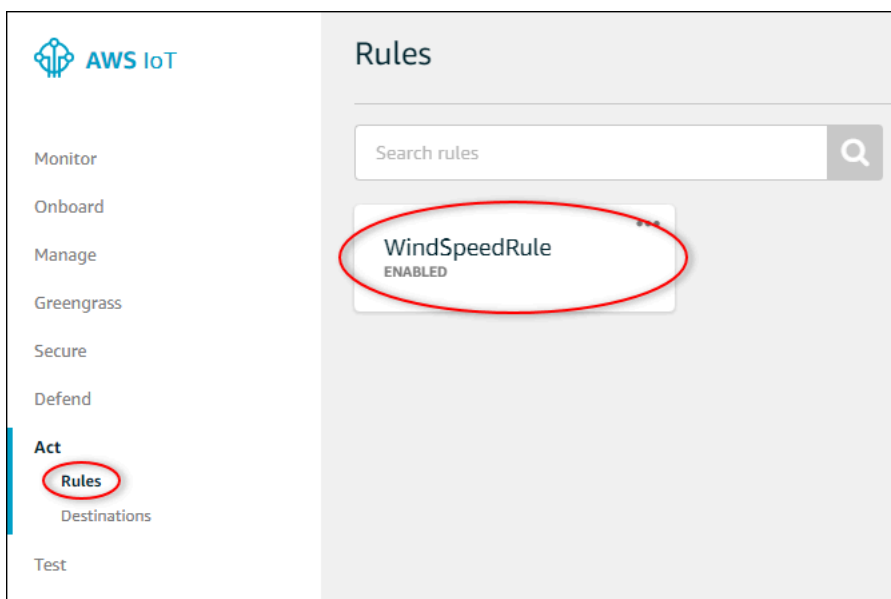
```
$aws/things/+/shadow/update/accepted
```

Problembehandlung bei einer Regel (DynamoDB)

Folgen Sie den Schritten in diesem Verfahren, um Fehler in Ihrer Regel zu beheben, falls die Demo-Asset-Daten nicht wie erwartet in der DynamoDB-Tabelle angezeigt werden. In diesem Verfahren konfigurieren Sie die Aktion zur Wiederveröffentlichung der Regel, um Fehlermeldungen im MQTT-Testclient anzuzeigen. Zur Fehlerbehebung können Sie auch die Protokollierung in CloudWatch Logs konfigurieren. Weitere Informationen finden Sie unter [Überwachen mit CloudWatch –Protokollen](#) im Entwicklerhandbuch für AWS IoT .

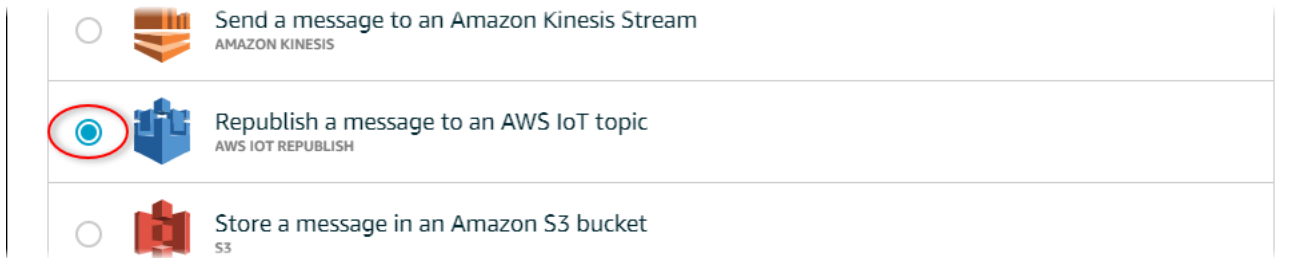
So fügen Sie einer Regel eine Aktion zur Wiederveröffentlichung eines Fehlers hinzu

1. Navigieren Sie zur [AWS IoT -Konsole](#).
2. Wählen Sie im linken Navigationsbereich Act (Agieren) und dann Rules (Regeln) aus.
3. Wählen Sie die Regel aus, die Sie zuvor erstellt haben.



4. Wählen Sie unter Error action (Fehleraktion) die Option Add action (Aktion hinzufügen) aus.

5. Wähle „Nachricht zu einem AWS IoT Thema erneut veröffentlichen“.



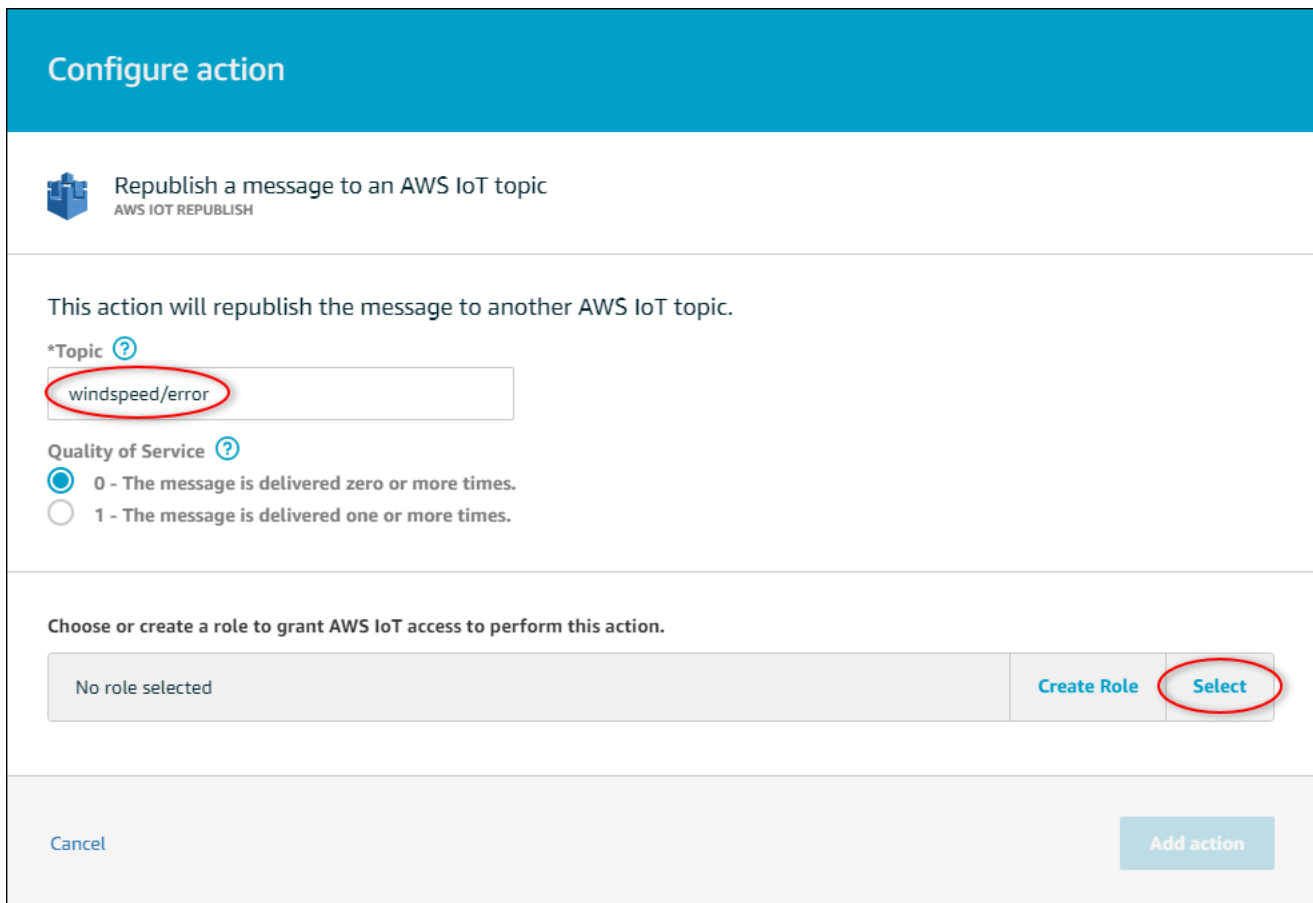
Send a message to an Amazon Kinesis Stream
AMAZON KINESIS

Republish a message to an AWS IoT topic
AWS IOT REPUBLISH

Store a message in an Amazon S3 bucket
S3

6. Klicken Sie unten auf der Seite auf Configure action (Aktion konfigurieren).

7. Geben Sie im Feld Thema **windspeed/error** ein. AWS IoT Core wird die Fehlermeldungen zu diesem Thema erneut veröffentlichen.



Configure action

Republish a message to an AWS IoT topic
AWS IOT REPUBLISH

This action will republish the message to another AWS IoT topic.

*Topic ?
windspeed/error

Quality of Service ?
☒ 0 - The message is delivered zero or more times.
☐ 1 - The message is delivered one or more times.

Choose or create a role to grant AWS IoT access to perform this action.

No role selected [Create Role](#) **Select**

[Cancel](#) [Add action](#)

8. Wählen Sie „Auswählen“, um AWS IoT Core Zugriff auf die Ausführung der Fehleraktion mithilfe der zuvor erstellten Rolle zu gewähren.

9. Wählen Sie Select (Auswählen) neben Ihrer Rolle aus.

Choose or create a role to grant AWS IoT access to perform this action.

No role selected	Refresh	Create Role	Close
Search for IAM roles			
WindSpeedDataRole	Select		

10. Wählen Sie Update Role (Rolle aktualisieren) aus, um der Rolle die zusätzlichen Berechtigungen hinzuzufügen.

1 - The message is delivered one or more times.

Choose or create a role to grant AWS IoT access to perform this action.

WindSpeedDataRole	Update Role	Create Role	Select

Cancel

Add action

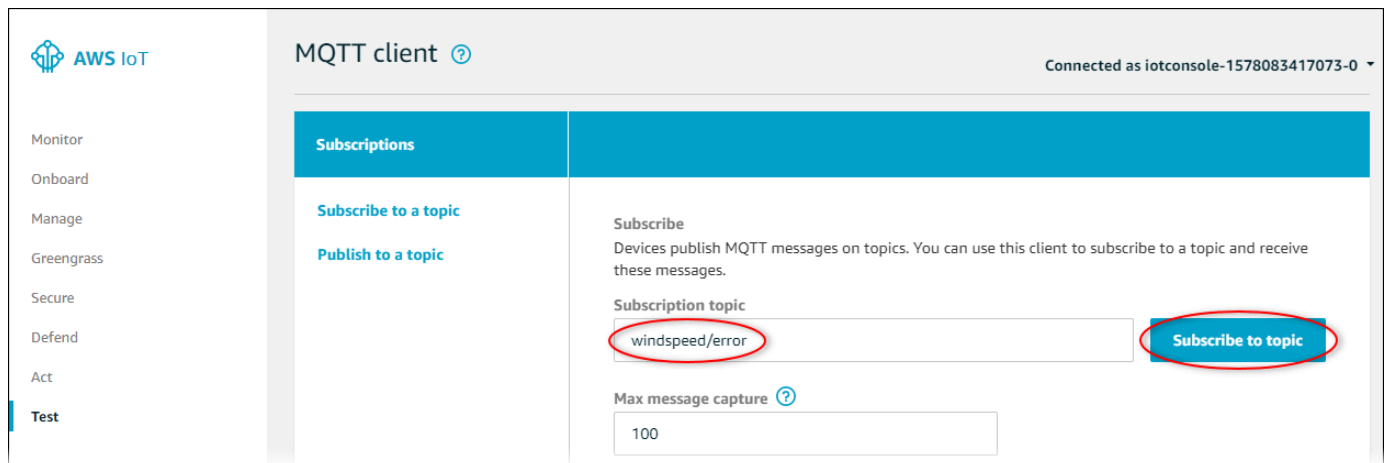
11. Wählen Sie Add action (Aktion hinzufügen) aus, um das Hinzufügen der Fehleraktion abzuschließen.
12. Wählen Sie den Zurück-Pfeil oben links auf der Konsole, um zur Startseite der AWS IoT Core-Konsole zurückzukehren.

Nachdem Sie die Aktion „Fehler erneut veröffentlichen“ eingerichtet haben, können Sie die Fehlermeldungen im MQTT-Testclient in AWS IoT Core anzeigen.

Im folgenden Verfahren abonnieren Sie das Fehlerthema im MQTT-Testclient.

So abonnieren Sie das Fehleraktionsthema.

1. Wählen Sie auf der linken Navigationsseite der AWS IoT Core-Konsole Test aus.
2. Geben Sie im Feld Subscription topic (Abonnementthema) „**windspeed/error**“ ein und wählen Sie Subscribe to topic (Thema abonnieren) aus.



3. Achten Sie darauf, dass Fehlermeldungen angezeigt werden, und erkunden Sie das `failures`-Array in einer Fehlermeldung, um die folgenden häufigsten Probleme zu diagnostizieren:

- Tippfehler in der Regelabfrageanweisung
- Unzureichende Rollenberechtigungen

Wenn keine Fehler angezeigt werden, überprüfen Sie, ob Ihre Regel aktiviert ist und ob Sie das in der Aktion „Fehler wiederveröffentlichen“ konfigurierte Thema abonniert haben. Wenn immer noch keine Fehler angezeigt werden, überprüfen Sie, ob Ihre Demo-Windparkkomponenten noch vorhanden sind und ob Sie Benachrichtigungen zu den Windgeschwindigkeitseigenschaften aktiviert haben. Wenn Ihre Demo-Assets abgelaufen sind und nicht mehr verfügbar sind AWS IoT SiteWise, können Sie eine neue Demo erstellen und die Regelabfrageanweisung aktualisieren, um das aktualisierte Asset-Modell und die aktualisierte Eigenschaft IDs widerzuspiegeln.

Problembehandlung bei Massenimport- und -exportvorgängen

Informationen zur Behandlung und Diagnose von Fehlern, die während eines Übertragungsauftrags auftreten, finden Sie in der AWS IoT TwinMaker `GetMetadataTransferJobAPI`:

1. Rufen Sie nach dem Erstellen und Ausführen eines Übertragungsauftrags die `GetMetadataTransferJobAPI` auf:

```
aws iottwinmaker get-metadata-transfer-job \
--metadata-transfer-job-id your_metadata_transfer_job_id \
```

```
--region us-east-1
```

2. Der Status des Jobs ändert sich in einen der folgenden Zustände:
 - COMPLETED
 - CANCELLED
 - ERROR
3. Die GetMetadataTransferJobAPI gibt ein [MetadataTransferJobProgress](#) Objekt zurück.
4. Das MetadataTransferJobProgress Objekt enthält die folgenden Parameter:
 - FailedCount: Gibt die Anzahl der Assets an, die während des Übertragungsvorgangs ausgefallen sind.
 - skippedCount: Gibt die Anzahl der Assets an, die während des Übertragungsvorgangs übersprungen wurden.
 - succeededCount: Gibt die Anzahl der Assets an, die während des Übertragungsvorgangs erfolgreich waren.
 - TotalCount: Gibt die Gesamtzahl der am Übertragungsprozess beteiligten Vermögenswerte an.
5. Zusätzlich wird vom API-Aufruf ein reportURL-Element zurückgegeben, das eine vorsignierte URL enthält. Wenn Ihr Übertragungsauftrag Fehler enthält, die untersucht werden müssen, können Sie unter dieser URL einen vollständigen Fehlerbericht herunterladen.

AWS IoT SiteWise Endpunkte und Kontingente

In den folgenden Abschnitten werden die Endpunkte und Kontingente für AWS IoT SiteWise beschrieben.

Themen

- [AWS IoT SiteWise Endpunkte](#)
- [AWS IoT SiteWise Kontingente](#)

AWS IoT SiteWise Endpunkte

Der Allgemeine AWS-Referenz Leitfaden listet die AWS IoT SiteWise Endpunkte für eine auf. AWS-KontoWeitere Informationen finden Sie im Leitfaden unter [AWS IoT SiteWise Endpunkte und Kontingente](#)Allgemeine AWS-Referenz .

AWS IoT SiteWise Kontingente

In den folgenden Tabellen werden die Kontingente in beschrieben. AWS IoT SiteWise Weitere Informationen zu Kontingenten und zur Beantragung von Kontingenterhöhungen finden Sie unter [AWS Servicekontingenten](#) im Allgemeine AWS-Referenz. Weitere Informationen zu AWS IoT SiteWise Kontingenten finden Sie unter [AWS IoT SiteWise Servicekontingenten](#) im Allgemeine AWS-Referenz.

Kontingente für AWS IoT SiteWise Vermögenswerte und Anlagemodelle

Ressource	Description	Kontingent	Einstellbar
Anzahl der Anlagemodelle in jedem AWS-Region für jedes AWS-Konto	Die maximale Anzahl von Asset-Modellen, die Sie in einem AWS-Region für einen erstellen können AWS-Konto.	10000	Ja
Anzahl der Assets in jedem Asset-Modell	Die maximale Anzahl von Assets, die Sie	10.000	Ja

Ressource	Description	Kontingent	Einstellbar
	für jedes Asset-Modell erstellen können.		
Anzahl der untergeordneten Vermögenswerte in jeder übergeordneten Anlage	Die maximale Anzahl von untergeordneten Vermögenswerten, die Sie einer übergeordneten Anlage zuordnen können.	2000	Ja
Tiefe der Hierarchiestruktur des Komponentenmodells	Die maximale Baumtiefe der Anlagenhierarchie für ein Anlagenmodell.	30	Ja
Anzahl der Hierarchiedefinitionen in jedem Anlagenmodell	Die maximale Anzahl von Hierarchiedefinitionen, die Sie in einem Anlagenmodell haben können.	30 —	Ja
Anzahl der Eigenschaften auf der Stammebene in jedem Asset-Modell	Die maximale Anzahl von <code>assetModelProperties</code> für jedes Anlagenmodell. Diese Anzahl beinhaltet <code>nonCompositeModelProperties</code> . Diese Quote gilt auch für jedes einzelne Asset, das mit diesem Asset-Modell erstellt wurde.	500	Ja

Ressource	Description	Kontingent	Einstellbar
Anzahl der Immobilien in einem Asset-Modell	Die maximale Anzahl von Eigenschaften eines Vermögensmodells vom Typ ASSET_MODEL oder COMPONENT_MODEL . Diese Anzahl wird durch die Kombination der Eigenschaften des Stammobjektmodells und aller enthaltenen component-model-based oder zusammengesetzten Inline-Modelle bestimmt. Diese Quote gilt auch für jedes einzelne Asset, das anhand dieses Asset-Modells erstellt wurde.	5000	Ja
Anzahl der Eigenschaften in jedem zusammengesetzten Modell	Die maximal zulässige Anzahl von Eigenschaften für zusammengesetzte Modelle. Außerdem die maximale Anzahl von Eigenschaften, die für ein Objektmodell des Typs zulässig sind COMPONENT_MODEL .	100	Ja

Ressource	Description	Kontingent	Einstellbar
Tiefe des Eigenschaftsbaums in einem Vermögensmodell	Beispielsweise hat ein Modell mit einer Transformationseigenschaft C, das eine Transformationseigenschaft B verbraucht, die wiederum eine Messeigenschaft A verbraucht, eine Tiefe von 3.	10	Nein
Anzahl der Vermögensmodelle in jedem Hierarchiebaum	Die maximale Anzahl von Anlagenmodellen, die Sie in einen einzelnen Hierarchiebaum aufnehmen können.	100	Ja

Ressource	Description	Kontingent	Einstellbar
Anzahl der direkt abhängigen Immobilien für ein Anlagemodell	Dieses Kontingent begrenzt die Anzahl der Eigenschaften, die direkt von einer einzelnen Eigenschaft abhängen können, wie in Eigenschaftsformelausdrücken definiert. Die Anzahl der abhängigen Eigenschaften muss größer sein als die Anzahl der direkt abhängigen Eigenschaften für ein Vermögensmodell. Beantragen Sie eine Erhöhung für beide Kontingente, wenn es für ein Vermögensmodell mehr direkt abhängige Immobilien als abhängige Immobilien gibt.	20	Ja
Anzahl der abhängigen Immobilien in einem Anlagenmodell	Dieses Kontingent begrenzt die Anzahl der Eigenschaften, die direkt oder indirekt von einer einzelnen Eigenschaft abhängen können, wie in Eigenschaftsformel ausdrücken definiert.	30	Nein

Ressource	Description	Kontingent	Einstellbar
Anzahl der zusammengesetzten Modelle in einem Anlagenmodell	Die maximale Anzahl von zusammengesetzten Modellen, die Sie in einem einzelnen Anlagenmodell haben können.	50	Ja
Tiefe des zusammengesetzten Modells	Die maximale Tiefe des zusammengesetzten Modellbaums in jedem Anlagenmodell, einschließlich Inline- und component-model-based Verbundmodellen.	2	Ja
Anzahl einzigartiger Anlagenmodelle, die dasselbe Komponentenmodell verwenden	Die maximale Anzahl einzigartiger Asset-Modelle, die mindestens ein component-model-based zusammengesetztes Modell haben, das direkt auf ein bestimmtes Asset-Modell vom Typ COMPONENT_MODEL verweist.	20	Ja

Ressource	Description	Kontingent	Einstellbar
Anzahl der Eigenschaftsvariablen in einem Eigenschaftsformel ausdruck	Der Ausdruck enthält beispielsweise zwei Eigenschaftsvariablen <code>temp</code> , <code>power</code> und <code>avg(power) + max(temp)</code> . Dies gilt auch für Ergebnisse von Transformationsberechnungen.	10	Nein
Anzahl der Funktionen in einem Eigenschaftsformelausdruck	Der Ausdruck enthält beispielsweise zwei Funktionen <code>max</code> , <code>avg</code> und <code>avg(power) + max(temp)</code> .	10	Nein

Kontingente für AWS IoT SiteWise Schnittstellen

Kontingente für AWS IoT SiteWise Schnittstellen

Ressource	Description	Kontingent	Einstellbar
Anzahl der Schnittstellen in jeder AWS-Region für jede AWS-Konto	Die maximale Anzahl von Schnittstellen, die Sie in einer AWS-Region für eine erstellen können AWS-Konto.	100	Ja
Anzahl der Eigenschaften in jeder Schnittstelle	Die maximale Anzahl von Eigenschaften, die Sie in einer Schnittstelle definieren können.	200	Ja

Ressource	Description	Kontingent	Einstellbar
Anzahl einzigartiger Asset-Modelle, die dasselbe Schnittstellenmodell verwenden	Die maximale Anzahl einzigartiger Anlagenmodelle, die dasselbe Schnittstellenmodell verwenden können.	500	Ja
Anzahl der Schnittstellen in jedem Hierarchiebaum	Die maximale Anzahl von Schnittstellen, die Sie in einen einzelnen Hierarchiebaum aufnehmen können.	100	Ja
Tiefe des Schnittstellen-Hierarchiebaums	Die maximale Tiefe des Schnittstellenhierarchiebaums.	30	Ja
Anzahl der Hierarchiedefinitionen in jeder Schnittstelle	Die maximale Anzahl von Hierarchiedefinitionen, die Sie in einer Schnittstelle haben können.	20	Ja

Kontingente für Daten zu AWS IoT SiteWise Vermögenswerten

Ressource	Description	Kontingent	Einstellbar
Anforderungsrate für Komponenteneigenschaftsdaten-API-Operationen	Die maximale Anzahl von API-Anfragen pro Sekunde für Immobiliendaten, die Sie AWS-Regionen jeweils ausführen können AWS-Konto.	1000	Ja

Ressource	Description	Kontingent	Einstellbar
	Dieses Kontingent gilt für API-Operationen wie GetAssetPropertyValue und BatchPutAssetPropertyValue .		
Anzahl der Datenpunkte pro Sekunde für jede Datenqualität für jede Anlageeigenschaft	Diese Quote gilt für die maximale Anzahl von timestamp-quality-value (TQV) Datenpunkten mit demselben Zeitstempel in Sekunden für jede Datenqualität für jede Anlageeigenschaft. Dies ist die maximale Anzahl von Datenpunkten guter, unsicherer und schlechter Qualität, die Sie für eine bestimmte Sekunde pro Komponenteneigenschaft speichern können.	10	Nein

Ressource	Description	Kontingent	Einstellbar
Anzahl der BatchPutAssetPropertyValue Einträge, die jede Sekunde in jede nicht attributierte Objekteigenschaft aufgenommen wurden, für jede für eine. AWS-Region AWS-Konto	Die maximale Anzahl von Einträgen in jeder Asset-Eigenschaft ohne Attribute für alle BatchPutAssetPropertyValue Quellen, einschließlich SiteWise Edge-Gateways, AWS IoT Core Regeln und API-Aufrufe.	10	Nein
Anzahl der pro Sekunde aufgenommenen BatchPutAssetPropertyValue Einträge für jede Attribut-Asset-Eigenschaft für jede in einem. AWS-Region AWS-Konto	Dieses Kontingent gilt für Einträge in jeder Attribut-Asset-Eigenschaft BatchPutAssetPropertyValue aus allen Quellen, einschließlich SiteWise Edge-Gateways, AWS IoT Core Regeln und API-Aufrufen.	1	Nein
Rate der übernommenen Datenpunkte	Die maximale Anzahl von timestamp-quality-value (TQV-) Datenpunkten, die pro Sekunde jeweils AWS-Region für einen aufgenommen werden. AWS-Konto	5000	Ja

Ressource	Description	Kontingent	Einstellbar
Rate anfragen für BatchGetAssetsPropertyAggregates	Die maximale Anzahl von BatchGetAssetsPropertyAggregates Anfragen pro Sekunde, die Sie AWS-Region in jeder Sekunde ausführen können AWS-Konto.	200	Ja
Rate der Anfragen für BatchGetAssetsPropertyValue	Die maximale Anzahl von BatchGetAssetsPropertyValue Anfragen pro Sekunde, die Sie AWS-Region in jeder Sekunde ausführen können AWS-Konto.	500	Ja
Rate der Anfragen für BatchGetAssetsPropertyValueHistory	Die maximale Anzahl von BatchGetAssetsPropertyValueHistory Anfragen pro Sekunde, die Sie ausführen können.	200	Ja

Ressource	Description	Kontingent	Einstellbar
Anzahl der pro Sekunde aufgenommenen BatchPutAssetPropertyValue Einträge für jede Objekteigenschaft AWS-Region in einem AWS-Konto.	Dieses Kontingent gilt für Einträge in jeder Asset-Eigenschaft BatchPutAssetPropertyValue aus allen Quellen, einschließlich SiteWise EdgeGateways, AWS IoT Core Regeln und API-Aufrufen.	10	Nein
Häufigkeit der GetAssetPropertyAggregates Anfragen und BatchGetAssetPropertyAggregates Eingabeabfragen für jede Anlageeigenschaft	Die maximale Gesamtzahl der GetAssetPropertyAggregates Anfragen und BatchGetAssetPropertyAggregates Einträge für jede Objekteigenschaft pro Sekunde AWS-Region in jedem Objekt AWS-Konto.	50	Nein

Ressource	Description	Kontingent	Einstellbar
Rate der GetAssetPropertyValue Anfragen und BatchGetAssetPropertyValues Eingabeabfragen für jede Anlageeigenschaft	Die maximale Gesamtzahl der GetAssetPropertyValue Anfragen und BatchGetAssetPropertyValues Einträge für jede Anlageeigenschaft pro Sekunde und AWS-Region in jeder einzelnen AWS-Konto.	500	Nein
Rate der GetAssetPropertyHistory Anfragen und BatchGetAssetPropertyHistory Eingabeabfragen für jede Anlageeigenschaft	Die maximale Gesamtzahl der GetAssetPropertyHistory Anfragen und BatchGetAssetPropertyHistory Einträge für jede Anlageeigenschaft pro Sekunde und AWS-Region in jeder einzelnen AWS-Konto.	30	Nein

Ressource	Description	Kontingent	Einstellbar
Rate der GetInterp olatedAss etPropert yValues Anfragen	Die maximale Anzahl von GetInterp olatedAss etPropert yValues Anfragen pro Sekunde, die Sie AWS-Region in jeder Sekunde ausführen können AWS-Konto.	500	Ja
Anzahl der Ergebniss e in jeder GetInterp olatedAss etPropert yValues Anfrage	Die maximale Anzahl von Ergebnissen, die für eine paginiert e GetInterp olatedAss etPropert yValues Anfrage zurückgegeben werden.	10	Ja

Ressource	Description	Kontingent	Einstellbar
Rate der abgerufenen Datenpunkte von GetAssetPropertyValueHistory und BatchGetAssetPropertyHistory	Die maximale Byterate (MB/Sekunde) der pro Sekunde abgerufenen Datenpunkte für jeden AWS-Region in einem Across und. AWS-Konto GetAssetPropertyValueHistory BatchGetAssetPropertyHistory Die für dieses Kontingent ausgewertete Antwortnutzlast verwendet Timestamp-Quality-Value (TQV) -Felder für jeden Datenpunkt und rundet die Bytegröße für jede API-Anfrage auf das nächste 4-KB-Inkrement. Timestamp-quality-value (TQV) - Datenpunkte, die jede Sekunde abgerufen werden, variieren je nach Datentyp:	100	Ja

Ressource	Description	Kontingent	Einstellbar
	- Ganzzahl — bis zu 5 Millionen TQV pro Sekunde - Doppelt — bis zu 4 Millionen TQV pro Sekunde - Boolean — bis zu 6 Millionen TQV pro Sekunde - Zeichenfolge — variiert je nach Größe der einzelnen Zeichenkettenwerte		
Anzahl der Eigenschaften, die von einer einzelnen Eigenschaft innerhalb eines verknüpften Asset-Modells abhängen	Die maximale Anzahl von Eigenschaften, die direkt oder indirekt von einer einzelnen Eigenschaft innerhalb eines verknüpften Vermögensmodells abhängen.	10	Ja

Kontingente für SiteWise Edge-Gateways

Ressource	Description	Kontingent	Einstellbar
Anzahl der SiteWise Edge-Gateways in jedem AWS-Region für ein AWS-Konto	Die maximale Anzahl von SiteWise Edge-Gateways, die Sie in einem AWS-Region	100	Ja

Ressource	Description	Kontingent	Einstellbar
	n für einen erstellen können. AWS-Konto		
Anzahl der OPC UA-Quellen in einem Edge-Gateway SiteWise	Die maximale Anzahl von OPC UA-Quellen, die Sie in einem SiteWise Edge-Gateway konfigurieren können.	100	Nein
Gesamtzahl der Ziele in einem SiteWise Edge-Gateway	Die maximale Anzahl von Zielen, die Sie in einem SiteWise Edge-Gateway konfigurieren können.	100	Nein

Kontingente für AWS IoT SiteWise Monitor

Note

Die SiteWise Monitor-Funktion ist für Neukunden nicht mehr verfügbar. Bestandskunden können den Service weiterhin wie gewohnt nutzen. Weitere Informationen finden Sie unter [Änderung der Verfügbarkeit SiteWise überwachen](#).

Ressource	Description	Kontingent	Einstellbar
Anzahl der Portale in jedem AWS-Region für ein AWS-Konto	Die maximale Anzahl von SiteWise Monitor-Portalen, die Sie in einem AWS-Region für einen erstellen können AWS-Konto.	100	Ja

Ressource	Description	Kontingent	Einstellbar
Anzahl der Projekte in einem Portal	Die maximale Anzahl von Projekten, die Sie in einem SiteWise Monitor-Portal erstellen können.	100	Ja
Anzahl der Dashboards in einem Projekt	Die maximale Anzahl von Dashboards, die Sie innerhalb eines Projekts in SiteWise Monitor erstellen können.	100	Ja
Anzahl der Root-Assets in einem Projekt	Die maximale Anzahl von Elementen der obersten Ebene, die Sie einem Projekt in SiteWise Monitor hinzufügen können.	1	Nein
Anzahl der Visualisierungen in einem Dashboard	Die maximale Anzahl visueller Elemente (wie Diagramme, Grafiken oder Tabellen), die Sie einem Dashboard in SiteWise Monitor hinzufügen können.	10	Ja

Ressource	Description	Kontingent	Einstellbar
Anzahl der Metriken in jeder Dashboard-Visualisierung	Die maximale Anzahl von Metriken oder Datenpunkten, die Sie in einer einzigen Visualisierung auf einem Dashboard in SiteWise Monitor anzeigen können.	5	Ja
Anzahl der Schwellenwerte für jede Dashboard-Visualisierung	Die maximale Anzahl von Schwellenwerten, die Sie für jede Visualisierung auf einem Dashboard in SiteWise Monitor festlegen können.	12	Nein

Kontingente für den AWS IoT SiteWise Massenimport und -export von Metadaten

Ressource	Description	Kontingent	Einstellbar
Anzahl der Metadaten transferaufträge in der Warteschlange	Die maximale Anzahl von PENDING Metadatentransferaufträgen in der Warteschlange.	10	Ja
Größe der Importdatei für den Metadaten transferauftrag	Die maximale Größe der importierten Datei (in MB).	100	Ja

Ressource	Description	Kontingent	Einstellbar
Anzahl der AWS IoT SiteWise Importresourcen in einem Job	Die maximale Anzahl von AWS IoT SiteWise Importresourcen in einem einzelnen Job. Eine Ressource umfasst Anlagen und Anlagenmodelle.	5000	Ja
Anzahl der AWS IoT SiteWise Exportresourcen in einem Job	Die maximale Anzahl von AWS IoT SiteWise Exportresourcen in einem einzelnen Job. Eine Ressource umfasst Anlagen und Anlagenmodelle.	5000	Ja

Kontingente für den AWS IoT SiteWise Massenimport von Daten

Ressource	Description	Kontingent	Einstellbar
Anzahl der laufenden Massenimportaufträge	Die maximale Anzahl von Massenimportaufträgen, die gleichzeitig ausgeführt werden können.	100	Nein
Größe der CSV-Datei	Die maximale CSV-Dateigröße (in GB) in einem Massenimportauftrag.	10	Nein

Ressource	Description	Kontingent	Einstellbar
Größe der unkomprimierten Parquet-Datei	Die maximale Dateigröße (in MB) für eine unkomprimierte Parquet-Datei in einem Massenimportauftrag.	256 MB	Nein
Größe der CSV-Datei für die gepufferte Aufnahme	Die maximale CSV-Dateigröße (in MB), wenn die gepufferte Aufnahme für einen Massenimportauftrag verwendet wird.	256 MB	Nein
Größe der unkomprimierten Parquet-Zeilengruppe	Die maximale Größe einer unkomprimierten Parquet-Reihengruppe.	64 MB	Nein
Anzahl der eindeutigen Messungen in einer Parquetdatei	Die maximale Anzahl von Einzelmessungen in einer Parquetdatei.	10000	Nein
Anzahl der Tage zwischen dem Zeitstempel in der Vergangenheit und dem heutigen Zeitpunkt für die gepufferte Aufnahme	Die maximale Anzahl von Tagen zwischen einem Zeitstempel in der Vergangenheit und dem heutigen Datum bei gepufferter Aufnahme.	30	Ja

Ressource	Description	Kontingent	Einstellbar
Anforderungsrate für jeden von beiden CreateBulkImportJobs AWS-Region AWS-Konto		10	Ja
Preis ListBulkImportJobs für jeden AWS-Region von beiden anfordern AWS-Konto		50	Ja
Preis DescribeBulkImportJobs für jeden AWS-Region von beiden anfordern AWS-Konto		50	Ja

Kontingente für die Drosselung der AWS IoT SiteWise Assistant-API

Ressource	Description	Kontingent	Einstellbar
Rate der Anfragen für den InvokeAssistant Betrieb	Die maximale Anzahl von Transaktionen pro Minute (TPM), die mit der AWS IoT SiteWise InvokeAssistant API in einem AWS-Konto durchgeführt werden können. Die TPM-Grenzwerte gelten für alle unterstützten	10	Nein

Ressource	Description	Kontingent	Einstellbar
	Regionen und können in einigen Regionen angepasst werden.		

Kontingente für die Erkennung von Anomalien

Ressource	Description	Kontingent	Einstellbar
Maximale Anzahl von Rechenmodellen	Die maximale Anzahl von Rechenmodellen.	1000	Ja
Maximale Anzahl von Eingabeeigenschaften für jedes Rechenmodell	Die maximale Anzahl von Eingabeeigenschaften, die ein Rechenmodell haben kann.	80	Nein
Maximale Anzahl von TQVs in Trainingsdaten (nach Resampling)	Die maximale Anzahl von TQVs in Trainingsdaten nach dem Resampling.	1.500.000	Nein
Maximale Anzahl von TQVs in den Bewertungsdaten (nach Resampling)	Die maximale Anzahl von TQVs in den Bewertungsdaten nach dem Resampling.	1.500.000	Nein
Minimale Zeitspanne der Trainingsdaten	Der Mindestzeitraum für Trainingsdaten. Details dazu finden Sie unter Machen Sie sich mit dem Mindestzeitraum vertraut.	14 Tage	Nein

Ressource	Description	Kontingent	Einstellbar
Maximale Anzahl gleichzeitiger ACTIVE Trainingsjobs	Die maximale Anzahl gleichzeitiger ACTIVE Trainingsjobs zu einem bestimmten Zeitpunkt.	25	Ja
Maximale Größe der Etikettendatendatei	Die maximale Größe der Etiketten datendatei.	2 MB	Nein
Maximale Anzahl ACTIVE hochfrequenter Inferenzen	Die maximale Anzahl ACTIVE hochfrequenter Inferenzen zu einem bestimmten Zeitpunkt.	250	Ja
Maximale Anzahl ACTIVE niederfrequenter Folgerungen	Die maximale Anzahl ACTIVE niederfrequenter Inferenzen zu einem bestimmten Zeitpunkt.	500	Ja
Maximale Zeit für die ACTIVE Trainingsdauer	Die maximale Zeit, für die das ACTIVE Training stattfinden kann.	24 Stunden	Ja

Kontingente für AWS IoT SiteWise SQL

Ressource	Description	Kontingent	Einstellbar
Rate der Anfragen für API SiteWise ExecuteQuery	Die maximale Anzahl von ExecuteQuery Anfragen pro Sekunde, die Sie	30	Ja

Ressource	Description	Kontingent	Einstellbar
	AWS-Region in jeder Sekunde ausführen können AWS-Konto.		
Maximale Anzahl gleichzeitig ausgeführter Abfragen	Die maximale Anzahl von Abfragen, die gleichzeitig ausgeführt werden können.	30	Ja

Dokumentenhistorie für dieAWS IoT SiteWiseBenutzer-Leitfaden

In der folgenden Tabelle wird die Dokumentation für diese Version von beschriebenAWS IoT SiteWise.

- API-Version: 02.12.2019

Änderung	Beschreibung	Datum
<u>Das Data Processing Pack (DPP) steht ab dem 7. November 2025 keinen Neukunden mehr zur Verfügung.</u>	DPP wird ab dem 7. November 2025 nicht mehr für Neukunden verfügbar sein. Wenn Sie die DPP-Funktion nutzen möchten, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen.	7. Oktober 2025
<u>AWS IoT SiteWise Monitor wird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein.</u>	AWS IoT SiteWise Monitor wird ab dem 7. November 2025 nicht mehr für Neukunden geöffnet sein. Wenn Sie die SiteWise Monitor-Funktion nutzen möchten, melden Sie sich vor diesem Datum an. Bestandskunden können den Service weiterhin wie gewohnt nutzen.	7. Oktober 2025
<u>Schnittstellen für Asset-Modelle</u>	Definieren Sie eine gemeinsame Struktur, die die Konsistenz zwischen den Anlagenmodellen gewährleistet und gleichzeitig Variation	5. August 2025

en bei der Implementierung zulässt.

[Native Unterstützung für Anomalien wurde hinzugefügt zu AWS IoT SiteWise](#)

AWS IoT SiteWiseDie native Anomalieerkennung ist ein Service für maschinelles Lernen (ML) zur Überwachung von Industrieanlagen, der abnormales Geräteverhalten erkennt und potenzielle Ausfälle identifiziert. Mit der systemeigenen Anomalieerkennung können Sie prädiktive Wartungsprogramme implementieren und suboptimale Geräteprozesse identifizieren.

28. Juli 2025

[Es wurden umfangreiche SQL-Funktionen und neue SQL-Datentypen, -Klauseln und -Operatoren zu vorhandenem SQL hinzugefügt](#)

Unterstützung für neue Datentypen, Klauseln und Operatoren hinzugefügt.

22. Juli 2025

[Verarbeiten und visualisieren Sie Daten mit SiteWise Edge- und Open-Source-Tools](#)

Integrieren Sie Open-Source-Tools wie Node-RED®, InfluxDB® und Grafana® auf AWS IoT SiteWise MQTT-enabled V3-Gateways. Verwenden Sie lokale Verarbeitung und Visualisierung, um Ihre industriellen Datenverwaltungsfunktionen zu verbessern.

3. Juli 2025

[Connect externe Anwendungen mit Benutzernamen und Passwörtern mit dem EMQX-Broker](#)

Unterstützung für die Verbindung externer Anwendungen mithilfe von Benutzernamen und Kennwörtern mit dem AWS IoT Greengrass EMQX-Broker hinzugefügt, der standardmäßig bei der Erstellung eines neuen V3-Gateways bereitgestellt wird. MQTT-enabled

01. Mai 2025

[Support für MQTT-enabled V3-Gateways auf Edge SiteWise](#)

Neue Funktionen hinzugefügt und veraltete Inhalte entfernt

26. Februar 2025

- Unterstützung für MQTT-enabled V3-Gateways hinzugefügt. Verbesserte Zielkonfiguration mithilfe von Pfadfiltern zum Abonnieren von MQTT-Themen, einschließlich direkter Datenaufnahme in Echtzeit AWS IoT SiteWise oder gepufferter Datenaufnahme mit Amazon S3.
- Version 3.0.0 des IoT SiteWise OPC UA Collector und Version 4.0.0 der SiteWise IoT-Publisher-Komponente für veröffentlicht. AWS IoT Greengrass V2
- Die vorherige Version der selbst gehosteten SiteWise Edge-Gateways wurde in Classic Streams, V2-Gateways umbenannt.
- Verweise auf AWS IoT Greengrass V1 in der SiteWise Edge-Dokumentation wurden entfernt, da die Verwendung nicht mehr unterstützt wird. AWS IoT SiteWise

Unterstützung für AWS IoT SiteWiseAssistant	Unterstützung für den AWS IoT SiteWise Assistenten hinzugefügt — einen generativen AI-powered Assistenten.	18. November 2024
Es wurden konfigurierbare Sitzungs-Timeouts für SiteWise Edge-APIs hinzugefügt	Es wurden konfigurierbare Sitzungs-Timeout-Einstellungen hinzugefügt, um Inaktivitätszeiträume für AWS OpsHub und SiteWise Edge-APIs zu verwalten.	31. Oktober 2024
Konfigurierbare Proxyeinstellungen für SiteWise Edge-APIs hinzugefügt	Es wurde die Verwaltung von Trust Store-Informationen hinzugefügt, um die HTTPS-Proxyunterstützung für SiteWise Edge-Gateways zu aktivieren.	31. Oktober 2024
Aktivieren Sie CORS für Edge-APIs SiteWise	CORS-Unterstützung für SiteWise Edge-APIs hinzugefügt, um einen sicheren domänenübergreifenden Zugriff auf Webanwendungen zu ermöglichen.	30. September 2024
Support für Datenquellen CloudRail und Litmus Edge Partnerdatenquellen	Unterstützung für beide Datenquellen CloudRail und Litmus Edge als Partnerdatenquellen hinzugefügt.	5. September 2024
Allgemeine Verfügbarkeit für die Ausführung von SiteWise Edge auf Siemens Industrial Edge	AWS IoT SiteWiseunterstützt jetzt die allgemeine Verfügbarkeit der Ausführung von SiteWise Edge auf Siemens Industrial Edge-Geräten.	24. Juli 2024

<u>Unterstützung für die Zeitstempelkonfiguration auf OPC UA-Datenquellen hinzugefügt</u>	AWS IoT SiteWiseunterstützt jetzt die Zeitstempelkonfiguration für OPC UA-Datenquellen.	24. Juli 2024
<u>Unterstützung für die Datentypkonvertierung auf OPC UA-Datenquellen hinzugefügt</u>	AWS IoT SiteWiseunterstützt jetzt die Datentypkonvertierung für nicht unterstützte OPC UA-Datentypen.	24. Juli 2024
<u>Unterstützung für die Ausführung einer Vorversion von SiteWise Edge auf Siemens Industrial Edge wurde hinzugefügt</u>	AWS IoT SiteWiseunterstützt jetzt die Ausführung einer SiteWise Edge-Vorschau auf Siemens Industrial Edge-Geräten.	26. November 2023
<u>Unterstützung für Warm-Tier-Speicher hinzugefügt</u>	AWS IoT SiteWiseunterstützt jetzt Warm Storage, eine vollständig verwaltete Speicherebene, die es Kunden erleichtert, Industriedaten sicher zu speichern und darauf zuzugreifen.	15. November 2023
<u>Unterstützung für benutzerdefinierte eindeutige Identifikatoren hinzugefügt</u>	AWS IoT SiteWiseunterstützt jetzt die Verwendung von benutzerdefinierten eindeutigen Identifikatoren für Anlagen, Objektmodelle, Eigenschaften und Hierarchien.	15. November 2023

<u>Unterstützung für die Erkennung multivariater Anomalien von Industrieanlagen hinzugefügt</u>	AWS IoT SiteWiseunterstützt jetzt die Erkennung multivariater Anomalien von Industrieanlagen durch die Integration von historischen und Echtzeit-Gerätedaten mit Amazon Lookout for Equipment.	15. November 2023
<u>Zusätzliche Unterstützung für die kosteneffiziente und skalierbare Erfassung von Zeitreihendaten in AWS IoT SiteWise</u>	AWS IoT SiteWiseunterstützt jetzt die kosteneffiziente und skalierbare Erfassung von Zeitreihendaten, die für analytische Anwendungsfälle benötigt werden.	15. November 2023
<u>Unterstützung für Massenimport, -export und -aktualisierung hinzugefügt</u>	AWS IoT SiteWiseunterstützt jetzt den Massenimport, den Export und die Aktualisierung von Metadaten für Industrieanlagen.	15. November 2023
<u>Unterstützung für Komponenten des Asset-Modells wurde hinzugefügt</u>	AWS IoT SiteWiseunterstützt jetzt Komponenten des Asset-Modells, um Industriekunden bei der Erstellung wiederverwendbarer Komponenten zu unterstützen.	15. November 2023
<u>Unterstützung für IoT-Dashboard-Anwendung hinzugefügt</u>	AWS IoT SiteWiseunterstützt jetzt eine Open-Source-Dashboard-Anwendung, mit der Sie Betriebsdaten visualisieren und mit ihnen interagieren können.	15. November 2023

Die serviceverknüpften Rollen für wurden aktualisiert AWS IoT SiteWise	AWS IoT SiteWise hat neue dienstbezogene Rollen und kann eine Metadaten suchabfrage für die AWS IoT TwinMaker Datenbank ausführen.	6. November 2023
Das Tagging für AWS IoT SiteWise Datenstream-Ressourcen wurde aktualisiert	Unterstützung für das Taggen von Datenstream-Ressourcen wurde hinzugefügt.	18. August 2022
Aktualisierte SiteWise Edge-Gateways	Sie können den Publisher jetzt so konfigurieren, dass er steuert, welche Daten vom Edge an die Cloud gesendet werden und in welcher Reihenfolge sie an die Cloud gesendet werden.	12. Januar 2022
Die AWS IoT SiteWise Demo wurde aktualisiert	Sie können die Demo jetzt verwenden, um ein SiteWise Monitor-Portal zu erstellen.	10. Januar 2022
Die Speicherverwaltung wurde aktualisiert	Sie können jetzt einen Aufbewahrungszeitraum definieren, um zu kontrollieren, wie lange Ihre Daten im Hot-Tier aufbewahrt werden.	29. November 2021
Unterstützung für die Verwaltung von Datenströmen hinzugefügt	Sie können jetzt Daten aufnehmen, AWS IoT SiteWise bevor Sie Asset-Modelle und Assets erstellen.	24. November 2021
Die Hierarchien der Asset-Modelle wurden aktualisiert	Ein untergeordnetes Anlagemodell kann jetzt mehreren übergeordneten Vermögensmodellen zugeordnet werden.	28. Oktober 2021

Start in der Region	Gestartet AWS IoT SiteWise in AWS GovCloud (US-West).	29. September 2021
Aktualisierte Funktionen	Die folgenden Funktionen wurden hinzugefügt • In Metriken können Sie verschachtelte Ausdrücke in Aggregationsfunktionen und temporalen Funktionen verwenden. • In Transformationen können Sie die Funktion pretrigger() verwenden, um den Wert einer Variablen vor der Eigenschaftensaktualisierung abzurufen, die die aktuelle Transformationsberechnung ausgelöst hat.	10. August 2021
Benutzerdefiniertes metrisches Zeitintervall	Unterstützung für benutzerdefinierte Zeitintervalle und Offsets in Metriken hinzugefügt.	3. August 2021
Einsatz AWS IoT SiteWise am Netzwerkrand	Die Funktion zur Kantenbearbeitung ist jetzt allgemein verfügbar.	29. Juli 2021
Daten nach Amazon S3 exportieren	AWS IoT SiteWise kann jetzt Daten nach Amazon S3 exportieren.	27. Juli 2021
VPC-Endpunkte () AWS PrivateLink	Der VPC-Schnittstellen-Endpunkt für die API-Operationen auf der Kontrollebene ist jetzt allgemein verfügbar.	15. Juli 2021

<u>Transformiert</u>	Transformationen können jetzt mehrere Variablen für Asset-Eigenschaften eingeben.	8. Juli 2021
<u>Die Funktion timestamp () wurde aktualisiert</u>	In Transformationen können Sie jetzt eine Variable als Argument für die timestamp () Funktion angeben.	16. Juni 2021
<u>Alarmer, allgemeine Verfügbarkeit</u>	Die Alarmfunktion ist jetzt allgemein verfügbar.	27. Mai 2021
<u>Modbus-TCP Protokolladapter Version 2 veröffentlicht</u>	Version 2 des <u>Modbus-TCP Protokolladapter-Connectors</u> ist verfügbar. Diese Version fügte Unterstützung für ASCII-, UTF8- und ISO8859-kodierte Quellzeichenfolgen hinzu.	24. Mai 2021
<u>Servicekontingente wurden aktualisiert</u>	Die folgenden Kontingente für die <u>GetInterpolatedAssetPropertyValues</u> API wurden hinzugefügt: Anzahl der GetInterpolatedAssetPropertyValues Anfragen, Anzahl der Ergebnisse pro GetInterpolatedAssetPropertyValues Anfrage und Anzahl der Tage zwischen dem Startdatum in der Vergangenheit und dem heutigen Datum für GetInterpolatedAssetPropertyValues .	29. April 2021

[FormelAusdrücke wurden aktualisiert](#)

Die folgenden Operatoren und Funktionen wurden hinzugefügt:

22. April 2021

- Die folgenden [Operatoren](#) wurden hinzugefügt: `<`, `>`, `<=`, `>=`, `=`, `!=`, `!`, `and`, `or`, `undnot`.
- Die folgende [Vergleichsfunktion](#) wurde hinzugefügt: `neq(x, y)`.
- Die folgenden [Zeichenkettenfunktionen](#) wurden hinzugefügt: `join()`, `format()`, und `f''`.

[VPC-Endpunkte \(\) AWS PrivateLink](#)

Es wurden Informationen zum Herstellen einer privaten Verbindung zwischen Ihrer Virtual Private Cloud (VPC) und den APIs der AWS IoT SiteWise Steuerungsebene hinzugefügt, indem Sie einen VPC-Schnittstellen-Endpunkt erstellen.

16. März 2021

[IAM-Föderation](#)

Die Administratoren und Benutzer Ihres SiteWise Monitor-Portals können sich jetzt mit ihren IAM-Anmeldedaten bei den ihnen zugewiesenen Portalen anmelden.

16. März 2021

[Start der Region](#)

AWS IoT SiteWise in China (Peking) eingeführt.

3. Februar 2021

[SiteWise IoT-Connector Version 10 veröffentlicht](#)

Version 10 des SiteWise IoT-Connectors ist verfügbar. Diese Version ist so konfiguriert, dass der `StreamManager`, die Handhabung verbessert wird, wenn die Quellverbindung verloren geht und wieder hergestellt wird. Diese Version akzeptiert auch OPC UA-Werte mit einem `ServerTimestamp`, wenn kein `SourceTimestamp` verfügbar ist.

22. Januar 2021

[Datums- und Uhrzeitfunktionen](#)

AWS IoT SiteWise unterstützt jetzt Datums- und Uhrzeitfunktionen.

21. Januar 2021

[Syntax der Funktion](#)

Sie können jetzt die Uniform Function Call Syntax (UFCS) für AWS IoT SiteWise Funktionen verwenden.

11. Januar 2021

[Integration mit Grafana](#)

Es wurden Informationen zur Visualisierung von AWS IoT SiteWise Daten in Grafana-Dashboards hinzugefügt.

15. Dezember 2020

[AWS IoT SiteWise Veröffentlichung der Funktion](#)

15. Dezember 2020

Sie können jetzt Ihre Daten mit Alarmen überwachen, Industriedaten an der Peripherie verarbeiten, Modbus TCP und Ethernet/IP Quellen für Ihr SiteWise Edge-Gateway verwenden, eingehende Daten mit Deadbands filtern und vieles mehr.

- Der Abschnitt „[Daten mit Alarmen überwachen](#)“ wurde hinzugefügt, in dem Sie Alarmer definieren, konfigurieren und darauf reagieren können. AWS IoT SiteWise
- Der Abschnitt [Edge-Verarbeitung](#) wurde hinzugefügt, in dem Sie die Verarbeitung Ihrer Industriedaten auf Ihren Edge-Geräten konfigurieren können.
- [Modbus TCP und Ethernet/IP](#) Abschnitte wurden der SiteWise Edge-Gateway-Quelldokumentation hinzugefügt.
- Der [Quellzielbereich](#) wurde hinzugefügt, mit dem Sie anpassen können, wohin Sie Ihre eingehenden Industriedaten senden.
- Es wurde der [OPC UA-Filterbereich](#) hinzugefügt, mit dem Sie die Häufigkeit

und Art der Daten steuern können, die von Ihrem lokalen Industrieserver an Ihr SiteWise Edge-Gateway gesendet werden.

[AWS IoT SiteWiseunterstützt jetzt vom Kunden verwaltete CMKs.](#)

AWS IoT SiteWiseunterstützt jetzt die Verschlüsselung mit vom Kunden verwalteten CMKs.

24. November 2020

[SiteWise IoT-Connector Version 8 veröffentlicht](#)

Version 8 des SiteWise IoT-Connectors ist verfügbar. Diese Version verbessert die Stabilität, wenn die Netzwerkonnektivität des Connectors unterbrochen wird.

19. November 2020

[Verwendung von Zeichenketten und Bedingungen in Formelausdrücken](#)

Es wurden Informationen zur Verwendung von Zeichenketten und bedingten Funktionen in Formelausdrücken für Transformationen und Metriken hinzugefügt.

16. November 2020

[Daten mithilfe AWS IoT Greengrass des Stream-Managers aufnehmen](#)

Es wurden Informationen darüber hinzugefügt, wie Sie mithilfe eines AWS IoT Greengrass Edge-Geräts umfangreiche IoT-Daten aus lokalen Datenquellen aufnehmen können.

16. September 2020

[VPC-Endpunkte \(\) AWS PrivateLink](#)

Es wurden Informationen zum Herstellen einer privaten Verbindung zwischen Ihrer Virtual Private Cloud (VPC) und den AWS IoT SiteWise Daten-APIs hinzugefügt, indem Sie einen VPC-Schnittstellen-Endpunkt erstellen.

4. September 2020

[SiteWise IoT-Connector Version 7 veröffentlicht](#)

Version 7 des SiteWise IoT-Connectors ist verfügbar. Diese Version behebt ein Problem mit SiteWise Edge-Gateway-Metriken.

14. August 2020

[IAM Identity Center-Benutzer von der AWS IoT SiteWise Konsole aus erstellen](#)

Es wurden Informationen darüber hinzugefügt, wie Sie IAM Identity Center-Benutzer in der AWS IoT SiteWise Konsole erstellen können. Sie können jetzt IAM Identity Center-Benutzer erstellen, wenn Sie Benutzer einem neuen oder vorhandenen Portal zuweisen. Das Tutorial „[Windparkdaten visualisieren und teilen](#)“ wurde aktualisiert, um diese Funktion nutzen zu können. Diese Änderung reduziert die Anzahl der Schritte im Tutorial.

4. August 2020

[Die Fehlerbehebung SiteWise am Edge-Gateway wurde verbessert](#)

Es wurden zusätzliche Informationen zur Fehlerbehebung bei einem SiteWise Edge-Gateway und zum [Exportieren des OPC UA-Client-Zertifikats](#) für eine Quelle hinzugefügt.

18. Juni 2020

[Dokumentation zu den Aufgaben in der Konsole](#)

Konsolenaufgabendokumentation für [Modellieren von industriellen Komponenten](#), [Abfragen von Komponenteneigenschaftendaten](#) und [Interaktion mit anderen Services](#) hinzugefügt. Sie können diese Anweisungen befolgen, um Aufgaben in der AWS IoT SiteWise-Konsole durchzuführen.

11. Juni 2020

[Tutorial zum Analysieren exportierter Daten](#)

Es wurde ein Tutorial hinzugefügt, dem Sie folgen können, um zu erfahren, wie Sie Amazon Athena verwenden, um Asset-Daten zu analysieren, die Sie mit der [CloudFormationExportfunktionsvorlage](#) nach Amazon S3 exportiert haben.

27. Mai 2020

[Die Verwendung von FormelAusdrücken wurde verbessert](#)

Es wurden detaillierte Informationen zum Verhalten von AWS IoT SiteWise Formeleigenschaften hinzugefügt und ein Beispiel für das Zählen gefilterter Datenpunkte hinzugefügt.

18. Mai 2020

[SiteWise IoT-Connector Version 6 veröffentlicht](#)

Version 6 des SiteWise IoT-Connectors ist verfügbar.

Diese Version bietet Unterstützung für CloudWatch Metriken und die automatische Erkennung neuer OPC UA-Tags. Das bedeutet, dass Sie Ihr SiteWise Edge-Gateway nicht neu starten müssen, wenn sich die Tags für Ihre OPC UA-Quellen ändern. Diese Version des Connectors erfordert Stream Manager und AWS IoT Greengrass Core-Software v1.10.0 oder höher.

29. April 2020

[AWS IoT SiteWise Veröffentlichung der Funktion](#)

AWS IoT SiteWise Veröffentlichung einer Funktion.

29. April 2020

Sie können jetzt SiteWise Edge-Gateways mit der API verwalten, Ihr Logo zu Portalen hinzufügen, SiteWise Edge-Gateway-Metriken anzeigen und vieles mehr.

- Der Abschnitt [Daten nach Amazon S3 exportieren](#) wurde mit einer CloudFormation Vorlage hinzugefügt, mit der Sie neue Datenwerte in einen Amazon S3 S3-Bucket exportieren können.
- Der Abschnitt [Konfiguration von Datenquellen](#) wurde hinzugefügt, der die Quellendokumentation für das SiteWise Edge-Gateway verbessert und die neuen SiteWise Edge-Gateway-APIs enthält.
- Der Abschnitt SiteWise Edge-Gateway-Metriken wurde hinzugefügt, in dem die von SiteWise Edge-Gateways veröffentlichten CloudWatch Metriken beschrieben werden.
- Der Abschnitt Konfiguration eines SiteWise Edge-Gateways auf Amazon EC2 wurde mit einer CloudFormation Vorlage hinzugefügt,

mit der Sie SiteWise Edge-Gateway-Abhängigkeiten auf einer Amazon EC2 EC2-Instance schnell konfigurieren können.

- Der Abschnitt mit den [Portal-Servicerollen](#) wurde hinzugefügt, in dem die neue Berechtigungsfunktion von SiteWise Monitor-Portalen beschrieben wird.
- Aktualisierung der [Portaldokumentation](#) für Portal-Servicerollen und Portal-Logos.
- Der Abschnitt „[AWS IoT SiteWiseRessourcen taggen](#)“ wurde hinzugefügt.
- Aktualisierung des Abschnitts [Erstellung von Dashboards \(CLI\)](#) für die neue Dashboard-Definitionsstruktur.
- Hinzufügung des Abschnitts [Sicherheit](#).

[Tutorial zum Visualisieren und Teilen von Windparkdaten im SiteWise Monitor-Tutorial](#)

Es wurde ein Tutorial hinzugefügt, dem Sie folgen können, um zu erfahren, wie Sie AWS IoT SiteWise Monitor Anlagendaten visualisieren und teilen können.

12. März 2020

[AWS IoT SiteWiseKonzepte](#)

Es wurde ein Glossar mit AWS IoT SiteWise Begriffen hinzugefügt, anhand dessen Sie sich über den Service und seine allgemeinen Begriffe informieren können.

5. März 2020

[Die AWS IoT Greengrass Installationsanweisungen wurden entfernt](#)

Die Installationsanweisungen für die AWS IoT Greengrass Core-Software wurden aus dem AWS IoT SiteWise Benutzerhandbuch entfernt. Das [AWS IoT Greengrass Developer Guide](#) bietet ein Geräte-Setup-Skript und Anweisungen zur Einrichtung AWS IoT Greengrass auf anderen Plattformen wie Amazon EC2 und Docker.

14. Februar 2020

[Verbessertes Einlesen von Daten mithilfe von Regeln AWS IoT Core](#)

Es wurden detaillierte Informationen [zur Verwendung](#) und [Problembehebung](#) der AWS IoT SiteWise Regelaktion hinzugefügt, mit der Sie Daten aus MQTT-Nachrichten aufnehmen können. AWS IoT Core

14. Februar 2020

[SiteWise IoT-Connector Version 5 veröffentlicht](#)

Version 5 des SiteWise IoT-Connectors ist verfügbar. Diese Version behebt ein Kompatibilitätsproblem mit der AWS IoT Greengrass Core-Software v1.9.4.

12. Februar 2020

[SiteWise IoT-Connector Version 4 veröffentlicht](#)

Version 4 des SiteWise IoT-Connectors ist verfügbar. Diese Version behebt ein Problem mit der Wiederverbindung des OPC UA-Servers.

7. Februar 2020

[Umstrukturierte Modellierung von Industrieanlagen](#)

Der Abschnitt „Aktualisieren von Komponenten und Modellen“ wurde in mehrere Themen innerhalb von „Modellierung industrieller Komponenten“ umstrukturiert.

4. Februar 2020

- [Komponenten- und Modellzustände](#)
- [Datenströme verwalten für AWS IoT SiteWise](#)
- [Attributwerte aktualisieren](#)
- [Anlagen zuordnen und deren Zuordnung aufheben](#)
- [Aktualisieren Sie Ressource n und Modelle](#)
- [Löschen Sie Objekte und Modelle in AWS IoT SiteWise](#)

[Tutorial zum Aufnehmen von Daten aus Dingen AWS IoT](#)

Es wurde ein Tutorial hinzugefügt, dem Sie folgen können, um zu erfahren, wie Sie eine AWS IoT SiteWise Regelaktion konfigurieren, um Daten aus einer neuen oder vorhandenen Flotte von AWS IoT Dingen aufzunehmen.

4. Februar 2020

[Das Abrufen von Daten wurde neu strukturiert AWS IoT SiteWise](#)

[Der Bereich zum Abrufen von Daten wurde in zwei Bereiche der obersten Ebene umstrukturiert: Abfragen von Werten und Aggregaten von Vermögenswerten und Interaktion mit anderen Diensten. AWS](#)

21. Januar 2020

[Tutorial zum Veröffentlichen von Eigenschaftswertaktualisierungen in Amazon DynamoDB](#)

Es wurde ein Tutorial hinzugefügt, dem Sie folgen können, um zu erfahren, wie Sie Benachrichtigungen über Eigenschaftswerte verwenden, um Asset-Daten in DynamoDB zu speichern.

8. Januar 2020

[Verwendung von Formelausdrücken](#)

Es wurde die Formelausdrucksreferenz hinzugefügt, um die Konstanten und Funktionen zu organisieren, die für die Verwendung in Transformations- und Metrikeigenschaften verfügbar sind. Umstrukturierung von [Komponenteneigenschaften](#) zu separaten Themen für jeden Eigenschaftstyp.

7. Januar 2020

[Verwenden von OPC-UA-Knotenfiltern](#)

Es wurden Informationen zur Verwendung von OPC UA-Knotenfiltern hinzugefügt, um die Leistung SiteWise des Edge-Gateways beim Hinzufügen von SiteWise Edge-Gateway-Quellen zu verbessern.

3. Januar 2020

Einen Connector aktualisieren	Es wurden Informationen zum Upgrade eines SiteWise Edge-Gateways hinzugefügt, wenn eine neue Connector-Version veröffentlicht wird.	30. Dezember 2019
SiteWise IoT-Connector Version 3 veröffentlicht	Version 3 des SiteWise IoT-Connectors ist verfügbar. Diese Version entfernt die Berechtigungsvoraussetzung für <code>iot:*</code> .	17. Dezember 2019
SiteWise IoT-Connector Version 2 veröffentlicht	Version 2 des SiteWise IoT-Connectors ist verfügbar. Diese Version bietet Unterstützung für mehrere geheime OPC UA-Ressourcen.	10. Dezember 2019
Dashboards erstellen () AWS CLI	Es wurden Informationen zum Erstellen eines Dashboards AWS IoT SiteWise Monitor unter Verwendung von hinzugefügt. AWS CLI	6. Dezember 2019

[AWS IoT SiteWise Version 2 veröffentlicht](#)

02. Dezember 2019

Veröffentlichte Vorschau für Version 2 von AWS IoT SiteWise. Sie können jetzt Daten über OPC UA, MQTT und HTTP aufnehmen, Ihre Daten in Asset-Hierarchien modellieren und Ihre Daten mit Monitor visualisieren. SiteWise

- Der Abschnitt [Komponentenmodellierung](#) wurde im Hinblick auf Änderungen an Komponenten, Komponentenmodellen und Komponentenhierarchien neu geschrieben.
- Der Abschnitt [zur Datenaufnahme](#) wurde aktualisiert und umfasst nun AWS IoT Greengrass Verbindungsschritte und Abschnitte zur Datenerfassung ohne Gateway.
- Der [AWS IoT SiteWise Monitor](#) Abschnitt und ein [separater Anwendungsfaden, der die Verwendung der Monitor-Webanwendung](#) zeigt, wurden hinzugefügt. SiteWise
- Es wurden die Abschnitte [Daten abfragen von AWS IoT SiteWise](#) und [Interagiere mit anderen AWS Diensten](#) hinzugefügt.

- Der Abschnitt [Erste Schritte](#) wurde umgeschrieben, um der Erfahrung der aktualisierten Demo zu entsprechen.

[AWS IoT SiteWise Version 1 veröffentlicht](#)

Erste Vorschau für Version 1 von veröffentlicht AWS IoT SiteWise.

25. Februar 2019

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.