



Guida per l'utente

AWS IoT SiteWise



AWS IoT SiteWise: Guida per l'utente

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

Che cos'è AWS IoT SiteWise?	1
Come funziona AWS IoT SiteWise	2
Inserisci dati industriali	2
Modella le risorse per contestualizzare i dati raccolti	3
Analizza utilizzando domande, allarmi e previsioni	4
Visualizza le operazioni	4
Archivia i dati	5
Integrazione con altri servizi	5
Casi d'uso per AWS IoT SiteWise	5
Settore manifatturiero	5
Settore alimentare	6
Energia e utenze	6
Lavorare con AWS SDKs	6
Concetti	7
Inizia a usare	14
Requisiti	14
Configura un account AWS	15
Registrati per un Account AWS	15
Crea un utente con accesso amministrativo	15
Usa la demo di avvio rapido	17
Crea la demo AWS IoT SiteWise	17
Eliminare la AWS IoT SiteWise demo	19
Tutorial	21
Calcola l'OEE	21
Prerequisiti	21
Come calcolare l'OEE	22
Ingerisci dati	24
Prerequisiti	25
Fase 1: Creare una AWS IoT politica	26
Fase 2: Creare qualsiasi AWS IoT cosa	28
Fase 3: Creazione di un modello di asset per dispositivi	30
Fase 4: Creazione di un modello di asset per il parco dispositivi	32
Passaggio 5: Creare e configurare una risorsa del dispositivo	34
Fase 6: Creare e configurare un parco di dispositivi	34

Passaggio 7: crea una regola in AWS IoT Core per inviare dati alle risorse del dispositivo	35
Passaggio 8: Esegui lo script del client del dispositivo	39
Passaggio 9: Pulisci le risorse dopo il tutorial	46
Visualizza e condividi i dati in Monitor SiteWise	47
Prerequisiti	49
Passaggio 1: crea un portale in Monitor SiteWise	49
Passaggio 2: accedi a un portale	53
Fase 3: Creare un progetto per un parco eolico	55
Fase 4: Creare una dashboard per visualizzare i dati del parco eolico	59
Fase 5: Esplora il portale	66
Passaggio 6: Pulisci le risorse dopo il tutorial	67
Pubblica su Amazon DynamoDB	70
Prerequisiti	70
Fase 1: Configurazione AWS IoT SiteWise per la pubblicazione degli aggiornamenti dei valori delle proprietà	71
Passaggio 2: crea una regola in Core AWS IoT	73
Fase 3: Creare una tabella DynamoDB	76
Fase 4: Configurare l'azione della regola DynamoDB	77
Fase 5: Esplora i dati in DynamoDB	78
Passaggio 6: Pulisci le risorse dopo il tutorial	79
Inserisci dati in AWS IoT SiteWise	83
Gestione dei flussi di dati	83
Configura autorizzazioni e impostazioni	85
Associa un flusso di dati a una proprietà dell'asset	86
Dissocia un flusso di dati da una proprietà dell'asset	87
Eliminare un flusso di dati	88
Aggiornare un alias di proprietà di un asset	89
Scenari comuni	90
Inserisci dati con AWS IoT SiteWise APIs	92
BatchPutAssetPropertyValue API	93
CreateBulkImportJob API	96
Usa le regole AWS IoT Core	104
Concedi l'accesso richiesto	104
Configura l'azione della regola	106
Riduci i costi con Basic Ingest	114
Usa le AWS IoT Events azioni	115

Usa AWS IoT Greengrass stream manager	116
Usa i gateway SiteWise Edge	117
Concetti chiave del gateway	117
Vantaggi dell'implementazione SiteWise di Edge	118
Ospita autonomamente un gateway	119
Requisiti	120
Crea un gateway	125
Installa il software gateway	128
Gateway V3 abilitati per MQTT	131
Stream classici, gateway V2	187
Aggiungi fonti di dati	202
Componenti per SiteWise Edge	244
Filtra le risorse	246
Supporto proxy e archivi attendibili	247
Usa APIs	253
Ospita un gateway su Siemens Industrial Edge	271
Sicurezza	272
Siemens Secure Storagee l'applicazione AWS IoT SiteWise Edge	272
Destinazioni per dispositivi Siemens Industrial Edge	273
Esegui la migrazione dall'applicazione di anteprima	274
Risoluzione dei problemi	275
AWS IoT SiteWise Registro delle modifiche delle applicazioni Edge	275
Requisiti	277
Crea un gateway	278
Creazione di una Siemens Databus user	279
Accedere all'applicazione	280
Installa l'applicazione	281
Aggiornare la configurazione di un'applicazione installata	283
Destinazioni e filtri di percorso	284
Comprendi le destinazioni	284
Comprendi i filtri di percorso	288
Aggiungi una destinazione in tempo reale	291
Aggiungi una destinazione con buffer utilizzando Amazon S3	300
Aggiungi filtri di percorso	307
Gestisci le destinazioni	310
Gestisci i gateway	316

Gestisci il tuo gateway SiteWise Edge con la console AWS IoT SiteWise	317
Gestisci i gateway SiteWise Edge utilizzando per AWS OpsHubAWS IoT SiteWise	318
Accedi al tuo gateway SiteWise Edge utilizzando le credenziali del sistema operativo locale	320
Gestisci il certificato del gateway SiteWise Edge	322
Cambia la versione dei pacchetti di componenti del gateway SiteWise Edge	323
Elenca SiteWise i gateway Edge	323
Descrivi un gateway Edge SiteWise	324
Crea un gateway Edge SiteWise	325
Aggiorna un SiteWise gateway Edge	326
Aggiorna la configurazione delle funzionalità del gateway	327
Tagga le risorse del gateway	328
Elenca i tag per un gateway	329
Rimuovere i tag da un gateway	330
Aggiorna la versione di un componente AWS IoT SiteWise	331
Eliminare un gateway SiteWise Edge	331
Gateway di backup e ripristino	333
Backup giornalieri dei dati metrici	333
Ripristina un gateway SiteWise Edge	334
Ripristina AWS IoT SiteWise i dati	335
Convalida backup e ripristini eseguiti correttamente	336
Gateway legacy ()AWS IoT Greengrass Version 1	336
Asset industriali modello	338
Stati di asset e modelli	340
Controllate lo stato di una risorsa	340
Controllate lo stato di un modello di asset o componente	342
Versioni del modello di asset	344
Recuperate la versione attiva di un modello di asset o di un modello di componente (console)	345
Recuperate la versione attiva di un modello di asset o di un modello di componente ()AWS CLI	346
Modelli compositi personalizzati (componenti)	347
Modelli compositi personalizzati in linea	348
Component-model-based modelli compositi personalizzati	350
Utilizzate i percorsi per fare riferimento alle proprietà personalizzate del modello composito	351

Configura oggetto IDs	353
Lavora con oggetti UUIDs	354
Usa esterno IDs	354
Crea modelli	356
Crea modelli di asset in AWS IoT SiteWise	357
Creazione di modelli di componenti	372
Definite le proprietà dei dati	376
Crea modelli compositi personalizzati (componenti)	458
Crea risorse	462
Creare una risorsa (console)	462
Crea una risorsa (AWS CLI)	463
Configura una nuova risorsa	465
Cerca risorse	465
Prerequisiti	465
Ricerca avanzata su Console AWS IoT SiteWise	466
Aggiorna i valori degli attributi	469
Associare e dissociare gli asset	472
Associa e dissocia gli asset (console)	472
Associa e dissocia le risorse (AWS CLI)	474
Aggiorna risorse e modelli	475
Aggiorna le risorse in AWS IoT SiteWise	476
Aggiorna i modelli di asset e i modelli dei componenti	477
Aggiorna modelli compositi personalizzati (componenti)	483
Blocco ottimistico per le scritture dei modelli di asset	486
Eliminare risorse e modelli in AWS IoT SiteWise	490
Eliminare risorse	491
Eliminare i modelli di asset	493
Operazioni in blocco con asset e modelli	495
Concetti e terminologia chiave	496
Funzionalità supportate	497
Prerequisiti per le operazioni in blocco	497
Eseguire un processo di importazione in blocco	500
Esegui un processo di esportazione in blocco	502
Monitoraggio dell'avanzamento dei lavori e gestione degli errori	506
Esempi di metadati di importazione	511
Esporta esempi di metadati	527

schema del processo di trasferimento dei metadati	529
Monitora i dati con allarmi	548
Tipi di allarmi	548
Stati di allarme	550
Proprietà dello stato di allarme	551
Definisci gli allarmi sui modelli di asset	554
Requisiti per le notifiche di allarme	557
Definisci gli allarmi AWS IoT Events	558
Definisci allarmi esterni	593
Configura gli allarmi sugli asset	595
Configura un valore di soglia (console)	595
Configura un valore di soglia (AWS CLI)	596
Configurare le impostazioni di notifica	598
Rispondi agli allarmi	600
Rispondere a un allarme (console)	601
Rispondi a un allarme (API)	604
Inserire uno stato di allarme esterno	605
Mappa i flussi di stato di allarme esterni	605
Inserisci i dati sullo stato dell'allarme	607
AWS IoT SiteWise Assistente	609
Configura l' AWS IoT SiteWise Assistente	609
Creazione di un set di dati	611
Modifica un set di dati	616
Eliminare un set di dati	618
AWS IoT SiteWise Domande dell'assistente	619
Monitora i dati con AWS IoT SiteWise Monitor	620
SiteWise Monitora i ruoli	621
Federazione SAML	623
SiteWise Monitora i concetti	624
Inizia con AWS IoT SiteWise Monitor (Classic)	626
Crea un portale	627
Configura il tuo portale	628
Invita gli amministratori	632
Aggiungere utenti del portale	635
Creazione di dashboard (CLI)	639
Attiva gli allarmi per i tuoi portali	645

Abilita il tuo portale all'edge	648
Amministra i tuoi portali	648
Inizia con AWS IoT SiteWise Monitor (AI-Aware)	658
Crea un portale	660
Configura il tuo portale	660
Amministra i tuoi portali	663
Eliminare un portale	667
Crea dashboard con AWS CLI	668
Accesso al portale	673
Crea un progetto	674
Aggiorna un progetto	675
Elimina un progetto	675
Creazione di un pannello di controllo	676
Aggiornare una dashboard	677
Eliminare una dashboard	678
Configura la dashboard	679
Interroga i dati da AWS IoT SiteWise	700
Interroga i valori degli asset correnti	701
Interroga il valore corrente di una proprietà dell'asset (console)	701
Interroga il valore corrente di una proprietà dell'asset ()AWS CLI	701
Interroga i valori storici delle proprietà degli asset	703
Interroga gli aggregati delle proprietà degli asset	705
Aggregati per una proprietà di asset (API)	705
Aggregati per una proprietà di un asset ()AWS CLI	707
AWS IoT SiteWise linguaggio di interrogazione	708
Prerequisiti	709
Riferimento al linguaggio di interrogazione	709
Interagisci con altri servizi	718
Comprendi le proprietà degli asset negli argomenti MQTT	719
Lavora con le notifiche	719
Attiva le notifiche sulle proprietà degli asset (console)	720
Attiva le notifiche sulle proprietà degli asset ()AWS CLI	720
Notifiche di interrogazione	722
Esportazione di dati su Amazon S3	725
Integra Grafana	726
Integrazione con AWS IoT TwinMaker	727

Abilitazione dell'integrazione	728
Integrazione e AWS IoT SiteWiseAWS IoT TwinMaker	728
Rileva le anomalie delle apparecchiature	729
Aggiungi una definizione di previsione (console)	731
Addestra una previsione (console)	734
Avvia o interrompi l'inferenza su una previsione (console)	735
Aggiungere una definizione di previsione (CLI)	736
Addestra una previsione e un'inferenza iniziale (CLI)	739
Addestra una previsione (CLI)	741
Avvia o interrompi l'inferenza su una previsione (CLI)	742
Gestisci l'archiviazione dei dati	745
Configurare le impostazioni di archiviazione	746
Impatto sulla conservazione dei dati	746
Configura per il livello caldo (console)	747
Configura per warm tier (AWS CLI) (.....	749
Configurazione per il livello freddo (console)	752
Configura per cold tier (AWS CLI)	755
Risolvere i problemi relativi alle impostazioni di archiviazione	760
Errore: il bucket non esiste	760
Errore: accesso negato al percorso Amazon S3	761
Errore: non è possibile assumere il ruolo ARN	761
Errore: accesso al bucket Amazon S3 interregionale non riuscito	762
Percorsi dei file e schemi di dati salvati nella fase fredda	762
Dati dell'attrezzatura (misurazioni)	762
Metriche, trasformazioni e aggregazioni	767
Metadati delle risorse	771
metadati della gerarchia degli asset	775
File di indice dei dati di archiviazione	778
Esempi di codice	779
Nozioni di base	783
Ciao AWS IoT SiteWise	784
Informazioni di base	788
Azioni	851
Sicurezza	927
Protezione dei dati	928
Riservatezza del traffico Internet	929

AWS IoT SiteWise Miglioramento del servizio Assistant Business	929
Crittografia dei dati	929
Crittografia a riposo	930
Crittografia in transito	933
Gestione delle chiavi	935
Gestione dell'identità e degli accessi	936
Destinatari	937
Autenticazione con identità	937
Come AWS IoT SiteWise funziona con IAM	941
Policy gestite	961
Ruoli collegati ai servizi	964
Imposta le autorizzazioni per gli allarmi	982
Prevenzione sostitutiva confusa tra diversi servizi in AWS IoT SiteWise	988
Risoluzione dei problemi di identità e accesso in	990
Convalida della conformità	992
Resilienza	993
Sicurezza dell'infrastruttura	994
Analisi della configurazione e delle vulnerabilità	994
Endpoint VPC	995
Operazioni API supportate	995
Creazione di un endpoint VPC dell'interfaccia	998
Accesso AWS IoT SiteWise tramite un endpoint VPC di interfaccia	999
Creazione di una policy di endpoint VPC	1000
Best practice di sicurezza	1001
Usa le credenziali di autenticazione sui tuoi server OPC UA	1001
Utilizzate modalità di comunicazione crittografate per i vostri server OPC UA	1002
Mantieni aggiornati i tuoi componenti	1002
Crittografa il file system del tuo gateway SiteWise Edge	1002
Accesso sicuro alla configurazione edge	1002
Protezione dei dati su Siemens Industrial Edge Management	1003
Concedi agli utenti di SiteWise Monitor le autorizzazioni minime possibili	1003
Non esporre informazioni riservate	1003
Segui le migliori pratiche di AWS IoT Greengrass sicurezza	1004
Consulta anche	1004
Registrazione e monitoraggio	1005
Monitora i registri di servizio	1005

Gestisci l'accesso AWS IoT SiteWise	1007
Esempio: voci dei file di AWS IoT SiteWise registro	1009
Monitora i log del gateway SiteWise Edge	1009
Usa Amazon CloudWatch Logs	1010
Usa i registri di servizio	1011
Monitoraggio con i CloudWatch parametri di Amazon	1013
AWS IoT Greengrass Version 2 metriche del gateway	1014
Registra le chiamate API con AWS CloudTrail	1023
AWS IoT SiteWise informazioni in CloudTrail	1023
AWS IoT SiteWise eventi di dati in CloudTrail	1024
AWS IoT SiteWise eventi di gestione in CloudTrail	1027
Esempio: voci dei file di AWS IoT SiteWise registro	1027
Assegnazione di tag alle risorse	1029
Usa i tag in AWS IoT SiteWise	1029
Tagga con AWS Management Console	1029
Tagga con l'API AWS IoT SiteWise	1029
Usa i tag con le policy IAM	1031
Risoluzione dei problemi	1033
Risoluzione dei problemi di un gateway	1033
Configura e accedi ai log del gateway Edge SiteWise	1033
Risoluzione dei problemi relativi SiteWise al gateway Edge	1034
Risoluzione dei problemi relativi all'applicazione AWS IoT SiteWise Edge su Siemens Industrial Edge	1040
Risoluzione dei problemi di integrazioni open source su Edge	1041
Risoluzione dei problemi AWS IoT Greengrass	1044
Risolvere i problemi relativi a un portale	1044
Gli utenti e gli amministratori non possono accedere al portale AWS IoT SiteWise	1044
Risolvere i problemi relativi a un' AWS IoT SiteWise azione delle regole	1046
Configura i log AWS IoT Core	1046
Configura un'azione di errore di ripubblicazione	1047
Risolvi i problemi relativi alle regole	1049
Risolvere i problemi relativi a una regola (AWS IoT SiteWise)	1051
Risoluzione dei problemi relativi a una regola (DynamoDB)	1053
Risolvi i problemi di importazione ed esportazione in blocco	1056
Punti finali e quote	1058
Endpoints	1058

Quote 1058

 AWS IoT SiteWise Limiti di limitazione dell'API Assistant 1076

 Quote per il rilevamento delle anomalie 1077

Cronologia dei documenti 1078

..... mc

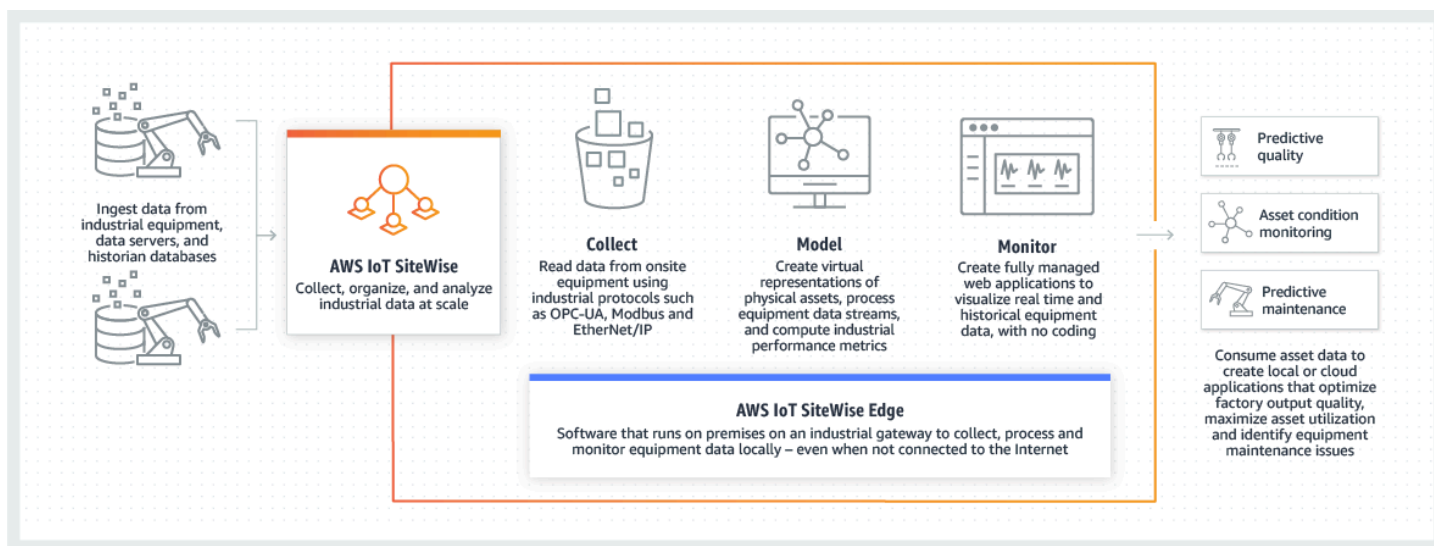
Che cos'è AWS IoT SiteWise?

AWS IoT SiteWise è un servizio gestito con cui è possibile raccogliere, archiviare, organizzare e monitorare i dati delle apparecchiature industriali su larga scala per aiutarvi a prendere decisioni migliori basate sui dati. È possibile utilizzarlo AWS IoT SiteWise per monitorare le operazioni tra le strutture, calcolare rapidamente metriche di prestazioni industriali comuni e creare applicazioni che analizzano i dati delle apparecchiature industriali per prevenire costosi problemi alle apparecchiature e ridurre le lacune nella produzione.

Con AWS IoT SiteWise Monitor, gli utenti operativi possono creare applicazioni web per visualizzare e analizzare i dati industriali in tempo reale. Per ottenere informazioni dettagliate sulle operazioni industriali, bisogna configurare e monitorare parametri quali il tempo medio tra i guasti e l'efficienza complessiva delle apparecchiature (OEE).

AWS IoT SiteWise Edge è un componente AWS IoT SiteWise che consente la raccolta, l'archiviazione e l'elaborazione di dati su dispositivi locali. Ciò è utile se hai un accesso limitato a Internet o hai bisogno di mantenere privati i tuoi dati.

Il diagramma seguente mostra l'architettura di base di AWS IoT SiteWise:



Argomenti

- [Come funziona AWS IoT SiteWise](#)
- [Casi d'uso per AWS IoT SiteWise](#)
- [Utilizzo di questo servizio con un SDK AWS](#)
- [AWS IoT SiteWise concetti](#)

Come funziona AWS IoT SiteWise

AWS IoT SiteWise offre un framework di modellazione delle risorse che è possibile utilizzare per creare rappresentazioni di dispositivi, processi e strutture industriali. Le rappresentazioni delle apparecchiature e dei processi sono denominate modelli di asset in. AWS IoT SiteWise Con i modelli di asset, definisci i dati grezzi da consumare e come trasformarli in metriche utili. [Crea e visualizza risorse e modelli per le tue operazioni industriali nella AWS IoT SiteWise console](#). Puoi anche configurare modelli di asset per raccogliere ed elaborare dati sull'edge o nel AWS cloud.

Argomenti

- [Inserisci dati industriali](#)
- [Modella le risorse per contestualizzare i dati raccolti](#)
- [Analizza utilizzando domande, allarmi e previsioni](#)
- [Visualizza le operazioni](#)
- [Archivia i dati](#)
- [Integrazione con altri servizi](#)

Inserisci dati industriali

Inizia a utilizzare AWS IoT SiteWise ingerendo dati industriali. L'acquisizione dei dati avviene in diversi modi:

- Inserimento diretto dai server in loco: utilizza protocolli come OPC UA per leggere i dati direttamente dai dispositivi in loco. Implementa il software SiteWise Edge gateway, compatibile con AWS IoT Greengrass V2, su un'ampia gamma di piattaforme come gateway industriali comuni o server virtuali. È possibile connettere fino a 100 server OPC UA a un singolo gateway. AWS IoT SiteWise Per ulteriori informazioni, consulta [AWS IoT SiteWise Requisiti del gateway Edge self-hosted](#).

Tieni presente che protocolli come Modbus TCP e Ethernet/IP (EIP) sono supportati dalla nostra partnership con Domatica nel contesto di. AWS IoT Greengrass V2

- Elaborazione dei dati Edge con pacchetti: migliora il tuo gateway SiteWise Edge aggiungendo pacchetti per abilitare funzionalità edge complete. Con SiteWise Edge, disponibile su AWS IoT Greengrass V2, l'elaborazione dei dati viene eseguita direttamente in loco prima di essere trasmessa in modo sicuro al AWS Cloud tramite un AWS IoT Greengrass flusso. Per ulteriori informazioni, consulta [Configura una sorgente OPC UA in SiteWise Edge](#).

- Inserimento adattivo tramite Amazon S3 con operazioni in blocco: quando lavori con un gran numero di asset o modelli di asset, utilizza operazioni in blocco per importare ed esportare in blocco risorse dai bucket Amazon S3. Per ulteriori informazioni, consulta [Operazioni in blocco con asset e modelli](#).
- Messaggi MQTT con regole di AWS IoT base: per i dispositivi collegati a AWS IoT Core che inviano messaggi MQTT, utilizza il motore AWS IoT Core rules per indirizzare tali messaggi a AWS IoT SiteWise. Se avete dispositivi collegati a AWS IoT Core che inviano messaggi [MQTT](#), utilizza il motore AWS IoT Core rules per indirizzare tali messaggi AWS IoT SiteWise. Per ulteriori informazioni, consulta [Inserisci dati per AWS IoT SiteWise utilizzare le regole AWS IoT Core](#).
- Inserimento di dati innescato da eventi: utilizza le azioni AWS IoT Events per configurare l'azione IoT per inviare dati AWS IoT Events a quando si verificano gli eventi. AWS IoT SiteWise Per ulteriori informazioni, consulta [Inserisci dati da AWS IoT SiteWise AWS IoT Events](#).
- AWS IoT SiteWise API: le tue applicazioni su Edge o nel cloud possono inviare dati direttamente a AWS IoT SiteWise Per ulteriori informazioni, consulta [Inserisci dati con AWS IoT SiteWise APIs](#).

Modella le risorse per contestualizzare i dati raccolti

Dopo aver acquisito i dati, puoi utilizzarli per creare rappresentazioni virtuali di risorse, processi e strutture creando modelli delle tue operazioni fisiche. Una risorsa, che rappresenta un dispositivo o un processo, trasmette flussi di dati al Cloud. AWS Le risorse possono anche significare raggruppamenti logici di dispositivi. Le gerarchie vengono create associando risorse per rispecchiare operazioni complesse. Queste gerarchie consentono alle risorse di accedere ai dati delle risorse secondarie associate. Gli asset vengono creati a partire da modelli di asset. I modelli di asset sono strutture dichiarative che standardizzano i formati degli asset. Riutilizza i componenti degli asset per l'organizzazione e la manutenibilità dei tuoi modelli. Per ulteriori informazioni, consulta [Asset industriali modello](#).

Con AWS IoT SiteWise, puoi configurare le tue risorse per trasformare i dati in entrata in metriche e trasformazioni contestuali.

- Trasforma il lavoro quando si ricevono i dati delle apparecchiature.
- Le metriche vengono calcolate a intervalli definiti dall'utente.

Le metriche e le trasformazioni sono applicabili sia alle risorse singole che a più risorse. AWS IoT SiteWise calcola automaticamente gli aggregati statistici di uso comune come media, somma

e conteggio, in vari intervalli di tempo relativi ai dati, alle metriche e alle trasformazioni delle apparecchiature.

Le risorse possono essere sincronizzate utilizzando AWS IoT TwinMaker. Per ulteriori informazioni, consulta [Integrazione e AWS IoT SiteWise](#) [AWS IoT TwinMaker](#).

Analizza utilizzando domande, allarmi e previsioni

Analizza la data raccolta AWS IoT SiteWise eseguendo interrogazioni e impostando allarmi. Puoi anche utilizzare Amazon Lookout per rilevare automaticamente le anomalie all'interno dei parametri e identificarne le cause principali.

- Imposta allarmi specifici per avvisare il team quando apparecchiature o processi si discostano dalle prestazioni ottimali, garantendo una rapida identificazione e risoluzione dei problemi. Per ulteriori informazioni, consulta [Monitora i dati con allarmi in AWS IoT SiteWise](#).
- Utilizza le operazioni AWS IoT SiteWise API per interrogare i valori correnti, i valori storici e gli aggregati delle proprietà degli asset su intervalli di tempo specifici. Per ulteriori informazioni, consulta [Interroga i dati da AWS IoT SiteWise](#).
- Usa il rilevamento delle anomalie con Amazon Lookout for Equipment per identificare e visualizzare i cambiamenti nelle apparecchiature o nelle condizioni operative. Con il rilevamento delle anomalie, puoi determinare misure di manutenzione preventiva per le tue operazioni. Questa integrazione consente ai clienti di sincronizzare i dati tra Amazon Lookout for Equipment AWS IoT SiteWise e Amazon. Per ulteriori informazioni, consulta [Rileva le anomalie con Lookout for Equipment](#).

Visualizza le operazioni

Configura SiteWise Monitor per creare applicazioni web per i tuoi dipendenti operativi. Le applicazioni web aiutano i dipendenti a visualizzare le vostre operazioni. Gestisci vari livelli di accesso per i tuoi dipendenti utilizzando IAM Identity Center o IAM. Configura accessi e autorizzazioni unici per ogni dipendente per visualizzare sottoinsiemi specifici di un'intera operazione industriale. AWS IoT SiteWise fornisce una [guida applicativa](#) per questi dipendenti per imparare a usare Monitor. SiteWise

Per ulteriori informazioni sulla visualizzazione delle operazioni, consulta. [Monitora i dati con AWS IoT SiteWise Monitor](#)

Archivia i dati

Puoi integrare lo storage di serie temporali con il tuo data lake industriale. AWS IoT SiteWise dispone di tre livelli di storage per i dati industriali:

- Un livello di archiviazione a caldo ottimizzato per applicazioni in tempo reale.
- Un livello di storage caldo ottimizzato per carichi di lavoro analitici.
- Un livello di cold storage gestito dal cliente che utilizza Amazon S3 per applicazioni di dati operativi con elevata tolleranza di latenza.

AWS IoT SiteWise ti aiuta a gestire i costi di storage conservando i dati recenti nel livello di archiviazione a caldo. Quindi, si definiscono le politiche di conservazione dei dati per spostare i dati storici su uno storage di livello caldo o freddo. Per ulteriori informazioni, consulta [Gestisci l'archiviazione dei dati in AWS IoT SiteWise](#).

Puoi anche importare ed esportare i metadati delle risorse. Per ulteriori informazioni, consulta [Metadati delle risorse](#).

Integrazione con altri servizi

AWS IoT SiteWise si integra con diversi AWS servizi per sviluppare una AWS IoT soluzione completa nel AWS cloud. Per ulteriori informazioni, consulta [Interagisci con altri AWS servizi](#).

Casi d'uso per AWS IoT SiteWise

AWS IoT SiteWise viene utilizzato in una varietà di settori per molte applicazioni di raccolta e analisi di dati industriali.

Raccogli dati in modo coerente da tutte le tue fonti per risolvere rapidamente i problemi. AWS IoT SiteWise offre il monitoraggio remoto per raccogliere i dati direttamente in loco o raccogliarli da più fonti in molte strutture. AWS IoT SiteWise fornisce la flessibilità necessaria per le soluzioni di dati IoT industriali.

Settore manifatturiero

AWS IoT SiteWise può semplificare il processo di raccolta e utilizzo dei dati dalle apparecchiature per individuare e ridurre al minimo le inefficienze, migliorando le operazioni industriali. AWS IoT SiteWise aiuta a raccogliere dati dalle linee e dalle apparecchiature di produzione. Con AWS IoT SiteWise, puoi trasferire i dati sul AWS cloud e creare metriche prestazionali per apparecchiature

e processi specifici. Puoi utilizzare le metriche prodotte per comprendere l'efficacia complessiva delle tue operazioni e identificare opportunità di innovazione e miglioramento. È inoltre possibile visualizzare il processo di produzione e identificare carenze nelle apparecchiature e nei processi, lacune di produzione o difetti del prodotto.

Settore alimentare

Le strutture dell'industria alimentare gestiscono un'enorme quantità di processi di trasformazione del cibo, tra cui la macinatura del grano per la farina, la macellazione e il confezionamento della carne, così come l'assemblaggio, la cottura e la congelazione di cibi precotti. Gli impianti di trasformazione alimentare spesso si estendono su più sedi e gli operatori degli impianti e delle apparecchiature si trovano in una posizione centralizzata per monitorare processi e apparecchiature. Ad esempio, le unità di refrigerazione valutano la manipolazione e la scadenza degli ingredienti. Monitorano la produzione di rifiuti tra le strutture per garantire l'efficienza operativa. Con AWS IoT SiteWise, puoi raggruppare i flussi di dati dei sensori provenienti da più sedi per linea di produzione e strutture in modo che i tuoi ingegneri di processo possano comprendere meglio e apportare miglioramenti tra le strutture.

Energia e utenze

Con AWS IoT SiteWise, è possibile risolvere i problemi relativi alle apparecchiature in modo più semplice ed efficiente. È possibile monitorare le prestazioni degli asset da remoto e in tempo reale. Accedi ai dati storici delle apparecchiature da qualsiasi luogo per individuare potenziali problemi, inviare risorse accurate e prevenire e risolvere i problemi più rapidamente.

Utilizzo di questo servizio con un SDK AWS

AWS i kit di sviluppo software (SDKs) sono disponibili per molti linguaggi di programmazione più diffusi. Ogni SDK fornisce un'API, esempi di codice, e documentazione che facilitano agli sviluppatori la creazione di applicazioni nel loro linguaggio preferito.

Documentazione sugli SDK	Esempi di codice
AWS SDK per C++	AWS SDK per C++ esempi di codice
AWS CLI	AWS CLI esempi di codice
AWS SDK per Go	AWS SDK per Go esempi di codice

Documentazione sugli SDK	Esempi di codice
AWS SDK per Java	AWS SDK per Java esempi di codice
AWS SDK per JavaScript	AWS SDK per JavaScript esempi di codice
AWS SDK per Kotlin	AWS SDK per Kotlin esempi di codice
AWS SDK per .NET	AWS SDK per .NET esempi di codice
AWS SDK per PHP	AWS SDK per PHP esempi di codice
AWS Strumenti per PowerShell	AWS Strumenti per PowerShell esempi di codice
AWS SDK per Python (Boto3)	AWS SDK per Python (Boto3) esempi di codice
AWS SDK per Ruby	AWS SDK per Ruby esempi di codice
SDK AWS per Rust	SDK AWS per Rust esempi di codice
AWS SDK per SAP ABAP	AWS SDK per SAP ABAP esempi di codice
AWS SDK per Swift	AWS SDK per Swift esempi di codice

Esempio di disponibilità

Non riesci a trovare quello che ti serve? Richiedi un esempio di codice utilizzando il link [Provide feedback \(Fornisci un feedback\)](#) nella parte inferiore di questa pagina.

AWS IoT SiteWise concetti

Di seguito sono riportati i concetti fondamentali di AWS IoT SiteWise:

Aggregazione

Gli aggregati sono metriche fondamentali, o misurazioni, che calcolano AWS IoT SiteWise automaticamente tutti i dati delle serie temporali. Per ulteriori informazioni, consulta [Interroga gli aggregati delle proprietà degli asset in AWS IoT SiteWise](#).

Asset

Quando si inseriscono o si inseriscono dati AWS IoT SiteWise dalle apparecchiature industriali, tutti i dispositivi, le apparecchiature e i processi vengono visualizzati come risorse. A ogni risorsa sono associati dati. Ad esempio, un'apparecchiatura potrebbe avere un numero di serie, un'ubicazione, una marca e un modello e una data di installazione. Potrebbe anche avere valori di serie temporali relativi a disponibilità, prestazioni, qualità, temperatura, pressione e altro. Raggruppa le risorse in gerarchie, consentendo alle risorse di accedere ai dati archiviati nelle risorse secondarie. Per ulteriori informazioni, consulta [Asset industriali modello](#).

Gerarchia di asset

Imposta gerarchie di asset per creare rappresentazioni logiche delle tue operazioni industriali. A tale scopo, definite una gerarchia in un modello di asset e associate gli asset creati da quel modello alla gerarchia specificata. Le metriche degli asset principali possono combinare i dati delle proprietà degli asset secondari, consentendovi di calcolare metriche che offrono informazioni dettagliate sull'attività complessiva o su una parte specifica di essa. Per ulteriori informazioni, consulta [Definite le gerarchie dei modelli di asset](#).

Modello di asset

Ogni risorsa viene creata utilizzando un modello di asset. I modelli di asset sono strutture che definiscono e standardizzano il formato degli asset. Garantiscono informazioni coerenti su più risorse dello stesso tipo, consentendoti di gestire i dati in risorse che rappresentano gruppi di dispositivi. In ogni modello di asset, è possibile definire [attributi](#), input serie temporali ([misure](#)), trasformazioni di serie temporali ([trasformazioni](#)), aggregazioni di serie temporali ([parametri](#)) e [gerarchie di asset](#). Per ulteriori informazioni, consulta [Asset industriali modello](#).

Decidete dove vengono elaborate le proprietà del vostro modello di asset configurando il modello di asset per l'edge. Utilizzate questa funzionalità per gestire e monitorare i dati degli asset sui dispositivi locali.

Proprietà di asset

Le proprietà degli asset sono le strutture all'interno di ciascun asset che contengono dati industriali. Ogni proprietà ha un tipo di dati e può anche avere un'unità. Una proprietà può essere un [attributo](#), una [misura](#), una [trasformazione](#) o un [parametro](#). Per ulteriori informazioni, consulta [Definite le proprietà dei dati](#).

Configura le proprietà degli asset per l'elaborazione periferica. Per ulteriori informazioni sull'elaborazione dei dati all'edge, consulta [Configura una sorgente OPC UA in SiteWise Edge](#).

Attributo

Gli attributi sono proprietà di una risorsa che in genere rimangono costanti, come il produttore del dispositivo o la posizione del dispositivo. Gli attributi possono avere valori preimpostati. Ogni risorsa creata da un modello di asset include i valori predefiniti degli attributi definiti in quel modello. Per ulteriori informazioni, consulta [Definisci dati statici \(attributi\)](#).

Dashboard

Ogni progetto contiene un set di pannelli di controllo. I pannelli di controllo forniscono un set di visualizzazioni per i valori di un set di asset. I proprietari del progetto creano i pannelli di controllo e le visualizzazioni contenute. Quando il proprietario di un progetto è pronto a condividere il set di pannelli di controllo, il proprietario può invitare i visualizzatori al progetto, consentendo loro di accedere a tutti i pannelli di controllo del progetto. Se si desidera un set diverso di visualizzatori per pannelli di controllo diversi, è necessario dividere i pannelli di controllo tra i progetti. Quando gli spettatori guardano le dashboard, possono personalizzare l'intervallo di tempo per esaminare dati specifici.

Set di dati

I set di dati sono raccolte di dati che rappresentano dati di serie temporali, dati e non-time-series dati non relativi alle apparecchiature, come orari dei turni, registri di manutenzione e database dei dipendenti. Supportano dati esterni e utilizzano AWS IoT SiteWise funzionalità analitiche. Include fonti di set di dati, schema del set di dati e parametri del set di dati. L' AWS IoT SiteWise Assistente utilizza set di dati che utilizzano indici Amazon Kendra.

Flusso di dati

Inserisci o inserisci dati industriali AWS IoT SiteWise ancor prima di creare modelli e asset di asset. AWS IoT SiteWise genera automaticamente flussi di dati per raccogliere flussi di dati grezzi dalle apparecchiature.

Alias del flusso di dati

Gli alias del flusso di dati consentono di identificare facilmente un flusso di dati. Ad esempio, l'alias `server1-windfarm/3/turbine/7/temperature` indica i valori di temperatura provenienti dalla turbina #7 del parco eolico #3. Il termine `server1` è il nome della fonte di dati che aiuta a identificare il server OPC UA ed `server1-` è un prefisso allegato a tutti i flussi di dati riportati da questo server OPC UA.

Associazione del flusso di dati

Dopo aver creato modelli e asset di asset, associate i flussi di dati alle proprietà degli asset definite negli asset per strutturare i dati. AWS IoT SiteWise può quindi utilizzare modelli e risorse di asset per gestire i dati in entrata dai flussi di dati. È inoltre possibile dissociare i flussi di dati dalle proprietà degli asset. Per ulteriori informazioni, consulta [Gestisci i flussi di dati per AWS IoT SiteWise](#).

Destinazioni

Le destinazioni in SiteWise Edge rappresentano gli endpoint a cui desideri inviare la telemetria o i dati elaborati. SiteWise Edge supporta l' AWS IoT SiteWise hot tier, l'ingestione bufferizzata o un bucket Amazon S3 come destinazioni. È possibile configurare le destinazioni per sottoscrivere argomenti MQTT specifici utilizzando filtri di percorso. Per ulteriori informazioni, consulta [Comprendi le destinazioni Edge AWS IoT SiteWise](#).

Formula

Ogni proprietà di [trasformazione](#) e [metrica](#) viene fornita con una formula che delinea come la proprietà trasforma o aggrega i dati. Queste formule includono input di proprietà, operatori e funzioni offerte da AWS IoT SiteWise. Per ulteriori informazioni, consulta [Usa espressioni di formule](#).

Misura

Le misurazioni sono proprietà di una risorsa che rappresenta i flussi di dati grezzi delle serie temporali del sensore provenienti da un dispositivo o un'apparecchiatura. Per ulteriori informazioni, consulta [Definisci i flussi di dati provenienti dalle apparecchiature \(misurazioni\)](#).

Parametro

Le metriche sono proprietà di una risorsa che rappresentano dati aggregati di serie temporali. Ogni metrica è accompagnata da un'espressione matematica ([formula](#)) che descrive come aggregare i punti dati e da un intervallo di tempo per il calcolo di tale aggregazione. Le metriche generano un singolo punto dati per ogni intervallo di tempo specificato. Per ulteriori informazioni, consulta [Aggrega i dati provenienti da proprietà e altre risorse \(metriche\)](#).

MQTT

MQTT (Message Queuing Telemetry Transport) è un protocollo di messaggistica leggero per sensori e dispositivi.

Pacchetti

SiteWise Gli edge gateway utilizzano pacchetti per determinare come raccogliere, elaborare e instradare i dati. Per ulteriori informazioni sui pacchetti disponibili per il gateway SiteWise Edge, consulta [the section called “Usa i pacchetti”](#).

Pacchetto di raccolta dati

Utilizza il pacchetto di raccolta dati in modo che il tuo gateway SiteWise Edge possa raccogliere i dati industriali e indirizzarli verso la AWS destinazione di tua scelta.

Pacchetto per l'elaborazione dei dati

Utilizza il pacchetto di elaborazione dati per elaborare, archiviare e recuperare i dati all'edge per un massimo di 30 giorni. Scambia dati elaborati dall'edge da e verso applicazioni locali tramite Edge. SiteWise APIs

OPC UA

OPC UA (Open Platform Communications Unified Architecture) è un protocollo di comunicazione per l'automazione industriale.

Filtri Path

Utilizza i filtri di percorso all'interno di un gateway per abbonarti agli argomenti MQTT e pubblicarli nelle destinazioni AWS IoT SiteWise supportate. Le fonti, le pipeline di elaborazione dei dati e le destinazioni basate su MQTT scambiano tutti dati utilizzando argomenti MQTT su un gateway V3 dotato di MQTT ospitato autonomamente. È possibile definire filtri per argomento per specificare i dati che si desidera importare o indirizzare verso destinazioni diverse.

Portal

Un AWS IoT SiteWise Monitor portale è un'applicazione web che puoi utilizzare per visualizzare e condividere i tuoi AWS IoT SiteWise dati. Un portale ha uno o più amministratori e contiene zero o più progetti.

Amministratore del portale

Ogni portale SiteWise Monitor dispone di uno o più amministratori del portale. Gli amministratori del portale utilizzano il portale per creare progetti che contengono raccolte di asset e pannelli di controllo. L'amministratore del portale assegna quindi asset e proprietari a ciascun progetto. Controllando l'accesso al progetto, gli amministratori del portale specificano quali asset possono vedere i proprietari e i visualizzatori del progetto.

Progetto

Ogni portale SiteWise Monitor contiene una serie di progetti. Ogni progetto ha un sottoinsieme di asset AWS IoT SiteWise associati. I proprietari del progetto creano uno o più pannelli di controllo per fornire un modo coerente per visualizzare i dati associati agli asset. I proprietari del progetto possono invitare i visualizzatori al progetto per consentire loro di visualizzare gli asset e i pannelli di controllo del progetto. Il progetto è l'unità di condivisione di base all'interno di SiteWise Monitor. I proprietari del progetto possono invitare gli utenti a cui l'AWS amministratore ha concesso l'accesso al portale. Un utente deve avere accesso a un portale prima che un progetto del portale possa essere condiviso con l'utente.

Proprietario del progetto

Ogni progetto SiteWise Monitor ha dei proprietari. I proprietari dei progetti creano visualizzazioni sotto forma di pannelli di controllo al fine di rappresentare i dati operativi in modo coerente. Una volta che i pannelli di controllo sono pronti per la condivisione, il proprietario del progetto può invitare i visualizzatori al progetto. I proprietari del progetto possono anche assegnare altri proprietari al progetto. I proprietari dei progetti possono configurare le soglie e le impostazioni di notifica per gli allarmi.

Visualizzatore del progetto

Ogni progetto SiteWise Monitor ha dei visualizzatori. I visualizzatori del progetto possono connettersi al portale per visualizzare i pannelli di controllo creati dai proprietari del progetto. In ogni dashboard, i visualizzatori dei progetti possono regolare l'intervallo di tempo per comprendere meglio i dati operativi. I visualizzatori del progetto possono visualizzare solo i pannelli di controllo nei progetti a cui hanno accesso. I visualizzatori del progetto possono confermare e posticipare gli allarmi.

Alias di proprietà

Avete la possibilità di creare alias sulle proprietà delle risorse, come il percorso del flusso di dati del server OPC UA (ad esempio, /company/windfarm/3/turbine/7/temperature), semplificando l'identificazione di una proprietà della risorsa durante l'acquisizione o il recupero dei dati dell'asset. Quando utilizzate un [gateway SiteWise Edge](#) per importare dati dai server, gli alias delle proprietà devono corrispondere ai percorsi dei flussi di dati grezzi. Per ulteriori informazioni, consulta [Gestisci i flussi di dati per AWS IoT SiteWise](#).

Notifica di proprietà

Quando abilitate le notifiche di proprietà per una proprietà di un asset, AWS IoT SiteWise pubblica un messaggio MQTT AWS IoT Core ogni volta che tale proprietà riceve un nuovo valore. Il

payload del messaggio include dettagli sull'aggiornamento del valore di quella proprietà. Utilizzate le notifiche sul valore delle proprietà per creare soluzioni che collegano i dati industriali AWS IoT SiteWise ad altri AWS servizi. Per ulteriori informazioni, consulta [Interagisci con altri AWS servizi](#).

SiteWise Gateway Edge

Un gateway SiteWise Edge viene installato presso la sede del cliente per raccogliere, gestire e indirizzare i dati. Un gateway SiteWise Edge si collega alle fonti di dati industriali tramite vari protocolli per raccogliere ed elaborare i dati, inviandoli al AWS cloud. SiteWise I gateway Edge possono anche connettersi alle fonti di dati dei partner. Per ulteriori informazioni, consulta [Usa i gateway AWS IoT SiteWise Edge](#).

Trasformazione

Le trasformazioni sono proprietà di una risorsa che rappresentano dati di serie temporali trasformati. Ogni trasformazione è accompagnata da un'espressione matematica ([formula](#)) che specifica come convertire i punti dati da un modulo all'altro. I punti dati trasformati mantengono una one-to-one relazione con i punti dati di input. Per ulteriori informazioni, consulta [Trasforma i dati \(trasformazioni\)](#).

Visualizzazione

In ogni dashboard, i proprietari del progetto decidono come visualizzare le proprietà e gli allarmi delle risorse associate al progetto. La disponibilità potrebbe essere rappresentata come un grafico a linee, mentre altri valori potrebbero essere visualizzati come grafici a barre o indicatori chiave di prestazione (KPIs). Gli allarmi vengono visualizzati al meglio come griglie di stato e linee temporali di stato. I proprietari del progetto personalizzano ogni visualizzazione per fornire la migliore comprensione dei dati per l'asset.

Inizia con AWS IoT SiteWise

Con AWS IoT SiteWise, puoi raccogliere, organizzare, analizzare e visualizzare i tuoi dati.

AWS IoT SiteWise fornisce una demo che puoi utilizzare per esplorare il servizio senza configurare una vera fonte di dati. Per ulteriori informazioni, consulta [Usa la AWS IoT SiteWise demo](#).

Puoi completare i seguenti tutorial per esplorare alcune funzionalità di: AWS IoT SiteWise

- [Inserisci dati AWS IoT SiteWise da oggetti AWS IoT](#)
- [Visualizza e condividi i dati dei parchi eolici in Monitor SiteWise](#)
- [Pubblica aggiornamenti dei valori delle proprietà su Amazon DynamoDB](#)

Per ulteriori informazioni, consulta i seguenti argomenti: AWS IoT SiteWise

- [Inserisci dati in AWS IoT SiteWise](#)
- [Asset industriali modello](#)
- [Configura le funzionalità edge su AWS IoT SiteWise Edge](#)
- [Monitora i dati con AWS IoT SiteWise Monitor](#)
- [Interroga i dati da AWS IoT SiteWise](#)
- [Interagisci con altri AWS servizi](#)

Argomenti

- [Requisiti](#)
- [Configura un account AWS](#)

Requisiti

Devi avere un AWS account per iniziare AWS IoT SiteWise. Se non ne hai uno, consulta la sezione seguente per istruzioni su come configurare un account.

Usa una regione dove AWS IoT SiteWise è disponibile. Per ulteriori informazioni, consulta [Endpoint e quote per AWS IoT SiteWise](#). È possibile utilizzare il selettore della regione in AWS Management Console per passare a una di queste regioni.

Configura un account AWS

Argomenti

- [Registrati per un Account AWS](#)
- [Crea un utente con accesso amministrativo](#)

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la <https://portal.aws.amazon.com/billing/registrazione>.
2. Segui le istruzioni online.

Parte della procedura di registrazione prevede la ricezione di una telefonata o di un messaggio di testo e l'immissione di un codice di verifica sulla tastiera del telefono.

Quando ti iscrivi a un Account AWS, Utente root dell'account AWS viene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire [attività che richiedono l'accesso di un utente root](#).

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a <https://aws.amazon.com> e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi [AWS Management Console](#) come proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

Per informazioni sull'accesso utilizzando un utente root, consulta la pagina [Signing in as the root user](#) della Guida per l'utente di Accedi ad AWS .

2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

Per istruzioni, consulta [Abilitare un dispositivo MFA virtuale per l'utente Account AWS root \(console\)](#) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

Per istruzioni, consulta [Abilitazione di AWS IAM Identity Center](#) nella Guida per l'utente di AWS IAM Identity Center .

2. In IAM Identity Center, assegna l'accesso amministrativo a un utente.

Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta [Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory](#) nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta [AWS Accedere al portale di accesso](#) nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni conforme alla best practice dell'applicazione di autorizzazioni con il privilegio minimo.

Segui le istruzioni riportate nella pagina [Creazione di un set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center .

2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

Per istruzioni, consulta [Aggiungere gruppi](#) nella Guida per l'utente di AWS IAM Identity Center .

Usa la AWS IoT SiteWise demo

Puoi esplorare facilmente AWS IoT SiteWise utilizzando la AWS IoT SiteWise demo. AWS IoT SiteWise fornisce la demo come AWS CloudFormation modello che è possibile implementare per creare modelli di asset, risorse e un portale SiteWise Monitor e generare dati di esempio per un massimo di una settimana.

Important

Una volta creata la demo, inizierai a farti pagare per le risorse create e utilizzate da questa demo.

Argomenti

- [Crea la demo AWS IoT SiteWise](#)
- [Eliminare la AWS IoT SiteWise demo](#)

Crea la demo AWS IoT SiteWise

Puoi creare la AWS IoT SiteWise demo dalla AWS IoT SiteWise console.

Note

La demo crea funzioni Lambda, una regola CloudWatch Events e i ruoli AWS Identity and Access Management (IAM) richiesti per la demo. Potresti vedere queste risorse nel tuo AWS account. Ti consigliamo di conservare queste risorse finché non hai terminato la demo. Se elimini le risorse, la demo potrebbe smettere di funzionare correttamente.

Per creare la demo nella AWS IoT SiteWise console

1. Vai alla [AWS IoT SiteWise console](#) e trova la SiteWise demo nell'angolo in alto a destra della pagina.
2. (Facoltativo) In SiteWise demo, modifica il campo Giorni di conservazione delle risorse demo per specificare per quanti giorni conservare la demo prima di eliminarla.

3. (Facoltativo) Per creare un portale di SiteWise monitoraggio per monitorare i dati di esempio, procedi come segue.

 Note

Ti verranno addebitate le risorse di SiteWise Monitor create e utilizzate in questa demo. Per ulteriori informazioni, consulta [SiteWise Monitor](#) nella sezione AWS IoT SiteWise Prezzi.

- a. Scegli Monitora le risorse.
- b. Scegli Autorizzazione.
- c. Scegli un ruolo IAM esistente che consenta agli utenti IAM federati di accedere al portale.

 Important

Il tuo ruolo IAM deve avere le seguenti autorizzazioni.

```
{
  "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": "*"
    }
  ]
}
```

Per ulteriori informazioni su come lavorare con SiteWise Monitor, consulta [Cos'è AWS IoT SiteWise Monitor?](#) nella Guida all'AWS IoT SiteWise Monitor applicazione.

4. Seleziona Create demo (Crea demo).

La creazione della demo richiederà circa tre minuti. Nel caso in cui l'operazione non riuscisse, verifica le autorizzazioni, probabilmente insufficienti, del tuo account. Passa a un account con autorizzazioni amministrative oppure esegui la procedura seguente per eliminare la demo e riprovare:

a. Scegli Delete demo (Elimina demo).

L'eliminazione della demo richiede circa 15 minuti.

b. Se la demo non viene eliminata, apri la [AWS CloudFormation console](#), scegli lo stack denominato lo TSite WiseDemoAssets e scegli Elimina nell'angolo in alto a destra.

c. Se la demo non riesce a eliminare nuovamente, segui i passaggi indicati nella AWS CloudFormation console per saltare le risorse che non sono state eliminate e riprova.

5. Una volta creata correttamente la demo, puoi esplorare le risorse e i dati della demo nella [AWS IoT SiteWise console](#).

Eliminare la AWS IoT SiteWise demo

La AWS IoT SiteWise demo si elimina automaticamente dopo una settimana o dopo il numero di giorni che hai scelto se hai creato lo stack demo dalla AWS CloudFormation console. Puoi eliminare la demo prima se hai finito di usare le risorse della demo. Inoltre puoi eliminare la demo se la demo non viene creata. Attieniti alla seguente procedura per eliminare manualmente la demo.

Per eliminare la demo AWS IoT SiteWise

1. Passare alla [console AWS CloudFormation](#).
2. Scegliere IoTSiteWiseDemoAssetsdall'elenco degli Stack.
3. Scegli Elimina.

Quando si elimina lo stack, tutte le risorse create per la demo vengono eliminate.

4. Nella finestra di dialogo di conferma scegliere Delete stack (Elimina stack).

L'eliminazione dello stack richiede circa 15 minuti. Se la demo non riuscisse a finalizzare l'eliminazione, seleziona nuovamente Delete (Elimina) nell'angolo in alto a destra. Se la demo non riesce a eliminare nuovamente, segui i passaggi indicati nella AWS CloudFormation console per ignorare le risorse che non sono state eliminate e riprova.

AWS IoT SiteWise tutorial

Benvenuto nella pagina AWS IoT SiteWise dei tutorial. Questa crescente raccolta di tutorial ti fornisce le conoscenze e le competenze necessarie per navigare nelle complessità di AWS IoT SiteWise. Questi tutorial offrono una vasta gamma di argomenti di base per soddisfare le tue esigenze. Man mano che approfondisci i tutorial, scopri informazioni preziose su vari aspetti di AWS IoT SiteWise.

Ogni tutorial utilizza un esempio di attrezzatura specifico. Questi tutorial sono destinati agli ambienti di test e utilizzano nomi di società, modelli, risorse, proprietà fittizi e così via. Il loro scopo è di fornire linee guida di carattere generico. I tutorial non sono destinati all'uso diretto in un ambiente di produzione senza un'attenta revisione e adattamento per soddisfare le esigenze specifiche dell'organizzazione.

Argomenti

- [Calcola l'OEE in AWS IoT SiteWise](#)
- [Inserisci dati AWS IoT SiteWise da oggetti AWS IoT](#)
- [Visualizza e condividi i dati dei parchi eolici in Monitor SiteWise](#)
- [Pubblica aggiornamenti dei valori delle proprietà su Amazon DynamoDB](#)

Calcola l'OEE in AWS IoT SiteWise

Questo tutorial fornisce un esempio specifico di come calcolare l'OEE (Overall Equipment Effectiveness, efficienza complessiva delle apparecchiature) di un processo di produzione. Trattandosi di un esempio, i tuoi calcoli o le formule reali dell'OEE potrebbero variare rispetto a quelli mostrati. In generale, l'OEE è definito da $\text{Availability} * \text{Quality} * \text{Performance}$. Per ulteriori informazioni sul calcolo dell'OEE, consulta [Overall equipment effectiveness](#) su Wikipedia.

Prerequisiti

Per completare questo tutorial, è necessario configurare l'acquisizione di dati per un dispositivo con i seguenti tre flussi di dati:

- **Equipment_State**— Un codice numerico che rappresenta lo stato della macchina, ad esempio inattività, guasto, arresto pianificato o funzionamento normale.
- **Good_Count**— Un flusso di dati in cui ogni punto dati contiene il numero di operazioni riuscite a partire dall'ultimo punto dati.

- **Bad_Count**— Un flusso di dati in cui ogni punto dati contiene il numero di operazioni non riuscite dall'ultimo punto dati.

Per configurare l'acquisizione di dati, consulta [Inserisci dati in AWS IoT SiteWise](#). In assenza di un'operazione industriale disponibile, è possibile creare uno script che generi e carichi dati esemplificativi tramite l'API di AWS IoT SiteWise .

Come calcolare l'OEE

In questo tutorial, si crea un modello di asset che calcola l'OEE in base a tre flussi di dati di input: **Equipment_State**, **Good_Count** e **Bad_Count**. In questo esempio, prendiamo in esame una macchina generica per il packaging, come quelle utilizzate per il confezionamento dello zucchero, delle patatine o della vernice. Nella [AWS IoT SiteWise console](#), crea un modello di AWS IoT SiteWise asset con le seguenti misurazioni, trasformazioni e metriche. Quindi, potete creare una risorsa per rappresentare la macchina per l'imballaggio e osservare come AWS IoT SiteWise calcola l'OEE.

Definisci le seguenti [misurazioni](#) per rappresentare i flussi di dati non elaborati provenienti dalla macchina confezionatrice.

Misurazioni

- **Equipment_State**— Un flusso di dati (o misurazione) che fornisce lo stato attuale della macchina confezionatrice in codici numerici:
 - **1024**— La macchina è inattiva.
 - **1020**— Un guasto, ad esempio un errore o un ritardo.
 - **1000**— Una sosta pianificata.
 - **1111**— Un'operazione normale.
- **Good_Count**— Un flusso di dati in cui ogni punto dati contiene il numero di operazioni riuscite a partire dall'ultimo punto dati.
- **Bad_Count**— Un flusso di dati in cui ogni punto dati contiene il numero di operazioni non riuscite dall'ultimo punto dati.

Utilizzando il flusso di dati di misurazione **Equipment_State** e i relativi codici, è possibile definire le seguenti [trasformazioni](#) (o misurazioni derivate). Le trasformazioni hanno una one-to-one relazione con le misurazioni grezze.

Trasformazioni

- `Idle = eq(Equipment_State, 1024)`— Un flusso di dati trasformato che contiene lo stato di inattività della macchina.
- `Fault = eq(Equipment_State, 1020)`— Un flusso di dati trasformato che contiene lo stato di guasto della macchina.
- `Stop = eq(Equipment_State, 1000)`— Un flusso di dati trasformato che contiene lo stato di arresto pianificato della macchina.
- `Running = eq(Equipment_State, 1111)`— Un flusso di dati trasformato che contiene il normale stato operativo della macchina.

Utilizzando le misurazioni non elaborate e quelle trasformate, è possibile definire i seguenti [parametri](#) che aggregano i dati della macchina per intervalli di tempo specificati. Per tutti i parametri che si vanno a definire in questa sezione bisogna scegliere lo stesso intervallo di tempo.

Metriche

- `Successes = sum(Good_Count)`— Il numero di pacchi riempiti con successo nell'intervallo di tempo specificato.
- `Failures = sum(Bad_Count)`— Il numero di pacchi riempiti senza successo nell'intervallo di tempo specificato.
- `Idle_Time = statetime(Idle)`— Il tempo di inattività totale della macchina (in secondi) per intervallo di tempo specificato.
- `Fault_Time = statetime(Fault)`— Il tempo totale di guasto della macchina (in secondi) per intervallo di tempo specificato.
- `Stop_Time = statetime(Stop)`— Il tempo di arresto totale pianificato della macchina (in secondi) per intervallo di tempo specificato.
- `Run_Time = statetime(Running)`— Il tempo totale di funzionamento della macchina (in secondi) senza problemi per un intervallo di tempo specificato.
- `Down_Time = Idle_Time + Fault_Time + Stop_Time`— Il tempo di inattività totale della macchina (in secondi) nell'intervallo di tempo specificato, calcolato come somma degli stati della macchina diversi da `Run_Time`.
- `Availability = Run_Time / (Run_Time + Down_Time)`— Il tempo di attività della macchina o la percentuale di tempo programmato in cui la macchina è disponibile a funzionare nell'intervallo di tempo specificato.

- $Quality = Successes / (Successes + Failures)$ — La percentuale della macchina di imballaggi riempiti con successo negli intervalli di tempo specificati.
- $Performance = ((Successes + Failures) / Run\_Time) / Ideal\_Run\_Rate$ — Le prestazioni della macchina nell'intervallo di tempo specificato, espresse in percentuale rispetto alla velocità di esecuzione ideale (in secondi) per il processo.

Ad esempio, l'`Ideal_Run_Rate` potrebbero essere 60 pacchetti al minuto (1 pacchetto al secondo). Se il `Ideal_Run_Rate` valore è espresso al minuto o all'ora, è necessario dividerlo per il fattore di conversione unitario appropriato, espresso `Run_Time` in secondi.

- $OEE = Availability * Quality * Performance$ — L'efficacia complessiva dell'attrezzatura della macchina nell'intervallo di tempo specificato. Questa formula calcola l'OEE come una frazione su 1.

Note

Se l'OEE è definito come una trasformazione, i valori di output vengono calcolati per ciascuno dei valori di input. Esiste la possibilità di generare valori imprevisti poiché la valutazione della trasformazione considera gli ultimi valori disponibili per tutte le proprietà che contribuiscono alla formula. Per gli aggiornamenti delle proprietà con lo stesso timestamp, i valori di output possono essere sovrascritti dagli aggiornamenti di altre proprietà in entrata. Ad esempio, quando si calcolano Disponibilità, Qualità e Prestazioni, l'OEE viene calcolato con gli ultimi punti dati disponibili per le altre due proprietà. Questi valori che contribuiscono condividono i timestamp e causano valori di output errati dell'OEE. L'ordine non è garantito per il calcolo delle trasformazioni.

Inserisci dati AWS IoT SiteWise da oggetti AWS IoT

Scopri come importare dati AWS IoT SiteWise da una serie di dispositivi utilizzando AWS IoT le ombre dei dispositivi in questo tutorial. Le ombre dei dispositivi sono oggetti JSON che memorizzano le informazioni sullo stato corrente di un dispositivo. AWS IoT Per ulteriori informazioni, consulta [Device shadow service nella Device Shadow](#) AWS IoT Guide.

Dopo aver completato questo tutorial, puoi impostare un'operazione in AWS IoT SiteWise base agli AWS IoT elementi. Utilizzando AWS IoT le cose, è possibile integrare l'operazione con altre utili funzionalità di AWS IoT. Ad esempio, è possibile configurare AWS IoT le funzionalità per eseguire le seguenti attività:

- Configura regole aggiuntive per lo streaming di dati verso [AWS IoT Events](#) [Amazon DynamoDB](#) e altri servizi. AWS Per ulteriori informazioni, consulta [Rules](#) nella AWS IoT Developer Guide.
- Indicizza, cerca e aggrega i dati del tuo dispositivo con il servizio di indicizzazione del AWS IoT parco veicoli. Per ulteriori informazioni, consulta il [servizio di indicizzazione del parco veicoli nella Guida per gli sviluppatori](#). AWS IoT
- Controlla e proteggi i tuoi dispositivi con. AWS IoT Device Defender Per ulteriori informazioni, consulta la sezione [AWS IoT Device Defender](#) nella Guida per gli sviluppatori di AWS IoT .

In questo tutorial, imparerai a trasferire i dati dalle ombre dei dispositivi AWS IoT Things agli asset in esso contenuti. AWS IoT SiteWise A tale scopo, create uno o più AWS IoT elementi ed eseguite uno script che aggiorna l'ombra del dispositivo di ogni elemento con i dati sull'utilizzo della CPU e della memoria. In questo tutorial usi i dati di utilizzo della CPU e della memoria per imitare i dati realistici del sensore. Quindi, crei una regola con un' AWS IoT SiteWise azione che invia questi dati a una risorsa AWS IoT SiteWise ogni volta che lo shadow del dispositivo si aggiorna. Per ulteriori informazioni, consulta [Inserisci dati per AWS IoT SiteWise utilizzare le regole AWS IoT Core](#).

Argomenti

- [Prerequisiti](#)
- [Fase 1: Creare una AWS IoT politica](#)
- [Fase 2: Creare e configurare qualsiasi AWS IoT cosa](#)
- [Fase 3: Creazione di un modello di asset per dispositivi](#)
- [Fase 4: Creazione di un modello di asset per il parco dispositivi](#)
- [Passaggio 5: Creare e configurare una risorsa del dispositivo](#)
- [Fase 6: Creare e configurare un parco di dispositivi](#)
- [Passaggio 7: crea una regola in AWS IoT Core per inviare dati alle risorse del dispositivo](#)
- [Passaggio 8: Esegui lo script del client del dispositivo](#)
- [Passaggio 9: Pulisci le risorse dopo il tutorial](#)

Prerequisiti

Per completare questo tutorial, è necessario quanto segue:

- Un AWS account. Se non lo hai, consultare [Configura un account AWS](#).

- Un computer di sviluppo che esegue Windows, macOS, Linux, o Unix per accedere a AWS Management Console. Per ulteriori informazioni, consulta [Nozioni di base su AWS Management Console](#).
- Un utente AWS Identity and Access Management (IAM) con autorizzazioni di amministratore.
- Python3 installato sul tuo computer di sviluppo o installato sul dispositivo che desideri registrare come dispositivo AWS IoT.

Fase 1: Creare una AWS IoT politica

In questa procedura, crea una AWS IoT politica che consenta ai tuoi AWS IoT oggetti di accedere alle risorse utilizzate in questo tutorial.

Per creare una AWS IoT politica

1. Accedi alla [AWS Management Console](#).
2. Consulta le [AWS regioni](#) in cui AWS IoT SiteWise è supportato. Passare a una di queste regioni supportate, se necessario.
3. Passare alla [console AWS IoT](#). Se viene visualizzato il pulsante Connect device, selezionarlo.
4. Nel riquadro di navigazione a sinistra, scegli Sicurezza, quindi scegli Politiche.
5. Scegli Create (Crea).
6. Inserisci un nome per la AWS IoT politica (ad esempio, **SiteWiseTutorialDevicePolicy**).
7. In Documento di politica, scegli JSON per inserire la seguente politica in formato JSON. Sostituisci **region** e **account-id** con la tua regione e l'ID dell'account, ad esempio **us-east-1** e **123456789012**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource": "arn:aws:iot:region:account-id:client/SiteWiseTutorialDevice*"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Publish",
      "Resource": [
```

```

    "arn:aws:iot:region:account-id:topic/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/update",
    "arn:aws:iot:region:account-id:topic/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/delete",
    "arn:aws:iot:region:account-id:topic/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/get"
  ]
},
{
  "Effect": "Allow",
  "Action": "iot:Receive",
  "Resource": [
    "arn:aws:iot:region:account-id:topic/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/update/accepted",
    "arn:aws:iot:region:account-id:topic/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/delete/accepted",
    "arn:aws:iot:region:account-id:topic/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/get/accepted",
    "arn:aws:iot:region:account-id:topic/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/update/rejected",
    "arn:aws:iot:region:account-id:topic/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/delete/rejected"
  ]
},
{
  "Effect": "Allow",
  "Action": "iot:Subscribe",
  "Resource": [
    "arn:aws:iot:region:account-id:topicfilter/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/update/accepted",
    "arn:aws:iot:region:account-id:topicfilter/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/delete/accepted",
    "arn:aws:iot:region:account-id:topicfilter/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/get/accepted",
    "arn:aws:iot:region:account-id:topicfilter/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/update/rejected",
    "arn:aws:iot:region:account-id:topicfilter/$aws/things/
    ${iot:Connection.Thing.ThingName}/shadow/delete/rejected"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "iot:GetThingShadow",

```

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

Questa politica consente ai AWS IoT dispositivi di stabilire connessioni e comunicare con le ombre dei dispositivi utilizzando messaggi MQTT. Per ulteriori informazioni sui messaggi MQTT, consulta [Che cos'è MQTT?](#) . Per interagire con AWS IoT le ombre dei dispositivi, i tuoi dispositivi pubblicano e ricevono messaggi MQTT su argomenti che iniziano con. `$aws/things/thing-name/shadow/` Questa politica incorpora una variabile di policy relativa agli oggetti nota come. `${iot:Connection.Thing.ThingName}` Questa variabile sostituisce il nome dell'oggetto connesso in ogni argomento. L'`iot:Connect` istruzione stabilisce delle limitazioni sui dispositivi che possono stabilire connessioni, assicurando che la variabile `thing` policy possa sostituire solo i nomi che iniziano con. `SiteWiseTutorialDevice`

Per ulteriori informazioni, vedere [Thing policy variables](#) nella AWS IoT Developer Guide.

Note

Questa policy si applica agli oggetti i cui nomi iniziano con `SiteWiseTutorialDevice`. Per utilizzare un nome diverso per gli oggetti, è necessario aggiornare la policy di conseguenza.

8. Scegli Create (Crea).

Fase 2: Creare e configurare qualsiasi AWS IoT cosa

In questa procedura, si crea e si configura qualsiasi AWS IoT cosa. È possibile designare il computer di sviluppo come qualsiasi AWS IoT cosa. Man mano che progredisci, ricorda che i principi che stai imparando qui possono essere applicati a progetti reali. Hai la flessibilità di creare e configurare AWS IoT cose su qualsiasi dispositivo in grado di eseguire un AWS IoT SDK, inclusi AWS IoT Greengrass FreerTOS. Per ulteriori informazioni, consulta la sezione [AWS IoT SDKs](#) nella Guida per gli sviluppatori di AWS IoT .

Per creare e configurare qualsiasi cosa AWS IoT

1. Aprire una riga di comando ed eseguire il comando seguente per creare una directory per questo tutorial.

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

2. Eseguire il seguente comando per creare una directory per i certificati dell'oggetto.

```
mkdir device1
```

Se si stanno creando altri oggetti, incrementa il numero nel nome della directory di conseguenza per tenere traccia di quali certificati appartengono a quale oggetto.

3. Passare alla [console AWS IoT](#).
4. Nel riquadro di navigazione a sinistra, scegli Tutti i dispositivi nella sezione Gestisci. Quindi, scegliere Things (Oggetti).
5. Se viene visualizzata la finestra di dialogo You don't have any things yet (Non hai ancora oggetti), selezionare Create a thing (Crea un oggetto). Altrimenti, scegli Crea cose.
6. Nella pagina Creazione di elementi, scegli Crea un singolo elemento, quindi scegli Avanti.
7. Nella pagina Specificare le proprietà dell'oggetto, inserisci un nome per l' AWS IoT oggetto (ad esempio **SiteWiseTutorialDevice1**), quindi scegli Avanti. Se si stanno creando altri oggetti, incrementare il numero nel nome dell'oggetto di conseguenza.

Important

Il nome dell'oggetto deve corrispondere al nome utilizzato nella politica creata nel Passaggio 1: Creazione di una AWS IoT politica. In caso contrario, il dispositivo non potrà connettersi a AWS IoT.

8. Nella pagina Configura il certificato del dispositivo - opzionale, scegli Genera automaticamente un nuovo certificato (consigliato), quindi scegli Avanti. I certificati consentono AWS IoT di identificare in modo sicuro i dispositivi.
9. Nella pagina Allega politiche al certificato - opzionale, seleziona la politica creata nel Passaggio 1: Creazione di una AWS IoT politica e scegli Crea oggetto.
10. Nella finestra di dialogo Scarica certificati e chiavi, procedi come segue:

- a. Scegliere i collegamenti Download per scaricare il certificato, la chiave pubblica e la chiave privata dell'oggetto. Salvare tutti e tre i file nella directory creata per i certificati dell'oggetto (ad esempio, `iot-sitewise-rule-tutorial/device1`).

 Important

Questo è l'unico momento in cui è possibile scaricare il certificato e le chiavi dell'oggetto, necessari per connettere correttamente il dispositivo ad AWS IoT.

- b. Scegliete il link Download per scaricare un certificato CA root. Salvare il certificato CA radice in `iot-sitewise-rule-tutorial`. Consigliamo di scaricare Amazon Root CA 1.

11. Seleziona Fatto.

Ora hai registrato qualsiasi AWS IoT cosa sul tuo computer. Esegui uno dei seguenti passaggi successivi:

- Continua con la Fase 3: Creazione di un modello di asset del dispositivo senza creare AWS IoT elementi aggiuntivi. È possibile completare questo tutorial con un solo oggetto.
- Ripetere la procedura descritta in questa sezione su un altro computer o dispositivo per creare altri oggetti AWS IoT . Per questo tutorial, si consiglia di seguire questa opzione in modo da poter inserire dati di utilizzo univoci della CPU e della memoria da più dispositivi.
- Ripetere la procedura descritta in questa sezione sullo stesso dispositivo (computer) per creare più oggetti AWS IoT . Ogni AWS IoT oggetto riceve dati simili sull'utilizzo della CPU e della memoria dal computer, quindi utilizza questo approccio per dimostrare l'acquisizione di dati non univoci da più dispositivi.

Fase 3: Creazione di un modello di asset per dispositivi

In questa procedura, create un modello di asset AWS IoT SiteWise per rappresentare i dispositivi che trasmettono i dati sull'utilizzo della CPU e della memoria. Per elaborare i dati in asset che rappresentano gruppi di dispositivi, i modelli di asset applicano informazioni coerenti su più asset dello stesso tipo. Per ulteriori informazioni, consulta [Asset industriali modello](#).

Per creare un modello di asset che rappresenta un dispositivo

1. Passare alla [console AWS IoT SiteWise](#).

2. Nel riquadro di navigazione a sinistra scegliere Models (Modelli).
3. Scegli Crea modello.
4. In Dettagli del modello, inserisci un nome per il tuo modello. Ad esempio, **SiteWise Tutorial Device Model**.
5. In Measurement definitions (Definizioni misurazione), effettuare le seguenti operazioni:
 - a. In Nome, inserisci **CPU Usage**.
 - b. In Unit (Unità), immettere %.
 - c. Lasciare Data type (Tipo di dati) come Double (Doppio).

Le proprietà di misurazione rappresentano i flussi di dati non elaborati di un dispositivo.

Per ulteriori informazioni, consulta [Definisci i flussi di dati provenienti dalle apparecchiature \(misurazioni\)](#).

6. Scegli Aggiungi nuova misurazione per aggiungere una seconda proprietà di misurazione.
7. Nella seconda riga sotto Measurement definitions (Definizioni misurazione), effettuare le seguenti operazioni:
 - a. In Nome, inserisci **Memory Usage**.
 - b. In Unit (Unità), immettere %.
 - c. Lasciare Data type (Tipo di dati) come Double (Doppio).
8. In Metric definitions (Definizioni parametri), effettuare le seguenti operazioni:
 - a. In Nome, inserisci **Average CPU Usage**.
 - b. In Formula (Formula), immettere **avg(CPU Usage)** . Scegliere CPU Usage dall'elenco di completamento automatico quando viene visualizzato.
 - c. In Time interval (Intervallo di tempo), immettere **5 minutes**.

Le proprietà dei parametri definiscono i calcoli di aggregazione che elaborano tutti i punti dati di input per un intervallo e generano un singolo punto dati per ogni intervallo. Questa proprietà dei parametri calcola l'utilizzo medio della CPU di ogni dispositivo ogni 5 minuti. Per ulteriori informazioni, consulta [Aggrega i dati provenienti da proprietà e altre risorse \(metriche\)](#).

9. Scegli Aggiungi nuova metrica per aggiungere una seconda proprietà metrica.
10. Nella seconda riga sotto Metric definitions (Definizioni parametri), effettuare le seguenti operazioni:

- a. In Nome, inserisci **Average Memory Usage**.
- b. In Formula (Formula), immettere **avg(Memory Usage)** . Scegliere Memory Usage dall'elenco di completamento automatico quando viene visualizzato.
- c. In Time interval (Intervallo di tempo), immettere **5 minutes**.

Questa proprietà dei parametri calcola l'utilizzo medio della memoria di ogni dispositivo ogni 5 minuti.

11. (Facoltativo) Aggiungere altri parametri che devono essere calcolati per ogni dispositivo. Alcune funzioni interessanti sono min e max. Per ulteriori informazioni, consulta [Usa espressioni di formule](#). Nella Fase 4: Creazione di un modello di asset per il parco dispositivi, crei una risorsa principale in grado di calcolare le metriche utilizzando i dati dell'intero parco dispositivi.
12. Scegli Crea modello.

Fase 4: Creazione di un modello di asset per il parco dispositivi

In questa procedura, create un modello di asset che simboleggi la vostra collezione di dispositivi. AWS IoT SiteWise All'interno di questo modello di asset, stabilisci una struttura che ti consente di collegare numerosi dispositivi a un unico asset globale della flotta. Successivamente, definisci le metriche nel modello di asset della flotta per consolidare i dati di tutti gli asset dei dispositivi connessi. Questo approccio fornisce informazioni complete sulle prestazioni collettive dell'intero parco veicoli.

Per creare un modello di asset che rappresenta un parco istanze dei dispositivi

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione a sinistra scegliere Models (Modelli).
3. Scegli Crea modello.
4. In Dettagli del modello, inserisci un nome per il tuo modello. Ad esempio, **SiteWise Tutorial Device Fleet Model**.
5. In Hierarchy definitions (Definizioni gerarchie), effettuare le seguenti operazioni:
 - a. In Hierarchy name (Nome gerarchia), immettere **Device**.
 - b. Nel Hierarchy model (Modello gerarchia), scegliere il modello di asset dispositivo (**SiteWise Tutorial Device Model**).

Una gerarchia definisce una relazione tra un modello di asset padre (parco istanze) e un modello di asset figlio (dispositivo). Gli asset padre possono accedere ai dati delle proprietà degli asset figlio. Quando si creano gli asset in un secondo momento, è necessario associare gli asset figlio agli asset padre in base a una definizione di gerarchia nel modello di asset padre. Per ulteriori informazioni, consulta [Definite le gerarchie dei modelli di asset](#).

6. In Metric definitions (Definizioni parametri), effettuare le seguenti operazioni:
 - a. In Nome, inserisci **Average CPU Usage**.
 - b. In Formula (Formula), immettere **avg(Device | Average CPU Usage)**. Quando viene visualizzato l'elenco di completamento automatico, selezionare Device per scegliere una gerarchia, quindi selezionare Average CPU Usage per scegliere il parametro dall'asset dispositivo creato in precedenza.
 - c. In Time interval (Intervallo di tempo), immettere **5 minutes**.

Questa proprietà del parametro calcola l'utilizzo medio della CPU di tutti gli asset dispositivo associati a un asset parco istanze tramite la gerarchia **Device**.

7. Scegli Aggiungi nuova metrica per aggiungere una seconda proprietà metrica.
8. Nella seconda riga sotto Metric definitions (Definizioni parametri), effettuare le seguenti operazioni:
 - a. In Nome, inserisci **Average Memory Usage**.
 - b. In Formula (Formula), immettere **avg(Device | Average Memory Usage)**. Quando viene visualizzato l'elenco di completamento automatico, selezionare Device per scegliere una gerarchia, quindi selezionare Average Memory Usage per scegliere il parametro dall'asset dispositivo creato in precedenza.
 - c. In Time interval (Intervallo di tempo), immettere **5 minutes**.

Questa proprietà del parametro calcola l'utilizzo medio della memoria di tutti gli asset dispositivo associati a un asset parco istanze tramite la gerarchia **Device**.

9. (Facoltativo) Aggiungere altri parametri che devono essere calcolati per il parco istanze dei dispositivi.
10. Scegli Crea modello.

Passaggio 5: Creare e configurare una risorsa del dispositivo

In questa procedura, si genera una risorsa del dispositivo basata sul modello di risorsa del dispositivo. Quindi, definisci gli alias per ogni proprietà di misurazione. Un alias di proprietà è una stringa univoca che identifica la proprietà di una risorsa. Successivamente, potete identificare una proprietà per il caricamento dei dati utilizzando gli alias anziché l'ID della risorsa e l'ID della proprietà. Per ulteriori informazioni, consulta [Gestisci i flussi di dati per AWS IoT SiteWise](#).

Per creare un asset dispositivo e definire gli alias di proprietà

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione a sinistra, scegli Asset.
3. Selezionare Create asset (Crea asset).
4. In Informazioni sul modello, scegliete il modello di asset del dispositivo, **SiteWise Tutorial Device Model**.
5. In Informazioni sulla risorsa, inserite un nome per la risorsa. Ad esempio, **SiteWise Tutorial Device 1**.
6. Selezionare Create asset (Crea asset).
7. Per il nuovo asset dispositivo, scegliere Edit (Modifica).
8. In CPU Usage, immettere **/tutorial/device/SiteWiseTutorialDevice1/cpu** come alias di proprietà. Includi il nome dell' AWS IoT oggetto nell'alias della proprietà, in modo da poter importare i dati da tutti i tuoi dispositivi utilizzando un'unica AWS IoT regola.
9. In Memory Usage, immettere **/tutorial/device/SiteWiseTutorialDevice1/memory** come alias di proprietà.
10. Scegli Save (Salva).

Se hai creato più AWS IoT elementi in precedenza, ripeti i passaggi da 3 a 10 per ogni dispositivo e incrementa di conseguenza il numero nel nome della risorsa e negli alias delle proprietà. Ad esempio, il nome del secondo asset dispositivo dovrebbe essere **SiteWise Tutorial Device 2** e i relativi alias di proprietà dovrebbero essere **/tutorial/device/SiteWiseTutorialDevice2/cpu** e **/tutorial/device/SiteWiseTutorialDevice2/memory**.

Fase 6: Creare e configurare un parco di dispositivi

In questa procedura, si crea un asset di parco dispositivi derivato dal modello di parco dispositivi utilizzato. Quindi, colleghi le risorse dei singoli dispositivi alla risorsa del parco dispositivi. Questa

associazione consente alle proprietà metriche dell'asset della flotta di compilare e analizzare i dati provenienti da più dispositivi. Questi dati forniscono una visione consolidata delle prestazioni collettive dell'intera flotta.

Per creare un asset parco istanze dei dispositivi e associare asset dispositivo

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione a sinistra, scegli Asset.
3. Selezionare Create asset (Crea asset).
4. In Informazioni sul modello, scegli il modello di asset della flotta di dispositivi, **SiteWise Tutorial Device Fleet Model**.
5. In Informazioni sulla risorsa, inserisci un nome per la risorsa. Ad esempio, **SiteWise Tutorial Device Fleet 1**.
6. Selezionare Create asset (Crea asset).
7. Per il nuovo asset parco istanze dei dispositivi, scegliere Edit (Modifica).
8. In Risorse associate a questa risorsa, scegliete Aggiungi risorsa associata ed effettuate le seguenti operazioni:
 - a. In Hierarchy (Gerarchia), scegliere Device. Questa gerarchia identifica la relazione di gerarchia tra gli asset parco istanze dei dispositivi e gli asset dispositivi. Questa gerarchia è stata definita nel modello di asset parco istanze dei dispositivi in precedenza in questo tutorial.
 - b. In Asset, scegliere l'asset dispositivo SiteWise Tutorial Device 1.
9. (Facoltativo) Se in precedenza avete creato più risorse per dispositivo, ripetete i passaggi da 8 a 10 per ogni risorsa del dispositivo che avete creato.
10. Scegli Save (Salva).

A questo punto gli asset dispositivo vengono visualizzati organizzati in gerarchia.

Passaggio 7: crea una regola in AWS IoT Core per inviare dati alle risorse del dispositivo

In questa procedura, stabilisci una regola in AWS IoT Core. La regola è progettata per interpretare i messaggi di notifica provenienti dalle ombre del dispositivo e trasmettere i dati alle risorse del dispositivo in AWS IoT SiteWise. Ogni volta che il dispositivo si aggiorna lo shadow, AWS IoT invia

un messaggio MQTT. Puoi creare una regola che intraprende un'azione quando le shadow del dispositivo cambiano in base al messaggio MQTT. In questo caso, l'obiettivo è gestire il messaggio di aggiornamento, estrarre i valori delle proprietà e trasmetterli alle risorse del dispositivo. AWS IoT SiteWise

Creare una regola con un' AWS IoT SiteWise azione

1. Passare alla [console AWS IoT](#).
2. Nel riquadro di navigazione a sinistra, scegli Routing dei messaggi, quindi scegli Regole.
3. Scegli Crea regola.
4. Inserisci un nome e una descrizione per la regola, quindi scegli Avanti.
5. Immettete la seguente istruzione SQL e scegliete Avanti.

```
SELECT
  *
FROM
  '$aws/things/+/shadow/update/accepted'
WHERE
  startsWith(topic(3), "SiteWiseTutorialDevice")
```

Questa istruzione di query di regola funziona perché il servizio shadow del dispositivo pubblica gli aggiornamenti shadow in `$aws/things/thingName/shadow/update/accepted`. Per ulteriori informazioni sulle ombre dei dispositivi, consulta [Device shadow service](#) nella AWS IoT Device Shadow Guide.

Nella clausola WHERE, questa istruzione di query di regola utilizza la funzione `topic(3)` per ottenere il nome dell'oggetto dal terzo segmento dell'argomento. Quindi, l'istruzione esclude i dispositivi con nomi che non corrispondono a quelli dei dispositivi del tutorial. Per ulteriori informazioni su AWS IoT SQL, consulta il [riferimento AWS IoT SQL](#) nella AWS IoT Developer Guide.

6. In Azioni relative alle regole, scegliete Invia i dati dei messaggi alle proprietà degli asset in AWS IoT SiteWise ed effettuate le seguenti operazioni:
 - a. Scegliere By property alias (Per alias di proprietà).
 - b. In Property alias (Alias proprietà), immettere **`/tutorial/device/${topic(3)}/cpu`**.

La `${...}` sintassi è un modello sostitutivo. AWS IoT valuta il contenuto all'interno delle parentesi. Questo modello di sostituzione esegue il pull del nome dell'oggetto dall'argomento

per creare un alias univoco per ogni oggetto. Per ulteriori informazioni, consulta [Modelli sostitutivi](#) nella Guida per gli sviluppatori AWS IoT

 Note

Poiché un'espressione in un modello di sostituzione viene valutata separatamente dall'istruzione SELECT, non puoi utilizzare un modello di sostituzione per fare riferimento a un alias creato utilizzando una clausola AS. È possibile fare riferimento solo alle informazioni presenti nel payload originale, oltre alle funzioni e agli operatori supportati.

- c. In Entry ID, facoltativo, inserisci. **`${concat(topic(3), "-cpu-", floor(state.reported.timestamp))}`**

Entry identifica IDs in modo univoco ogni tentativo di inserimento del valore. Se una voce restituisce un errore, è possibile trovare l'ID voce nell'output dell'errore per risolvere il problema. Il modello di sostituzione in questo ID voce combina il nome dell'oggetto e il timestamp restituito del dispositivo. Ad esempio, l'ID voce risultante potrebbe essere simile a SiteWiseTutorialDevice1-cpu-1579808494.

- d. In Time in seconds (Ora in secondi), immettere **`${floor(state.reported.timestamp)}`**.

Questo modello di sostituzione calcola l'ora in secondi dal timestamp restituito del dispositivo. In questo tutorial, i dispositivi restituiscono il timestamp in secondi nel formato di ora epoch Unix come numero a virgola mobile.

- e. In Offset in nanos - opzionale, inserisci. **`${floor((state.reported.timestamp % 1) * 1E9)}`**

Questo modello di sostituzione calcola l'offset in nanosecondi dall'ora in secondi convertendo la parte decimale del timestamp restituito del dispositivo.

 Note

AWS IoT SiteWise richiede che i dati abbiano un timestamp corrente in Unix Epoch Time. Se i dispositivi non restituiscono l'ora in modo accurato, è possibile che l'ora corrente restituita dal motore di regole AWS IoT sia [timestamp\(\)](#). Questa funzione segnala l'ora in millisecondi, quindi è necessario aggiornare i parametri dell'ora dell'azione di regola ai seguenti valori:

- In Time in seconds (Ora in secondi), immettere **`${floor(timestamp() / 1E3)}`**.
- In Offset in nanos (Offset in nanosecondi), immettere **`${(timestamp() % 1E3) * 1E6}`**.

f. In Data type (Tipo di dati), scegliere Double (Doppio).

Questo tipo di dati deve corrispondere al tipo di dati della proprietà dell'asset definita nel modello di asset.

- g. In Valore, immetti **`${state.reported.cpu}`**. Nei modelli di sostituzione, è possibile utilizzare l'operatore `.` per recuperare un valore all'interno di una struttura JSON.
- h. Scegliere Add entry (Aggiungi voce) per aggiungere una nuova voce per la proprietà di utilizzo della memoria e completare nuovamente i passaggi seguenti per questa proprietà:
- Scegliere By property alias (Per alias di proprietà).
 - In Property alias (Alias proprietà), immettere **`/tutorial/device/${topic(3)}/memory`**.
 - In Entry ID - facoltativo, inserisci **`${concat(topic(3), "-memory-", floor(state.reported.timestamp))}`**
 - In Time in seconds (Ora in secondi), immettere **`${floor(state.reported.timestamp)}`**.
 - In Offset in nanos - opzionale, inserisci **`${floor((state.reported.timestamp % 1) * 1E9)}`**
 - In Data type (Tipo di dati), scegliere Double (Doppio).
 - In Valore, immetti **`${state.reported.memory}`**.
 - In Ruolo IAM, scegli Crea nuovo ruolo per creare un ruolo IAM per questa azione della regola. Questo ruolo consente di AWS IoT inviare dati alle proprietà della flotta di dispositivi e della relativa gerarchia degli asset.
 - Inserisci il nome del ruolo e scegli Crea.
7. (Facoltativo) Configurare un'azione di errore che è possibile utilizzare per risolvere i problemi della regola. Per ulteriori informazioni, consulta [Risolvere i problemi relativi a una regola \(\)AWS IoT SiteWise](#).
8. Scegli Next (Successivo).
9. Controlla le impostazioni e scegli Crea per creare la regola.

Passaggio 8: Esegui lo script del client del dispositivo

In questo tutorial, non stai utilizzando un dispositivo reale per riportare i dati. Invece, esegui uno script per aggiornare l'ombra del dispositivo in uso con l'utilizzo della CPU e della memoria per imitare i dati reali dei sensori. AWS IoT Per eseguire lo script, è necessario prima installare Python i pacchetti richiesti. In questa procedura, si Python installano i pacchetti richiesti e quindi si esegue lo script client del dispositivo.

Per configurare ed eseguire lo script client del dispositivo

1. Passare alla [console AWS IoT](#).
2. Nella parte inferiore del riquadro di spostamento a sinistra scegliere Settings (Impostazioni).
3. Salvare l'endpoint personalizzato da utilizzare con lo script client del dispositivo. Usare questo endpoint per interagire con le shadow dell'oggetto. Questo endpoint è univoco per l'account nella regione corrente.

L'endpoint personalizzato è simile all'esempio seguente.

```
identifier.iot.region.amazonaws.com
```

4. Aprire una riga di comando ed eseguire il comando seguente per passare alla directory creata in precedenza nel tutorial.

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

5. Eseguire il comando seguente per installare SDK per dispositivi AWS IoT per Python.

```
pip3 install AWSIoTPythonSDK
```

Per ulteriori informazioni, consulta [SDK per dispositivi AWS IoT per Python](#) la Guida per AWS IoT gli sviluppatori

6. Eseguire il seguente comando per installare psutil, un processo multiplatforma e una libreria di utilità di sistema.

```
pip3 install psutil
```

Per ulteriori informazioni, consultare [psutil](#) nell'indice dei pacchetti Python.

7. Creare un file chiamato `thing_performance.py` nella directory `iot-sitewise-rule-tutorial` e quindi copiare il seguente codice Python nel file.

```
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 {}".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. Esegui `thing_performance.py` dalla riga di comando con i parametri seguenti:

- `-n, --thingName` — Nome del tuo oggetto, ad esempio **SiteWiseTutorialDevice1**.
- `-e, --endpoint` — L' AWS IoT endpoint personalizzato salvato in precedenza in questa procedura.
- `-r, --rootCA` — Il percorso del certificato CA AWS IoT principale.
- `-c, --cert` — Il percorso del tuo certificato AWS IoT Thing.
- `-k, --key` — Il percorso della chiave privata del tuo AWS IoT Thing Certificate.
- `-d, --requestDelay` — (Facoltativo) Il tempo di attesa, in secondi, tra un aggiornamento dello shadow del dispositivo e l'altro. Il valore predefinito è 1 secondo.
- `-v, --enableLogging` — (Facoltativo) Se questo parametro è presente, lo script stampa i messaggi di debug da. SDK per dispositivi AWS IoT per Python

Il comando è simile al seguente esempio:

```

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

```

Se state eseguendo lo script per altre AWS IoT operazioni, aggiornate di conseguenza il nome dell'oggetto e la directory del certificato.

9. Provare ad aprire e chiudere i programmi sul dispositivo per vedere come cambiano gli utilizzi della CPU e della memoria. Lo script stampa ogni lettura dell'utilizzo della CPU e della memoria. Se lo script carica i dati nel servizio shadow del dispositivo correttamente, l'output dello script è simile all'esempio seguente.

```

[SiteWiseTutorialDevice1]
CPU:    24.6%

```

Memory: 85.2%

[SiteWiseTutorialDevice1]

Update request e6686e44-fca0-44db-aa48-3ca81726f3e3 accepted

10. Attenersi alla seguente procedura per verificare che lo script aggiorni la shadow del dispositivo:

- a. Passare alla [console AWS IoT](#).
- b. Nel riquadro di navigazione a sinistra, scegli Tutti i dispositivi, quindi scegli Cose.
- c. Scegli quello che preferisci, SiteWiseTutorialDevice.
- d. Scegliete la scheda Device Shadows, scegliete Classic Shadow e verificate che lo stato Shadow sia simile al seguente esempio.

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

Se lo stato shadow della cosa è vuoto o non assomiglia all'esempio precedente, controlla che lo script sia in esecuzione e che la connessione sia avvenuta correttamente. AWS IoT

Se lo script continua a scadere durante la connessione a AWS IoT, verifica che la [policy relativa alle cose](#) sia configurata in base a questo tutorial.

11. Attenersi alla seguente procedura per verificare che l'azione di regola invii dati ad AWS IoT SiteWise:

- a. Passare alla [console AWS IoT SiteWise](#).
- b. Nel riquadro di navigazione a sinistra, scegli Asset.
- c. Scegliere la freccia accanto all'asset parco istanze dei dispositivi (SiteWise Tutorial Device Fleet 1 1) per espandere la gerarchia degli asset, quindi scegliere l'asset dispositivo (SiteWise Tutorial Device 1).
- d. Scegliere Measurements (Misurazioni).
- e. Verificare che le celle Latest value (Valore più recente) abbiano i valori per le proprietà CPU Usage e Memory Usage.

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. Se le proprietà CPU Usage e Memory Usage non hanno i valori più recenti, aggiornare la pagina. Se i valori non vengono visualizzati dopo pochi minuti, consultare [Risolvere i problemi relativi a una regola \(\)AWS IoT SiteWise](#).

Il tutorial è stato completato. Per esplorare le visualizzazioni dei dati in tempo reale, è possibile configurare un portale in AWS IoT SiteWise Monitor. Per ulteriori informazioni, consulta [Monitora i dati con AWS IoT SiteWise Monitor](#). In caso contrario, è possibile premere CTRL+C nel prompt dei comandi per arrestare lo script client del dispositivo. È improbabile che il programma Python invii un numero di messaggi tale da comportare un addebito, ma è consigliabile arrestare il programma al termine delle operazioni.

Passaggio 9: Pulisci le risorse dopo il tutorial

Dopo aver completato il tutorial sull'acquisizione di dati dagli AWS IoT oggetti, ripulisci le risorse per evitare di incorrere in costi aggiuntivi.

Per eliminare risorse gerarchiche in AWS IoT SiteWise

1. [Vai alla console AWS IoT SiteWise](#)
2. Nel riquadro di navigazione a sinistra, scegli Asset.
3. Quando eliminate delle risorse in AWS IoT SiteWise, dovete prima dissociarle.

Completare la procedura seguente per dissociare gli asset del dispositivo dall'asset parco istanze dei dispositivi:

- a. Scegliete la risorsa del parco dispositivi (SiteWise Tutorial Device Fleet 1).
- b. Scegli Modifica.
- c. In Assets associated to this asset (Asset associati a questo asset), scegliere Disassociate (Dissocia) per ogni asset dispositivo associato a questo asset parco istanze dei dispositivi.
- d. Scegli Save (Salva).

A questo punto gli asset dispositivo non sono più organizzati in gerarchia.

4. Scegliere l'asset dispositivo (SiteWise Tutorial Device 1).
5. Scegliere Delete (Elimina).
6. Nella finestra di dialogo di conferma immettere **Delete** e quindi scegliere Delete (Elimina).
7. Ripeti i passaggi da 4 a 6 per ogni risorsa del dispositivo e per la risorsa del parco dispositivi (SiteWise Tutorial Device Fleet 1).

Per eliminare i modelli di asset gerarchici in AWS IoT SiteWise

1. Passare alla [console AWS IoT SiteWise](#).
2. Se non è già stato fatto, eliminare gli asset del dispositivo e del parco istanze dei dispositivi. Per ulteriori informazioni, consultare [la procedura precedente](#). Non è possibile eliminare un modello se sono presenti asset creati da quel modello.
3. Nel riquadro di navigazione a sinistra scegliere Models (Modelli).
4. Scegliere l'asset parco istanze dei dispositivi (SiteWise Tutorial Device Fleet Model).

Quando eliminate i modelli di asset gerarchici, iniziate eliminando prima il modello di asset principale.

5. Scegliere Delete (Elimina).
6. Nella finestra di dialogo di conferma immettere **Delete** e quindi scegliere Delete (Elimina).
7. Ripetere i passaggi da 4 a 6 per il modello di asset del dispositivo (SiteWise Tutorial Device Model).

Per disabilitare o eliminare una regola in AWS IoT Core

1. Passare alla [console AWS IoT](#).
2. Nel riquadro di navigazione a sinistra, scegli Routing dei messaggi, quindi scegli Regole.
3. Seleziona la tua regola e scegli Elimina.
4. Nella finestra di dialogo di conferma, inserisci il nome della regola, quindi scegli Elimina.

Visualizza e condividi i dati dei parchi eolici in Monitor SiteWise

Questo tutorial spiega come visualizzare e condividere dati industriali tramite applicazioni web gestite, note come portali. AWS IoT SiteWise Monitor Ogni portale comprende progetti e offre la flessibilità di scegliere quali dati sono accessibili all'interno di ciascun progetto. Quindi, specifica le persone

della tua organizzazione che possono accedere a ciascun portale. I tuoi utenti accedono ai portali utilizzando AWS IAM Identity Center gli account, in modo che tu possa utilizzare il tuo archivio di identità esistente o uno store gestito da AWS.

Tu e gli utenti con le autorizzazioni sufficienti potete creare i pannelli di controllo in ogni progetto per visualizzare i dati di settore in modo significativo. Quindi, gli utenti possono visualizzare questi pannelli di controllo per ottenere rapidamente informazioni dettagliate sui dati e monitorare le operazioni. Puoi configurare autorizzazioni amministrative o di sola lettura per ogni progetto e per ogni utente della società. Per ulteriori informazioni, consulta [Monitora i dati con AWS IoT SiteWise Monitor](#).

Nel corso del tutorial, migliorerai la AWS IoT SiteWise demo, fornendo un set di dati di esempio per un parco eolico. Si configura un portale in SiteWise Monitor, si crea un progetto e si creano dashboard per visualizzare i dati del parco eolico. Il tutorial copre anche la creazione di utenti aggiuntivi, insieme all'assegnazione delle autorizzazioni per possedere o visualizzare il progetto e le dashboard associate.

Note

Quando usi SiteWise Monitor, ti viene addebitato un costo per utente che accede a un portale (al mese). In questo tutorial vengono creati tre utenti, ma dovrà accedere solo un utente. Dopo aver completato questo tutorial, ti vengono addebitati i costi per un utente. Per ulteriori informazioni, consulta [AWS IoT SiteWise Prezzi](#).

Argomenti

- [Prerequisiti](#)
- [Passaggio 1: crea un portale in Monitor SiteWise](#)
- [Passaggio 2: accedi a un portale](#)
- [Fase 3: Creare un progetto per un parco eolico](#)
- [Fase 4: Creare una dashboard per visualizzare i dati del parco eolico](#)
- [Fase 5: Esplora il portale](#)
- [Passaggio 6: Pulisci le risorse dopo il tutorial](#)

Prerequisiti

Per completare questo tutorial, è necessario quanto segue:

- Un Account AWS. Se non lo hai, consultare [Configura un account AWS](#).
- Un computer di sviluppo in esecuzione Windows, macOS, Linux, oppure Unix per accedere a AWS Management Console. Per ulteriori informazioni, consulta la pagina [Che cos'è AWS Management Console?](#).
- Un utente AWS Identity and Access Management (IAM) con autorizzazioni di amministratore.
- Una demo di un parco AWS IoT SiteWise eolico funzionante. Quando configuri la demo, definisce i modelli e gli asset AWS IoT SiteWise e trasmette loro i dati per rappresentare un parco eolico. Per ulteriori informazioni, consulta [Usa la AWS IoT SiteWise demo](#).
- Se hai abilitato IAM Identity Center nel tuo account, accedi al tuo account di AWS Organizations gestione. Per ulteriori informazioni, consulta [Concetti e terminologia di AWS Organizations](#). Se non hai abilitato IAM Identity Center, lo abiliterai in questo tutorial e imposterai il tuo account come account di gestione.

Se non riesci ad accedere al tuo account di AWS Organizations gestione, puoi completare parzialmente il tutorial purché nella tua organizzazione sia presente un utente IAM Identity Center. In questo caso, puoi creare il portale e le dashboard, ma non puoi creare nuovi utenti IAM Identity Center da assegnare ai progetti.

Passaggio 1: crea un portale in Monitor SiteWise

In questa procedura crei un portale in AWS IoT SiteWise Monitor. Ogni portale è un'applicazione web gestita a cui tu e i tuoi utenti potete accedere con AWS IAM Identity Center gli account. Con IAM Identity Center, puoi utilizzare l'archivio di identità esistente della tua azienda o crearne uno gestito da AWS. I dipendenti della tua azienda possono accedere senza creare file separati Account AWS.

Per creare un portale

1. Accedere alla [console AWS IoT SiteWise](#).
2. Controlla gli [AWS IoT SiteWise endpoint e le quote](#) dove AWS IoT SiteWise è supportato e cambia regione, se necessario. È necessario eseguire la AWS IoT SiteWise demo nella stessa regione.
3. Nel riquadro di navigazione a sinistra scegliere Portals (Portali).

4. Selezionare Create portal (Crea portale).
5. Se hai già abilitato IAM Identity Center, vai al passaggio 6. Altrimenti, completa i seguenti passaggi per abilitare IAM Identity Center:
 - a. Nella pagina Abilita AWS IAM Identity Center (SSO), inserisci il tuo indirizzo e-mail, nome e cognome per creare un utente IAM Identity Center che diventi amministratore del portale. Utilizza un indirizzo e-mail a cui puoi accedere in modo da ricevere un'e-mail per impostare una password per il tuo nuovo utente IAM Identity Center.

In un portale, l'amministratore del portale crea i progetti e assegna gli utenti ai progetti. Sarà possibile configurare altri utenti in un secondo momento.

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

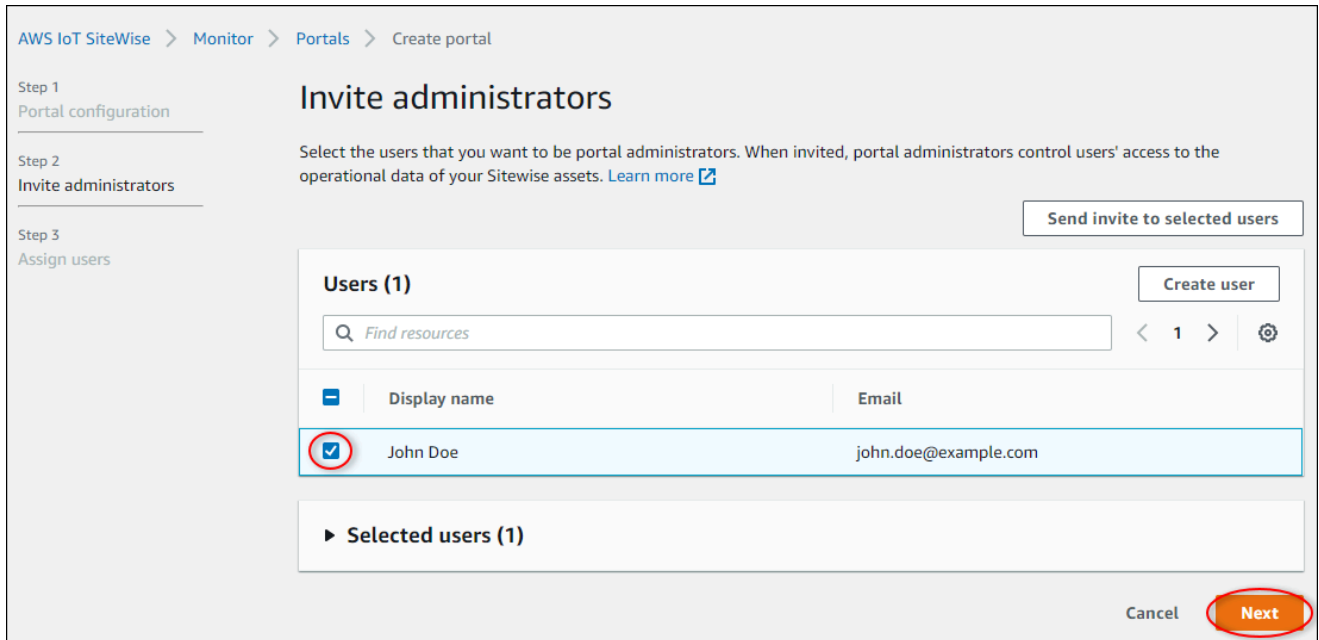
Last name
Doe

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

Cancel **Create user**

- b. Selezionare Create user (Crea utente).
6. Nella pagina Portal configuration (Configurazione portale) completare i passaggi seguenti:
 - a. Immettere un nome per il portale, ad esempio **WindFarmPortal**.
 - b. (Facoltativo) Inserire una descrizione del portale. Se si dispone di più portali, utilizzare descrizioni significative per tenere traccia dei contenuti di ciascuno.
 - c. (Facoltativo) Carica un'immagine da visualizzare nel portale.
 - d. Immettete un indirizzo e-mail che gli utenti del portale possano contattare in caso di problemi con il portale e necessitano dell'aiuto dell' AWS amministratore dell'azienda per risolverlo.
 - e. Selezionare Create portal (Crea portale).

7. Nella pagina Invita amministratori, puoi assegnare gli utenti di IAM Identity Center al portale come amministratori. Gli amministratori del portale gestiscono le autorizzazioni e i progetti all'interno di un portale. In questa pagina, effettuate le seguenti operazioni:
 - a. Seleziona un utente come amministratore del portale. Se hai abilitato IAM Identity Center in precedenza in questo tutorial, seleziona l'utente che hai creato.



- b. Facoltativamente, è possibile avvalersi del comando Send invite to selected users (Manda un invito agli utenti selezionati). Il tuo client di posta elettronica si apre e nel corpo del messaggio viene visualizzato un invito. È possibile personalizzare il messaggio e-mail prima di inviarlo agli amministratori del portale. È inoltre possibile inviare l'e-mail agli amministratori del portale in un secondo momento. Se stai provando SiteWise Monitor per la prima volta e sarai l'amministratore del portale, non devi inviarti un'e-mail.
 - c. Scegli Next (Successivo).
8. Nella pagina Assegna utenti, puoi assegnare gli utenti di IAM Identity Center al portale. Gli amministratori del portale possono successivamente assegnare questi utenti come proprietari o visualizzatori del progetto. I proprietari dei progetti possono creare dashboard nei progetti. I visualizzatori dei progetti hanno accesso in sola lettura ai progetti a cui sono assegnati. In questa pagina, puoi creare utenti IAM Identity Center da aggiungere al portale.

Note

Se non hai effettuato l'accesso al tuo account di AWS Organizations gestione, non puoi creare utenti IAM Identity Center. Scegli Assegna utenti per creare il portale senza utenti del portale, quindi salta questo passaggio.

In questa pagina, effettuate le seguenti operazioni:

- a. Completa due volte i seguenti passaggi per creare due utenti IAM Identity Center:
 - i. Scegli Crea utente per aprire una finestra di dialogo in cui inserire i dettagli per il nuovo utente.
 - ii. Inserisci un indirizzo e-mail, nome e cognome per il nuovo utente. IAM Identity Center invia all'utente un'e-mail per consentirgli di impostare la password. Se desideri accedere al portale come questi utenti, scegli un indirizzo email a cui puoi accedere. Ogni indirizzo e-mail deve essere unico. Gli utenti accedono al portale utilizzando il loro indirizzo e-mail come nome utente.

Create user [X]

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

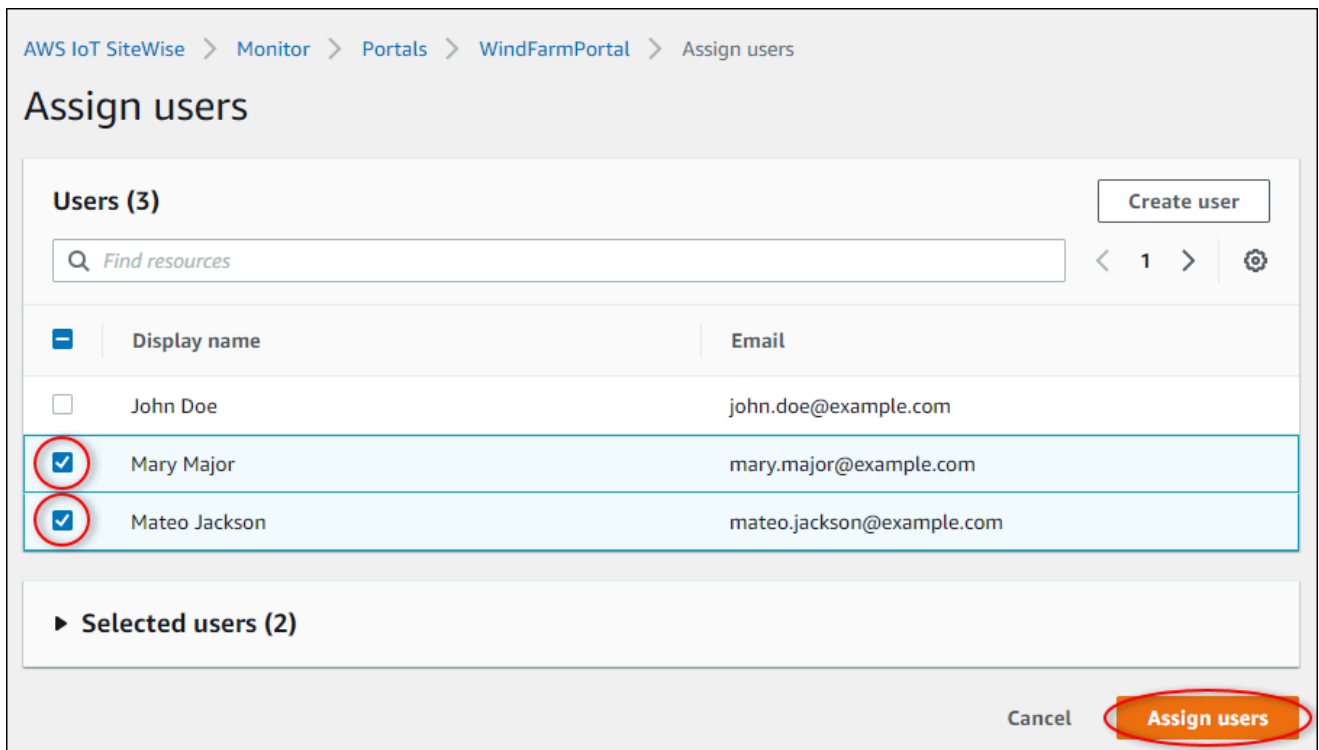
Email address
mary.major@example.com

First name
Mary

Last name
Major

Cancel Create user

- iii. Selezionare Create user (Crea utente).
- b. Seleziona i due utenti IAM Identity Center che hai creato nel passaggio precedente.



- c. Scegli Assegna utenti per aggiungere questi utenti al portale.

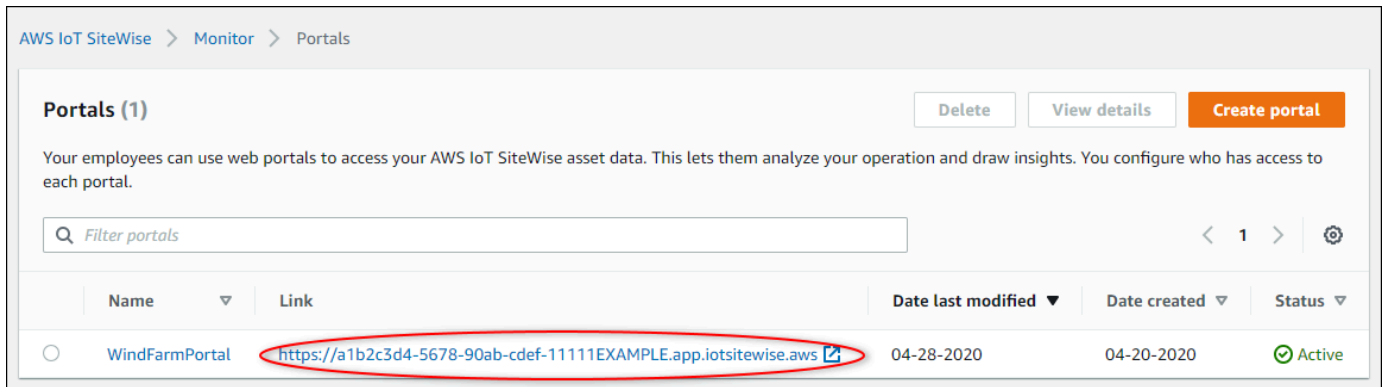
La pagina dei portali si apre con il nuovo portale elencato.

Passaggio 2: accedi a un portale

In questa procedura, accedi al nuovo portale utilizzando l' AWS IAM Identity Center utente che hai aggiunto al portale.

Per accedere a un portale

1. Nella pagina Portals (Portali) scegliere il collegamento del nuovo portale per aprire il portale in una nuova scheda.



2. Se hai creato il tuo primo utente IAM Identity Center in precedenza nel tutorial, utilizza i seguenti passaggi per creare una password per il tuo utente:

- Controlla la tua email per vedere l'oggetto Invitation to join AWS IAM Identity Center.
- Apri l'e-mail di invito e scegli Accept invitation.
- Nella nuova finestra, imposta una password per il tuo utente IAM Identity Center.

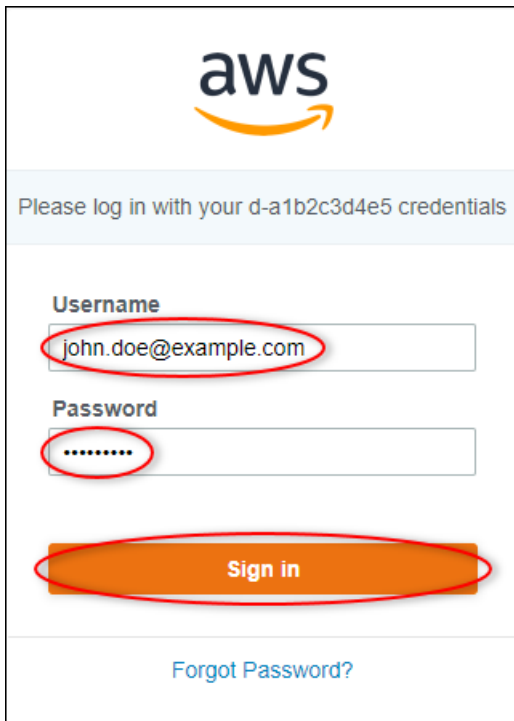
Se desideri accedere successivamente al portale come secondo e terzo utente IAM Identity Center che hai creato in precedenza, puoi anche completare questi passaggi per impostare le password per quegli utenti.

Note

Se non hai ricevuto un'e-mail, puoi generare una password per il tuo utente nella console IAM Identity Center. Per ulteriori informazioni, consulta [Reimpostazione della password utente di IAM Identity Center per un utente finale](#) nella Guida per l'AWS IAM Identity Center utente.

3. Accedi al tuo IAM Identity Center Username e Password. Se hai creato il tuo utente IAM Identity Center in precedenza in questo tutorial, Username è l'indirizzo e-mail dell'utente amministratore del portale che hai creato.

Tutti gli utenti del portale, incluso l'amministratore del portale, devono accedere con le proprie credenziali utente IAM Identity Center. Tali credenziali, in genere, non corrispondono a quelle utilizzate per accedere alla AWS Management Console.



4. Scegliere Sign in.

Viene visualizzato il portale.

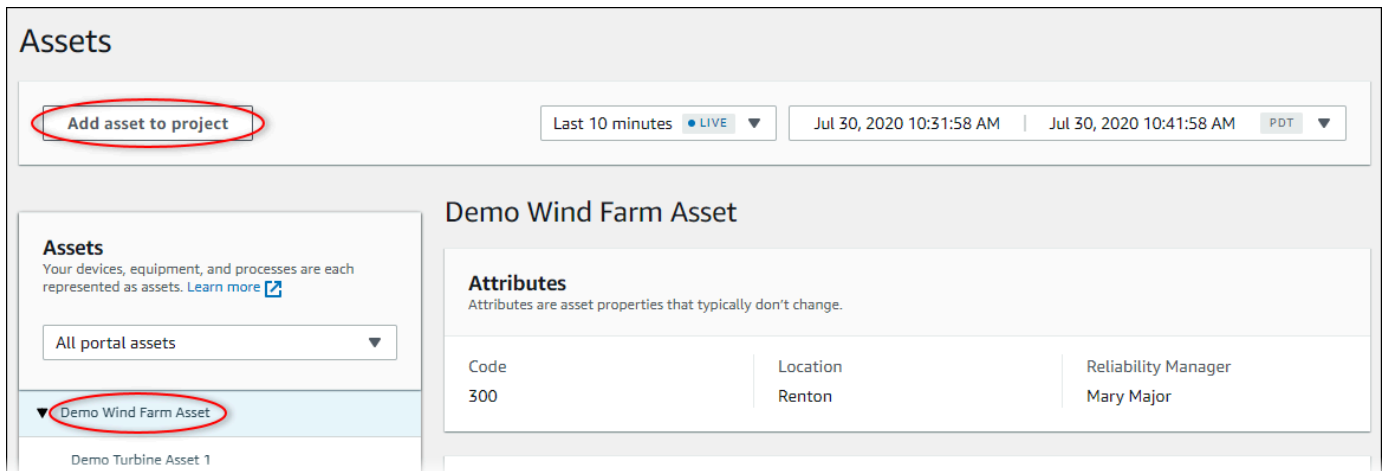
Fase 3: Creare un progetto per un parco eolico

In questa procedura crei un progetto nel portale. I progetti sono risorse che definiscono una serie di autorizzazioni, risorse e dashboard, che puoi configurare per visualizzare i dati delle risorse in quel progetto. Con i progetti, è possibile definire chi ha accesso a sottoinsiemi specifici dell'operazione e come vengono visualizzati i dati di tali sottoinsiemi. È possibile assegnare agli utenti del portale come proprietari o visualizzatori di ciascun progetto. I proprietari del progetto possono creare dashboard per visualizzare i dati e condividere il progetto con altri utenti. I visualizzatori del progetto possono visualizzare i pannelli di controllo ma non modificarli. Per ulteriori informazioni sui ruoli in SiteWise Monitor, consulta. [SiteWise Monitora i ruoli](#)

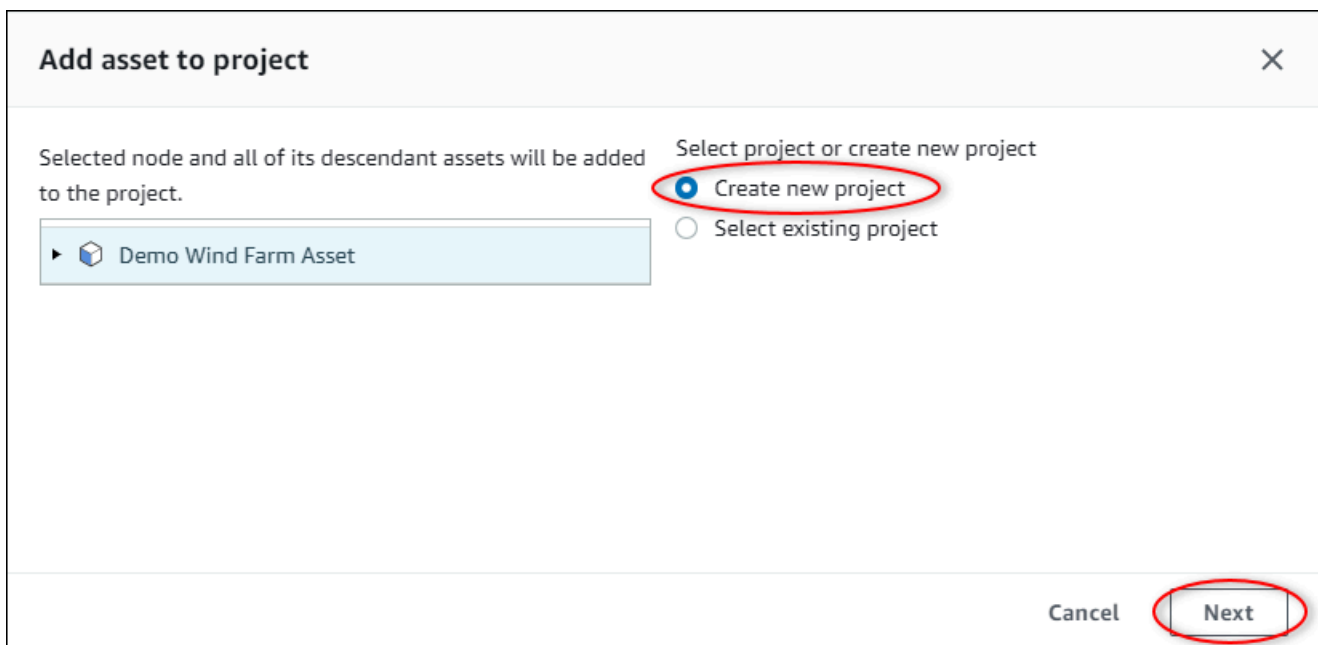
Per creare un progetto di centrale eolica

1. Nel riquadro di navigazione a sinistra del portale, scegli la scheda Risorse. Nella pagina Risorse, puoi esplorare tutte le risorse disponibili nel portale e aggiungere risorse ai progetti.

2. Nel browser delle risorse, scegliete Demo Wind Farm Asset. Quando scegli una risorsa, puoi esplorarne i dati storici e in tempo reale. Puoi anche premere Shift per selezionare più risorse e confrontarne i dati side-by-side.
3. Scegli Aggiungi risorsa al progetto in alto a sinistra. I progetti contengono i pannelli di controllo che gli utenti del portale possono visualizzare per esplorare i dati. Ogni progetto ha accesso a un sottoinsieme delle tue risorse in AWS IoT SiteWise. Quando si aggiunge un asset a un progetto, tutti gli utenti con accesso a tale progetto possono anche accedere ai dati relativi all'asset e ai relativi elementi figlio.



4. Nella finestra di dialogo Aggiungi risorsa al progetto, scegli Crea nuovo progetto, quindi scegli Avanti.



5. Nella finestra di dialogo Crea nuovo progetto, inserisci un nome e una descrizione del progetto per il tuo progetto, quindi scegli Aggiungi risorsa al progetto.

Create new project ✕

Project name

The project name can have up to 256 characters.

Project description

The project description can have up to 2048 characters.

Cancel Previous Add asset to project

Viene visualizzata la pagina del nuovo progetto.

6. Nella pagina del progetto, puoi aggiungere utenti del portale come proprietari o visualizzatori del progetto.

Note

Se non hai effettuato l'accesso al tuo account di AWS Organizations gestione, potresti non avere utenti del portale da assegnare a questo progetto, quindi puoi saltare questo passaggio.

In questa pagina, procedi come segue:

- a. In Proprietari del progetto, scegli Aggiungi proprietari o Modifica utenti.

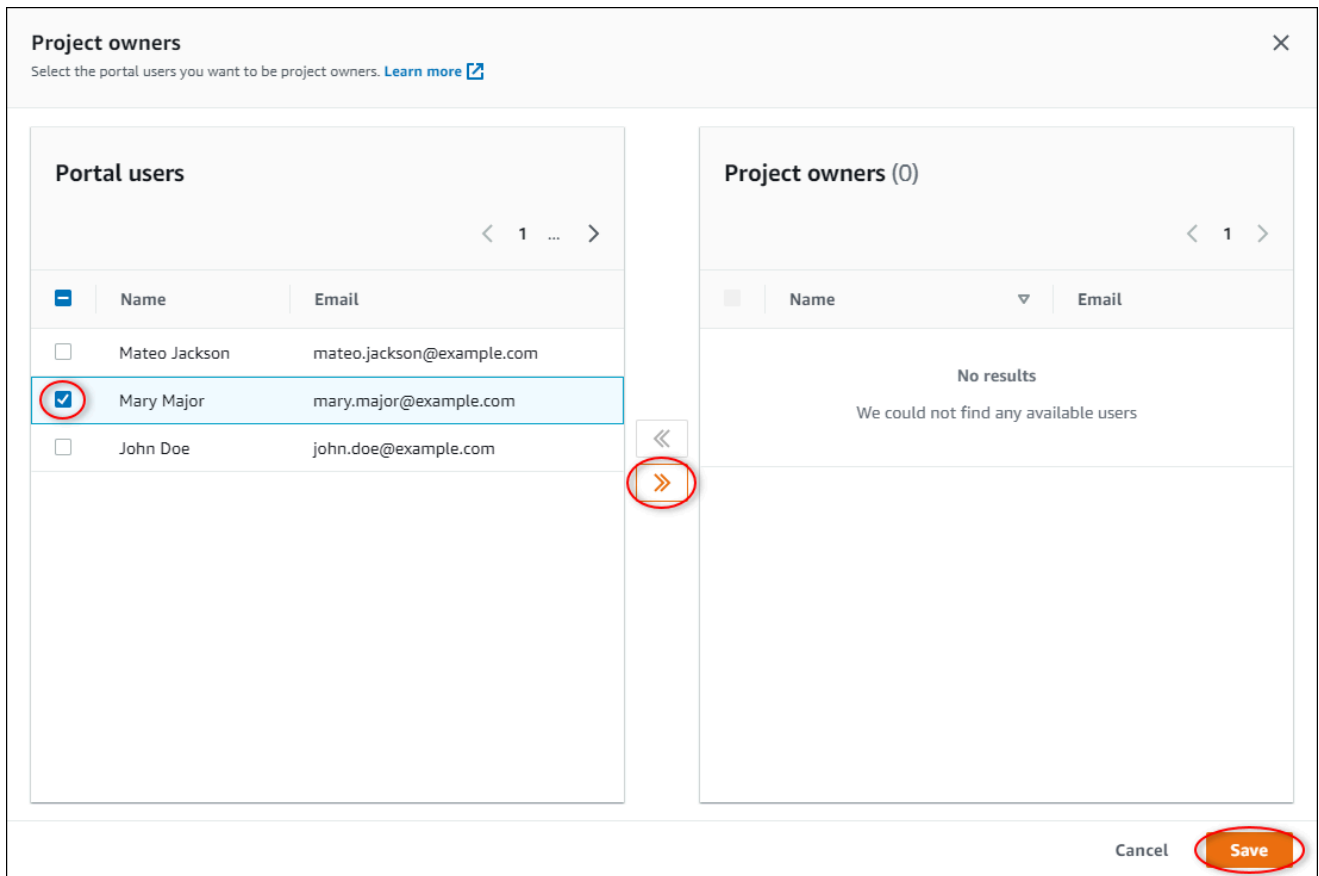
Project owners Send invitations Remove owners Edit owners

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

< 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. Scegli l'utente da aggiungere come proprietario del progetto (ad esempio, Mary Major), quindi scegliete l'icona >>.



- c. Seleziona Salva.

Il tuo utente IAM Identity Center Mary Major può accedere a questo portale per modificare le dashboard di questo progetto e condividerlo con altri utenti in questo portale.

- d. In Visualizzatori del progetto, scegli Aggiungi visualizzatori o Modifica utenti.
- e. Scegli l'utente da aggiungere come visualizzatore del progetto (ad esempio, Mateo Jackson), quindi scegliete l'icona >>.
- f. Seleziona Salva.

Il tuo utente IAM Identity Center Mateo Jackson può accedere a questo portale per visualizzare, ma non modificare, le dashboard del progetto del parco eolico.

Fase 4: Creare una dashboard per visualizzare i dati del parco eolico

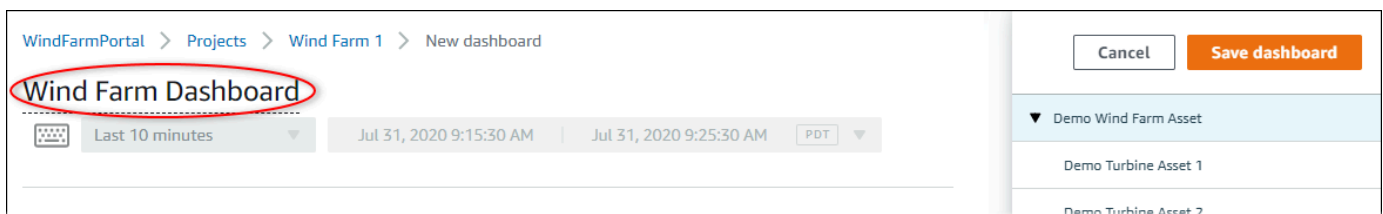
In questa procedura crei i pannelli di controllo per visualizzare i dati demo della centrale eolica. I pannelli di controllo contengono visualizzazioni personalizzabili dei dati degli asset del progetto. Ogni visualizzazione può avere un tipo diverso, ad esempio un grafico a linee, un grafico a barre o un indicatore chiave di prestazione (KPI). Puoi scegliere il tipo di visualizzazione più adatto ai dati. I proprietari dei progetti possono modificare le dashboard, mentre i visualizzatori dei progetti possono solo visualizzare le dashboard per ottenere informazioni dettagliate.

Per creare un pannello di controllo con visualizzazioni

1. Nella pagina del nuovo progetto, scegli Crea dashboard per creare una dashboard e aprirne la pagina di modifica.

Nella pagina di modifica di un pannello di controllo è possibile trascinare le proprietà degli asset dalla gerarchia degli asset nel pannello di controllo per creare le visualizzazioni. Quindi è possibile modificare il titolo della visualizzazione, i titoli della legenda, il tipo, le dimensioni e la posizione per ogni visualizzazione nel pannello di controllo.

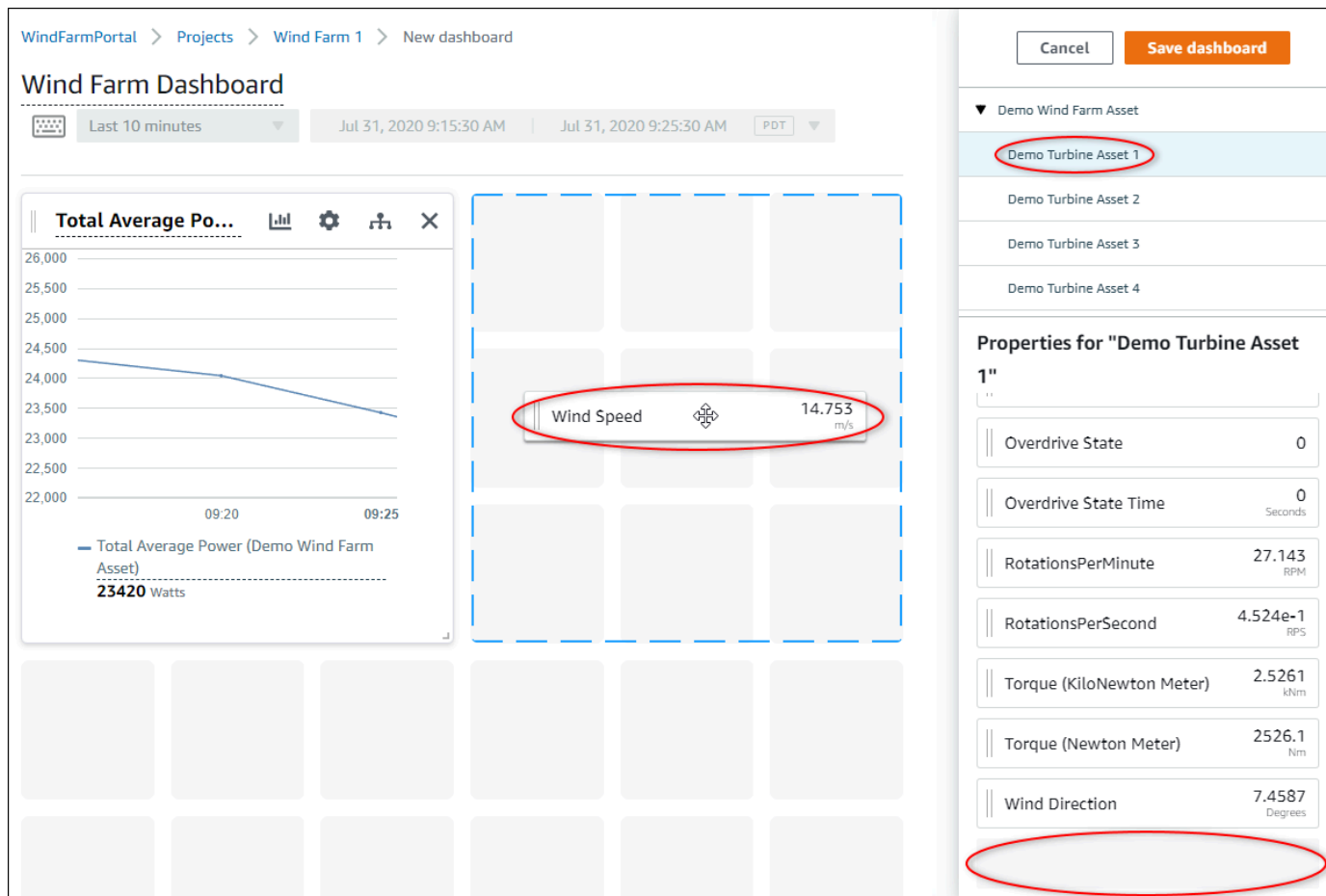
2. Inserisci un nome per la tua dashboard.



3. Trascina Total Average Power dagli Demo Wind Farm Asset nella dashboard per creare una visualizzazione.

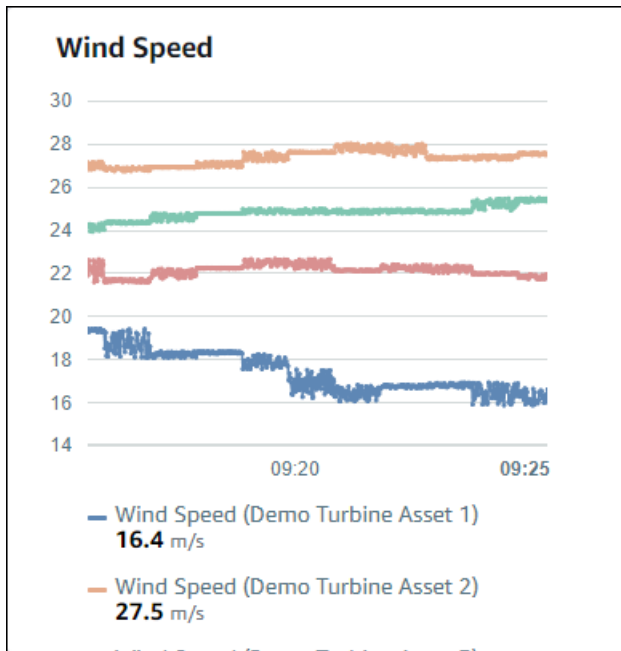
The screenshot displays the 'Wind Farm Dashboard' in the AWS IoT SiteWise console. The breadcrumb navigation at the top reads 'WindFarmPortal > Projects > Wind Farm 1 > New dashboard'. The dashboard title is 'Wind Farm Dashboard'. Below the title, there are filters for 'Last 10 minutes', a date range from 'Jul 31, 2020 9:15:30 AM' to 'Jul 31, 2020 9:25:30 AM', and a time zone selector set to 'PDT'. The main area contains a grid of 18 placeholder widgets. One widget, 'Total Average Power', is highlighted with a red oval and shows a value of '24038 Watts'. On the right side, there is a sidebar with a 'Cancel' button and a 'Save dashboard' button. Below these buttons is a section titled 'Demo Wind Farm Asset' containing a list of assets: 'Demo Turbine Asset 1', 'Demo Turbine Asset 2', 'Demo Turbine Asset 3', and 'Demo Turbine Asset 4'. At the bottom of the sidebar is a section titled 'Properties for "Demo Wind Farm Asset"' which includes a 'Code' field with the value '300' and a 'Total Overdrive State Time' field with the value '0 seconds'. A red oval highlights the empty space between the 'Code' and 'Total Overdrive State Time' fields.

4. Scegliere Demo Turbine Asset 1 per mostrare le proprietà di quella risorsa, quindi trascinare Wind Speed dalla dashboard per creare una visualizzazione della velocità del vento.

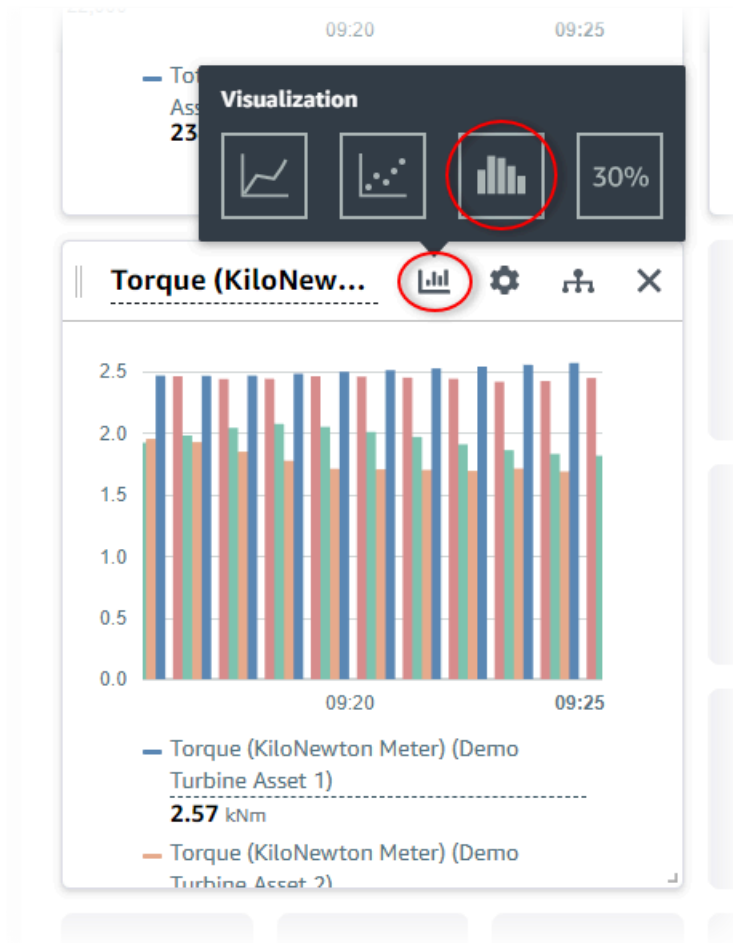


5. Add Wind Speed alla nuova visualizzazione della velocità del vento per ciascuno Demo Turbine Asset 2, 3, e 4(in quest'ordine).

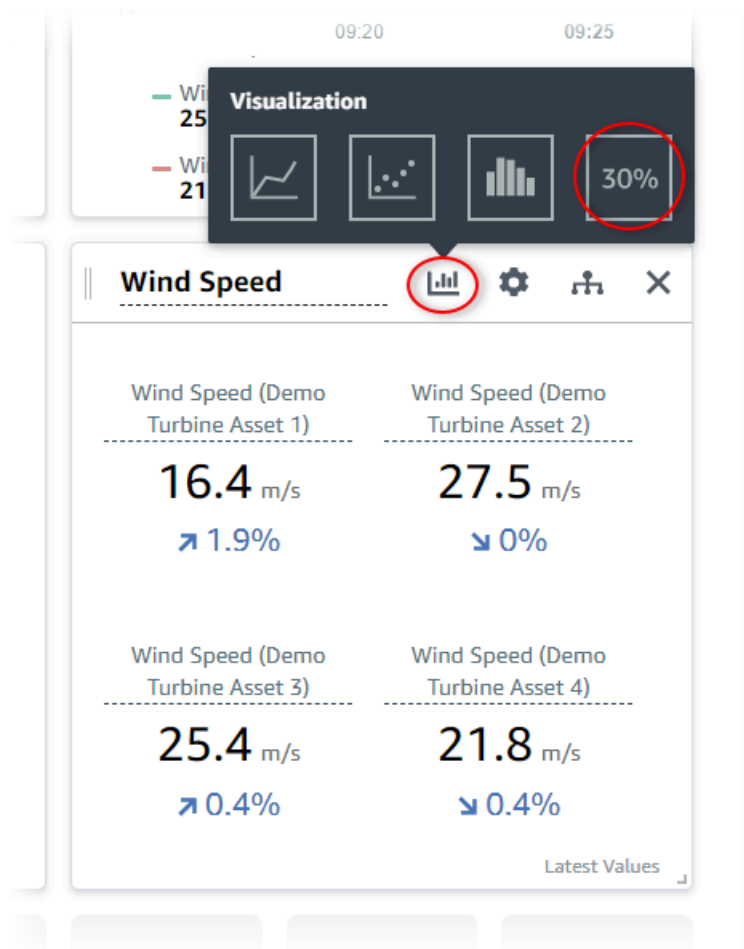
Vostro Wind Speedla visualizzazione dovrebbe essere simile alla schermata seguente.



6. Ripetere i passaggi 4 e 5 per le turbine eoliche» Torque (KiloNewton Meter) proprietà per creare una visualizzazione della coppia delle turbine eoliche.
7. Scegli l'icona del tipo di visualizzazione per Torque (KiloNewton Meter) visualizzazione, quindi scegli l'icona del grafico a barre.

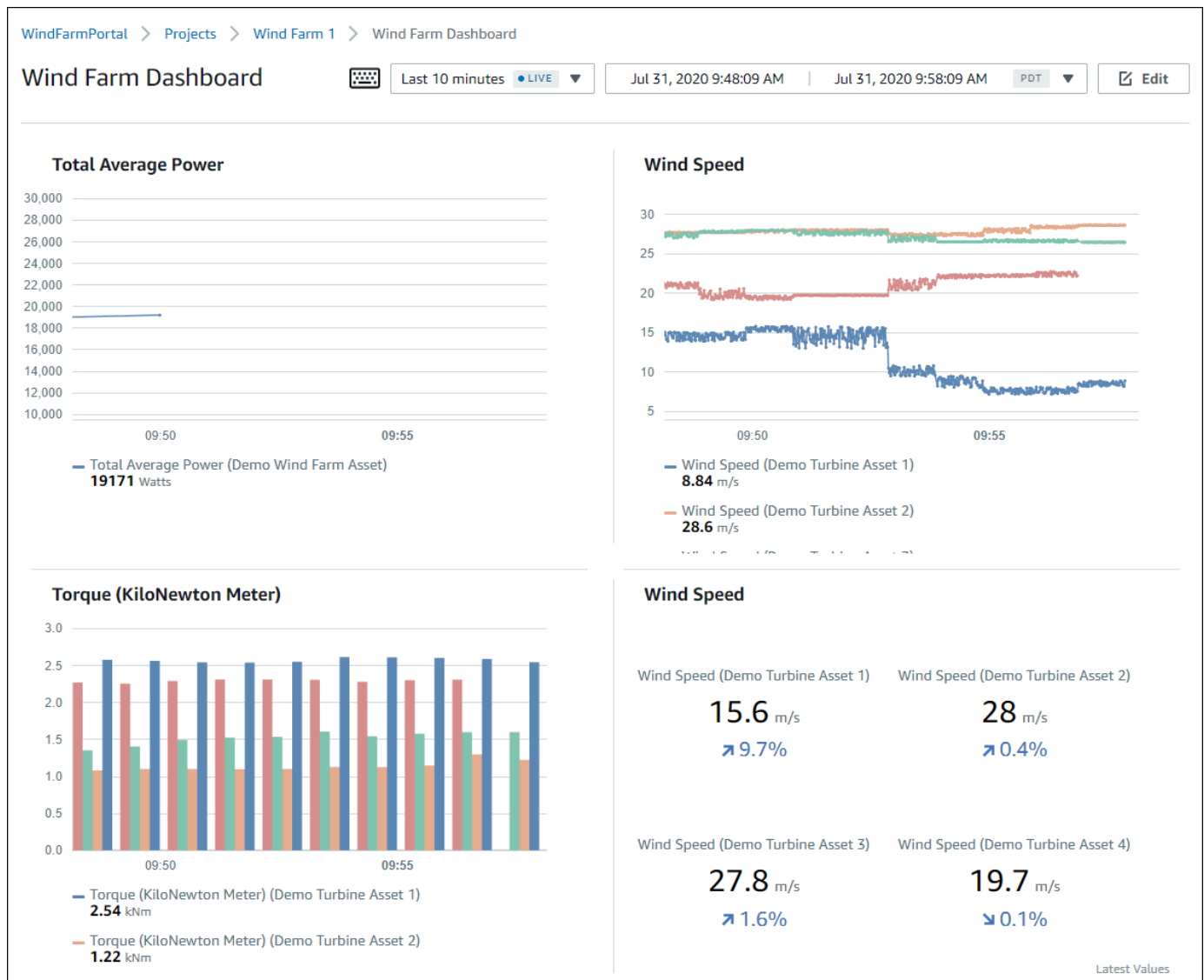


8. Ripeti i passaggi 4 e 5 per le turbine eoliche Wind Direction proprietà per creare una visualizzazione della direzione del vento.
9. Scegli l'icona del tipo di visualizzazione per Wind Direction visualizzazione, quindi scegli l'icona del grafico KPI (30%).



10. (Facoltativo) Apportare altre modifiche al titolo della visualizzazione, ai titoli della legenda, al tipo, alle dimensioni e alla posizione per ogni visualizzazione in base alle esigenze.
11. Scegli Salva dashboard in alto a destra per salvare la dashboard.

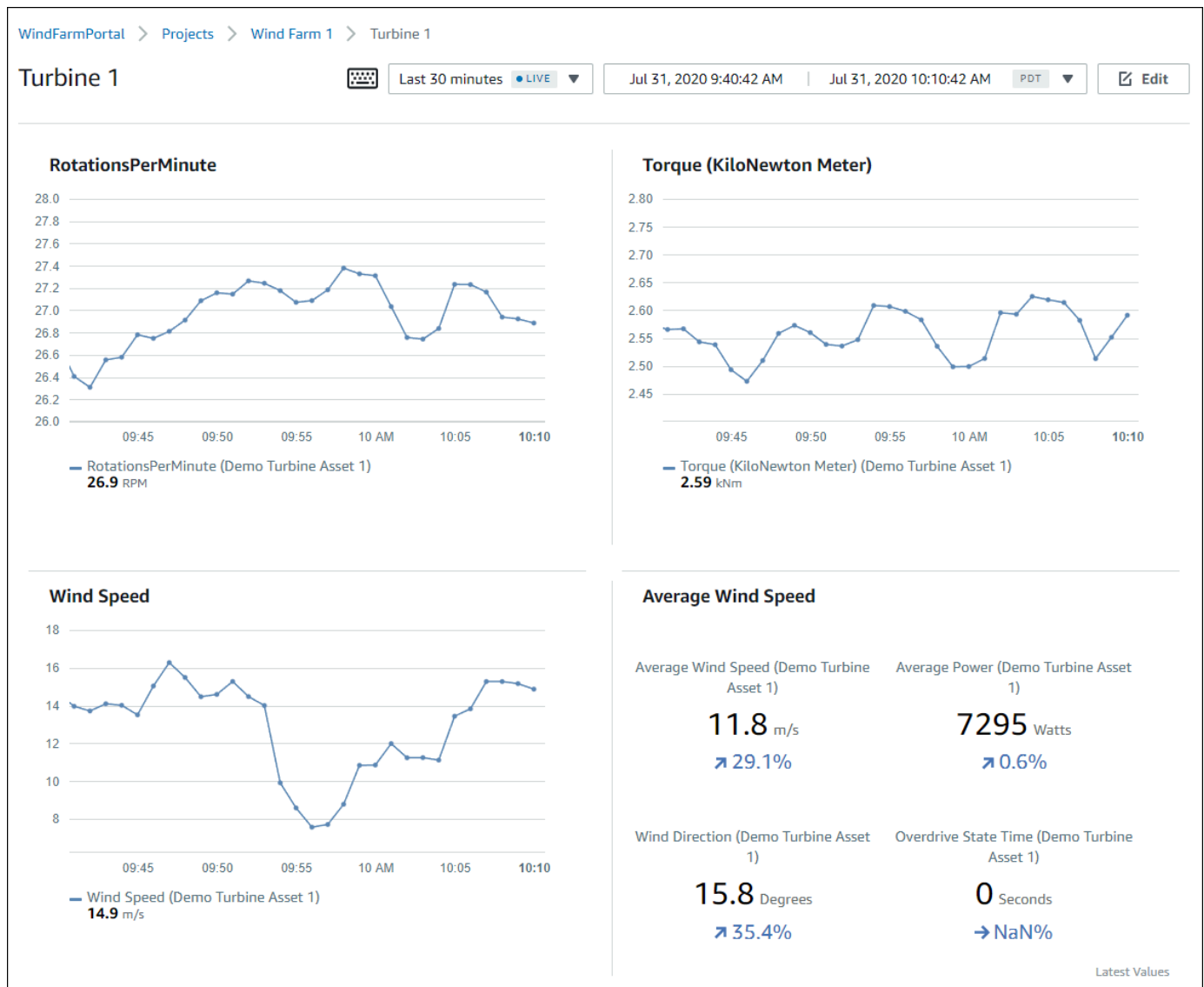
Il pannello di controllo è simile allo screenshot seguente.



12. (Facoltativo) Creare un pannello di controllo aggiuntivo per ogni asset turbina eolica.

Come procedura consigliata, consigliamo di creare un pannello di controllo per ogni asset in modo che i visualizzatori dei progetti possano esaminare eventuali problemi relativi a ogni singolo asset. È possibile aggiungere solo fino a 5 asset per ogni visualizzazione, pertanto è necessario creare più pannelli di controllo per gli asset gerarchici in molti scenari.

Un pannello di controllo per una turbina eolica demo è simile allo screenshot seguente.



13. (Facoltativo) Modificare la timeline o selezionare i punti dati in una visualizzazione per esplorare i dati nel pannello di controllo. Per ulteriori informazioni, consulta [Visualizzazione dei dashboard](#) nella Guida all'AWS IoT SiteWise Monitor applicazione.

Fase 5: Esplora il portale

In questa procedura, puoi esplorare il portale come utente con meno autorizzazioni rispetto a un amministratore del AWS IoT SiteWise portale.

Per esplorare il portale e completare il tutorial

- (Facoltativo) Se hai aggiunto altri utenti al progetto come proprietari o visualizzatori, puoi accedere al portale come tali utenti. Ciò consente di esplorare il portale come utente con meno autorizzazioni rispetto a un amministratore del portale.

 Important

Ti viene addebitato un costo per ogni utente che accede a un portale. Per ulteriori informazioni, consulta [AWS IoT SiteWise Prezzi](#).

Per esplorare il portale con altri utenti, procedi come segue:

- a. Scegli Esci nella parte inferiore sinistra del portale per uscire dall'applicazione web.
- b. Scegli Esci in alto a destra nel portale applicativo IAM Identity Center per disconnetterti dal tuo utente IAM Identity Center.
- c. Accedi al portale come utente IAM Identity Center che hai assegnato come proprietario del progetto o visualizzatore del progetto. Per ulteriori informazioni, consulta [Passaggio 2: accedi a un portale](#).

Hai completato il tutorial. Una volta terminata l'esplorazione del parco eolico dimostrativo in SiteWise Monitor, segui la procedura successiva per ripulire le risorse.

Passaggio 6: Pulisci le risorse dopo il tutorial

Una volta completato il tutorial puoi ripulire le risorse. Non viene addebitato alcun costo per AWS IoT SiteWise se gli utenti non accedono al portale, ma puoi eliminare il portale e gli utenti Elenco Centro identità AWS IAM. Gli asset della centrale eolica demo vengono eliminati alla fine della durata scelta al momento della creazione della demo oppure è possibile eliminare la demo manualmente. Per ulteriori informazioni, consulta [Eliminare la AWS IoT SiteWise demo](#).

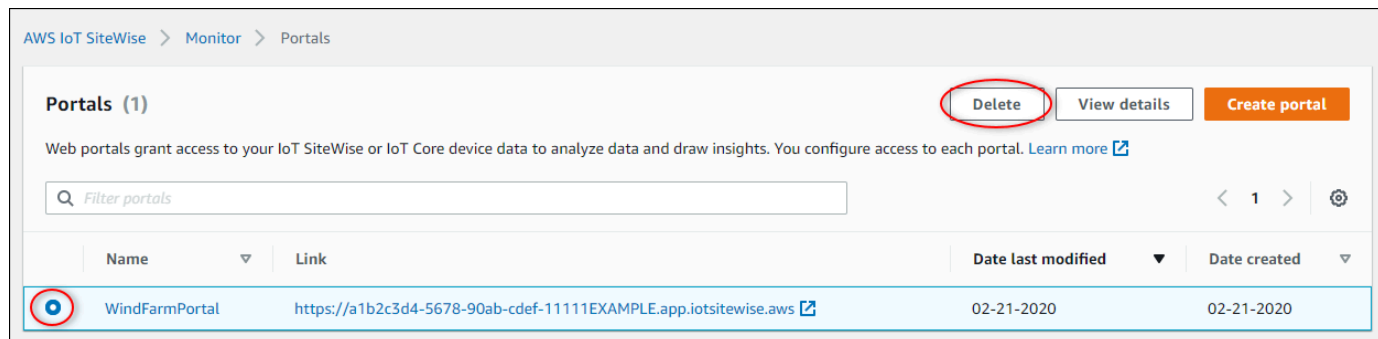
Utilizza le seguenti procedure per eliminare gli utenti del portale e di IAM Identity Center.

Per eliminare un portale

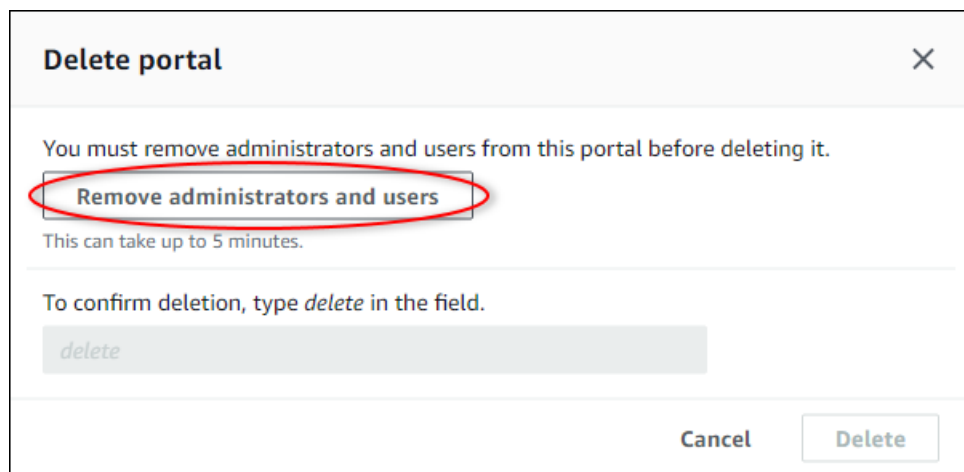
1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione a sinistra scegliere Portals (Portali).

3. Scegli il tuo portale WindFarmPortal, quindi scegli Elimina.

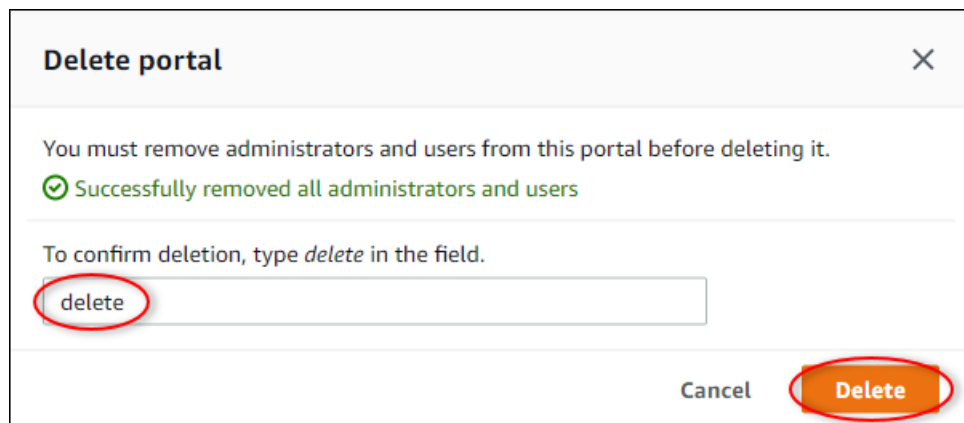
Quando si elimina un portale o un progetto, gli asset associati ai progetti eliminati non sono interessati.



4. Nella finestra di dialogo Elimina portale, scegli Rimuovi amministratori e utenti.

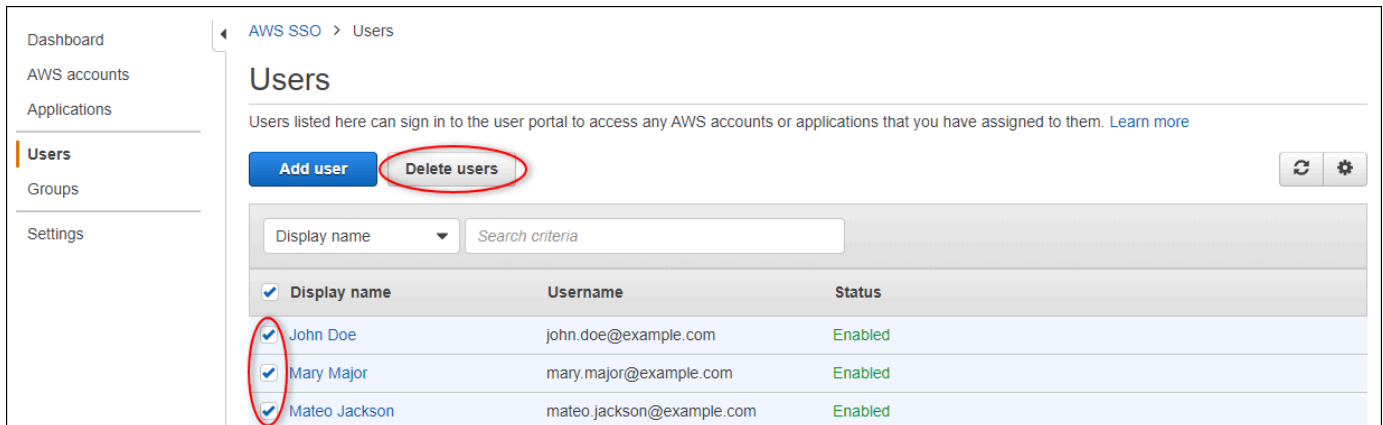


5. Immettere **delete** per confermare l'eliminazione, quindi scegliere Delete (Elimina).

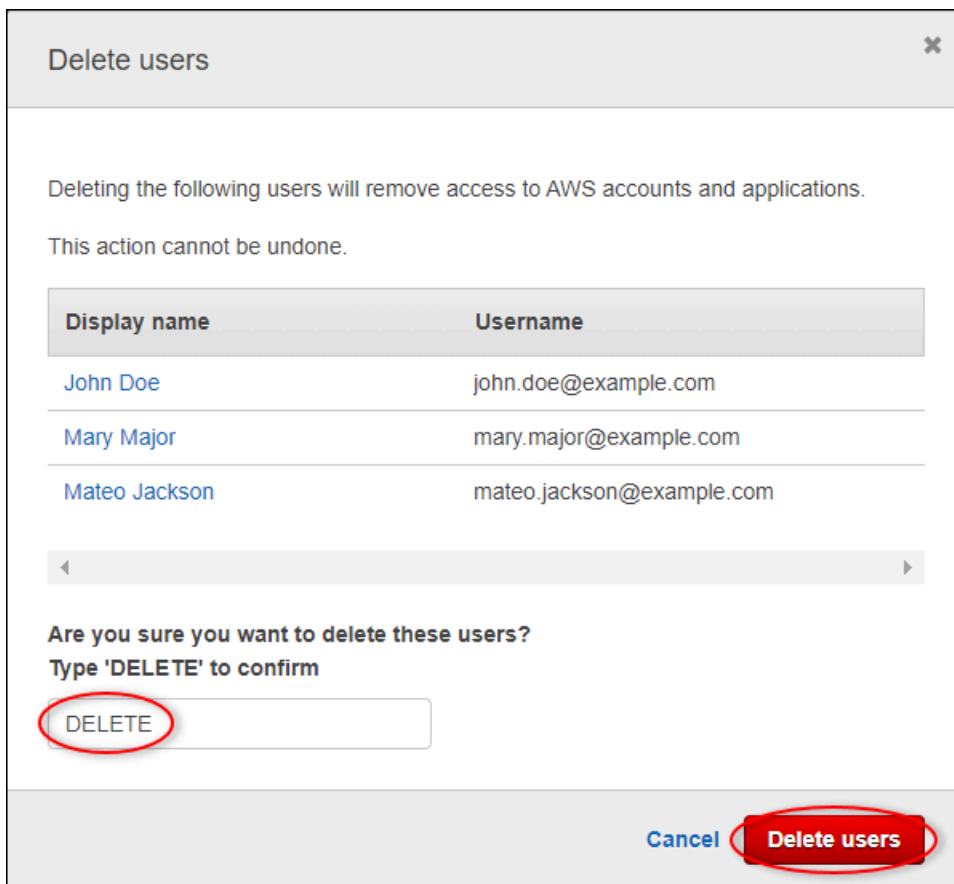


Per eliminare gli utenti di IAM Identity Center

1. Accedi alla [console IAM Identity Center](#).
2. Nel riquadro di navigazione a sinistra, seleziona Users (Utenti).
3. Selezionare la casella di controllo per ogni utente da eliminare, quindi scegliere Delete users (Elimina utenti).



4. Nella finestra di dialogo Elimina utenti, inserisci **DELETE**, quindi scegli Elimina utenti.



Pubblica aggiornamenti dei valori delle proprietà su Amazon DynamoDB

Questo tutorial introduce un modo pratico per archiviare i dati utilizzando [Amazon](#) DynamoDB, semplificando l'accesso ai dati storici degli asset senza dover interrogare ripetutamente l'API. AWS IoT SiteWise Dopo aver completato questo tutorial, puoi creare un software personalizzato che utilizza i dati degli asset, ad esempio una mappa in tempo reale della velocità e della direzione del vento su un intero parco eolico. Se desideri monitorare e visualizzare i tuoi dati senza implementare una soluzione software personalizzata, consulta [Monitora i dati con AWS IoT SiteWise Monitor](#)

In questo tutorial, ti baserai sulla AWS IoT SiteWise demo che fornisce un set di dati di esempio per un parco eolico. Puoi configurare gli aggiornamenti dei valori delle proprietà dalla demo del parco eolico per inviare dati, tramite le regole AWS IoT Core, a una tabella DynamoDB che crei. Quando abiliti gli aggiornamenti dei valori delle proprietà, AWS IoT SiteWise invia i dati AWS IoT Core nei messaggi MQTT. Quindi, definisci le regole di AWS IoT base che eseguono azioni, come l'azione DynamoDB, a seconda del contenuto di tali messaggi. Per ulteriori informazioni, consulta [Interagisci con altri AWS servizi](#).

Argomenti

- [Prerequisiti](#)
- [Fase 1: Configurazione AWS IoT SiteWise per la pubblicazione degli aggiornamenti dei valori delle proprietà](#)
- [Passaggio 2: crea una regola in Core AWS IoT](#)
- [Fase 3: Creare una tabella DynamoDB](#)
- [Fase 4: Configurare l'azione della regola DynamoDB](#)
- [Fase 5: Esplora i dati in DynamoDB](#)
- [Passaggio 6: Pulisci le risorse dopo il tutorial](#)

Prerequisiti

Per completare questo tutorial, è necessario quanto segue:

- Un AWS account. Se non lo hai, consultare [Configura un account AWS](#).
- Un computer di sviluppo che esegue Windows, macOS, Linux o Unix per accedere a. AWS Management Console Per ulteriori informazioni, consulta [Cos'è il? AWS Management Console](#)

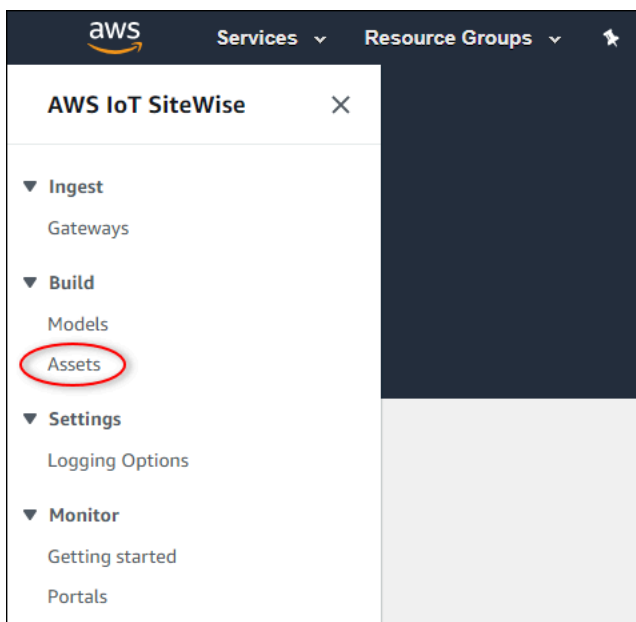
- Un utente IAM con autorizzazioni da amministratore.
- Una demo di un AWS IoT SiteWise parco eolico funzionante. Quando configuri la demo, definisce i modelli e gli asset AWS IoT SiteWise e trasmette loro i dati per rappresentare un parco eolico. Per ulteriori informazioni, consulta [Usa la AWS IoT SiteWise demo](#).

Fase 1: Configurazione AWS IoT SiteWise per la pubblicazione degli aggiornamenti dei valori delle proprietà

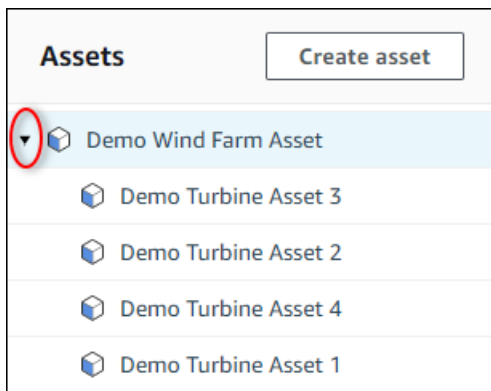
In questa procedura, abilitate le notifiche sul valore delle proprietà sugli asset della vostra turbina dimostrativa Wind Speed proprietà. Dopo aver abilitato le notifiche sui valori delle proprietà, AWS IoT SiteWise pubblica ogni aggiornamento del valore in un messaggio MQTT su AWS IoT Core.

Per abilitare le notifiche di aggiornamento del valore della proprietà sulle proprietà degli asset

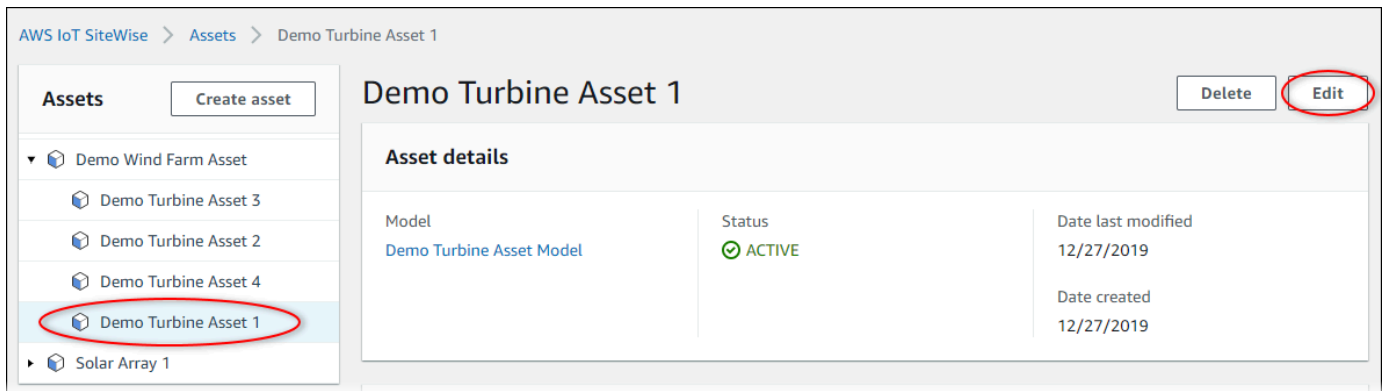
1. Accedere alla [console AWS IoT SiteWise](#).
2. Controlla gli [AWS IoT SiteWise endpoint e le quote](#) dove AWS IoT SiteWise è supportato e cambia AWS regione, se necessario. Passa a una regione in cui stai eseguendo la AWS IoT SiteWise demo.
3. Nel riquadro di navigazione a sinistra, scegli Asset.



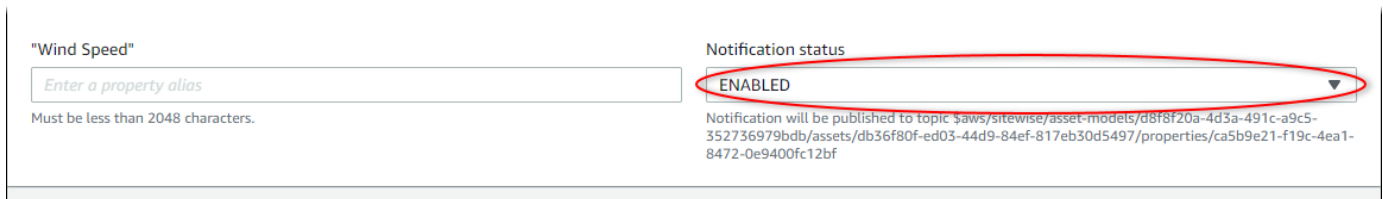
4. Scegli la freccia accanto a Demo Wind Farm Asset per espandere la gerarchia degli asset del parco eolico.



5. Scegli una turbina demo e seleziona Modifica.



6. Aggiornamento di Wind Speed stato di notifica della proprietà è impostato su ENABLED.



7. Scegli Salva risorsa nella parte inferiore della pagina.
8. Ripetere i passaggi da 5 a 7 per ogni risorsa della turbina demo.
9. Scegliete una turbina dimostrativa (ad esempio, Demo Turbine Asset 1).
10. Scegliere Measurements (Misurazioni).
11. Scegli l'icona di copia accanto a Wind Speed proprietà per copiare l'argomento della notifica negli appunti. Salvare l'argomento di notifica da utilizzare più avanti in questo tutorial. È sufficiente registrare l'argomento di notifica da una turbina.

Torque (KiloNewton Meter)	–	⊖ Disabled	–	2.128123
Wind Speed	–	✔ Enabled	<code>\$aws/sitewise/asset-models/d8f8f20a-4d3a-491c-a9c5-352736979bdb/assets/db36f80f-ed03-44d9-84ef-817eb30d5497/properties/ca5b9e21-f19c-4ea1-8472-0e9400fc12bf</code>	26.49812

L'argomento di notifica dovrebbe essere simile all'esempio seguente.

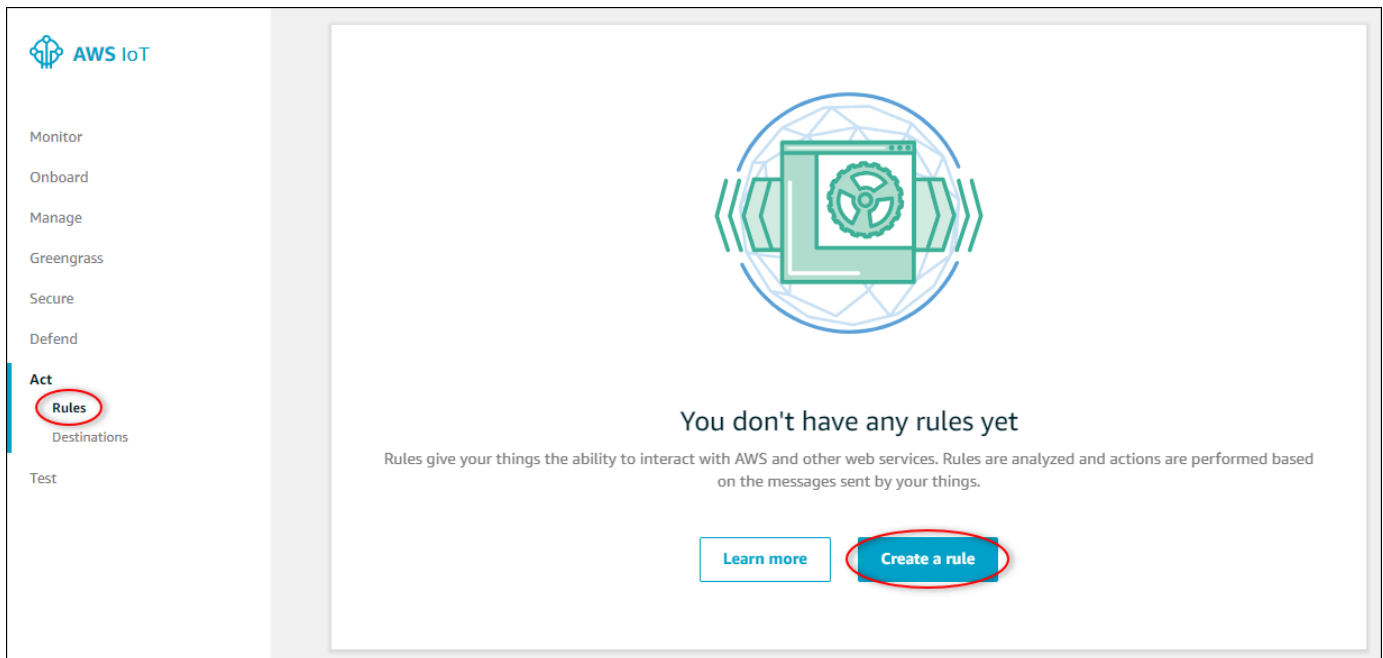
```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/  
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-  
cdef-33333EXAMPLE
```

Passaggio 2: crea una regola in Core AWS IoT

In questa procedura, crei una regola in AWS IoT Core che analizza i messaggi di notifica del valore della proprietà e inserisce i dati in una tabella Amazon DynamoDB. AWS IoT Core regola di base analizzano i messaggi MQTT ed eseguono azioni in base al contenuto e all'argomento di ciascun messaggio. Quindi, crei una regola con un'azione DynamoDB per inserire dati in una tabella DynamoDB che crei come parte di questo tutorial.

Per creare una regola con un'azione DynamoDB

1. Passare alla [console AWS IoT](#). Se viene visualizzato il pulsante Get started (Inizia), sceglierlo.
2. Nel riquadro di navigazione sinistro scegliere Atti e quindi Regole.



3. Se viene visualizzata la finestra di dialogo You don't have any rules yet (Non hai ancora regole), selezionare Create a rule (Crea una regola). In caso contrario, scegliere Create (Crea).
4. Inserire un nome e una descrizione per la regola.

Create a rule

Create a rule to evaluate messages sent by your things and specify what to do when a message is received (for example, write data to a DynamoDB table or invoke a Lambda function).

Name

Description

5. Individuare l'argomento di notifica salvato in precedenza in questo tutorial.

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

Sostituisci l'ID della risorsa (l'ID dopo `assets/`) nell'argomento con un `+`. Questo seleziona la proprietà della velocità del vento per tutte le turbine eoliche dimostrative. Il filtro dell'argomento `+` accetta tutti i nodi da un singolo livello in un argomento. L'argomento dovrebbe essere simile all'esempio seguente.

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

6. Immettere la seguente istruzione di query per la regola. Sostituire l'argomento nella sezione FROM con l'argomento di notifica.

```
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. In **Set one or more actions** (Imposta una o più operazioni), scegliere **Add action** (Aggiungi operazione).

8. Nella pagina **Seleziona un'azione**, scegli **Dividi messaggio in più colonne di una tabella DynamoDB (Dynamo). DBv2**

9. Scegliere **Configura azione** nella parte inferiore della pagina.
10. Nella pagina **Configure action** (Configura operazione), scegli **Create a new resource** (Crea una nuova risorsa).

La console DynamoDB si apre in una nuova scheda. Tenere aperta la scheda **Azione regola** mentre si completano le procedure riportate di seguito.

Fase 3: Creare una tabella DynamoDB

In questa procedura, crei una tabella Amazon DynamoDB per ricevere i dati sulla velocità del vento dall'azione della regola.

Per creare una tabella DynamoDB

1. Nella dashboard della console DynamoDB, scegli Crea tabella.
2. Immettere un nome per la tabella.

Create DynamoDB table Tutorial ?

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* ⓘ

Primary key* Partition key

ⓘ

☒ Add sort key

ⓘ

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

☒ Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".
- Encryption at Rest with DEFAULT encryption type.

ⓘ You do not have the required role to enable Auto Scaling by default.
Please refer to [documentation](#).

+ Add tags **NEW!**

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

Cancel Create

3. Per Chiave primaria esegui queste operazioni:
 - a. Inserisci **timestamp** come chiave di partizione.
 - b. Scegli il tipo Numero.
 - c. Selezionare la casella di controlloAggiungi chiave di ordinamento).

- d. Immettere **asset** come chiave di ordinamento e lasciare il tipo di chiave di ordinamento predefinito String.
4. Scegli Create (Crea) .

Quando l'avviso tabella in fase di creazione scompare, la tabella è pronta.

5. Tornare alla scheda con la pagina Configura azione. Tieni aperta la scheda DynamoDB mentre completi le seguenti procedure.

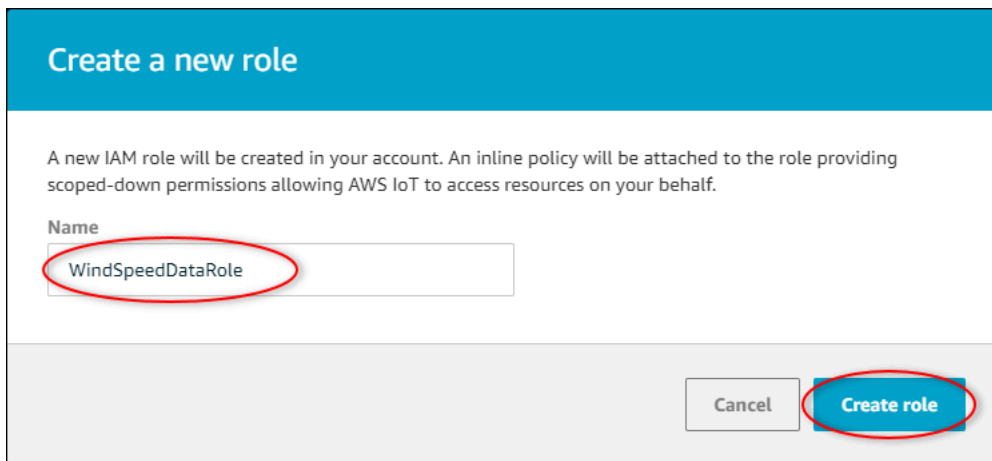
Fase 4: Configurare l'azione della regola DynamoDB

In questa procedura, configuri l'azione della regola di Amazon DynamoDB per inserire i dati dagli aggiornamenti dei valori delle proprietà nella tua nuova tabella DynamoDB.

Per configurare l'azione della regola DynamoDB

1. Nella pagina Configura azione, aggiorna l'elenco dei nomi delle tabelle e scegli la nuova tabella DynamoDB.

2. Scegli Crea ruolo per creare un ruolo IAM che conceda l'accesso AWS IoT Core per eseguire l'azione della regola.
3. Fornire un nome ruolo e selezionare Crea ruolo.



4. Selezionare Add action (Aggiungi operazione).
5. Scegliere Crea regola nella parte inferiore della pagina per completare la creazione della regola.

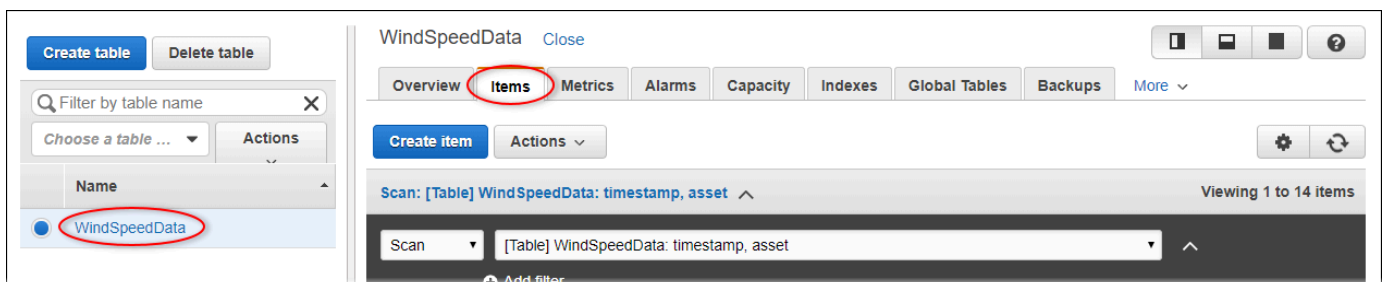
I dati degli asset dimostrativi dovrebbero iniziare a comparire nella tabella DynamoDB.

Fase 5: Esplora i dati in DynamoDB

In questa procedura, esplori i dati sulla velocità del vento degli asset dimostrativi nella tua nuova tabella Amazon DynamoDB.

Per esplorare i dati degli asset in DynamoDB

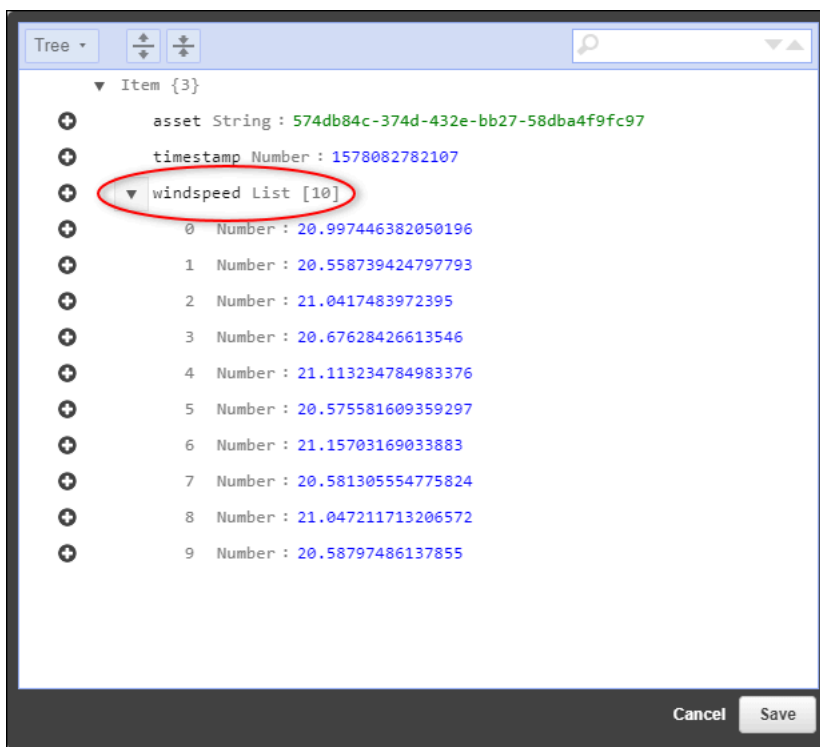
1. Tornate alla scheda con la tabella DynamoDB aperta.
2. Nella tabella creata in precedenza, scegliere la scheda Elementi per visualizzare i dati nella tabella. Aggiorna la pagina se non vedi righe nella tabella. Se le righe non vengono visualizzate dopo alcuni minuti, consulta [Risoluzione dei problemi relativi a una regola \(DynamoDB\)](#).



3. In una riga nella tabella, scegliere l'icona di modifica per espandere i dati.

	timestamp ⓘ	asset	windspeed
☐	1578093637414	db36f80f-ed03-44d9-84ef-817eb30d5497	[{"N": "40.18707553698584"}, {"N": "40.20834808480326"}, {"N": "40.21081344172715"}]
☐	1578093637422	db36f80f-ed03-44d9-84ef-817eb30d5497	[{"N": "40.21081344172715"}, {"N": "40.218280888809424"}, {"N": "40.22691091326525"}]
☐	1578093637451	db36f80f-ed03-44d9-84ef-817eb30d5497	[{"N": "40.218912043562895"}, {"N": "40.22691091326525"}, {"N": "40.22876939941959"}]
☐	1578093637453	db36f80f-ed03-44d9-84ef-817eb30d5497	[{"N": "40.22876939941959"}, {"N": "40.21820505495924"}, {"N": "40.21081344172715"}]

4. Scegli la freccia accanto a windspeedstruttura per espandere l'elenco dei punti dati sulla velocità del vento. Ogni elenco riporta una serie di punti di dati sulla velocità del vento inviati AWS IoT SiteWise dalla demo del parco eolico. Se si imposta un'azione regola per uso personale, è possibile che si desideri un formato di dati diverso. Per ulteriori informazioni, consulta [Interroga le notifiche relative alle proprietà degli asset in AWS IoT SiteWise](#).



Ora che hai completato il tutorial, disabilita o elimina la regola ed elimina la tabella DynamoDB per evitare di incorrere in costi aggiuntivi. Per ripulire le risorse, consulta. [Passaggio 6: Pulisci le risorse dopo il tutorial](#)

Passaggio 6: Pulisci le risorse dopo il tutorial

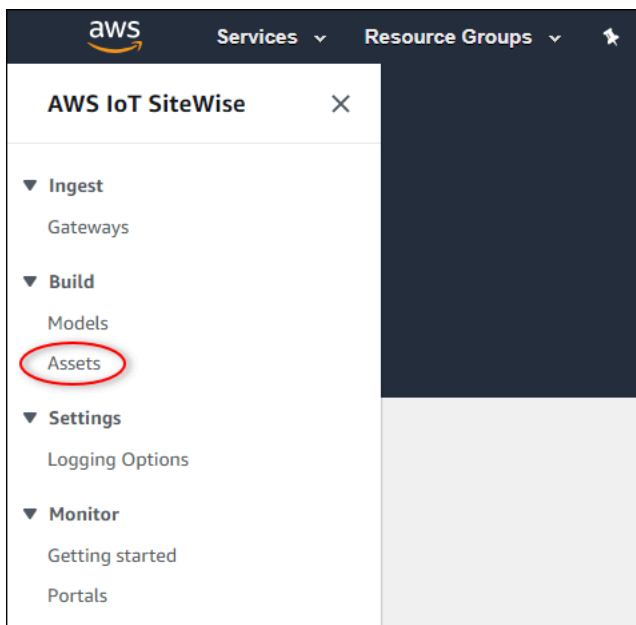
Dopo aver completato questo tutorial, puoi pulire le risorse per evitare di incorrere in costi aggiuntivi. Gli asset dimostrativi del parco eolico vengono eliminati al termine della durata scelta al momento

della creazione della demo. Puoi anche eliminare la demo manualmente. Per ulteriori informazioni, consulta [Eliminare la AWS IoT SiteWise demo](#).

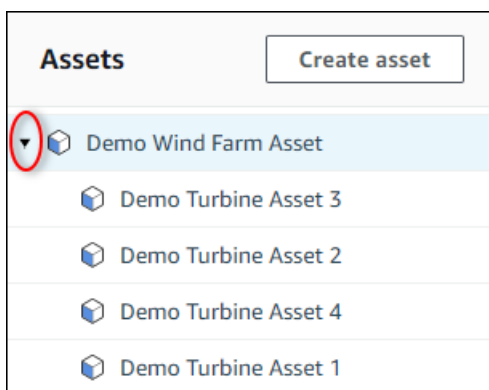
Utilizza le seguenti procedure per disabilitare le notifiche di aggiornamento dei valori delle proprietà (se non hai eliminato la demo), disabilitare o eliminare la AWS IoT regola ed eliminare la tabella DynamoDB.

Per disabilitare le notifiche di aggiornamento del valore della proprietà sulle proprietà degli asset

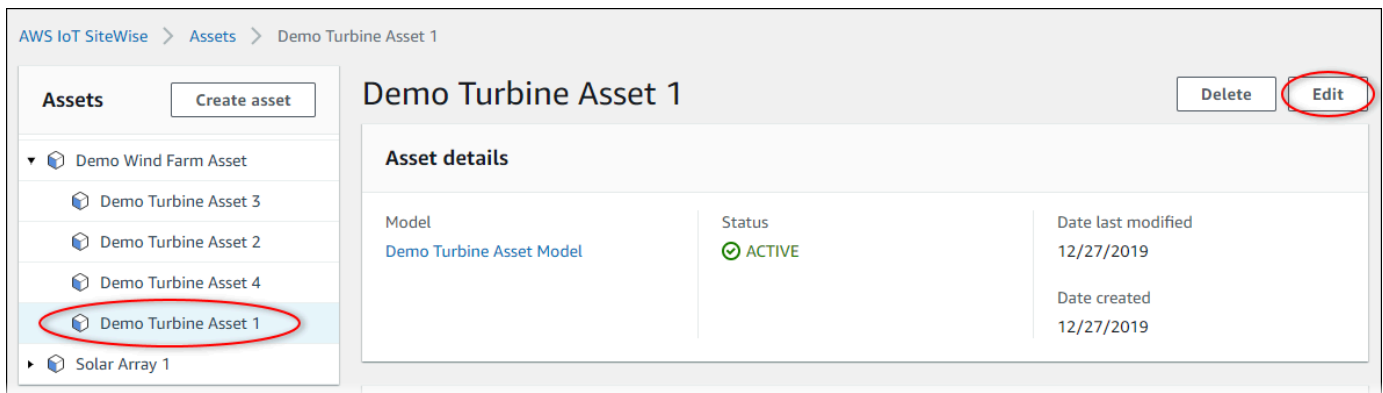
1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione a sinistra, scegli Asset.



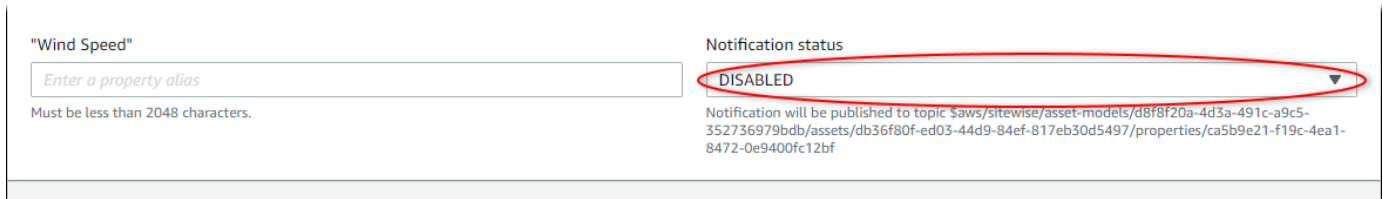
3. Scegli la freccia accanto a Demo Wind Farm Asset per espandere la gerarchia degli asset del parco eolico.



4. Scegli una turbina demo e seleziona Modifica.



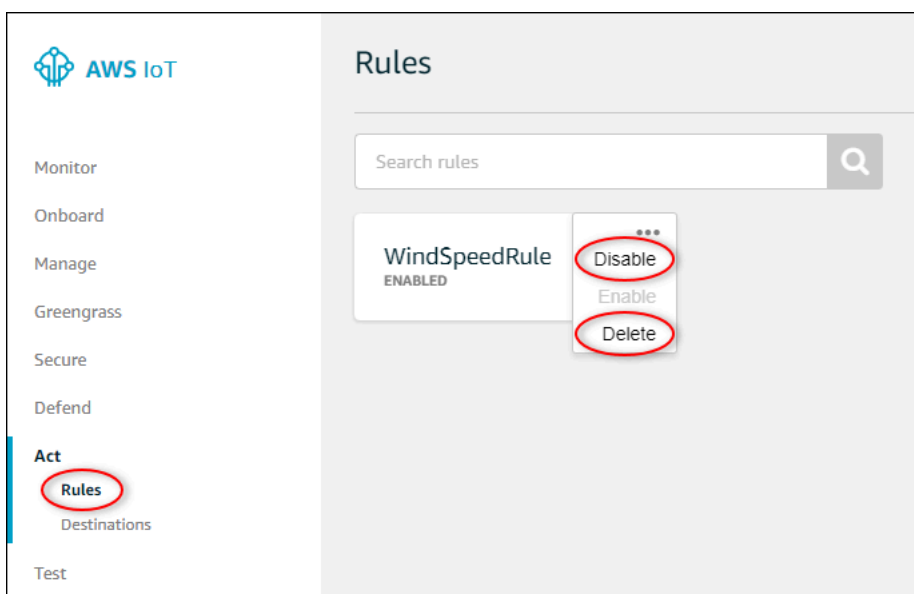
5. Aggiornamento di Wind Speedlo stato di notifica della proprietà è impostato su DISABILITATO.



6. Scegli Salva risorsa nella parte inferiore della pagina.
7. Ripetere i passaggi da 4 a 6 per ogni asset turbina demo.

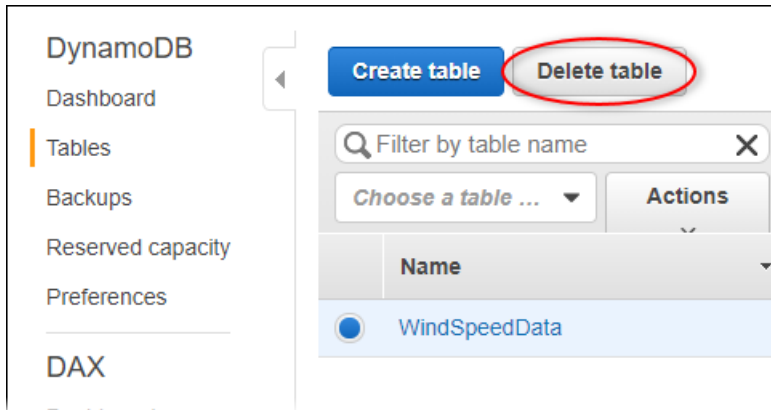
Per disabilitare o eliminare una regola in AWS IoT Core

1. Passare alla [console AWS IoT](#).
2. Nel riquadro di navigazione sinistro scegliere Atti e quindi Regole.
3. Scegli il menu della regola e seleziona Disattiva o Elimina.

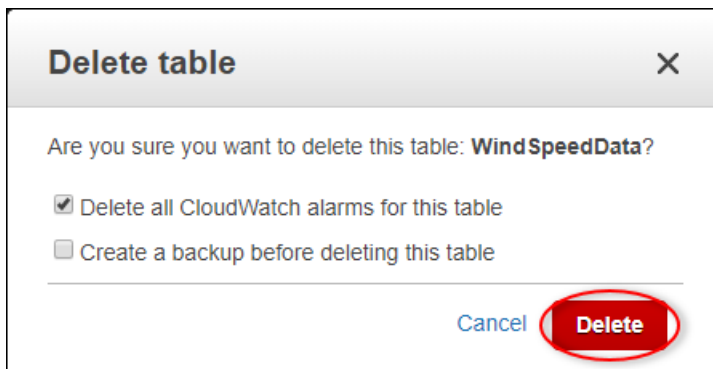


Per eliminare una tabella DynamoDB

1. Accedere alla console [DynamoDB](#).
2. Nel riquadro di navigazione a sinistra, selezionare Tables (Tabelle).
3. Scegli la tabella che hai creato in precedenza, WindSpeedData.
4. Seleziona Delete Table (Elimina tabella).



5. Nella finestra di dialogo Delete table (Elimina tabella) scegliere Delete (Elimina).



Inserisci dati in AWS IoT SiteWise

AWS IoT SiteWise è progettato per raccogliere e correlare in modo efficiente i dati industriali con gli asset corrispondenti, che rappresentano vari aspetti delle operazioni industriali. Questa documentazione si concentra sugli aspetti pratici dell'acquisizione dei dati e offre diversi metodi adattati a diversi casi d'uso industriali. AWS IoT SiteWise Per le istruzioni sulla creazione di un'operazione industriale virtuale, consulta [Asset industriali modello](#).

È possibile inviare dati industriali a AWS IoT SiteWise utilizzando una delle seguenti opzioni:

- AWS IoT SiteWise Edge: utilizza il [gateway SiteWise Edge](#) come intermediario tra AWS IoT SiteWise e i tuoi server di dati. AWS IoT SiteWise fornisce AWS IoT Greengrass componenti che è possibile implementare su qualsiasi piattaforma in grado di eseguire la configurazione AWS IoT Greengrass di un SiteWise gateway Edge. Questa opzione supporta il collegamento con il protocollo server [OPC UA](#).
- AWS IoT SiteWise API: utilizza l'[AWS IoT SiteWise API](#) per caricare dati da qualsiasi altra fonte. Utilizza la nostra [BatchPutAssetPropertyValue](#) API di streaming per l'inserimento in pochi secondi o l'[CreateBulkImportJob](#) API orientata ai batch per facilitare l'ingestione economica in batch di grandi dimensioni.
- AWS IoT Regole di base: utilizza le regole di [AWS IoT base per caricare dati dai messaggi MQTT](#) pubblicati da un oggetto o da un altro servizio. AWS IoT AWS
- AWS IoT Events azioni: utilizza le [AWS IoT Events azioni attivate](#) da eventi specifici in. AWS IoT Events Questo metodo è adatto per scenari in cui il caricamento dei dati è legato al verificarsi di eventi.
- AWS IoT Greengrass stream manager: utilizza [AWS IoT Greengrass stream manager](#) per caricare dati da fonti di dati locali utilizzando un dispositivo edge. Questa opzione si adatta a situazioni in cui i dati provengono da postazioni locali o periferiche.

Questi metodi offrono una gamma di soluzioni per la gestione dei dati da diverse fonti. Approfondisci i dettagli di ciascuna opzione per ottenere una comprensione completa delle funzionalità di acquisizione dei dati offerte. AWS IoT SiteWise

Gestisci i flussi di dati per AWS IoT SiteWise

Un flusso di dati è la risorsa che contiene i dati storici delle serie temporali. Ogni flusso di dati è identificato da un alias univoco, che semplifica la registrazione dell'origine di ogni dato. I flussi di dati

vengono creati automaticamente AWS IoT SiteWise quando vengono ricevuti i dati della prima serie temporale. Se i dati della prima serie temporale vengono identificati con un alias, AWS IoT SiteWise crea un nuovo flusso di dati con quell'alias, a condizione che a quell'alias non sia già assegnato alcuna proprietà della risorsa. In alternativa, se i primi dati della serie temporale vengono identificati con un ID di risorsa e un ID di proprietà, AWS IoT SiteWise crea un nuovo flusso di dati e lo associa alla proprietà dell'asset.

Esistono due modi per assegnare un alias a una proprietà dell'asset. Il metodo utilizzato dipende dal fatto che i dati vengano inviati per AWS IoT SiteWise primi o che venga creata prima una risorsa.

- Se i dati vengono inviati per AWS IoT SiteWise primi, questo crea automaticamente un flusso di dati con l'alias assegnato. Quando la risorsa viene creata in un secondo momento, utilizzate l' [AssociateTimeSeriesToAssetProperty](#) API per associare il flusso di dati e il relativo alias alla proprietà della risorsa.
- Se una risorsa viene creata per prima, utilizzate l' [UpdateAssetProperty](#) API per assegnare un alias a una proprietà della risorsa. Quando i dati vengono successivamente inviati a AWS IoT SiteWise, il flusso di dati viene creato automaticamente e associato alla proprietà dell'asset.

Attualmente, è possibile associare solo flussi di dati alle misurazioni. Le misurazioni sono un tipo di proprietà degli asset che rappresentano i flussi di dati grezzi dei sensori dei dispositivi, come i valori di temperatura con data e ora o i valori di rotazione al minuto (RPM) con data e ora.

Quando queste misurazioni definiscono metriche o trasformazioni, i dati in entrata attivano calcoli specifici. È importante notare che una proprietà di un asset può essere collegata solo a un flusso di dati alla volta.

AWS IoT SiteWise utilizza TimeSeries la risorsa Amazon Resource Name (ARN) per determinare i costi di storage. Per ulteriori informazioni, consulta [AWS IoT SiteWise Prezzi](#).

Le seguenti sezioni mostrano come utilizzare la AWS IoT SiteWise console o l'API per gestire i flussi di dati.

Argomenti

- [Configura autorizzazioni e impostazioni](#)
- [Associa un flusso di dati a una proprietà dell'asset](#)
- [Dissocia un flusso di dati da una proprietà dell'asset](#)
- [Eliminare un flusso di dati](#)

- [Aggiorna l'alias di proprietà di un asset](#)
- [Scenari comuni](#)

Configura autorizzazioni e impostazioni

I flussi di dati vengono creati automaticamente AWS IoT SiteWise quando vengono ricevuti i primi dati della serie temporale. Se i dati importati non sono associati a una proprietà della risorsa, AWS IoT SiteWise crea un nuovo flusso di dati dissociato che è configurabile per essere associato a una proprietà della risorsa. Configura il controllo di accesso del gateway a cui invia i dati AWS IoT SiteWise, utilizzando le policy IAM per specificare il tipo di dati da importare.

La seguente policy IAM disabilita l'inserimento di dati disassociati dal gateway, pur consentendo l'inserimento di dati nei flussi di dati associati a una proprietà dell'asset:

Example Policy utente IAM che disabilita l'inserimento di dati disassociati dal gateway

```
{
  "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 Policy utente IAM che disabilita l'inserimento di tutti i dati dal gateway

```

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

```

Associa un flusso di dati a una proprietà dell'asset

Gestisci i tuoi flussi di dati utilizzando o. Console AWS IoT SiteWise AWS CLI

Console

Usa la AWS IoT SiteWise console per gestire i tuoi flussi di dati.

Per gestire i flussi di dati (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Flussi di dati.
3. Scegli un flusso di dati filtrando in base all'alias del flusso di dati o selezionando Flussi di dati non associati nel menu a discesa del filtro.

4. Seleziona il flusso di dati da aggiornare. È possibile selezionare più flussi di dati. Fai clic su Gestisci flussi di dati in alto a destra.
5. Seleziona il flusso di dati da associare da Aggiorna associazioni di flussi di dati e fai clic sul pulsante Scegli la misurazione.
6. Nella sezione Scegli la misurazione, trova la proprietà di misurazione degli asset corrispondente. Seleziona la misura, quindi fai clic su Scegli.
7. Esegui i passaggi 4 e 5 per gli altri flussi di dati selezionati nel passaggio 3. Assegna le proprietà delle risorse a tutti i flussi di dati.
8. Scegliete Aggiorna per confermare le modifiche. Viene visualizzato un banner di conferma dell'avvenuto aggiornamento.

AWS CLI

Per associare un flusso di dati (identificato dal relativo alias) a una proprietà dell'asset (identificata dal suo ID), esegui il comando seguente:

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

Dissocia un flusso di dati da una proprietà dell'asset

Console

Utilizza la AWS IoT SiteWise console per dissociare il flusso di dati dalla proprietà di una risorsa.

Per dissociare i flussi di dati da una proprietà della risorsa (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Flussi di dati.
3. Scegli un flusso di dati filtrando in base all'alias del flusso di dati o selezionando Flussi di dati associati nel menu a discesa del filtro.
4. Seleziona il flusso di dati da dissociare. La colonna degli alias del flusso di dati deve contenere un alias. Le colonne Asset name e Asset property name devono contenere i valori

della proprietà dell'asset a cui è associato il flusso di dati. È possibile selezionare più flussi di dati.

5. Fai clic su Gestisci flussi di dati in alto a destra.
6. Nella sezione Aggiorna le associazioni dei flussi di dati, fai clic su X nella colonna Nome della misurazione. Uno submitted stato dovrebbe apparire nella colonna Stato.
7. Scegli Aggiorna per confermare le modifiche. Il flusso di dati è ora dissociato dalla proprietà dell'asset e l'alias viene ora utilizzato per identificare il flusso di dati.

AWS CLI

Per dissociare un flusso di dati da una proprietà dell'asset (identificata dalla relativa ID s e dal relativo alias), esegui il comando seguente:

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

Il flusso di dati è ora dissociato dalla proprietà dell'asset e l'alias viene utilizzato per identificare il flusso di dati. L'alias non è più associato alla proprietà dell'asset, poiché ora è associato al flusso di dati.

Eliminare un flusso di dati

Quando una proprietà viene rimossa da un modello di asset, AWS IoT SiteWise elimina le proprietà e i relativi flussi di dati da tutte le risorse gestite dal modello di asset. Elimina inoltre tutte le proprietà e i relativi flussi di dati di una risorsa quando la risorsa viene eliminata. Se i dati di un flusso di dati devono essere conservati, è necessario dissociarli dalla proprietà dell'asset prima di essere eliminati.

Warning

Quando una proprietà viene eliminata da una risorsa, viene eliminato anche il flusso di dati associato. Per preservare il flusso di dati, dissociatelo prima dalla proprietà della risorsa, prima di eliminare la proprietà dal modello di asset o eliminare la risorsa.

Console

Utilizzate la AWS IoT SiteWise console per dissociare il flusso di dati dalla proprietà di una risorsa.

Per eliminare un flusso di dati (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Flussi di dati.
3. Scegli un flusso di dati filtrandolo in base all'alias del flusso di dati.
4. Seleziona il flusso di dati da eliminare. È possibile selezionare più flussi di dati.
5. Scegli il pulsante Elimina per eliminare il flusso di dati.

AWS CLI

Utilizza l' [DeleteTimeSeries](#) API per eliminare un flusso di dati specifico, tramite il relativo alias.

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

Aggiorna l'alias di proprietà di un asset

Gli alias devono essere univoci all'interno di una AWS regione. Ciò include gli alias sia delle proprietà degli asset che dei flussi di dati. Non assegnate un alias a una proprietà della risorsa, se quell'alias è utilizzato da un'altra proprietà o da un altro flusso di dati.

Console

Utilizzate la AWS IoT SiteWise console per aggiornare l'alias di una proprietà della risorsa.

Per aggiornare un alias di proprietà di un asset (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Asset.
3. Selezionate la risorsa dalla tabella.
4. Fai clic sul pulsante Edit (Modifica).

5. Selezionate il tipo di proprietà nella tabella Proprietà.
6. Trova la proprietà e digita il nuovo alias nel campo di testo dell'alias della proprietà.
7. Fate clic sul pulsante Salva per salvare le modifiche.

AWS CLI

Per aggiornare un alias su una proprietà dell'asset, esegui il comando seguente:

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

Note

Se le notifiche di proprietà sono attualmente abilitate, devono essere fornite nuovamente per garantire che continuino a essere abilitate.

Scenari comuni

Sposta un flusso di dati

Per modificare l'associazione di un flusso di dati con un'altra proprietà dell'asset, dissociate innanzitutto il flusso di dati dalla proprietà corrente dell'asset. Quando si dissocia un flusso di dati da una proprietà della risorsa, deve essere assegnato un alias a quella proprietà della risorsa.

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

Ora riassegna il flusso di dati alla nuova proprietà dell'asset.

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

Errore durante l'assegnazione di un alias a una proprietà dell'asset

Quando si utilizza l'UpdateAssetPropertyAPI per assegnare un alias a una proprietà, è possibile che venga visualizzato il seguente messaggio di errore:

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

Questo messaggio di errore indica che l'alias non è assegnato alla proprietà, perché è attualmente utilizzato da un'altra proprietà o da un flusso di dati.

Ciò accade se i dati vengono importati AWS IoT SiteWise con un alias. Quando i dati vengono inviati con un alias non utilizzato da un altro flusso di dati o proprietà di un asset, viene creato un nuovo flusso di dati con quell'alias. Le due opzioni seguenti risolvono il problema.

- Utilizza l'AssociateTimeSeriesToAssetPropertyAPI per associare il flusso di dati al relativo alias alla proprietà dell'asset.
- Interrompi temporaneamente l'ingestione dei dati ed elimina il flusso di dati. Utilizza l'UpdateAssetPropertyAPI per assegnare l'alias alla proprietà dell'asset, quindi riattiva l'inserimento dei dati.

Errore durante l'associazione di un flusso di dati a una proprietà dell'asset

Quando si associa un flusso di dati a una proprietà dell'asset, viene visualizzato il seguente messaggio di errore.

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

Questo messaggio di errore indica che la proprietà dell'asset è già associata a un flusso di dati contenente dati. Tale flusso di dati deve essere dissociato o eliminato prima di associare un altro flusso di dati alla proprietà della risorsa.

 Note

Quando si dissocia un flusso di dati da una proprietà dell'asset, l'alias assegnato alla proprietà viene assegnato al flusso di dati. Affinché l'alias rimanga assegnato alla proprietà, assegnate un nuovo alias a quella proprietà prima di dissociare il flusso di dati.

Per conservare i dati memorizzati nella proprietà dell'asset, procedi come segue:

- Assicurati che nessun dato venga inserito nella proprietà dell'asset, per evitare la creazione di un nuovo flusso di dati.
- Utilizza l'UpdateAssetPropertyAPI per impostare un nuovo alias da assegnare al flusso di dati attualmente assegnato.
- Utilizza l'DisassociateTimeSeriesFromAssetPropertyAPI per dissociare il flusso di dati corrente dalla proprietà dell'asset.
- Utilizza l'AssociateTimeSeriesToAssetPropertyAPI per associare il flusso di dati desiderato alla proprietà dell'asset.

Se i dati memorizzati nella proprietà dell'asset devono essere eliminati, procedi come segue:

- Assicuratevi che nessun dato venga importato nella proprietà dell'asset, per evitare la creazione di un nuovo flusso di dati.
- Utilizza l>DeleteTimeSeriesAPI per eliminare il flusso di dati attualmente assegnato.
- Utilizza l'AssociateTimeSeriesToAssetPropertyAPI per associare il flusso di dati desiderato alla proprietà dell'asset.

Inserisci dati con AWS IoT SiteWise APIs

AWS IoT SiteWise APIs Utilizzateli per inviare dati industriali con data e ora agli attributi e alle proprietà di misurazione degli asset. APIs Accetta strutture contenenti payload (TQV). timestamp-quality-value

BatchPutAssetPropertyValue API

Utilizzare l'operazione [BatchPutAssetPropertyValue](#) per caricare i dati. Con questa operazione, puoi caricare più immissioni di dati alla volta per raccogliere dati da più dispositivi e inviarli tutti in un'unica richiesta.

Important

L'[BatchPutAssetPropertyValue](#) operazione è soggetta alle seguenti quote:

- Fino a 10 [iscrizioni](#) per richiesta.
- Fino a 10 [valori di proprietà](#) (punti dati TQV) per ingresso.
- AWS IoT SiteWise rifiuta tutti i dati con un timestamp datato a più di 7 giorni nel passato o a più di 10 minuti nel futuro.

Per ulteriori informazioni su queste quote, consulta la voce [BatchPutAssetPropertyValue](#) nella documentazione di riferimento delle API di AWS IoT SiteWise .

Per identificare la proprietà di una risorsa, specificate una delle seguenti opzioni:

- La proprietà `assetId` e `propertyId` della risorsa a cui vengono inviati i dati.
- `IlpropertyAlias`, che è un alias del flusso di dati (ad esempio, `/company/windfarm/3/turbine/7/temperature`). Per utilizzare questa opzione, è necessario prima impostare l'alias della proprietà dell'asset. Per impostare gli alias delle proprietà, vedere. [Gestisci i flussi di dati per AWS IoT SiteWise](#)

L'esempio seguente illustra come inviare le letture della temperatura e delle rotazioni al minuto (RPM) di una turbina eolica da un payload archiviato in un file JSON.

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

Il payload in di esempio `batch-put-payload.json` ha il seguente contenuto.

```
{
  "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"
    }
  ]
}

```

```
]
}
```

Specificando `enablePartialEntryProcessing` as è `true` possibile l'ingestione di tutti i valori che non provocano errori. Il comportamento predefinito è `false`. Se un valore non è valido, l'intera voce non viene inserita.

Ogni voce nel payload contiene un `entryId` che è possibile definire come una qualsiasi stringa univoca. Se una richiesta non riesce, ciascun errore conterrà l'`entryId` della richiesta corrispondente in modo che sia possibile sapere quale richiesta riprovare.

Ogni struttura nell'elenco di `propertyValues` è una struttura timestamp-quality-value (TQV) che contiene `value`, `a` e facoltativamente `timestamp`. `quality`

- `value`— Una struttura che contiene uno dei seguenti campi, a seconda del tipo di proprietà impostata:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`
- `nullValue`— Una struttura con il seguente campo che indica il tipo di valore della proprietà con valore Null e qualità o. BAD UNCERTAIN
 - `valueType`— Enumerazione di {"B», «D», «S», «I"}
- `timestamp`— Una struttura che contiene l'ora attuale dell'epoca Unix in secondi, `timeInSeconds`. È inoltre possibile impostare la `offsetInNanos` chiave nella `timestamp` struttura se si dispone di dati temporalmente precisi. AWS IoT SiteWise rifiuta tutti i punti dati con `timestamp` più vecchi di 7 giorni nel passato o più recenti di 10 minuti nelle future.
- `quality`— (Facoltativo) Una delle seguenti stringhe di qualità:
 - `GOOD`— (Impostazione predefinita) I dati non sono interessati da alcun problema.
 - `BAD`— I dati sono interessati da un problema, ad esempio un guasto del sensore.
 - `UNCERTAIN`— I dati sono influenzati da un problema come l'imprecisione del sensore.

Per ulteriori informazioni su come AWS IoT SiteWise gestisce la qualità dei dati nei calcoli, vedi [Qualità dei dati nelle espressioni di formule](#).

CreateBulkImportJob API

Usa l'`CreateBulkImportJobAPI` per importare grandi quantità di dati da Amazon S3. I tuoi dati devono essere salvati in formato CSV in Amazon S3. I file di dati possono avere le seguenti colonne.

Note

I dati precedenti alle 00:00:00 UTC del 1° gennaio 1970 non sono supportati.
Per identificare la proprietà di un asset, specificate una delle seguenti opzioni.

- La proprietà `ASSET_ID` e `PROPERTY_ID` della risorsa a cui stai inviando i dati.
- `IIALIAS`, che è un alias del flusso di dati (ad esempio, `/company/windfarm/3/turbine/7/temperature`). Per utilizzare questa opzione, è necessario prima impostare l'alias della proprietà dell'asset. Per informazioni relative all'impostazione degli alias delle proprietà, consultare [the section called "Gestione dei flussi di dati"](#).

- `ALIAS`— L'alias che identifica la proprietà, ad esempio il percorso del flusso di dati del server OPC UA (ad esempio, `/company/windfarm/3/turbine/7/temperature`). Per ulteriori informazioni, consulta [Gestisci i flussi di dati per AWS IoT SiteWise](#).
- `ASSET_ID`— L'ID della risorsa.
- `PROPERTY_ID`— L'ID della proprietà dell'asset.
- `DATA_TYPE`— Il tipo di dati della proprietà può essere uno dei seguenti.
 - `STRING`— Una stringa con un massimo di 1024 byte.
 - `INTEGER`— Un numero intero con segno a 32 bit con intervallo `[-2.147.483.648, 2.147.483.647]`.
 - `DOUBLE`— Un numero in virgola mobile con intervallo `[-10^100, 10^100]` e precisione doppia IEEE 754.
 - `BOOLEAN`— `true` oppure `false`.
- `TIMESTAMP_SECONDS`— Il timestamp del punto dati, in Unix Epoch Time.
- `TIMESTAMP_NANO_OFFSET`— L'offset in nanosecondi coperto da `TIMESTAMP_SECONDS`.
- `QUALITY`— (Facoltativo) La qualità del valore della proprietà del bene. Il valore può essere uno dei seguenti.
 - `GOOD`— (Impostazione predefinita) I dati non sono interessati da alcun problema.
 - `BAD`— I dati sono interessati da un problema, ad esempio un guasto del sensore.

- **UNCERTAIN**— I dati sono influenzati da un problema come l'imprecisione del sensore.

Per ulteriori informazioni su come AWS IoT SiteWise gestisce la qualità dei dati nei calcoli, consulta [Qualità dei dati nelle espressioni delle formule](#).

- **VALUE**— Il valore della proprietà dell'asset.

Example file di dati in formato.csv

```
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 fornisce le seguenti operazioni API per creare un processo di importazione in blocco e ottenere informazioni su un lavoro esistente.

- [CreateBulkImportJob](#)— Crea un nuovo processo di importazione in blocco.
- [DescribeBulkImportJob](#)— Recupera informazioni su un processo di importazione in blocco.
- [ListBulkImportJob](#)— Recupera un elenco impaginato di riepiloghi di tutti i lavori di importazione in blocco.

Crea un processo di importazione in blocco AWS IoT SiteWise ()AWS CLI

Utilizza l'operazione [CreateBulkImportJob](#) API per trasferire dati da Amazon S3 a AWS IoT SiteWise. L'[CreateBulkImportJob](#) API consente l'ingestione di grandi volumi di dati storici e l'ingestione bufferizzata di flussi di dati analitici in piccoli lotti. Fornisce una primitiva economica per l'ingestione dei dati. Gli esempi seguenti utilizzano AWS CLI.

⚠ Important

Prima di creare un processo di importazione in blocco, è necessario abilitare AWS IoT SiteWise il livello caldo o il livello freddo. AWS IoT SiteWise Per ulteriori informazioni, consulta [Configurare le impostazioni di archiviazione in AWS IoT SiteWise](#).

L'[CreateBulkImportJob](#) API supporta l'inserimento di dati storici AWS IoT SiteWise con l'opzione di impostare il parametro. `adaptive-ingestion-flag`

- Se impostata su `false`, l'API acquisisce i dati storici senza attivare calcoli o notifiche.
- Se impostata su `true`, l'API acquisisce nuovi dati, calcola le metriche e trasforma i dati per ottimizzare le analisi e le notifiche continue entro sette giorni.

Esegui il comando seguente. `file-name` Sostituiscilo con il nome del file che contiene la configurazione del processo di importazione in blocco.

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

Example Configurazione del processo di importazione in blocco

Di seguito sono riportati alcuni esempi di impostazioni di configurazione:

- Sostituisci `adaptive-ingestion-flag` con `true` o `false`.
 - Se impostato su `false`, il processo di importazione in blocco inserisce i dati storici in. AWS IoT SiteWise
 - Se impostato su `true`, il processo di importazione in blocco esegue le seguenti operazioni:
 - Inserisce nuovi dati in. AWS IoT SiteWise
 - Calcola metriche e trasformazioni e supporta notifiche per i dati con un timestamp entro sette giorni.
- Sostituisci `delete-files-after-import-flag` con `true` per eliminare i dati dal bucket di dati Amazon S3 dopo averli inseriti in uno storage di livello caldo. AWS IoT SiteWise
- Sostituisci `amzn-s3-demo-bucket` con `-for-errors` il nome del bucket Amazon S3 a cui vengono inviati gli errori associati a questo processo di importazione in blocco.
- Sostituisci `amzn-s3-demo-bucket` con `-for-errors-prefix` il prefisso del bucket Amazon S3 a cui vengono inviati gli errori associati a questo processo di importazione in blocco.

Amazon S3 utilizza il prefisso come nome di cartella per organizzare i dati nel bucket. Ogni oggetto Amazon S3 ha una chiave che è il suo identificatore univoco nel bucket. Per ogni oggetto in un bucket è presente esattamente una chiave. Il prefisso deve terminare con una barra (/). Per ulteriori informazioni, consulta [Organizing object using prefixes](#) nella Amazon Simple Storage Service User Guide.

- Sostituisci `amzn-s3-demo-bucket` con `-data` il nome del bucket Amazon S3 da cui vengono importati i dati.
- Sostituisci `data-bucket-key` con la chiave dell'oggetto Amazon S3 che contiene i tuoi dati. Ogni oggetto ha una chiave che è un identificatore univoco. Ogni oggetto ha esattamente una chiave.
- Sostituiscilo `data-bucket-version-id` con l'ID della versione per identificare una versione specifica dell'oggetto Amazon S3 che contiene i tuoi dati. Questo parametro è facoltativo.
- Sostituisci `column-name` con il nome della colonna specificato nel file.csv.
- `job-name` Sostituiscilo con un nome univoco che identifichi il processo di importazione in blocco.
- Sostituisci `job-role-arn` con il ruolo IAM che consente di AWS IoT SiteWise leggere i dati di Amazon S3.

Note

Assicurati che il tuo ruolo disponga delle autorizzazioni mostrate nell'esempio seguente. Sostituisci `amzn-s3-demo-bucket` con `-data` il nome del bucket Amazon S3 che contiene i tuoi dati. Inoltre, `amzn-s3-demo-bucket-for-errors` sostituiscilo con il nome del bucket Amazon S3 a cui vengono inviati gli errori associati a questo processo di importazione in blocco.

```
{
  "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"
}
```

Descrivi un processo di AWS IoT SiteWise importazione in blocco ()AWS CLI

Utilizza l'operazione [DescribeBulkImportJob](#) API per recuperare informazioni su uno specifico processo di importazione in blocco. AWS IoT SiteWise Questa operazione restituisce dettagli come lo stato del lavoro, l'ora di creazione e le informazioni sull'errore in caso di esito negativo del processo. È possibile utilizzare questa operazione per monitorare l'avanzamento del lavoro e risolvere i problemi. Per utilizzarla `DescribeBulkImportJob`, è necessario l'ID del lavoro dell'`CreateBulkImportJob` operazione. L'API restituisce le seguenti informazioni:

- Elenco dei file importati, incluse le posizioni e le chiavi dei bucket Amazon S3
- Posizione del report di errore (se applicabile)
- Dettagli di configurazione del lavoro, come il formato del file e i nomi delle colonne CSV
- Data di creazione di posti di lavoro e ultimo aggiornamento
- Stato attuale del lavoro (ad esempio, se il lavoro è in corso, completato o non riuscito)
- Ruolo IAM (ARN) utilizzato per il processo di importazione

Per i lavori completati, esamina i risultati per confermare la corretta integrazione dei dati. Se un processo fallisce, esamina i dettagli dell'errore per diagnosticare e risolvere i problemi.

Sostituiscilo *job-ID* con l'ID del processo di importazione in blocco che desideri recuperare.

```
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"
}

```

Elenca i lavori di importazione AWS IoT SiteWise in blocco (AWS CLI)

Utilizza l'operazione [ListBulkImportJobs](#) API per recuperare un elenco di riepiloghi per i lavori di importazione in blocco in AWS IoT SiteWise. Questa operazione offre un modo efficiente per monitorare e gestire i processi di importazione dei dati. Restituisce le seguenti informazioni chiave per ogni processo:

- Job ID. Un identificatore univoco per ogni processo di importazione in blocco
- Nome del lavoro. Il nome che hai assegnato al lavoro durante la sua creazione
- Stato attuale. Lo stato corrente del lavoro (ad esempio, COMPLETED, RUNNING, FAILED)

ListBulkImportJobs è particolarmente utile per ottenere una panoramica completa di tutti i lavori di importazione in blocco. Questo può aiutarti a tenere traccia di più importazioni di dati, identificare eventuali lavori che richiedono attenzione e mantenere un flusso di lavoro organizzato. L'operazione supporta l'impaginazione, che consente di recuperare in modo efficiente un gran numero di riepiloghi dei lavori. È possibile utilizzare il lavoro IDs restituito da questa operazione con l'[DescribeBulkImportJob](#) operazione per recuperare informazioni più dettagliate su lavori specifici. Questo processo in due fasi consente innanzitutto di ottenere una visione di alto livello di tutti i lavori e quindi di approfondire i dettagli dei lavori di interesse. Durante l'utilizzo ListBulkImportJobs, puoi applicare filtri per restringere i risultati. Ad esempio, è possibile filtrare i lavori in base al loro stato per recuperare solo i lavori completati o solo i lavori in esecuzione. Questa funzionalità consente di concentrarsi sulle informazioni più pertinenti per l'attività corrente. L'operazione restituisce anche un nextToken messaggio se ci sono più risultati disponibili. È possibile utilizzare questo token nelle chiamate successive per recuperare la serie successiva di riepiloghi dei lavori, in modo da scorrere ripetutamente tutti i processi di importazione in blocco anche se ne è presente un numero elevato. L'esempio seguente mostra come utilizzare ListBulkImportJobs with per recuperare un elenco AWS CLI di lavori completati.

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

Example Filtro Risposta per i lavori completati

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



```
}
```

Questo comando dimostra come `ListBulkImportJobs` recuperare un elenco di lavori completati con errori. Il massimo è impostato su 50 risultati e stiamo utilizzando un token successivo per i risultati impaginati.

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

Inserisci dati per AWS IoT SiteWise utilizzare le regole AWS IoT Core

Invia dati AWS IoT SiteWise a AWS IoT oggetti e altri AWS servizi utilizzando le regole in AWS IoT Core. Le regole trasformano i messaggi MQTT ed eseguono azioni per interagire con AWS i servizi. L'azione della AWS IoT SiteWise regola inoltra i dati dei messaggi all'[BatchPutAssetPropertyValue](#) operazione dall' AWS IoT SiteWise API. Per ulteriori informazioni, consulta [Regole](#) e [AWS IoT SiteWise azioni](#) nella Guida per gli AWS IoT sviluppatori.

Per seguire un tutorial che illustra i passaggi necessari per configurare una regola che acquisisce i dati attraverso le ombre dei dispositivi, consulta. [Inserisci dati AWS IoT SiteWise da oggetti AWS IoT](#)

Puoi anche inviare dati da altri AWS IoT SiteWise AWS servizi. Per ulteriori informazioni, consulta [Interagisci con altri AWS servizi](#).

Argomenti

- [Concedi AWS IoT l'accesso richiesto](#)
- [Configura l'azione della AWS IoT SiteWise regola](#)
- [Riduci i costi con Basic Ingest in AWS IoT SiteWise](#)

Concedi AWS IoT l'accesso richiesto

I ruoli IAM vengono utilizzati per controllare le AWS risorse a cui ogni regola ha accesso. Prima di creare una regola, devi creare un ruolo IAM con una policy che consenta alla regola di eseguire azioni sulla AWS risorsa richiesta. AWS IoT assume questo ruolo quando esegue una regola.

Se create l'azione della regola nella AWS IoT console, potete scegliere una risorsa principale per creare un ruolo che abbia accesso a una gerarchia di risorse selezionata. Per ulteriori informazioni

su come definire manualmente un ruolo per una regola, consulta [Concessione AWS IoT delle autorizzazioni di accesso richieste e Pass role](#) nella Developer Guide.AWS IoT

Per l'azione della AWS IoT SiteWise regola, è necessario definire un ruolo che consenta `iotsitewise:BatchPutAssetPropertyValue` l'accesso alle proprietà degli asset a cui la regola invia i dati. Per migliorare la sicurezza, è possibile specificare un percorso di gerarchia AWS IoT SiteWise degli asset nella Condition proprietà.

La policy di attendibilità di esempio riportata di seguito consente l'accesso a un asset specifico e ai relativi asset figlio.

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

Rimuovilo Condition dalla policy per consentire l'accesso a tutte le tue risorse. La policy di attendibilità di esempio riportata di seguito consente l'accesso a tutti gli asset nella regione corrente.

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

```
}
```

Configura l'azione della AWS IoT SiteWise regola

L'azione della AWS IoT SiteWise regola invia i dati dal messaggio MQTT che ha avviato la regola alle proprietà dell'asset in. AWS IoT SiteWise È possibile caricare più immissioni di dati in diverse proprietà degli asset contemporaneamente, per inviare aggiornamenti per tutti i sensori di un dispositivo in un unico messaggio. Inoltre puoi caricare più punti dati contemporaneamente per ogni inserimento di dati.

Note

Quando invii dati a AWS IoT SiteWise con l'azione della regola, i dati devono soddisfare tutti i requisiti dell'BatchPutAssetPropertyValueoperazione. Ad esempio, i dati non possono avere un timestamp precedente a 7 giorni dall'epoca Unix corrente. Per ulteriori informazioni, consulta [Inserimento di dati con l'API AWS IoT SiteWise](#).

Per ogni inserimento di dati nell'azione di regola, puoi identificare una proprietà dell'asset e specificare il timestamp, la qualità e il valore di ciascun punto dati per la proprietà dell'asset. L'azione di regola prevede stringhe per tutti i parametri.

Per identificare una proprietà dell'asset, specifica uno dei seguenti valori:

- Asset ID (ID asset) (`assetId`) e Property ID (ID proprietà) (`propertyId`) della proprietà dell'asset a cui invii i dati. Puoi trovare l'Asset ID e l'ID della proprietà utilizzando la Console AWS IoT SiteWise. Se conosci l'Asset ID, puoi utilizzare la chiamata AWS CLI `aws iot:DescribeAsset` per trovare l'ID della proprietà.
- Property alias (Alias proprietà) (`propertyAlias`), che è un alias del flusso di dati (ad esempio, `/company/windfarm/3/turbine/7/temperature`). Per utilizzare questa opzione, è necessario prima impostare l'alias della proprietà dell'asset. Per informazioni relative all'impostazione degli alias delle proprietà, consultare [Gestisci i flussi di dati per AWS IoT SiteWise](#).

Per il timestamp di ogni voce, utilizzate il timestamp riportato dall'apparecchiatura o il timestamp fornito da AWS IoT Core. Il timestamp ha due parametri:

- Tempo in secondi (`timeInSeconds`) — L'ora dell'epoca Unix, in secondi, in cui il sensore o l'apparecchiatura ha riportato i dati.

- Offset in nanos (offsetInNanos) — (Facoltativo) L'offset in nanosecondi dal tempo in secondi.

Important

Se il timestamp è una stringa, ha una parte decimale o non è espresso in secondi, rifiuta la richiesta. AWS IoT SiteWise È necessario convertire il timestamp in secondi e offset in nanosecondi. Utilizza le funzionalità del motore delle AWS IoT regole per convertire il timestamp. Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Ottenere timestamp per dispositivi che non riportano l'ora esatta](#)
- [Conversione di timestamp in formato stringa](#)

È possibile utilizzare modelli sostitutivi per diversi parametri dell'azione per eseguire calcoli, richiamare funzioni e recuperare valori dal payload dei messaggi. Per ulteriori informazioni, consulta Modelli [sostitutivi nella Guida per gli sviluppatori](#).AWS IoT

Note

Poiché un'espressione in un modello di sostituzione viene valutata separatamente dall'istruzione SELECT, non puoi utilizzare un modello di sostituzione per fare riferimento a un alias creato utilizzando una clausola AS. È possibile fare riferimento solo alle informazioni presenti nel payload originale, oltre alle funzioni e agli operatori supportati.

Argomenti

- [Ottenere timestamp per dispositivi che non riportano l'ora esatta](#)
- [Conversione di timestamp in formato stringa](#)
- [Conversione di stringhe di timestamp con precisione in nanosecondi](#)
- [Esempi di configurazioni di regole](#)
- [Risoluzione dei problemi dell'azione di regola](#)

Ottenere timestamp per dispositivi che non riportano l'ora esatta

[Se il sensore o l'apparecchiatura non riporta dati temporali accurati, recupera l'ora attuale di Unix dal motore delle AWS IoT regole con timestamp \(\)](#). Questa funzione restituisce il tempo in millisecondi,

quindi è necessario convertire il valore in tempo in secondi e l'offset in nanosecondi. A tale scopo, utilizzate le seguenti conversioni:

- Per Time in seconds (Ora in secondi) (`timeInSeconds`), utilizza **`${floor(timestamp() / 1E3)}`** per convertire l'ora da millisecondi in secondi.
- Per Offset in nanos (Offset in nanosecondi) (`offsetInNanos`), utilizza **`${(timestamp() % 1E3) * 1E6}`** per calcolare l'offset del timestamp in nanosecondi.

Conversione di timestamp in formato stringa

Se il sensore o l'apparecchiatura riporta i dati temporali in formato stringa (ad esempio, `2020-03-03T14:57:14.699Z`), usa [time_to_epoch](#) (String, String). Questa funzione inserisce il timestamp e lo schema di formato come parametri e restituisce l'ora in millisecondi. Quindi, è necessario convertire il tempo in tempo in secondi e l'offset in nanosecondi. A tale scopo, utilizzate le seguenti conversioni:

- Per Tempo in secondi (`timeInSeconds`), utilizzare **`${floor(time_to_epoch("2020-03-03T14:57:14.699Z", "yyyy-MM-dd'T'HH:mm:ss'Z'") / 1E3)}`** per convertire la stringa del timestamp in millisecondi e quindi in secondi.
- Per Offset in nanos (`offsetInNanos`), usa per calcolare l'offset in nanosecondi della stringa del **`${(time_to_epoch("2020-03-03T14:57:14.699Z", "yyyy-MM-dd'T'HH:mm:ss'Z'") % 1E3) * 1E6}`** timestamp.

Note

La `time_to_epoch` funzione supporta stringhe di timestamp con precisione fino a un millisecondo. Per convertire stringhe con precisione in microsecondi o nanosecondi, configura una AWS Lambda funzione richiamata dalla regola per convertire il timestamp in valori numerici. Per ulteriori informazioni, consulta [Conversione di stringhe di timestamp con precisione in nanosecondi](#).

Conversione di stringhe di timestamp con precisione in nanosecondi

Se il dispositivo invia informazioni sul timestamp in formato stringa con precisione in nanosecondi (ad esempio, `2020-03-03T14:57:14.699728491Z`) utilizza la procedura seguente per configurare

l'azione della regola. Potete creare una AWS Lambda funzione che converta il timestamp da una stringa in Time in seconds (**timeInSeconds**) e Offset in nanos (). `offsetInNanos` Quindi, usa [aws_lambda \(functionArn, inputJSON\)](#) nei parametri di azione della regola per richiamare quella funzione Lambda e utilizzare l'output nella regola.

Note

Questa sezione contiene istruzioni avanzate che presuppongono che si abbia familiarità con la creazione delle risorse seguenti:

- Funzioni Lambda. Per ulteriori informazioni, consulta [Creare la prima funzione Lambda nella Guida](#) per gli AWS Lambda sviluppatori.
- AWS IoT regole con la AWS IoT SiteWise rule action. Per ulteriori informazioni, consulta [Inserisci dati per AWS IoT SiteWise utilizzare le regole AWS IoT Core](#).

Per creare un'azione di AWS IoT SiteWise regola che analizzi le stringhe del timestamp

1. Crea una funzione Lambda con le seguenti proprietà:

- Nome della funzione: utilizza un nome di funzione descrittivo (ad esempio, **ConvertNanosecondTimestampFromString**).
- Runtime — Usa un runtime Python 3, come Python 3.11 (). `python3.11`
- Autorizzazioni: crea un ruolo con `AWS LambdaBasicExecutionRole` autorizzazioni Lambda di base ().
- Livelli: aggiungi il `AWS SDK Pandas livello -Python311` per l'utilizzo della funzione Lambda. `numpy`
- Codice funzione: utilizzate il seguente codice di funzione, che utilizza un argomento di stringa denominato `timestamp` e restituisce e valori per quel timestamp. `timeInSeconds` `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
    }
```

Questa funzione Lambda inserisce stringhe di timestamp in formato ISO 8601 utilizzando `datetime64` from NumPy

Note

Se le stringhe del timestamp non sono in formato ISO 8601, puoi implementare una soluzione con pandas che definisce il formato del timestamp. [Per ulteriori informazioni, vedi `pandas.to\_datetime`](#).

2. Quando configuri l' AWS IoT SiteWise azione per la tua regola, usa i seguenti modelli sostitutivi per Time in seconds () e Offset in nanos (). **timeInSeconds** `offsetInNanos` Questi modelli di sostituzione presuppongono che il payload del messaggio contenga la stringa timestamp in timestamp. La funzione `aws_lambda` utilizza una struttura JSON per il suo secondo parametro, quindi è possibile modificare i modelli di sostituzione sottostanti, se necessario.
 - Per Time in seconds (Tempo in secondi) (`timeInSeconds`), utilizzare il seguente modello di sostituzione.

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

- Per Offset in nanos (Offset in nanosecondi) (`offsetInNanos`), utilizzare il seguente modello di sostituzione.

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

Per ogni parametro, sostituisci *region* e *account-id* con la tua regione e l'ID dell'account. AWS Se hai usato un nome diverso per la tua funzione Lambda, cambia anche quello.

3. Concedi AWS IoT le autorizzazioni per richiamare la tua funzione con l'autorizzazione. `lambda:InvokeFunction` Per ulteriori informazioni, consultare [aws_lambda\(functionArn, inputJson\)](#).
4. Verifica la tua regola (ad esempio, usa il client di test AWS IoT MQTT) e verifica che AWS IoT SiteWise riceva i dati che invii.

Se la regola non funziona come previsto, consultare [Risolvere i problemi relativi a un' AWS IoT SiteWise azione delle regole](#).

Note

Questa soluzione richiama la funzione Lambda due volte per ogni stringa di timestamp. Puoi creare un'altra regola per ridurre il numero di chiamate alla funzione Lambda se la regola gestisce più punti dati con lo stesso timestamp in ogni payload.

A tale scopo, create una regola con un'azione di ripubblicazione che richiami la Lambda e pubblichi il payload originale con la stringa del timestamp convertita in `and. timeInSeconds offsetInNanos`. Quindi, create una regola con un'azione di regola per consumare il payload convertito AWS IoT SiteWise . Con questo approccio, riduci il numero di volte in cui la regola richiama la Lambda ma aumenti il numero di azioni della AWS IoT regola eseguite. Considerare il prezzo di ciascun servizio se si applica questa soluzione al caso d'uso.

Esempi di configurazioni di regole

Questa sezione contiene esempi di configurazioni di regole per creare una regola con un' AWS IoT SiteWise azione.

Example Azione di regola di esempio che utilizza alias di proprietà come argomenti dei messaggi

L'esempio seguente crea una regola con un' AWS IoT SiteWise azione che utilizza l'argomento (tramite [topic \(\)](#)) come alias di proprietà per identificare le proprietà degli asset. Utilizzate questo esempio per definire una regola per l'importazione di dati di doppio tipo in tutte le turbine eoliche di tutti i parchi eolici. Questo esempio richiede la definizione di alias di proprietà su tutte le proprietà

degli asset delle turbine. È necessario definire una seconda regola simile per importare dati di tipo intero.

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

Il payload di esempio in `sitewise-rule-payload.json` contiene quanto segue.

```
{  
  "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"  
      }  
    }  
  ]  
}
```

Con questa azione di regola, inviate il seguente messaggio a un alias di proprietà di una turbina eolica (ad esempio, `/company/windfarm/3/turbine/7/temperature`) come argomento per l'inserimento dei dati.

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

Example Azione di regola di esempio che utilizza `timestamp()` per determinare l'ora

L'esempio seguente crea una regola con un' AWS IoT SiteWise azione che identifica la proprietà di un asset IDs e utilizza [timestamp\(\)](#) per determinare l'ora corrente.

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

Il payload di esempio in `sitewise-rule-payload.json` contiene quanto segue.

```
{
  "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"
    }
}
]
```

Con questa azione della regola, inviate il seguente messaggio a `my/asset/property/topic` importare i dati.

```
{
  "type": "double",
  "value": "38.3"
}
```

Risoluzione dei problemi dell'azione di regola

Per risolvere l'azione della AWS IoT SiteWise regola in AWS IoT Core, configura i CloudWatch registri o configura un'azione di errore di ripubblicazione per la regola. Per ulteriori informazioni, consulta [Risolvere i problemi relativi a un' AWS IoT SiteWise azione delle regole](#).

Riduci i costi con Basic Ingest in AWS IoT SiteWise

AWS IoT Core [offre una funzionalità chiamata Basic Ingest che è possibile utilizzare per inviare dati AWS IoT Core senza incorrere in AWS IoT costi di messaggistica](#). Basic Ingest ottimizza il flusso di dati per i carichi di lavoro di inserimento con elevati volumi di dati rimuovendo il broker dei messaggi di pubblicazione/sottoscrizione dal percorso di inserimento. Puoi utilizzare Basic Ingest se conosci le regole a cui devono essere instradati i messaggi.

Per utilizzare Basic Ingest, invii i messaggi direttamente a una regola specifica utilizzando l'argomento speciale `$aws/rules/rule-name`. Ad esempio, per inviare un messaggio a una regola denominata `SiteWiseWindFarmRule`, invii un messaggio all'argomento `$aws/rules/SiteWiseWindFarmRule`.

Se l'azione di regola utilizza modelli di sostituzione contenenti [topic\(Decimal\)](#), puoi passare l'argomento originale alla fine dell'argomento speciale di Basic Ingest, ad esempio `$aws/rules/rule-name/original-topic`. Ad esempio, per utilizzare Basic Ingest con l'alias di proprietà della centrale eolica di esempio della sezione precedente, puoi inviare messaggi al seguente argomento.

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

 Note

L'esempio precedente include una seconda barra (//) perché AWS IoT rimuove il prefisso Basic Ingest (\$aws/rules/*rule-name*/) dall'argomento visibile all'azione della regola. In questo esempio, la regola riceve l'argomento /company/windfarm/3/turbine/7/temperature.

Per ulteriori informazioni, consulta [Ridurre i costi di messaggistica con Basic Ingest](#) nella Guida per gli sviluppatori AWS IoT.

Inserisci dati da AWS IoT SiteWise AWS IoT Events

Con AWS IoT Events, puoi creare applicazioni complesse di monitoraggio degli eventi per la tua flotta IoT nel AWS Cloud. Utilizza l'azione IoT in AWS IoT Events per inviare dati alle proprietà degli asset AWS IoT SiteWise quando si verifica un evento.

 Note

Avviso di fine del supporto: il 20 maggio 2026, AWS terminerà il supporto per AWS IoT Events. Dopo il 20 maggio 2026, non potrai più accedere alla AWS IoT Events console o AWS IoT Events alle risorse. Per ulteriori informazioni, consulta [AWS IoT Events Fine del supporto](#).

AWS IoT Events è progettato per semplificare lo sviluppo di applicazioni di monitoraggio degli eventi per dispositivi e sistemi IoT all'interno del AWS Cloud. Utilizzando AWS IoT Events, puoi:

- Rileva e rispondi a cambiamenti, anomalie o condizioni specifiche nella tua flotta IoT.
- Migliora l'efficienza operativa e abilita la gestione proattiva del tuo ecosistema IoT.

Grazie all'integrazione con AWS IoT SiteWise through the AWS IoT SiteWise action, ne AWS IoT Events estende le funzionalità, consentendoti di aggiornare automaticamente le proprietà degli asset AWS IoT SiteWise in risposta a eventi specifici. Questa interazione può semplificare l'inserimento e la gestione dei dati. Può anche fornirti informazioni utili.

Per ulteriori informazioni, consulta i seguenti argomenti nella Guida per gli AWS IoT Events sviluppatori:

- [Che cos'è AWS IoT Events?](#)
- [Operazioni AWS IoT Events](#)
- [SiteWise Azione IoT](#)

Usa AWS IoT Greengrass stream manager in AWS IoT SiteWise

AWS IoT Greengrass stream manager è una funzionalità di integrazione che facilita il trasferimento di flussi di dati da fonti locali al AWS cloud. Funge da livello intermedio che gestisce i flussi di dati, consentendo ai dispositivi che operano all'edge di raccogliere e archiviare i dati prima che vengano inviati AWS IoT SiteWise, per ulteriori analisi ed elaborazioni.

Aggiungi una destinazione di dati configurando una fonte locale sulla AWS IoT SiteWise console. Puoi anche utilizzare stream manager nella tua AWS IoT Greengrass soluzione personalizzata per importare dati. AWS IoT SiteWise

Note

Per importare dati da fonti OPC UA, configura un gateway AWS IoT SiteWise Edge che funzioni su. AWS IoT Greengrass Per ulteriori informazioni, consulta [Usa i gateway AWS IoT SiteWise Edge](#).

Per ulteriori informazioni su come configurare una destinazione per i dati di origine locale, consulta. [Comprendi le destinazioni Edge AWS IoT SiteWise](#)

Per ulteriori informazioni su come importare dati utilizzando stream manager in una AWS IoT Greengrass soluzione personalizzata, consulta i seguenti argomenti nella Guida per gli AWS IoT Greengrass Version 2 sviluppatori:

- [Che cos'è AWS IoT Greengrass?](#)
- [Gestisci i flussi di dati sul core AWS IoT Greengrass](#)
- [Esportazione dei dati nelle proprietà degli asset AWS IoT SiteWise](#)

Usa i gateway AWS IoT SiteWise Edge

AWS IoT SiteWise Edge estende le funzionalità del cloud agli ambienti periferici industriali, abilitando l'elaborazione, l'analisi e il processo decisionale locali dei dati. SiteWise Edge si integra con AWS IoT SiteWise e altri AWS servizi per fornire soluzioni IoT industriali complete. I gateway fungono da intermediario tra le apparecchiature industriali e AWS IoT SiteWise.

SiteWise e i gateway Edge funzionano su due diversi obiettivi di implementazione:

- AWS IoT Greengrass V2
- Siemens Industrial Edge

È possibile utilizzare un gateway SiteWise Edge per raccogliere dati sull'edge e pubblicarli sul cloud. Per i gateway in esecuzione AWS IoT Greengrass, puoi anche elaborare i dati all'edge utilizzando modelli e asset di asset.

L'applicazione AWS IoT SiteWise Edge on Siemens Industrial Edge supporta l'integrazione tra apparecchiature industriali e AWS IoT SiteWise consente di aggregare ed elaborare i dati grezzi delle macchine ed eseguire analisi localmente prima di inviare dati raffinati al Cloud. AWS

Concetti chiave dei SiteWise gateway Edge

SiteWise Edge offre diverse funzioni utili per l'edge computing in ambienti industriali.

Raccolta ed elaborazione locali dei dati

Supporta la raccolta di dati da asset industriali utilizzando protocolli come OPC-UA e MQTT. I gateway funzionano su AWS IoT Greengrass dispositivi Core o. Siemens Industrial Edge

Funzionamento offline

Continua a raccogliere ed elaborare dati durante le interruzioni di Internet, sincronizzandosi con il cloud quando la connettività viene ripristinata.

Edge computing con componenti AWS IoT Greengrass

Utilizza l'editor SiteWise IoT per inoltrare i dati al cloud e al AWS IoT SiteWise processore per trasformazioni e calcoli locali. Sia l'editor che il processore sono AWS IoT Greengrass

V2 componenti. Per ulteriori informazioni sui AWS IoT Greengrass componenti, vedere [AWS-provided components](#).

Integrazione con AWS IoT SiteWise per estendere le funzionalità cloud

Funziona con le funzionalità AWS IoT SiteWise cloud, estendendo modelli di asset, analisi e dashboard fino all'edge.

Per i gateway con un pacchetto di elaborazione dati abilitato, puoi utilizzare AWS OpsHub for per gestire centralmente AWS IoT SiteWise i tuoi SiteWise gateway Edge. AWS OpsHub fornisce funzionalità di gestione e monitoraggio remoti. Per ulteriori informazioni, consulta [Gestisci i gateway SiteWise Edge utilizzando per AWS OpsHubAWS IoT SiteWise](#).

Integrazione delle fonti di dati dei partner

Connetti una fonte di dati partner al tuo gateway e ricevi dati dal partner nel tuo gateway SiteWise Edge e nel AWS cloud. Per ulteriori informazioni, consulta [Fonti di dati dei partner sui SiteWise gateway Edge](#).

Visualizzazione locale sull'edge

Fornisce dashboard personalizzati per informazioni in tempo reale sull'edge.

Monitora i dati localmente nella tua struttura utilizzando i portali SiteWise Monitor sui tuoi dispositivi locali. Per ulteriori informazioni, consulta [Attivazione del AWS IoT SiteWise portale all'edge](#).

Vantaggi dell'implementazione SiteWise di Edge

SiteWise Edge offre numerosi vantaggi che possono migliorare in modo significativo le operazioni industriali e i processi decisionali.

- Informazioni operative in tempo reale senza ritardi nell'elaborazione del cloud
- Continuità operativa in ambienti disconnessi
- Larghezza di banda e costi di archiviazione ridotti grazie alla preelaborazione perimetrale
- Maggiore affidabilità con la capacità di prendere decisioni locali basate sui dati

Ospita autonomamente un gateway AWS IoT SiteWise Edge con AWS IoT Greengrass V2

Configura AWS IoT SiteWise Edge per raccogliere, elaborare e visualizzare localmente i dati dalle apparecchiature industriali prima di inviarli al cloud. Utilizzo dell'hosting autonomo. AWS IoT Greengrass Version 2

Un gateway AWS IoT SiteWise Edge funge da intermediario tra le apparecchiature industriali e AWS IoT SiteWise. In esecuzione AWS IoT Greengrass Version 2, il gateway SiteWise Edge supporta la raccolta e l'elaborazione dei dati in locale. Monitora i dati localmente all'interno della struttura tramite i portali SiteWise Monitor sui dispositivi locali con il pacchetto di elaborazione dati abilitato e AWS OpsHub installato.

Esistono due tipi di gateway ospitati autonomamente:

Gateway V3 abilitato per MQTT

L'architettura gateway V3 abilitata per MQTT offre migliori capacità di acquisizione dei dati. Utilizza il protocollo MQTT per una comunicazione dati efficiente e offre destinazioni di dati configurabili. Queste includono opzioni per l'inserimento di dati nel buffer tramite Amazon S3, nonché l'ingestione di dati in tempo reale. È possibile implementare filtri di percorso per sottoscrivere argomenti MQTT specifici, abilitando la raccolta mirata dei dati. Si noti che il gateway V3 abilitato per MQTT non supporta la funzionalità Data Processing Pack. Per ulteriori informazioni, consulta [Gateway V3 compatibili con MQTT per Edge AWS IoT SiteWise](#).

Stream classici, gateway V2

Il gateway Classic Streams, V2 rappresenta l'architettura gateway Edge tradizionale AWS IoT SiteWise. È ideale per le implementazioni SiteWise Edge esistenti e per gli utenti abituati al flusso di lavoro consolidato. Sebbene il gateway Classic Streams, V2 supporti il pacchetto di elaborazione dati, tieni presente che i dati generati dal pacchetto di elaborazione dati non possono essere importati tramite Amazon S3. Usa il gateway Classic Streams, V2 se devi mantenere la compatibilità con le distribuzioni esistenti o se hai bisogno della funzionalità del pacchetto di elaborazione dati. Per ulteriori informazioni, consulta [Stream classici, gateway V2 per Edge AWS IoT SiteWise](#).

Argomenti

- [AWS IoT SiteWise Requisiti del gateway Edge self-hosted](#)

- [Crea un gateway Edge ospitato autonomamente SiteWise](#)
- [Installa il software AWS IoT SiteWise Edge gateway sul tuo dispositivo locale](#)
- [Gateway V3 compatibili con MQTT per Edge AWS IoT SiteWise](#)
- [Stream classici, gateway V2 per Edge AWS IoT SiteWise](#)
- [Aggiungi fonti di dati al tuo gateway AWS IoT SiteWise Edge](#)
- [AWS IoT Greengrass componenti per AWS IoT SiteWise Edge](#)
- [Filtra le risorse su un gateway SiteWise Edge](#)
- [Configura il supporto proxy e gestisci gli archivi affidabili per AWS IoT SiteWise Edge](#)
- [Utilizzare AWS IoT SiteWise APIs sul bordo](#)

AWS IoT SiteWise Requisiti del gateway Edge self-hosted

AWS IoT SiteWise I gateway edge funzionano AWS IoT Greengrass V2 come un insieme di AWS IoT Greengrass componenti che supportano la raccolta, l'elaborazione e la pubblicazione dei dati in locale. Per configurare un gateway SiteWise Edge funzionante AWS IoT Greengrass V2, crea un gateway in Cloud AWS ed esegui il software SiteWise Edge gateway per configurare il dispositivo locale. Quando si utilizza il AWS Management Console per creare il gateway SiteWise Edge, viene fornito uno script di installazione. Esegui questo script sul dispositivo gateway di destinazione per configurare il software e le dipendenze necessari.

Requisiti locali del dispositivo

I dispositivi locali devono soddisfare i seguenti requisiti per installare ed eseguire il software SiteWise Edge gateway.

- Supporta la versione del software AWS IoT Greengrass V2 Core [v2.3.0 o successiva](#). Per ulteriori informazioni, consulta [Requisiti nella Guida](#) per gli AWS IoT Greengrass Version 2 sviluppatori.
- Una delle seguenti piattaforme supportate:
 - Sistema operativo: Ubuntu 20.04 o versione successiva
Architettura: x86_64 (AMD64) o (Aarch64) ARMv8
 - Sistema operativo: Red Hat Enterprise Linux (RHEL) 8
Architettura: x86_64 (AMD64) o (Aarch64) ARMv8
 - Sistema operativo: Amazon Linux 2

Architettura: x86_64 (AMD64) o ARMv8 (Aarch64)

- Sistema operativo: Debian 11

Architettura: x86_64 (AMD64) o (Aarch64) ARMv8

- Sistema operativo: Windows Server 2019 e versioni successive

Architettura: x86_64 () AMD64

Note

Le piattaforme ARM supportano solo i gateway SiteWise Edge con Data Collection Pack. Il pacchetto di elaborazione dati non è supportato.

- Minimo 4 GB di RAM.
- Almeno 10 GB di spazio su disco disponibili per il software SiteWise Edge gateway.
- Configura il dispositivo locale per assicurarti che le porte corrette siano accessibili. Per un elenco completo degli endpoint di servizio in uscita richiedi, consulta Endpoint di [servizio richiesti per AWS IoT SiteWise i gateway](#) Edge.
- Java Runtime Environment (JRE) versione 11 o successiva. Java deve essere disponibile nella variabile di PATH ambiente del dispositivo. Per utilizzare Java per sviluppare componenti personalizzati, è necessario installare un Java Development Kit (JDK). Ti consigliamo di utilizzare [Amazon Corretto](#) o [OpenJDK](#).

Bucket Amazon S3 da inserire nell'elenco dei dispositivi locali

Configura il tuo dispositivo locale per fornire l'accesso al firewall al seguente bucket Amazon S3. Configura l'accesso in base alle rispettive regioni dei tuoi dispositivi.

Regione	Endpoint
Asia Pacifico (Tokyo)	https://iot-sitewise-gateway-ap-northeast-1-785558802005.s3.ap-northeast-1.amazonaws.com
Asia Pacifico (Seoul)	https://iot-sitewise-gateway-ap-northeast-2-310055672453.s3.ap-northeast-2.amazonaws.com

Regione	Endpoint
Asia Pacifico (Mumbai)	https://iot-sitewise-gateway-ap-south-1-677656657204.s3.ap-south-1.amazonaws.com
Asia Pacifico (Singapore)	https://iot-sitewise-gateway-ap-southeast-1-475191558554.s3.ap-southeast-1.amazonaws.com
Asia Pacifico (Sydney)	https://iot-sitewise-gateway-ap-southeast-2-396319432685.s3.ap-southeast-2.amazonaws.com
Canada (Centrale)	https://iot-sitewise-gateway-ca-central-1-842060018567.s3.ca-central-1.amazonaws.com
Cina (Pechino)	https://iot-sitewise-gateway-cn-north-1-237124890262.s3.cn-north-1.amazonaws.com.cn
Europa (Francoforte)	https://iot-sitewise-gateway-eu-central-1-748875242063.s3.eu-central-1.amazonaws.com
Europa (Irlanda)	https://iot-sitewise-gateway-eu-west-1-383414315062.s3.eu-west-1.amazonaws.com
Stati Uniti orientali (Virginia settentrionale)	https://iot-sitewise-gateway-us-east-1-223558168232.s3.us-east-1.amazonaws.com / https://iot-sitewise-gateway-us-east-1-223558168232.s3.amazonaws.com
Stati Uniti orientali (Ohio)	https://iot-sitewise-gateway-us-east-2-005072661813.s3.us-east-2.amazonaws.com
AWS GovCloud (Stati Uniti occidentali)	https://iot-sitewise-gateway-us-gov-west-1-599984565679.s3.us-gov-west-1.amazonaws.com/
US West (Oregon)	https://iot-sitewise-gateway-us-west-2-502577205460.s3.us-west-2.amazonaws.com

Requisiti del pacchetto di elaborazione dati

- Se si prevede di utilizzare il pacchetto di elaborazione dati all'edge con AWS IoT SiteWise, il dispositivo locale deve inoltre soddisfare i seguenti requisiti:
 - Dispone di un processore quad-core x86 a 64 bit.
 - Dispone di almeno 16 GB di RAM.
 - Se utilizzato, dispone di almeno 32 GB di RAM Microsoft Windows.
 - Aveva almeno 256 GB di spazio libero su disco.
 - Il dispositivo locale deve consentire il traffico di rete in entrata sulla porta 443.
 - Le seguenti porte sono riservate all'uso da AWS IoT SiteWise: 80, 443, 3001, 4569, 4572, 8000, 8081, 8082, 8084, 8085, 8445, 8086, 9000, 9500, 11080 e 50010. L'utilizzo di una porta riservata per il traffico può comportare l'interruzione della connessione.

Note

Il componente AWS IoT Greengrass V2 Stream manager ha i propri requisiti. Per ulteriori informazioni, consulta [Configuration](#) nella AWS IoT Greengrass Version 2 Developer Guide.

- I requisiti minimi di spazio su disco e capacità di elaborazione dipendono da una serie di fattori specifici dell'implementazione e del caso d'uso.
- Lo spazio su disco necessario per memorizzare nella cache i dati relativi alla connettività Internet intermittente dipende dai seguenti fattori:
 - Numero di flussi di dati caricati
 - Punti dati per flusso di dati al secondo
 - Dimensione di ciascun punto dati
 - Velocità di comunicazione
 - Tempo previsto di inattività della rete
- La capacità di elaborazione necessaria per il polling e il caricamento dei dati dipende dai seguenti fattori:
 - Numero di flussi di dati caricati
 - Punti dati per flusso di dati al secondo

Configura le autorizzazioni per utilizzare SiteWise i gateway Edge

È necessario disporre delle seguenti autorizzazioni per utilizzare SiteWise i gateway Edge:

Note

Se utilizzi la AWS IoT SiteWise console per creare il tuo gateway SiteWise Edge, queste autorizzazioni vengono aggiunte automaticamente.

- Il ruolo IAM per il tuo gateway SiteWise Edge deve consentirti di utilizzare un gateway SiteWise Edge su un AWS IoT Greengrass V2 dispositivo per elaborare i dati del modello di asset e i dati degli asset.

Il ruolo consente al seguente servizio di assumere il ruolo: `credentials.iot.amazonaws.com`.

Dettagli dell'autorizzazione

Il ruolo deve avere le seguenti autorizzazioni:

- `iotsitewise`— Consente ai responsabili di recuperare i dati dei modelli di asset e i dati degli asset dall'edge.
- `iot`— Consente ai AWS IoT Greengrass V2 dispositivi di interagire con. AWS IoT
- `logs`— Consente ai AWS IoT Greengrass V2 dispositivi di inviare log ad Amazon CloudWatch Logs.
- `s3`— Consente ai AWS IoT Greengrass V2 dispositivi di scaricare elementi di componenti personalizzati da Amazon S3.

```
{
  "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": "*"
}
```

Crea un gateway Edge ospitato autonomamente SiteWise

Usa la AWS IoT SiteWise console o AWS CLI per creare un gateway SiteWise Edge ospitato autonomamente. Questa procedura descrive in dettaglio come creare un gateway SiteWise Edge ospitato autonomamente da installare sul proprio hardware. Per informazioni sulla creazione di un gateway SiteWise Edge che funziona su Siemens Industrial Edge, consulta. [Ospita un gateway SiteWise Edge su Siemens Industrial Edge](#)

Creare un gateway SiteWise Edge

Console

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Edge gateway.
3. Selezionare Create gateway (Crea gateway).
4. Per Scegli l'obiettivo di implementazione, scegli un gateway ospitato autonomamente.
5. Seleziona un gateway V3 abilitato per MQTT o un gateway V2 Classic Streams. Per ulteriori informazioni su ciascuna opzione, vedere. [Ospita autonomamente un gateway AWS IoT](#)

[SiteWise Edge con AWS IoT Greengrass V2](#) Il gateway V3 abilitato per MQTT è consigliato per le sue funzionalità pronte per il futuro.

6. Nella sezione Configurazione del gateway, inserisci un nome per il tuo gateway SiteWise Edge o usa il nome generato da AWS IoT SiteWise
7. In Greengrass Device OS, seleziona il sistema operativo del dispositivo su cui installerai questo gateway SiteWise Edge.

 Note

Il pacchetto di elaborazione dati è disponibile solo su piattaforme x86. È disponibile solo sui flussi Classic, gateway V2

8. (Facoltativo) Per elaborare e organizzare i dati sull'edge, in Funzionalità Edge, seleziona Data Processing Pack.

 Note

Per concedere ai gruppi di utenti presenti nella directory aziendale l'accesso a questo gateway SiteWise Edge, consulta [Configura la funzionalità edge in SiteWise Edge](#)

9. (Facoltativo) In configurazione avanzata, effettuate le seguenti operazioni:
 - Per il dispositivo principale Greengrass, scegli una delle seguenti opzioni:
 - Configurazione predefinita: utilizza AWS automaticamente le impostazioni predefinite per creare un dispositivo principale Greengrass in AWS IoT Greengrass V2
 1. Immettere un nome per il dispositivo principale Greengrass o utilizzare il nome generato da AWS IoT SiteWise
 - Configurazione avanzata: scegli questa opzione se desideri utilizzare un dispositivo principale Greengrass esistente o crearne uno manualmente.
 1. Scegli un dispositivo principale Greengrass o scegli Crea dispositivo principale Greengrass per crearne uno nella console AWS IoT Greengrass V2. Per ulteriori informazioni, consulta [Configurazione di altri dispositivi AWS IoT Greengrass V2](#) nella Guida per sviluppatori AWS IoT Greengrass Version 2.
10. Selezionare Create gateway (Crea gateway).

11. Nella finestra di dialogo del programma di installazione del gateway Generate SiteWise Edge, scegli Genera e scarica. AWS IoT SiteWise genera automaticamente un programma di installazione che è possibile utilizzare per configurare il dispositivo locale.

 Important

Non puoi rigenerare questo file. Assicurati di salvare il file di installazione in una posizione sicura perché utilizzerai il file in un secondo momento.

AWS CLI

Per creare un gateway self-hosted utilizzando il AWS CLI, fornisci un nome per il gateway, specifica la piattaforma e la versione del gateway. Esistono molte altre opzioni che è possibile specificare durante la creazione di un gateway. Per ulteriori informazioni, vedete [create-gateway](#) nel AWS CLI Command Reference for AWS IoT SiteWise

Per utilizzare questo esempio, sostituisci i segnaposto inseriti dall'utente con le tue informazioni.

```
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`— Un nome univoco per il gateway.
- `gateway-platform`— specifica la configurazione della piattaforma gateway. Per i gateway ospitati autonomamente, immettere. `greengrassV2` Per ulteriori informazioni, consulta [Opzioni](#) nella sezione `create-gateway` di Command Reference for AWS CLI AWS IoT SiteWise
- `coreDeviceThingName`— Il nome dell'oggetto AWS IoT per il tuo dispositivo AWS IoT Greengrass V2 principale.
- `coreDeviceOperatingSystem`— Il sistema operativo del dispositivo principale in AWS IoT Greengrass V2. La specificazione del sistema operativo è obbligatoria `gateway-version 3` e non applicabile per `gateway-version 2`. Le opzioni includono: `LINUX_AARCH64`, `LINUX_AMD64`, e `WINDOWS_AMD64`
- `gateway-version`— La versione del gateway.

- Per creare un gateway V3 abilitato per MQTT, utilizzare 3 per la versione gateway.
- Per creare un gateway Classic Streams, V2, utilizzatelo per la versione gateway. 2
- `cli-input-json`— Un file JSON contenente i parametri della richiesta.

Ora che hai creato il gateway SiteWise Edge, [installa il software AWS IoT SiteWise Edge gateway sul tuo dispositivo locale](#).

Installa il software AWS IoT SiteWise Edge gateway sul tuo dispositivo locale

Dopo aver creato un gateway AWS IoT SiteWise Edge, installa il software SiteWise Edge gateway sul tuo dispositivo locale. SiteWise Il software Edge Gateway può essere installato su dispositivi locali su cui sono installati sistemi operativi Linux o Microsoft Windows server.

Important

Assicurati che il dispositivo locale si connetta a Internet.

Linux

La procedura seguente utilizza SSH per connettersi al dispositivo locale. In alternativa, è possibile utilizzare un'unità flash USB o altri strumenti per trasferire il file di installazione sul dispositivo locale. Se non desideri utilizzare SSH, vai al passaggio 2: installa il software gateway SiteWise Edge riportato di seguito.

Prerequisiti SSH

Prima di connetterti al dispositivo tramite SSH, completa i seguenti prerequisiti.

- Linux e macOS: scarica e installa OpenSSH. Per ulteriori informazioni, consulta <https://www.openssh.com>.

Passaggio 1: copia il programma di installazione sul tuo dispositivo gateway Edge SiteWise

Le seguenti istruzioni spiegano come connettersi al dispositivo locale utilizzando un client SSH.

1. Per connetterti al tuo dispositivo, esegui il seguente comando in una finestra di terminale sul tuo computer, sostituendo *username* e *IP* con un nome utente con privilegi e indirizzo IP elevati.

```
ssh username@IP
```

2. Per trasferire il file di installazione AWS IoT SiteWise generato sul dispositivo gateway SiteWise Edge, esegui il comando seguente.

 Note

- Sostituiscilo *path-to-saved-installer* con il percorso sul computer utilizzato per salvare il file di installazione e il nome del file di installazione.
- Sostituiscilo *IP-address* con l'indirizzo IP del dispositivo locale.
- Sostituiscilo *directory-to-receive-installer* con il percorso sul dispositivo locale che usi per ricevere il file di installazione.

```
scp path-to-saved-installer.sh user-name@IP-address:directory-to-receive-installer
```

Passaggio 2: installare il software SiteWise Edge gateway

Nelle seguenti procedure, esegui i comandi in una finestra di terminale sul tuo dispositivo gateway SiteWise Edge.

1. Assegna al file di installazione l'autorizzazione di esecuzione.

```
chmod +x path-to-installer.sh
```

2. Eseguire il programma di installazione.

```
sudo ./path-to-installer.sh
```

Windows Server

Prerequisiti

È necessario disporre dei seguenti prerequisiti per installare il software SiteWise Edge gateway:

- Microsoft WindowsServer 2019 o versione successiva installata
- Privilegi di amministratore
- PowerShell versione 5.1 o successiva installata
- SiteWise Il programma di installazione di Edge Gateway è stato scaricato su Windows Server, dove verrà eseguito il provisioning

Passaggio 1: Esegui PowerShell come amministratore

1. Sul server Windows in cui desideri installare SiteWise Edge gateway, accedi come amministratore.
2. Entra PowerShell nella barra di ricerca di Windows.
3. Nei risultati della ricerca, apri il menu contestuale (fai clic con il pulsante destro del mouse) sull' PowerShell app Windows. Scegli Esegui come amministratore.

Passaggio 2: installa il software SiteWise Edge gateway

Esegui i seguenti comandi in una finestra di terminale sul tuo dispositivo SiteWise Edge Gateway.

1. Sblocca il programma di installazione del gateway SiteWise Edge.

```
unblock-file path-to-installer.ps1
```

2. Esegui il programma di installazione.

```
./path-to-installer.ps1
```

Note

Se l'esecuzione dello script è disabilitata sul sistema, modifica la politica di esecuzione dello script in RemoteSigned.

```
Set-ExecutionPolicy RemoteSigned
```

Il passaggio successivo dipende dal tipo di gateway self-hosted di cui hai bisogno. Continua su [Gateway V3 compatibili con MQTT per Edge AWS IoT SiteWise](#) o. [Stream classici, gateway V2 per Edge AWS IoT SiteWise](#)

Gateway V3 compatibili con MQTT per Edge AWS IoT SiteWise

AWS IoT SiteWise può utilizzare gateway V3 abilitati per MQTT, che rappresentano un progresso significativo nell'architettura del gateway Edge. SiteWise Questo tipo di gateway sfrutta il protocollo MQTT (Message Queuing Telemetry Transport) per la comunicazione dei dati, offrendo maggiore flessibilità ed efficienza nelle implementazioni IoT industriali.

Il gateway V3 abilitato per MQTT utilizza MQTT per il trasferimento dei dati, abilitando un protocollo di rete leggero e di pubblicazione e sottoscrizione che trasporta in modo efficiente i messaggi tra i dispositivi e il cloud. Puoi configurare varie destinazioni di dati, tra cui l'inserimento di dati in tempo reale direttamente in Amazon S3 AWS IoT SiteWise e l'inserimento di dati nel buffer. Per consentire una raccolta precisa dei dati, puoi implementare filtri di percorso per sottoscrivere argomenti MQTT specifici.

I gateway V3 compatibili con MQTT sono dotati di una destinazione preconfigurata in tempo reale con filtri impostati su «#» (tutti gli argomenti), che è possibile personalizzare o rimuovere in base alle esigenze. Per semplificare la gestione dei dati, in ogni gateway può esistere una sola destinazione in tempo reale.

L'architettura abilitata per MQTT si differenzia notevolmente dal gateway Classic Streams, V2. Mentre la V2 utilizza un approccio basato sul flusso, la V3 utilizza MQTT, offrendo destinazioni di dati e opzioni di filtraggio più configurabili. Tuttavia, tieni presente che V3 non supporta il pacchetto di elaborazione dati, disponibile nella V2.

Il gateway V3 abilitato per MQTT offre diversi vantaggi:

- Scalabilità migliorata, grazie alla natura leggera di MQTT, che consente una migliore gestione di numerosi dispositivi e la trasmissione di dati ad alta frequenza.
- Controllo migliorato dei dati tramite filtri di percorso, che consentono una gestione granulare della raccolta dei dati e riducono il trasferimento e l'elaborazione non necessari dei dati.
- Gestione flessibile dei dati, che consente la configurazione tra elaborazione in tempo reale e archiviazione bufferizzata in base a esigenze specifiche.
- Allineamento con i moderni standard di comunicazione IoT, ponendo le basi per futuri miglioramenti e integrazioni.

Prendi in considerazione l'adozione del gateway V3 dotato di MQTT per le nuove implementazioni, soprattutto quando sono necessarie opzioni flessibili di inserimento dei dati e un controllo preciso sulla raccolta dei dati.

Note

Per le implementazioni o gli scenari esistenti che richiedono il pacchetto di elaborazione dati, il gateway Classic Streams, V2 rimane un'opzione valida.

Offrendo entrambi i tipi di gateway, ti AWS IoT SiteWise assicura di poter scegliere la soluzione più adatta alle tue esigenze IoT industriali specifiche, sia che tu dia priorità alle funzionalità MQTT avanzate o alla compatibilità con i sistemi esistenti.

Destinazioni e filtri di percorso

Visualizza i seguenti argomenti per ulteriori informazioni sulle destinazioni e sui filtri di percorso nei gateway abilitati per MQTT:

- [Comprendi le destinazioni Edge AWS IoT SiteWise](#)
- [Aggiungi una AWS IoT SiteWise destinazione Edge in tempo reale](#)
- [Aggiungi una destinazione AWS IoT SiteWise con buffer utilizzando Amazon S3](#)
- [Comprendi i filtri di percorso per AWS IoT SiteWise le destinazioni Edge](#)
- [Aggiungi filtri di percorso alle destinazioni AWS IoT SiteWise Edge](#)
- [Gestisci le destinazioni AWS IoT SiteWise Edge](#)

Connect applicazioni esterne al broker EMQX

Questa guida spiega come connettere applicazioni esterne al gateway AWS IoT SiteWise Edge tramite un broker EMQX sul gateway V3 implementato e abilitato per MQTT. Le applicazioni esterne possono includere strumenti di monitoraggio personalizzati, software di visualizzazione di terze parti o sistemi legacy che devono interagire con i dati industriali all'edge.

Tratteremo i passaggi di configurazione sia per Linux che per gli Microsoft Windows ambienti, tra cui la configurazione di distribuzione EMQX, la configurazione TLS per connessioni sicure e le regole di autorizzazione per controllare l'accesso ad argomenti specifici.

 Note

EMQX non è un fornitore o fornitore di Edge. AWS IoT SiteWise

 Important

Per proteggere le connessioni al gateway, consigliamo vivamente di utilizzare l'autenticazione basata su certificati tramite la funzionalità di autenticazione del dispositivo client. AWS IoT Greengrass Questo metodo offre una solida sicurezza tramite l'autenticazione TLS reciproca (mTLS). Per ulteriori informazioni, consulta [Connect client a dispositivi core](#) nella AWS IoT Greengrass Version 2 Developer Guide.

Se non sei in grado di utilizzare l'autenticazione basata su certificati, segui questa guida per configurare l'autenticazione tramite nomi utente e password.

Prerequisiti

- Un gateway V3 abilitato per SiteWise Edge MQTT che è stato distribuito ed è online
- Accesso all'host del gateway
- Accesso alle AWS IoT Greengrass console AWS IoT SiteWise e

Argomenti

- [Formato di payload dei messaggi per il broker EMQX su Edge AWS IoT SiteWise](#)
- [Configurare il broker EMQX](#)
- [Connect un'applicazione al broker EMQX su Edge AWS IoT SiteWise](#)
- [Imposta le regole di autorizzazione per AWS IoT SiteWise Edge in EMQX](#)

Formato di payload dei messaggi per il broker EMQX su Edge AWS IoT SiteWise

Affinché il componente SiteWise editore IoT utilizzi i dati dall'applicazione esterna e li pubblichi sul AWS IoT SiteWise cloud, il payload inviato al broker deve soddisfare requisiti specifici.

La comprensione del formato del payload è fondamentale per una comunicazione MQTT efficace con Edge. AWS IoT SiteWise Sebbene il processo di configurazione della connessione sia trattato

nelle sezioni successive, presentiamo innanzitutto i requisiti del payload per aiutarvi a pianificare l'implementazione.

Requisiti dell'argomento MQTT

Non ci sono restrizioni sulla struttura degli argomenti MQTT, incluso il numero di livelli o caratteri utilizzati. Tuttavia, è consigliabile che l'argomento corrisponda al `propertyAlias` campo del payload.

Example Esempio di alias di proprietà

Se l'argomento MQTT è `site1/line1/compressor1/temperature`, assicurati che le `propertyAlias` corrispondano.

```
{
  "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
      }
    }
  ]
}
```

Struttura del payload JSON

[Il payload dei messaggi MQTT è scritto in JSON e segue il formato del PutAssetPropertyValueEntry messaggio definito nell'API Reference.](#) [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

Affinché un messaggio sia considerato valido, può essere soddisfatta solo una delle seguenti condizioni:

- `propertyAlias` è impostato, oppure
- Entrambe `assetId` e due `propertyId` sono impostate

`PutAssetPropertyValueEntry` ha un `entryId` campo che non è obbligatorio in questo contesto.

Configurare il broker EMQX

Questa sezione spiega come aggiungere nomi utente e password. Spiega anche come stabilire una connessione TLS da una fonte esterna utilizzando il nome utente e la password aggiunti. È possibile configurare il broker EMQX utilizzando Linux o. Microsoft Windows

Note

Per configurare il broker, è necessario un dispositivo principale configurato con la configurazione EMQX predefinita nel gateway V3 abilitato per MQTT.

⚠ Important

Dopo aver completato questa procedura, consigliamo vivamente di configurare le regole di autorizzazione. Per ulteriori informazioni, consulta [Imposta le regole di autorizzazione per AWS IoT SiteWise Edge in EMQX](#). Le regole di autorizzazione per gli utenti aggiunti migliorano la sicurezza.

Aggiornare la configurazione di distribuzione EMQX per l'autenticazione

Per aggiornare la configurazione di distribuzione EMQX per l'autenticazione

1. Passare alla [console AWS IoT SiteWise](#).
2. Nella barra di navigazione a sinistra, scegli Edge gateway nella sezione Edge.
3. Scegli il gateway da configurare.
4. Nella sezione di configurazione del gateway Edge, copia il valore del dispositivo principale Greengrass. Salvalo per un uso successivo.
5. Apri la [AWS IoT console](#).
6. Nella barra di navigazione a sinistra, nella sezione Gestisci, scegli Dispositivi Greengrass, quindi Implementazioni.
7. Trova il valore principale del dispositivo che hai salvato in precedenza e scegli quel link per aprire la distribuzione.
8. Scegli il pulsante a discesa Azioni, quindi Rivedi.
9. Leggi il messaggio che appare, quindi scegli Revise deployment. Viene visualizzata la pagina Specificare la destinazione.
10. Scegli Avanti fino a raggiungere il passaggio Configura componenti.
11. Seleziona il pulsante `aws.greengrass.clientdevices.mqtt.EMQX radio`.
12. Scegli il pulsante Configura componente. Viene visualizzata una pagina di configurazione per il componente.
13. In Aggiornamento della configurazione, scegli Ripristina la configurazione predefinita per la versione del componente: 2.*.*.
14. Inserisci la seguente configurazione nella sezione Configurazione da unire in base al tuo sistema operativo.

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
    -p 127.0.0.1:18083:18083 -v emqx-data:/opt/emqx/data -e
    EMQX_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"  
}
```

Il `dockerOptions` campo è solo per i gateway Linux.

15. Scegli Conferma.
16. Scegli Avanti fino a raggiungere la fase di revisione.
17. Seleziona Deploy (Implementa).
18. Una volta completata la distribuzione, procedi al passaggio successivo.

Abilita l'autenticazione con nome utente e password

Questa sezione mostra come aggiungere nomi utente e password tramite la GUI del dashboard EMQX.

Note

Le istruzioni relative a EMQX fornite sono solo di riferimento. Poiché la documentazione e le funzionalità di EMQX possono cambiare nel tempo e non conserviamo tale documentazione, consigliamo di consultare la documentazione [ufficiale di EMQX per le informazioni più aggiornate](#).

EMQX Dashboard

Per abilitare l'autenticazione di nome utente e password tramite il pannello di controllo EMQX

1. Assicurati di essere all'interno dell'host del gateway.
2. Apri una finestra del browser e visita <http://localhost:18083/>.

3. Inserisci il nome utente **admin** e la password predefiniti di **public**. Per ulteriori informazioni, vedere [EMQX Dashboard nei documenti](#) EMQX.
4. Dopo l'accesso, ti viene richiesto di cambiare la password. Aggiorna la password per continuare a utilizzare il pannello di controllo EMQX.
5. Nella barra di navigazione a sinistra, scegli l'icona a forma di scudo, quindi Autenticazione.
6. Nella riga Database integrato, scegli il pulsante Utenti.
7. Scegli il pulsante con l'icona del segno più per aggiungere utenti. Viene visualizzata una schermata Aggiungi.
8. Immettere un nome utente e una password per l'utente dell'applicazione esterna.
9. Scegli Save (Salva). Il nome utente scelto viene visualizzato nella tabella della pagina Autenticazione.

Note

Le regole di autorizzazione esistenti o predefinite si applicano al nuovo utente. Si consiglia di rivederle e adattarle alle esigenze delle applicazioni esterne.

EMQX Management with Linux

Utilizza lo strumento AWS IoT SiteWise CLI EMQX all'indirizzo. `/greengrass/v2/bin/swe-emqx-cli`

Per abilitare l'autenticazione di nome utente e password tramite EMQX Management utilizzando Linux

1. Modificare la password dell'amministratore eseguendo il seguente comando:

```
/greengrass/v2/bin/swe-emqx-cli admin change-pwd
```

2. Quando richiesto, procedi come segue:
 1. Immettete l'utente amministratore corrente (l'impostazione predefinita è `admin`) e la password (l'impostazione predefinita è `public`).
 2. Inserisci e conferma la tua nuova password.

In caso di successo, viene visualizzato il seguente messaggio:

```
admin password changed successfully
```

3. Aggiungi utenti per applicazioni esterne eseguendo il comando seguente:

```
/greengrass/v2/bin/swe-emqx-cli users add
```

4. Quando richiesto, effettuate le seguenti operazioni:

1. Immettete il nome utente per il nuovo utente.
2. Inserisci e conferma la password per il nuovo utente.

In caso di successo, viene visualizzato il seguente messaggio:

```
User '[username]' created successfully
```

5. Verifica la configurazione utente eseguendo il comando seguente:

```
/greengrass/v2/bin/swe-emqx-cli users list
```

L'output mostra tutti gli utenti configurati:

```
Users:  
- [your-added-username]  
  
Total users: 1
```

EMQX Management with Windows

Utilizzate lo strumento AWS IoT SiteWise CLI EMQX in una delle seguenti posizioni:

- PowerShell: C:\greengrass\v2\bin\swe-emqx-cli.ps1
- Prompt dei comandi: C:\greengrass\v2\bin\swe-emqx-cli.bat

Per abilitare l'autenticazione di nome utente e password tramite EMQX Management utilizzando Windows

1. Modificare la password dell'amministratore eseguendo il comando seguente:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 admin change-pwd
```

2. Quando richiesto, procedi come segue:

1. Immettete l'utente amministratore corrente (l'impostazione predefinita è `admin`) e la password (l'impostazione predefinita è `public`).
2. Inserisci e conferma la tua nuova password.

In caso di successo, viene visualizzato il seguente messaggio:

```
admin password changed successfully
```

3. Aggiungi utenti per applicazioni esterne eseguendo il comando seguente:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 users add
```

4. Quando richiesto, effettuate le seguenti operazioni:

1. Immettete il nome utente per il nuovo utente.
2. Inserisci e conferma la password per il nuovo utente.

In caso di successo, viene visualizzato il seguente messaggio:

```
User '[username]' created successfully
```

5. Verifica la configurazione utente eseguendo il comando seguente:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 users list
```

L'output mostra tutti gli utenti configurati:

```
Users:  
- [your-added-username]
```

Total users: 1

Connect un'applicazione al broker EMQX su Edge AWS IoT SiteWise

Il broker EMQX utilizza Transport Layer Security (TLS) sulla porta 8883 per crittografare tutte le comunicazioni, garantendo la protezione dei dati durante la trasmissione. Questa sezione illustra i passaggi per stabilire connessioni tra le applicazioni e il broker EMQX. Seguire questi passaggi aiuta a mantenere l'integrità e la riservatezza dei dati industriali. Il processo di connessione prevede due approcci principali: l'utilizzo del rilevamento automatico degli IP tramite componenti o la configurazione manuale dei nomi e degli indirizzi IP DNS come Subject Alternative Names (SANs) nei certificati TLS. Ogni metodo presenta i propri vantaggi a seconda della configurazione della rete e dei requisiti di sicurezza. Questa documentazione ti guiderà attraverso entrambe le opzioni.

Argomenti

- [Configura TLS per connessioni sicure al broker EMQX su Edge AWS IoT SiteWise](#)
- [Verifica la connessione del broker EMQX su Edge AWS IoT SiteWise](#)
- [Usa la tua CA](#)
- [Aprire la porta 8883 per le connessioni firewall esterne](#)

Configura TLS per connessioni sicure al broker EMQX su Edge AWS IoT SiteWise

Per impostazione predefinita, AWS IoT Greengrass genera un certificato del server TLS per il broker EMQX firmato dall'autorità di certificazione (CA) del dispositivo principale. Per ulteriori informazioni, consulta [Connessione dei dispositivi client a un dispositivo AWS IoT Greengrass Core con un broker MQTT](#).

Recupera il certificato TLS

Per ottenere il certificato CA, esegui il seguente comando sull'host del gateway:

Linux

Esegui il comando seguente in una sessione di shell sull'host del gateway:

```
/greengrass/v2/bin/swe-emqx-cli cert
```

Questo comando visualizza la posizione del certificato e ne stampa il contenuto.

In alternativa, è possibile salvare il certificato in un file utilizzando questo comando:

```
/greengrass/v2/bin/swe-emqx-cli cert --output /path/to/certificate.pem
```

Windows

Esegui il comando seguente in una PowerShell sessione sull'host del gateway:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 cert
```

Questo comando visualizza la posizione del certificato e ne stampa il contenuto.

In alternativa, è possibile salvare il certificato in un file utilizzando questo comando:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 cert --output C:\path\to\certificate.pem
```

La CLI individua automaticamente il certificato indipendentemente dal percorso esatto sul sistema.

Copia il contenuto del file `ca.pem` nell'applicazione esterna che stai collegando al broker. Salvalo come `BrokerCoreDeviceCA.pem`.

Aggiungi nomi/indirizzi IP DNS personalizzati al certificato del server TLS

Il nome alternativo dell'oggetto (SAN) sul certificato generato da è. `AWS IoT Greengrass localhost`. Quando si stabilisce una connessione TLS dall'esterno dell'host del gateway, la fase di verifica TLS non riesce perché il nome host del broker non corrisponde al nome host del certificato del `localhost` server.

Per risolvere il problema della mancata corrispondenza del nome host, AWS IoT Greengrass offre due modi per gestire gli endpoint dei dispositivi principali. Questa sezione copre entrambe le opzioni. Per informazioni più dettagliate, consulta [Gestire gli endpoint principali dei dispositivi](#) nella Guida per gli AWS IoT Greengrass Version 2 sviluppatori.

- Per connetterti al broker EMQX utilizzando l'indirizzo IP del dispositivo principale, utilizza la sezione `Automated IP Discovery`.
- Per connettersi al broker EMQX utilizzando un nome DNS anziché un indirizzo IP, si utilizza la sezione `Gestione manuale`.

Automated IP discovery

Questa opzione consente al dispositivo principale di rilevare automaticamente il proprio indirizzo IP e aggiungerlo come Subject Alternative Name (SAN) al certificato del broker.

1. Aggiungi il `aws.greengrass.clientdevices.IPDetector` componente alla distribuzione del tuo dispositivo principale.
2. Implementa le modifiche sul tuo dispositivo
3. Attendi il completamento della distribuzione.

Una volta completata la distribuzione, puoi stabilire una connessione TLS sicura utilizzando l'indirizzo IP del broker.

L'indirizzo IP viene aggiunto automaticamente come SAN al certificato del broker.

Manual DNS and IP Configuration

Puoi aggiungere manualmente nomi DNS e indirizzi IP come Subject Alternative Names (SANs) al tuo certificato TLS. Questo metodo è utile quando è stato configurato un nome DNS per l'host del gateway.

Important

Se stai utilizzando il `IPDetector` componente, rimuovilo dalla distribuzione prima di procedere. Il `IPDetector` componente sostituisce le configurazioni manuali degli endpoint.

Per configurare manualmente gli endpoint

1. Passare alla [console AWS IoT SiteWise](#).
2. Nella barra di navigazione a sinistra, scegli Edge gateway nella sezione Edge.
3. Scegli il gateway da configurare.
4. Nella sezione di configurazione del gateway Edge, scegli l'URL del dispositivo principale Greengrass. Viene visualizzata la pagina del dispositivo principale.
5. Scegli la scheda Dispositivi client.
6. Scegli Gestisci endpoint.

7. Nella finestra di dialogo Gestisci endpoint, inserisci i nomi DNS e gli eventuali indirizzi IP che desideri aggiungere. SANs Usa la porta 8883.
8. Scegli Aggiorna.

Il certificato del server TLS del broker si aggiorna automaticamente per includere i nuovi endpoint.

Per verificare l'aggiornamento del certificato del server TLS tramite Linux

1. Avvia una sessione di shell sul tuo host gateway.

```
docker exec emqx openssl x509 -in ./data/cert.pem -text -noout | grep -A1  
"Subject Alternative Name"
```

2. Il comando restituisce un output simile al seguente:

```
X509v3 Subject Alternative Name:  
DNS:endpoint_you_added, DNS:localhost
```

3. Verifica che il tuo endpoint compaia nell'elenco di SANs.

Per verificare l'aggiornamento del certificato del server TLS tramite Windows

1. Avvia una sessione di shell sull'host del gateway.

```
(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. Il comando restituisce un output simile al seguente:

```
Subject Alternative Name:  
DNS Name=your-endpoint  
DNS Name=localhost
```

3. Verifica che l'endpoint che hai aggiunto sia nell'elenco di SANs.

Verifica la connessione del broker EMQX su Edge AWS IoT SiteWise

Dopo aver configurato il broker EMQX con certificati TLS e credenziali di autenticazione, è importante verificare che la configurazione funzioni correttamente. Il test della connessione aiuta a garantire che le configurazioni di sicurezza siano implementate correttamente e che i client possano stabilire correttamente connessioni crittografate con il broker. Questa sezione illustra come testare la connessione del broker utilizzando il client Mosquitto Command Line Interface (CLI), uno strumento client MQTT ampiamente utilizzato che supporta la crittografia e l'autenticazione TLS.

Usa il client CLI Mosquitto per testare la connessione del broker EMQX

In questo passaggio utilizzeremo il client CLI mosquitto per testare la nostra configurazione e assicurarci di poterci connettere correttamente al broker utilizzando il nome utente e la password che abbiamo creato in precedenza. Per eseguire i seguenti `BrokerCoreDeviceCA.pem` passaggi nella Fase 3: Configurazione di TLS.

```
mosquitto_sub -h hostname/ip address \  
  -p 8883 \  
  -t "#" \  
  -q 1 \  
  -u username -P password \  
  --cafile BrokerCoreDeviceCA.pem
```

Note

Potresti ricevere un errore `SSL:Verify` se l' `hostname/IP` indirizzo a cui ti stai connettendo non corrisponde al Subject Alternative Name (SAN) riportato sul certificato CA che stai trasmettendo al client. Vedi «Aggiungere names/IP indirizzi DNS personalizzati al certificato del server TLS» nel Passaggio 3: Configurazione di TLS per informazioni su come ottenere un certificato con la SAN corretta.

A questo punto, tutti gli utenti hanno accesso alla pubblicazione e alla sottoscrizione di tutti gli argomenti sul broker. Passa a [Imposta le regole di autorizzazione per AWS IoT SiteWise Edge in EMQX](#).

Usa la tua CA

AWS IoT Greengrass descrive come configurare il proprio componente di autenticazione del dispositivo client per utilizzare la propria autorità di certificazione (CA). Il componente di

autenticazione del dispositivo client (`aws.greengrass.clientdevices.Auth`) autentica i dispositivi client e autorizza le azioni dei dispositivi client. Per ulteriori informazioni, consulta [Using your own certificate authority](#) nella Developer Guide.AWS IoT Greengrass Version 2

Per utilizzare la tua CA, aggiungi il `aws.greengrass.clientdevices.Auth` componente alla tua distribuzione in modo da poter specificare una configurazione personalizzata.

Aprire la porta 8883 per le connessioni firewall esterne

Linux

Nella regola del firewall dell'host Linux, aggiungi una regola in entrata per la porta 8883 per consentire le connessioni in entrata dall'esterno dell'host gateway. Se sono presenti firewall, assicurati che le connessioni TLS in entrata sulla porta 8883 siano consentite.

Windows

Nella regola del firewall Microsoft Windows dell'host, aggiungi una regola in entrata per la porta 8883 per consentire le connessioni in entrata dall'esterno dell'host gateway. Assicurati che la regola sia una regola di autorizzazione, di tipo port, specificando la porta 8883. Puoi configurarlo in base alla tua configurazione di rete per consentire le connessioni dalle tue applicazioni esterne al broker.

Imposta le regole di autorizzazione per AWS IoT SiteWise Edge in EMQX

EMQX supporta l'aggiunta di regole di autorizzazione basate su identificatori come nome utente, indirizzo IP o ID client. Ciò è utile se si desidera limitare il numero di applicazioni esterne che si connettono a varie operazioni o argomenti.

Argomenti

- [Configura l'autorizzazione utilizzando il database integrato con Linux](#)
- [Configura l'autorizzazione utilizzando il database integrato con Windows](#)
- [Aggiornare la configurazione di distribuzione EMQX per l'autorizzazione](#)
- [Aggiungi regole di autorizzazione tramite il pannello di controllo EMQX per gli utenti](#)

Configura l'autorizzazione utilizzando il database integrato con Linux

Quando si configurano le regole di autorizzazione, sono disponibili due scelte di configurazione che dipendono dalla configurazione di distribuzione.

- **Docker**— Se stai eseguendo un'installazione standard senza Litmus Edge, usa la configurazione del gateway Docker bridge. Questo è in genere il caso in cui sono stati distribuiti solo componenti AWS IoT SiteWise.
- **Litmus Edge**— Se l'avete Litmus Edge installato sul gateway, utilizzate la configurazione della sottorete di rete.

Note

Se configurate inizialmente il Docker bridge gateway e successivamente lo installate Litmus Edge, riconfigurate le regole di autorizzazione utilizzando l'opzione di Litmus Edge sottorete di rete per garantire una comunicazione corretta tra tutti i componenti.

Per aggiungere regole di autorizzazione di base

1. Verificare che il broker EMQX sia distribuito e funzionante.
2. Avvia una sessione di shell sul tuo host gateway.

Docker without Litmus Edge

Per Docker un'installazione standard senza Litmus Edge, esegui:

```
/greengrass/v2/bin/swe-emqx-cli acl init
```

Litmus Edge network subnet

Se lo utilizzi Litmus Edge, determina l'IP della sottorete di rete Litmus Edge:

```
docker network inspect LitmusNetwork | grep IPAM -A9
```

Annota il valore di sottorete dall'output ed esegui il comando seguente. Sostituisci `litmus_subnet_ip` con il valore Subnet del passaggio precedente.

```
/greengrass/v2/bin/swe-emqx-cli acl init litmus_subnet_ip
```

Lo strumento crea e applica automaticamente le regole di autorizzazione per consentire le connessioni dall'indirizzo IP fornito al broker. Consente l'accesso a tutti gli argomenti. Ciò include il raccogliatore IoT SiteWise OPC UA e l'editore SiteWise IoT.

3. Passa a [Aggiornare la configurazione di distribuzione EMQX per l'autorizzazione](#).

Configura l'autorizzazione utilizzando il database integrato con Windows

Questa sezione descrive la configurazione delle regole di autorizzazione utilizzando il database integrato per le distribuzioni Windows.

Per aggiungere regole di autorizzazione di base

1. Verificare che il broker EMQX sia distribuito e funzionante.
2. Esegui lo strumento AWS IoT SiteWise CLI EMQX:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 acl init
```

Lo strumento crea e applica automaticamente le regole ACL che consentono le connessioni da localhost (127.0.0.1) al broker. Consente l'accesso a tutti gli argomenti. Ciò include il raccogliatore IoT SiteWise OPC UA e l'editore SiteWise IoT.

3. Passa a [Aggiornare la configurazione di distribuzione EMQX per l'autorizzazione](#).

Aggiornare la configurazione di distribuzione EMQX per l'autorizzazione

Per aggiornare la configurazione di distribuzione EMQX per l'autorizzazione

1. Passare alla [console AWS IoT SiteWise](#).
2. Nella barra di navigazione a sinistra, scegli Edge gateway nella sezione Edge.
3. Scegli il gateway da configurare.
4. Nella sezione di configurazione del gateway Edge, copia il valore del dispositivo principale Greengrass. Salvalo per un uso successivo.
5. Apri la [AWS IoT console](#).
6. Nella barra di navigazione a sinistra, nella sezione Gestisci, scegli Dispositivi Greengrass, quindi Implementazioni.

7. Trova il valore principale del dispositivo che hai salvato in precedenza e scegli quel link per aprire la distribuzione.
8. Scegli il pulsante a discesa Azioni, quindi Rivedi.
9. Leggi il messaggio che appare, quindi scegli Revise deployment. Viene visualizzata la pagina Specificare la destinazione.
10. Scegli Avanti fino a raggiungere il passaggio Configura componenti.
11. Seleziona il pulsante `aws.greengrass.clientdevices.mqtt.EMQX` radio.
12. Scegli il pulsante Configura componente. Viene visualizzata una pagina di configurazione per il componente.
13. In Aggiornamento della configurazione, scegli Ripristina la configurazione predefinita per la versione del componente: 2.*.*.
14. Incolla il seguente contenuto nella sezione Configurazione da unire in base al tuo sistema operativo.

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. Scegli Conferma.
16. Scegli Avanti fino a raggiungere la fase di revisione.
17. Seleziona Deploy (Implementa).

Note

Da questo momento in poi, non puoi modificare il file ACL per aggiornare le regole di autorizzazione. In alternativa, è possibile procedere a [Aggiungi regole di autorizzazione tramite il pannello di controllo EMQX per gli utenti](#) dopo una distribuzione riuscita.

Aggiungi regole di autorizzazione tramite il pannello di controllo EMQX per gli utenti

È possibile aggiungere o aggiornare le regole di autorizzazione utilizzando il dashboard EMQX o lo strumento AWS IoT SiteWise EMQX CLI. Lo strumento AWS IoT SiteWise CLI EMQX gestisce l'autorizzazione utilizzando il database integrato di EMQX.

Note

L'aggiunta di regole di autorizzazione è una fase di configurazione avanzata che richiede la comprensione dei pattern tematici di MQTT e del controllo degli accessi. Per ulteriori informazioni sulla creazione di regole di autorizzazione utilizzando il database integrato di EMQX, vedere [Use Built-in Database in the EMQX Docs](#).

Note

Le istruzioni relative a EMQX fornite sono solo di riferimento. Poiché la documentazione e le funzionalità di EMQX possono cambiare nel tempo e non conserviamo tale documentazione, consigliamo di consultare la documentazione [ufficiale di EMQX per le informazioni più aggiornate](#).

EMQX dashboard

Questa procedura mostra come aggiungere regole di autorizzazione al dashboard EMQX.

Il dashboard EMQX è accessibile solo dall'host del gateway. Se tenti di connetterti dall'esterno dell'host del gateway, non puoi accedere alla dashboard.

Per aggiungere regole di autorizzazione utilizzando il pannello di controllo EMQX

1. Assicurati di essere all'interno dell'host del gateway.
2. Apri una finestra del browser e visita <http://localhost:18083/>.
3. Accedi al pannello di controllo EMQX. Questa procedura presuppone che tu abbia cambiato le credenziali di accesso predefinite con qualcosa di tua scelta. Per ulteriori informazioni sulla configurazione iniziale, vedere. [Abilita l'autenticazione con nome utente e password](#)
4. Scegli l'icona a forma di scudo, quindi Autorizzazione dal menu a discesa.
5. Scegli il pulsante Autorizzazioni nella riga Database integrato.
6. Nella sezione Autorizzazione del database integrato, aggiungi o aggiorna le regole di autorizzazione degli utenti per le tue esigenze aziendali. Per ulteriori indicazioni sulla creazione di regole, vedere la sezione [Utilizzo del database integrato](#) nei documenti EMQX.

AWS IoT SiteWise CLI tool using Linux

Per gestire le regole di autorizzazione utilizzando lo strumento AWS IoT SiteWise CLI EMQX in Linux:

- Aggiungi regole di autorizzazione per un utente utilizzando il seguente formato:

```
/greengrass/v2/bin/swe-emqx-cli auth add your-username your-action your-permission your-topic [your-action-permission-topic]
```

Example Aggiungere regole di autorizzazione per un utente

Questo esempio mostra come aggiungere regole per un utente denominato `system1`:

```
/greengrass/v2/bin/swe-emqx-cli auth add system1 \  
  publish allow "sensors/#" \  
  subscribe allow "control/#" \  
  
```

```
all deny "#"
```

Example : Visualizza le regole di autorizzazione per un utente

Per visualizzare le regole di autorizzazione per gli `system1` utenti, esegui il comando seguente:

```
/greengrass/v2/bin/swe-emqx-cli auth list system1
```

Example : Visualizza tutte le regole di autorizzazione esistenti

Per visualizzare tutte le regole di autorizzazione attualmente in uso, esegui il comando seguente:

```
/greengrass/v2/bin/swe-emqx-cli auth list
```

Example : elimina tutte le regole di autorizzazione per un utente

Per eliminare tutte le regole di autorizzazione applicate a un determinato utente, esegui il comando seguente:

```
/greengrass/v2/bin/swe-emqx-cli auth delete system1
```

Ti viene richiesto di confermare l'eliminazione.

AWS IoT SiteWise CLI tool using Windows

Per gestire le regole di autorizzazione utilizzando lo strumento AWS IoT SiteWise CLI EMQX in: Windows PowerShell

- Aggiungere regole di autorizzazione per un utente utilizzando il seguente formato:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth add your-username your-action your-permission your-topic [your-action-permission-topic]
```

Example : aggiungi regole di autorizzazione per un utente

Questo esempio mostra come aggiungere regole per un utente denominato `system1`:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth add system1 `
```

```
publish allow "sensors/#" `
subscribe allow "control/#" `
all deny "#"
```

Example : Visualizza le regole di autorizzazione per un utente

Per visualizzare le regole di autorizzazione per gli system1 utenti, esegui il comando seguente:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth list system1
```

Example : Visualizza tutte le regole di autorizzazione esistenti

Per visualizzare tutte le regole di autorizzazione attualmente in uso, esegui il comando seguente:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth list
```

Example : elimina tutte le regole di autorizzazione per un utente

Per eliminare tutte le regole di autorizzazione applicate a un determinato utente, esegui il comando seguente:

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth delete system1
```

Ti viene richiesto di confermare l'eliminazione.

Elabora e visualizza i dati con SiteWise Edge e strumenti open source

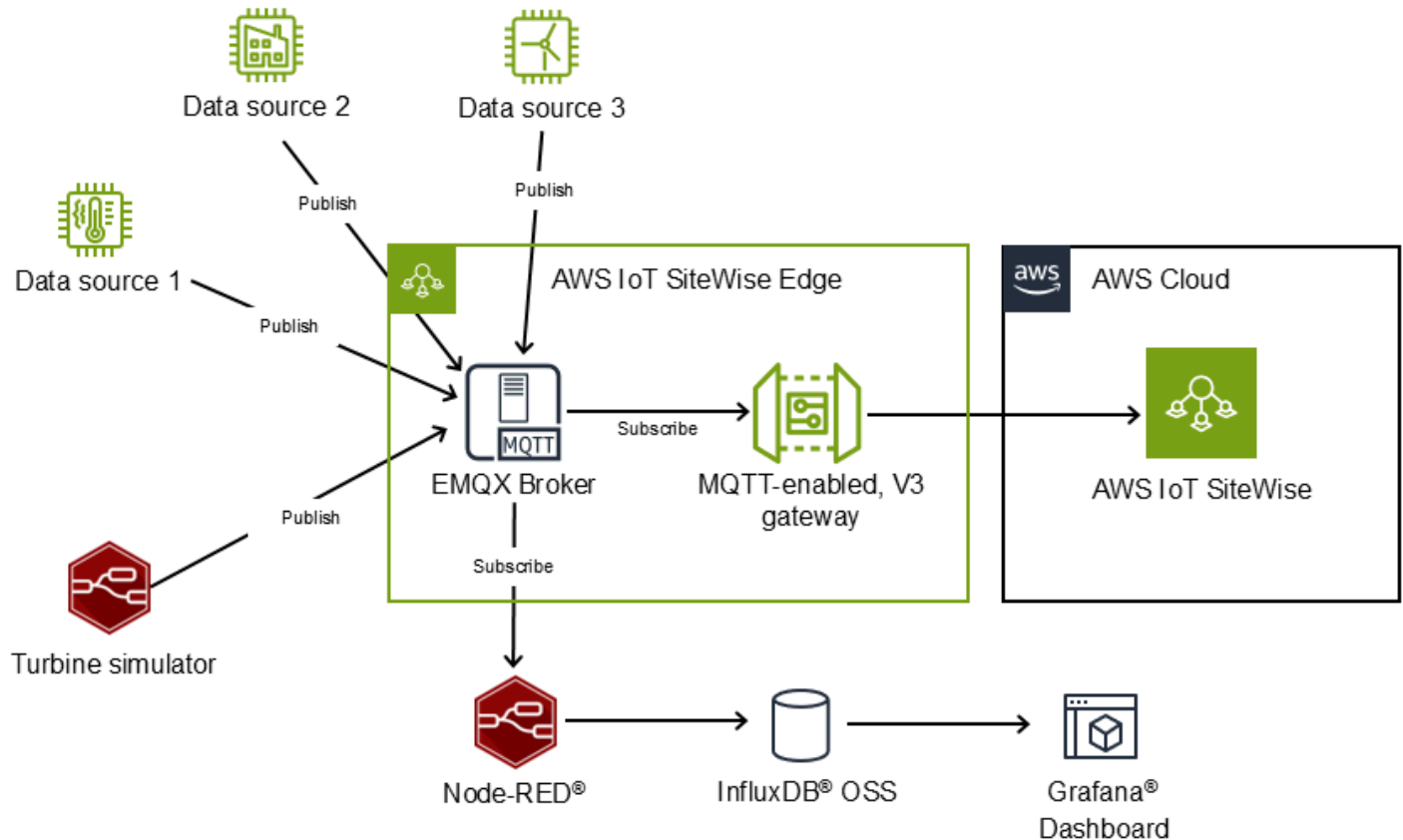
Configura i gateway abilitati per AWS IoT SiteWise Edge MQTT con strumenti open source per l'elaborazione e la visualizzazione locali per migliorare le capacità di gestione dei dati industriali.

Con SiteWise Edge, puoi creare una pipeline di elaborazione dei dati locale utilizzando strumenti esterni open source. [Usa Node-RED® per archiviare dati di serie temporali con InfluxDB® e monitorare le operazioni tramite i dashboard Grafana®.](#)

Node-RED elabora e trasforma i flussi di dati, mentre InfluxDB fornisce l'archiviazione dei dati in serie temporali. Grafana mostra i tuoi dati operativi in tempo reale. Usa questi strumenti con SiteWise Edge per sincronizzare i dati tra il tuo ambiente locale e il Cloud AWS, offrendoti sia approfondimenti locali immediati che funzionalità di analisi a lungo termine basate sul cloud.

Note

Node-RED®, InfluxDB® e Grafana® non sono vendor o fornitori di Edge. SiteWise

**Note**

In questa guida, utilizziamo la versione open source di [Grafana](#) SiteWise for Edge anziché il servizio [Amazon Managed Grafana](#).

Opzioni di implementazione

Puoi implementare questa soluzione utilizzando uno dei due approcci. Con una configurazione Microsoft Windows manuale, puoi controllare la configurazione e l'integrazione dei componenti con la tua infrastruttura. Con Linux, puoi usare Docker per distribuire componenti preconfigurati nei contenitori.

Scegli il metodo che soddisfa i tuoi requisiti operativi.

- [Configura manualmente le integrazioni open source \(Windows\)](#)— Per configurazioni personalizzate o infrastrutture esistenti
- [Configura integrazioni open source con Docker \(Linux\)](#)— Per un'implementazione rapida con componenti preconfigurati

Panoramica degli esempi di parchi eolici

Questa guida utilizza un esempio di parco eolico per dimostrare come monitorare la velocità del vento per una turbina in un parco eolico. Questo scenario pratico illustra le esigenze comuni di monitoraggio industriale in cui la visibilità locale e basata sul cloud è preziosa per l'efficienza operativa.

Con questa integrazione, puoi:

- Raccogli dati dalle apparecchiature industriali utilizzando un gateway AWS IoT SiteWise Edge
- Elabora i dati localmente con Node-RED, InfluxDB e Grafana
- Archivia i dati localmente usando InfluxDB
- Monitora i dati in tempo reale utilizzando le dashboard Grafana

In questa guida, utilizziamo l'esempio di un parco eolico. Utilizziamo Node-RED per simulare una turbina che genera dati sulla velocità del vento. Node-RED traduce il payload di dati, pubblica i dati sul broker SiteWise Edge MQTT, si abbona per ricevere dati dal broker e archivia i dati localmente in InfluxDB. Questo approccio garantisce che tutti i dati operativi siano disponibili sia localmente per un accesso immediato che nel cloud per ulteriori analisi. Implementando questo modello, si ottiene la resilienza contro le interruzioni della rete pur mantenendo la capacità di eseguire analisi avanzate in. Cloud AWS Grafana si connette a InfluxDB per il monitoraggio locale, fornendo agli operatori una visibilità in tempo reale sulle metriche senza dipendenze dal cloud. Un gateway abilitato per SiteWise Edge MQTT si connette allo stesso broker MQTT a cui inviare i dati AWS IoT SiteWise, creando un ponte tra le operazioni edge e i servizi basati su cloud.

Puoi utilizzare i tuoi dati e le tue configurazioni per creare un flusso di lavoro simile personalizzato in base ai tuoi requisiti industriali specifici, indipendentemente dal fatto che tu stia monitorando apparecchiature di produzione, infrastrutture di pubblica utilità o altri asset industriali.

Requisiti per le integrazioni open source

Prima di implementare integrazioni open source con SiteWise Edge, assicurati che il tuo ambiente soddisfi i requisiti necessari.

- Requisiti hardware: l'hardware del gateway deve soddisfare i requisiti per i gateway SiteWise Edge. Per ulteriori informazioni, vedere [AWS IoT SiteWise Requisiti del gateway Edge self-hosted](#) per i gateway V3 compatibili con MQTT e [Requisiti per l'applicazione AWS IoT SiteWise Edge](#)

 Important

Quando distribuisce componenti open source aggiuntivi, assicurati che l'hardware soddisfi i requisiti per InfluxDB, Node-RED e Grafana.

- La configurazione di rete deve supportare sia la comunicazione locale tra i componenti sia la connettività cloud per Edge. SiteWise
- Tutti i servizi devono essere eseguiti sullo stesso host.

Considerazioni relative alla sicurezza

Si consiglia di crittografare tutte le comunicazioni tra i componenti, soprattutto quando si accede alle interfacce da reti non locali. Implementa controlli di accesso adeguati per ogni componente e segui le AWS migliori pratiche per la configurazione del gateway AWS IoT SiteWise Edge e AWS la sicurezza dell'account.

Ambiente di sviluppo

Questa guida illustra Node-RED, InfluxDB e Grafana in esecuzione e a cui si accede localmente su un host gateway. Per le implementazioni di produzione che richiedono l'accesso esterno, implementa misure di sicurezza tra cui crittografia, autenticazione e autorizzazione TLS. Segui le best practice di sicurezza di ogni applicazione.

Software di terze parti

Questa soluzione utilizza software di terze parti non gestito da AWS, tra cui InfluxDB, Node-RED, Grafana e il plug-in. `node-red-contrib-influxdb` Prima della distribuzione, assicuratevi che questi componenti siano conformi ai requisiti di sicurezza, agli standard di conformità e alle politiche di governance della vostra organizzazione.

 Important

Questa guida fa riferimento e utilizza software di terze parti non di proprietà o gestito da AWS. Prima dell'implementazione, assicuratevi che tutti i componenti soddisfino i requisiti di

sicurezza, conformità e governance. Mantieni tutto il software aggiornato con le ultime patch di sicurezza e segui le migliori pratiche per proteggere l'implementazione perimetrale. InfluxDB, Node-RED, Grafana non sono vendor o fornitori di Edge. SiteWise

Altre considerazioni

Considerate questi fattori aggiuntivi quando implementate integrazioni open source con Edge. SiteWise

- Utilizza le versioni più recenti di tutti i servizi, gli strumenti e i componenti.
- Filtra e aggrega i dati localmente prima della trasmissione nel cloud per ridurre AWS IoT SiteWise i costi di acquisizione dei dati. Configura i periodi di conservazione dei dati appropriati in InfluxDB e dimensiona correttamente l'hardware del gateway. Per ulteriori informazioni, consulta [Prezzi di AWS IoT SiteWise](#).
- Implementa procedure di backup regolari per tutti i dati.
- Monitora l'utilizzo delle risorse sul gateway e configura i limiti di risorse appropriati per ogni componente. Implementa politiche di conservazione dei dati in InfluxDB per gestire l'utilizzo del disco.

Configura manualmente le integrazioni open source (Windows)

Usa questa guida per creare manualmente un bucket di serie temporali per i dati sulla velocità del vento che si collega a Grafana® e Node-RED®.

Installa e configura manualmente Node-RED, InfluxDB® e Grafana per controllare la configurazione della distribuzione. Microsoft Windows Puoi archiviare e gestire i dati delle serie temporali dai tuoi dispositivi utilizzando InfluxDB.

Prerequisiti di configurazione manuale

Prima di iniziare, completa questi requisiti:

Note

Esegui tutti i servizi (SiteWise Edge, InfluxDB, Node-RED e Grafana) sullo stesso host.

- Installa un gateway V3 abilitato per MQTT. Per ulteriori informazioni, consulta [Gateway V3 compatibili con MQTT per Edge AWS IoT SiteWise](#).
- Installa ed esegui questi servizi localmente:
 - InfluxDB OSS v2. [Per i passaggi di installazione, consulta Installa InfluxDB](#).
 - Nodo rosso. Per i passaggi di installazione, vedi [Installare Node-RED](#) localmente.
 - Grafana. Per i passaggi di installazione, consulta [Installare Grafana](#).

Configura l'archiviazione locale con InfluxDB

Con InfluxDB®, puoi archiviare localmente i dati delle serie temporali dai tuoi dispositivi. Lo scopo della capacità di archiviazione locale è mantenere la visibilità operativa durante le interruzioni della rete e ridurre la latenza per le applicazioni che richiedono tempi critici. È possibile eseguire analisi e visualizzazioni all'edge pur avendo la possibilità di inoltrare selettivamente i dati al cloud.

In questa sezione, crei un bucket di serie temporali per i dati sulla velocità del vento delle turbine e generi un token API per la connettività Grafana® e Node-RED®. Il bucket InfluxDB funge da contenitore di archiviazione dedicato per i dati delle serie temporali, in modo simile a un database nei sistemi tradizionali. Il token API consente un accesso programmatico sicuro ai dati.

Per configurare InfluxDB

1. [Dopo aver completato i passaggi preliminari e aver verificato che tutti gli strumenti siano in esecuzione sullo stesso host, apri il browser Web e vai su `http://127.0.0.1:8086`.](#)
2. (Facoltativo) Abilita la crittografia TLS per una maggiore sicurezza. Per ulteriori informazioni, consulta [Abilitare la crittografia TLS](#) in. InfluxData Documentation
3. Crea un bucket InfluxDB di serie temporali per archiviare i dati da Node-RED. Il bucket fungerà da contenitore dedicato per i dati del parco eolico, consentendoti di organizzare e gestire le politiche di conservazione specifiche per questo set di dati. Per ulteriori informazioni, consulta [Gestire i bucket](#) in. InfluxData Documentation
4. (Facoltativo) Configura il periodo di conservazione dei dati per la tua edge location. L'impostazione di periodi di conservazione appropriati aiuta a gestire le risorse di storage in modo efficiente rimuovendo automaticamente i dati più vecchi che non sono più necessari per le operazioni locali.

Per informazioni sulla conservazione dei dati, consulta [Conservazione dei dati in InfluxDB](#) nel. InfluxData Documentation

5. Genera un token API per il bucket. Questo token consentirà una comunicazione sicura tra InfluxDB e altri componenti come Node-RED e Grafana. In questo modo, solo i servizi autorizzati possono leggere o scrivere nel tuo archivio dati. Per ulteriori informazioni, consulta [Creare un token](#) in InfluxData Documentation.

Dopo aver completato questi passaggi, puoi archiviare i dati delle serie temporali nell'istanza InfluxDB, fornendo una base per la persistenza e l'analisi locali dei dati nell'ambiente edge.

Configura i flussi Node-RED per l'integrazione dei dati AWS IoT SiteWise

Con Node-RED®, puoi implementare due flussi per gestire i dati tra i tuoi dispositivi e AWS IoT SiteWise. Questi flussi collaborano per creare una soluzione completa di gestione dei dati che affronti il flusso di dati sia locale che cloud.

- Flusso di pubblicazione dei dati: pubblicazione nel cloud. Il flusso di pubblicazione dei dati invia i dati a AWS IoT SiteWise. Questo flusso simula un dispositivo a turbina generando i dati dei sensori, traducendoli in AWS IoT SiteWise formato e pubblicandoli nel broker SiteWise Edge MQTT. Ciò consente di sfruttare le funzionalità cloud AWS IoT SiteWise di archiviazione, analisi e integrazione con altri servizi. AWS

Per ulteriori informazioni, consulta [Configura il flusso di pubblicazione dei dati](#).

- Flusso di conservazione dei dati: archivia i dati all'edge. Il flusso di conservazione dei dati si iscrive al broker SiteWise Edge MQTT per ricevere i dati, tradurli in formato InfluxDB® e archivarli localmente per il monitoraggio. Questo storage locale fornisce l'accesso immediato ai dati operativi, riduce la latenza per le applicazioni urgenti e garantisce la continuità durante le interruzioni della rete.

Per ulteriori informazioni, consulta [Configura il flusso di conservazione dei dati](#).

Questi due flussi interagiscono per garantire che i dati vengano inviati AWS IoT SiteWise e archiviati localmente per un accesso immediato.

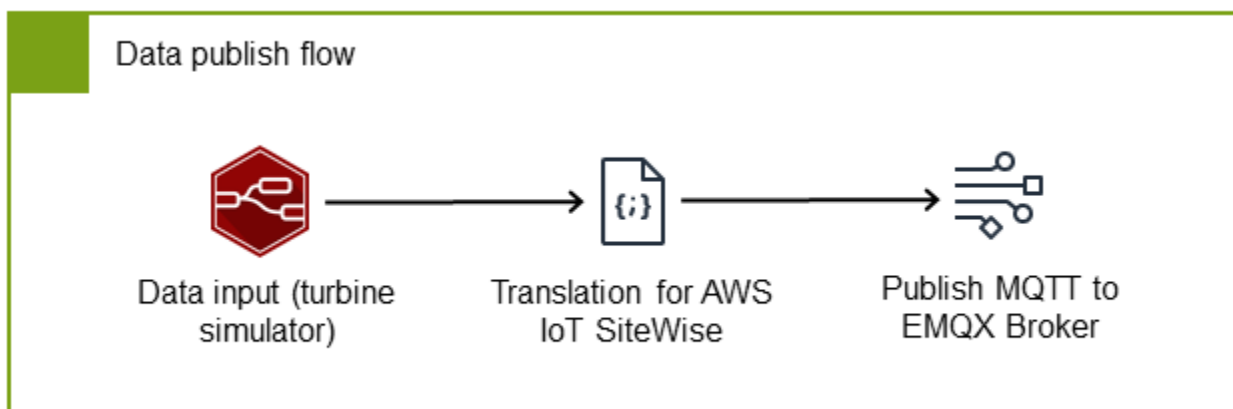
[Per accedere alla console Node-RED, vai su `http://127.0.0.1:1880`](#). Per informazioni sull'attivazione di TLS, consulta [Abilitare](#) la crittografia TLS.

Configura il flusso di pubblicazione dei dati

Il flusso di pubblicazione dei dati utilizza tre nodi per creare una pipeline che invia i dati industriali al cloud. Questo flusso è essenziale per consentire l'analisi basata sul cloud, l'archiviazione a lungo

termine e l'integrazione con altri servizi. AWS Innanzitutto, i dati simulati del dispositivo vengono inviati al broker SiteWise Edge MQTT. Il gateway raccoglie i dati dal broker e li trasmette al AWS IoT SiteWise cloud, dove è possibile sfruttare potenti funzionalità di analisi e visualizzazione.

- Inserimento dati: riceve i dati del dispositivo dalle apparecchiature industriali o dai simulatori
- Traduttore di dati per AWS IoT SiteWise: traduce i dati in AWS IoT SiteWise formato per garantire la compatibilità con il gateway Edge SiteWise
- Editore MQTT: pubblica i dati sul broker SiteWise Edge MQTT, rendendoli disponibili sia agli utenti locali che a quelli cloud



Configura il nodo di immissione dei dati

In questo esempio, il nodo di immissione dei dati utilizza un dispositivo simulato di turbina eolica che genera dati sulla velocità del vento. Questo nodo funge da punto di ingresso per i dati industriali, indipendentemente dal fatto che provengano da fonti simulate (come nel nostro esempio) o da apparecchiature industriali reali negli ambienti di produzione.

Utilizziamo un formato JSON personalizzato per il payload di dati per fornire una struttura standardizzata che funzioni in modo efficiente sia con gli strumenti di elaborazione locali che con il servizio cloud. AWS IoT SiteWise Questo formato include metadati essenziali come timestamp e indicatori di qualità oltre ai valori di misurazione effettivi, consentendo una gestione completa dei dati e il monitoraggio della qualità in tutta la pipeline. Importa il nodo di iniezione per ricevere dati simulati in questo formato JSON standardizzato con timestamp, indicatori di qualità e valori.

[Per ulteriori informazioni sul nodo di iniezione Node-RED, consulta la sezione Inject nella Documentazione. Node-RED](#)

Il simulatore di turbine genera dati sulla velocità del vento ogni secondo in questo formato JSON standardizzato:

Example : Payload di dati sulla turbina

```
{
  name: string,          // Property name/identifier
  timestamp: number,     // Epoch time in nanoseconds
  quality: "GOOD" | "UNCERTAIN" | "BAD",
  value: number | string | boolean
}
```

Questo formato offre diversi vantaggi:

- Il name campo identifica la proprietà o la misurazione specifica e consente di tracciare più punti dati dallo stesso dispositivo
- La misurazione timestamp in nanosecondi garantisce un tracciamento preciso del tempo per un'analisi storica accurata
- L'qualityindicatore consente di filtrare e gestire i dati in base alla loro affidabilità
- Il value campo flessibile supporta diversi tipi di dati per adattarsi a diverse uscite di sensori

Example : Nodo di iniezione di un simulatore di turbina

```
[
  {
    "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"
    ]
  ]
}
]

```

Configura un nodo per la traduzione dei dati

Il gateway SiteWise Edge richiede dati in un formato specifico per garantire la compatibilità con il AWS IoT SiteWise cloud. Il nodo Translator è un componente importante che converte i dati di input nel formato di AWS IoT SiteWise payload richiesto. Questa fase di traduzione garantisce che i dati industriali possano essere elaborati, archiviati e successivamente analizzati correttamente nell' AWS IoT SiteWise ambiente cloud.

Standardizzando il formato dei dati in questa fase, si abilita l'integrazione tra i dispositivi perimetrali e il servizio cloud, dove è possibile utilizzare funzionalità di modellazione, analisi e visualizzazione degli asset. Usa questa struttura:

Example : Struttura del payload per l'analisi dei dati SiteWise Edge

```

{
  "propertyAlias": "string",

```

```

"propertyValues": [
  {
    "value": {
      "booleanValue": boolean,
      "doubleValue": number,
      "integerValue": number,
      "stringValue": "string"
    },
    "timestamp": {
      "timeInSeconds": number,
      "offsetInNanos": number
    },
    "quality": "GOOD" | "UNCERTAIN" | "BAD",
  }
]
}

```

Note

Abbinata `propertyAlias` alla tua gerarchia di argomenti MQTT (ad esempio, `/Renton/WindFarm/Turbine/WindSpeed`). Ciò garantisce che i dati siano associati correttamente alla proprietà corretta dell'asset in AWS IoT SiteWise. Per ulteriori informazioni, consulta il concetto «Alias del flusso di dati» in [AWS IoT SiteWise concetti](#).

1. Importa il nodo funzionale di esempio per la traduzione del AWS IoT SiteWise payload. Questa funzione gestisce la conversione dal formato di input standardizzato al formato AWS IoT SiteWise compatibile, garantendo la corretta formattazione del timestamp, gli indicatori di qualità e la digitazione dei valori.

```

[
  {
    "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
\",
    \"boolean\": (x) => \"booleanValue\",
    \"string\": (x) =>
    \"stringValue\",
  }\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\": [
    [
      \"string\"
    ]
  ]
}
]

```

2. Verificate che il JavaScript codice traduca correttamente i dati sulla velocità del vento. La funzione svolge diverse attività importanti:

- Estrae il nome della proprietà dall'input e lo imposta come PropertyAlias
- Converte il timestamp da millisecondi al formato richiesto di secondi e nanosecondi
- Conserva l'indicatore di qualità per il monitoraggio dell'affidabilità dei dati
- Rileva automaticamente il tipo di valore e lo formatta in base ai requisiti AWS IoT SiteWise

3. Connetti il nodo al flusso, collegandolo tra il nodo di input dei dati e l'editore MQTT.

Per indicazioni sulla scrittura di una funzione specifica per le vostre esigenze aziendali, consultate [Writing Functions](#) nella documentazione Node-RED

Configurare l'editore MQTT

Dopo la traduzione, i dati sono pronti per la pubblicazione sul broker SiteWise Edge MQTT.

Configura l'editore MQTT con queste impostazioni per inviare dati al broker SiteWise Edge MQTT:

Per importare il nodo di uscita MQTT

1. Importa un nodo di configurazione in uscita MQTT utilizzando. "type": "mqtt out" I nodi di uscita MQTT consentono di condividere la configurazione di un broker.
2. Inserite le coppie chiave-valore per le informazioni relative alla connessione del broker MQTT e al routing dei messaggi.

Importa il nodo di esempio. 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": ""  
  }  
]
```

Il nodo di uscita MQTT di esempio crea la connessione MQTT con le seguenti informazioni:

- Server: 127.0.0.1
- Porta: 1883
- Protocollo: MQTT V5

Quindi, il nodo di uscita MQTT configura il routing dei messaggi con le seguenti informazioni:

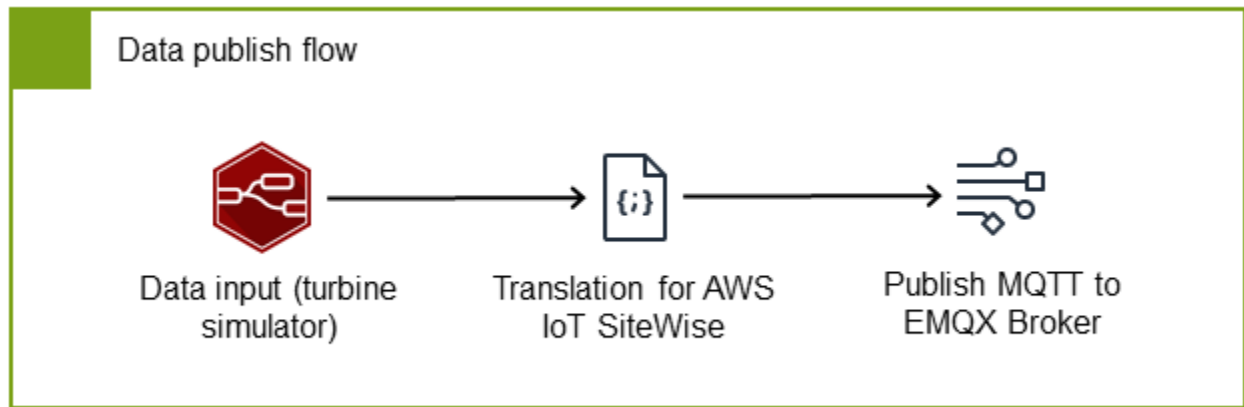
- Argomento: /Renton/WindFarm/Turbine/WindSpeed
- QoS: 1

Implementa e verifica i nodi

Dopo aver configurato i tre nodi del flusso di pubblicazione dei dati, segui questi passaggi per distribuire il flusso e verificare che i dati vengano trasmessi correttamente a AWS IoT SiteWise

Per distribuire e verificare le connessioni

1. Connect i tre nodi come mostrato nel flusso di pubblicazione dei dati.



2. Scegli Deploy per applicare tutte le modifiche alla connessione dei nodi.
3. Vai alla [AWS IoT SiteWise console](#) e scegli Flussi di dati.
4. Assicurati che il prefisso Alias sia selezionato nel menu a discesa. Quindi, cerca l'alias. / Renton/WindFarm/Turbine/WindSpeed

Se vedi l'alias corretto nella tua ricerca, significa che hai distribuito il flusso e verificato la trasmissione dei dati.

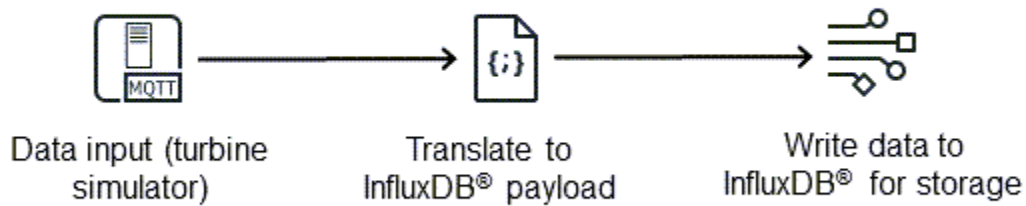
Configura il flusso di conservazione dei dati

Il flusso di conservazione dei dati può essere utilizzato per mantenere la visibilità operativa ai margini. Ciò è utile durante le interruzioni della rete o quando è necessario l'accesso immediato ai dati. Questo flusso si iscrive al broker MQTT per ricevere i dati del dispositivo, li converte in formato InfluxDB® e li archivia localmente. Implementando questo flusso, si crea un archivio dati locale resiliente a cui gli operatori possono accedere senza dipendere dal cloud, che consente il monitoraggio in tempo reale e il processo decisionale a livello perimetrale.

Il flusso è composto da tre componenti chiave che interagiscono per garantire che i dati vengano acquisiti e archiviati correttamente:

- Client di abbonamento MQTT: riceve i dati dal broker, garantendo l'acquisizione di tutti i dati industriali pertinenti
- Traduttore InfluxDB: converte il AWS IoT SiteWise payload in formato InfluxDB, preparando i dati per un'archiviazione efficiente delle serie temporali
- Writer InfluxDB: gestisce l'archiviazione locale, garantendo la persistenza e la disponibilità dei dati per le applicazioni locali

Data retention flow



Configura il client di abbonamento MQTT

- Configura il client di abbonamento MQTT in Node-RED per ricevere dati dal broker MQTT EMQX importando l'esempio seguente. AWS IoT SiteWise

Example : MQTT nel nodo

```
[
  {
    "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": ""
}
]
```

Questo abbonamento garantisce che tutti i dati rilevanti pubblicati sul broker vengano acquisiti per l'archiviazione locale, fornendo una registrazione completa delle vostre operazioni industriali. Il nodo utilizza gli stessi parametri di connessione MQTT della [Configurare l'editore MQTT](#) sezione, con le seguenti impostazioni di abbonamento:

- Argomento — /Renton/WindFarm/Turbine/WindSpeed
- QoS — 1

Per ulteriori informazioni, vedere [Connect to an MQTT Broker](#) nella Node-RED Documentazione.

Configura il traduttore InfluxDB

[InfluxDB organizza i dati utilizzando tag per l'indicizzazione e campi per i valori.](#) Questa organizzazione ottimizza le prestazioni delle query e l'efficienza di archiviazione per i dati delle serie temporali. Importa il nodo funzionale di esempio che contiene il JavaScript codice per convertire il AWS IoT SiteWise payload in formato InfluxDB. Il traduttore divide le proprietà in due gruppi:

- Tag: proprietà di qualità e nome per un'indicizzazione efficiente
- Campi: timestamp (in millisecondi dall'epoca) e valore

Example : nodo funzionale di traduzione in un payload InfluxDB

```
[
  {
    "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(millisecons_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"
      ]
    ]
  }
]
```

Per ulteriori opzioni di configurazione, consulta il [node-red-contrib-influxdb](#) repository Node-RED. GitHub

Configura lo scrittore InfluxDB

Il nodo writer InfluxDB è il componente finale del flusso di conservazione dei dati, responsabile della memorizzazione dei dati industriali nel database locale InfluxDB. Questa archiviazione locale è importante per mantenere la visibilità operativa durante le interruzioni della rete e fornire l'accesso immediato ai dati per applicazioni che richiedono un intervento urgente.

1. Installa il `node-red-contrib-influxdb` pacchetto tramite l'opzione Manage palette. Questo pacchetto fornisce i nodi necessari per connettere Node-RED con InfluxDB.
2. Aggiungi un nodo di uscita InfluxDB al tuo flusso. Questo nodo gestirà l'effettiva scrittura dei dati nel database InfluxDB.
3. Configura le proprietà del server per stabilire una connessione sicura alla tua istanza InfluxDB:
 - a. Imposta la versione su 2.0: specifica che ti stai connettendo a InfluxDB v2.x, che utilizza un'API diversa rispetto alle versioni precedenti
 - b. Imposta l'URL su: rimanda alla tua `http://127.0.0.1:8086` istanza locale di InfluxDB
 - c. Inserisci il tuo token di autenticazione InfluxDB. Questo token sicuro autorizza la connessione al tuo database. Il token è stato generato durante la [Configura l'archiviazione locale con InfluxDB](#) procedura.
4. Specificate i parametri della posizione di archiviazione per definire dove e come verranno archiviati i dati:
 - a. Inserisci il nome della tua organizzazione InfluxDB: l'organizzazione è uno spazio di lavoro per un gruppo di utenti, a cui appartengono i bucket e le dashboard. [Per ulteriori informazioni, consulta Gestire le organizzazioni in. InfluxData Documentation](#)
 - b. Specificate il bucket InfluxDB (ad esempio, `WindFarmData`): il bucket è equivalente a un database nei sistemi tradizionali e funge da contenitore per i dati delle serie temporali
 - c. Imposta la misurazione InfluxDB (ad esempio, `TurbineData`): la misurazione è simile a una tabella nei database relazionali, che organizza i punti dati correlati

Note

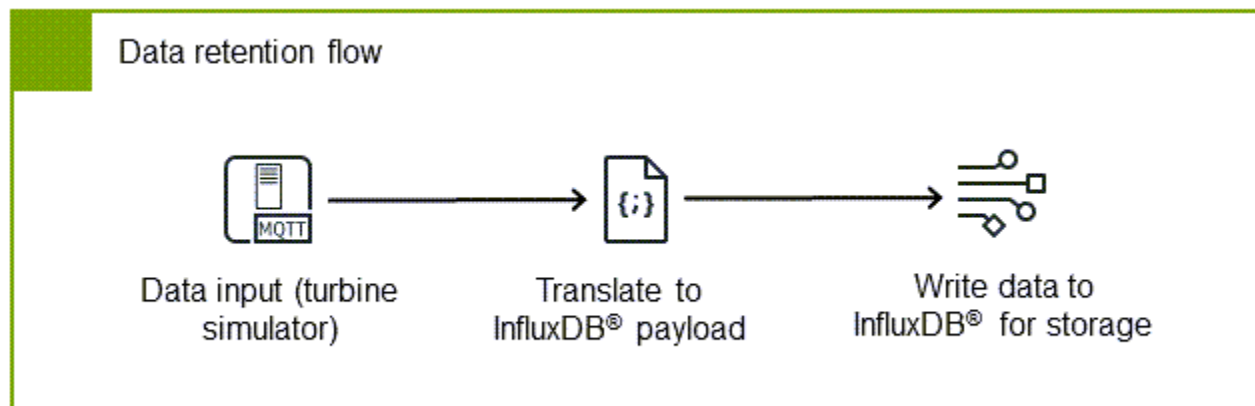
Trova il nome della tua organizzazione nella barra laterale sinistra dell'istanza InfluxDB. I concetti di organizzazione, bucket e misurazione sono fondamentali per il modello di

organizzazione dei dati di InfluxDB. [Per ulteriori informazioni, consulta la documentazione di InfluxDB.](#)

Implementa e verifica il flusso di conservazione

Dopo aver configurato tutti i componenti del flusso di conservazione dei dati, è necessario implementare e verificare che il sistema funzioni correttamente. Questa verifica garantisce che i dati industriali vengano archiviati correttamente a livello locale per l'accesso e l'analisi immediati.

1. Connect i tre nodi come illustrato nel diagramma del flusso di conservazione dei dati. Questo crea una pipeline completa dalla sottoscrizione dei dati all'archiviazione locale.



2. Scegli Deploy per applicare le modifiche e attivare il flusso. Questo avvia il processo di raccolta e archiviazione dei dati.
3. Usa InfluxDB Data Explorer per interrogare e visualizzare i tuoi dati. Questo strumento consente di verificare che i dati vengano archiviati correttamente e di creare visualizzazioni iniziali dei dati delle serie temporali.

In Data Explorer, dovreste essere in grado di vedere le misurazioni della velocità del vento registrate nel tempo, a conferma del corretto funzionamento dell'intera pipeline, dalla generazione dei dati all'archiviazione locale.

Per ulteriori informazioni, vedere [Query in Data Explorer](#) in InfluxData Documentation

Con l'implementazione sia del flusso di pubblicazione dei dati che del flusso di conservazione dei dati, ora disponi di un sistema completo che invia i dati al AWS IoT SiteWise cloud mantenendo una copia locale per un accesso e una resilienza immediati. Questo approccio a doppio percorso ti

garantisce di ottenere i vantaggi dell'analisi e dello storage basati sul cloud mantenendo al contempo la visibilità operativa all'edge.

Configura Grafana per Edge SiteWise

Grafana® ti consente di creare dashboard di monitoraggio locali in tempo reale per i tuoi dati industriali. Visualizzando i dati archiviati in InfluxDB®, è possibile fornire agli operatori informazioni immediate sulle prestazioni delle apparecchiature, sull'efficienza dei processi e sui potenziali problemi. Questa visibilità ai margini è importante per le operazioni urgenti e per mantenere la continuità durante le interruzioni della rete.

Configura la fonte di dati

Il collegamento di Grafana al database InfluxDB crea un potente livello di visualizzazione per i dati industriali. Questa connessione consente dashboard di monitoraggio in tempo reale che gli operatori possono utilizzare per prendere decisioni informate senza dipendere dal cloud.

1. Accedi alla tua istanza Grafana localmente accedendo a <http://127.0.0.1:3000> nel tuo browser. Se è necessario abilitare TLS, puoi fare riferimento a [Configurare Grafana HTTPS per il traffico web sicuro](#) nella Grafana Labs Documentazione.
2. Aggiungi una fonte di dati InfluxDB che punti al bucket di serie temporali InfluxDB in cui Node-RED scrive i dati. Ad esempio, WindFarmData. Questa connessione stabilisce il collegamento tra i dati memorizzati e la piattaforma di visualizzazione.
3. Per istruzioni dettagliate, consulta [Configurare l'origine dati InfluxDB](#) nella Documentazione. Grafana Labs

Crea una dashboard Grafana per SiteWise i dati Edge

La creazione di una dashboard è il passaggio finale nella creazione di una soluzione di monitoraggio locale. I dashboard forniscono rappresentazioni visive dei dati industriali, semplificando l'identificazione di tendenze, anomalie e potenziali problemi a colpo d'occhio.

- Segui la guida per creare una dashboard. Per ulteriori informazioni, consulta [Crea la tua prima dashboard](#) nella Grafana Labs Documentazione. Questo modello presuppone che il bucket abbia un nome WindFarmData e che la misurazione sia. TurbineData

Puoi anche utilizzare la guida rapida importando il modello di dashboard di esempio fornito per creare rapidamente una dashboard con un grafico delle serie temporali per i dati generati da

Node-RED nella sezione precedente. Questo modello fornisce un punto di partenza che puoi personalizzare per soddisfare le tue esigenze di monitoraggio specifiche.

```
{
  "__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": ""
}

```

Configura integrazioni open source con Docker (Linux)

Per un processo di implementazione semplificato, puoi utilizzare Docker per configurare Node-RED®, InfluxDB® e Grafana® su un ambiente Linux. Questo metodo utilizza contenitori preconfigurati, che consentono un'implementazione rapida e una gestione più semplice dei componenti.

Prerequisiti di configurazione di Docker

Prima di iniziare, verifica di avere quanto segue:

- Un gateway V3 abilitato per MQTT. Per ulteriori informazioni, consulta [Gateway V3 compatibili con MQTT per Edge AWS IoT SiteWise](#).
- Il plugin Docker Compose. Per i passaggi di installazione, consulta [Installare il Docker Compose plug-in](#) nella documentazione dei manuali Docker.

Distribuisci i servizi

Questa implementazione esegue SiteWise Edge, InfluxDB, Node-RED e Grafana sullo stesso host.

Configura l'ambiente

1. Ottieni l'accesso root:

```
sudo -i
```

2. Crea un file.env o esporta queste variabili di ambiente:

```
export INFLUXDB_PASSWORD=your-secure-influxdb-password  
export INFLUXDB_TOKEN=your-secure-influxdb-token  
export GRAFANA_PASSWORD=your-secure-grafana-password
```

Configura la rete Docker

- Crea una rete bridge usando il nome SiteWiseEdgeNodeRedDemoNetwork.

```
docker network create --driver=bridge SiteWiseEdgeNodeRedDemoNetwork
```

Aggiorna la distribuzione SiteWise Edge

1. Passa alla [AWS IoT console](#)
2. Scegli Dispositivi Greengrass nel menu di navigazione a sinistra nella sezione Gestisci, quindi Dispositivi principali.
3. Seleziona il dispositivo principale collegato al tuo SiteWise Edge Gateway.
4. Scegli la scheda Distribuzioni, quindi seleziona il valore Deployment ID.
5. Scegli Azioni, quindi seleziona Rivedi.
6. Leggi il messaggio pop-up, quindi scegli Revise Deployment.
7. Nella Fase 2 - Seleziona i componenti, seleziona i seguenti componenti e poi scegli Avanti.
 - `aws.greengrass.clientdevices.mqtt.EMQX`
 - `aws.iot.SiteWiseEdgePublisher`
8. Nella Fase 3 - Configurazione dei componenti, selezionate il valore del `aws.greengrass.clientdevices.mqtt.EMQX` componente e aggiungete la seguente configurazione di rete:

```
{
  "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. Scegli Next (Successivo).
10. Nel Passaggio 4 - Configurazione delle impostazioni avanzate, scegli Avanti.

11. Scegli Deploy

Avvia i servizi

1. Avvia i servizi utilizzando il file Docker Compose. Esegui il comando seguente nella directory contenente il `compose.yaml` file.

```
docker compose up -d
```

2. Crea un tunnel SSH per accedere ai servizi:

```
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
```

Questa distribuzione crea i seguenti servizi in: `SiteWiseEdgeNodeRedDemoNetwork` network

InfluxDB v2 (porta 8086)

Include organizzazione preconfigurata (`iot-sitewise-edge`), bucket `WindFarmData` InfluxDB e credenziali di amministratore

Node-red (porta 1880)

Include nodi InfluxDB e flussi preconfigurati per l'integrazione AWS IoT SiteWise

Grafana (porto 3000)

Include utente amministratore, origine dati InfluxDB e dashboard di monitoraggio

Accedi ai servizi

Dopo la distribuzione, accedi ai servizi utilizzando le seguenti URLs credenziali:

Note

È possibile accedere a ciascun servizio dal proprio host o dal computer gateway.

Dettagli di accesso al servizio

Servizio	URL	Credenziali
Node-red	http://127.0.0.1:1880	Non sono richieste credenziali
InfluxDB	http://127.0.0.1:8086	Nome utente: admin Password: \$INFLUXDB_PASSWORD
Grafana	http://127.0.0.1:3000	Nome utente: admin Password: \$GRAFANA_PASSWORD

Verifica della distribuzione

Per assicurarti che la distribuzione abbia esito positivo, esegui i seguenti controlli:

1. Per Node-RED, verifica la presenza di due flussi precaricati:
 - Flusso di pubblicazione dei dati
 - Flusso di conservazione dei dati
2. Infatti AWS IoT SiteWise, nella AWS IoT SiteWise console, conferma la presenza di un flusso di dati con l'alias/Renton/WindFarm/Turbine/WindSpeed.
3. Per InfluxDB, usa Data Explorer per verificare la memorizzazione dei dati nella TurbineData misurazione all'interno del bucket. WindFarmData
4. Per Grafana, visualizza la dashboard per confermare la visualizzazione dei dati delle serie temporali generati da Node-RED.

Elabora i dati per le integrazioni open source

I dati possono essere elaborati (ad esempio trasformazione o aggregazione), in fasi diverse utilizzando vari strumenti, ognuno dei quali soddisfa requisiti di monitoraggio diversi.

Elabora i dati con i nodi Node-RED

Trasforma i tuoi dati in tempo reale utilizzando i nodi di elaborazione integrati Node-RED®. Configura questi nodi tramite la console Node-RED per creare la tua pipeline di dati.

Nodi di trasformazione dei dati

Trasforma singoli punti dati, in modo simile a Transforms in AWS IoT SiteWise, utilizzando questi nodi:

- **change node:** esegue semplici modifiche di valore sui dati.
- **function node:** consente JavaScript trasformazioni personalizzate per l'elaborazione di dati complessi.

Nodi di calcolo delle metriche

Combina più punti dati in un unico output, in modo simile a Metrics in AWS IoT SiteWise, utilizzando questi nodi:

- **nodo batch:** raggruppa più messaggi per l'elaborazione in batch.
- **join node:** combina più flussi di dati in un unico output.
- **nodo aggregatore:** calcola le metriche aggregate da più punti dati.

[Per ulteriori opzioni di nodo, consulta la libreria Node-RED.](#)

Crea attività InfluxDB

Sebbene Node-RED eccella nell'elaborazione dei dati di base con una configurazione rapida, i calcoli metrici complessi possono diventare difficili nella programmazione basata sul flusso. InfluxDB® Tasks fornisce un'alternativa tramite script Flux pianificati per esigenze di elaborazione avanzate.

Usa InfluxDB Tasks per:

- Aggregazioni statistiche su set di dati di grandi dimensioni
- Operazioni matematiche su più proprietà
- Misurazioni derivate da più fonti

Caratteristiche dell'attività

- Esecuzione pianificata: esegue attività basate su espressioni cron
- Elaborazione in batch: ottimizza le operazioni per i dati di serie temporali
- Ripristino degli errori: riprova automaticamente le operazioni non riuscite
- Monitoraggio: monitora l'esecuzione tramite registri dettagliati

Gestisci le attività tramite l'interfaccia utente, l'API o la CLI di InfluxDB. Per ulteriori informazioni, consulta [Elaborare i dati con](#) le attività di InfluxDB.

Usa le trasformazioni Grafana

Trasforma la visualizzazione dei dati in Grafana® senza modificare i dati di origine in InfluxDB. Le trasformazioni Grafana si applicano solo al livello di visualizzazione.

- Visual Builder: crea trasformazioni senza scrivere codice
- Anteprima dal vivo: visualizza i risultati della trasformazione in tempo reale
- Multi-Source: elabora i dati da più fonti di database
- Archiviazione efficiente: trasforma i dati al momento della visualizzazione senza archiviare risultati intermedi

Per ulteriori informazioni, consulta [Transform](#) data.

Risoluzione dei problemi relativi alle integrazioni open source

Per ulteriori informazioni sulla risoluzione dei problemi relativi alle integrazioni open source per i gateway SiteWise Edge, consulta. [Risoluzione dei problemi di integrazioni open source su Edge](#)

Stream classici, gateway V2 per Edge AWS IoT SiteWise

Comprendi le caratteristiche e i limiti degli stream Classic, dei gateway V2 per Edge. AWS IoT SiteWise

Il gateway Classic Streams, V2 mantiene le funzionalità tradizionali utilizzate nelle AWS IoT SiteWise implementazioni precedenti prima dell'introduzione dei gateway V3 abilitati per MQTT. Questi gateway SiteWise Edge sono considerati stream Classic, gateway V2. Mantengono la compatibilità con le versioni precedenti e sono funzionali con il pacchetto di elaborazione dati. Sebbene il

gateway Classic Streams, V2 offre prestazioni affidabili per le configurazioni esistenti, presenta delle limitazioni rispetto alle opzioni di gateway più recenti. In particolare, questo tipo di gateway non è completamente compatibile con le funzionalità avanzate disponibili nella destinazione gateway V3 abilitata per MQTT. Per utilizzare il protocollo di messaggistica MQTT, è possibile creare un nuovo gateway V3 abilitato per MQTT. Per ulteriori informazioni, consulta [Gateway V3 compatibili con MQTT per Edge AWS IoT SiteWise](#).

Argomenti

- [Utilizzate i pacchetti per raccogliere ed elaborare dati in Edge SiteWise](#)
- [Configura il componente publisher AWS IoT SiteWise](#)
- [Destinazioni e gestore di stream AWS IoT Greengrass](#)
- [Configura le funzionalità edge su AWS IoT SiteWise Edge](#)
- [Configura l'elaborazione dei dati edge per AWS IoT SiteWise modelli e asset](#)

Utilizzate i pacchetti per raccogliere ed elaborare dati in Edge SiteWise

AWS IoT SiteWise Gli edge gateway utilizzano diversi pacchetti per determinare come raccogliere ed elaborare i dati.

Attualmente sono disponibili i seguenti pacchetti:

- **Pacchetto di raccolta dati:** utilizza questo pacchetto per raccogliere i dati industriali e indirizzarli verso destinazioni AWS cloud. Per impostazione predefinita, questo pacchetto è abilitato automaticamente per il gateway SiteWise Edge.
- **Pacchetto di elaborazione dati:** utilizza questo pacchetto per abilitare la comunicazione tramite gateway SiteWise Edge con modelli e asset di asset configurati dall'edge. È possibile utilizzare la configurazione edge per controllare quali dati degli asset elaborare ed elaborare in loco. È quindi possibile inviare i dati a AWS IoT SiteWise o ad altri AWS servizi. Per ulteriori informazioni sul pacchetto di elaborazione dati, vedere [the section called “Configura l'elaborazione dei dati edge”](#).

Pacchetti di aggiornamento

Important

L'aggiornamento delle versioni dei pacchetti di elaborazione dati precedenti (e incluse) 2.0.x alla versione 2.1.x comporterà la perdita dei dati delle misurazioni archiviate localmente.

SiteWise Gli edge gateway utilizzano pacchetti diversi per determinare come raccogliere ed elaborare i dati. È possibile utilizzare la AWS IoT SiteWise console per aggiornare i pacchetti.

Per aggiornare i pacchetti (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nella barra di navigazione a sinistra, scegli Edge gateway nella sezione Edge.
3. Nell'elenco dei gateway, scegli il gateway SiteWise Edge con i pacchetti che desideri aggiornare.
4. Nella sezione Configurazione del gateway, scegli Aggiornamenti software disponibili.
5. Nella pagina Modifica versioni del software, scegli Aggiornamenti.

 Note

È possibile aggiornare solo i pacchetti abilitati. Per trovare l'elenco dei pacchetti abilitati per questo gateway SiteWise Edge, scegli Panoramica, quindi consulta la sezione Funzionalità Edge.

6. Nella pagina di modifica delle versioni del software, nella sezione Aggiornamenti dei componenti del Gateway, procedi come segue:
 - Per aggiornare il collettore OPC UA, scegli una versione, quindi scegli Deploy.
 - Per aggiornare Publisher, scegli una versione, quindi scegli Deploy.
 - Per aggiornare il pacchetto di elaborazione dati, scegli una versione, quindi scegli Distribuisci.
7. Quando hai finito di distribuire nuove versioni, scegli Fine.

Se riscontri problemi durante l'aggiornamento dei pacchetti, consulta [Impossibile distribuire i pacchetti sui gateway Edge SiteWise](#)

Configura il componente publisher AWS IoT SiteWise

Dopo aver creato un gateway AWS IoT SiteWise Edge e installato il software, puoi configurare il componente publisher in modo che il gateway SiteWise Edge possa esportare i dati nel AWS Cloud. Utilizza il componente publisher per abilitare funzionalità aggiuntive o configurare le impostazioni predefinite. Per ulteriori informazioni, consulta [AWS IoT SiteWise Publisher](#) nella AWS IoT Greengrass Version 2 Developer Guide.

 Note

La configurazione dell'editore varia in base al tipo di gateway utilizzato. Per i gateway Classic Stream, V2, usa lo spazio dei nomi. `iotsitewise:publisher:2` Per i gateway V3 abilitati per MQTT, utilizzate lo spazio dei nomi. `iotsitewise:publisher:3`

Console

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Edge gateway.
3. Seleziona il gateway SiteWise Edge per il quale desideri configurare l'editore.
4. Nella sezione Configurazione di Publisher, scegli Modifica
5. Per Ordine di pubblicazione, scegliete una delle seguenti opzioni:
 - Pubblica prima i dati più vecchi: per impostazione predefinita, il gateway SiteWise Edge pubblica prima i dati più vecchi sul cloud.
 - Pubblica prima i dati più recenti: il gateway SiteWise Edge pubblica prima i dati più recenti sul cloud.
6. (Facoltativo) Se non desideri che il gateway SiteWise Edge comprima i tuoi dati, deseleziona Attiva la compressione durante il caricamento dei dati.
7. (Facoltativo) Se non desideri pubblicare vecchi dati, scegli Escludi dati scaduti e procedi come segue:
 - Per Periodo limite, inserisci un valore e scegli un'unità. Il periodo limite deve essere compreso tra cinque minuti e sette giorni. Ad esempio, se il periodo limite è di tre giorni, i dati più vecchi di tre giorni non vengono pubblicati sul cloud.
8. (Facoltativo) Per configurare impostazioni personalizzate sulla gestione dei dati sul dispositivo locale, scegli Impostazioni di archiviazione locale ed esegui le seguenti operazioni:
 - a. Per Periodo di conservazione, inserisci un numero e scegli un'unità. Il periodo di conservazione deve essere compreso tra un minuto e 30 giorni e deve essere superiore o uguale al periodo di rotazione. Ad esempio, se il periodo di conservazione è di 14 giorni, il gateway SiteWise Edge elimina tutti i dati sull'edge che sono più vecchi del periodo limite specificato dopo che sono stati archiviati per 14 giorni.

- b. Per Periodo di rotazione, inserisci un numero e scegli un'unità. Il periodo di rotazione deve essere superiore a un minuto e pari o inferiore al periodo di conservazione. Ad esempio, supponiamo che il periodo di rotazione sia di due giorni, il gateway SiteWise Edge raggruppa e salva i dati più vecchi del periodo limite in un unico file. Per i gateway ospitati autonomamente AWS IoT Greengrass V2, il gateway SiteWise Edge trasferisce un batch di dati nella seguente directory locale una volta ogni due giorni: `/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports`
- c. Per Capacità di archiviazione, inserisci un valore maggiore o uguale a 1. Se la capacità di archiviazione è di 2 GB, il gateway SiteWise Edge inizia a eliminare i dati quando più di 2 GB di dati vengono archiviati localmente.

9. Scegli Save (Salva).

AWS CLI

Utilizza l'[UpdateGatewayCapabilityConfiguration](#) API per configurare l'editore.

Imposta il parametro `capabilityNamespace` su `iotsitewise:publisher:2`.

Example : configurazione Publisher per i gateway Classic Stream, V2

Lo spazio dei nomi dell'editore: `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
        }
    }
]
}
}

```

L'editore fornisce i seguenti parametri di configurazione che è possibile personalizzare:

SiteWisePublisherConfiguration

publishingOrder

L'ordine in cui i dati vengono pubblicati nel cloud. Il valore di questo parametro può essere uno dei seguenti:

- **TIME_ORDER**(Pubblica prima i dati più vecchi): per impostazione predefinita, i dati più vecchi vengono pubblicati per primi nel cloud.
- **RECENT_DATA**(Pubblica prima i dati più recenti): i dati più recenti vengono pubblicati prima nel cloud.

enableCompression

Imposta questa opzione per `true` comprimere i dati prima della pubblicazione. La compressione dei dati può ridurre l'utilizzo della larghezza di banda.

dropPolicy

(Facoltativo) Una policy che controlla quali dati vengono pubblicati sul cloud.

cutoffAge

L'età massima dei dati da pubblicare è specificata in giorni, ore e minuti. Ad esempio `7d` o `1d7h16m`. I dati più vecchi di quelli specificati non vengono inviati a AWS IoT SiteWise.

I dati precedenti al periodo limite non vengono pubblicati nel cloud. L'età limite deve essere compresa tra cinque minuti e sette giorni.

È possibile utilizzare mh, e d quando si specifica un'età limite. Nota che m rappresenta i minuti, h rappresenta le ore e d rappresenta i giorni.

`exportPolicy`

(Facoltativo) Una policy che gestisce l'archiviazione dei dati all'edge. Questa politica si applica ai dati precedenti alla data limite.

`retentionPeriod`

Il gateway SiteWise Edge elimina tutti i dati sull'edge che sono precedenti al periodo limite dall'archiviazione locale dopo che sono stati archiviati per il periodo di conservazione specificato. Il periodo di conservazione deve essere compreso tra un minuto e 30 giorni e deve essere superiore o uguale al periodo di rotazione.

È possibile utilizzare mh, e d quando si specifica un periodo di conservazione. Nota che m rappresenta i minuti, h rappresenta le ore e d rappresenta i giorni.

`rotationPeriod`

L'intervallo di tempo durante il quale raggruppare e salvare i dati precedenti al periodo limite in un singolo file. Il gateway SiteWise Edge trasferisce un batch di dati nella seguente directory locale alla fine di ogni periodo di rotazione: `./greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports` Il periodo di rotazione deve essere maggiore di un minuto, e uguale o inferiore al periodo di conservazione.

È possibile utilizzare mh, e d quando si specifica un periodo di rotazione. Nota che m rappresenta i minuti, h rappresenta le ore e d rappresenta i giorni.

`exportSizeLimitGB`

La dimensione massima consentita per i dati archiviati localmente, in GB. Se questa quota viene violata, il gateway SiteWise Edge inizia a eliminare i primi dati fino a quando la dimensione dei dati archiviati localmente non è uguale o inferiore alla quota. Il valore di questo parametro deve essere maggiore o uguale a 1.

`SiteWiseS3PublisherConfiguration`

`accessRoleArn`

Il ruolo di accesso che AWS IoT SiteWise autorizza la gestione del bucket Amazon S3 su cui stai pubblicando.

streamToS3ConfigMapping

Una serie di configurazioni che associa uno stream a una configurazione Amazon S3.

streamName

Lo stream da cui leggere e pubblicare nella configurazione di Amazon S3.

targetBucketArn

Il bucket in ARN cui pubblicare.

publishPolicy

publishFrequency

La frequenza con cui il gateway SiteWise Edge pubblica nel bucket Amazon S3.

localSizeLimitGB

La dimensione massima dei file scritti sul disco locale. Se questa soglia viene superata, l'editore pubblica tutti i dati memorizzati nel buffer nella relativa destinazione.

siteWiseImportPolicy

enableSiteWiseStorageImport

Imposta questa opzione `true` su per importare i dati da un bucket Amazon S3 allo storage. AWS IoT SiteWise

enableDeleteAfterImport

Imposta questa opzione `true` su per eliminare il file nel bucket Amazon S3 dopo l'ingestione nello storage. AWS IoT SiteWise

Destinazioni e gestore di stream AWS IoT Greengrass

AWS IoT Greengrass stream manager consente di inviare dati alle seguenti Cloud AWS destinazioni: canali in entrata AWS IoT Analytics, flussi in Amazon Kinesis Data Streams, proprietà delle risorse o oggetti AWS IoT SiteWise in Amazon Simple Storage Service (Amazon S3). Per ulteriori informazioni, consulta [Manage data stream on the AWS IoT Greengrass](#) Core nella Developer Guide. AWS IoT Greengrass Version 2

Example : struttura dei messaggi del flusso di dati

L'esempio seguente mostra la struttura dei messaggi del flusso di dati richiesta trasmessa dallo AWS IoT Greengrass stream manager.

```
{
  "assetId": "string",
  "propertyAlias": "string",
  "propertyId": "string",
  "propertyValues": [
    {
      "quality": "string",
      "timestamp": {
        "offsetInNanos": number,
        "timeInSeconds": number
      },
      "value": {
        "booleanValue": boolean,
        "doubleValue": number,
        "integerValue": number,
        "stringValue": "string"
      }
    }
  ]
}
```

Note

Il messaggio del flusso di dati deve includere (assetIdpropertyId) o propertyAlias nella sua struttura.

assetId

(Facoltativo) L'ID della risorsa da aggiornare.

propertyAlias

(Facoltativo) L'alias che identifica la proprietà, ad esempio il percorso del flusso di dati del server OPC UA. Per esempio:

```
/company/windfarm/3/turbine/7/temperature
```

Per ulteriori informazioni, consulta [Gestire i flussi di dati](#) nella Guida per l'utente.AWS IoT SiteWise

propertyId

(Facoltativo) L'ID della proprietà dell'asset per questa voce.

propertyValues

(Obbligatorio) L'elenco dei valori delle proprietà da caricare. È possibile specificare fino a 10 elementi `propertyValues` dell'array.

quality

(Facoltativo) La qualità del valore della proprietà dell'asset.

timestamp

(Obbligatorio) Il timestamp del valore della proprietà dell'asset.

offsetInNanos

(Facoltativo) L'offset in nanosecondi da. `timeInSeconds`

timeInSeconds

(Obbligatorio) La data del timestamp, in secondi, nel formato Unix epoch. I dati frazionari in nanosecondi sono forniti da. `offsetInNanos`

value

(Obbligatorio) Il valore della proprietà dell'asset.

Note

Nel `value` campo può esistere solo uno dei seguenti valori.

booleanValue

(Facoltativo) Dati della proprietà Asset di tipo Boolean (`true`o`false`).

doubleValue

(Facoltativo) Dati relativi alla proprietà dell'asset di tipo double (numero a virgola mobile).

integerValue

(Facoltativo) Dati relativi alla proprietà dell'asset di tipo intero (numero intero).

stringValue

(Facoltativo) Dati delle proprietà dell'asset di tipo string (sequenza di caratteri).

Configura le funzionalità edge su AWS IoT SiteWise Edge

È possibile utilizzare AWS IoT SiteWise Edge per raccogliere e archiviare temporaneamente i dati in modo da poter organizzare ed elaborare i dati del dispositivo localmente. Abilitando l'elaborazione edge, puoi scegliere di inviare solo dati aggregati per ottimizzare l'utilizzo della larghezza Cloud AWS di banda e i costi di archiviazione nel cloud. Utilizzando AWS IoT SiteWise components with AWS IoT Greengrass, puoi raccogliere ed elaborare i dati sull'edge prima di inviarli a Cloud AWS, oppure gestirli in locale utilizzando Edge. SiteWise APIs

La raccolta dei dati avviene tramite pacchetti di dati e AWS IoT SiteWise componenti eseguiti su AWS IoT Greengrass.

Note

- AWS IoT SiteWise conserva i dati edge sui gateway SiteWise Edge per un massimo di 30 giorni. Il periodo di conservazione dei dati dipende dallo spazio disponibile su disco del dispositivo.
- Se il gateway SiteWise Edge è stato disconnesso da Cloud AWS per 30 giorni, il [Data Processing Pack](#) viene disattivato automaticamente.

Argomenti

- [Configura la funzionalità edge in SiteWise Edge](#)

Configura la funzionalità edge in SiteWise Edge

AWS IoT SiteWise fornisce i seguenti pacchetti che il gateway SiteWise Edge può utilizzare per determinare come raccogliere ed elaborare i dati. Seleziona i pacchetti per abilitare le funzionalità edge per il tuo gateway SiteWise Edge.

- Il pacchetto di raccolta dati consente al gateway SiteWise Edge di raccogliere dati da più server OPC UA e quindi di esportarli dall'edge al AWS cloud. Diventa attivo dopo aver aggiunto fonti di dati al gateway SiteWise Edge.
- Il pacchetto di elaborazione dati consente al gateway SiteWise Edge di elaborare i dati delle apparecchiature sull'edge. Ad esempio, puoi utilizzare modelli di asset per calcolare metriche e trasformazioni. Per ulteriori informazioni sui modelli di asset e sugli asset, consulta [Asset industriali modello](#)

Note

- Il pacchetto di elaborazione dati è disponibile solo su piattaforme x86.

Per configurare le funzionalità edge

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Edge gateway.
3. Seleziona il gateway SiteWise Edge per il quale desideri attivare le funzionalità edge.
4. Nella sezione Funzionalità Edge, scegli Modifica
5. Nella sezione Funzionalità Edge, seleziona Abilita il pacchetto di elaborazione dati (comporta costi aggiuntivi).
6. (Facoltativo) Nella sezione Connessione LDAP Edge, è possibile concedere ai gruppi di utenti presenti nella directory aziendale l'accesso a questo gateway SiteWise Edge. I gruppi di utenti possono utilizzare le credenziali LDAP (Lightweight Directory Access Protocol) per accedere al SiteWise gateway Edge. Quindi possono utilizzare le AWS OpsHub AWS IoT SiteWise applicazioni, le operazioni AWS IoT SiteWise API o altri strumenti per gestire il gateway SiteWise Edge. Per ulteriori informazioni, consulta [Gestione dei gateway SiteWise Edge](#).

Note

È inoltre possibile utilizzare Linux o Microsoft Windows le credenziali per accedere al gateway SiteWise Edge. Per ulteriori informazioni, consulta [Accedi al tuo gateway SiteWise Edge utilizzando le credenziali del sistema operativo Linux](#).

- a. Seleziona Attivato.

- b. Per Nome del provider, inserisci un nome per il tuo provider LDAP.
- c. Per Nome host o indirizzo IP, inserisci il nome host o l'indirizzo IP del tuo server LDAP.
- d. Per Porta, inserisci un numero di porta.
- e. Per Nome distinto di base (DN), immettere un nome distinto (DN) per la base.

Sono supportati i seguenti tipi di attributi: CommonName (CN), LocalityName (L), Name (ST), stateOrProvince OrganizationName (O), (OU), CountryName organizationalUnitName (C), StreetAddress (STREET), DomainComponent (DC) e userid (UID).

- f. Per il DN del gruppo di amministratori, inserisci un DN.
- g. Per il DN del gruppo di utenti, inserisci un DN.

7. Scegli Save (Salva).

Ora che hai attivato le funzionalità edge sul tuo gateway SiteWise Edge, devi configurare il tuo modello di asset per l'edge. La configurazione edge del modello di asset specifica dove vengono calcolate le proprietà degli asset. È possibile calcolare tutte le proprietà sull'edge oppure configurare le proprietà del modello di asset separatamente. [Le proprietà del modello di asset includono metriche, trasformazioni e misurazioni.](#)

Per ulteriori informazioni sulle proprietà degli asset, consulta [the section called “Definite le proprietà dei dati”](#)

Dopo aver creato il modello di asset, puoi configurarlo per l'edge. Per ulteriori informazioni sulla configurazione del modello di asset per l'edge, consulta [the section called “Create un modello di asset \(console\)”](#).

Note

I modelli di asset e i dashboard vengono sincronizzati automaticamente tra il AWS Cloud e il gateway SiteWise Edge ogni 10 minuti. È inoltre possibile eseguire la sincronizzazione manualmente dall'applicazione gateway SiteWise Edge locale.

Configura l'elaborazione dei dati edge per AWS IoT SiteWise modelli e asset

È possibile utilizzare AWS IoT SiteWise Edge per raccogliere, archiviare, organizzare e monitorare i dati delle apparecchiature a livello locale. È possibile utilizzare SiteWise Edge per modellare i dati

industriali e SiteWise Monitor per creare dashboard in cui il personale operativo possa visualizzare i dati localmente. Puoi elaborare i dati localmente e inviarli al AWS Cloud oppure elaborarli in locale utilizzando l'API. AWS IoT SiteWise

Con AWS IoT SiteWise Edge, puoi elaborare i dati grezzi localmente e scegliere di inviare solo dati aggregati al AWS Cloud per ottimizzare l'utilizzo della larghezza di banda e i costi di archiviazione nel cloud.

Note

- AWS IoT SiteWise conserva i dati edge sui gateway SiteWise Edge per un massimo di 30 giorni. Il periodo di conservazione dei dati dipende dallo spazio disponibile su disco del dispositivo.
- Se il gateway SiteWise Edge è stato disconnesso dal AWS cloud per 30 giorni, [Configura una sorgente OPC UA in SiteWise Edge](#) viene disattivato automaticamente.

Configura un modello di asset per l'elaborazione dei dati su SiteWise Edge

È necessario configurare il modello di asset per l'edge prima di poter elaborare i dati del gateway SiteWise Edge sull'edge. La configurazione edge del modello di asset specifica dove vengono calcolate le proprietà degli asset. Puoi scegliere di calcolare tutte le proprietà sull'edge e inviare i risultati al AWS Cloud oppure personalizzare dove calcolare separatamente le proprietà di ciascuna risorsa. Per ulteriori informazioni, consulta [Configura l'elaborazione dei dati edge per AWS IoT SiteWise modelli e asset](#).

Le proprietà delle risorse includono metriche, trasformazioni e misurazioni:

- Le metriche sono i dati aggregati della risorsa in un periodo di tempo specificato. Puoi calcolare nuove metriche utilizzando i dati metrici esistenti. AWS IoT SiteWise invia sempre le tue metriche al AWS Cloud per l'archiviazione a lungo termine. AWS IoT SiteWise calcola le metriche sul AWS Cloud per impostazione predefinita. Puoi configurare il tuo modello di asset per calcolare le tue metriche all'edge. AWS IoT SiteWise invia i risultati elaborati al Cloud. AWS
- Le trasformazioni sono espressioni matematiche che mappano i punti dati delle proprietà di un asset da un modulo all'altro. Le trasformazioni possono utilizzare le metriche come dati di input e devono essere calcolate e archiviate nella stessa posizione dei relativi input. Se configuri un input metrico per il calcolo sull'edge, calcola AWS IoT SiteWise anche la trasformazione associata sull'edge.

- Le misurazioni sono formattate come dati grezzi che il dispositivo raccoglie e invia al Cloud per impostazione predefinita. AWS Puoi configurare il tuo modello di asset per archiviare questi dati sul tuo dispositivo locale.

Per ulteriori informazioni sulle proprietà degli asset, consulta [the section called “Definite le proprietà dei dati”](#).

Dopo aver creato il modello di asset, puoi configurarlo per l'edge. Per ulteriori informazioni sulla configurazione del modello di asset per l'edge, consulta [the section called “Create un modello di asset \(console\)”](#).

Note

I modelli di asset e i dashboard vengono sincronizzati automaticamente tra il AWS Cloud e il gateway SiteWise Edge ogni 10 minuti. Puoi anche sincronizzare manualmente da [Gestione dei gateway SiteWise Edge](#)

È possibile utilizzare AWS IoT SiteWise REST APIs e AWS Command Line Interface (AWS CLI) per interrogare il gateway SiteWise Edge alla ricerca di dati sull'edge. Prima di interrogare il gateway SiteWise Edge per la ricerca di dati sull'edge, è necessario soddisfare i seguenti prerequisiti:

- Le credenziali devono essere impostate per il REST. APIs Per ulteriori informazioni sull'impostazione delle credenziali, vedere [the section called “Gestisci i gateway”](#)
- L'endpoint SDK deve puntare all'indirizzo IP del gateway Edge. SiteWise Puoi trovare ulteriori informazioni nella documentazione del tuo SDK. Ad esempio, consulta [Specificare gli endpoint personalizzati nella Guida](#) per gli AWS SDK for Java 2.x sviluppatori.
- Il certificato SiteWise Edge Gateway deve essere registrato. Puoi trovare ulteriori informazioni sulla registrazione del certificato SiteWise Edge gateway nella documentazione del tuo SDK. Ad esempio, consulta la [registrazione dei pacchetti di certificati in Node.js nella Guida](#) per gli sviluppatori. AWS SDK for Java 2.x

Per ulteriori informazioni sull'interrogazione dei dati con AWS IoT SiteWise, vedere [Interroga i dati da AWS IoT SiteWise](#)

Aggiungi fonti di dati al tuo gateway AWS IoT SiteWise Edge

Dopo aver configurato un gateway AWS IoT SiteWise Edge, puoi aggiungere e configurare fonti di dati su cui importare dati da apparecchiature industriali locali. AWS IoT SiteWise SiteWise Edge supporta vari protocolli, tra cui OPC UA, e molti altri protocolli disponibili tramite fonti di dati partner. Queste fonti consentono al gateway di connettersi ai server locali e recuperare i dati industriali. Configurando le fonti di dati, è possibile importare dati da diverse fonti di dati e quindi associare i flussi di dati alle proprietà degli asset, consentendo una modellazione completa degli asset industriali e la mappatura dei dati. AWS IoT SiteWise

Argomenti

- [Fonti di dati OPC UA per gateway Edge AWS IoT SiteWise](#)
- [Fonti di dati dei partner sui SiteWise gateway Edge](#)

Fonti di dati OPC UA per gateway Edge AWS IoT SiteWise

Dopo aver configurato un gateway AWS IoT SiteWise Edge, è possibile configurare le fonti di dati in modo che il gateway SiteWise Edge possa importare dati dalle apparecchiature industriali locali verso AWS IoT SiteWise. Ogni fonte rappresenta un server locale, ad esempio un server OPC UA, a cui il gateway SiteWise Edge collega e recupera i flussi di dati industriali. Per ulteriori informazioni sulla configurazione di un gateway SiteWise Edge, vedere. [Crea un gateway Edge ospitato autonomamente SiteWise](#)

Il tipo di gateway, i gateway V3 abilitati per MQTT rispetto ai gateway Classic Stream, V2, influenza il modo in cui vengono gestiti i dati OPC UA. In Classic stream, i gateway V2 e le sorgenti dati OPC UA vengono aggiunte direttamente alla configurazione del gateway SiteWise IoT publisher. Ogni fonte di dati è associata al gateway e il routing dei dati è configurato individualmente per ciascuna fonte. Al contrario, utilizzando gateway V3 abilitati per MQTT, le fonti di dati OPC UA vengono convertite in argomenti MQTT e gestite tramite destinazioni centralizzate. Per ulteriori informazioni su ciascun tipo, vedere e. [Gateway V3 compatibili con MQTT per Edge AWS IoT SiteWise Stream classici, gateway V2 per Edge AWS IoT SiteWise](#)

Note

AWS IoT SiteWise riavvia il gateway SiteWise Edge ogni volta che aggiungi o modifichi una fonte. Il gateway SiteWise Edge non inserirà dati durante l'aggiornamento della configurazione di origine. Il tempo necessario per riavviare il gateway SiteWise Edge dipende

dal numero di tag presenti nelle sorgenti del gateway SiteWise Edge. Il tempo di riavvio può variare da pochi secondi (per un gateway SiteWise Edge con pochi tag) a diversi minuti (per un gateway SiteWise Edge con molti tag).

Dopo aver creato le fonti, puoi associare i flussi di dati alle proprietà delle risorse. Per ulteriori informazioni su come creare e utilizzare gli asset, consulta [Asset industriali modello](#).

Puoi visualizzare le CloudWatch metriche per verificare che una fonte di dati sia connessa. AWS IoT SiteWise Per ulteriori informazioni, consulta [AWS IoT Greengrass Version 2 metriche del gateway](#).

Attualmente, AWS IoT SiteWise supporta i seguenti protocolli di origine dati:

- [OPC UA](#) — Protocollo di comunicazione A machine-to-machine (M2M) per l'automazione industriale.

Support per protocolli industriali aggiuntivi

SiteWise Edge supporta un'ampia gamma di protocolli industriali attraverso l'integrazione con partner di fonti di dati. Queste partnership consentono la connettività con oltre 200 protocolli diversi, compatibili con vari sistemi e dispositivi industriali.

Per un elenco dei partner disponibili per le fonti di dati, consulta. [SiteWise Opzioni di origine dati dei partner Edge Gateway](#)

Configura una sorgente OPC UA in SiteWise Edge

È possibile utilizzare la AWS IoT SiteWise console o la funzionalità di un gateway SiteWise Edge per definire e aggiungere una sorgente OPC UA al gateway SiteWise Edge per rappresentare un server OPC UA locale.

Argomenti

- [Configura una sorgente OPC UA \(console\)](#)
- [Configurare una fonte OPC UA \(AWS CLI\)](#)

Configura una sorgente OPC UA (console)

È possibile utilizzare la console per configurare la sorgente OPC UA con la seguente procedura.

 Note

Attenzione: la duplicazione TQVs può comportare una doppia ricarica.

Per configurare una sorgente OPC UA utilizzando la console AWS IoT SiteWise

1. Passare alla [console AWS IoT SiteWise](#).
2. Nella barra di navigazione a sinistra, scegli Edge gateway nella sezione Edge.
3. Seleziona il gateway SiteWise Edge per aggiungere una fonte OPC UA.
4. Scegli Aggiungi origine dati
5. Immettete un nome per la fonte.
6. Compilare il campo Local endpoint (Endpoint locale) del server dell'origine dati. L'endpoint può essere l'indirizzo IP o il nome host. È inoltre possibile aggiungere un numero di porta all'endpoint locale. Ad esempio, l'endpoint locale potrebbe avere il seguente aspetto:
opc.tcp://203.0.113.0:49320
7. (Facoltativo) Per Node ID for selection, aggiungi filtri di nodo per limitare i flussi di dati che vengono importati nel Cloud. AWS Per impostazione predefinita, i gateway SiteWise Edge utilizzano il nodo radice di un server per importare tutti i flussi di dati. È possibile utilizzare i filtri dei nodi per ridurre il tempo di avvio e l'utilizzo della CPU del gateway SiteWise Edge includendo solo i percorsi dei dati utilizzati per la modellazione. AWS IoT SiteWise Per impostazione predefinita, i gateway SiteWise Edge caricano tutti i percorsi OPC UA tranne quelli che iniziano con. /Server/ Per definire i filtri dei nodi OPC UA, è possibile utilizzare i percorsi dei nodi * e ** i caratteri jolly. Per ulteriori informazioni, consulta [Usa i filtri dei nodi OPC UA in Edge SiteWise](#).
8. Le destinazioni variano tra i gateway V3 abilitati per MQTT e i gateway Classic Stream V2.
 - Gli stream classici, le destinazioni gateway V2 hanno una relazione 1:1 con l'origine. Ogni fonte invia dati a una destinazione particolare.
 - Le destinazioni gateway V3 abilitate per MQTT vengono configurate separatamente perché il modello hub and spoke consente di centralizzare la configurazione e la gestione di più fonti di dati su gateway diversi. Per configurare le destinazioni in un gateway V3, vedere. [Comprendi le destinazioni Edge AWS IoT SiteWise](#)

Classic streams, V2 gateway destinations

- AWS IoT SiteWise in tempo reale: scegli questa opzione per inviare i dati direttamente allo AWS IoT SiteWise storage. Acquisisci e monitora i dati in tempo reale all'edge.
- AWS IoT SiteWise Memorizzato nel buffer con Amazon S3: invia dati in formato Parquet ad Amazon S3 e quindi importali nello storage. AWS IoT SiteWise Scegli questa opzione per importare i dati in batch e archiviare i dati storici in modo conveniente. Puoi configurare la posizione preferita del bucket Amazon S3 e la frequenza con cui desideri che i dati vengano caricati su Amazon S3. Puoi anche scegliere cosa fare con i dati dopo l'ingestione. AWS IoT SiteWise Puoi scegliere di rendere i dati disponibili sia AWS IoT SiteWise in Amazon S3 che in Amazon S3 oppure puoi scegliere di eliminarli automaticamente da Amazon S3 dopo che sono stati importati in. AWS IoT SiteWise
- Il bucket Amazon S3 è un meccanismo di staging e buffering e supporta file in formato Parquet.
- Se selezioni la casella di controllo Importa dati nello AWS IoT SiteWise storage, i dati vengono caricati prima in Amazon S3 e poi nello AWS IoT SiteWise storage.
 - Se selezioni la casella di controllo Elimina dati da Amazon S3, i dati vengono eliminati da Amazon S3 dopo essere stati importati nello storage. SiteWise
 - Se deselezioni la casella di controllo Elimina dati da Amazon S3, i dati vengono archiviati sia in Amazon S3 che in storage. SiteWise
- Se deselezioni la casella di controllo Importa dati nello AWS IoT SiteWise storage, i dati vengono archiviati solo in Amazon S3. Non viene importato nello SiteWise storage.

Visita [Gestisci l'archiviazione dei dati](#) per i dettagli sulle varie opzioni di archiviazione AWS IoT SiteWise offerte. Per ulteriori informazioni sulle opzioni di prezzo, consulta la pagina [AWS IoT SiteWise dei prezzi](#).

- AWS IoT Greengrass stream manager: utilizza AWS IoT Greengrass stream manager per inviare dati alle seguenti destinazioni AWS cloud: canali in AWS IoT Analytics, flussi in Amazon Kinesis Data Streams, proprietà degli asset o oggetti AWS IoT SiteWise in Amazon Simple Storage Service (Amazon S3). Per ulteriori informazioni, consulta [Manage data stream on the AWS IoT Greengrass](#) Core nella Developer Guide. AWS IoT Greengrass Version 2

Inserisci un nome per lo AWS IoT Greengrass stream.

MQTT-enabled, V3 gateway destinations

1. [Gateway V3 compatibili con MQTT per Edge AWS IoT SiteWise](#) Per informazioni su come aggiungere le destinazioni pertinenti, consulta la pagina.
 2. Ritorna a questa procedura dopo aver aggiunto le destinazioni di origine.
9. Nel riquadro Configurazione avanzata, puoi fare quanto segue:
- a. Scegli una modalità di sicurezza dei messaggi per le connessioni e i dati in transito tra il server di origine e il gateway SiteWise Edge. Questo campo è la combinazione della politica di sicurezza OPC UA e della modalità di sicurezza dei messaggi. Scegliete la stessa politica di sicurezza e la stessa modalità di sicurezza dei messaggi che avete specificato per il vostro server OPC UA.
 - b. Se la tua fonte richiede l'autenticazione, scegli un AWS Secrets Manager segreto dall'elenco di configurazione dell'autenticazione. Il gateway SiteWise Edge utilizza le credenziali di autenticazione contenute in questo segreto quando si connette a questa fonte di dati. È necessario allegare segreti al AWS IoT Greengrass componente del gateway SiteWise Edge per utilizzarli per l'autenticazione delle fonti di dati. Per ulteriori informazioni, consulta [the section called "Configura l'autenticazione della fonte di dati"](#).
-  **Tip**

Il server di dati potrebbe disporre di un'opzione denominata Allow anonymous login (Consenti accesso anonimo). Se questa opzione è impostata su Yes (Sì), l'origine non richiede l'autenticazione.
- c. (Facoltativo) È possibile attivare un prefisso del flusso di dati selezionando Attiva prefisso del flusso di dati - opzionale.
 - Immettete un prefisso per il flusso di dati. Il gateway SiteWise Edge aggiunge questo prefisso a tutti i flussi di dati provenienti da questa fonte. Utilizzare un prefisso del flusso di dati per distinguere tra flussi di dati con lo stesso nome da origini diverse. Ogni flusso di dati deve avere un nome univoco all'interno del tuo account.
 - d. (Facoltativo) Scegliete un'opzione di conversione del tipo di dati per convertire i tipi di dati OPC UA non supportati in stringhe prima di inserirli. AWS IoT SiteWise Converte i valori degli array con tipi di dati semplici in stringhe JSON e i tipi di dati in stringhe ISO 8601 DateTime . Per ulteriori informazioni, consulta [Conversione di tipi di dati non supportati](#).

- e. (Facoltativo) Per i gruppi di proprietà, scegliete Aggiungi nuovo gruppo.
 - i. Immettete un nome per il gruppo di proprietà.
 - ii. Per le proprietà:
 - 1. Per i percorsi dei nodi, aggiungi i filtri dei nodi OPC UA per limitare i percorsi OPC UA su cui vengono caricati. AWS IoT SiteWise Il formato è simile all'ID del nodo per la selezione.
 - iii. Per le impostazioni del gruppo, effettuate le seguenti operazioni:
 - 1. Per l'impostazione della qualità dei dati, scegliete il tipo di qualità dei dati che desiderate che AWS IoT SiteWise Collector inserisca.
 - 2. Per l'impostazione della modalità di scansione, configurate le proprietà standard dell'abbonamento utilizzando la modalità di scansione. È possibile selezionare Sottoscrivi o Sondaggio. Per ulteriori informazioni sulla modalità di scansione, vedere [the section called “Filtra gli intervalli di inserimento dei dati”](#).

Subscribe

Per inviare ogni punto dati

- i. Scegli Iscriviti e imposta quanto segue:
 - A. [Trigger di modifica dei dati](#): la condizione che avvia un avviso di modifica dei dati.
 - B. [Dimensione della coda di sottoscrizione](#): la profondità della coda su un server OPC-UA per una particolare metrica in cui vengono messe in coda le notifiche per gli elementi monitorati.
 - C. Intervallo di [pubblicazione dell'abbonamento: l'intervallo](#) (in millisecondi) del ciclo di pubblicazione specificato al momento della creazione dell'abbonamento.
 - D. Intervallo di istantanea - Facoltativo: l'impostazione del timeout della frequenza delle istantanee per garantire che Edge acquisisca un flusso costante di dati. AWS IoT SiteWise
 - E. Velocità di scansione: la velocità con cui si desidera che il gateway SiteWise Edge legga i registri. AWS IoT SiteWise calcola automaticamente la velocità di scansione minima consentita per il SiteWise gateway Edge.

- F. Timestamp: il timestamp da includere nei punti dati OPC UA. Puoi utilizzare il timestamp del server o il timestamp del tuo dispositivo.

**Note**

Utilizza la versione 2.5.0 o successiva del componente IoT SiteWise OPC UA Collector. Se si utilizza la funzionalità di timestamp con le versioni precedenti, gli aggiornamenti della configurazione non riescono. Per ulteriori informazioni, consulta [Aggiorna la versione di un componente AWS IoT SiteWise](#).

- ii. Nelle impostazioni Deadband, configura un tipo di Deadband. Il tipo di banda morta controlla quali dati ti invia la fonte e quali dati scarta. AWS IoT SiteWise Per ulteriori informazioni sull'impostazione della banda morta, consulta [the section called "Filtra gli intervalli di inserimento dei dati"](#)
- Nessuno: il server associato invia tutti i punti dati per questo gruppo di proprietà.
 - Percentuale: il server associato invia solo i dati che non rientrano in una percentuale specificata dell'intervallo dei dati. Questo intervallo viene calcolato dal server in base all'unità tecnica minima e massima definita per ciascun nodo. Se il server non supporta bande morte percentuali o non dispone di unità tecniche definite, il gateway calcola l'intervallo utilizzando i valori minimo e massimo forniti di seguito.
 - Assoluto: il server associato invia solo dati che non rientrano in un intervallo specifico.
- A. Imposta il valore Deadband come percentuale dell'intervallo di dati sulla banda morta.
- B. (Facoltativo) Specificate un valore minimo e massimo per l'intervallo di banda morta utilizzando Intervallo minimo - opzionale e Intervallo massimo - opzionale.

Poll

Per inviare punti dati a un intervallo specifico

- Scegliete Sondaggio e impostate quanto segue:

- A. Velocità di scansione: la velocità con cui desideri che il gateway SiteWise Edge legga i tuoi registri. AWS IoT SiteWise calcola automaticamente la velocità di scansione minima consentita per il SiteWise gateway Edge.
- B. Timestamp: il timestamp da includere nei punti dati OPC UA. Puoi utilizzare il timestamp del server o il timestamp del tuo dispositivo.

**Note**

Utilizza la versione 2.5.0 o successiva del componente IoT SiteWise OPC UA Collector. Se si utilizza la funzionalità di timestamp con le versioni precedenti, gli aggiornamenti della configurazione non riescono. Per ulteriori informazioni, consulta [Aggiorna la versione di un componente AWS IoT SiteWise](#).

**Note**

Le impostazioni Deadband sono applicabili se hai selezionato Iscriviti nelle impostazioni della modalità di scansione.

10. Scegli Save (Salva).

Configurare una fonte OPC UA (AWS CLI)

È possibile definire sorgenti di dati OPC UA per un gateway SiteWise Edge utilizzando il AWS CLI. A tale scopo, create un file JSON di configurazione della funzionalità OPC UA e utilizzate il [update-gateway-capability-configuration](#) comando per aggiornare la configurazione del gateway SiteWise Edge. È necessario definire tutte le sorgenti OPC UA in un'unica configurazione di funzionalità.

MQTT-enabled, V3 gateway

Questa funzionalità ha il seguente spazio dei nomi.

- `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"
      }
    },
    "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

Questa funzionalità ha il seguente spazio dei nomi.

- `iotsitewise:opcuacollector:2`

Sintassi della richiesta

```

{
  "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
    }
  }
]
}
]
}

```

Corpo della richiesta

sources

Un elenco di strutture di definizione dei sorgenti OPC UA, ciascuna delle quali contiene le seguenti informazioni:

name

Un nome descrittivo univoco per l'origine.

endpoint

Una struttura endpoint contenente le seguenti informazioni:

certificateTrust

Una struttura delle policy di attendibilità dei certificati contenente le seguenti informazioni:

type

La modalità di attendibilità dei certificati per l'origine. Seleziona una delle seguenti opzioni:

- **TrustAny**— Il gateway SiteWise Edge si fida di qualsiasi certificato quando si connette alla fonte OPC UA.
- **X509**— Il gateway SiteWise Edge si fida di un certificato X.509 quando si connette alla fonte OPC UA. Se si sceglie questa opzione, è necessario definire `certificateBody` in `certificateTrust`. È inoltre possibile definire `certificateChain` in `certificateTrust`.

certificateBody

(Facoltativo) Il corpo di un certificato X.509.

Questo campo è obbligatorio se si sceglie X509 per `type` in `certificateTrust`.

certificateChain

(Facoltativo) La catena di attendibilità per un certificato X.509.

Questo campo viene utilizzato solo se si sceglie X509 per `type` in `certificateTrust`.

endpointUri

L'endpoint locale della sorgente OPC UA. Ad esempio, l'endpoint locale potrebbe essere simile a `opc.tcp://203.0.113.0:49320`.

securityPolicy

La politica di sicurezza da utilizzare per proteggere i messaggi letti dalla fonte OPC UA. Seleziona una delle seguenti opzioni:

- **NONE**— Il gateway SiteWise Edge non protegge i messaggi provenienti dalla fonte OPC UA. Ti consigliamo di scegliere una politica di sicurezza diversa. Se si sceglie questa opzione, è necessario anche selezionare **NONE** per `messageSecurityMode`.
- **BASIC256_SHA256**— La politica Basic256Sha256 di sicurezza.

- AES128_SHA256_RSA0AEP— La politica Aes128_Sha256_Rsa0aep di sicurezza.
- AES256_SHA256_RSAPSS— La politica Aes256_Sha256_RsaPss di sicurezza.
- BASIC128_RSA15— (Obsoleta) La politica Basic128Rsa15 di sicurezza è obsoleta nelle specifiche OPC UA perché non è più considerata sicura. Ti consigliamo di scegliere una politica di sicurezza diversa. Per ulteriori informazioni, vedere [Profile SecurityPolicy — Basic128Rsa15](#).
- BASIC256— (Obsoleto) La politica di Basic256 sicurezza è obsoleta nelle specifiche OPC UA perché non è più considerata sicura. Ti consigliamo di scegliere una politica di sicurezza diversa. Per ulteriori informazioni, vedere [SecurityPolicy — Basic256](#).

 Important

Se scegli una politica di sicurezza diversa da NONE, devi scegliere SIGN o SIGN_AND_ENCRYPT per `messageSecurityMode`. È inoltre necessario configurare il server di origine in modo che consideri attendibile il gateway SiteWise Edge. Per ulteriori informazioni, consulta [Configura i server OPC UA in modo che si affidino al gateway AWS IoT SiteWise Edge](#).

`messageSecurityMode`

La modalità di sicurezza dei messaggi da utilizzare per proteggere le connessioni alla sorgente OPC UA. Seleziona una delle seguenti opzioni:

- NONE— Il gateway SiteWise Edge non protegge le connessioni alla fonte OPC UA. Ti consigliamo di scegliere una modalità di sicurezza dei messaggi diversa. Se si sceglie questa opzione, è necessario anche selezionare NONE per `securityPolicy`.
- SIGN— I dati in transito tra il gateway SiteWise Edge e la fonte OPC UA sono firmati ma non crittografati.
- SIGN_AND_ENCRYPT— I dati in transito tra il gateway e la fonte OPC UA sono firmati e crittografati.

 Important

Se si sceglie una modalità di sicurezza dei messaggi diversa da NONE, è necessario sceglierne una `securityPolicy` diversa da NONE. È inoltre necessario configurare il server di origine in modo che consideri attendibile il gateway SiteWise Edge. Per

ulteriori informazioni, consulta [Configura i server OPC UA in modo che si affidino al gateway AWS IoT SiteWise Edge](#).

identityProvider

Una struttura del provider di identità che contiene le seguenti informazioni:

type

Il tipo di credenziali di autenticazione richieste dall'origine. Seleziona una delle seguenti opzioni:

- Anonymous— La fonte non richiede l'autenticazione per la connessione.
- Username— L'origine richiede un nome utente e una password per la connessione. Se si sceglie questa opzione, è necessario definire `usernameSecretArn` in `identityProvider`.

usernameSecretArn

(Facoltativo) L'ARN di un AWS Secrets Manager segreto. Il gateway SiteWise Edge utilizza le credenziali di autenticazione contenute in questo segreto quando si connette a questa fonte. È necessario collegare segreti al SiteWise connettore IoT del gateway SiteWise Edge per utilizzarli per l'autenticazione dell'origine. Per ulteriori informazioni, consulta [Configurare l'autenticazione dell'origine dati per SiteWise Edge](#).

Questo campo è obbligatorio se si sceglie Username per `type` in `identityProvider`.

nodeFilterRules

Un elenco di strutture di regole di filtro dei nodi che definiscono i percorsi dei flussi di dati OPC UA da inviare al AWS Cloud. Puoi utilizzare i filtri dei nodi per ridurre il tempo di avvio e l'utilizzo della CPU del gateway SiteWise Edge includendo solo i percorsi dei dati utilizzati per la modellazione. AWS IoT SiteWise Per impostazione predefinita, i gateway SiteWise Edge caricano tutti i percorsi OPC UA tranne quelli che iniziano con. `/Server/` Per definire i filtri dei nodi OPC UA, è possibile utilizzare i percorsi dei nodi `*` e `**` i caratteri jolly. Per ulteriori informazioni, consulta [Usa i filtri dei nodi OPC UA in Edge SiteWise](#).

Ogni struttura dell'elenco deve contenere le seguenti informazioni:

action

L'operazione per questa regola di filtro dei nodi. Puoi scegliere le seguenti opzioni:

- **INCLUDE**— Il gateway SiteWise Edge include solo flussi di dati che soddisfano questa regola.

definition

Una struttura di regole del filtro dei nodi che contiene le seguenti informazioni:

type

Il tipo di percorso del filtro dei nodi per questa regola. Puoi scegliere le seguenti opzioni:

- **OpcUaRootPath**— Il gateway SiteWise Edge valuta questo percorso di filtro del nodo rispetto alla radice della gerarchia dei percorsi OPC UA.

rootPath

Il percorso del filtro del nodo da valutare rispetto alla radice della gerarchia dei percorsi OPC UA. Questo percorso deve iniziare con. /

measurementDataStreamPrefix

Una stringa da aggiungere a tutti i flussi di dati dalla fonte. Il gateway SiteWise Edge aggiunge questo prefisso a tutti i flussi di dati provenienti da questa fonte. Utilizzare un prefisso del flusso di dati per distinguere tra flussi di dati con lo stesso nome da origini diverse. Ogni flusso di dati deve avere un nome univoco all'interno del tuo account.

typeConversions

I tipi di conversioni disponibili per i tipi di dati OPC UA non supportati. Ogni tipo di dati viene convertito in stringhe. Per ulteriori informazioni, consulta [Conversione di tipi di dati non supportati](#).

array

Il tipo di dati di matrice semplice che viene convertito in stringhe. Puoi scegliere le seguenti opzioni:

- **JsonArray**— Indica che avete scelto di convertire i tipi di dati dell'array semplice in stringhe.

datetime

Il tipo di DateTime dati convertito in stringhe. Puoi scegliere le seguenti opzioni:

- **ISO8601String**— Indica che avete scelto di convertire i tipi di dati ISO 8601 in stringhe.

destination

Configurazione per la destinazione dei tag OPC UA. I gateway V3 con stream classico, v2 e MQTT hanno configurazioni diverse per le destinazioni.

type

Il tipo di destinazione.

`streamName`— solo per stream classici, gateway V2

Il nome del flusso . Il nome dello stream deve essere univoco.

`streamBufferSize`— solo per stream classici, gateway V2

La dimensione del buffer dello stream. Questo è importante per gestire il flusso di dati da fonti OPC UA.

propertyGroups

(Facoltativo) L'elenco dei gruppi di proprietà che definiscono deadband e scanMode richiesti dal protocollo.

name

Il nome del gruppo di proprietà. Dovrebbe essere un identificatore univoco.

deadband

Il deadband valore definisce la modifica minima del valore di un punto dati che deve avvenire prima che i dati vengano inviati al cloud. Contiene le seguenti informazioni:

type

I tipi di banda morta supportati. Puoi scegliere le seguenti opzioni:

- **ABSOLUTE**— Un valore fisso che specifica la modifica assoluta minima richiesta per considerare un punto dati sufficientemente significativo da essere inviato al cloud.
- **PERCENT**— Un valore dinamico che specifica la modifica minima richiesta come percentuale del valore dell'ultimo punto dati inviato. Questo tipo di banda morta è utile quando i valori dei dati variano notevolmente nel tempo.

value

Il valore della banda morta. Quando `type` è **ABSOLUTE**, questo valore è un doppio senza unità. Quando `type` è **PERCENT**, questo valore è un doppio tra 1 e 100.

eguMin

(Facoltativo) L'unità tecnica minima quando si utilizza una PERCENT banda morta. Si imposta questa impostazione se il server OPC UA non ha unità tecniche configurate.

eguMax

(Facoltativo) Il valore massimo dell'unità tecnica quando si utilizza una banda PERCENT morta. Si imposta questa impostazione se il server OPC UA non ha unità tecniche configurate.

timeoutMilliseconds

La durata in millisecondi prima del timeout. Il valore minimo è. 100

scanMode

La scanMode struttura che contiene le seguenti informazioni:

type

I tipi supportati di scanMode. I valori accettati sono POLL e EXCEPTION.

rate

L'intervallo di campionamento per la modalità di scansione.

timestampToReturn

La fonte del timestamp. Puoi scegliere le seguenti opzioni:

- SOURCE_TIME— Utilizza il timestamp del tuo dispositivo.
- SERVER_TIME— Utilizza il timestamp del tuo server.

Note

Da utilizzare TimestampToReturn con la versione 2.5.0 o successiva del componente IoT SiteWise OPC UA Collector. Se si utilizza questa funzionalità con versioni precedenti, gli aggiornamenti della configurazione non riescono. Per ulteriori informazioni, consulta [Aggiorna la versione di un componente AWS IoT SiteWise](#).

nodeFilterRuleDefinitions

(Facoltativo) Un elenco di percorsi di nodi da includere nel gruppo di proprietà. I gruppi di proprietà non possono sovrapporsi. Se non si specifica un valore per questo campo, il gruppo contiene tutti i percorsi sotto la radice e non è possibile creare gruppi di proprietà aggiuntivi. La struttura `nodeFilterRuleDefinitions` contiene le seguenti informazioni:

type

`OpcUaRootPath` è l'unico tipo supportato. Ciò specifica che il valore di `rootPath` è un percorso relativo alla radice dello spazio di navigazione OPC UA.

rootPath

Un elenco delimitato da virgole che specifica i percorsi (relativi alla radice) da includere nel gruppo di proprietà.

Esempi di configurazione delle funzionalità aggiuntivi per stream Classic, gateway V2 (AWS CLI)

L'esempio seguente definisce una configurazione della funzionalità del gateway OPC UA SiteWise Edge a partire da un payload archiviato in un file JSON.

```
aws iotsitewise update-gateway-capability-configuration \
--capability-namespace "iotsitewise:opcuacollector:2" \
--capability-configuration file://opc-ua-configuration.json
```

Example : configurazione del codice sorgente OPC UA

Il `opc-ua-configuration.json` file seguente definisce una configurazione sorgente OPC UA di base e non sicura.

```
{
  "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 : configurazione del codice sorgente OPC UA con gruppi di proprietà definiti

Il `opc-ua-configuration.json` file seguente definisce una configurazione sorgente OPC UA di base e non sicura con gruppi di proprietà definiti.

```

{
  "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 : configurazione del codice sorgente OPC UA con proprietà

Il seguente esempio JSON `opc-ua-configuration.json` definisce una configurazione di origine OPC UA con le seguenti proprietà:

- Considera attendibile qualsiasi certificato.
- Utilizza la politica BASIC256 di sicurezza per proteggere i messaggi.
- Usa la modalità SIGN_AND_ENCRYPT per proteggere le connessioni.
- Utilizza le credenziali di autenticazione memorizzate in un segreto di Secrets Manager.
- Filtra i flussi di dati tranne quelli il cui percorso inizia con `/WindFarm/2/WindTurbine/`.
- Aggiunge `/Washington` all'inizio di ogni percorso del flusso di dati per distinguere tra questo "Parco eolico #2" e un "Parco eolico #2" in un'altra area.

```

{
  "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 : configurazione del codice sorgente OPC UA con certificato attendibile

Il seguente esempio JSON `opc-ua-configuration.json` definisce una configurazione di origine OPC UA con le seguenti proprietà:

- Considera attendibile un determinato certificato X.509.
- Utilizza la politica BASIC256 di sicurezza per proteggere i messaggi.
- Usa la modalità SIGN_AND_ENCRYPT per proteggere le connessioni.

```

{
  "sources": [
    {
      "name": "Wind Farm #3",
      "endpoint": {
        "certificateTrust": {
          "type": "X509",
          "certificateBody": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w
0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZ
WF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC01BTsBDb25zb2x1MRIw
EAYDVQQDEw1UZXRhbnQ2IjYwMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvi5
jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTEwNDI1MjA0NTIxWjCBiDELMAkGA1UEBh
MCVVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBb
WF6b24xFDASBgNVBAwTC01BTsBDb25zb2x1MRIwEAYDVQQDEw1UZXRhbnQ2IjYwMx
HzAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvi5jb20wgZ8wDQYJKoZIhvcNAQEE
BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ21uUSfwfEvySwTC2XADZ4nB+BLyGVI
k60CpiwsZ3G93vUEI03IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
ITx0USQv7c7ugFFDzQGBzZswY6786m86gpEIbb30hjZnzcvcQAaRHhd1QWIMm2nr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4nUhVVxYUntneD9+h8Mg9q6q+auN
KyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6Guo
EDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjStbNYiytVbZPQUQ5Yaxu2jXnimvw

```

```

3rrszlaEXAMPLE=
    -----END CERTIFICATE-----",
    "certificateChain": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w
0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZ
WF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAstC0lBTSBDb25zb2x1MRIw
EAYDVQQDEw1UZXR0Q21sYWxhZAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvi5
jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTEwNDI1MjA0NTIxWjCBiDELMAkGA1UEBh
MCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBb
WF6b24xFDASBgNVBAstC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZXR0Q21sYWxh
ZAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvi5jb20wgZ8wDQYJKoZIhvcNAQEE
BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ21uUSfwfEvySwTC2XADZ4nB+BLygVI
k60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
ITx0USQv7c7ugFFDzQGBzZswY6786m86gpEIbb30hjZnzcVQAaRHhd1QWIMm2nr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4nUhVVxYUntneD9+h8Mg9q6q+auN
KyExzyLwax1Aoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6Guo
EDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjSTbNYiytVbZPQUQ5Yaxu2jXnimvw
3rrszlaEXAMPLE=
    -----END CERTIFICATE-----"
    },
    "endpointUri": "opc.tcp://203.0.113.2:49320",
    "securityPolicy": "BASIC256",
    "messageSecurityMode": "SIGN_AND_ENCRYPT",
    "identityProvider": {
        "type": "Anonymous"
    },
    "nodeFilterRules": []
    },
    "measurementDataStreamPrefix": ""

}
]
}

```

Configura i server OPC UA in modo che si affidino al gateway AWS IoT SiteWise Edge

Se scegli un'opzione `messageSecurityMode` diversa da Nessuno quando configuri la tua sorgente OPC UA, devi abilitare i server di origine a considerare attendibile il AWS IoT SiteWise gateway Edge. Il gateway SiteWise Edge genera un certificato che il server di origine potrebbe richiedere. Il processo varia a seconda dei server di origine. Per ulteriori informazioni, consulta la documentazione relativa ai server in uso.

La procedura seguente descrive i passaggi di base.

Per consentire a un server OPC UA di considerare attendibile il SiteWise gateway Edge

1. Apri l'interfaccia per configurare il tuo server OPC UA.
2. Inserite il nome utente e la password per l'amministratore del server OPC UA.
3. Individua Trusted Clients nell'interfaccia, quindi scegli AWS IoT SiteWise Gateway Client.
4. Scegliere Trust (Considera attendibile).

Esportazione del certificato client OPC UA

Alcuni server OPC UA richiedono l'accesso al file di certificato del client OPC UA per considerare attendibili il gateway Edge. SiteWise Se ciò si applica ai server OPC UA, è possibile utilizzare la seguente procedura per esportare il certificato client OPC UA dal gateway Edge. SiteWise Quindi, puoi importare il certificato sul tuo server OPC UA.

Per esportare il file del certificato del client OPC UA per una fonte

1. Eseguire il comando seguente per passare alla directory che contiene il file del certificato. Sostituisci *sitewise-work* con il percorso di archiviazione locale per la cartella di lavoro *aws.iot.SiteWiseEdgeCollectorOpcua* Greengrass e *source-name* sostituisilo con il nome dell'origine dati.

Per impostazione predefinita, la cartella di lavoro Greengrass si trova */greengrass/v2/work/aws.iot.SiteWiseEdgeCollectorOpcua* su Linux e *C:/greengrass/v2/work/aws.iot.SiteWiseEdgeCollectorOpcua* su. Microsoft Windows

```
cd /sitewise-work/source-name/opcua-certificate-store
```

2. Il certificato client OPC UA del gateway SiteWise Edge per questa fonte è contenuto nel *aws-iot-opcua-client.pfx* file.

Eseguire il comando seguente per esportare il certificato in un file *.pem* chiamato *aws-iot-opcua-client-certificate.pem*.

```
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. Trasferisci il file del certificato dal gateway SiteWise Edge al server OPC UA. *aws-iot-opcua-client-certificate.pem*

Per fare ciò, è possibile utilizzare software comuni come il programma `scp` per trasferire il file utilizzando il protocollo SSH. Per ulteriori informazioni, consultare la pagina [Secure copy](#) su Wikipedia.

Note

Se il tuo gateway SiteWise Edge è in esecuzione su Amazon Elastic Compute Cloud (Amazon EC2) e ti connetti ad esso per la prima volta, devi configurare i prerequisiti per la connessione. Per ulteriori informazioni, consulta [Connect to your Linux using SSH](#) nella Amazon EC2 User Guide.

4. Importa il file del certificato sul server OPC UA per fidarti del gateway SiteWise Edge. `aws-iot-opcua-client-certificate.pem` Le fasi variano a seconda dei server di origine in uso. Consultare la documentazione relativa al server.

Filtra gli intervalli di inserimento dei dati con OPC UA

È possibile controllare il modo in cui si inseriscono i dati con una fonte OPC UA utilizzando la modalità di scansione e gli intervalli di banda morta. Queste funzionalità consentono di controllare il tipo di dati da importare e come e quando il server e il gateway SiteWise Edge si scambiano queste informazioni.

Raccogli o filtra i dati in base alla qualità

È possibile configurare le impostazioni di qualità dei dati per controllare quali dati vengono raccolti dalla fonte OPC UA. La fonte di dati include la valutazione della qualità come metadati quando la invia. Puoi selezionare una o tutte le seguenti opzioni:

- Good
- Bad
- Uncertain

Gestire valori NaN o null

SiteWise Edge supporta la raccolta e la gestione di valori NaN e null.

- NaN (Not a Number): rappresenta risultati numerici non definiti o non rappresentabili.

- Nullo: indica i dati mancanti.

Il collettore IoT SiteWise OPC UA acquisisce valori NaN e Null con qualità BAD o UNCERTAIN. Questi valori speciali vengono scritti nel flusso locale, consentendo una raccolta di dati più completa.

Controlla la frequenza di raccolta dei dati con la modalità Scan

È possibile configurare la modalità di scansione OPC UA per controllare il modo in cui si raccolgono i dati dalla fonte OPC UA. Puoi scegliere la modalità di abbonamento o di sondaggio.

- Modalità di abbonamento: la fonte OPC UA raccoglie i dati da inviare al gateway SiteWise Edge alla frequenza definita dalla velocità di scansione. Il server invia i dati solo quando il valore è cambiato, quindi questa è la frequenza massima di ricezione dei dati del gateway SiteWise Edge.
- Modalità polling: il gateway SiteWise Edge esegue il polling della sorgente OPC UA a una frequenza prestabilita definita dalla velocità di scansione. Il server invia i dati indipendentemente dal fatto che il valore sia cambiato, quindi il gateway SiteWise Edge riceve sempre i dati a questo intervallo.

 Note

L'opzione della modalità polling ha la precedenza sulle impostazioni della banda morta per questa fonte.

Filtra l'ingestione di dati OPC UA con intervalli di banda morta

Puoi applicare una banda morta ai tuoi gruppi di proprietà di origine OPC UA per filtrare e scartare determinati dati invece di inviarli al Cloud. AWS Una banda morta specifica una finestra di fluttuazioni previste nei valori dei dati in entrata dalla fonte OPC UA. Se i valori rientrano in questa finestra, il server OPC UA non li invierà al Cloud. AWS Puoi utilizzare il filtro a banda morta per ridurre la quantità di dati che stai elaborando e inviando al Cloud. AWS Per sapere come configurare le sorgenti OPC UA per il tuo gateway SiteWise Edge, consulta. [Fonti di dati OPC UA per gateway Edge AWS IoT SiteWise](#)

 Note

Il server elimina tutti i dati che rientrano nella finestra specificata dalla banda morta. Non puoi recuperare questi dati scartati.

Tipi di zone morte

È possibile specificare due tipi di bande morte per il gruppo di proprietà del server OPC UA. Questi consentono di scegliere la quantità di dati da inviare al AWS Cloud e la quantità da scartare.

- **Percentuale:** si specifica una finestra utilizzando una percentuale della fluttuazione prevista nel valore di misurazione. Il server calcola la finestra esatta in base a questa percentuale e invia al AWS Cloud i dati che superano i limiti della finestra. Ad esempio, specificando un valore di banda morta del 2% su un sensore con un intervallo compreso tra -100 gradi Fahrenheit e +100 gradi Fahrenheit, si indica al server di inviare dati al Cloud quando il valore cambia di 4 gradi Fahrenheit o più. AWS

Note

Facoltativamente, puoi specificare un valore di banda morta minimo e massimo per questa finestra se il server di origine non definisce le unità tecniche. Se non viene fornito un intervallo di unità tecniche, il server OPC UA utilizza per impostazione predefinita l'intervallo completo del tipo di dati di misurazione.

- **Assoluto:** si specifica una finestra utilizzando unità esatte. Ad esempio, se si specifica un valore di banda morta pari a 2 su un sensore, si indica al server di inviare dati al AWS Cloud quando il loro valore cambia di almeno 2 unità. È possibile utilizzare la banda morta assoluta per ambienti dinamici in cui sono regolarmente previste fluttuazioni durante le normali operazioni.

Timeout con banda morta

Facoltativamente, puoi configurare un'impostazione di timeout per la banda morta. Dopo questo timeout, il server OPC UA invia il valore di misurazione corrente anche se rientra nella fluttuazione prevista della banda morta. È possibile utilizzare l'impostazione del timeout per garantire AWS IoT SiteWise l'acquisizione di un flusso costante di dati in ogni momento, anche quando i valori non superano la finestra di banda morta definita.

Usa i filtri dei nodi OPC UA in Edge SiteWise

Quando si definiscono le fonti di dati OPC UA per un gateway SiteWise Edge, è possibile definire filtri di nodo. I filtri dei nodi consentono di limitare i percorsi del flusso di dati che il gateway SiteWise Edge invia al cloud. Puoi utilizzare i filtri dei nodi per ridurre il tempo di avvio e l'utilizzo della CPU del gateway SiteWise Edge includendo solo i percorsi dei dati utilizzati per la modellazione AWS IoT SiteWise. Per impostazione predefinita, i gateway SiteWise Edge caricano tutti i percorsi OPC UA

tranne quelli che iniziano con. /Server/ È possibile utilizzare i caratteri jolly * e ** nei filtri dei nodi per includere più percorsi dei flussi di dati con un solo filtro. Per informazioni su come configurare le sorgenti OPC UA per il gateway SiteWise Edge, consulta. [Fonti di dati OPC UA per gateway Edge AWS IoT SiteWise](#)

Note

AWS IoT SiteWise riavvia il gateway SiteWise Edge ogni volta che aggiungi o modifichi una fonte. Il gateway SiteWise Edge non inserirà dati durante l'aggiornamento della configurazione di origine. Il tempo necessario per riavviare il gateway SiteWise Edge dipende dal numero di tag presenti nelle sorgenti del gateway SiteWise Edge. Il tempo di riavvio può variare da pochi secondi (per un gateway SiteWise Edge con pochi tag) a diversi minuti (per un gateway SiteWise Edge con molti tag).

La tabella seguente elenca i caratteri jolly che è possibile utilizzare per filtrare le fonti di dati OPC UA.

jolly del filtro dei nodi OPC UA

Carattere jolly	Descrizione
*	Corrisponde a un singolo livello in un percorso dei flussi di dati.
**	Corrisponde a più livelli in un percorso dei flussi di dati.

Note

Se configuri una fonte con un filtro ampio e successivamente modifichi la fonte per utilizzare un filtro più restrittivo, AWS IoT SiteWise interrompe l'archiviazione dei dati che non corrispondono al nuovo filtro.

Example : Scenario che utilizza filtri di nodo

Quelli che seguono sono ipotetici flussi di dati:

- /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

Utilizzando i flussi di dati precedenti, è possibile definire filtri di nodo per limitare i dati da includere dalla fonte OPC UA.

- Per selezionare tutti i nodi in questo esempio, usa / o. /**/ È possibile includere più directory o cartelle con i caratteri jolly **.
- Per selezionare tutti i flussi di dati PLC, si possono utilizzare i caratteri /*/*/*PLC* o /**/PLC*.
- Per selezionare tutti i contatori in questo esempio, utilizzate /**/Counter* o /*/*/*Counter*.
- Per selezionare tutti i contatori da Line 2, utilizzare /**/Line 2/Counter*.

Conversione di tipi di dati non supportati

Facoltativamente, abilita la conversione dei tipi di dati AWS IoT SiteWise per array e DateTime tipi di dati semplici. AWS IoT SiteWise non supporta tutti i tipi di dati OPC UA. Quando invii dati non supportati al tuo flusso di AWS IoT Greengrass dati, tali dati vengono persi. Tuttavia, convertendo i tipi di dati nativi non supportati in stringhe, puoi importare i dati anziché eliminarli AWS IoT SiteWise. AWS IoT SiteWise serializza i dati convertiti in modo da poter successivamente utilizzare le proprie funzioni per riconvertire le stringhe al tipo di dati originale a valle, se necessario.

È possibile aggiornare le impostazioni di conversione dei tipi di dati per un'origine dati in qualsiasi momento e ogni origine dati può avere le proprie impostazioni.

Quando aggiungi fonti di dati in Console AWS IoT SiteWise, ci sono due caselle di controllo in Conversione del tipo di dati nella configurazione avanzata. Puoi indicare quali tipi di dati convertire in stringhe.

Inoltre, il collettore IoT SiteWise OPC UA può accettare valori NaN o null sull'edge.

- Converti i valori degli array con tipi di dati semplici in stringhe JSON
- Converti DateTime i valori in stringhe ISO 8601

Prerequisito

- Utilizza la versione 2.5.0 o successiva del collettore [IoT SiteWise OPC UA](#).

Limitazioni

Queste sono le limitazioni per la conversione dei tipi di dati OPC UA in stringhe in AWS IoT SiteWise

- La conversione di tipi di dati complessi non è supportata.
- I limiti delle stringhe dopo la conversione sono di 1024 byte. Se la stringa è più lunga di 1024 byte, viene rifiutata da AWS IoT SiteWise

Configurare l'autenticazione dell'origine dati per SiteWise Edge

Se il tuo server OPC UA richiede credenziali di autenticazione per la connessione, puoi utilizzarle AWS Secrets Manager per creare e distribuire un codice segreto sul tuo SiteWise gateway Edge. AWS Secrets Manager crittografa i segreti sul dispositivo per proteggere il nome utente e la password finché non è necessario utilizzarli. Per ulteriori informazioni sul componente AWS IoT Greengrass Secret Manager, consulta [Secret manager](#) nella AWS IoT Greengrass Version 2 Developer Guide.

Per informazioni sulla gestione dell'accesso ai segreti di Secrets Manager, vedere:

- [Chi ha i permessi per accedere ai tuoi AWS Secrets Manager segreti.](#)
- [Determinare se una richiesta è consentita o rifiutata all'interno di un account.](#)

Fase 1: Creare segreti di autenticazione all'origine

Puoi utilizzarlo AWS Secrets Manager per creare un segreto di autenticazione per la tua fonte di dati. Nel campo segreto, definisci **username** coppie **password** chiave-valore che contengono i dettagli di autenticazione per la tua fonte di dati.

Creazione di un segreto (console)

1. Passare alla [console AWS Secrets Manager](#).
2. Scegli Archivia un nuovo segreto.
3. In Tipo segreto, scegli Altro tipo di segreti.
4. In Coppie chiave/valore, procedi come segue:

1. Nella prima casella di immissione, immettete **username** e nella seconda casella di immissione inserite il nome utente.
2. Scegli Aggiungi riga.
3. Nella prima casella di immissione, inserisci **password** e nella seconda casella di immissione inserisci la password.
5. Per la chiave di crittografia, seleziona aws/secretsmanager, quindi scegli Avanti.
6. Nella pagina Salva una nuova pagina segreta, inserisci un nome segreto.
7. (Facoltativo) Inserisci una descrizione che ti aiuti a identificare questo segreto, quindi scegli Avanti.
8. (Facoltativo) In Memorizza una nuova pagina segreta, attiva Rotazione automatica. Per ulteriori informazioni, consulta [Ruotare i segreti](#) nella Guida per l'AWS Secrets Manager utente.
9. Specificate un programma di rotazione.
10. Scegli una funzione Lambda in grado di ruotare questo segreto, quindi scegli Avanti.
11. Controlla le configurazioni segrete, quindi scegli Store.

Per autorizzare l'interazione con il gateway SiteWise Edge AWS Secrets Manager, il ruolo IAM del gateway SiteWise Edge deve consentire l'azione `secretsmanager:GetSecretValue`. Puoi utilizzare il dispositivo principale Greengrass per cercare la policy IAM. Per ulteriori informazioni sull'aggiornamento di una policy IAM, consulta [Modifica delle policy IAM](#) nella Guida per l'AWS Identity and Access Management utente.

Example policy

Sostituisci *secret-arn* con l'Amazon Resource Name (ARN) del segreto creato nel passaggio precedente. Per ulteriori informazioni su come ottenere l'ARN di un segreto, consulta [Find secrets AWS Secrets Manager in](#) the AWS Secrets Manager User Guide.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Effect": "Allow",
      "Resource": [
```

```
        "secret-arn"  
    ]  
}  
]  
}
```

Passaggio 2: distribuisce i segreti sul tuo dispositivo gateway SiteWise Edge

Puoi utilizzare la AWS IoT SiteWise console per distribuire segreti sul tuo gateway SiteWise Edge.

Per distribuire un segreto (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegliere Gateways.
3. Dall'elenco dei gateway, scegli il gateway SiteWise Edge di destinazione.
4. Nella sezione Configurazione del gateway, scegli il collegamento del dispositivo principale Greengrass per aprire il AWS IoT Greengrass core associato al gateway SiteWise Edge.
5. Nel riquadro di navigazione, scegli Implementazioni.
6. Scegli la distribuzione di destinazione, quindi scegli Rivedi.
7. Nella pagina Specificare la destinazione, scegli Avanti.
8. Nella pagina Seleziona componenti, nella sezione Componenti pubblici, disattiva Mostra solo i componenti selezionati.
9. Cerca e scegli aws.greengrass. SecretManagercomponente, quindi scegli Avanti.
10. Dall'elenco Componenti selezionati, scegli aws.greengrass. SecretManagercomponente, quindi scegli Configura componente.
11. Nel campo Configurazione da unire, aggiungi il seguente oggetto JSON.

Note

Sostituisci *secret-arn* con l'ARN del segreto creato nel passaggio precedente. Per ulteriori informazioni su come ottenere l'ARN di un segreto, consulta [Find secrets AWS Secrets Manager in](#) the AWS Secrets Manager User Guide.

```
{  
  "cloudSecrets": [  
    {
```

```
    "arn": "secret-arn"  
  }  
]  
}
```

12. Scegli Conferma.
13. Scegli Next (Successivo).
14. Nella pagina Configura impostazioni avanzate, scegli Avanti.
15. Rivedi le configurazioni di distribuzione, quindi scegli Distribuisci.

Passaggio 3: aggiungere configurazioni di autenticazione

È possibile utilizzare la AWS IoT SiteWise console per aggiungere configurazioni di autenticazione al gateway SiteWise Edge.

Per aggiungere configurazioni di autenticazione (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Dall'elenco dei gateway, scegli il gateway SiteWise Edge di destinazione.
3. Dall'elenco Sorgenti dati, scegli l'origine dati di destinazione, quindi scegli Modifica.
4. Nella pagina Aggiungi un'origine dati, scegli Configurazione avanzata.
5. Per la configurazione dell'autenticazione, scegli il segreto che hai distribuito nel passaggio precedente.
6. Seleziona Salva.

Fonti di dati dei partner sui SiteWise gateway Edge

Quando si utilizza un gateway AWS IoT SiteWise Edge, è possibile connettere un'origine dati partner al gateway SiteWise Edge e ricevere dati dal partner nel gateway SiteWise Edge e nel AWS cloud. Queste fonti di dati dei partner sono AWS IoT Greengrass componenti sviluppati in collaborazione tra AWS e il partner. Quando aggiungi una fonte di dati partner, AWS IoT SiteWise creerà questo componente e lo distribuirà sul tuo gateway SiteWise Edge.

Note

È possibile aggiungere una fonte di dati per ogni partner in ogni gateway.

Per aggiungere un'origine dati partner, procedi come segue:

1. [Aggiungi una fonte di dati partner in SiteWise Edge](#)
2. Vai al portale web del partner, ove applicabile, e configura l'origine dati del partner in modo che si connetta al gateway SiteWise Edge.

Argomenti

- [Sicurezza](#)
- [Configura Docker sul tuo gateway SiteWise Edge](#)
- [Aggiungi una fonte di dati partner in SiteWise Edge](#)
- [SiteWise Opzioni di origine dati dei partner Edge Gateway](#)

Sicurezza

Nell'ambito del [modello di responsabilità condivisa](#) tra AWS i nostri clienti e i nostri partner, di seguito viene descritto chi è responsabile dei diversi aspetti della sicurezza:

Responsabilità del cliente

- Controllo del partner.
- Configurazione dell'accesso alla rete fornito al partner.
- Monitoraggio per un utilizzo ragionevole delle risorse della macchina SiteWise Edge gateway (CPU, memoria e file system).

AWS responsabilità

- Isolare il partner dalle risorse AWS cloud del cliente ad eccezione di quelle necessarie al partner. In questo caso, AWS IoT SiteWise ingestione.
- Limitazione della soluzione partner a un uso ragionevole delle risorse della macchina del gateway SiteWise Edge (CPU e memoria).

Responsabilità del partner

- Utilizzo di impostazioni predefinite sicure.
- Mantenimento della soluzione sicura nel tempo tramite patch e altri aggiornamenti appropriati.
- Mantenere riservati i dati dei clienti.

Configura Docker sul tuo gateway SiteWise Edge

AWS IoT SiteWise fornisce un'immagine Docker che consente di eseguire l'applicazione SiteWise Edge su varie piattaforme e ambienti. Questa Docker immagine incapsula tutti i componenti e le dipendenze necessari per raccogliere, elaborare e inviare dati dalle apparecchiature industriali al Cloud. AWS Utilizzando l'immagine Docker, puoi distribuire ed eseguire l'applicazione SiteWise Edge su host compatibili con Docker, come server, dispositivi edge o servizi container basati sul cloud.

Per aggiungere un'origine dati partner, è necessario installare [Docker Engine](#) 1.9.1 o versione successiva sul dispositivo locale.

Note

La versione 20.10 è l'ultima versione verificata per funzionare con il SiteWise software Edge gateway.

Verify Docker è installato

Per verificare Docker che sia installato, esegui il comando seguente da un terminale connesso al gateway SiteWise Edge:

```
docker info
```

Se il comando restituisce un `docker is not recognized` risultato o Docker è installata una versione precedente di, [installa Docker Engine](#) prima di continuare.

Configura Docker

L'utente di sistema che esegue un componente del Docker contenitore deve disporre delle autorizzazioni di root o di amministratore oppure è necessario configurarlo Docker per eseguirlo come utente non root o non amministratore.

Nei dispositivi Linux, è necessario aggiungere un `ggc_user` utente al gruppo senza il `docker` quale chiamare i comandi. Docker `sudo`

Per aggiungere `ggc_user` al `docker` gruppo l'utente non root utilizzato per eseguire i componenti del Docker contenitore, esegui il comando seguente:

```
sudo usermod -aG docker ggc_user
```

Per ulteriori informazioni, consulta i [passaggi successivi all'installazione di Linux per Docker Engine](#).

Aggiungi una fonte di dati partner in SiteWise Edge

Per connettere una fonte di dati partner al tuo gateway SiteWise Edge, aggiungila come fonte di dati. Quando lo aggiungi come fonte di dati, AWS IoT SiteWise distribuirà un AWS IoT Greengrass componente privato al tuo gateway SiteWise Edge.

Prerequisiti

Per aggiungere un'origine dati partner, devi fare quanto segue:

- Per EasyEdge eCloudRail, crea un account con il partner, quindi associa gli account.
- [Configura Docker sul tuo gateway SiteWise Edge](#)

Crea un gateway SiteWise Edge con una fonte di dati partner

Se desideri creare un nuovo gateway SiteWise Edge, completa i passaggi indicati in [Crea un gateway Edge ospitato autonomamente SiteWise](#). Dopo aver creato SiteWise Edge gateway, segui i passaggi indicati [Aggiungi una fonte di dati partner a un gateway SiteWise Edge esistente](#) per aggiungere una fonte di dati partner.

Aggiungi una fonte di dati partner a un gateway SiteWise Edge esistente

1. Passare alla [console AWS IoT SiteWise](#).
2. Nella barra di navigazione a sinistra, scegli Edge gateway nella sezione Edge.
3. Scegli il gateway SiteWise Edge a cui desideri connettere la fonte di dati del partner.
4. In Origini dati, scegli Aggiungi origine dati.
5. Nella schermata Aggiungi origine dati, scegli un tipo di origine per selezionare il partner che connette il tuo gateway SiteWise Edge. Ogni fonte di dati ha le proprie opzioni di configurazione. Esistono due categorie di fonti di dati: AWS fonti e fonti partner.

Utilizzando una fonte dati partner, puoi selezionare una fonte per gateway. Per un elenco delle opzioni di integrazione dei partner di origine dati, consulta [SiteWise Opzioni di origine dati dei partner Edge Gateway](#). Tieni presente che puoi aggiungere fino a 100 fonti di dati OPC UA (AWS fonti). Per iniziare a usare le fonti di dati OPC UA, vedi. [Fonti di dati OPC UA per gateway Edge AWS IoT SiteWise](#)

6. Immettete un nome per la fonte.
7. Seleziona la scheda della fonte dati in basso e segui la procedura di configurazione.

CloudRail

Gran parte della CloudRail configurazione viene eseguita nel CloudRail portale dopo aver salvato l'origine dati per il gateway SiteWise Edge. Tuttavia, è richiesta l'autorizzazione della connessione.

Note

La CloudRail connessione è disponibile solo su Linux.

1. [Crea un CloudRail account](#) a cui iniziare a connetterti AWS IoT SiteWise.
2. Assicurati che Docker sia installato sul tuo gateway. Per ulteriori informazioni, consulta [Configura Docker sul tuo gateway SiteWise Edge](#).
3. Leggi l'accordo di autorizzazione all'accesso e alla distribuzione, quindi scegli Autorizza. La selezione della casella concede al AWS partner l'accesso alla tua fonte di dati e AWS consente l'implementazione sul componente del partner.

Note

Il prefisso di misurazione, opzionale, è impostato all'interno del portale. CloudRail

Note

Il software del partner viene sviluppato, gestito e supportato dal AWS partner. AWS non è responsabile dell'interfaccia, della configurazione o del software.

Per ulteriori informazioni, consulta [CloudRail](#).

EasyEdge

Gran parte della EasyEdge configurazione viene eseguita nel EasyEdge portale dopo aver salvato l'origine dati per il gateway SiteWise Edge. Tuttavia, è richiesta l'autorizzazione della connessione.

 Note

La EasyEdge connessione è disponibile solo su Linux.

1. [Crea un EasyEdge account](#) a cui iniziare a connetterti AWS IoT SiteWise.
2. Assicurati che Docker sia installato sul tuo gateway. Per ulteriori informazioni, consulta [Configura Docker sul tuo gateway SiteWise Edge](#).
3. Leggi l'accordo di autorizzazione all'accesso e alla distribuzione, quindi scegli Autorizza. La selezione della casella concede al AWS partner l'accesso alla tua fonte di dati e AWS consente l'implementazione sul componente del partner.

 Note

Il prefisso di misurazione, opzionale, è impostato all'interno del portale. EasyEdge

 Note

Il software del partner viene sviluppato, gestito e supportato dal AWS partner. AWS non è responsabile dell'interfaccia, della configurazione o del software.

Per ulteriori informazioni, consulta [EasyEdge](#).

Litmus Edge

È possibile attivare la Litmus configurazione in due modi. Attiva Litmus Edge direttamente AWS IoT SiteWise utilizzando le informazioni del Litmus Edge Manager portale. In alternativa, puoi attivarlo Litmus Edge manualmenteLitmus Edge Manager. AWS IoT SiteWise

 Note

La Litmus Edge connessione è disponibile solo su Linux.

Da attivare utilizzando un codice Litmus Edge di attivazione su AWS IoT SiteWise

Utilizzare questa procedura per aggiungere una fonte di Litmus Edge dati con un codice di Litmus Edge attivazione su Console AWS IoT SiteWise.

1. Seleziona Attiva ora usando un codice. Vengono visualizzate opzioni di configurazione aggiuntive.
2. Inserisci il Litmus Edge Manager per connetterti Litmus Edge al tuo gateway SiteWise Edge. Per ulteriori informazioni, consulta la [Fase 3a: Impostazione dell'endpoint per la gestione dei dati e dei dispositivi](#) nella Litmus Edge Manager documentazione.
3. Fornisci il codice di Litmus Edge Manager attivazione su cui eseguire l'attivazione Litmus Edge AWS IoT SiteWise
4. Facoltativamente, fornisci AWS IoT SiteWise il certificato Litmus Edge Manager CA. Il certificato Litmus Edge impedisce l'attivazione su un dispositivo non autorizzato. Litmus Edge Manager
5. Assicurati che Docker sia installato sul tuo gateway. Per ulteriori informazioni, consulta [Configura Docker sul tuo gateway SiteWise Edge](#).

 Note

AWS IoT SiteWise distribuisce l'applicazione partner come contenitore Docker. L'applicazione viene distribuita con NET_ADMIN funzionalità tali da consentire la gestione del contenitore Litmus Edge Docker tramite. Litmus Edge Manager Litmus Edgerichiede questo accesso privilegiato per funzionare sui tuoi dispositivi. Per ulteriori informazioni sui requisiti di Litmus Edge Docker, consulta [l'installazione di Docker](#) nella QuickStart Guida nella documentazione di Litmus Edge.

6. Leggi l'accordo di autorizzazione all'accesso e alla distribuzione, quindi scegli Autorizza. La selezione della casella concede al AWS partner l'accesso alla tua fonte di dati e AWS consente l'implementazione sul componente del partner.

Da attivare manualmente tramite Litmus Edge

1. Seleziona Attiva in seguito Litmus Edge.
2. Assicurati che Docker sia installato sul tuo gateway. Per ulteriori informazioni, consulta [Configura Docker sul tuo gateway SiteWise Edge](#).

 Note

AWS IoT SiteWise distribuisce l'applicazione partner come contenitore Docker. L'applicazione viene distribuita con NET_ADMIN funzionalità tali da consentire la gestione del contenitore Litmus Edge Docker tramite. Litmus Edge Manager Litmus Edgerichiede questo accesso privilegiato per funzionare sui tuoi dispositivi. Per ulteriori informazioni sui requisiti di Litmus Edge Docker, consulta [l'installazione di Docker](#) nella QuickStart Guida nella documentazione di Litmus Edge.

3. Leggi l'accordo di autorizzazione all'accesso e alla distribuzione, quindi scegli Autorizza. La selezione della casella concede al AWS partner l'accesso alla tua fonte di dati e AWS consente l'implementazione sul componente del partner.
4. Una volta completata la distribuzione, segui le istruzioni dell'[interfaccia utente Web Access the Litmus Edge](#) contenute nella documentazione. Litmus Edge QuickStart Guide

 Note

Il software del partner viene sviluppato, gestito e supportato dal AWS partner. AWS non è responsabile dell'interfaccia, della configurazione o del software.

Per ulteriori informazioni, consulta [Litmus Edge](#).

8. Scegli Save (Salva).

SiteWise Opzioni di origine dati dei partner Edge Gateway

AWS IoT SiteWise consente di collegare e inserire dati da varie fonti di dati dei partner, come apparecchiature industriali, sensori e altri sistemi di terze parti. Per connettere una fonte di dati partner, devi seguire alcuni passaggi, tra cui la configurazione dell'origine dati a cui inviare i dati AWS

IoT SiteWise, l'impostazione delle autorizzazioni e dell'autenticazione necessarie e la mappatura dei dati ai tuoi modelli di asset. Questo processo garantisce che i dati dei partner siano perfettamente integrati nel vostro AWS IoT SiteWise ambiente, consentendovi di monitorarli e analizzarli insieme alle altre fonti di dati.

Questa sezione elenca i partner disponibili per l'integrazione di fonti di dati di terze parti sui gateway SiteWise Edge. Utilizza le informazioni riportate di seguito per configurare un'origine dati partner.

Note

Puoi aggiungere una fonte di dati per ogni partner in ogni gateway

CloudRail

Portale:

<https://devices.cloudrail.com/>

Requisiti

Per ulteriori informazioni sui CloudRail requisiti, consulta le [domande frequenti](#) sul CloudRail sito Web.

CloudRaildocumentazione:

[Edge Computing: SiteWise Edge](#)

EasyEdge

Portale:

<https://studio.easyedge.io/>

Requisiti

[EasyEdgerequisiti](#): informazioni sui EasyEdge requisiti, inclusi gli endpoint e le porte necessari per la configurazione del firewall. Nota: è necessario un EasyEdge account per accedere a questa documentazione.

EasyEdgedocumentazione:

[EasyEdgeper AWS](#)

Litmus Edge

Accesso a Litmus Edge Manager:

Per accedere a Litmus Edge, configura un account [Litmus Edge Manager](#).

Requisiti

[Litmus EdgeRequisiti](#): configurazioni e requisiti di sistema consigliati per l'implementazione.

Litmus Edge

Litmusdocumentazione:

- [Integrazione con AWS IoT SiteWise](#)
- [Documentazione Litmus Edge](#)

AWS IoT Greengrass componenti per AWS IoT SiteWise Edge

SiteWise Edge utilizza AWS IoT Greengrass componenti per raccogliere, elaborare e trasmettere dati industriali all'edge. Questi componenti interagiscono per consentire l'elaborazione locale dei dati e la perfetta integrazione con il servizio AWS IoT SiteWise cloud.

SiteWise Editore IoT

Il componente SiteWise editore IoT (`aws.iot.SiteWiseEdgePublisher`) è responsabile di:

- Trasmissione sicura dei dati raccolti al servizio cloud AWS IoT SiteWise
- Gestione del buffering dei dati e dei nuovi tentativi durante i problemi di connettività

Per ulteriori informazioni sulla configurazione dell'editore per SiteWise Edge, consulta [Configura il componente publisher AWS IoT SiteWise](#). Inoltre, per ulteriori informazioni sul componente publisher, consulta [IoT SiteWise Publisher](#) nella AWS IoT Greengrass Version 2 Developer Guide.

SiteWise Processore IoT

Il componente SiteWise del processore IoT (`aws.iot.SiteWiseEdgeProcessor`) esegue le seguenti attività:

- Esecuzione di trasformazioni e calcoli dei dati all'edge
- Implementazione locale delle definizioni e dei calcoli delle proprietà degli asset
- Riduzione del volume dei dati aggregando o filtrando i dati prima della trasmissione

Per ulteriori informazioni sul componente processore, consulta [SiteWise Processore IoT](#) nella AWS IoT Greengrass Version 2 Developer Guide.

Collettore IoT SiteWise OPC UA

Il componente IoT SiteWise OPC UA collector (`aws.iot.SiteWiseEdgeCollectorOpcua`) è progettato per:

- Connect ai server OPC UA in ambienti industriali
- Raccogli dati dalle fonti di dati OPC UA in modo efficiente
- Trasforma i dati OPC UA in un formato compatibile con AWS IoT SiteWise

Per ulteriori informazioni sul componente OPC UA Collector, consulta [IoT SiteWise OPC UA Collector](#) nella Developer Guide.AWS IoT Greengrass Version 2

Simulatore di sorgenti dati IoT SiteWise OPC UA

Il componente del simulatore di origine dati IoT SiteWise OPC UA (`aws.iot.SiteWiseEdgeOpcuaDataSourceSimulator`) offre le seguenti funzionalità:

- Avvia un server OPC UA locale che genera dati di esempio
- Simula una fonte di dati che può essere letta dal componente AWS IoT SiteWise OPC UA Collector su un gateway AWS IoT SiteWise
- Consente l'esplorazione delle AWS IoT SiteWise funzionalità utilizzando i dati di esempio generati

Questo componente è particolarmente utile per scopi di test e sviluppo, poiché consente di simulare fonti di dati industriali senza la necessità di apparecchiature fisiche.

Per ulteriori informazioni sul componente di simulazione dell'origine dati, consulta il [simulatore di sorgenti dati IoT SiteWise OPC UA](#) nella Guida per gli AWS IoT Greengrass Version 2 sviluppatori.

Questi AWS IoT Greengrass componenti funzionano per abilitare SiteWise la funzionalità Edge. L' SiteWise editore IoT garantisce l'invio affidabile dei dati al cloud, il SiteWise processore IoT gestisce i calcoli locali e l'ottimizzazione dei dati e il collettore IoT SiteWise OPC UA facilita l'integrazione con i protocolli industriali comuni.

Note

Per utilizzare questi componenti, è necessario averli installati AWS IoT Greengrass V2 o successivamente installati sui dispositivi periferici. La corretta configurazione di ogni componente è importante per prestazioni ottimali di SiteWise Edge.

Filtra le risorse su un gateway SiteWise Edge

Puoi utilizzare il filtro edge per gestire in modo più efficiente le tue risorse inviando solo un sottoinsieme di risorse a uno specifico gateway SiteWise Edge da utilizzare nell'elaborazione dei dati. Se le risorse sono disposte in una struttura ad albero, o padre-figlio, è possibile impostare una policy IAM collegata al ruolo IAM di un gateway SiteWise Edge che consenta solo l'invio della radice dell'albero, o genitore, e dei relativi figli a un gateway Edge specifico. SiteWise

Note

Se stai organizzando le risorse esistenti in una struttura ad albero, dopo aver creato la struttura, accedi a ogni risorsa esistente che hai aggiunto alla struttura e scegli Modifica, quindi scegli Salva per assicurarti che AWS IoT SiteWise riconosca la nuova struttura.

Imposta il filtraggio dei bordi

Configura il filtro edge sul tuo gateway SiteWise Edge aggiungendo la seguente policy IAM al ruolo IAM del gateway SiteWise Edge, sostituendola `<root-asset-id>` con l'ID dell'asset root che desideri inviare al gateway SiteWise Edge.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssociatedAssets"
      ],
      "Resource": "arn:aws:iotsitewise:*:*:asset/*",
      "Condition": {
        "StringNotLike": {
          "iotsitewise:assetHierarchyPath": "/<root-asset-id>*"
        }
      }
    }
  ]
}
```

Se al momento ci sono risorse sul gateway SiteWise Edge che desideri rimuovere, accedi al gateway SiteWise Edge ed esegui il comando seguente per forzare la sincronizzazione del gateway SiteWise Edge AWS IoT SiteWise eliminando la cache.

```
sudo rm /greengrass/v2/work/aws.iot.SiteWiseEdgeProcessor/sync-app/  
sync_resource_bundles/edge.json
```

Configura il supporto proxy e gestisci gli archivi affidabili per AWS IoT SiteWise Edge

In AWS IoT SiteWise Edge, configura e gestisci gli archivi di fiducia per configurare il supporto proxy per i tuoi dispositivi edge. Per prima cosa, configura la configurazione del proxy, quindi configura gli archivi attendibili. È possibile configurare i trust store durante l'installazione del gateway o manualmente dopo la creazione del gateway.

- Proxy: facilitano la connettività tra i dispositivi e i AWS servizi periferici in vari ambienti di rete.
- Trust Stores: garantisce connessioni sicure gestendo certificati affidabili. Le configurazioni corrette consentono di rispettare le politiche di sicurezza della rete, abilitare la comunicazione in ambienti di rete con restrizioni e ottimizzare il trasferimento dei dati tra dispositivi periferici e servizi cloud.

SiteWise Edge utilizza più archivi di fiducia per diversi tipi di componenti, garantendo un flusso di dati sicuro ed efficiente dai dispositivi periferici al cloud. È possibile configurare trust store e proxy su un gateway esistente o durante il processo di installazione durante la creazione di un nuovo gateway.

Requisiti per le configurazioni trust store e proxy

Prima di configurare un trust store o installare SiteWise Edge con le impostazioni proxy, assicurati di soddisfare i prerequisiti. Esistono diversi requisiti di implementazione in base all'utilizzo dei componenti e ai requisiti di funzionalità.

Requisiti di supporto proxy

- L'URL del tuo server proxy. L'URL deve includere le informazioni sull'utente, il numero di porta dell'host. Ad esempio, `scheme://[userinfo@]host[:port]`.
 - `scheme`— Deve essere HTTP o HTTPS
 - (Facoltativo)`userinfo`: informazioni sul nome utente e sulla password
 - `host`— Il nome host o l'indirizzo IP del server proxy

- `port`— Il numero di porta
- Un elenco di indirizzi per bypassare il proxy.
- (Facoltativo) Il file del certificato CA proxy se si utilizza un proxy HTTPS con un certificato autofirmato.

Requisiti del Trust Store

- Per una funzionalità completa del pacchetto di elaborazione dati con proxy HTTPS, è necessario aggiornare tutti e tre gli archivi attendibili.
- Se utilizzi solo il raccoglitore IoT SiteWise OPC UA e l' SiteWise editore IoT, aggiorna i certificati AWS IoT Greengrass Core e Java trust store alla versione più recente.

Le migliori pratiche per le configurazioni edge del trust store e del server proxy

Per una manutenzione continua e per mantenere il massimo livello di sicurezza nell'ambiente perimetrale:

- Rivedi e aggiorna regolarmente le impostazioni del proxy per allinearle ai requisiti di sicurezza della rete.
- Monitora la connettività del gateway e il flusso di dati per garantire una corretta comunicazione proxy
- Gestite e aggiornate gli archivi di fiducia in base alle politiche di gestione dei certificati della vostra organizzazione
- Puoi implementare e seguire le nostre migliori pratiche consigliate per comunicazioni sicure in ambienti periferici, come:
- Documenta le configurazioni del proxy e del trust store per una visibilità operativa
- Segui le pratiche di sicurezza della tua organizzazione per la gestione delle credenziali

Queste pratiche aiutano a mantenere operazioni sicure e affidabili per i gateway SiteWise Edge, pur rimanendo in linea con le politiche di sicurezza più ampie.

Configura le impostazioni proxy durante AWS IoT SiteWise l'installazione del gateway Edge

È possibile configurare AWS IoT SiteWise Edge in modo che funzioni con un server proxy durante l'installazione del gateway. Lo script di installazione supporta sia i proxy HTTP che HTTPS e può configurare automaticamente gli archivi di fiducia per connessioni proxy sicure.

Quando si esegue lo script di installazione con le impostazioni proxy, esegue diverse attività importanti:

- Convalida il formato e i parametri dell'URL del proxy per garantire che siano specificati correttamente.
- Scarica e installa le dipendenze richieste tramite il proxy configurato.
- Se viene fornito un certificato CA proxy, questo viene aggiunto al certificato CA AWS IoT Greengrass principale e importato in Java. KeyStore
- Configura AWS IoT Greengrass (utilizzato da SiteWise Edge) per utilizzare il proxy per tutte le connessioni in uscita.
- Completa l'installazione di SiteWise Edge con le configurazioni proxy e trust store appropriate.

Per configurare le impostazioni proxy durante l'installazione del software gateway

1. Crea un gateway SiteWise Edge. Per ulteriori informazioni, consultare [Crea un gateway Edge ospitato autonomamente SiteWise](#) e [Installa il software AWS IoT SiteWise Edge gateway sul tuo dispositivo locale](#).
2. Esegui lo script di installazione con le impostazioni proxy appropriate per il tuo ambiente. Sostituisci i segnaposto con le informazioni proxy specifiche

Sostituisci ciascuno dei seguenti elementi:

- `-p, --proxy-url` — L'URL del server proxy. L'URL deve essere uno `http` o `https`.
- `-n, --no-proxy` — Un elenco di indirizzi separati da virgole per aggirare il proxy.
- (Facoltativo) `-c, --proxy-ca-cert` — Percorso del file del certificato CA del proxy.
- (Facoltativo) `-j, --javastorepass` — La KeyStore password Java. La password predefinita è `changeit`.

Linux

Per i sistemi Linux, utilizzate la seguente struttura di comandi:

```
sudo ./install.sh -p proxy-url -n no-proxy-addresses [-c proxy-ca-cert-path] [-j javastorepass]
```

Windows

Per Microsoft Windows i sistemi che utilizzano PowerShell, usa questa struttura di comandi:

```
.\install.ps1 -ProxyUrl proxy-url -NoProxyAddresses no-proxy-addresses [-ProxyCaCertPath proxy-ca-cert-path] [-JavaStorePass javastorepass]
```

Risoluzione dei problemi durante l'installazione con proxy

Per ulteriori informazioni sulla risoluzione dei problemi di trust store relativi a un gateway SiteWise Edge, consulta [Problemi di installazione con proxy](#).

Configura manualmente i trust store per il supporto del proxy HTTPS in AWS IoT SiteWise Edge

Quando configuri i componenti AWS IoT SiteWise Edge per la connessione tramite un proxy HTTPS, aggiungi il certificato del server proxy agli archivi di fiducia appropriati. SiteWise Edge utilizza più archivi di fiducia per proteggere le comunicazioni. Esistono tre trust store e il loro utilizzo dipende dal tipo di componente SiteWise Edge utilizzato nell'implementazione del gateway.

Gli archivi di fiducia vengono aggiornati automaticamente durante il processo di installazione quando vengono fornite le impostazioni proxy.

- [Configura un AWS IoT Greengrass Core Component Trust Store](#)— Il certificato CA AWS IoT Greengrass principale è incluso negli archivi attendibili per verificare l'autenticità dei AWS servizi.

Questo trust store consente AWS IoT Greengrass ai componenti di comunicare in modo sicuro con AWS i servizi tramite il proxy, verificando al contempo l'autenticità di tali servizi.

- [Configura un archivio di fiducia dei componenti basato su Java](#)— Java KeyStore (JKS) è il principale trust store utilizzato dai componenti basati su Java per le connessioni. SSL/TLS

Le applicazioni Java si affidano al JKS per stabilire connessioni sicure. Ad esempio, se utilizzi l' SiteWise editore IoT o il raccoglitore IoT SiteWise OPC UA, basati su Java, dovrai configurare questo trust store. Ciò garantisce che questi componenti possano comunicare in modo sicuro tramite il proxy HTTPS durante l'invio di dati al cloud o la raccolta di dati dai server OPC UA.

- [Configurazione dell'archivio di fiducia dei componenti a livello di sistema](#)— Quando si utilizzano i proxy HTTPS, i relativi certificati devono essere aggiunti agli archivi di fiducia appropriati per consentire connessioni sicure.

Quando si utilizzano i proxy HTTPS, i relativi certificati devono essere aggiunti agli archivi di fiducia appropriati per consentire connessioni sicure. Ciò è necessario perché i componenti a livello di sistema, spesso scritti in linguaggi come Rust o Go, si basano sull'archivio di fiducia del sistema anziché sul JKS di Java. Ad esempio, se utilizzi utilità di sistema che devono comunicare tramite il proxy (ad esempio per gli aggiornamenti software o la sincronizzazione dell'ora), dovrai configurare il trust store a livello di sistema. Ciò garantisce che questi componenti e utilità possano stabilire connessioni sicure tramite il proxy.

Configura un AWS IoT Greengrass Core Component Trust Store

Per le funzioni AWS IoT Greengrass Core che utilizzano la CA root di Amazon:

1. Individua il file del certificato in `/greengrass/v2/AmazonRootCA1.pem`
2. Aggiungi il certificato radice del proxy HTTPS (autofirmato) a questo file.

```
-----BEGIN CERTIFICATE-----
MIIETCCA 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
ADA5MQswCQYDVQQGEwJVUzEPMA0tMVT8Q0tPHRh8jrdkGA1UEChMGMDV3QQDExBBKW
... content of root CA certificate ...
o/ufQJQWUCyziar1hem9uMRkwFwYVPSHCb2XV4cdFyQzR1K1dZwgJcIQ6XUDgHaa
```

```
5MsI+yMRQ+hDaXJioblXgjUka642M4UwtBV8oK2xJNDd2ZhwLnoQdeXeGADKkpy  
rqXRfKoQnoZsG4q5WTP46EXAMPLE  
-----END CERTIFICATE-----
```

Configura il proxy HTTPS su un gateway stabilito

È possibile aggiungere il supporto proxy a un gateway consolidato connettendosi alla porta 443 anziché alla porta 8883. Per ulteriori informazioni sull'utilizzo di un server proxy, consulta [Connect on port 443 o tramite un proxy di rete](#) nella AWS IoT Greengrass Version 2 Developer Guide. Se si crea un nuovo gateway, è possibile impostare la configurazione del proxy durante l'installazione del gateway. Per ulteriori informazioni, consulta [Configura le impostazioni proxy durante AWS IoT SiteWise l'installazione del gateway Edge](#).

Quando si utilizza un proxy HTTPS con AWS IoT Greengrass SiteWise Edge, il software sceglie automaticamente tra HTTP e HTTPS per le connessioni proxy in base all'URL fornito.

Important

Aggiorna tutti gli archivi di fiducia richiesti prima di tentare la connessione tramite un proxy HTTPS.

Configura un archivio di fiducia dei componenti basato su Java

Per l' SiteWise editore IoT, il collettore IoT SiteWise OPC UA e i servizi Java inclusi nel pacchetto di elaborazione dati, la posizione predefinita del Java trust store è `$JAVA_HOME/jre/lib/security/cacerts`

Per aggiungere un certificato

1. Crea un file per archiviare il certificato del server proxy, ad esempio `proxy.crt`.

Note

Crea il file in anticipo utilizzando il certificato del server proxy.

2. Aggiungi il file all'archivio di fiducia di Java utilizzando il seguente comando:

```
sudo keytool -import -alias proxyCert -keystore /usr/lib/jvm/java-11-openjdk-amd64/lib/security/cacerts -file proxy.crt
```

3. Quando richiesto, usa la password predefinita: `changeit`

Configurazione dell'archivio di fiducia dei componenti a livello di sistema

Per i componenti scritti in Rust, Go e altri linguaggi che utilizzano il trust store di sistema:

Linux

Sistemi Linux: aggiungi certificati a `/etc/ssl/certs/ca-certificates.crt`

Windows

Microsoft Windows sistemi: per configurare il trust store, segui la procedura [Certificate Store](#) nella documentazione di Microsoft Ignite.

Windows offre diversi archivi di certificati, inclusi archivi separati per gli ambiti utente e computer, ciascuno con diversi archivi secondari. Per la maggior parte delle configurazioni SiteWise Edge, consigliamo di aggiungere certificati all'archivio. `COMPUTER | Trusted Root Certification Authorities` Tuttavia, a seconda dei requisiti di configurazione e sicurezza specifici, potrebbe essere necessario utilizzare un archivio diverso.

Risoluzione dei problemi relativi al trust store

Per ulteriori informazioni sulla risoluzione dei problemi di trust store relativi a un gateway SiteWise Edge, consulta [Problemi relativi al Trust Store](#).

Utilizzare AWS IoT SiteWise APIs sul bordo

AWS IoT SiteWise ne fornisce un sottoinsieme APIs, oltre a uno specifico APIs per l'edge, che consente un'interazione senza interruzioni con i modelli di asset e gli asset associati distribuiti sull'edge. Questi modelli di asset devono essere configurati per funzionare sull'edge. Per ulteriori informazioni, consulta [Configura un modello di asset per l'elaborazione dei dati su SiteWise Edge](#) le istruzioni dettagliate su questo processo di configurazione.

Dopo averli configurati APIs, potete recuperare dati completi sui modelli di asset e sui singoli asset. Il recupero di informazioni su modelli di asset, asset, dashboard, portale e progetto può aiutarvi a

monitorare portali e dashboard implementati e ad accedere ai dati sugli asset raccolti a livello edge. Ciò fornisce un host centrale nella rete con cui interagire AWS IoT SiteWise senza richiedere una chiamata all'API Web.

Argomenti

- [Tutti i dispositivi AWS IoT SiteWise Edge disponibili APIs](#)
- [Solo Edge da utilizzare con dispositivi edge APIs AWS IoT SiteWise](#)
- [Abilita CORS su Edge AWS IoT SiteWise APIs](#)
- [Configura i timeout di sessione per Edge AWS IoT SiteWise](#)
- [Tutorial: Elenca i modelli di asset su un gateway AWS IoT SiteWise Edge](#)

Tutti i dispositivi AWS IoT SiteWise Edge disponibili APIs

AWS IoT SiteWise offre una varietà di dispositivi edge APIs da utilizzare in modo da poter completare le attività localmente sul dispositivo. Alcune delle soluzioni edge disponibili APIs includono il recupero di modelli di asset, la creazione e l'aggiornamento delle proprietà degli asset e l'invio di flussi di dati al cloud. Sfruttandoli APIs, è possibile creare soluzioni in grado di funzionare in ambienti con connettività di rete intermittente o limitata.

Disponibile AWS IoT SiteWise APIs

Sui dispositivi edge AWS IoT SiteWise APIs sono disponibili quanto segue:

- [ListAssetModels](#)
- [DescribeAssetModel](#)
- [ListAssets](#)
- [DescribeAsset](#)
- [DescribeAssetProperty](#)
- [ListAssociatedAssets](#)
- [GetAssetPropertyAggregates](#)
- [GetAssetPropertyValue](#)
- [GetAssetPropertyValueHistory](#)
- [ListDashboards](#)
- [ListPortals](#)
- [ListProjectAssets](#)

- [ListProjects](#)
- [DescribeDashboard](#)
- [DescribePortal](#)
- [DescribeProject](#)

Disponibile solo per gli spigoli APIs

Quanto segue APIs viene utilizzato localmente sui dispositivi periferici:

- [Autenticazione](#)— Usa questa API per ottenere le credenziali temporanee SigV4 che utilizzerai per effettuare chiamate API.

Solo Edge da utilizzare con dispositivi edge APIs AWS IoT SiteWise

Oltre a AWS IoT SiteWise APIs quelli disponibili sul bordo, ce ne sono di specifici. Quelle specifiche per gli spigoli sono descritte APIs di seguito.

Autenticazione

Ottiene le credenziali dal gateway SiteWise Edge. Dovrai aggiungere utenti locali o connetterti al sistema tramite LDAP o un pool di utenti Linux. Per ulteriori informazioni sull'aggiunta di utenti, consulta [LDAP o pool](#) di [utenti Linux](#).

Sintassi della richiesta

```
POST /authenticate HTTP/1.1
Content-type: application/json
{
  "username": "string",
  "password": "string",
  "authMechanism": "string"
}
```

Parametri di richiesta URI

La richiesta non utilizza parametri URI.

Corpo della richiesta

La richiesta accetta i seguenti dati in formato JSON.

username

Il nome utente utilizzato per convalidare la chiamata di richiesta.

Tipo: stringa

Campo obbligatorio: sì

password

La password dell'utente che richiede le credenziali.

Tipo: stringa

Campo obbligatorio: sì

Meccanismo di autenticazione

Il metodo di autenticazione per convalidare questo utente nell'host.

Tipo: stringa

Valori validi: ldap, linux, winnt

Campo obbligatorio: sì

Sintassi della risposta

```
HTTP/1.1 200
Content-type: application/json
{
  "accessKeyId": "string",
  "secretAccessKey": "string",
  "sessionToken": "string",
  "region": "edge"
}
```

Elementi di risposta

Se l'operazione riesce, il servizio restituisce una risposta HTTP 200.

I seguenti dati vengono restituiti in formato JSON.

accessKeyId

L'ID della chiave di accesso che identifica le credenziali di sicurezza temporanee.

Vincoli di lunghezza: lunghezza minima di 16. La lunghezza massima è 128 caratteri.

Modello: `[\w]*`

`secretAccessKey`

La chiave di accesso segreta che può essere utilizzata per firmare le richieste.

Tipo: stringa

`sessionToken`

Il token che gli utenti devono passare all'API del servizio per utilizzare le credenziali temporanee.

Tipo: stringa

`Regione`

La regione a cui ti rivolgi per le chiamate API.

Tipo: CONSTANT - edge

Errori

`IllegalArgumentException`

La richiesta è stata respinta perché il corpo del documento fornito non era valido. Il messaggio di errore descrive l'errore specifico.

Codice di stato HTTP: 400

`AccessDeniedException`

L'utente non dispone di credenziali valide basate sull'attuale Identity Provider. Il messaggio di errore descrive il meccanismo di autenticazione.

Codice di stato HTTP: 403

`TooManyRequestsException`

La richiesta ha raggiunto il limite di tentativi di autenticazione. Il messaggio di errore contiene la quantità di tempo di attesa prima che vengano effettuati nuovi tentativi di autenticazione.

Codice di stato HTTP: 429

Abilita CORS su Edge AWS IoT SiteWise APIs

L'abilitazione di CORS (Cross-Origin Resource Sharing) su AWS IoT SiteWise Edge APIs consente alle applicazioni Web di comunicare direttamente con diversi APIs domini. Ciò consente un'integrazione perfetta, lo scambio di dati in tempo reale e l'accesso ai dati tra domini senza server o soluzioni alternative intermedie. Le impostazioni CORS possono essere configurate per specificare le origini consentite, garantendo un accesso controllato tra le origini.

Note

CORS è disponibile per la versione 3.3.1 e successive di Questa funzione è disponibile per la versione 3.3.1 e successive del componente. `aws.iot.SiteWiseEdgeProcessor` Per ulteriori informazioni, consulta il [AWS IoT SiteWise processore nella Guida](#) per gli AWS IoT Greengrass Version 2 sviluppatori.

Per abilitare CORS su Edge SiteWise APIs

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Edge gateway.
3. Seleziona il gateway SiteWise Edge per il quale desideri abilitare CORS. È possibile abilitare CORS sul tipo di AWS IoT Greengrass V2 distribuzione.
4. Nella sezione Configurazione del gateway, scegli il dispositivo principale Greengrass associato.
5. Nella scheda Implementazioni, in Dispositivi Greengrass, seleziona il link di distribuzione appropriato.
6. In Azioni scegli Rivedi, quindi Rivedi distribuzione.

Important

La creazione di una configurazione modificata abilitata per CORS sostituisce la configurazione corrente del dispositivo.

7. Nel passaggio 1, Specificare l'obiettivo, fornire un nome opzionale per identificare la distribuzione.
8. Nel passaggio 2, Selezione dei componenti: facoltativo, è possibile lasciare tutte le selezioni correnti così come sono e scegliere Avanti.

9. Nel passaggio 3, Configura i componenti: opzionale, seleziona `aws.iot. SiteWiseEdgeProcessore` scegli Configura componente.
10. Nella sezione Aggiornamento della configurazione, in Configurazione da unire, inserisci il seguente codice JSON:

```
{  
  "AWS_SITEWISE_EDGE_ACCESS_CONTROL_ALLOW_ORIGIN": "*"   
}
```

Note

L'utilizzo di `*` as come valore per `AWS_SITEWISE_EDGE_ACCESS_CONTROL_ALLOW_ORIGIN` consente tutte le origini. Per gli ambienti di produzione, si consiglia di specificare l'origine esatta URLs per una maggiore sicurezza.

11. Scegli Conferma.
12. Scegli Avanti per procedere con i passaggi rimanenti fino ad arrivare alla Fase 5, Revisione.
13. Controlla le modifiche alla configurazione, quindi scegli Deploy per applicare le modifiche al tuo gateway SiteWise Edge.

Note

In alternativa, puoi abilitare CORS impostando la variabile `AWS_SITEWISE_EDGE_ACCESS_CONTROL_ALLOW_ORIGIN` ambientale globale `*` su sul tuo AWS IoT SiteWise gateway.

Note

Per un proxy autenticato, `userinfo` deve essere incluso nel `url` campo della configurazione del proxy anziché come campi separati `username` e `password`.

Una volta completata la distribuzione, CORS viene abilitato sull'API SiteWise Edge, consentendo alle origini specificate di inviare richieste multiorigine all'API.

Configura i timeout di sessione per Edge AWS IoT SiteWise

SiteWise Edge consente di configurare i timeout di sessione per l' SiteWise API Edge. Questa funzionalità migliora la sicurezza interrompendo automaticamente le sessioni inattive dopo un periodo di tempo specificato. Questa sezione guida l'utente attraverso il processo di configurazione del timeout della sessione utilizzando. Console AWS IoT SiteWise

Note

La configurazione del timeout della sessione è disponibile per la versione 3.4.0 e successive del componente. `aws.iot.SiteWiseEdgeProcessor` Per ulteriori informazioni, consulta il [AWS IoT SiteWise processore nella Guida](#) per gli AWS IoT Greengrass Version 2 sviluppatori.

Per configurare un timeout di sessione per un gateway SiteWise Edge

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Edge gateway.
3. Scegli il gateway SiteWise Edge in cui desideri configurare il timeout della sessione.

Note

È possibile configurare il timeout della sessione in base al tipo di AWS IoT Greengrass V2 distribuzione.

4. Nella sezione Configurazione del gateway, scegli il dispositivo principale Greengrass associato.
5. Nella scheda Implementazioni, in Dispositivi Greengrass, seleziona il link di distribuzione appropriato.
6. In Azioni scegli Rivedi. Leggi l'avviso, quindi scegli Rivedi la distribuzione.

Important

La creazione di una configurazione di timeout della sessione rivista sostituisce la configurazione corrente del dispositivo.

7. Nel passaggio 1, Specificate l'obiettivo, fornite un nome opzionale per identificare la distribuzione rivista, quindi scegliete Avanti.

8. Nel passaggio 2, Selezione dei componenti: facoltativo, è possibile lasciare tutte le selezioni correnti così come sono e scegliere Avanti.
9. Nel passaggio 3, Configura i componenti: opzionale, seleziona aws.iot. SiteWiseEdgeProcessore scegli Configura componente.
10. Nella sezione Aggiornamento della configurazione, in Configurazione da unire, inserisci il seguente codice JSON:

```
{  
  "AWS_SITEWISE_EDGE_SESSION_TIMEOUT_MINUTES": "240"  
}
```

11. Imposta il valore AWS_SITEWISE_EDGE_SESSION_TIMEOUT_MINUTES in minuti. I valori di timeout della sessione possono essere compresi tra 1 minuto e 10080 minuti (7 giorni). Il valore predefinito è 240 minuti (4 ore).
12. Scegli Conferma.
13. Scegli Avanti per procedere con i passaggi rimanenti fino ad arrivare al Passaggio 5, Revisione.
14. Controlla le modifiche alla configurazione, quindi scegli Deploy per applicare le modifiche al tuo gateway SiteWise Edge.

Note

In alternativa, puoi configurare il timeout della sessione impostando la variabile ambientale globale AWS_SITEWISE_EDGE_SESSION_TIMEOUT_MINUTES sul valore desiderato (in minuti) sul tuo gateway Edge. SiteWise

Una volta completata la distribuzione, la nuova configurazione del timeout della sessione viene applicata all'API Edge. SiteWise

Tutorial: Elenca i modelli di asset su un gateway AWS IoT SiteWise Edge

Puoi utilizzare un AWS IoT SiteWise APIs sottoinsieme di quelli disponibili e quelli specifici APIs per Edge per interagire con i modelli di asset e i relativi asset sull'edge. Questo tutorial ti illustrerà come ottenere credenziali temporanee per un gateway AWS IoT SiteWise Edge e ottenere un elenco dei modelli di asset sul gateway Edge. SiteWise

Prerequisiti

Nei passaggi di questo tutorial puoi utilizzare una varietà di strumenti. Per utilizzare questi strumenti, assicurati di aver installato i prerequisiti corrispondenti.

Per completare questo tutorial, è necessario quanto segue:

- A: distribuito e funzionante [AWS IoT SiteWise Requisiti del gateway Edge self-hosted](#)
- Accesso al gateway SiteWise Edge nella stessa rete tramite la porta 443.
- [OpenSSL installato](#)
- [\(AWS OpsHub per AWS IoT SiteWise\) L'AWS OpsHub applicazione for AWS IoT SiteWise](#)
- (curl) [curl](#) installato
- [\(Python\) urllib3 installato](#)
- [\(Python\) Python3 installato](#)
- [\(Python\) Boto3 installato](#)
- (Python) installato [BotoCore](#)

Fase 1: Ottenete un certificato firmato per il servizio SiteWise Edge Gateway

Per stabilire una connessione TLS al gateway APIs Available at the SiteWise Edge, è necessario un certificato affidabile. È possibile generare questo certificato utilizzando un OpenSSL o for AWS OpsHub . AWS IoT SiteWise

OpenSSL

Note

È necessario che sia installato [OpenSSL](#) per eseguire questo comando.

Apri un terminale ed esegui il comando seguente per ottenere un certificato firmato dal gateway SiteWise Edge. Sostituisci `<sitewise_gateway_ip>` con l'IP del gateway SiteWise Edge.

```
openssl s_client -connect <sitewise_gateway_ip>:443 </dev/null 2>/dev/null | openssl  
x509 -outform PEM > GatewayCert.pem
```

AWS OpsHub for AWS IoT SiteWise

Puoi usare AWS OpsHub per AWS IoT SiteWise. Per ulteriori informazioni, consulta [Gestione dei gateway SiteWise Edge](#).

In questo tutorial viene utilizzato il percorso assoluto del certificato gateway SiteWise Edge scaricato. Esegui il comando seguente per esportare il percorso completo del certificato, sostituendolo `<absolute_path_to_certificate>` con il percorso del certificato:

```
export PATH_TO_CERTIFICATE='<absolute_path_to_certificate>'
```

Passaggio 2: ottieni il nome host SiteWise del gateway Edge

Note

È necessario che sia installato [OpenSSL](#) per eseguire questo comando.

Per completare il tutorial è necessario il nome host del gateway Edge. SiteWise Per ottenere il nome host del gateway SiteWise Edge, esegui quanto segue, sostituendolo `<sitewise_gateway_ip>` con l'IP del gateway SiteWise Edge:

```
openssl s_client -connect <sitewise_gateway_ip>:443 </dev/null 2>/dev/null | grep -Po  
'CN = \K.*' | head -1
```

Esegui il comando seguente per esportare il nome host da utilizzare in seguito, sostituendolo `<your_edge_gateway_hostname>` con il nome host del gateway Edge: SiteWise

```
export GATEWAY_HOSTNAME='<your_edge_gateway_hostname>'
```

Passaggio 3: Ottieni credenziali temporanee per il tuo gateway Edge SiteWise

Ora che hai il certificato firmato e il nome host del tuo gateway SiteWise Edge, devi ottenere credenziali temporanee per poter funzionare APIs sul gateway. È possibile ottenere queste credenziali tramite AWS OpsHub AWS IoT SiteWise o direttamente dal gateway SiteWise Edge utilizzando. APIs

 Important

Le credenziali scadono ogni 4 ore, quindi è consigliabile ottenerle immediatamente prima di utilizzare il APIs gateway Edge. SiteWise Non memorizzate nella cache le credenziali per più di 4 ore.

Ottieni credenziali temporanee utilizzando for AWS OpsHubAWS IoT SiteWise

 Note

È necessario installare l'[AWS IoT SiteWise applicazione AWS OpsHub for](#).

Per utilizzare l' AWS IoT SiteWise applicazione AWS OpsHub per ottenere le credenziali temporanee, procedi come segue:

1. Accedere all'applicazione.
2. Seleziona Impostazioni.
3. Per Autenticazione, scegli Copia credenziali.
4. Espandi l'opzione più adatta al tuo ambiente e scegli Copia.
5. Salva le credenziali per utilizzarle in seguito.

Ottieni credenziali temporanee utilizzando l'API del gateway SiteWise Edge

Per utilizzare l'API SiteWise Edge gateway per ottenere le credenziali temporanee è possibile utilizzare uno script Python o curl, per prima cosa è necessario disporre di un nome utente e una password per SiteWise il gateway Edge. I gateway SiteWise Edge utilizzano l'autenticazione e l'autorizzazione SigV4. [Per ulteriori informazioni sull'aggiunta di utenti, consulta LDAP o pool di utenti Linux](#). Queste credenziali verranno utilizzate nei passaggi seguenti per ottenere le credenziali locali sul gateway SiteWise Edge necessarie per utilizzare il. AWS IoT SiteWise APIs

Python

 Note

[È necessario installare urllib3 e Python3.](#)

Per ottenere le credenziali usando Python

1. Crea un file chiamato `get_credentials.py` e poi copia il seguente codice al suo interno.

```
'''
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 the AWS IoT SiteWise Edge 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. Esegui `get_credentials.py` dal terminale sostituendo `<gateway_username>` e `<gateway_password>` con le credenziali che hai creato.

```
python3 get_credentials.py -e https://$GATEWAY_HOSTNAME -c $PATH_TO_CERTIFICATE  
-u '<gateway_username>' -p '<gateway_password>' -m 'linux'
```

curl

Note

È necessario che [curl](#) sia installato.

Per ottenere le credenziali usa curl

1. Esegui il seguente comando dal terminale sostituendo `<gateway_username>` e `<gateway_password>` con le credenziali che hai creato.

```
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"  
}
```

La risposta dovrebbe essere simile alla seguente:

```
{  
  "username": "sweuser",  
  "accessKeyId": "<accessKeyId>",  
  "secretAccessKey": "<secretAccessKey>",  
  "sessionToken": "<sessionToken>",  
  "sessionExpiryTime": "2022-11-17T04:51:40.927095Z",  
  "authMechanism": "linux",  
  "role": "edge-user"  
}
```

2. Eseguire il seguente comando dal terminale.

```
export AWS_ACCESS_KEY_ID=<accessKeyId>
export AWS_SECRET_ACCESS_KEY=<secretAccessKey>
export AWS_SESSION_TOKEN=<sessionToken>
export AWS_REGION=edge
```

Passaggio 4: Ottieni un elenco dei modelli di asset sul gateway SiteWise Edge

Ora che disponete di un certificato firmato, del nome host del gateway SiteWise Edge e delle credenziali temporanee per il gateway SiteWise Edge, potete utilizzare l'`ListAssetModelsAPI` per ottenere un elenco dei modelli di asset sul gateway SiteWise Edge.

Python

Note

[È necessario installare Python3 e Boto3. BotoCore](#)

Per ottenere l'elenco dei modelli di asset usando Python

1. Crea un file chiamato `list_asset_model.py` e poi copia il seguente codice al suo interno.

```
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. Esegui `list_asset_model.py` dal terminale.

```
python3 list_asset_model.py
```

curl

 Note

È necessario installare [curl](#).

Per ottenere l'elenco dei modelli di asset che utilizzano curl

Esegui il seguente comando dal terminale.

```
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"
```

La risposta dovrebbe essere simile alla seguente:

```
{
  "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  
}
```

Ospita un gateway SiteWise Edge su Siemens Industrial Edge

Ospita il tuo gateway su Siemens Industrial Edge utilizzando l'applicazione AWS IoT SiteWise Edge. Allo stesso modo AWS IoT Greengrass V2, puoi ottimizzare i processi di produzione o migliorare i flussi di lavoro operativi utilizzando SiteWise Edge on. Siemens Industrial Edge

Puoi importare dati dal tuo dispositivo Siemens Industrial Edge al tuo AWS account eseguendo un gateway SiteWise Edge sul dispositivo. Per fare ciò, richiedi l'accesso all'applicazione AWS IoT SiteWise Edge al team di supporto SiteWise Edge. Quindi, crea una risorsa gateway SiteWise Edge con un obiettivo di implementazione del dispositivo Siemens Industrial Edge: nuovo. Successivamente, scaricate il file di configurazione e caricatelo nell'applicazione tramite il Siemens Industrial Edge Management portale. Per ulteriori informazioni sull'esecuzione delle applicazioni su Siemens Industrial Edge, incluso come configurare le Siemens risorse richieste, vedi [Cos'è Industrial Edge?](#) nella documentazione Siemens.

Note

Siemens non è un fornitore o fornitore di Edge. SiteWise II Siemens Industrial Edge Marketplace è un marketplace indipendente.

Argomenti

- [Sicurezza](#)
- [Siemens Secure Storage e l'applicazione AWS IoT SiteWise Edge](#)
- [Destinazioni per dispositivi Siemens Industrial Edge](#)
- [Esegui la migrazione dall'applicazione di anteprima](#)
- [Risoluzione dei problemi](#)
- [AWS IoT SiteWise Registro delle modifiche delle applicazioni Edge](#)
- [Requisiti per l'applicazione AWS IoT SiteWise Edge](#)

- [Crea un gateway per Siemens Industrial Edge](#)
- [Crea un file Siemens Databus user per l'applicazione](#)
- [Accedi all'applicazione AWS IoT SiteWise Edge](#)
- [Installa l'applicazione su un dispositivo Siemens](#)
- [Aggiornare la configurazione dell'applicazione AWS IoT SiteWise Edge](#)

Sicurezza

Come parte del [modello di responsabilità condivisa](#) tra AWS i nostri clienti e i nostri partner, di seguito viene descritto chi è responsabile dei diversi aspetti della sicurezza:

Responsabilità del cliente

- Controllo del partner.
- Configurazione dell'accesso alla rete fornito al partner.
- Protezione fisica del dispositivo su cui è installato SiteWise Edge.

AWS responsabilità

- Isolare il partner dalle risorse AWS cloud del cliente.

Responsabilità del partner

- Utilizzo di impostazioni predefinite sicure.
- Mantenimento della soluzione sicura nel tempo tramite patch e altri aggiornamenti appropriati.
- Mantenere riservati i dati dei clienti.
- Verifica di altre applicazioni disponibili nel marketplace dei partner.

Siemens Secure Storage e l'applicazione AWS IoT SiteWise Edge

Per proteggere le credenziali e i segreti necessari per eseguire l'applicazione AWS IoT SiteWise Edge, Siemens Industrial Edge fornisce meccanismi per archiviare in modo sicuro le credenziali sul dispositivo. L'applicazione AWS IoT SiteWise Edge non verrà eseguita su un dispositivo se non supporta l'archiviazione sicura di queste credenziali. Gli errori di esecuzione causati dalla mancanza del supporto Secure Storage vengono registrati nei file di registro.

Per installare ed eseguire l'applicazione AWS IoT SiteWise Edge sono necessarie le seguenti versioni minime del sistema operativo. Aggiorna i tuoi dispositivi alle versioni più recenti per installare l'applicazione.

- Per i dispositivi virtuali: IEVD versione 1.19 o successiva
- Per dispositivi fisici: IED-OS versione 2.2 o successiva

L'applicazione AWS IoT SiteWise Edge attiva non Siemens Industrial Edge verrà eseguita finché non avrai aggiornato il dispositivo.

Destinazioni per dispositivi Siemens Industrial Edge

Quando si utilizza l'applicazione AWS IoT SiteWise Edge su Siemens Industrial Edge, le destinazioni aiutano a preparare i dati prima di inviarli AWS IoT SiteWise per ulteriori analisi e distribuzioni. Puoi configurare le impostazioni di destinazione dei dati per l'inserimento di dati nel buffer utilizzando Amazon S3 o utilizzare l'ingestione di dati in tempo reale. Entrambi consentono di abbonarsi agli argomenti MQTT utilizzando filtri di percorso sulla destinazione di distribuzione del dispositivo.

Siemens Industrial Edge

Il target Siemens Industrial Edge di implementazione sul gateway supporta due metodi principali di gestione dei dati:

- AWS IoT SiteWise impostazioni in tempo reale: i dati vengono inviati direttamente al AWS IoT SiteWise momento della raccolta
- AWS IoT SiteWise memorizzato nel buffer utilizzando le impostazioni di Amazon S3: i dati vengono raccolti e archiviati temporaneamente in batch prima di essere inviati ad Amazon S3

Per ulteriori informazioni sulla configurazione di queste opzioni, consulta e. [Aggiungi una destinazione AWS IoT SiteWise con buffer utilizzando Amazon S3](#) [Aggiungi una AWS IoT SiteWise destinazione Edge in tempo reale](#)

Prefissi per i filtri di percorso

I filtri di percorso per i gateway che utilizzano obiettivi di Siemens Industrial Edge distribuzione combinano l'argomento e il nome del flusso di dati per creare un identificatore univoco per i dati. L'argomento combinato con il nome del flusso di dati viene chiamato prefisso nei gateway. Siemens Industrial Edge Ciò si differenzia dai gateway ospitati autonomamente, in cui i filtri di percorso si basano esclusivamente su argomenti MQTT.

Example Struttura dei filtri di percorso per i flussi di dati Siemens

Un tipico filtro di percorso per un flusso di dati Siemens include sia il percorso dell'argomento che il nome del flusso di dati:

```
ie/d/device1/application1/datastream1
```

Dove:

- `ie/d/` è il prefisso richiesto per i flussi di dati Siemens
- `device1/application1` rappresenta il percorso gerarchico
- `datastream1` è il nome specifico del flusso di dati

Note

Quando lavori con flussi di Siemens Industrial Edge dati, assicurati di includere sia gli argomenti dei metadati (`ie/m/`) che dei dati (`ie/d/`) nei filtri dei percorsi per ricevere informazioni complete sui flussi di dati.

Destinazioni e filtri di percorso

Visualizza i seguenti argomenti per ulteriori informazioni sulle destinazioni e sui filtri di percorso nei gateway abilitati per MQTT:

- [Comprendi le destinazioni Edge AWS IoT SiteWise](#)
- [Aggiungi una AWS IoT SiteWise destinazione Edge in tempo reale](#)
- [Aggiungi una destinazione AWS IoT SiteWise con buffer utilizzando Amazon S3](#)
- [Comprendi i filtri di percorso per AWS IoT SiteWise le destinazioni Edge](#)
- [Aggiungi filtri di percorso alle destinazioni AWS IoT SiteWise Edge](#)
- [Gestisci le destinazioni AWS IoT SiteWise Edge](#)

Esegui la migrazione dall'applicazione di anteprima

Se hai eseguito SiteWise Edge Siemens Industrial Edge durante la fase di anteprima, dovrai eseguire l'aggiornamento dalla versione di anteprima, la versione 1.0.1, alla versione più recente. Effettua le seguenti operazioni per migrare:

1. Crea nuovi gateway SiteWise Edge. Per ulteriori informazioni, consulta [Crea un gateway per Siemens Industrial Edge](#).

2. Creare un nuovo Siemens Databus user per ogni nuovo gateway. Per ulteriori informazioni, consulta [Crea un file Siemens Databus user per l'applicazione](#).
3. Disinstalla l'applicazione gateway AWS IoT SiteWise Edge versione 1.0.1 sul tuo IED.

Note

Preparatevi alle interruzioni del flusso di dati riconfigurando le AWS IoT SiteWise risorse precedentemente utilizzate dalla versione di anteprima dell' AWS IoT SiteWise applicazione Edge. Sebbene la cronologia dei dati venga preservata, esiste la possibilità di perdita di dati durante la reinstallazione del nuovo gateway.

4. Eliminare i gateway SiteWise Edge creati durante l'anteprima in. [Console AWS IoT SiteWise](#)
5. Installa l'applicazione AWS IoT SiteWise Edge gateway su IED utilizzando il nuovo file di configurazione del gateway. Per ulteriori informazioni, consulta [Installa l'applicazione su un dispositivo Siemens](#).

Important

L'installazione del nuovo gateway sovrascrive la versione di anteprima dell'applicazione SiteWise Edge. Non è possibile tornare alla versione 1.0.1 dopo aver installato la versione 2.0.0.

Dopo aver configurato il nuovo gateway Siemens Databus user, i dati fluiscono verso le tue proprietà.

Puoi anche aggiornare l'applicazione SiteWise Edge direttamente dalla versione precedente. Tuttavia, è ancora necessaria una nuova configurazione del gateway.

Risoluzione dei problemi

Per risolvere i problemi relativi al gateway SiteWise Edge sul Siemens Industrial Edge dispositivo, consulta. [Risoluzione dei problemi relativi all'applicazione AWS IoT SiteWise Edge su Siemens Industrial Edge](#)

Puoi anche accedere a [AWS Re:post](#) per trovare le risposte alle tue domande.

AWS IoT SiteWise Registro delle modifiche delle applicazioni Edge

La tabella seguente descrive le modifiche in ogni versione dell'applicazione AWS IoT SiteWise Edge.

Versione	Modifiche	
3.0.0	Nuove funzionalità • Aggiunge opzioni di configurazione aggiuntive nella AWS IoT SiteWise console per i gateway installati su Siemens Industrial Edge: • Aggiungi e configura destinazioni Amazon S3. • Rimuovi la destinazione AWS IoT SiteWise in tempo reale. • Aggiungi filtri per ogni destinazione per controllare il routing dei dati. • Aggiunge l'opzione per una configurazione globale del prefisso per inviare tutti i dati dall' SiteWise editore IoT nel Siemens IE programma di installazione. Il prefisso viene applicato dopo il filtraggio. Correzioni di bug e miglioramenti • Aggiunge la possibilità di riprovare le connessioni al Siemens Industrial Edge Databus se il tentativo iniziale dell'applicazione AWS IoT SiteWise Edge fallisce.	
2.0.1	Correzioni di bug e miglioramenti • Risolve un problema per cui l'app entrava in stato di errore e si chiudeva se non era in grado di recuperare le credenziali all'avvio AWS . Aggiunge il supporto per riprovare fino al corretto recupero delle credenziali.	
2.0.0	• L'applicazione AWS IoT SiteWise Edge è ora disponibile a livello generale. • L'applicazione richiede Siemens IEVD versione 1.19 o Siemens IED-OS versione 2.2. • Miglioramenti delle prestazioni: utilizzo ridotto della memoria e della CPU.	

Versione	Modifiche	
	- Miglioramenti del debug: ora puoi caricare un file di configurazione opzionale per abilitare i registri di debug. - Miglioramenti della sicurezza: l'applicazione consente di archiviare in modo sicuro le credenziali sul dispositivo SecureStorage API. - Dockervalore digest: sha256:4a960f29234a190ebb5224c1fd0f3e99faafccc4cb3d93ca13fef247b6656d18	
1.0.1	Rilascio iniziale	

Requisiti per l'applicazione AWS IoT SiteWise Edge

Per eseguire AWS IoT SiteWise Edge su Siemens Industrial Edge, è necessario quanto segue:

- Un account [Siemens Digital Exchange Platform](#).
- Un account Siemens Industrial Edge Hub (iehub).
- Un'istanza Siemens Industrial Edge Management.
 - Il servizio di configurazione dell'app IE. Per ulteriori informazioni, [Installing the IE App Configuration Service manually](#) consulta la Siemens Industrial Edge Management documentazione.
- Accesso alla versione 2.0.1 o successiva dell'applicazione AWS IoT SiteWise Edge. Per ulteriori informazioni, consulta [Accedi all'applicazione AWS IoT SiteWise Edge](#).
- Un Siemens Industrial Edge dispositivo (IED) o un dispositivo Siemens Industrial Edge virtuale (IEVD).
 - Almeno 15 GB di spazio su disco per i requisiti hardware.
 - 1 GB di RAM con 1 GB aggiuntivo di memoria di swap.
 - Configurazione del dispositivo per consentire il traffico in uscita sulle porte 443 e 8883.
 - Un processore x86-64 bit.
 - Siemens Industrial Edge Management versione 1.13.10 o successiva.
 - Conformità del dispositivo ai requisiti. Siemens Secure Storage
 - Sui dispositivi virtuali, IEVD versione 1.19 o successiva.

- Sui dispositivi fisici, IED-OS versione 2.2 o successiva.
- La versione più recente della Docker Compose.
- Docker Engine versione 18.091 o successiva.
- Accesso al dominio richiesto. Per ulteriori informazioni, consulta [AWS IoT SiteWise punti finali](#).

Crea un gateway per Siemens Industrial Edge

Dopo avere gli account Siemens e le istanze IEM appropriati, è possibile creare un gateway SiteWise Edge di tipo Siemens Industrial Edge.

Note

Assicurati di soddisfare tutti i requisiti per l'esecuzione di un dispositivo. Siemens Industrial Edge Management Per ulteriori informazioni, consulta [Requisiti per l'applicazione AWS IoT SiteWise Edge](#).

Per creare il file di configurazione

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Edge gateway.
3. Selezionare Create gateway (Crea gateway).
4. Per il tipo di implementazione, scegli il dispositivo Siemens Industrial Edge.
5. Inserisci un nome per il tuo gateway SiteWise Edge o utilizza il nome generato da AWS IoT SiteWise.
6. (Facoltativo) In configurazione avanzata, effettuate le seguenti operazioni:
 - Immettete un nome per l' AWS IoT Core oggetto o utilizzate il nome generato da AWS IoT SiteWise.
7. Selezionare Create gateway (Crea gateway).
8. Nella finestra di dialogo Genera il file di configurazione del gateway SiteWise Edge, scegli Genera e scarica. AWS IoT SiteWise genera automaticamente un file di configurazione che utilizzerai per configurare l'applicazione AWS IoT SiteWise Edge.

 Important

Il file di configurazione del gateway viene utilizzato per eseguire il backup e il ripristino dell'applicazione AWS IoT SiteWise Edge. Salva il file di configurazione del gateway SiteWise Edge [AWS Secrets Manager](#) per archivarlo e gestirlo in modo sicuro. Secrets Manager archivia, gestisce e recupera in modo sicuro le informazioni sensibili.

Crea un file Siemens Databus user per l'applicazione

AWS IoT SiteWise Edge on Siemens Industrial Edge acquisisce i dati dall'Siemens Databus applicazione. Per connettere SiteWise Edge a Siemens Databus, è necessario un dispositivo Siemens Databus user che fornisca l'accesso ai dati su cui si desidera trasferire in modo sicuro. AWS IoT SiteWise Per iniziare, crea un file Siemens Databus user e quindi fornisci le credenziali all'applicazione SiteWise Edge.

Per creare una Siemens Databus user

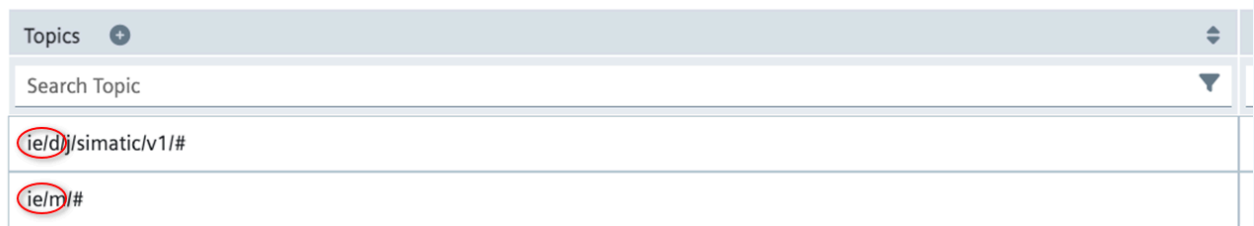
1. Nel tuo Siemens Industrial Edge Management caso, scegli Edge Management nella sezione Applicazioni della piattaforma.
2. Scegli l'icona Connessioni dati.
3. Seleziona Databus. Viene visualizzato un elenco dei dispositivi collegati.
4. Seleziona il dispositivo da connettere all'applicazione AWS IoT SiteWise Edge.
5. Scegli Avvia. Viene visualizzato il pulsante Databus Configurator per il dispositivo selezionato.
6. Crea un utente per il tuo dispositivo Edge in Utenti. Per ulteriori informazioni sulla creazione di un utente, consulta [Utenti](#) nella documentazione di Siemens Industrial Edge Management.
7. Seleziona gli argomenti a cui Siemens Databus dovrebbe avere accesso. Questi argomenti limitano l'accesso a cui AWS IoT SiteWise Edge può accedere.

 Important

Tutti gli argomenti a cui Siemens Databus user ha accesso vengono pubblicati su AWS IoT SiteWise.

Note

Siemens Databus users deve accedere sia agli argomenti relativi ai dati che ai metadati. Gli argomenti che iniziano con `ie/d` sono argomenti relativi ai dati. E gli argomenti che iniziano con `ie/m` sono argomenti relativi ai metadati. Condividi gli argomenti in coppia in modo che SiteWise Edge abbia accesso sia ai dati che ai metadati per ogni rispettivo argomento.



8. Imposta le autorizzazioni appropriate per la tua Siemens Databus configurazione.

Dopo aver creato la Siemens Databus configurazione, puoi installare l'applicazione AWS IoT SiteWise Edge sul tuo Siemens Industrial Edge Management. Per ulteriori informazioni, consulta [Installa l'applicazione su un dispositivo Siemens](#).

Facoltativamente, puoi anche configurare destinazioni e filtri di percorso per il tuo Siemens Industrial Edge gateway. Per ulteriori informazioni, consulta [Destinazioni e filtri di percorso](#).

Accedi all'applicazione AWS IoT SiteWise Edge

Per accedere all'applicazione AWS IoT SiteWise Edge su Siemens Industrial Edge, [invia un'e-mail](#) per richiedere l'accesso al team di supporto SiteWise Edge.

Includi le seguenti informazioni nell'e-mail:

- Il tuo nome e le tue informazioni di contatto
- Company name (Nome dell'azienda)
- Siemens Industrial EdgeID inquilino

Installa l'applicazione su un dispositivo Siemens

Dopo aver ottenuto l'accesso all'applicazione AWS IoT SiteWise Edge inviando un'e-mail al team di supporto SiteWise Edge per Siemens Industrial Edge, assegna l'applicazione a un'istanza di Siemens Industrial Edge Management. Quindi, puoi installare l'applicazione AWS IoT SiteWise Edge sul tuo dispositivo.

Per installare l'applicazione AWS IoT SiteWise Edge

1. Verificate che il Docker digest fornito all'interno Siemens Industrial Edge Management corrisponda alla versione più recente elencata in [AWS IoT SiteWise Registro delle modifiche delle applicazioni Edge](#)

Per ulteriori informazioni sulla localizzazione del valore del Docker digest per Siemens, consulta la sezione [Gestione di un'app](#) nel Siemens Industrial Edge dispositivo della documentazione Siemens.

Siemens Industrial Edge Management supporta una versione dell'applicazione AWS IoT SiteWise Edge alla volta. Esegui questo passaggio per assicurarti di utilizzare la versione più recente dell'applicazione prima di installare l'applicazione AWS IoT SiteWise Edge sul tuo Siemens Industrial Edge dispositivo.

2. Assegna l'applicazione AWS IoT SiteWise Edge a Siemens Industrial Edge Management. Per ulteriori informazioni, consulta [Gestione di un'app](#) nella sezione Industrial Edge Management della Siemens documentazione.
3. All'interno di Edge Management, sfoglia il catalogo di AWS IoT SiteWise Edge e sceglierlo.
4. Scegli Installa.

Note

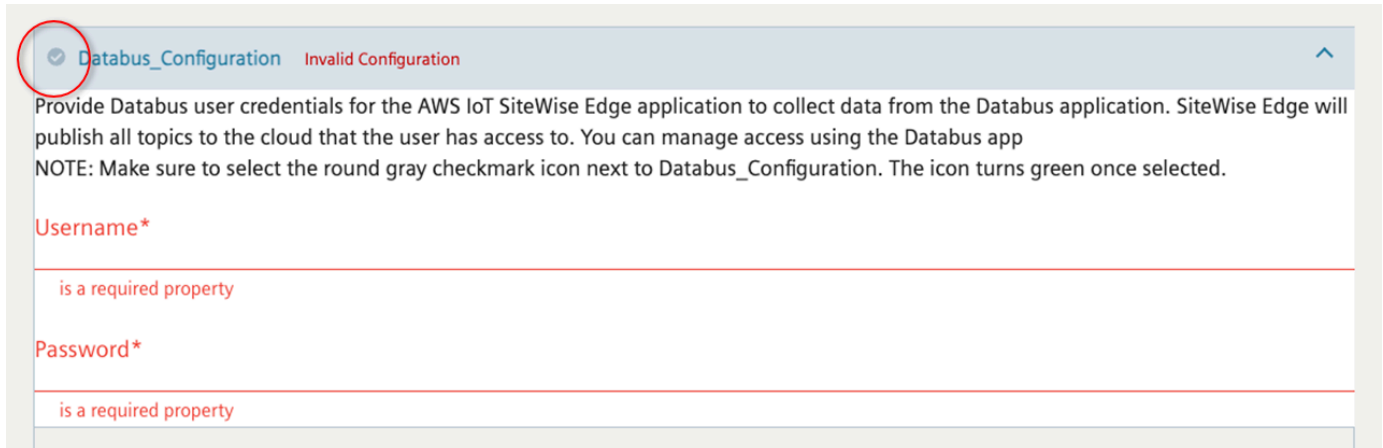
Se viene visualizzato un pulsante Contattaci, selezionalo e segui i passaggi per richiedere l'accesso all'applicazione AWS IoT SiteWise Edge su Siemens Industrial Edge. Per ulteriori informazioni, consulta [Accedi all'applicazione AWS IoT SiteWise Edge](#).

5. Seleziona Databus_Configuration nelle opzioni di configurazione dello schema.
6. Immettete il nome utente e la password per la configurazione Databus. Per ulteriori informazioni sulla creazione di un file Siemens Databus user, vedere [Crea un file Siemens Databus user per l'applicazione](#).

7. Scegli la piccola icona rotonda grigia con il segno di spunta accanto a Databus_Configuration per colorare l'icona in verde.

 Note

Le configurazioni di input si applicano solo se l'icona del segno di spunta passa da grigia a verde. In caso contrario, la configurazione di input viene ignorata.



 Databus_Configuration Invalid Configuration ^

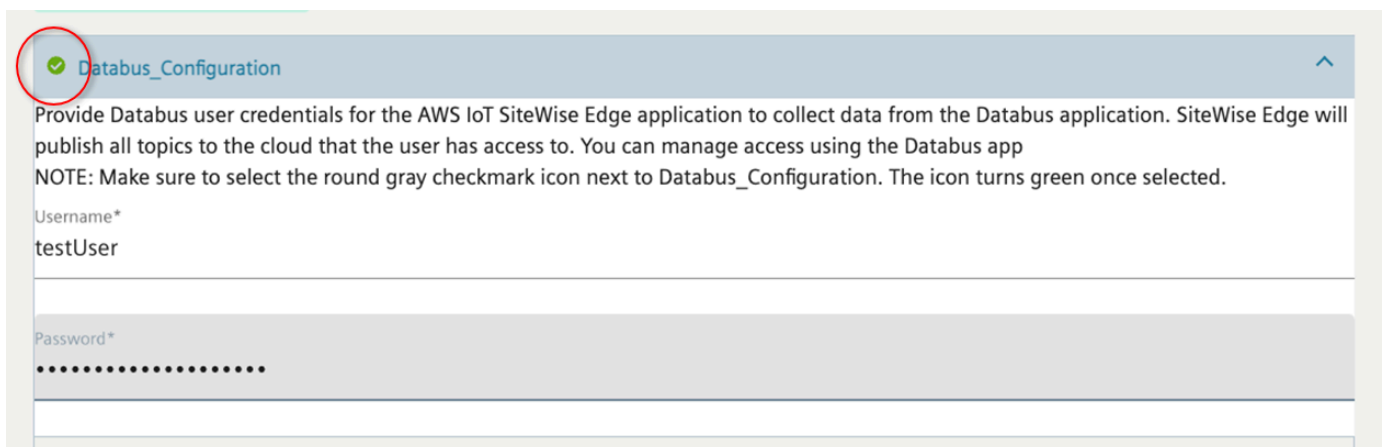
Provide Databus user credentials for the AWS IoT SiteWise Edge application to collect data from the Databus application. SiteWise Edge will publish all topics to the cloud that the user has access to. You can manage access using the Databus app
NOTE: Make sure to select the round gray checkmark icon 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. Scegli Avanti per passare ad Altre configurazioni, dove puoi caricare il file di configurazione del gateway.
9. Scegliete SiteWise_Edge_Gateway_Config come posizione per caricare il file di configurazione del gateway.

Note

Assicuratevi di scegliere `_Edge_Gateway_Config` anziché `_Edge_Support_Config_Optional`. SiteWise SiteWise

10. Selezionate il dispositivo per installare l'applicazione.

11. Selezionare Install now (Installa ora).

Facoltativamente, puoi configurare il componente publisher per esportare i dati nel AWS Cloud. Per ulteriori informazioni, consulta [configurare il componente AWS IoT SiteWise publisher](#).

Per configurare le destinazioni per il Siemens Industrial Edge gateway, consulta [Destinazioni e filtri di percorso](#).

Aggiornare la configurazione dell'applicazione AWS IoT SiteWise Edge

Ci sono alcuni aspetti da considerare quando si aggiorna la configurazione di un'applicazione AWS IoT SiteWise Edge. Siemens Industrial Edge

Note

Qualsiasi modifica alla configurazione dell'applicazione AWS IoT SiteWise Edge richiede il riavvio dell'applicazione.

Motivi per riavviare l'applicazione AWS IoT SiteWise Edge

- Una novità Siemens Databus user per l'applicazione AWS IoT SiteWise Edge.
- Una modifica al file di configurazione del gateway (il file `SiteWise_Edge_Gateway_Config`).
- Un aggiornamento della configurazione del proxy (che richiede anche un riavvio IEVD completo)
- Per abilitare i registri di debug per i problemi di debug

Riavvio dell'applicazione

1. Nel tuo Siemens Industrial Edge Management caso, scegli Edge Management nella sezione Applicazioni della piattaforma.

2. Scegli Le mie app installate.
3. Seleziona l'applicazione AWS IoT SiteWise Edge.
4. Scegli Riavvia.

Destinazioni e filtri di percorso

Le destinazioni in AWS IoT SiteWise Edge offrono un modo flessibile ed efficiente per gestire il flusso dei dati industriali dai dispositivi periferici al cloud. Questa sezione spiega come configurare le destinazioni, utilizzare i filtri di percorso per indirizzare flussi di dati specifici e scegliere il tipo di destinazione giusto per il proprio caso d'uso.

È possibile utilizzare destinazioni e filtri di percorso su gateway V3 con hosting autonomo e gateway V3 e utilizzati insieme all'applicazione Edge ospitata su. AWS IoT SiteWise Siemens Industrial Edge I filtri di destinazione e percorso non funzionano con i gateway Classic Streams e V2.

Argomenti

- [Comprendi le destinazioni Edge AWS IoT SiteWise](#)
- [Comprendi i filtri di percorso per AWS IoT SiteWise le destinazioni Edge](#)
- [Aggiungi una AWS IoT SiteWise destinazione Edge in tempo reale](#)
- [Aggiungi una destinazione AWS IoT SiteWise con buffer utilizzando Amazon S3](#)
- [Aggiungi filtri di percorso alle destinazioni AWS IoT SiteWise Edge](#)
- [Gestisci le destinazioni AWS IoT SiteWise Edge](#)

Comprendi le destinazioni Edge AWS IoT SiteWise

Utilizza le destinazioni AWS IoT SiteWise Edge per determinare dove inviare i dati di origine. Puoi scegliere la destinazione dei dati in base alle caratteristiche specifiche di cui hai bisogno, come convenienza, bassa latenza o requisiti di archiviazione. Integra i dati dei dispositivi acquisiti dai AWS IoT SiteWise nostri partner o dalle applicazioni personalizzate per pubblicare e sottoscrivere filtri di percorso (argomenti) sull'edge. È quindi possibile modellare, trasferire e archiviare i dati del dispositivo nel cloud.

Note

Per sfruttare appieno tutte le funzionalità di destinazione sui gateway ospitati autonomamente, esegui l'upgrade alle versioni più recenti di IoT SiteWise Publisher e IoT

SiteWise OPC UA Collector. Il supporto per lo streaming continua sui gateway Classic Streams e V2 per mantenere la compatibilità con le configurazioni esistenti. Per ulteriori informazioni, consulta [Stream classici, gateway V2 per Edge AWS IoT SiteWise](#)

Argomenti

- [In che modo le destinazioni SiteWise Edge migliorano la gestione dei dati](#)
- [Tipi di destinazione](#)
- [Confronta le funzionalità di destinazione tra le versioni del gateway](#)
- [Limitazioni di destinazione](#)
- [Casi d'uso per le destinazioni SiteWise Edge](#)

In che modo le destinazioni SiteWise Edge migliorano la gestione dei dati

Esporta i dati dall'edge AWS IoT SiteWise in tempo reale o in batch utilizzando Amazon S3.

Le destinazioni migliorano la flessibilità e la scalabilità nel tuo ambiente. AWS IoT SiteWise Le destinazioni implementano un modello di gestione dei dati centralizzato, in cui le fonti pubblicano i dati su un sistema centrale. Le destinazioni determinano dove vengono inviati i dati utilizzando i filtri di percorso. Le destinazioni possono sottoscrivere più filtri di percorso.

I gateway abilitati per MQTT, ospitati autonomamente o in esecuzione su Siemens Industrial Edge, utilizzano MQTT per la comunicazione locale e sono dotati di una destinazione predefinita in tempo reale con filtri impostati su. # Ciò significa che, per impostazione predefinita, tutti i messaggi su tutti gli argomenti vengono pubblicati nella destinazione in tempo reale. AWS IoT SiteWise Per ulteriori informazioni, consulta [Comprendi i filtri di percorso per AWS IoT SiteWise le destinazioni Edge](#). È possibile aggiungere una destinazione in tempo reale in ogni gateway.

Tipi di destinazione

Quando configuri una destinazione per il tuo gateway, hai due opzioni principali: configurazione in tempo reale utilizzando AWS IoT SiteWise e configurazione bufferizzata con Amazon S3. Ogni tipo di destinazione ha il proprio set di impostazioni e considerazioni.

AWS IoT SiteWise impostazioni in tempo reale

Scegli questa opzione per inviare i dati direttamente allo storage di AWS IoT SiteWise livello più elevato per facilitare l'acquisizione e il monitoraggio dei dati in tempo reale. Le impostazioni in

tempo reale gestiscono il flusso di dati, in particolare quando un gateway presenta problemi di connettività con il cloud. Durante la perdita della connessione, i dati vengono temporaneamente archiviati localmente sul gateway. Una volta ristabilita la connessione, i dati memorizzati vengono inviati automaticamente al cloud.

È possibile modificare vari aspetti del processo di pubblicazione dei dati, ad esempio la quantità massima di dati da archiviare localmente, la velocità con cui i dati vengono inviati al cloud al momento della riconnessione e quando eliminare i dati una volta raggiunta la capacità di archiviazione.

Per ulteriori informazioni sui livelli AWS IoT SiteWise di storage, consulta, [Gestisci l'archiviazione dei dati in AWS IoT SiteWise](#)

AWS IoT SiteWise bufferizzato utilizzando le impostazioni di Amazon S3

Questo tipo di destinazione consente di bufferizzare i dati localmente sul gateway e di inviarli periodicamente a un bucket Amazon S3 in batch. I dati vengono archiviati nell'efficiente formato Parquet, ottimizzato per i carichi di lavoro analitici. Una volta che i dati sono in Amazon S3, puoi importarli AWS IoT SiteWise per l'archiviazione, l'elaborazione e l'analisi.

Scegli questa opzione per importare i dati in batch e archiviare i dati storici in modo conveniente. Puoi configurare la posizione preferita del bucket Amazon S3 e la frequenza con cui desideri che i dati vengano caricati su Amazon S3. Puoi anche scegliere cosa fare con i dati dopo l'ingestione. AWS IoT SiteWise Puoi scegliere di rendere i dati disponibili sia SiteWise in Amazon S3 che in Amazon S3 oppure puoi scegliere di eliminarli automaticamente da Amazon S3.

Confronta le funzionalità di destinazione tra le versioni del gateway

La funzionalità di destinazione nei gateway compatibili con MQTT semplifica la gestione del flusso di dati. Le destinazioni semplificano la gestione dei dati attraverso la configurazione centralizzata del routing dei dati verso vari endpoint. Questo approccio elimina la necessità di configurazioni complesse di singoli flussi, rendendo l'intero sistema più flessibile e facile da gestire.

In confronto, Classic Streams, V2 gateway, SiteWise Edge trasmette i dati dalle fonti di dati agli editori tramite AWS IoT Greengrass stream, configurando le destinazioni dei dati individualmente per ciascuna fonte di dati.

Con la funzionalità di AWS IoT SiteWise destinazione, la configurazione del routing dell'editore viene consolidata. La configurazione della destinazione consente di gestire destinazioni e filtri di percorso

in modo centralizzato. È possibile aggiungere facilmente una destinazione, gestire i filtri di percorso, eliminare filtri o destinazioni non necessari, a seconda delle esigenze.

Inoltre, la funzionalità di destinazione utilizza MQTT (Message Queuing Telemetry Transport), un protocollo standard di settore ampiamente utilizzato nelle applicazioni IoT industriali. Questa adozione di MQTT aiuta a facilitare l'integrazione con vari dispositivi e sistemi. AWS IoT SiteWise

Limitazioni di destinazione

Le attuali limitazioni per le destinazioni sui gateway SiteWise Edge includono:

- Il pacchetto di elaborazione dati non è supportato sui gateway abilitati per MQTT.
- Il supporto dei tipi di dati è limitato ai tipi di dati. AWS IoT SiteWise Per informazioni sull'attivazione della conversione dei tipi di dati, consulta [Conversione di tipi di dati non supportati](#).

Casi d'uso per le destinazioni SiteWise Edge

SiteWise Le destinazioni Edge vengono utilizzate in diverse applicazioni. Ecco alcuni esempi chiave:

Automazione industriale, Monitoraggio in tempo reale e manutenzione predittiva

In ambienti industriali, i sensori e i dispositivi presenti in fabbrica possono pubblicare dati su SiteWise Edge. Le destinazioni possono essere configurate per filtrare e indirizzare i dati pertinenti, consentendo il monitoraggio e l'analisi in tempo reale delle prestazioni delle macchine. È possibile abbonarsi agli argomenti MQTT pertinenti utilizzando i filtri di percorso, elaborare i dati e quindi pubblicare i dati elaborati. In questo modo, puoi indirizzare selettivamente i dati elaborati verso servizi di analisi AWS cloud o sistemi locali. I produttori possono quindi implementare strategie di manutenzione predittiva, ottimizzare i processi di produzione e ridurre i tempi di inattività.

Edifici intelligenti, Efficienza energetica e ottimizzazione dell'occupazione

I sistemi di automazione degli edifici generano flussi di dati per monitorare e controllare vari aspetti di un edificio, come i sistemi HVAC, l'illuminazione e il controllo degli accessi. Con SiteWise Edge, questi flussi di dati possono essere acquisiti, elaborati e indirizzati verso destinazioni diverse. I facility manager possono configurare le destinazioni per filtrare e inoltrare i dati pertinenti, abilitando funzionalità avanzate come le misure di efficienza energetica e l'ottimizzazione dell'occupazione, garantendo al contempo la privacy e la conformità dei dati.

Questi casi d'uso dimostrano come la funzionalità Destinations di SiteWise Edge possa essere sfruttata in vari settori per acquisire, elaborare e instradare i dati in modo efficiente. Ciò consente funzionalità avanzate come il monitoraggio in tempo reale, la manutenzione predittiva, l'efficienza energetica e la diagnostica remota, garantendo al contempo la privacy e la conformità dei dati.

Comprendi i filtri di percorso per AWS IoT SiteWise le destinazioni Edge

Argomenti

- [Requisiti del filtro dei percorsi](#)
- [Procedure consigliate per i filtri di percorso](#)
- [Filtri di percorso per server OPC UA](#)
- [Caratteri speciali nei nomi dei filtri di percorso](#)

Ogni destinazione è configurata per instradare i dati verso AWS IoT SiteWise o verso Amazon S3. I filtri di percorso consentono di selezionare dati specifici da filtrare quando si ricevono messaggi MQTT per una destinazione. I filtri di percorso rappresentano i nomi logici dei flussi di dati e fungono da sottoscrizioni agli argomenti MQTT desiderati.

In MQTT, i dati sono organizzati in argomenti, che sono stringhe gerarchiche separate da barre (). / Ad esempio, un dispositivo potrebbe pubblicare dati sulla temperatura sull'argomento. `home/livingroom/sensor1/temperature` Qui `home/livingroom/sensor1` rappresenta il percorso o il nome logico del sensore ed `temperature` è il tipo di dati da pubblicare.

È possibile utilizzare i filtri di percorso per sottoscrivere argomenti specifici o una serie di argomenti utilizzando i caratteri jolly (+e#). Il carattere + jolly corrisponde a un singolo livello nella gerarchia degli argomenti. Ad esempio, `home/+sensor1/temperature` `home/livingroom/sensor1/temperature` corrisponderebbe a `e. home/bedroom/sensor1/temperature` Il # carattere jolly, se usato alla fine di un filtro, corrisponde a più livelli.

È inoltre possibile utilizzare una varietà di caratteri generalmente non consentiti nella specifica MQTT all'interno del nome di un filtro di percorso. Questi caratteri non funzionano come caratteri jolly quando vengono utilizzati all'interno di un nome. AWS IoT SiteWise converte questi caratteri utilizzando la codifica per garantire la conformità MQTT preservando al contempo la struttura di denominazione originale. Questa funzionalità è particolarmente utile per adattarsi alle convenzioni di denominazione esistenti di altri sistemi. Per ulteriori informazioni, consulta [Caratteri speciali nei nomi dei filtri di percorso](#).

Selezionando attentamente i filtri di percorso appropriati, è possibile controllare quali dati vengono inviati a una destinazione specifica. Personalizza il flusso di dati in base ai requisiti del tuo sistema IoT utilizzando filtri di percorso.

Requisiti del filtro dei percorsi

Quando inserite i filtri di percorso utilizzando il Console AWS IoT SiteWise, tenete presente quanto segue:

- I filtri di percorso sono delimitati da una nuova riga, con ogni riga che rappresenta un filtro di percorso separato.
- I filtri di percorso individuali possono avere una dimensione compresa tra 1 e 65.535 byte.
- Un filtro di percorso non può essere vuoto.
- I valori nulli (U+0000) non sono consentiti.
- È possibile inserire fino a 100 filtri di percorso o 65.535 caratteri alla volta, a seconda del limite raggiunto per primo.
- Il limite complessivo è di 20.000 filtri di percorso per tutte le destinazioni su un gateway combinato.
- È possibile utilizzare %, #+, e \$ caratteri all'interno dei nomi dei filtri di percorso, tuttavia li converte AWS IoT SiteWise automaticamente nella codifica URI.

Procedure consigliate per i filtri di percorso

Quando crei filtri di percorso per le tue AWS IoT SiteWise destinazioni, prendi in considerazione le seguenti strategie per gestire efficacemente i tuoi dati.

- Strutturate i filtri in modo da rispecchiare la gerarchia dei dispositivi. Ad esempio, in un ambiente di produzione `factory/+/machine/#`, acquisisce i dati da tutte le macchine su diverse linee di produzione.
- Utilizza livelli specifici per tipi di dispositivi, ubicazioni o funzioni. Ad esempio, `factory/assembly-line/robot/temperature`. Oppure, nell'agricoltura intelligente `farm/+/crop/+/moisture`, per monitorare i livelli di umidità di varie colture in diversi campi.
- Sfrutta le wildcard in modo strategico: utilizzale + per le variazioni di un singolo livello e per # acquisire tutti i livelli successivi. Ad esempio `building/+/+/energy-consumption`, tiene traccia del consumo di energia in diverse zone e piani di un edificio. Ciò presuppone che il primo + catturi tutti i piani e il secondo tutte le zone.

- Bilancia specificità e flessibilità creando filtri sufficientemente specifici per acquisire dati pertinenti ma sufficientemente flessibili per adattarsi alle modifiche future. Ad esempio, `site/+/
equipment-type/+/
measurement` consente l'aggiunta di nuovi siti o tipi di apparecchiature senza modificare la struttura del filtro.

Testa a fondo i tuoi filtri per assicurarti che acquisiscano i dati previsti e si allineino con l'architettura e gli obiettivi del tuo sistema IoT.

Filtri di percorso per server OPC UA

Per i server OPC UA, i filtri di percorso devono corrispondere ai nomi dei tag OPC UA. Il livello finale del filtro di percorso deve corrispondere esattamente al nome del tag OPC UA. Ad esempio, se il tuo tag OPC UA è `Device1.Temperature`, il tuo filtro di percorso potrebbe esserlo. `factory/line1/
Device1.Temperature` È possibile utilizzare i caratteri jolly nei livelli precedenti, ad esempio `factory/+/
Device1.Temperature` per acquisire il tag su più linee di produzione. Se nei nomi dei filtri di percorso sono presenti caratteri speciali, consulta [Caratteri speciali nei nomi dei filtri di percorso](#) per ulteriori informazioni.

Caratteri speciali nei nomi dei filtri di percorso

AWS IoT SiteWise contiene caratteri comunemente usati nei protocolli industriali come OPC UA, che in genere non sono consentiti nei nomi degli argomenti MQTT standard. Questa funzionalità facilita l'integrazione più fluida dei sistemi industriali con architetture basate su MQTT.

Note

Sebbene la nostra gestione speciale dei caratteri sia utile per l'integrazione e la migrazione, si consiglia di allinearsi alle convenzioni di denominazione MQTT standard per le nuove implementazioni, quando possibile, per garantire una compatibilità più ampia.

Quando riceve dati da fonti industriali, AWS IoT SiteWise normalizza i nomi degli argomenti utilizzando la codifica URI per i caratteri speciali:

- `%` diventa `%25` (codificato prima come carattere di escape)
- `#` diventa `%23`
- `+` diventa `%2B`

- \$diventa %24 (solo all'inizio di un argomento)

Questa codifica garantisce che i dati di origine contenenti questi caratteri MQTT speciali possano essere utilizzati in modo sicuro come nomi di argomenti MQTT, preservando al contempo le convenzioni di denominazione industriale originali.

Example : caratteri speciali nei nomi dei filtri di percorso

Ecco alcuni esempi di come i nomi degli argomenti industriali potrebbero apparire nei filtri di AWS IoT SiteWise percorso:

- Factory1/Line#2/Sensor+3diventa Factory1/Line%232/Sensor%2B3
- Plant%A/Unit\$1/Tempdiventa Plant%25A/Unit%241/Temp
- Site1/#Section/+Nodediventa Site1/%23Section/%2BNode

Quando crei abbonamenti o visualizzi i nomi degli argomenti in AWS IoT SiteWise, vedrai le versioni originali non codificate. La codifica viene gestita automaticamente per garantire la conformità MQTT.

Aggiungi una AWS IoT SiteWise destinazione Edge in tempo reale

Il tipo di destinazione in tempo reale consente di trasmettere i dati IoT direttamente dai dispositivi e dai gateway AWS IoT SiteWise allo storage in tempo reale. Questa opzione è ideale per i casi d'uso che richiedono l'inserimento e l'elaborazione immediati dei dati man mano che vengono generati, senza la necessità di operazioni di batch o buffering. È possibile configurare una sola destinazione in tempo reale in ogni gateway, poiché trasmette i dati in modo continuo. AWS IoT SiteWise

Note

La duplicazione TQVs può comportare un doppio addebito.

Per aggiungere una destinazione in tempo reale

Usa la AWS IoT SiteWise console o aggiungi una destinazione in tempo reale AWS CLI al tuo gateway abilitato per SiteWise Edge MQTT.

Console

1. Apri la [AWS IoT SiteWise console](#).

2. Nel riquadro di navigazione, scegli Edge gateway.
3. Seleziona il gateway a cui desideri aggiungere una destinazione.
4. Nella sezione Destinazioni, scegli Aggiungi destinazione.
5. Nella pagina Aggiungi destinazione, inserisci i dettagli della destinazione:
 - a. Un nome per la destinazione nel campo Nome destinazione.
 - b. Seleziona il AWS IoT SiteWise tempo reale per il tipo di destinazione.
6. Configura l'ordine di pubblicazione del gateway impostando l'ordine di pubblicazione su Pubblica prima i dati più vecchi o Pubblica prima i dati più recenti. Per impostazione predefinita, il gateway pubblica per primi i dati più vecchi.
7. Utilizza Tempo di attesa massimo per il batch per impostare il tempo massimo di attesa per l'editore prima di inviare un batch di dati a AWS IoT SiteWise. Questa impostazione si applica a ogni alias. I dati vengono archiviati localmente fino a quando:
 - Il tempo impostato è trascorso, oppure
 - Sono state time-quality-value ricevute 10 voci (TQV) per l'alias

La condizione soddisfatta per prima attiva l'invio del batch al cloud.

8. Per comprimere i dati caricati, seleziona la casella di controllo Attiva la compressione durante il caricamento dei dati. Consentire al gateway di comprimere i dati prima di caricarli sul cloud riduce l'utilizzo della larghezza di banda.
9. Per filtrare i dati scaduti degli editori, seleziona la casella di controllo Escludi dati scaduti. Questa selezione invia solo i dati attivi e correnti a. AWS IoT SiteWise
10. Nel campo Periodo limite, inserisci la frequenza con cui i dati devono essere considerati scaduti all'interno del set di dati. È possibile determinare se i dati vengono conteggiati in termini di minuti o giorni. Il periodo limite minimo è di cinque minuti. Il periodo limite massimo è di sette giorni.
11. Configura facoltativamente le impostazioni di archiviazione locale:
 - a. Imposta la frequenza del periodo di conservazione: la quantità di tempo in cui il gateway archivia localmente i dati più vecchi del periodo limite. Il periodo di conservazione minimo è di un minuto.

Il periodo di conservazione massimo è di 30 giorni ed è maggiore o uguale al periodo di rotazione.

- b. Imposta il periodo di rotazione: l'intervallo di tempo da specificare quando si salvano dati precedenti al periodo limite per un singolo file. Il gateway trasferisce un batch di dati nella seguente directory locale alla fine di ogni periodo di rotazione: `/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports`

La conservazione deve essere superiore a un minuto e pari al periodo di conservazione.

- c. Fornisci il valore della capacità di archiviazione (GB) per impostare la dimensione massima dei dati archiviati localmente in GB. Se i dati superano la dimensione massima di archiviazione locale determinata, il gateway inizia a eliminare prima i dati più vecchi. Il gateway continua a eliminare finché la dimensione dei dati archiviati localmente non è uguale o inferiore alla quota.

La capacità di archiviazione deve essere maggiore o uguale a un GB.

12. Aggiungi filtri di percorso alla tua destinazione. Per ulteriori informazioni, consultare [Aggiungi filtri di percorso alle destinazioni AWS IoT SiteWise Edge](#).

Per ulteriori informazioni, consulta [Tipi di destinazione](#).

AWS CLI

Example : crea una nuova destinazione AWS IoT SiteWise in tempo reale

Utilizza l'[UpdateGatewayCapabilityConfiguration](#) API per configurare l'editore.

Imposta il parametro `capabilityNamespace` su `iotsitewise:publisher:3`.

```
{
  "sources": [
    {
      "type": "MQTT"
    }
  ],
  "destinations": [
    {
      "type": "SITEWISE_REALTIME",
      "name": "your-destination-name",
      "config": {
        "publishingOrder": "TIME_ORDER",
        "enableCompression": true,
        "maxBatchWaitTime": "10s"
      }
    }
  ],
}
```

```

    "filters": [
      {
        "type": "PATH",
        "config": {
          "paths": [
            "#"
          ]
        }
      }
    ]
  }
]
}

```

Per aggiornare una destinazione esistente AWS IoT SiteWise in tempo reale, utilizza innanzitutto l'API `DescribeGatewayCapabilityConfiguration` per trovare il `destinationId`.

Example : aggiorna una destinazione AWS IoT SiteWise in tempo reale

Utilizza l'[UpdateGatewayCapabilityConfiguration](#) API per configurare l'editore.

Imposta il parametro `capabilityNamespace` su `iotsitewise:publisher:3`.

```

{
  "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": [
            "#"
          ]
        }
      }
    ]
  }
]
}

```

Le seguenti opzioni di configurazione sono specifiche per i gateway che utilizzano lo `iotsitewise:publisher:3` spazio dei nomi.

`sources`

Definisce le fonti di dati a cui trasferire i dati dalle apparecchiature industriali. AWS IoT SiteWise Per i gateway compatibili con MQTT, utilizzare. MQTT

Tipo: matrice di oggetti

Campo obbligatorio: sì

`destinations`

Definisce dove inviare i dati. Le destinazioni sono in tempo reale o memorizzate nel buffer utilizzando Amazon S3. È richiesto almeno un oggetto di destinazione, ma puoi aggiungere un array vuoto. È possibile avere una destinazione in tempo reale per ogni gateway. Per ulteriori informazioni, consulta [Comprendi le destinazioni Edge AWS IoT SiteWise](#).

Tipo: matrice di oggetti

Campo obbligatorio: sì

id

L'identificatore univoco della destinazione. Puoi fornire un ID di destinazione esistente o lasciarlo vuoto. Se non si specifica un ID, per impostazione predefinita viene generato un UUID.

Tipo: string

Campo obbligatorio: no

type

Tipo di destinazione. Le opzioni includono: SITEWISE_REALTIME e. SITEWISE_BUFFERED

- SITEWISE_REALTIME— Invia i dati direttamente allo AWS IoT SiteWise storage in tempo reale.
- SITEWISE_BUFFERED— Invia dati ad Amazon S3 in batch in formato Parquet, quindi importali nello storage. AWS IoT SiteWise

Tipo: stringa

Campo obbligatorio: sì

name

Un nome univoco per la destinazione.

Tipo: stringa

Campo obbligatorio: sì

config

Configurazione specifica per il tipo di destinazione in formato JSON. La configurazione varia tra destinazioni in tempo reale e destinazioni con buffer.

Tipo: oggetto

Campo obbligatorio: sì

Ordine di pubblicazione

Determina l'ordine in cui i dati vengono pubblicati. I dati vengono pubblicati in base al relativo timestamp. Le opzioni includono e. TIME_ORDER RECENT_DATA

- TIME_ORDER(impostazione predefinita): pubblica prima i dati più vecchi.

- **RECENT_DATA**— Pubblica prima i dati più recenti.

Tipo: string

Campo obbligatorio: no

Abilita la compressione

Se impostato su `true`, abilita la compressione dei dati prima dell'invio a AWS IoT SiteWise. Consentire al gateway di comprimere i dati prima di caricarli sul cloud riduce l'utilizzo della larghezza di banda. Il valore predefinito è `true`.

Tipo: Booleano

Campo obbligatorio: no

DropPolicy

Definisce come gestire i dati più vecchi.

Tipo: oggetto

Campo obbligatorio: no

- **cutoffAge**

L'età massima dei dati da pubblicare specificata in giorni, ore e minuti. Ad esempio `7d` o `1d7h16m`. I dati più vecchi di quelli specificati non vengono inviati a AWS IoT SiteWise.

I dati precedenti al periodo limite non vengono pubblicati nel cloud. L'età limite deve essere compresa tra cinque minuti e sette giorni.

È possibile utilizzare `mh`, e `d` quando si specifica un'età limite. Nota che `m` rappresenta i minuti, `h` rappresenta le ore e `d` rappresenta i giorni.

Tipo: stringa

Campo obbligatorio: sì

- **exportPolicy**

Definisce come gestire i dati che superano l'età limite.

Tipo: oggetto

Campo obbligatorio: no

- `retentionPeriod`

Il gateway SiteWise Edge elimina tutti i dati sull'edge che sono precedenti al periodo limite dall'archiviazione locale dopo che sono stati archiviati per il periodo di conservazione specificato. Il periodo di conservazione deve essere compreso tra un minuto e 30 giorni e deve essere superiore o uguale al periodo di rotazione.

È possibile utilizzare `mh`, e `d` quando si specifica un periodo di conservazione. Nota che `m` rappresenta i minuti, `h` rappresenta le ore e `d` rappresenta i giorni.

Tipo: string

Campo obbligatorio: no

- `rotationPeriod`

L'intervallo di tempo durante il quale raggruppare e salvare i dati precedenti al periodo limite in un singolo file. Il gateway SiteWise Edge trasferisce un batch di dati nella seguente directory locale alla fine di ogni periodo di rotazione: `./greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports` Il periodo di rotazione deve essere maggiore di un minuto, e uguale o inferiore al periodo di conservazione.

È possibile utilizzare `mh`, e `d` quando si specifica un periodo di rotazione. Nota che `m` rappresenta i minuti, `h` rappresenta le ore e `d` rappresenta i giorni.

Tipo: string

Campo obbligatorio: no

- `exportSizeLimitGB`

La dimensione massima consentita per i dati archiviati localmente, in GB. Se questa quota viene violata, il gateway SiteWise Edge inizia a eliminare i primi dati fino a quando la dimensione dei dati archiviati localmente non è uguale o inferiore alla quota. Il valore di questo parametro deve essere maggiore o uguale a 1.

Tipo: integer

Campo obbligatorio: no

maxBatchWaitTime

Imposta un tempo massimo di attesa per l'editore prima di inviare un batch di dati a AWS IoT SiteWise. Questa impostazione si applica a ogni alias. I dati vengono archiviati localmente fino a quando:

- Il tempo impostato è trascorso, oppure
- Sono state time-quality-value ricevute 10 voci (TQV) per l'alias

Usare mh, e d per specificare un orario limite. Nota che m rappresenta i minuti, h rappresenta le ore e d rappresenta i giorni.

Tipo: string

Campo obbligatorio: no

filters

Filtri da applicare ai dati. È richiesto almeno un filtro.

Tipo: stringa

Campo obbligatorio: sì

type

Tipo di filtro. Utilizza PATH.

Tipo: stringa

Campo obbligatorio: sì

config

Configurazione specifica per il tipo di filtro in formato JSON. È richiesto almeno un oggetto, ma l'array può essere vuoto.

Tipo: oggetto

Campo obbligatorio: sì

- paths

Una serie di filtri di percorso. Per ulteriori informazioni, consulta [Comprendi i filtri di percorso per AWS IoT SiteWise le destinazioni Edge](#). Il percorso predefinito è #.

Tipo: matrice di stringhe

Campo obbligatorio: sì

Aggiungi una destinazione AWS IoT SiteWise con buffer utilizzando Amazon S3

Il tipo di destinazione memorizzata nel buffer consente di risparmiare sui costi di importazione AWS IoT SiteWise se non sono necessari i dati in tempo reale. Consente di archiviare temporaneamente i dati IoT in un bucket Amazon S3 prima di importarli. AWS IoT SiteWise In alternativa, puoi semplicemente caricare i tuoi dati su S3 per l'archiviazione, indipendentemente dal fatto che tu abbia intenzione di importarli in. AWS IoT SiteWise Ciò è utile per raggruppare e memorizzare i dati dai dispositivi e dai gateway prima di inserirli. AWS IoT SiteWise Con questa opzione, i dati vengono caricati nel bucket S3 specificato in formato Parquet a una frequenza configurata. È quindi possibile importare questi dati in un AWS IoT SiteWise archivio per ulteriori analisi ed elaborazioni.

Per aggiungere una destinazione bufferizzata utilizzando Amazon S3

Usa la AWS IoT SiteWise console o AWS CLI aggiungi una destinazione che bufferizza i dati utilizzando Amazon S3 al SiteWise tuo gateway Edge.

Console

Usa AWS Management Console per aggiungere una AWS IoT SiteWise destinazione bufferizzata utilizzando Amazon S3.

1. Apri la [AWS IoT SiteWise console](#).
2. Nel pannello di navigazione, scegli Edge gateway.
3. Seleziona il gateway a cui desideri aggiungere una destinazione.
4. Nella sezione Destinazioni, scegli Aggiungi destinazione.
5. Nella pagina Aggiungi destinazione, inserisci i dettagli della destinazione:
 - a. Un nome per la destinazione nel campo Nome destinazione.
 - b. Seleziona AWS IoT SiteWise bufferizzato utilizzando Amazon S3 per il tipo di destinazione. AWS IoT SiteWise memorizzato nel buffer con Amazon S3 invia i dati ad Amazon Simple Storage Service in batch, in formato Parquet, quindi li importa nello storage. AWS IoT SiteWise

6. Inserisci l'URL di Amazon S3 per la posizione in cui desideri archiviare i dati del gateway. Puoi cercare il percorso scegliendo Browse S3. Una volta aggiunto un bucket, puoi anche visualizzarlo scegliendo Visualizza.
7. Specificate la frequenza con cui il gateway deve caricare i dati su Amazon S3 inserendo un intervallo di tempo e selezionando un incremento di tempo per la frequenza di caricamento dei dati. Il valore della frequenza deve essere maggiore di 0 e minore o uguale a 30 giorni.
8. Nelle impostazioni di archiviazione dei dati, stabilisci cosa fare con i dati del gateway dopo averli importati in AWS IoT SiteWise. Esistono due decisioni da prendere in merito all'archiviazione dei dati:
 - Se desideri copiare i dati importati nell' AWS IoT SiteWise archivio, seleziona la casella di controllo Copia i dati nell'archivio. Questa opzione duplica i dati importati dal AWS IoT SiteWise bucket Amazon S3 configurato nello storage.
 - Se scegli di importare i dati dal tuo bucket Amazon S3 nello AWS IoT SiteWise storage, puoi anche specificare se i dati importati devono essere eliminati una volta completata l'importazione. Seleziona la casella di controllo Elimina dati da Amazon S3 per eliminare la data importata dal bucket Amazon S3 configurato dopo averla importata nello storage. AWS IoT SiteWise
9. Aggiungi filtri di percorso alla tua destinazione. Per ulteriori informazioni, consultare [Aggiungi filtri di percorso alle destinazioni AWS IoT SiteWise Edge](#).

AWS CLI

Example : creazione di una nuova AWS IoT SiteWise destinazione con buffer utilizzando Amazon S3

Utilizza l'[UpdateGatewayCapabilityConfiguration](#) API per configurare l'editore.

Imposta il parametro `capabilityNamespace` su `iotsitewise:publisher:3`.

```
{
  "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 : aggiorna una AWS IoT SiteWise destinazione con buffer utilizzando Amazon S3

Per aggiornare una destinazione esistente AWS IoT SiteWise in tempo reale, utilizza innanzitutto l'DescribeGatewayCapabilityConfigurationAPI per trovare la destinationId

Lo spazio dei nomi dell'editore: 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": [
            "#"
          ]
        }
      }
    ]
  }
]
}

```

Le seguenti opzioni di configurazione sono specifiche per i gateway abilitati per MQTT che utilizzano lo spazio dei nomi. `iotsitewise:publisher:3`

`sources`

Definisce le fonti di dati a cui trasferire i dati dalle apparecchiature industriali. AWS IoT SiteWise Per i gateway compatibili con MQTT, utilizzare. MQTT

Tipo: matrice di oggetti

Campo obbligatorio: sì

destinations

Definisce dove inviare i dati. Le destinazioni sono in tempo reale o memorizzate nel buffer utilizzando Amazon S3. È richiesto almeno un oggetto di destinazione, ma puoi aggiungere un array vuoto. È possibile avere una destinazione in tempo reale per ogni gateway. Per ulteriori informazioni, consulta [Comprendi le destinazioni Edge AWS IoT SiteWise](#).

Tipo: matrice di oggetti

Campo obbligatorio: sì

id

L'identificatore univoco della destinazione. Puoi fornire un ID di destinazione esistente o lasciarlo vuoto per generare automaticamente un nuovo ID per la destinazione.

Tipo: string

Campo obbligatorio: no

type

Tipo di destinazione. Le opzioni includono: SITEWISE_REALTIME e SITEWISE_BUFFERED. Scegli SITEWISE_BUFFERED.

- SITEWISE_REALTIME(impostazione predefinita): invia i dati direttamente allo AWS IoT SiteWise storage in tempo reale. Per ulteriori informazioni, consulta [Aggiungi una AWS IoT SiteWise destinazione Edge in tempo reale](#).
- SITEWISE_BUFFERED— Invia dati ad Amazon S3 in batch in formato Parquet, quindi importali nello storage. AWS IoT SiteWise

Tipo: stringa

Campo obbligatorio: sì

name

Un nome univoco per la destinazione.

Tipo: stringa

Campo obbligatorio: sì

config

Configurazione specifica per il tipo di destinazione in formato JSON. La configurazione varia tra destinazioni in tempo reale e destinazioni con buffer.

Tipo: oggetto

Campo obbligatorio: sì

`targetBucketArn`

Il bucket in ARN cui pubblicare. Scegli lo stesso Regione AWS per entrambi AWS IoT SiteWise e per Amazon S3. Se viene scelto un prefisso, deve avere tra 1 e 255 caratteri.

Note

AWS IoT SiteWise, incluso il gateway, avrà accesso all'intero bucket S3 specificato. Ti consigliamo di utilizzare un bucket dedicato per l'inserimento di dati nel buffer.

Tipo: stringa

Campo obbligatorio: sì

`publishPolicy`

Dettagli della politica di pubblicazione.

Tipo: oggetto

Campo obbligatorio: sì

`publishFrequency`

La frequenza con cui il gateway SiteWise Edge pubblica nel bucket Amazon S3. La frequenza di caricamento dei dati su Amazon S3 deve essere superiore a 0 minuti e inferiore o uguale a 30 giorni. È possibile utilizzare `mh`, e `d` quando si specifica una frequenza di pubblicazione. Nota che `m` rappresenta i minuti, `h` rappresenta le ore e `d` rappresenta i giorni. Il valore predefinito è 15 minuti.

Tipo: stringa

Campo obbligatorio: sì

localSizeLimitGB

La dimensione massima dei file scritti sul disco locale in GB. Se questa soglia viene superata, l'editore pubblica tutti i dati memorizzati nel buffer nella relativa destinazione.

Tipo: integer

Campo obbligatorio: sì

siteWiseImportPolicy

Dettagli della politica di importazione per l'importazione dei dati in. AWS IoT SiteWise

Tipo: oggetto

Campo obbligatorio: sì

enableSiteWiseStorageImport

Imposta questa opzione `true` su per importare i dati da un bucket Amazon S3 allo storage. AWS IoT SiteWise Inizialmente crea una copia dei dati in. AWS IoT SiteWise Quindi, se imposti su `enableDeleteAfterImport true`, i dati in S3 vengono eliminati dopo la copia su. AWS IoT SiteWise Si applicano le implicazioni relative ai prezzi. Il valore predefinito è `true`.

Tipo: Booleano

Campo obbligatorio: sì

enableDeleteAfterImport

Imposta questa opzione `true` su per eliminare il file nel bucket Amazon S3 dopo l'ingestione nello storage. AWS IoT SiteWise Il valore predefinito è `true`.

Tipo: Booleano

Campo obbligatorio: sì

bulkImportJobRoleArn

L'ARN del ruolo IAM che AWS IoT SiteWise presuppone la lettura dei dati bufferizzati da Amazon S3 durante l'ingestione dei dati. Questo ruolo viene utilizzato quando un dispositivo periferico chiama per avviare il processo di importazione in blocco AWS IoT SiteWise APIs .

 Note

Se `enableSiteWiseStorageImport` impostato su `true`, questo parametro è obbligatorio.

Tipo: string

Campo obbligatorio: no

Aggiungi filtri di percorso per la tua destinazione. Per ulteriori informazioni, consulta [Aggiungi filtri di percorso alle destinazioni AWS IoT SiteWise Edge](#).

Aggiungi filtri di percorso alle destinazioni AWS IoT SiteWise Edge

Aggiungi filtri di percorso a una destinazione. I filtri di percorso utilizzano la sintassi degli argomenti MQTT, dove # è un carattere jolly che corrisponde a un numero qualsiasi di livelli e + un carattere jolly che corrisponde a un singolo livello. È possibile aggiungere più destinazioni a un gateway, ognuna con il proprio set di filtri di percorso abbonati alla telemetria dell'apparecchiatura.

Siemens Industrial Edge gateway utilizzano un prefisso per la compatibilità. Per ulteriori informazioni, consulta [Prefissi per i filtri di percorso](#).

Console

Per aggiungere filtri di percorso

1. Apri la [AWS IoT SiteWise console](#).
2. Nel riquadro di navigazione, scegli Edge gateway.
3. Seleziona il gateway a cui desideri aggiungere i filtri di percorso.
4. Nella sezione Filtri di percorso sotto Aggiungi destinazione, scegli Aggiungi filtro di percorso.
5. Inserisci il filtro del percorso a cui desideri sottoscrivere questa destinazione. Puoi usare caratteri jolly (#e+) per iscriverti a più percorsi.
6. Scegli Aggiungi filtro di percorso per aggiungere il filtro del percorso all'elenco.
7. Ripeti i passaggi per aggiungere filtri di percorso aggiuntivi, se necessario.
8. Dopo aver aggiunto tutti i filtri di percorso richiesti, scegli Crea.

AWS CLI for self-hosted gateways

Example : configurazione del filtro di percorso

```
{
  "destinations": [
    {
      ...
    }
  ],
  "filters": [
    {
      "type": "PATH",
      "config": {
        "paths": [
          "home/+/sensor1/temperature",
          "home/livingroom/sensor1/temperature",
          "home/livingroom/sensor1/temperature",
          "building/#"
        ]
      }
    }
  ]
}
```

AWS CLI for Siemens IEGateways

Example : Configurazione del prefisso per i filtri di percorso

Acquisisci tutti i dati utilizzando i prefissi data (ie/d) e metadata (ie/m) per ogni filtro di percorso.

```
{
  "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

Copia i filtri di percorso tra le destinazioni scaricando l'elenco dei filtri di percorso. Per ulteriori informazioni, consulta [Scarica tutti i filtri di percorso in una destinazione \(console\)](#).

Carica i filtri di percorso in blocco

Per caricare i filtri di percorso in blocco, usa un file CSV o di testo. AWS IoT SiteWise rimuove automaticamente i duplicati esatti quando carichi i file. Ad esempio, `windfarm/site1/` e `windfarm/site1/` sono duplicati esatti che vengono rilevati AWS IoT SiteWise perché la stringa è esattamente la stessa. I duplicati parziali non vengono rimossi e comportano costi aggiuntivi. Ad esempio, `windfarm/#` e `windfarm/site1` sono argomenti sovrapposti perché `windfarm/site1` sono già inclusi in `windfarm/#`.

Note

Evita i duplicati per evitare costi aggiuntivi. Il file caricato deve essere in formato.csv o.txt. Non può contenere alcuna intestazione e deve essere composto da un'unica colonna. Nella colonna, elenca i tuoi filtri di percorso, con ogni filtro su una riga separata. Nel file non devono essere incluse altre informazioni.

Requisiti di caricamento dei file

Questi sono requisiti aggiuntivi per il filtro dei percorsi.

- Puoi caricare un file.csv o .txt. Altri formati di file non sono supportati.
- I file CSV (.csv) non possono avere intestazioni e devono contenere solo una colonna.
- Puoi avere un filtro di percorso su ogni riga.
- I file caricati non possono essere vuoti.
- Quando viene utilizzato # come jolly, deve essere l'ultimo carattere nel filtro degli argomenti. Ad esempio, topic/# o come personaggio indipendente a un determinato livello di argomento. Tuttavia, tieni presente che # può essere utilizzato anche come carattere normale all'interno del nome di un livello di argomento, ad esempio factory/machine#1/topic. Per ulteriori informazioni, consulta [Caratteri speciali nei nomi dei filtri di percorso](#).
- Puoi anche usare il + personaggio. Ad esempio, factory/+/temp utilizzatelo per misurare tutte le temperature delle fabbriche anziché factory/machine2/temp factory/machine3/temp singolarmente.

Gestisci le destinazioni AWS IoT SiteWise Edge

Dopo aver aggiunto le destinazioni, è possibile eseguire diverse operazioni per gestirle, ad esempio modificare le configurazioni delle destinazioni, eliminare le destinazioni e gestire i filtri dei percorsi.

Modificare una destinazione

Seleziona il pulsante di opzione accanto alla destinazione nella tabella e scegli il pulsante Modifica per modificare una destinazione.

Console

Per modificare una destinazione utilizzando la AWS IoT SiteWise console

1. Apri la [AWS IoT SiteWise console](#).
2. Nella barra di navigazione a sinistra, scegli Edge gateway nella sezione Edge.
3. Seleziona il gateway appropriato.
4. Nella sezione Destinazioni, scegli la destinazione che desideri modificare, quindi scegli Modifica.
5. Modifica la destinazione, quindi scegli Salva.

AWS CLI

Per modificare una destinazione utilizzando AWS CLI

- È possibile modificare una destinazione modificando le informazioni di configurazione della funzionalità JSON.

```
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

Non puoi aggiornare la destinazione `type` o `capability-namespace`. Ad esempio, non puoi passare da un tipo `SITewise_REALTIME` a `SITewise_BUFFERED`. È possibile avere una destinazione in tempo reale per ogni gateway abilitato per MQTT.

Eliminare una destinazione

Se non hai più bisogno di una destinazione, puoi eliminarla dal tuo gateway SiteWise Edge.

Console

Per eliminare una destinazione utilizzando la AWS IoT SiteWise console

1. Apri la [AWS IoT SiteWise console](#).
2. Nella barra di navigazione a sinistra, scegli Edge gateway nella sezione Edge.
3. Seleziona il gateway appropriato.
4. Nella sezione Destinazioni, scegli la destinazione che desideri eliminare, quindi scegli Elimina. Viene visualizzata una schermata di conferma.
5. Per confermare la scelta di eliminare la destinazione, digita «elimina» nella casella di conferma.

AWS CLI

Per eliminare una destinazione utilizzando AWS CLI

- Eliminare la configurazione della funzionalità del gateway specificando l'ID del gateway e modificando la configurazione della funzionalità per rimuovere la destinazione che si desidera eliminare.

```
aws iotsitewise update-gateway-capability-configuration \
  --gateway-id your-gateway-id \
  --capability-namespace "iotsitewise:publisher:3" \
  --capability-configuration '{
    "sources": [
      {
        "type": "MQTT"
      }
    ]
  }
```

```
],  
  "destinations": []  
}'
```

Note

L'array delle destinazioni può essere vuoto ([]), ma l'oggetto di destinazione stesso deve essere incluso nella configurazione delle funzionalità.

Scarica tutti i filtri di percorso in una destinazione (console)

Scarica un file CSV contenente tutti i filtri di percorso nella AWS IoT SiteWise console. Puoi utilizzare un elenco scaricato di filtri di percorso per condividere facilmente gli elenchi di filtri di percorso tra le destinazioni del gateway.

Per scaricare un file CSV di tutti i filtri di percorso utilizzando la console AWS IoT SiteWise

1. Apri la [AWS IoT SiteWise console](#).
2. Nella barra di navigazione a sinistra, scegli Edge gateway nella sezione Edge.
3. Seleziona il gateway contenente i filtri di percorso.
4. Scegli Aggiungi destinazione o Modifica destinazione.
5. Vai alla sezione Filtri Path e scegli Scarica CSV.

Note

Il file CSV include tutti i filtri di percorso in una particolare destinazione, indipendentemente da quelli selezionati dall'elenco dei filtri di percorso.

Modifica un filtro di percorso

Puoi modificare i singoli filtri di percorso per definire i dati ricevuti dalla tua destinazione.

Console

Utilizzando la AWS IoT SiteWise console, è possibile modificare ogni singolo filtro di percorso all'interno di ogni rispettiva casella di testo.

Per modificare un filtro di percorso utilizzando la AWS IoT SiteWise console

1. Apri la [AWS IoT SiteWise console](#).
2. Nella barra di navigazione a sinistra, scegli Edge gateway nella sezione Edge.
3. Seleziona il gateway contenente i filtri di percorso.
4. Seleziona la destinazione appropriata.
5. Scegli Modifica.
6. Scegliete la casella di testo per la riga contenente il filtro del percorso che desiderate modificare.
7. Aggiorna il testo del filtro del percorso, assicurandoti che la casella di controllo del filtro del percorso modificato sia selezionata.
8. Scegli Save (Salva).

AWS CLI

Per modificare i filtri di percorso per una destinazione utilizzando il AWS CLI, recuperate prima la configurazione corrente, modificatela e quindi aggiornatela utilizzando il `update-gateway-capability-configuration` comando.

Per modificare un filtro di percorso utilizzando AWS CLI

1. Recupera la configurazione corrente delle funzionalità:

```
aws iotsitewise describe-gateway-capability-configuration \
  --gateway-id your-gateway-id \
  --capability-namespace "iotsitewise:publisher:3" \
  --query "capabilityConfiguration"
```

2. Modifica il JSON per modificare i filtri di percorso in base alle esigenze.
3. Aggiorna la configurazione delle funzionalità con i filtri di percorso modificati:

```
aws iotsitewise update-gateway-capability-configuration \
  --gateway-id your-gateway-id \
```

```
--capability-namespace "iotsitewise:publisher:3" \  
--capability-configuration json-containing-your-updated-path-filters
```

Eliminare un filtro di percorso

È possibile eliminare i filtri di percorso per una destinazione per controllare i dati che riceve dalle fonti MQTT e dalle pipeline di elaborazione dei dati.

Console

Per eliminare un filtro di percorso utilizzando la console AWS IoT SiteWise

1. Apri la [AWS IoT SiteWise console](#).
2. Nella barra di navigazione a sinistra, scegli Edge gateway nella sezione Edge.
3. Seleziona il gateway contenente i filtri di percorso.
4. Seleziona la destinazione appropriata.
5. Scegli Modifica.
6. Nella schermata Modifica destinazione, nella sezione Filtri di percorso, seleziona uno o più filtri di percorso da eliminare.
7. Scegliere Delete (Elimina). Viene visualizzato un messaggio di conferma dell'eliminazione. Se desideri procedere con l'eliminazione dei filtri dei percorsi, scegli Elimina nella schermata di conferma.

AWS CLI

Per eliminare una destinazione utilizzando AWS CLI

- Eliminare un filtro di percorso rimuovendolo dalla configurazione delle funzionalità.

```
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

L'array dei filtri può essere vuoto ([]), ma l'oggetto filters stesso deve essere incluso nella configurazione delle funzionalità.

Gestione dei gateway SiteWise Edge

Puoi utilizzare le operazioni della AWS IoT SiteWise console e dell'API per gestire i gateway AWS IoT SiteWise Edge. È inoltre possibile utilizzare l'applicazione [AWS OpsHubAWS IoT SiteWise per Windows per](#) gestire alcuni aspetti del gateway SiteWise Edge dal dispositivo locale.

Si consiglia vivamente di utilizzare l' AWS IoT SiteWise applicazione AWS OpsHub for per monitorare l'utilizzo del disco sul dispositivo locale. Puoi anche monitorare i CloudWatch parametri `Gateway.AvailableDiskSpace` e `Gateway.UsedPercentageDiskSpace` Amazon e creare allarmi per ricevere notifiche quando lo spazio su disco si sta esaurendo. Per ulteriori informazioni

sugli CloudWatch allarmi Amazon, consulta [Creare un CloudWatch allarme basato su una soglia statica](#).

Assicurati che il tuo dispositivo disponga di spazio sufficiente per i dati in arrivo. Quando lo spazio sul dispositivo locale sta per esaurire, il servizio elimina automaticamente una piccola quantità di dati con i timestamp più vecchi per fare spazio ai dati futuri.

Per verificare se il servizio ha eliminato i tuoi dati, procedi come segue:

1. Accedi all' AWS IoT SiteWise applicazione AWS OpsHub for.
2. Seleziona Impostazioni.
3. Per i registri, specifica un intervallo di tempo, quindi scegli Scarica.
4. Decomprimi il file di registro.
5. Se il file di registro contiene il seguente messaggio, il servizio ha eliminato i dati: sono stati eliminati *number* byte di dati per evitare che lo spazio di archiviazione del gateway SiteWise Edge si esaurisca.

Gestisci il tuo gateway SiteWise Edge con la console AWS IoT SiteWise

Puoi utilizzare la AWS IoT SiteWise console per configurare, aggiornare e monitorare tutti i gateway SiteWise Edge del tuo AWS account.

[È possibile visualizzare i gateway SiteWise Edge accedendo alla pagina Edge Gateways nella console.AWS IoT SiteWise](#) Per accedere alla pagina dei dettagli del gateway Edge per un gateway specifico, scegli il nome di un gateway Edge.

Dalla scheda Panoramica della pagina dei dettagli del gateway Edge, è possibile effettuare le seguenti operazioni:

- Nella sezione Origini dati, aggiorna la configurazione dell'origine dati e configura fonti di dati aggiuntive
- Scegli Open CloudWatch metrics per visualizzare il numero di punti dati acquisiti per origine dati nella console delle metriche CloudWatch
- Nella sezione Funzionalità Edge, aggiungi pacchetti di dati al tuo gateway SiteWise Edge facendo clic su Modifica
- Nella sezione Configurazione del gateway, visualizza lo stato di connettività dei tuoi gateway SiteWise Edge

- Nella sezione Configurazione di Publisher, visualizza lo stato di sincronizzazione del gateway SiteWise Edge e la configurazione del componente AWS IoT SiteWise Publisher

Dalla scheda Aggiornamenti della pagina dei dettagli del gateway Edge, è possibile visualizzare le versioni correnti dei componenti e dei pacchetti distribuite sul gateway Edge. Qui è anche possibile distribuire nuove versioni, quando sono disponibili.

Gestisci i gateway SiteWise Edge utilizzando per AWS OpsHubAWS IoT SiteWise

Utilizzi l' AWS IoT SiteWise applicazione AWS OpsHub for per gestire e monitorare i gateway SiteWise Edge ospitati autonomamente. Questa applicazione offre le seguenti opzioni di monitoraggio e gestione:

- In Panoramica, è possibile effettuare le seguenti operazioni:
 - Visualizza i dettagli del gateway SiteWise Edge che ti aiutano a ottenere informazioni dettagliate sui dati del dispositivo SiteWise Edge Gateway, a identificare i problemi e a migliorare le prestazioni del gateway SiteWise Edge.
 - Visualizza SiteWise i portali di monitoraggio che monitorano i dati provenienti da server e apparecchiature locali all'edge. Per ulteriori informazioni, consultate [Cosa contiene la Guida AWS IoT SiteWise Monitor all'AWS IoT SiteWise Monitor applicazione](#).
- In Health, c'è una dashboard che mostra i dati dal gateway SiteWise Edge. Gli esperti del settore, come gli ingegneri di processo, possono utilizzare la dashboard per visualizzare una panoramica del comportamento dei gateway SiteWise Edge.
- In Risorse, visualizza le risorse distribuite sul dispositivo locale e l'ultimo valore raccolto o calcolato per le proprietà delle risorse.
- In Impostazioni, puoi fare quanto segue:
 - Se il Data Processing Pack è installato, visualizza le informazioni di configurazione del gateway SiteWise Edge e sincronizza le risorse con il AWS Cloud.
 - Scarica i file di autenticazione che puoi utilizzare per accedere al gateway SiteWise Edge utilizzando altri strumenti.
 - Scarica i log che puoi utilizzare per risolvere i problemi relativi al gateway Edge. SiteWise
 - Visualizza i AWS IoT SiteWise componenti distribuiti sul gateway Edge. SiteWise

⚠ Important

È necessario utilizzare quanto segue AWS OpsHub per AWS IoT SiteWise:

- Il dispositivo locale e l'AWS IoT SiteWise applicazione AWS OpsHub for devono essere connessi alla stessa rete.
- Il pacchetto di elaborazione dati deve essere abilitato.

Per gestire i gateway SiteWise Edge utilizzando AWS OpsHub

1. Scarica e installa l'applicazione [AWS OpsHubAWS IoT SiteWise per Windows](#).
2. Aprire l'applicazione .
3. Se non disponi di credenziali locali configurate per il gateway, segui i passaggi seguenti [Accedi al tuo gateway SiteWise Edge utilizzando le credenziali del sistema operativo locale](#) per configurarle.
4. Puoi accedere al tuo gateway SiteWise Edge con le tue credenziali Linux o Lightweight Directory Access Protocol (LDAP). Per accedere al gateway SiteWise Edge, esegui una delle seguenti operazioni:

Linux

1. Per il nome host o l'indirizzo IP, inserisci il nome host o l'indirizzo IP del dispositivo locale.
2. Per l'autenticazione, scegli Linux.
3. Per Nome utente, inserisci il nome utente del tuo sistema operativo Linux.
4. Per Password, inserisci la password del tuo sistema operativo Linux.
5. Selezionare Sign in (Accedi).

LDAP

1. Per Nome host o indirizzo IP, inserisci il nome host o l'indirizzo IP del tuo dispositivo locale.
2. Per l'autenticazione, scegli LDAP.
3. Per Nome utente, inserisci il nome utente del tuo LDAP.
4. Per Password, inserisci la password del tuo LDAP.

5. Selezionare Sign in (Accedi).

Accedi al tuo gateway SiteWise Edge utilizzando le credenziali del sistema operativo locale

Oltre al Lightweight Directory Access Protocol (LDAP), puoi utilizzare le credenziali Linux o Windows per accedere al tuo gateway Edge ospitato autonomamente SiteWise .

Important

Per accedere al gateway SiteWise Edge con credenziali Linux, è necessario attivare il pacchetto di elaborazione dati per il gateway Edge. SiteWise

Accedi al tuo gateway SiteWise Edge utilizzando le credenziali del sistema operativo Linux

I passaggi seguenti presuppongono l'utilizzo di un dispositivo con Ubuntu. Se utilizzi una distribuzione Linux diversa, consulta la documentazione pertinente per il tuo dispositivo.

Per creare un pool di utenti Linux

1. Per creare un gruppo di amministratori, esegui il comando seguente.

```
sudo groupadd --system SWE_ADMIN_GROUP
```

Gli utenti del SWE_ADMIN_GROUP gruppo possono consentire l'accesso amministrativo al gateway SiteWise Edge.

2. Per creare un gruppo di utenti, esegui il comando seguente.

```
sudo groupadd --system SWE_USER_GROUP
```

Gli utenti del SWE_USER_GROUP gruppo possono consentire l'accesso in sola lettura per il gateway SiteWise Edge.

3. Per aggiungere un utente al gruppo di amministratori, esegui il comando seguente. Sostituisci *user-name* e *password* con il nome utente e la password che desideri aggiungere.

```
sudo useradd -p $(openssl passwd -1 password) user-name
```

4. Per aggiungere un utente a uno SWE_ADMIN_GROUP o all'altro SWE_USER_GROUP, sostituiscilo *user-name* con il nome utente aggiunto nel passaggio precedente.

```
sudo usermod -a -G SWE_ADMIN_GROUP user-name
```

È ora possibile utilizzare il nome utente e la password per accedere al gateway SiteWise Edge sull'AWS IoT SiteWise applicazione AWS OpsHub for.

Accedi al tuo gateway SiteWise Edge utilizzando le credenziali di Windows

I passaggi seguenti presuppongono l'utilizzo di un dispositivo con Windows.

Important

La sicurezza è una responsabilità condivisa tra te AWS e te. Crea una politica di password efficace con almeno 12 caratteri e una combinazione di lettere maiuscole, minuscole, numeri e simboli. Inoltre, imposta le regole di Windows Firewall per consentire il traffico in entrata sulla porta 443 e bloccare il traffico in entrata su tutte le altre porte.

Per creare un pool di utenti di Windows Server

1. Esegui PowerShell come amministratore.
 - a. Sul server Windows in cui desideri installare SiteWise Edge Gateway, accedi come amministratore.
 - b. Entra PowerShell nella barra di ricerca di Windows.
 - c. Nei risultati della ricerca, fai clic con il pulsante destro del mouse sull' PowerShell app Windows. Scegli Esegui come amministratore.
2. Per creare un gruppo di amministratori, esegui il comando seguente.

```
net localgroup SWE_ADMIN_GROUP /add
```

Devi essere un utente del SWE_ADMIN_GROUP gruppo per consentire l'accesso da amministratore al gateway SiteWise Edge.

3. Per creare un gruppo di utenti, esegui il comando seguente.

```
net localgroup SWE_USER_GROUP /add
```

È necessario essere un utente del SWE_USER_GROUP gruppo per consentire l'accesso in modalità di sola lettura al gateway SiteWise Edge.

4. Per aggiungere un utente, esegui il comando seguente. Sostituisci *user-name* e *password* con il nome utente e la password che desideri creare.

```
net user user-name password /add
```

5. Per aggiungere un utente al gruppo di amministratori, esegui il comando seguente. Sostituisci *user-name* con il nome utente che desideri aggiungere.

```
net localgroup SWE_ADMIN_GROUP user-name /add
```

È ora possibile utilizzare il nome utente e la password per accedere al gateway SiteWise Edge sull'AWS IoT SiteWise applicazione AWS OpsHub for.

Gestisci il certificato del gateway SiteWise Edge

È possibile utilizzare SiteWise Monitor e applicazioni di terze parti, come Grafana, sui dispositivi gateway SiteWise Edge. Queste applicazioni richiedono una connessione TLS al servizio. SiteWise I gateway Edge attualmente utilizzano un certificato autofirmato. Se si utilizza un browser per aprire le applicazioni, ad esempio un portale SiteWise Monitor, è possibile che venga visualizzato un avviso relativo alla presenza di un certificato non attendibile.

Di seguito viene illustrato come scaricare il certificato affidabile dall'AWS IoT SiteWise applicazione AWS OpsHub for.

1. Accedere all'applicazione.
2. Seleziona Impostazioni.
3. Per Autenticazione, scegli Scarica certificato.

Quanto segue presuppone che utilizzi Google Chrome o FireFox. Se utilizzi un browser diverso, consulta la documentazione pertinente del tuo browser. Per aggiungere il certificato scaricato nel passaggio precedente a un browser, esegui una delle seguenti operazioni:

- Se utilizzi Google Chrome, segui la sezione [Configurazione dei certificati](#) nella documentazione della Guida di Google Chrome Enterprise.
- Se utilizzi Firefox, segui le istruzioni [Per caricare il certificato nel browser Mozilla o Firefox](#) nella documentazione di Oracle.

Cambia la versione dei pacchetti di componenti del gateway SiteWise Edge

È possibile utilizzare la AWS IoT SiteWise console per modificare la versione dei pacchetti di componenti sui gateway SiteWise Edge.

Per modificare la versione di un pacchetto di componenti SiteWise Edge Gateway

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione a sinistra, selezionare Gateways (Gateway).
3. Seleziona il gateway SiteWise Edge per il quale desideri modificare le versioni del pacchetto.
4. In Configurazione del gateway, scegli Visualizza versioni del software.
5. Nella pagina Modifica versioni del software, per il pacchetto di cui desideri aggiornare la versione, seleziona la versione che desideri distribuire e scegli Distribuisci.
6. Seleziona Fatto.

Elenca SiteWise i gateway Edge

Console

Per elencare i gateway SiteWise Edge

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Edge gateway.
3. Visualizza l'elenco di tutti i tuoi gateway SiteWise Edge.

AWS CLI

Per elencare i gateway utilizzati AWS CLI, procedi nel seguente modo:

- Usa il comando `list-gateways` per visualizzare tutti i tuoi gateway:

```
aws iotsitewise list-gateways
```

Questo comando restituisce un elenco dei gateway con i relativi nomi e altre IDs informazioni.

È inoltre possibile specificare i parametri di impaginazione:

```
aws iotsitewise list-gateways --max-results 10 --next-token your-token
```

Per ulteriori informazioni, vedere [list-gateways](#) nel Command Reference. AWS CLI

Descrivi un gateway Edge SiteWise

Console

Per visualizzare i dettagli del gateway

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Edge gateway.
3. Scegli il nome del gateway di cui desideri visualizzare i dettagli.
4. Visualizza i dettagli del gateway nella pagina dei dettagli del gateway Edge.

AWS CLI

Per ottenere informazioni dettagliate sull'utilizzo di un gateway specifico AWS CLI, procedi nel seguente modo:

- Usa il comando `describe-gateway` con l'ID gateway:

```
aws iotsitewise describe-gateway --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Questo comando restituisce informazioni dettagliate sul gateway.

Per ulteriori informazioni, vedere [describe-gateway](#) nel Command Reference. AWS CLI

Crea un gateway Edge SiteWise

Console

Per creare un gateway SiteWise Edge

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Edge gateway.
3. Selezionare Create gateway (Crea gateway).
4. Inserisci un nome per il tuo gateway.
5. Seleziona il gruppo Greengrass come gateway.
6. Facoltativamente, aggiungi tag al tuo gateway.
7. Scegli Create (Crea).

AWS CLI

Per creare un nuovo SiteWise gateway IoT utilizzando AWS CLI, procedi nel seguente modo:

- Usa il comando `create-gateway` per creare un nuovo gateway:

```
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"
  }'
```

Questo comando restituisce l'ID e l'ARN del nuovo gateway:

```
{
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
  "gatewayArn": "arn:aws:iotsitewise:us-east-1:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
```

```
}
```

Per ulteriori informazioni, consulta [create-gateway](#) nel Command Reference. AWS CLI

Aggiorna un SiteWise gateway Edge

Console

Per aggiornare un gateway SiteWise Edge

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Edge gateway.
3. Seleziona il gateway che desideri aggiornare.
4. Scegli Modifica.
5. Aggiorna il nome del gateway o altre impostazioni secondo necessità.
6. Scegli Save (Salva).

AWS CLI

Per aggiornare un gateway esistente utilizzando AWS CLI, procedi nel seguente modo:

- Usa il comando `update-gateway` per aggiornare il nome di un gateway:

```
aws iotsitewise update-gateway \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
  --gateway-name "UpdatedGatewayName"
```

Questo comando non produce alcun output se ha esito positivo.

Per ulteriori informazioni, vedere [update-gateway](#) nel Command Reference. AWS CLI

Aggiorna la configurazione delle funzionalità del gateway

Console

Per aggiornare la configurazione delle funzionalità del gateway

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Edge gateway.
3. Scegli il nome del gateway che desideri aggiornare.
4. Nella sezione Origini dati, scegli Modifica.
5. Aggiorna la configurazione dell'origine dati secondo necessità.
6. Scegli Save (Salva).

AWS CLI

Per aggiornare la configurazione delle funzionalità di un gateway utilizzando AWS CLI, procedi nel seguente modo:

- Usa il `update-gateway-capability-configuration` comando per aggiornare la configurazione delle funzionalità:

```
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": ""
      }
    ]
  }
```



```
}  
]  
'
```

Questo comando restituisce lo spazio dei nomi e lo stato di sincronizzazione della funzionalità:

```
{  
  "capabilityNamespace": "iotsitewise:opcuacollector:1",  
  "capabilitySyncStatus": "CONFIGURING"  
}
```

Per ulteriori informazioni, vedere [update-gateway-capability-configuration](#) nel AWS CLI Command Reference.

Tagga le risorse del gateway

Console

Per etichettare una risorsa gateway

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Edge gateway.
3. Scegli il nome del gateway a cui vuoi taggare.
4. Seleziona la scheda Tags (Tag).
5. Scegliere Gestisci tag.
6. Scegli Aggiungi nuovo tag e inserisci una chiave e un valore per ogni tag.
7. Scegli Save (Salva).

AWS CLI

Per aggiungere tag a un gateway utilizzando AWS CLI, segui questi passaggi:

- Usa il comando `tag-resource` per aggiungere tag a un gateway:

```
aws iotsitewise tag-resource \
```

```
--resource-arn "arn:aws:iotsitewise:us-east-1:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE" \
--tags '{
  "Department": "Operations",
  "Project": "FactoryAutomation"
}'
```

Questo comando non produce alcun output se ha esito positivo.

Per ulteriori informazioni, vedere [tag-resource](#) nel Command Reference. AWS CLI

Elenca i tag per un gateway

Console

Per elencare i tag per un gateway

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Edge gateway.
3. Scegli il nome del gateway per il quale desideri visualizzare i tag.
4. Seleziona la scheda Tags (Tag).
5. Visualizza l'elenco dei tag associati al gateway.

AWS CLI

Per elencare i tag associati a un gateway che utilizza AWS CLI, procedi nel seguente modo:

- Usa il `list-tags-for-resource` comando per elencare i tag per un gateway:

```
aws iotsitewise list-tags-for-resource \
  --resource-arn "arn:aws:iotsitewise:us-east-1:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
```

Questo comando restituisce i tag associati al gateway:

```
{
  "tags": {
    "Environment": "Production",
```

```
    "Location": "Factory1",  
    "Department": "Operations",  
    "Project": "FactoryAutomation"  
  }  
}
```

Per ulteriori informazioni, vedere [list-tags-for-resource](#) nel AWS CLI Command Reference.

Rimuovere i tag da un gateway

Console

Per rimuovere tag da un gateway

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Edge gateway.
3. Scegli il nome del gateway da cui desideri rimuovere i tag.
4. Seleziona la scheda Tags (Tag).
5. Scegliere Gestisci tag.
6. Scegli l'icona di rimozione (X) accanto a ogni tag che desideri rimuovere.
7. Scegli Save (Salva).

AWS CLI

Per rimuovere i tag da un gateway utilizzando AWS CLI, procedi nel seguente modo:

- Usa il comando `untag-resource` per rimuovere i tag da un gateway:

```
aws iotsitewise untag-resource \  
  --resource-arn "arn:aws:iotsitewise:us-east-1:123456789012:gateway/  
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE" \  
  --tag-keys '["Project", "Department"]'
```

Questo comando non produce alcun output se ha esito positivo.

Per ulteriori informazioni, vedere [untag-resource](#) nel Command Reference. AWS CLI

Aggiorna la versione di un componente AWS IoT SiteWise

Aggiorna il componente AWS IoT SiteWise gateway sul tuo dispositivo AWS IoT Greengrass principale per assicurarti l'accesso alle funzionalità più recenti, ai miglioramenti delle prestazioni e alle patch di sicurezza.

Per aggiornare un AWS IoT SiteWise componente su AWS IoT Greengrass

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione a sinistra, scegli Edge gateway.
3. Seleziona il gateway da modificare e scegli Modifica.
4. In Edge Capabilities, in Versioni software, scegli Aggiornamenti software disponibili. Viene visualizzata la pagina Modifica versioni del software.
5. Scegli la versione del componente.

Note

Si consiglia di selezionare l'ultima versione disponibile. Keeping Gateway Components up-to-date aiuta a mantenere una funzionalità ottimale per la raccolta e l'elaborazione dei dati industriali.

6. Seleziona Deploy (Implementa). Ciò avvia una AWS IoT Greengrass V2 distribuzione per aggiornare il AWS IoT SiteWise componente sul gateway.

Eliminare un gateway SiteWise Edge

Console

Per eliminare il gateway SiteWise Edge

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Edge gateway.
3. Scegli il gateway che desideri eliminare.
4. Scegliere Delete (Elimina).
5. Per confermare che desideri eliminare il gateway, digita «elimina», quindi scegli Elimina nella finestra che appare.

AWS CLI

Per eliminare un gateway utilizzando AWS CLI, procedi nel seguente modo:

1. Elenca i gateway per identificare l'ID del gateway che desideri eliminare.

```
aws iotsitewise list-gateways
```

Questo comando restituisce un elenco dei gateway con i loro IDs nomi e altre informazioni:

```
{
  "gatewaySummaries": [
    {
      "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
      "gatewayName": "ExampleCorpGateway",
      "gatewayCapabilitySummaries": [
        {
          "capabilityNamespace": "iotsitewise:opcuacollector:1",
          "capabilitySyncStatus": "IN_SYNC"
        }
      ],
      "creationDate": 1588369971.457,
      "lastUpdateDate": 1588369971.457
    }
  ]
}
```

2. Eliminare il gateway specificandone l'ID:

```
aws iotsitewise delete-gateway --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Questo comando non produce alcun output se ha esito positivo.

Note

Quando si elimina un gateway, alcuni file del gateway rimangono nel file system del gateway.

3. Per verificare che il gateway sia stato eliminato, puoi elencare nuovamente i gateway:

```
aws iotsitewise list-gateways
```

Il gateway eliminato non dovrebbe più apparire nell'elenco.

Per ulteriori informazioni, vedere [delete-gateway](#) nel AWS CLI Command Reference.

Backup e ripristino dei gateway SiteWise Edge

Questo argomento spiega come ripristinare i gateway SiteWise Edge e il backup dei dati metrici. Se riscontri problemi con un gateway SiteWise Edge danneggiato sullo stesso computer e devi risolvere il problema, leggi la AWS IoT SiteWise documentazione [Risoluzione dei problemi SiteWise](#) relativi al gateway Edge.

Note

La guida fornita in questo argomento si riferisce SiteWise ai gateway Edge installati nella AWS IoT Greengrass V2 versione 2.1.0 o successiva.

Backup giornalieri dei dati metrici

La creazione di un backup è importante se desideri trasferire o ripristinare i dati su una nuova macchina. Il backup dei dati riduce notevolmente il rischio di perdita dei dati operativi durante un processo di trasferimento o ripristino.

Questa sezione si applica ai gateway che utilizzano il pacchetto di elaborazione dati. Per ulteriori informazioni sul pacchetto di elaborazione dati, vedere [Configura un modello di asset per l'elaborazione dei dati su SiteWise Edge](#).

Il percorso della cartella influxdb è il seguente:

Linux

```
/greengrass/v2/work/aws.iot.SiteWiseEdgeProcessor/influxdb
```

Windows

```
C:\greengrass\v2\work\aws.iot.SiteWiseEdgeProcessor\influxdb
```

Ti consigliamo di eseguire il backup dell'intera cartella con tutto il contenuto sottostante.

Ti consigliamo di eseguire periodicamente il backup dei dati metrici da 1.0 SiteWise Edge su un disco rigido esterno o sul cloud. AWS

Ripristina un gateway SiteWise Edge

Prima di tentare di ripristinare un gateway SiteWise Edge, assicurati che tutti i dispositivi perimetrali collegati al gateway siano fermi o disconnessi.

Utilizza la seguente procedura per ripristinare un gateway SiteWise Edge:

1. Utilizza lo script di installazione scaricato durante la creazione del gateway SiteWise Edge per ripristinare il gateway SiteWise Edge sul nuovo computer. Leggi la procedura di [installazione del software SiteWise Edge gateway sul tuo dispositivo locale](#) per configurare il gateway SiteWise Edge.

Se perdi o non riesci a trovare lo script di installazione, contatta [l'assistenza AWS clienti](#).
2. Una volta installato il gateway SiteWise Edge, accedi alla [AWS IoT Greengrass console](#).
3. Per ridistribuire i componenti, vai su Gestisci, quindi in AWS IoT Greengrass Dispositivi seleziona Dispositivi principali.
4. Nella tabella dei dispositivi AWS IoT Greengrass principali, seleziona il dispositivo principale corrispondente al tuo gateway SiteWise Edge.
5. Una volta nella pagina del dispositivo, apri la scheda Distribuzioni e seleziona il tuo ID di distribuzione, si aprirà la pagina Distribuzioni con l'ID selezionato.
6. Una volta che sei nella pagina Distribuzioni, in alto a destra premi il pulsante Azioni e seleziona l'opzione Revisione. per avviare una nuova distribuzione. Configura la distribuzione. Se desideri mantenere la distribuzione così com'è, passa a Review and Deploy.
7. Attendi che lo stato di distribuzione diventi. Completed

Note

Inoltre, saranno necessari alcuni minuti prima che tutti i componenti di SiteWise Edge vengano configurati e funzionati completamente.

Ripristina AWS IoT SiteWise i dati

Utilizzare la seguente procedura per ripristinare i dati su un nuovo computer.

1. Copiare la influxdb cartella sul nuovo computer.
2. Arresta il SiteWise EdgeProcessor componente eseguendo il seguente comando nel tuo terminale:

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. Individua il percorso in cui hai eseguito il backup dei dati ed esegui il seguente comando:

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. Riavvia il SiteWiseEdgeProcessor componente:

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
```

Convalida backup e ripristini eseguiti correttamente

Utilizza questa procedura per convalidare i dati di backup e i ripristini del gateway Edge. SiteWise

Note

Questa procedura richiede l'installazione di AWS OpsHub AWS IoT SiteWise. Per ulteriori informazioni, vedere [Gestione dei gateway SiteWise Edge utilizzando AWS OpsHub for AWS IoT SiteWise](#).

1. Aperto AWS OpsHub per AWS IoT SiteWise.
2. Nella pagina Impostazioni SiteWise Edge Gateway, controllate lo stato di ogni componente elencato nella tabella Componenti. Verificate che il colore di stato sia verde e che il display indichi RUNNING.
3. Convalida i dati precedenti nella dashboard del portale per verificare che i dati passati e i nuovi dati siano entrambi configurati correttamente. Si verificherà un periodo di inattività tra i dati passati e quelli nuovi. Dovresti fare attenzione a non visualizzare una durata in cui non vengono raccolti punti dati.

Se riscontri problemi con il backup o il ripristino di un gateway SiteWise Edge, consulta i seguenti argomenti sulla risoluzione dei problemi: Risoluzione dei problemi di [un gateway AWS IoT SiteWise Edge](#).

Gateway legacy ()AWS IoT Greengrass Version 1

Note

SiteWise Gli edge gateway in esecuzione AWS IoT Greengrass V1 sono disponibili solo se hai iniziato a utilizzare questa funzionalità prima del 29 luglio 2021. Per ulteriori informazioni sull'esecuzione di un AWS IoT SiteWise gateway utilizzando AWS IoT Greengrass V2,

consulta [Ospita autonomamente un gateway AWS IoT SiteWise Edge con AWS IoT Greengrass V2](#).

SiteWise Gli edge gateway ora funzionano esclusivamente su AWS IoT Greengrass V2, fornendo funzionalità avanzate e prestazioni migliorate per le applicazioni IoT industriali. Quest'ultima versione AWS IoT Greengrass V2 rappresenta un'evoluzione dell'architettura, costruita su un moderno framework basato su componenti che consente l'implementazione modulare del software. Semplifica l'installazione tramite un programma di installazione unificato, offrendo al contempo agli sviluppatori una maggiore flessibilità nell'implementazione di componenti personalizzati e nell'esecuzione di test locali. Il modello basato sui componenti consente una gestione più efficiente delle risorse e introduce un approccio di configurazione semplificato tramite ricette di componenti. Questo design facilita una migliore gestione delle dipendenze tra i componenti, supporta pratiche di distribuzione continua e fornisce funzionalità CLI avanzate per lo sviluppo locale. Inoltre, AWS IoT Greengrass V2 centralizza la gestione della configurazione AWS IoT Core e offre funzionalità di registrazione e monitoraggio migliorate, il tutto protetto da un modello di autorizzazioni di sicurezza più granulare.

Per ulteriori informazioni su come iniziare a utilizzare i gateway Edge, SiteWise AWS IoT Greengrass V2 [AWS IoT SiteWise Requisiti del gateway Edge self-hosted](#) Queste risorse forniscono step-by-step istruzioni sulla configurazione dei gateway, sulla configurazione delle fonti di dati e sulla gestione dell'infrastruttura IoT industriale.

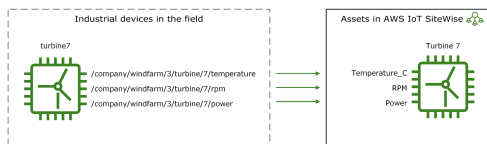
Note

Poiché AWS continua a innovare e migliorare i propri servizi IoT, si consiglia di rimanere aggiornati con le funzionalità e i miglioramenti più recenti. Controlla regolarmente la AWS IoT Greengrass documentazione AWS IoT SiteWise e le nuove funzionalità che possono ottimizzare ulteriormente le tue soluzioni IoT industriali.

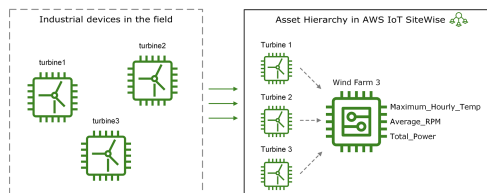
Asset industriali modello

Puoi creare rappresentazioni virtuali delle tue operazioni industriali con AWS IoT SiteWise gli asset. Una risorsa rappresenta un dispositivo, un'apparecchiatura o un processo che carica uno o più flussi di dati sul cloud. AWS Ad esempio, un dispositivo di asset può essere una turbina eolica che invia le misurazioni delle serie temporali della temperatura dell'aria, della velocità di rotazione dell'elica e dell'uscita di potenza alle proprietà degli asset in AWS IoT SiteWise.

Ogni flusso di dati corrisponde a un alias di proprietà univoco. L'alias `/company/windfarm/3/turbine/7/temperature`, ad esempio, identifica in modo univoco il flusso di dati relativi alla temperatura proveniente dalla turbina #7 nel parco eolico #3. È possibile configurare le AWS IoT SiteWise risorse per trasformare i dati di misurazione in entrata utilizzando espressioni matematiche, ad esempio per convertire i dati di temperatura da gradi Celsius a gradi Fahrenheit.



Un asset può anche rappresentare un raggruppamento logico di dispositivi, ad esempio un intero parco eolico. È possibile associare gli asset ad altri asset per creare gerarchie di asset che rappresentano operazioni industriali complesse. Le risorse possono accedere ai dati all'interno delle risorse secondarie associate. In questo modo, è possibile utilizzare AWS IoT SiteWise le espressioni per calcolare metriche aggregate, come la potenza netta in uscita di un parco eolico.



È necessario creare ogni risorsa a partire da un modello di asset. I modelli di asset sono strutture dichiarative che standardizzano il formato degli asset. I modelli di asset applicano informazioni coerenti su più asset dello stesso tipo in modo da poter elaborare i dati in risorse che rappresentano gruppi di dispositivi. Nel diagramma precedente, si utilizza lo stesso modello di asset per tutte e tre le turbine, contraddistinte da un insieme di proprietà in comune.

È inoltre possibile creare modelli di componenti. Un modello di componente è un tipo speciale di modello di asset che è possibile includere nei modelli di asset o in altri modelli di componenti.

È possibile utilizzare i modelli di componenti per definire sottoassiemi riutilizzabili comuni, come sensori, motori e così via, da condividere tra più modelli di asset.

Dopo averne definito i modelli, è possibile creare gli asset industriali. Per creare un asset, bisogna innanzitutto selezionarne un modello ACTIVE. e successivamente inserire le informazioni del caso, quali gli attributi e alias del flusso di dati. Nel diagramma precedente, si creano tre asset turbina da un modello di asset e quindi si associano gli alias del flusso di dati come `/company/windfarm/3/turbine/7/temperature` per ogni turbina.

Potete anche aggiornare ed eliminare risorse, modelli di asset e modelli di componenti esistenti. Quando si aggiorna un modello, ogni asset su di esso basato assimila le modifiche apportate. Quando aggiornate un modello di componente, ciò si applica a ogni asset basato su ogni modello di asset che fa riferimento al modello di componente.

I modelli di asset possono essere molto complessi, ad esempio quando si modella un'apparecchiatura complicata con molti sottocomponenti. Per contribuire a mantenere tali modelli di asset organizzati e gestibili, potete utilizzare modelli compositi personalizzati per raggruppare proprietà correlate o riutilizzare componenti condivisi. Per ulteriori informazioni, consulta [Modelli compositi personalizzati \(componenti\)](#).

Argomenti

- [Stati di asset e modelli](#)
- [Versioni del modello di asset](#)
- [Modelli compositi personalizzati \(componenti\)](#)
- [Configura l'oggetto AWS IoT SiteWise IDs](#)
- [Crea modelli di asset e componenti per AWS IoT SiteWise](#)
- [Create risorse per i modelli di asset in AWS IoT SiteWise](#)
- [Cerca risorse su Console AWS IoT SiteWise](#)
- [Aggiorna i valori degli attributi](#)
- [Associare e dissociare gli asset](#)
- [Aggiorna risorse e modelli](#)
- [Eliminare risorse e modelli in AWS IoT SiteWise](#)
- [Operazioni in blocco con asset e modelli](#)

Stati di asset e modelli

Quando create, aggiornate o eliminate una risorsa, un modello di asset o un modello di componente, la propagazione delle modifiche richiede tempo. AWS IoT SiteWise risolve queste operazioni in modo asincrono e aggiorna lo stato di ogni risorsa. Ogni asset, modello di asset e modello di componente ha un campo di stato che contiene lo stato della risorsa e qualsiasi messaggio di errore, se applicabile. Lo stato può avere uno dei seguenti valori:

- **ACTIVE**— La risorsa è attiva. Questo è l'unico stato in cui è possibile interrogare e interagire con risorse, modelli di asset e modelli di componenti.
- **CREATING**— La risorsa è in fase di creazione.
- **UPDATING**— La risorsa è in fase di aggiornamento.
- **DELETING**— La risorsa viene eliminata.
- **PROPAGATING**— (Solo modelli di asset e modelli di componenti) Le modifiche si stanno propagando a tutte le risorse dipendenti (dal modello di asset agli asset o dal modello di componente ai modelli di asset).
- **FAILED**— La risorsa non è stata convalidata durante un'operazione di creazione o aggiornamento, probabilmente a causa di un riferimento circolare in un'espressione. È possibile eliminare le risorse presenti nello **FAILED** stato.

Alcune delle operazioni di creazione, aggiornamento ed eliminazione eseguono una risorsa, un modello di asset o un modello di componente in uno stato diverso da quello in cui **ACTIVE** l'operazione è risolta. AWS IoT SiteWise Per interrogare o interagire con una risorsa dopo aver eseguito una di queste operazioni, è necessario attendere che lo stato cambi a **ACTIVE**. In caso contrario, le tue richieste avranno esito negativo.

Argomenti

- [Controllate lo stato di una risorsa](#)
- [Controllate lo stato di un modello di asset o componente](#)

Controllate lo stato di una risorsa

Puoi utilizzare la AWS IoT SiteWise console o l'API per verificare lo stato di una risorsa.

Argomenti

- [Verifica lo stato di una risorsa \(console\)](#)
- [Controlla lo stato di una risorsa \(AWS CLI\)](#)

Verifica lo stato di una risorsa (console)

Per verificare lo stato di un asset nella console AWS IoT SiteWise , attieniti alla procedura descritta di seguito.

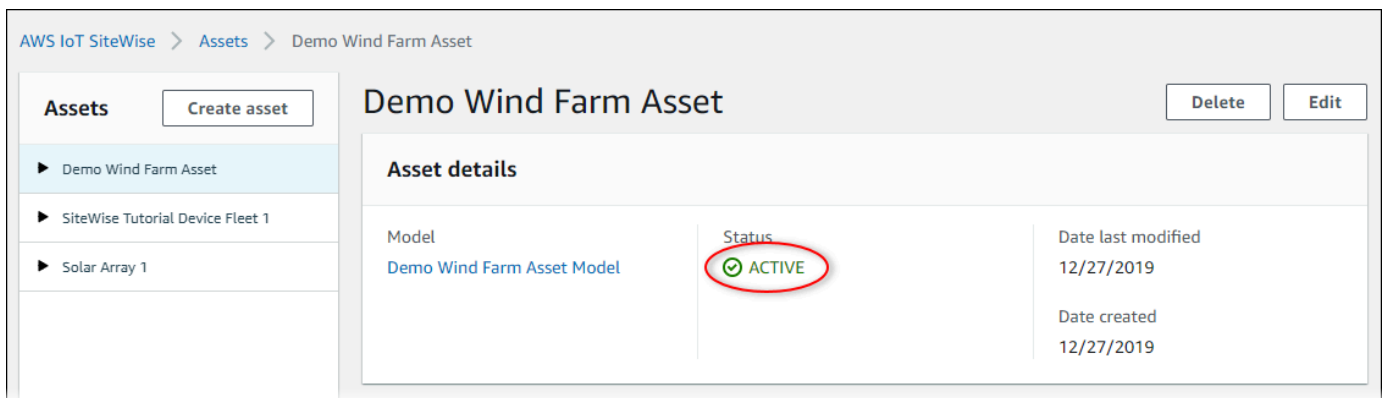
Per controllare lo stato di un asset (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Asset.
3. Scegli l'asset da controllare.

Tip

Puoi scegliere l'icona a forma di freccia per espandere una gerarchia di asset e trovare il tuo asset.

4. Trova Stato nel pannello Dettagli dell'asset.



Controlla lo stato di una risorsa (AWS CLI)

Puoi usare il AWS Command Line Interface (AWS CLI) per controllare lo stato di una risorsa.

Per verificare lo stato di una risorsa, utilizzate l'[DescribeAsset](#) operazione con il `assetId` parametro.

Per controllare lo stato di una risorsa (AWS CLI)

- Immetti il seguente comando per descrivere l'asset. Sostituisci *asset-id* con l'ID o l'ID esterno della risorsa. L'ID esterno è un ID definito dall'utente. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise .

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

L'operazione restituisce una risposta contenente i dettagli dell'asset. La risposta contiene un `assetStatus` oggetto con la seguente struttura:

```
{
  ...
  "assetStatus": {
    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  }
}
```

Lo stato dell'asset si trova in `assetStatus.state` dell'oggetto JSON.

Controllate lo stato di un modello di asset o componente

È possibile utilizzare la AWS IoT SiteWise console o l'API per verificare lo stato di un modello di asset o di un modello di componente.

Argomenti

- [Verifica lo stato di un modello di asset o di un modello di componente \(console\)](#)
- [Verifica lo stato di un modello di asset o di un modello di componente \(\)AWS CLI](#)

Verifica lo stato di un modello di asset o di un modello di componente (console)

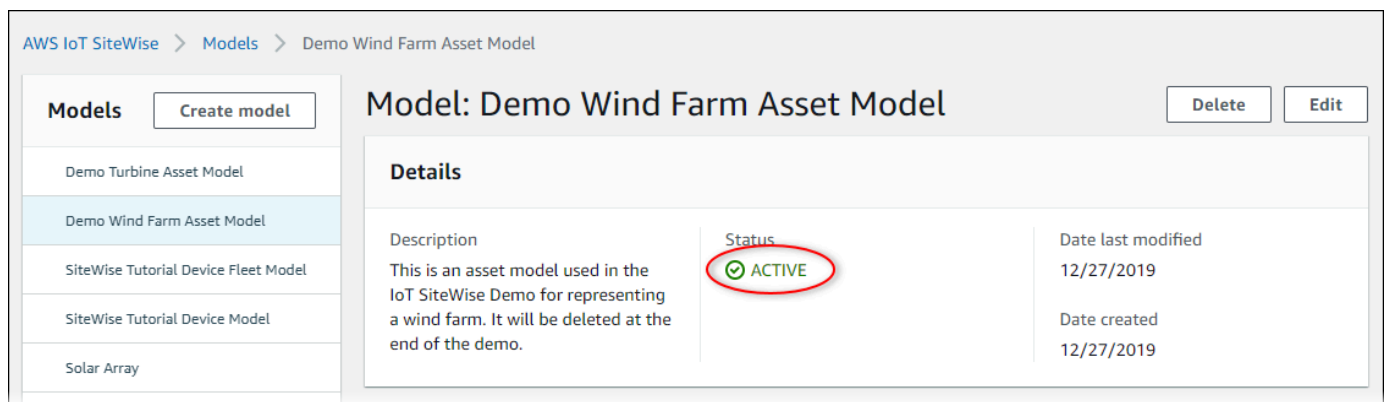
Utilizzate la seguente procedura per verificare lo stato di un modello di asset o di un modello di componente nella AWS IoT SiteWise console.

Tip

I modelli di asset e i modelli di componenti sono entrambi elencati in Modelli nel pannello di navigazione. Il pannello Dettagli del modello di asset o del modello di componente selezionato indica di che tipo si tratta.

Per verificare lo stato di un modello di asset o di un modello di componente (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione selezionare Models (Modelli).
3. Scegliete il modello da controllare.
4. Trova Stato nel pannello Dettagli.



Verifica lo stato di un modello di asset o di un modello di componente (AWS CLI)

È possibile utilizzare il AWS CLI per verificare lo stato di un modello di asset o di un modello di componente.

Per verificare lo stato di un modello di asset o di un modello di componente, utilizzate l'[DescribeAssetModel](#) operazione con il `assetModelId` parametro.

Tip

AWS CLI Definisce i modelli di componenti come un tipo di modello di asset. Pertanto, si utilizza la stessa [DescribeAssetModel](#) operazione per entrambi i tipi di modello. Il `assetModelType` campo nella risposta indica se è un `ASSET_MODEL` o un `COMPONENT_MODEL`.

Per verificare lo stato di un modello di asset o di un modello di componente (AWS CLI)

- Eseguite il comando seguente per descrivere il modello. `asset-model-id` Sostituitelo con l'ID o l'ID esterno del modello di asset o del modello di componente. L'ID esterno è un ID definito dall'utente. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise .

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

L'operazione restituisce una risposta che contiene i dettagli del modello. La risposta contiene un oggetto `assetModelStatus` che ha la seguente struttura.

```
{
  ...
  "assetModelStatus": {
    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  }
}
```

Lo stato del modello si trova `assetModelStatus.state` nell'oggetto JSON.

Versioni del modello di asset

AWS IoT SiteWise supporta l'elaborazione asincrona delle operazioni di creazione e aggiornamento su modelli di asset e modelli di componenti. Inoltre aggiorna lo stato del modello.

AWS IoT SiteWise propaga le modifiche di un modello valido nelle richieste di creazione e aggiornamento alle risorse dipendenti (dal modello di asset agli asset o dal modello di componente ai modelli di asset). Quindi colloca il modello in `ACTIVE` uno stato.

Se la definizione del modello fornita non è valida, AWS IoT SiteWise colloca il modello in uno `FAILED` stato. Le modifiche non vengono propagate alle risorse dipendenti. Le risorse dipendenti si riferiscono all'ultima definizione del modello propagata quando il modello era in uno `ACTIVE` stato.

In base alle informazioni di cui sopra, le definizioni dei modelli hanno due tipi di versioni del modello:

1. **Versione più recente:** la definizione più recente accettata come parte di una richiesta di creazione o aggiornamento.
2. **Versione attiva:** l'ultima definizione è stata elaborata correttamente e lo stato del modello è `ACTIVE`.

Per impostazione predefinita, i dettagli della versione più recente del modello vengono restituiti quando le descrizioni APIs vengono richiamate su un modello di asset o un modello di componente. Esistono scenari in cui è necessaria la versione attiva del modello di asset o del modello di componente. Di seguito sono riportati gli scenari di esempio:

- Un'operazione di aggiornamento con una definizione non valida pone il modello di asset in uno `FAILED` stato. È necessario annullare le modifiche recuperando la versione attiva del modello di asset e creando un'altra richiesta di aggiornamento che faccia riferimento a questa definizione valida.
- **AWS IoT SiteWise** Esiste un'applicazione in cui i clienti possono visualizzare gli asset e i modelli di asset corrispondenti. Quando un utente fa riferimento alla definizione del modello di asset corrispondente a un particolare asset e il modello di asset si trova in uno `FAILED` stato transitorio o transitorio `UPDATING/PROPAGATING`, l'ultima versione restituisce la definizione del modello di asset che non è ancora stata propagata agli asset in questione. In questo caso, è necessario restituire ai clienti la versione attiva del modello di asset.

Argomenti

- [Recuperate la versione attiva di un modello di asset o di un modello di componente \(console\)](#)
- [Recuperate la versione attiva di un modello di asset o di un modello di componente \(\)AWS CLI](#)

Recuperate la versione attiva di un modello di asset o di un modello di componente (console)

Seguite questa procedura per recuperare la versione attiva di un modello di asset o di un modello di componente nella AWS IoT SiteWise console.

 Tip

I modelli di asset e i modelli di componenti sono entrambi elencati in Modelli nel pannello di navigazione. Il pannello Dettagli del modello di asset o del modello di componente selezionato indica di che tipo si tratta.

Per recuperare la versione attiva di un modello di asset o di un modello di componente (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione selezionare Models (Modelli).
3. Scegliete il modello per recuperare la versione attiva.
 - a. Se il modello è in uno ACTIVE stato, stai visualizzando la sua versione attiva.
 - b. Se il modello è in uno FAILED stato transitorio UPDATING o transitorio, trova la voce Visualizza versione attiva in Stato nel pannello Dettagli. PROPAGATING

Recuperate la versione attiva di un modello di asset o di un modello di componente (AWS CLI)

Utilizzate il AWS CLI per recuperare la versione attiva di un modello di asset o di un modello di componente.

Per recuperare la versione attiva di un modello di asset o di un modello di componente, utilizzate l'[DescribeAssetModel](#) operazione con il `assetModelVersion` parametro.

 Tip

AWS CLI definisce i modelli di componenti come un tipo di modello di asset. Pertanto, si utilizza la stessa [DescribeAssetModel](#) operazione per entrambi i tipi di modello. Il `assetModelType` campo nella risposta indica se è un `ASSET_MODEL` o un `COMPONENT_MODEL`.

Per recuperare la versione attiva di un modello di asset o di un modello di componente ()AWS CLI

- Eseguite il comando seguente per descrivere il modello. *asset-model-id* Sostituitelo con l'ID o l'ID esterno del modello di asset o del modello di componente. L'ID esterno è un ID definito dall'utente. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise .

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id --asset-model-version ACTIVE
```

L'operazione restituisce una risposta con i dettagli del modello. La risposta contiene un `assetModelStatus` oggetto con la seguente struttura.

```
{
  ...
  "assetModelName": "string",
  "assetModelProperties": [ ... ],
  ...,
  "assetModelVersion": "string"
}
```

Modelli composti personalizzati (componenti)

Quando si modella un asset industriale particolarmente complesso, ad esempio un macchinario complicato composto da molte parti, può diventare una sfida mantenere i modelli di asset organizzati e gestibili.

In questi casi, puoi aggiungere modelli composti personalizzati, o componenti, se utilizzi la console, ai modelli di asset e ai modelli di componenti esistenti. Questi vi aiutano a rimanere organizzati raggruppando le proprietà correlate e riutilizzando le definizioni dei sottocomponenti.

Esistono due tipi di modelli composti personalizzati:

- I modelli composti personalizzati in linea definiscono un insieme di proprietà raggruppate che si applicano al modello di asset o al modello di componente a cui appartiene il modello composito personalizzato. Li utilizzate per raggruppare le proprietà correlate. Sono costituiti da un nome, una descrizione e un insieme di proprietà del modello di asset. Non sono riutilizzabili.

- I modelli compositi `component-model-based` personalizzati C fanno riferimento a un modello di componente che si desidera includere nel modello di asset o nel modello di componente. Li utilizzate per includere sottoassiemi standard nel modello. Sono costituiti da un nome, una descrizione e l'ID del modello di componente a cui fanno riferimento. Non hanno proprietà proprie; il modello di componente a cui si fa riferimento fornisce le proprietà associate a tutti gli asset creati.

Le seguenti sezioni illustrano come utilizzare modelli compositi personalizzati nei progetti.

Argomenti

- [Modelli compositi personalizzati in linea](#)
- [Component-model-based modelli compositi personalizzati](#)
- [Utilizzate i percorsi per fare riferimento alle proprietà personalizzate del modello composito](#)

Modelli compositi personalizzati in linea

I modelli compositi personalizzati in linea consentono di organizzare il modello di asset raggruppando le proprietà correlate.

Ad esempio, supponete di voler modellare un asset robotico. Il robot include un servomotore, un alimentatore e una batteria. Ciascuna di queste parti costituenti ha le proprie proprietà che si desidera includere nel modello. È possibile definire un modello di asset denominato `robot_model` con proprietà come le seguenti.

- `robot_model`
 - `servo_status` (numero intero)
 - `servo_position` (doppio)
 - `powersupply_status` (numero intero)
 - `powersupply_temperature` (doppio)
 - `battery_status` (numero intero)
 - `battery_charge` (doppio)

Tuttavia, in alcuni casi, potrebbero esserci molti sottoassiemi oppure i sottoassiemi stessi potrebbero avere molte proprietà. In questi casi, potrebbero esserci così tante proprietà da rendere complicato

il riferimento e la gestione in un unico elenco semplice alla radice del modello, come nell'esempio precedente.

Per gestire tali situazioni, è possibile utilizzare un modello composito personalizzato in linea per raggruppare le proprietà. Un modello composito personalizzato in linea è un modello composito personalizzato che definisce le proprie proprietà. Ad esempio, puoi modellare il tuo robot come segue.

- `robot_model`
 - `servo`
 - `status`(numero intero)
 - `position`(doppio)
 - `powersupply`
 - `status`(numero intero)
 - `temperature` (doppio)
 - `battery`
 - `status`(numero intero)
 - `charge`(doppio)

Nell'esempio precedente, `servopowersupply`, e `battery` sono i nomi dei modelli compositi personalizzati in linea definiti all'interno del modello di `robot_model` asset. Ciascuno di questi modelli compositi definisce quindi le proprie proprietà.

Note

In questo caso, ogni modello composito personalizzato definisce le proprie proprietà, in modo che tutte le proprietà facciano parte del modello di asset stesso (`robot_model` in questo caso). Queste proprietà non sono condivise con altri modelli di asset o modelli di componenti. Ad esempio, se avete creato un altro modello di asset che aveva anche un modello composito personalizzato in linea chiamato `servo`, apportare una modifica all'`servo` interno non `robot_model` influirebbe sulla `servo` definizione dell'altro modello di asset.

Se desiderate implementare tale condivisione (ad esempio, avere una sola definizione per un `servo`, condivisibile da tutti i vostri modelli di asset), dovrete invece creare un modello a

componenti relativo al servo e quindi creare modelli component-model-based compositi che vi facciano riferimento. Per i dettagli, consultate la sezione seguente.

Per informazioni su come creare modelli compositi personalizzati in linea, vedere [Crea modelli compositi personalizzati \(componenti\)](#).

Component-model-based modelli compositi personalizzati

È possibile creare un modello di componente in AWS IoT SiteWise per definire un sottoassieme standard riutilizzabile. Dopo aver creato un modello di componente, potete aggiungere riferimenti ad esso negli altri modelli di asset e modelli di componenti. A tale scopo, aggiungete un modello composito component-model-based personalizzato a qualsiasi modello in cui desiderate fare riferimento al componente. È possibile aggiungere riferimenti al componente da molti modelli o più volte all'interno dello stesso modello.

In questo modo, è possibile evitare di duplicare le stesse definizioni tra i modelli. Inoltre, semplifica la manutenzione dei modelli, poiché qualsiasi modifica apportata a un modello componente si rifletterà su tutti i modelli di asset che lo utilizzano.

Ad esempio, supponete che l'impianto industriale disponga di molti tipi di apparecchiature che utilizzano tutte lo stesso tipo di servomotore. Alcuni di essi hanno molti servomotori in un'unica apparecchiatura. Crei un modello di asset per ogni tipo di apparecchiatura, ma non vuoi duplicare la definizione di servo ogni volta. Volete modellarlo una sola volta e utilizzarlo nei vari modelli di asset. Se successivamente apporti una modifica alla definizione di servo, questa verrà aggiornata in tutti i tuoi modelli e asset.

Per modellare il robot dell'esempio precedente in questo modo, potete definire servomotori, alimentatori e batterie come modelli di componenti, in questo modo.

- `servo_component_model`
 - `status`(numero intero)
 - `position`(doppio)
- `powersupply_component_model`
 - `status`(numero intero)

- temperature (doppio)

- battery__component_model
 - status(numero intero)
 - charge(doppio)

È quindi possibile definire modelli di asset, ad esempio `robot_model` che fanno riferimento a questi componenti. Più modelli di asset possono fare riferimento allo stesso modello di componente. È inoltre possibile fare riferimento allo stesso modello di componente più volte in un unico modello di asset, ad esempio se il robot è dotato di più servomotori.

- robot_model
 - servo1(riferimento:) servo_component_model
 - servo2(riferimento:servo_component_model)
 - servo3(riferimento:servo_component_model)
 - powersupply (riferimento:powersupply_component_model)
 - battery(riferimento:battery_component_model)

Per informazioni su come creare modelli di componenti, vedere [Crea modelli di componenti](#).

Per informazioni su come fare riferimento ai modelli di componenti in altri modelli, vedere [Crea modelli compositi personalizzati \(componenti\)](#).

Utilizzate i percorsi per fare riferimento alle proprietà personalizzate del modello composito

Quando create una proprietà su un modello di asset, un modello di componente o un modello composito personalizzato, potete farvi riferimento da altre proprietà che ne utilizzano il valore, come [trasformazioni](#) e [metriche](#).

AWS IoT SiteWise offre diversi modi per fare riferimento alla proprietà. Il modo più semplice è spesso quello di utilizzare l'ID della proprietà. Tuttavia, se la proprietà a cui desiderate fare riferimento si

trova su un modello composito personalizzato, potrebbe essere più utile farvi riferimento invece tramite il percorso.

Un percorso è una sequenza ordinata di segmenti di percorso che specifica una proprietà in termini di posizione tra i modelli compositi annidati all'interno di un modello di asset e di un modello composito.

Ottieni i percorsi delle proprietà

Puoi ottenere il percorso di una proprietà dal path campo della sua [AssetModelProperty](#).

Ad esempio, supponete di avere un modello di asset `robot_model` che contiene un modello servo composito personalizzato con una proprietà `position`. Se chiami [DescribeAssetModelCompositeModel](#) on servo, la `position` proprietà elencherà un path campo simile al seguente:

```
"path": [  
  {  
    "id": "asset model ID",  
    "name": "robot_model"  
  },  
  {  
    "id": "composite model ID",  
    "name": "servo"  
  },  
  {  
    "id": "property ID",  
    "name": "position"  
  }  
]
```

Utilizzo dei percorsi delle proprietà

È possibile utilizzare un percorso di proprietà quando si definisce una proprietà che fa riferimento ad altre proprietà, ad esempio una trasformazione o una metrica.

Una proprietà utilizza una variabile per fare riferimento a un'altra proprietà. Per ulteriori informazioni sull'utilizzo delle variabili, vedere [Usa le variabili nelle espressioni delle formule](#).

Quando si definisce una variabile per fare riferimento a una proprietà, è possibile utilizzare l'ID della proprietà o il relativo percorso.

Per definire una variabile che utilizza il percorso della proprietà di riferimento, specificate il `propertyPath` campo del relativo valore.

Ad esempio, per definire un modello di asset con una metrica che fa riferimento a una proprietà utilizzando un percorso, potete passare un payload come questo a: [CreateAssetModel](#)

```
{
  ...
  "assetModelProperties": [
    {
      ...
      "type": {
        "metric": {
          ...
          "variables": [
            {
              "name": "variable name",
              "value": {
                "propertyPath": [
                  path segments
                ]
              }
            }
          ]
        }
      },
      ...
    },
    ...
  ],
  ...
}
```

Configura l'oggetto AWS IoT SiteWise IDs

AWS IoT SiteWise definisce vari tipi di oggetti persistenti, come asset, modelli di asset, proprietà e gerarchie. Tutti questi oggetti dispongono di identificatori univoci che è possibile utilizzare per recuperarli, aggiornarli ed eliminarli.

AWS IoT SiteWise offre ai clienti diverse opzioni per la creazione di ID. AWS IoT SiteWise ne genera uno automaticamente per te al momento della creazione dell'oggetto. Gli utenti possono anche fornire i propri dati IDs ai tuoi oggetti.

Argomenti

- [Lavora con oggetti UUIDs](#)
- [Usa esterno IDs](#)

Lavora con oggetti UUIDs

Ogni oggetto persistente AWS IoT SiteWise ha un [UUID](#) per identificarlo. Ad esempio, i modelli di asset hanno un ID del modello di asset, gli asset hanno un ID di asset e così via. Questo ID viene assegnato al momento della creazione dell'oggetto e rimane invariato per tutta la durata dell'oggetto.

Quando crei un nuovo oggetto, per impostazione predefinita AWS IoT SiteWise genera un ID univoco per te. Puoi anche fornire il tuo ID al momento della creazione in formato UUID.

Note

UUIDs deve essere univoco a livello globale all'interno AWS della regione in cui è stato creato e per lo stesso tipo di oggetto. Quando AWS IoT SiteWise genera automaticamente un ID per te, è sempre unico. Se scegli il tuo ID, assicurati che sia unico.

Ad esempio, se crei un nuovo modello di asset chiamando [CreateAssetModel](#), puoi fornire il tuo UUID nel `assetModelId` campo opzionale della richiesta.

Al contrario, se si omette `assetModelId` dalla richiesta, AWS IoT SiteWise genera un UUID per il nuovo modello di asset.

Usa esterno IDs

Per definire il proprio ID in un formato diverso dall'UUID, è possibile assegnare un ID esterno. Ad esempio, puoi farlo se riutilizzi un ID che stai utilizzando in un sistema che non lo è o se desideri renderlo più AWS leggibile dall'uomo. IDs Gli esterni hanno un formato più flessibile. Puoi usarli per fare riferimento ai tuoi oggetti nelle operazioni AWS IoT SiteWise API in cui altrimenti utilizzeresti l'UUID.

Come il UUIDs, ogni ID esterno deve essere unico nel suo contesto. Ad esempio, non puoi avere due modelli di asset con lo stesso ID esterno. Inoltre, come il UUIDs, un oggetto può avere un solo ID esterno nel corso della sua vita, che non può cambiare.

Differenze tra esterno IDs e UUIDs

L'esterno IDs differisce UUIDs nei seguenti modi:

- Ogni oggetto ha un UUID, ma gli oggetti esterni IDs sono opzionali.
- AWS IoT SiteWise non genera mai elementi esterni. IDs Li fornisci tu stesso.
- Se l'oggetto non ne ha già uno, puoi assegnare un ID esterno in qualsiasi momento.

Formato esterno IDs

Un ID esterno valido ha le seguenti proprietà:

- Ha una lunghezza compresa tra 2 e 128 caratteri.
- Il primo e l'ultimo carattere devono essere alfanumerici (A-Z, a-z, 0-9).
- I caratteri diversi dal primo e dall'ultimo devono essere alfanumerici oppure devono essere uno dei seguenti: `_` `-` `.` `:`

Ad esempio, un ID esterno deve essere conforme alla seguente espressione regolare:

```
[a-zA-Z0-9][a-zA-Z0-9_\-.:]*[a-zA-Z0-9]+
```

Oggetti di riferimento con oggetti esterni IDs

In molti punti in cui è possibile fare riferimento a un oggetto utilizzando il relativo UUID, è possibile utilizzare invece il relativo ID esterno, se ne ha uno. A tale scopo, aggiungete l'ID esterno alla stringa `externalId`:

Ad esempio, supponiamo di avere un modello di asset il cui UUID (asset model ID) è `a1b2c3d4-5678-90ab-cdef-11111EXAMPLE`, che ha anche l'ID esterno. `myExternalId`. Chiama [DescribeAssetModel](#) per avere maggiori dettagli al riguardo. È possibile utilizzare uno dei seguenti valori come valore di `assetModelId`:

- Con lo stesso Asset Model ID (UUID): `a1b2c3d4-5678-90ab-cdef-11111EXAMPLE`
- Con l'ID esterno: `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

Il `externalId`: prefisso, di per sé, non fa parte dell'ID esterno. È necessario fornire il prefisso solo quando si fornisce un ID esterno a un'operazione API che accetta uno UUIDs o più elementi esterni. IDs Ad esempio, fornite il prefisso quando interrogate o aggiornate un oggetto esistente.

Quando definite un ID esterno per un oggetto, ad esempio quando create un modello di asset, non includete il prefisso.

È possibile utilizzare external IDs anziché UUIDs in questo modo per molte operazioni API AWS IoT SiteWise, ma non per tutte. Ad esempio, il [GetAssetPropertyValue](#), deve usare UUIDs; non supporta l'utilizzo di ID esterni.

Per determinare se una particolare operazione API supporta questo utilizzo, consulta l'[API Reference](#).

Crea modelli di asset e componenti per AWS IoT SiteWise

AWS IoT SiteWise I modelli di asset e i modelli di componenti favoriscono la standardizzazione dei dati industriali. I modelli di asset e componenti rappresentano la struttura e le proprietà degli asset industriali e dei relativi componenti. I modelli di asset definiscono l'asset complessivo, ad esempio una turbina eolica o una linea di produzione. I modelli di componenti rappresentano i singoli componenti che compongono l'asset, come pale, generatori o sensori. Creando questi modelli, è possibile organizzare e strutturare i dati degli asset in modo da riflettere le relazioni e le gerarchie reali delle apparecchiature industriali, semplificando il monitoraggio, l'analisi e la manutenzione.

Un modello di asset o modello di componente contiene un nome, una descrizione, proprietà di asset e (facoltativamente) modelli composti personalizzati che raggruppano le proprietà o che fanno riferimento ai modelli di componenti per i sottoassiemi.

In AWS IoT SiteWise, potete creare modelli di asset e modelli di componenti per rappresentare la struttura e le proprietà degli asset industriali e dei relativi componenti.

- Utilizzate un modello di asset per creare risorse. Oltre alle funzionalità sopra elencate, un modello di asset può contenere anche definizioni gerarchiche che definiscono le relazioni tra gli asset.
- Un modello di componente rappresenta un sottoassieme all'interno di un modello di asset o di un altro modello di componente. Quando create un modello di componente, potete aggiungere riferimenti ad esso nei modelli di asset e in altri modelli di componenti. Tuttavia, non è possibile creare risorse direttamente dai modelli di componenti.

Dopo aver creato un modello di asset o un modello di componente, potete creare modelli composti personalizzati per raggruppare le proprietà o fare riferimento a modelli di componenti esistenti.

Per informazioni dettagliate su come creare modelli di asset e modelli di componenti, consultate le seguenti sezioni.

Argomenti

- [Crea modelli di asset in AWS IoT SiteWise](#)
- [Crea modelli di componenti](#)
- [Definite le proprietà dei dati](#)
- [Crea modelli composti personalizzati \(componenti\)](#)

Crea modelli di asset in AWS IoT SiteWise

AWS IoT SiteWise i modelli di asset favoriscono la standardizzazione dei dati industriali. Un modello include un nome e una descrizione, oltre a proprietà e definizioni gerarchiche di asset. Ad esempio, è possibile definire un modello di turbina eolica con proprietà di temperatura, rotazioni al minuto (RPM) e potenza. Quindi, è possibile definire un modello di parco eolico con una proprietà di potenza netta e una definizione di gerarchia di turbine eoliche.

Note

- Si consiglia di modellare l'operazione partendo dai nodi di livello inferiore. Ad esempio, crea il modello di turbina eolica prima di creare il modello di centrale eolica. Le definizioni della gerarchia delle risorse contengono riferimenti a modelli di asset esistenti. Con questo approccio, puoi definire le gerarchie di asset durante la creazione dei modelli.
- I modelli di asset non possono contenere altri modelli di asset. Se dovete definire un modello a cui potete fare riferimento come sottoassieme all'interno di un altro modello,

dovreste invece creare un modello componente-->. Per ulteriori informazioni, consulta [Crea modelli di componenti](#).

Le sezioni seguenti descrivono come utilizzare la AWS IoT SiteWise console o l'API per creare modelli di asset. Nelle sezioni seguenti vengono inoltre descritti i diversi tipi di proprietà e gerarchie degli asset che puoi utilizzare per creare modelli.

Argomenti

- [Create un modello di asset \(console\)](#)
- [Crea un modello di asset \(AWS CLI\)](#)
- [Modelli di asset di esempio](#)
- [Definite le gerarchie dei modelli di asset](#)

Create un modello di asset (console)

È possibile utilizzare la AWS IoT SiteWise console per creare un modello di asset. La AWS IoT SiteWise console offre varie funzionalità, come il completamento automatico delle formule, che possono aiutarti a definire modelli di asset validi.

Per creare un modello di asset (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione selezionare Models (Modelli).
3. Scegli Crea modello.
4. Nella pagina Crea modello, esegui le operazioni seguenti:
 - a. Immetti un nome per il modello di asset, ad esempio **Wind Turbine** o **Wind Turbine Model1**. Questo nome deve essere univoco rispetto a tutti i modelli del tuo account in questa regione.
 - b. (Facoltativo) Aggiungete un ID esterno per il modello. Si tratta di un ID definito dall'utente. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise .
 - c. (Facoltativo) Aggiungi le definizioni misurazione per il modello. Le misurazioni rappresentano flussi di dati provenienti dall'apparecchiatura. Per ulteriori informazioni, consulta [Definisci i flussi di dati provenienti dalle apparecchiature \(misurazioni\)](#).

- d. (Facoltativo) Aggiungi le definizioni di trasformazione per il modello. Le trasformazioni sono formule che mappano i dati da un modulo all'altro. Per ulteriori informazioni, consulta [Trasforma i dati \(trasformazioni\)](#).
- e. (Facoltativo) Aggiungi le definizioni parametro per il modello. Le metriche sono formule che aggregano i dati su intervalli di tempo. Le metriche possono inserire dati dalle risorse associate, in modo da poter calcolare valori che rappresentano l'operazione o un sottoinsieme dell'operazione. Per ulteriori informazioni, consulta [Aggrega i dati provenienti da proprietà e altre risorse \(metriche\)](#).
- f. (Facoltativo) Aggiungi le definizioni gerarchiche per il modello. Le gerarchie sono relazioni tra risorse. Per ulteriori informazioni, consulta [Definite le gerarchie dei modelli di asset](#).
- g. (Facoltativo) Aggiungi i tag per il modello di asset. Per ulteriori informazioni, consulta [Tagga le tue AWS IoT SiteWise risorse](#).
- h. Scegli Crea modello.

Quando si crea un modello di asset, la AWS IoT SiteWise console accede alla pagina del nuovo modello. In questa pagina puoi vedere lo stato del modello che inizialmente è CREAZIONE IN CORSO. Questa pagina si aggiorna automaticamente, quindi attendi l'aggiornamento dello stato del modello.

Note

Il processo di creazione del modello di asset può richiedere fino a qualche minuto, nei casi di maggiore complessità. Dopo che lo stato del modello di asset è ATTIVO, potete utilizzare il modello di asset per creare risorse. Per ulteriori informazioni, consulta [Stati di asset e modelli](#).

- 5. (Facoltativo) Dopo aver creato il modello di asset, potete configurare il modello di asset per l'edge. Per ulteriori informazioni su SiteWise Edge, consulta [Configura le funzionalità edge su AWS IoT SiteWise Edge](#).
 - a. Nella pagina del modello, scegli Configure for Edge.
 - b. Nella pagina di configurazione del modello, scegli la configurazione del bordo per il tuo modello. Questo controlla dove AWS IoT SiteWise è possibile calcolare e archiviare le proprietà associate a questo modello di asset. Per ulteriori informazioni sulla configurazione del modello per l'edge, consulta. [Configura una sorgente OPC UA in SiteWise Edge](#)

- c. Per la configurazione edge personalizzata, scegliete la posizione in cui desiderate AWS IoT SiteWise calcolare e archiviare ciascuna delle proprietà del modello di asset.

 Note

Le trasformazioni e le metriche associate devono essere configurate per la stessa posizione. Per ulteriori informazioni sulla configurazione del modello per l'edge, consulta [Configura una sorgente OPC UA in SiteWise Edge](#)

- d. Scegli Save (Salva). Nella pagina del modello, la configurazione di Edge dovrebbe ora essere configurata.

Crea un modello di asset (AWS CLI)

È possibile utilizzare AWS Command Line Interface (AWS CLI) per creare un modello di asset.

Utilizza l'operazione [CreateAssetModel](#) per creare un modello di asset con proprietà e gerarchie. Questa operazione prevede un payload con la seguente struttura.

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition
}
```

Per creare un modello di asset (AWS CLI)

1. Crea un file denominato `asset-model-payload.json` e copia il seguente oggetto JSON nel file.

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "",
  "assetModelDescription": "",
  "assetModelProperties": [

  ],
  "assetModelHierarchies": [
```

```
],  
  "assetModelCompositeModels": [  
  
  ]  
}
```

2. Utilizza l'editor di testo JSON preferito per modificare il file `asset-model-payload.json` come segue:
 - a. Immetti un nome (`assetModelName`) per il modello di asset, ad esempio **Wind Turbine** o **Wind Turbine Model**. In questo caso, questo nome deve essere univoco per tutti i modelli di asset e i modelli di componenti del tuo account Regione AWS.
 - b. (Facoltativo) Inserite un ID esterno (`assetModelExternalId`) per il modello di asset. Si tratta di un ID definito dall'utente. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise .
 - c. (Facoltativo) Immetti una descrizione (`assetModelDescription`) per il modello di asset o rimuovi la coppia chiave-valore `assetModelDescription`.
 - d. (Facoltativo) Definisci le proprietà dell'asset (`assetModelProperties`) per il modello. Per ulteriori informazioni, consulta [Definite le proprietà dei dati](#).
 - e. (Facoltativo) Definisci le gerarchie dell'asset (`assetModelHierarchies`) per il modello. Per ulteriori informazioni, consulta [Definite le gerarchie dei modelli di asset](#).
 - f. (Facoltativo) Definite gli allarmi per il modello. Gli allarmi monitorano altre proprietà in modo da poter identificare quando le apparecchiature o i processi richiedono attenzione. Ogni definizione di allarme è un modello composito (`assetModelCompositeModels`) che standardizza l'insieme di proprietà utilizzate dall'allarme. Per ulteriori informazioni, consultare [Monitora i dati con allarmi in AWS IoT SiteWise](#) e [Definisci gli allarmi sui modelli di asset in AWS IoT SiteWise](#).
 - g. (Facoltativo) Aggiungi i tag (`tags`) per il modello di asset. Per ulteriori informazioni, consulta [Tagga le tue AWS IoT SiteWise risorse](#).
3. Esegui il comando seguente per creare un modello di asset dalla definizione nel file JSON.

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

L'operazione restituisce una risposta contenente l'`assetModelId` univoco al quale fai riferimento quando crei un asset. La risposta contiene anche lo stato del modello

(`assetModelStatus.state`) che inizialmente è `CREATING`. Lo stato del modello di asset è `CREATING` fino a quando le modifiche non si propagano.

Note

Il processo di creazione del modello di asset può richiedere fino a qualche minuto, nei casi di maggiore complessità. Per verificare lo stato attuale del modello di asset, utilizzate l'[DescribeAssetModel](#) operazione specificando il `assetModelId`. Una volta contrassegnato dallo stato `ACTIVE`, il modello potrà essere utilizzato per creare asset. Per ulteriori informazioni, consulta [Stati di asset e modelli](#).

4. (Facoltativo) Crea modelli composti personalizzati per il tuo modello di asset. Con i modelli composti personalizzati, è possibile raggruppare le proprietà all'interno del modello o includere un sottoassieme facendo riferimento a un modello di componente. Per ulteriori informazioni, consulta [Crea modelli composti personalizzati \(componenti\)](#).

Modelli di asset di esempio

Questa sezione contiene esempi di definizioni di modelli di asset che è possibile utilizzare per creare modelli di asset con `awscli`. AWS IoT SiteWise SDKs Questi modelli di asset rappresentano una turbina eolica e un parco eolico. Gli asset delle turbine eoliche acquisiscono i dati grezzi dei sensori e calcolano valori come la potenza e la velocità media del vento. Gli asset del parco eolico calcolano valori come la potenza totale per tutte le turbine eoliche del parco eolico.

Argomenti

- [Modello di asset turbina eolica](#)
- [Modello di asset centrale eolica](#)

Modello di asset turbina eolica

Il seguente modello di asset rappresenta una turbina in una centrale eolica. La turbina eolica acquisisce i dati dei sensori per calcolare valori come la potenza e la velocità media del vento.

Note

Questo modello di esempio è simile al modello di turbina eolica mostrato nella demo. AWS IoT SiteWise Per ulteriori informazioni, consulta [Usa la 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": []
}

```

Modello di asset centrale eolica

Il seguente modello di asset rappresenta una centrale eolica che comprende più turbine eoliche. Questo modello di asset definisce una [gerarchia](#) rispetto al modello di turbina eolica. Ciò consente al parco eolico di calcolare i valori (come la potenza media) a partire dai dati di tutte le turbine eoliche del parco eolico.

Note

Questo modello di esempio è simile al modello di parco eolico illustrato nella AWS IoT SiteWise demo. Per ulteriori informazioni, consulta [Usa la AWS IoT SiteWise demo](#).

Questo modello di asset dipende da [Modello di asset turbina eolica](#). Sostituisci i valori `propertyId` e `childAssetModelId` con quelli di un modello di asset turbina eolica esistente.

```

{
  "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"
        }
    ]
}

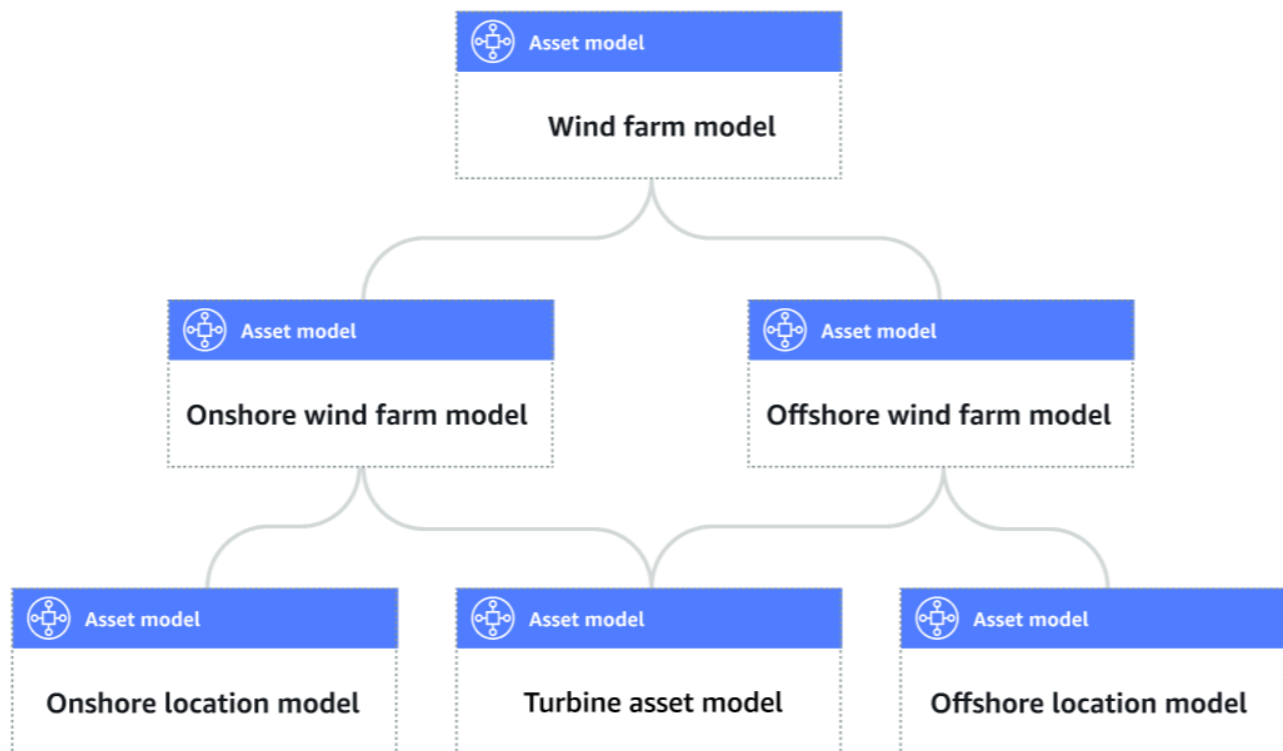
```

Definite le gerarchie dei modelli di asset

È possibile definire gerarchie di modelli di asset per creare associazioni logiche tra i modelli di asset utilizzati nelle operazioni industriali. Ad esempio, è possibile definire un parco eolico composto da parchi eolici onshore e offshore. Un parco eolico onshore contiene una turbina e una posizione onshore. Un parco eolico offshore contiene una turbina e un'ubicazione offshore.



Asset model hierarchy



Quando si associa un modello di asset figlio a un modello di asset principale tramite una gerarchia, le metriche del modello di asset principale possono inserire dati dalle metriche del modello di asset figlio. È possibile utilizzare le gerarchie e le metriche dei modelli di asset per calcolare statistiche che forniscono informazioni dettagliate sulla propria operazione o su un sottoinsieme di essa. Per ulteriori informazioni, consulta [Aggrega i dati provenienti da proprietà e altre risorse \(metriche\)](#).

Ogni gerarchia definisce una relazione tra un modello di asset principale e un modello di asset figlio. In un modello di asset principale, è possibile definire più gerarchie per lo stesso modello di asset

figlio. Ad esempio, se nei parchi eolici sono presenti due diversi tipi di turbine eoliche, in cui tutte le turbine eoliche sono rappresentate dallo stesso modello di asset, è possibile definire una gerarchia per ogni tipo. Quindi, è possibile definire le metriche nel modello del parco eolico per calcolare statistiche indipendenti e combinate per ogni tipo di turbina eolica.

Un modello di asset principale può essere associato a più modelli di asset secondari. Ad esempio, se disponete di un parco eolico onshore e un parco eolico offshore rappresentati da due diversi modelli di asset, potete associare questi modelli di asset allo stesso modello di asset del parco eolico principale.

Un modello di asset figlio può anche essere associato a più modelli di asset principali. Ad esempio, se disponete di due diversi tipi di parchi eolici, in cui tutte le turbine eoliche sono rappresentate dallo stesso modello di asset, potete associare il modello di asset delle turbine eoliche a diversi modelli di asset di parchi eolici.

Note

Quando si definisce una gerarchia di modelli di asset, il modello di asset figlio deve essere ACTIVE o avere una versione precedente. ACTIVE Per ulteriori informazioni, consulta [Stati di asset e modelli](#).

Dopo aver definito i modelli di asset gerarchici e aver creato gli asset, puoi associare gli asset per completare la relazione padre-figlio. Per ulteriori informazioni, consultare [Create risorse per i modelli di asset in AWS IoT SiteWise](#) e [Associare e dissociare gli asset](#).

Argomenti

- [Definite le gerarchie dei modelli di asset \(console\)](#)
- [Definire le gerarchie degli asset \(AWS CLI\)](#)

Definite le gerarchie dei modelli di asset (console)

Quando si definisce una gerarchia per un modello di asset nella AWS IoT SiteWise console, si specificano i seguenti parametri:

- Nome della gerarchia: il nome della gerarchia, ad esempio. **Wind Turbines**
- Modello gerarchico: il modello di asset secondario.

- ID esterno della gerarchia (opzionale): si tratta di un ID definito dall'utente. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise .

Per ulteriori informazioni, consulta [Create un modello di asset \(console\)](#).

Definire le gerarchie degli asset (AWS CLI)

Quando definite una gerarchia per un modello di asset con l' AWS IoT SiteWise API, specificate i seguenti parametri:

- name— Il nome della gerarchia, ad esempio. **Wind Turbines**
- childAssetModelId— L'ID o l'ID esterno del modello di asset secondario per la gerarchia. È possibile utilizzare l'[ListAssetModels](#) operazione per trovare l'ID di un modello di asset esistente.

Example Definizione della gerarchia di esempio

L'esempio seguente mostra una gerarchia di modelli di asset che rappresenta la relazione tra un parco eolico e le turbine eoliche. Questo oggetto è un esempio di. [AssetModelHierarchy](#) Per ulteriori informazioni, consulta [Crea un modello di asset \(AWS CLI\)](#).

```
{
  ...
  "assetModelHierarchies": [
    {
      "name": "Wind Turbines",
      "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-1111EXAMPLE"
    },
  ],
}
```

Crea modelli di componenti

Utilizzate i modelli di AWS IoT SiteWise componenti per definire sottoassiemi a cui potete fare riferimento dai modelli di asset o da altri modelli di componenti. In questo modo, è possibile riutilizzare la definizione del componente su più altri modelli o più volte all'interno dello stesso modello.

Il processo di definizione di un modello di componente è molto simile alla definizione di un modello di asset. Come un modello di asset, un modello di componente ha un nome, una descrizione e

proprietà di asset. Tuttavia, i modelli di componenti non possono includere definizioni della gerarchia degli asset, poiché i modelli di componenti stessi non possono essere utilizzati per creare risorse direttamente. Inoltre, i modelli di componenti non possono definire allarmi.

Ad esempio, è possibile definire un componente per un servomotore con proprietà di temperatura del motore, temperatura dell'encoder e resistenza di isolamento. Quindi, potete definire un modello di asset per apparecchiature che contengono servomotori, come una macchina CNC.

Note

- Si consiglia di modellare l'operazione partendo dai nodi di livello inferiore. Ad esempio, create il componente del servomotore prima di creare il modello di asset della macchina CNC. I modelli di asset contengono riferimenti a modelli di componenti esistenti.
- Non è possibile creare una risorsa direttamente da un modello di componente. Per creare una risorsa che utilizza il componente, è necessario creare un modello di risorsa per la risorsa. Quindi, create un modello composito personalizzato che faccia riferimento al componente. Per ulteriori informazioni sulla creazione di modelli di asset, consulta [Crea modelli di asset in AWS IoT SiteWise](#). Per ulteriori informazioni sulla creazione di modelli compositi personalizzati, consulta [Crea modelli compositi personalizzati \(componenti\)](#).

Le sezioni seguenti descrivono come utilizzare l' AWS IoT SiteWise API per creare modelli di componenti.

Argomenti

- [Crea un modello di componente \(AWS CLI\)](#)
- [Esempio di modello di componente](#)

Crea un modello di componente (AWS CLI)

È possibile utilizzare AWS Command Line Interface (AWS CLI) per creare un modello di componente.

Utilizzate l'[CreateAssetModel](#) operazione per creare un modello di componente con proprietà. Questa operazione prevede un payload con la seguente struttura:

```
{
  "assetModelType": "COMPONENT_MODEL",
  "assetModelName": "String",
```

```
"assetModelDescription": "String",  
"assetModelProperties": Array of AssetModelProperty,  
}
```

Per creare un modello di componente (AWS CLI)

1. Create un file chiamato `component-model-payload.json` e poi copiate il seguente oggetto JSON nel file:

```
{  
  "assetModelType": "COMPONENT_MODEL",  
  "assetModelName": "",  
  "assetModelDescription": "",  
  "assetModelProperties": [  
  
  ]  
}
```

2. Utilizza l'editor di testo JSON preferito per modificare il file `component-model-payload.json` come segue:
 - a. Immettete un nome (`assetModelName`) per il modello del componente, ad esempio **Servo Motor** o **Servo Motor Model**. In questo caso, questo nome deve essere univoco per tutti i modelli di asset e i modelli di componenti del tuo account Regione AWS.
 - b. (Facoltativo) Inserite un ID esterno (`assetModelExternalId`) per il modello del componente. Si tratta di un ID definito dall'utente. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise .
 - c. (Facoltativo) Immetti una descrizione (`assetModelDescription`) per il modello di asset o rimuovi la coppia chiave-valore `assetModelDescription`.
 - d. (Facoltativo) Definite le proprietà degli asset (`assetModelProperties`) per il modello del componente. Per ulteriori informazioni, consulta [Definite le proprietà dei dati](#).
 - e. (Facoltativo) Aggiungi i tag (`tags`) per il modello di asset. Per ulteriori informazioni, consulta [Tagga le tue AWS IoT SiteWise risorse](#).
3. Eseguite il comando seguente per creare un modello di componente dalla definizione nel file JSON.

```
aws iotsitewise create-asset-model --cli-input-json file://component-model-payload.json
```

L'operazione restituisce una risposta che contiene il `assetModelId` riferimento a cui si fa riferimento quando si aggiunge un riferimento al modello di componente in un modello di asset o in un altro modello di componente. La risposta contiene anche lo stato del modello (`assetModelStatus.state`) che inizialmente è `CREATING`. Lo stato del modello di componente è valido `CREATING` fino alla propagazione delle modifiche.

Note

Il processo di creazione del modello di componente può richiedere fino a qualche minuto per i modelli complessi. Per verificare lo stato corrente del modello di componente, utilizzate l'[DescribeAssetModel](#) operazione specificando il `assetModelId`. Una volta raggiunto lo stato del modello di componente `ACTIVE`, potete aggiungere riferimenti al modello di componente nei modelli di asset o in altri modelli di componenti. Per ulteriori informazioni, consulta [Stati di asset e modelli](#).

4. (Facoltativo) Create modelli composti personalizzati per il modello di componente. Con i modelli composti personalizzati, è possibile raggruppare le proprietà all'interno del modello o includere un sottoinsieme facendo riferimento a un altro modello di componente. Per ulteriori informazioni, consulta [Crea modelli composti personalizzati \(componenti\)](#).

Esempio di modello di componente

Questa sezione contiene un esempio di definizione del modello di componente che è possibile utilizzare per creare un modello di componente con AWS CLI and AWS IoT SiteWise SDKs. Questo modello di componente rappresenta un servomotore che può essere utilizzato all'interno di un'altra apparecchiatura, ad esempio una macchina CNC.

Argomenti

- [Modello di componente del servomotore](#)

Modello di componente del servomotore

Il seguente modello di componente rappresenta un servomotore che può essere utilizzato all'interno di apparecchiature come macchine CNC. Il servomotore fornisce varie misurazioni, come temperature e resistenza elettrica. Queste misurazioni sono disponibili come proprietà sugli asset creati da modelli di asset che fanno riferimento al modello dei componenti del servomotore.


```
{
  "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"
    }
  ]
}
```

Definite le proprietà dei dati

Le proprietà degli asset sono le strutture all'interno di ogni asset che contengono i dati degli asset. Esistono vari tipi di proprietà:

- **Attributi:** proprietà generalmente statiche di una risorsa, come il produttore del dispositivo o l'area geografica. Per ulteriori informazioni, consulta [Definisci dati statici \(attributi\)](#).
- **Misurazioni:** i flussi di dati dei sensori del dispositivo raw di un asset, come i valori della velocità di rotazione con data e ora o i valori di temperatura con data e ora in gradi Celsius. Una misurazione è definita da un alias del flusso di dati. Per ulteriori informazioni, consulta [Definisci i flussi di dati provenienti dalle apparecchiature \(misurazioni\)](#).
- **Trasformazioni:** i valori delle serie temporali trasformati di una risorsa, come i valori di temperatura con data e ora in gradi Fahrenheit. Una trasformazione è definita da un'espressione con le sue variabili. Per ulteriori informazioni, consulta [Trasforma i dati \(trasformazioni\)](#).
- **Metriche:** i dati di una risorsa aggregati in un intervallo di tempo specificato, ad esempio la temperatura media oraria. Un parametro è definito da un intervallo di tempo e un'espressione con le sue variabili. Le espressioni metriche possono inserire le proprietà metriche degli asset

associati, in modo da poter calcolare le metriche che rappresentano l'operazione o un sottoinsieme dell'operazione. Per ulteriori informazioni, consulta [Aggrega i dati provenienti da proprietà e altre risorse \(metriche\)](#).

Per ulteriori informazioni, consulta [Crea modelli di asset in AWS IoT SiteWise](#).

Per un esempio di come utilizzare misurazioni, trasformazioni e parametri per calcolare l'efficienza complessiva delle apparecchiature (OEE), consulta [Calcola l'OEE in AWS IoT SiteWise](#).

Argomenti

- [Definisci dati statici \(attributi\)](#)
- [Definisci i flussi di dati provenienti dalle apparecchiature \(misurazioni\)](#)
- [Trasforma i dati \(trasformazioni\)](#)
- [Aggrega i dati provenienti da proprietà e altre risorse \(metriche\)](#)
- [Usa espressioni di formule](#)

Definisci dati statici (attributi)

Gli attributi delle risorse rappresentano informazioni generalmente statiche, come il produttore del dispositivo o la posizione geografica. Ogni asset include gli attributi del modello da cui è stato creato.

Argomenti

- [Definire gli attributi \(console\)](#)
- [Definisci gli attributi \(\)AWS CLI](#)

Definire gli attributi (console)

Quando si definisce un attributo per un modello di asset nella AWS IoT SiteWise console, si specificano i seguenti parametri:

- Nome: il nome della proprietà.
- Valore predefinito: (Facoltativo) Il valore predefinito per questo attributo. Gli asset creati dal modello hanno questo valore per l'attributo. Per ulteriori informazioni su come sostituire il valore predefinito in un asset creato da un modello, consulta [Aggiorna i valori degli attributi](#).
- Tipo di dati: il tipo di dati della proprietà, che è uno dei seguenti:

- **String**: una stringa con un massimo di 1024 byte.
- **Numero intero**: un numero intero con segno a 32 bit con intervallo [-2.147.483.648, 2.147.483.647].
- **Doppio**: un numero in virgola mobile con intervallo [-10¹⁰⁰, 10¹⁰⁰] e precisione doppia IEEE 754.
- **false** Booleano — **true** o.
- **ID esterno**: (Facoltativo) Si tratta di un ID definito dall'utente. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise .

Per ulteriori informazioni, consulta [Create un modello di asset \(console\)](#).

Definisci gli attributi (AWS CLI)

Quando definite un attributo per un modello di asset con l' AWS IoT SiteWise API, specificate i seguenti parametri:

- **name**— Il nome della proprietà.
- **defaultValue**— (Facoltativo) Il valore predefinito per questo attributo. Gli asset creati dal modello hanno questo valore per l'attributo. Per ulteriori informazioni su come sostituire il valore predefinito in un asset creato da un modello, consulta [Aggiorna i valori degli attributi](#).
- **dataType**— Il tipo di dati della proprietà, che è uno dei seguenti:
 - **STRING**— Una stringa con un massimo di 1024 byte.
 - **INTEGER**— Un numero intero con segno a 32 bit con intervallo [-2.147.483.648, 2.147.483.647].
 - **DOUBLE**— Un numero in virgola mobile con intervallo [-10¹⁰⁰, 10¹⁰⁰] e precisione doppia IEEE 754.
 - **BOOLEAN**— **true** oppure **false**
- **externalId**— (Facoltativo) Si tratta di un ID definito dall'utente. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise .

Example Definizione di attributo di esempio

Nell'esempio seguente viene illustrato un attributo che rappresenta il numero di modello di un asset con un valore predefinito. Questo oggetto è un esempio di un oggetto [AssetModelProperty](#) che contiene un [attributo](#). Puoi specificare questo oggetto come parte del payload della richiesta

[CreateAssetModel](#) per creare una proprietà dell'attributo. Per ulteriori informazioni, consulta [Crea un modello di asset \(AWS CLI\)](#).

```
{
  ...
  "assetModelProperties": [
    {
      "name": "Model number",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "BLT123"
        }
      }
    }
  ],
  ...
}
```

Definisci i flussi di dati provenienti dalle apparecchiature (misurazioni)

Una misurazione rappresenta il flusso di dati grezzi del sensore di un dispositivo, ad esempio valori di temperatura con data e ora o valori di rotazioni al minuto (RPM).

Argomenti

- [Definisci le misurazioni \(console\)](#)
- [Definisci le misurazioni \(\)AWS CLI](#)

Definisci le misurazioni (console)

Quando definite una misurazione per un modello di asset nella AWS IoT SiteWise console, specificate i seguenti parametri:

- Nome: il nome della proprietà.
- Unità — (Facoltativo) L'unità scientifica della proprietà, ad esempio mm o Celsius.
- Tipo di dati: il tipo di dati della proprietà, che è uno dei seguenti:
 - String: una stringa con un massimo di 1024 byte.
 - Numero intero: un numero intero con segno a 32 bit con intervallo [-2.147.483.648, 2.147.483.647].

- **Doppio**: un numero in virgola mobile con intervallo $[-10^{100}, 10^{100}]$ e precisione doppia IEEE 754.
- **false** Booleano — **true** o.
- **ID esterno**: (Facoltativo) Si tratta di un ID definito dall'utente. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise .

Per ulteriori informazioni, consulta [Create un modello di asset \(console\)](#).

Definisci le misurazioni (AWS CLI)

Quando definite una misurazione per un modello di asset con l' AWS IoT SiteWise API, specificate i seguenti parametri:

- **name**— Il nome della proprietà.
- **dataType**— Il tipo di dati della proprietà, che è uno dei seguenti:
 - **STRING**— Una stringa con un massimo di 1024 byte.
 - **INTEGER**— Un numero intero con segno a 32 bit con intervallo $[-2.147.483.648, 2.147.483.647]$.
 - **DOUBLE**— Un numero in virgola mobile con intervallo $[-10^{100}, 10^{100}]$ e precisione doppia IEEE 754.
 - **BOOLEAN**— **true** oppure **false**
- **unit**— (Facoltativo) L'unità scientifica della proprietà, ad esempio mm o Celsius.
- **externalId**— (Facoltativo) Si tratta di un ID definito dall'utente. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise .

Example Definizione di misurazione di esempio

Nell'esempio seguente viene illustrata una misurazione che rappresenta le letture del sensore di temperatura di un asset. Questo oggetto è un esempio di un oggetto [AssetModelProperty](#) che contiene una [misurazione](#). Puoi specificare questo oggetto come parte del payload della richiesta [CreateAssetModel](#) per creare una proprietà di misurazione. Per ulteriori informazioni, consulta [Crea un modello di asset \(AWS CLI\)](#).

La struttura [di misurazione](#) è una struttura vuota quando si definisce un modello di asset perché successivamente si configura ogni risorsa per utilizzare flussi di dati di dispositivo unici. Per ulteriori

informazioni su come collegare la proprietà di misurazione di un asset al flusso di dati dei sensori di un dispositivo, consulta [Gestisci i flussi di dati per AWS IoT SiteWise](#).

```
{
  ...
  "assetModelProperties": [
    {
      "name": "Temperature C",
      "dataType": "DOUBLE",
      "type": {
        "measurement": {}
      },
      "unit": "Celsius"
    }
  ],
  ...
}
```

Trasforma i dati (trasformazioni)

Le trasformazioni sono espressioni matematiche che mappano i punti dati delle proprietà degli asset da un modulo all'altro. Un'espressione di trasformazione è costituita da variabili di proprietà degli asset, valori letterali, operatori e funzioni. I punti dati trasformati mantengono una one-to-one relazione con i punti dati di input. AWS IoT SiteWise calcola un nuovo punto dati trasformato ogni volta che una delle proprietà di input riceve un nuovo punto dati.

Note

Per gli aggiornamenti delle proprietà con lo stesso timestamp, i valori di output possono essere sovrascritti dagli aggiornamenti di altre proprietà in entrata.

Ad esempio, se l'asset ha un flusso di misurazione della temperatura denominato `Temperature_C` con unità in Celsius, è possibile convertire ogni punto dati in Fahrenheit con la formula $\text{Temperature\_F} = 9/5 * \text{Temperature\_C} + 32$. Ogni volta AWS IoT SiteWise che riceve un punto dati nel flusso `Temperature_C` di misurazione, il `Temperature_F` valore corrispondente viene calcolato in pochi secondi e disponibile come proprietà. `Temperature_F`

Se la trasformazione contiene più di una variabile, il punto dati che arriva prima avvia immediatamente il calcolo. Consideriamo un esempio in cui un produttore di componenti utilizza una

trasformazione per monitorare la qualità del prodotto. Utilizzando uno standard diverso in base al tipo di parte, il produttore utilizza le seguenti misurazioni per rappresentare il processo:

- **Part_Number**- Una stringa che identifica il tipo di parte.
- **Good_Count**- Un numero intero che aumenta di uno se la parte soddisfa lo standard.
- **Bad_Count**- Un numero intero che aumenta di uno se la parte non soddisfa lo standard.

Il produttore crea anche una trasformazione `Quality_Monitor`, che è uguale `if(eq(Part_Number, "BLT123") and (Bad_Count / (Good_Count + Bad_Count) > 0.1), "Caution", "Normal")` a.

Questa trasformazione monitora la percentuale di parti difettose prodotte per un tipo di parte specifico. Se il numero di parte è BLT123 e la percentuale di parti difettose supera il 10 per cento (0,1), la trasformazione viene restituita. "Caution" Altrimenti, la trasformazione viene "Normal" restituita.

Note

- Se `Part_Number` riceve un nuovo punto dati prima di altre misurazioni, la `Quality_Monitor` trasformazione utilizza il nuovo `Part_Number` valore e i `Bad_Count` valori `Good_Count` e più recenti. Per evitare errori, `Good_Count` reimpostatelo `Bad_Count` prima del successivo ciclo di produzione.
- Utilizzate [le metriche](#) se desiderate valutare le espressioni solo dopo che tutte le variabili hanno ricevuto nuovi punti dati.

Argomenti

- [Definisci le trasformazioni \(console\)](#)
- [Definisci le trasformazioni \(\)AWS CLI](#)

Definisci le trasformazioni (console)

Quando definite una trasformazione per un modello di asset nella AWS IoT SiteWise console, specificate i seguenti parametri:

- **Nome**: il nome della proprietà.

- **Unità** — (Facoltativo) L'unità scientifica della proprietà, ad esempio mm o Celsius.
- **Tipo di dati**: il tipo di dati della trasformazione, che può essere Double o String.
- **ID esterno**: (Facoltativo) Si tratta di un ID definito dall'utente. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise .
- **Formula**: l'espressione di trasformazione. Le espressioni di trasformazione non possono utilizzare funzioni di aggregazione o funzioni temporali. Per aprire la funzione di completamento automatico, inizia a digitare o premi il tasto freccia giù. Per ulteriori informazioni, consulta [Usa espressioni di formule](#).

Important

Le trasformazioni possono immettere proprietà di tipo intero, doppio, booleano o stringa. I valori booleani vengono convertiti in 0 (false) e 1 (true).

Le trasformazioni devono inserire una o più proprietà che non sono attributi e un numero qualsiasi di proprietà degli attributi. AWS IoT SiteWise calcola un nuovo punto dati trasformato ogni volta che la proprietà di input che non è un attributo riceve un nuovo punto dati. I nuovi valori degli attributi non avviano gli aggiornamenti delle trasformazioni. La stessa frequenza di richiesta per le operazioni API relative ai dati delle proprietà degli asset si applica ai risultati del calcolo delle trasformazioni.

Le espressioni delle formule possono generare solo valori doppi o stringhe. Le espressioni annidate possono generare altri tipi di dati, ad esempio stringhe, ma la formula nel suo insieme deve restituire un numero o una stringa. È possibile utilizzare la [funzione jp](#) per convertire una stringa in un numero. Il valore booleano deve essere 1 (vero) o 0 (falso). Per ulteriori informazioni, consulta [Valori indefiniti, infiniti e di overflow](#).

Per ulteriori informazioni, consulta [Create un modello di asset \(console\)](#).

Definisci le trasformazioni (AWS CLI)

Quando definite una trasformazione per un modello di asset con l' AWS IoT SiteWise API, specificate i seguenti parametri:

- **name**— Il nome della proprietà.
- **unit**— (Facoltativo) L'unità scientifica della proprietà, ad esempio mm o Celsius.
- **dataType**— Il tipo di dati della trasformazione, che deve essere DOUBLE o STRING.

- **externalId**— (Facoltativo) Si tratta di un ID definito dall'utente. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise.
- **expression**— L'espressione di trasformazione. Le espressioni di trasformazione non possono utilizzare funzioni di aggregazione o funzioni temporali. Per ulteriori informazioni, consulta [Usa espressioni di formule](#).
- **variables**— L'elenco di variabili che definisce le altre proprietà della risorsa da utilizzare nell'espressione. Ogni struttura di variabile contiene un nome semplice da utilizzare nell'espressione e una struttura **value** che specifica la proprietà da collegare alla variabile stessa. La struttura **value** contiene le seguenti informazioni:
 - **propertyId**— L'ID della proprietà da cui immettere i valori. È possibile utilizzare il nome della proprietà anziché il relativo ID.

Important

Le trasformazioni possono immettere proprietà di tipo intero, doppio, booleano o stringa. I valori booleani vengono convertiti in 0 (false) e 1 (true).

Le trasformazioni devono inserire una o più proprietà che non sono attributi e un numero qualsiasi di proprietà degli attributi. AWS IoT SiteWise calcola un nuovo punto dati trasformato ogni volta che la proprietà di input che non è un attributo riceve un nuovo punto dati. I nuovi valori degli attributi non avviano gli aggiornamenti delle trasformazioni. La stessa frequenza di richiesta per le operazioni API relative ai dati delle proprietà degli asset si applica ai risultati del calcolo delle trasformazioni.

Le espressioni delle formule possono generare solo valori doppi o stringhe. Le espressioni annidate possono generare altri tipi di dati, ad esempio stringhe, ma la formula nel suo insieme deve restituire un numero o una stringa. È possibile utilizzare la [funzione jp](#) per convertire una stringa in un numero. Il valore booleano deve essere 1 (vero) o 0 (falso). Per ulteriori informazioni, consulta [Valori indefiniti, infiniti e di overflow](#).

Example definizione di trasformazione

Nell'esempio seguente viene illustrata una proprietà di trasformazione che converte i dati di misurazione della temperatura di un asset da Celsius a Fahrenheit. Questo oggetto è un esempio di un oggetto [AssetModelProperty](#) che contiene un [Transform](#). Puoi specificare questo oggetto come parte del payload della richiesta [CreateAssetModel](#) per creare una proprietà di trasformazione. Per ulteriori informazioni, consulta [Crea un modello di asset \(AWS CLI\)](#).

```

{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "Temperature F",
      "dataType": "DOUBLE",
      "type": {
        "transform": {
          "expression": "9/5 * temp_c + 32",
          "variables": [
            {
              "name": "temp_c",
              "value": {
                "propertyId": "Temperature C"
              }
            }
          ]
        }
      },
      "unit": "Fahrenheit"
    }
  ],
  ...
}

```

Example definizione di trasformazione che contiene tre variabili

L'esempio seguente dimostra una proprietà di trasformazione che restituisce un messaggio di avviso ("Caution") se più del 10 per cento delle BLT123 parti non soddisfa lo standard. Altrimenti, restituisce un messaggio informativo ("Normal").

```

{
  ...
  "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"
        }
      }
    ]
  }
}
...
}

```

Aggrega i dati provenienti da proprietà e altre risorse (metriche)

Le metriche sono espressioni matematiche che utilizzano funzioni di aggregazione per elaborare tutti i punti dati di input e generare un singolo punto dati per intervallo di tempo specificato. Un parametro, ad esempio, può calcolare la temperatura oraria media da un flusso di dati di temperatura.

I parametri possono immettere dati dai parametri degli asset associati, in modo da poter calcolare le statistiche che forniscono informazioni dettagliate sull'operazione o su un sottoinsieme dell'operazione. Un parametro, ad esempio, può calcolare la temperatura oraria media di tutte le turbine eoliche di una centrale eolica. Per ulteriori informazioni su come definire le associazioni tra asset, consulta [Definite le gerarchie dei modelli di asset](#).

Le metriche possono anche inserire dati da altre proprietà senza aggregare i dati su ogni intervallo di tempo. Se specifichi un [attributo](#) in una formula, AWS IoT SiteWise utilizza il valore [più recente](#) per quell'attributo quando calcola la formula. Se specifichi una metrica in una formula, AWS IoT SiteWise utilizza l'[ultimo](#) valore per l'intervallo di tempo in cui calcola la formula. Ciò significa che puoi definire

metriche come $OEE = Availability * Quality * Performance$ AvailabilityQuality, dove e Performance sono tutte le altre metriche sullo stesso modello di asset.

AWS IoT SiteWise inoltre, calcola automaticamente un set di metriche di aggregazione di base per tutte le proprietà degli asset. Per ridurre i costi di calcolo, puoi utilizzare questi aggregati anziché definire parametri personalizzati per i calcoli di base. Per ulteriori informazioni, consulta [Interroga gli aggregati delle proprietà degli asset in AWS IoT SiteWise](#).

Argomenti

- [Definisci le metriche \(console\)](#)
- [Definisci le metriche \(\)AWS CLI](#)

Definisci le metriche (console)

Quando definisci una metrica per un modello di asset nella AWS IoT SiteWise console, specifichi i seguenti parametri:

- Nome: il nome della proprietà.
- Tipo di dati: il tipo di dati della trasformazione, che può essere Double o String.
- ID esterno: (Facoltativo) Si tratta di un ID definito dall'utente. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise .
- Formula: l'espressione metrica. Le espressioni metriche possono utilizzare [funzioni di aggregazione](#) per immettere dati da una proprietà per tutte le risorse associate in una gerarchia. Inizia a digitare o premi il tasto freccia giù per aprire la funzione di completamento automatico. Per ulteriori informazioni, consulta [Usa espressioni di formule](#).

Important

Le metriche possono essere solo proprietà di tipo intero, doppio, booleano o stringa. I valori booleani vengono convertiti in 0 (false) e 1 (true).

Le variabili di input in una espressione parametrica devono presentare lo stesso intervallo di tempo del parametro di output.

Le espressioni delle formule possono generare solo valori doppi o stringhe. Le espressioni annidate possono generare altri tipi di dati, ad esempio stringhe, ma la formula nel suo insieme deve restituire un numero o una stringa. È possibile utilizzare la [funzione jp](#) per

convertire una stringa in un numero. Il valore booleano deve essere 1 (vero) o 0 (falso). Per ulteriori informazioni, consulta [Valori indefiniti, infiniti e di overflow](#).

- Intervallo di tempo: l'intervallo di tempo metrico. AWS IoT SiteWise supporta i seguenti intervalli di tempo a finestra rotante, in cui ogni intervallo inizia alla fine di quello precedente:
 - da 1 minuto a 1 minuto, calcolato alla fine di ogni minuto (00:00:00, 00:01:00, 12:02:00 e così via).
 - Da 5 minuti a 5 minuti, calcolati alla fine di ogni cinque minuti a partire dall'ora (00:00:00, 12:05:00, 00:10:00 e così via).
 - da 15 minuti a 15 minuti, calcolati alla fine di ogni quindici minuti a partire dall'ora (00:00:00, 12:15:00, 00:30:00 e così via).
 - da 1 ora a 1 ora (60 minuti), calcolata alla fine di ogni ora in formato UTC (00:00:00, 01:00:00, 02:00:00 e così via).
 - 1 giorno — 1 giorno (24 ore), calcolato alla fine di ogni giornata in UTC (00:00 di lunedì, 00:00 di martedì e così via).
 - da 1 settimana a 1 settimana (7 giorni), calcolata alla fine di ogni domenica in UTC (ogni lunedì alle 00:00).
- Intervallo personalizzato: puoi inserire qualsiasi intervallo di tempo compreso tra un minuto e una settimana.
- Data di offset: (Facoltativa) La data di riferimento a partire dalla quale aggregare i dati.
- Tempo di offset: (Facoltativo) L'ora di riferimento a partire dalla quale aggregare i dati. L'ora di offset deve essere compresa tra 00:00:00 e 23:59:59.
- Fuso orario di offset - (Facoltativo) Il fuso orario per l'offset. Se non è specificato, il fuso orario di offset predefinito è l'Universal Coordinated Time (UTC).

Fusi orari supportati

- (UTC+ 00:00) Universal Coordinated Time
- (UTC+ 01:00) Ora centrale europea
- (UTC+ 02:00) Europa orientale
- (UTC03+:00) Ora dell'Africa orientale
- (UTC+ 04:00) Ora del Vicino Oriente
- (UTC+ 05:00) Ora del Pakistan di Lahore
- (UTC+ 05:30) Ora solare dell'India

- (UTC+ 06:00) Ora solare del Bangladesh
- (UTC+ 07:00) Ora solare del Vietnam
-
- (UTC+ 09:00) Ora solare del Giappone
- (UTC+ 09:30) Ora centrale dell'Australia
- (UTC+ 10:00) Ora orientale dell'Australia
- (UTC+ 11:00) Ora solare delle Salomone
- (UTC+ 12:00) Ora solare della Nuova Zelanda
- (UTC- 11:00) Ora delle Isole Midway
- (UTC- 10:00) Ora solare delle Hawaii
- (UTC- 09:00) Ora solare dell'Alaska
- (UTC- 08:00) Ora solare del Pacifico
- (UTC- 07:00) Ora solare di Phoenix
- (UTC- 06:00) Ora solare centrale
- (UTC- 05:00) Ora solare orientale
- (UTC- 04:00) Ora di Porto Rico e Isole Vergini Americane
- (UTC- 03:00) Ora solare dell'Argentina
- (UTC- 02:00) Ora della Georgia del Sud
- (UTC- 01:00) Ora dell'Africa centrale

Example intervallo di tempo personalizzato con un offset (console)

L'esempio seguente mostra come definire un intervallo di tempo di 12 ore con un offset il 20 febbraio 2021 alle 18:30:30 (PST).

Per definire un intervallo personalizzato con un offset

1. Per Intervallo di tempo, scegliete Intervallo personalizzato.
2. Per Intervallo di tempo, effettuate una delle seguenti operazioni:
 - Immettete **12**, quindi scegliete le ore.
 - Inserisci **720**, quindi scegli minuti.
 - Inserisci **43200**, quindi scegli secondi.

 Important

L'intervallo di tempo deve essere un numero intero indipendentemente dall'unità.

3. Per la data di offset, scegli 20/02/2021.
4. Per Tempo di offset, immettete. **18:30:30**
5. Per il fuso orario di Offset, scegliete (UTC- 08:00) Pacific Standard Time.

Se crei la metrica il 1° luglio 2021, prima o alle 18:30:30 (PST), otterrai il primo risultato di aggregazione il 1° luglio 2021 alle 18:30:30 (PST). Il secondo risultato di aggregazione è il 2 luglio 2021 alle 06:30:30 (PST) e così via.

Definisci le metriche ()AWS CLI

Quando definite una metrica per un modello di asset con l' AWS IoT SiteWise API, specificate i seguenti parametri:

- **name**— Il nome della proprietà.
- **dataType**— Il tipo di dati della metrica, che può essere DOUBLE oSTRING.
- **externalId**— (Facoltativo) Si tratta di un ID definito dall'utente. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise.
- **expression**— L'espressione metrica. Le espressioni metriche possono utilizzare [funzioni di aggregazione](#) per immettere dati da una proprietà per tutte le risorse associate in una gerarchia. Per ulteriori informazioni, consulta [Usa espressioni di formule](#).
- **window**— L'intervallo di tempo e l'offset della finestra di rotazione della metrica, in cui ogni intervallo inizia quando termina quello precedente:
 - **interval**— L'intervallo di tempo per la finestra di rotazione. L'intervallo di tempo deve essere compreso tra un minuto e una settimana.
 - **offsets**— L'offset della finestra ribaltabile.

Per ulteriori informazioni, consulta [TumblingWindow](#) nella documentazione di riferimento dell'API AWS IoT SiteWise .

Example intervallo di tempo personalizzato con un offset ()AWS CLI

L'esempio seguente mostra come definire un intervallo di tempo di 12 ore con un offset il 20 febbraio 2021 alle 18:30:30 (PST).

```
{
  "window": {
    "tumbling": {
      "interval": "12h",
      "offset": " 2021-07-23T18:30:30-08"
    }
  }
}
```

Se crei la metrica il 1° luglio 2021, prima o alle 18:30:30 (PST), otterrai il primo risultato di aggregazione il 1° luglio 2021 alle 18:30:30 (PST). Il secondo risultato di aggregazione è il 2 luglio 2021 alle 06:30:30 (PST) e così via.

- **variables**— L'elenco di variabili che definisce le altre proprietà della risorsa o delle risorse secondarie da utilizzare nell'espressione. Ogni struttura di variabile contiene un nome semplice da utilizzare nell'espressione e una struttura **value** che specifica la proprietà da collegare alla variabile stessa. La struttura **value** contiene le seguenti informazioni:
 - **propertyId**— L'ID della proprietà da cui estrarre i valori. È possibile utilizzare il nome della proprietà anziché il relativo ID se la proprietà è definita nel modello corrente (anziché definita in un modello da una gerarchia).
 - **hierarchyId**— (Facoltativo) L'ID della gerarchia da cui interrogare gli asset secondari per la proprietà. È possibile utilizzare il nome della definizione della gerarchia anziché il relativo ID. Se omettete questo valore, AWS IoT SiteWise trova la proprietà nel modello corrente.

Important

Le metriche possono essere solo proprietà di tipo intero, doppio, booleano o stringa. I valori booleani vengono convertiti in 0 (false) e (true). 1

Le variabili di input in una espressione parametrica devono presentare lo stesso intervallo di tempo del parametro di output.

Le espressioni delle formule possono generare solo valori doppi o stringhe. Le espressioni annidate possono generare altri tipi di dati, ad esempio stringhe, ma la formula nel suo

insieme deve restituire un numero o una stringa. È possibile utilizzare la [funzione jp](#) per convertire una stringa in un numero. Il valore booleano deve essere 1 (vero) o 0 (falso). Per ulteriori informazioni, consulta [Valori indefiniti, infiniti e di overflow](#).

- `unit`— (Facoltativo) L'unità scientifica della proprietà, ad esempio mm o Celsius.

Example Definizione del parametro di esempio

Nell'esempio seguente viene illustrata una proprietà parametro che aggrega i dati di misurazione della temperatura di un asset per calcolare la temperatura oraria massima in Fahrenheit. Questo oggetto è un esempio di un [AssetModelProperty](#) che contiene una [metrica](#). Puoi specificare questo oggetto come parte del payload della richiesta [CreateAssetModel](#) per creare una proprietà di parametro. Per ulteriori informazioni, consulta [Crea un modello di asset \(AWS CLI\)](#).

```
{
  ...
  "assetModelProperty": [
    ...
    {
      "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 Esempio di definizione della metrica che inserisce dati dalle risorse associate

L'esempio seguente mostra una proprietà metrica che aggrega i dati sulla potenza media di più turbine eoliche per calcolare la potenza media totale di un parco eolico. [Questo oggetto è un esempio di un oggetto che contiene una metrica AssetModelProperty](#). Puoi specificare questo oggetto come parte del payload della richiesta [CreateAssetModel](#) per creare una proprietà di parametro.

```

{
  ...
  "assetModelProperty": [
    ...
    {
      "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"
    }
  ],
  ...
}

```

```
}
```

Usa espressioni di formule

Con le espressioni di formula, è possibile definire le funzioni matematiche per trasformare e aggregare i dati industriali grezzi e ottenere informazioni dettagliate sulla propria operazione. Le espressioni di formule combinano valori letterali, operatori, funzioni e variabili per elaborare i dati. Per ulteriori informazioni su come definire le proprietà degli asset che utilizzano espressioni di formule, consulta [Trasforma i dati \(trasformazioni\)](#) e [Aggrega i dati provenienti da proprietà e altre risorse \(metriche\)](#). Le trasformazioni e le metriche sono proprietà delle formule.

Argomenti

- [Usa le variabili nelle espressioni delle formule](#)
- [Usa valori letterali nelle espressioni delle formule](#)
- [Usa gli operatori nelle espressioni delle formule](#)
- [Usa le costanti nelle espressioni delle formule](#)
- [Usa le funzioni nelle espressioni delle formule](#)
- [Tutorial sulle espressioni di formule](#)

Usa le variabili nelle espressioni delle formule

Le variabili rappresentano le proprietà AWS IoT SiteWise degli asset nelle espressioni delle formule. Utilizzate le variabili per inserire valori da altre proprietà degli asset nelle espressioni, in modo da poter elaborare i dati provenienti da proprietà costanti ([attributi](#)), flussi di dati non elaborati ([misurazioni](#)) e altre proprietà delle formule.

Le variabili possono rappresentare le proprietà degli asset dello stesso modello di asset o dei modelli di asset secondari associati. Solo le formule metriche possono inserire variabili da modelli di asset secondari.

Le variabili vengono identificate con nomi diversi nella console e nell'API.

- AWS IoT SiteWise console: utilizza i nomi delle proprietà degli asset come variabili nelle espressioni.
- AWS IoT SiteWise API (AWS CLI, AWS SDKs): definisci le variabili con la [ExpressionVariable](#) struttura, che richiede un nome di variabile e un riferimento a una proprietà dell'asset. Il nome della variabile può contenere lettere minuscole, numeri e caratteri di

sottolineatura. Quindi, utilizzate i nomi delle variabili per fare riferimento alle proprietà delle risorse nelle espressioni.

I nomi delle variabili distinguono tra maiuscole e minuscole

Per ulteriori informazioni, consulta [Definizione delle trasformazioni](#) e [Definizione delle metriche](#).

Utilizzate le variabili per fare riferimento alle proprietà

Il valore di una variabile definisce la proprietà a cui fa riferimento. AWS IoT SiteWise offre diversi modi per eseguire questa operazione.

- Per ID della proprietà: è possibile specificare l'ID univoco della proprietà (UUID) per identificarla.
- Per nome: se la proprietà appartiene allo stesso modello di asset, è possibile specificarne il nome nel campo ID della proprietà.
- Per percorso: un valore variabile può fare riferimento a una proprietà tramite il relativo percorso. Per ulteriori informazioni, consulta [Utilizzate i percorsi per fare riferimento alle proprietà personalizzate del modello composito](#).

Note

Le variabili non sono supportate dalla AWS IoT SiteWise console. Sono utilizzate dall' AWS IoT SiteWise API, tra cui AWS Command Line Interface (AWS CLI) e AWS SDKs.

Una variabile ricevuta in una risposta da AWS IoT SiteWise include informazioni complete sul valore, inclusi l'ID e il percorso.

Tuttavia, quando si passa una variabile a AWS IoT SiteWise (ad esempio, in una chiamata «crea» o «aggiorna»), è sufficiente specificare solo una di queste. Ad esempio, se si specifica il percorso, non è necessario fornire l'ID.

Usa valori letterali nelle espressioni delle formule

AWS IoT SiteWise supporta l'uso di valori letterali nelle espressioni e nelle formule. I letterali sono valori fissi che rappresentano un tipo di dati specifico. In AWS IoT SiteWise, è possibile definire numeri e stringhe letterali nelle espressioni delle formule. I valori letterali possono essere utilizzati in vari contesti, tra cui trasformazioni di dati, condizioni di allarme e calcoli di visualizzazione.

- Numeri

Usa numeri e notazione scientifica per definire numeri interi e doppi. È possibile utilizzare la [notazione E per esprimere numeri con notazione scientifica](#).

Esempi: 1, 2.0, .9, -23.1 7.89e3 3.4E-5

- Stringhe

Usa i caratteri ' (virgolette) e " (virgolette doppie) per definire le stringhe. Il tipo di citazione per l'inizio e la fine devono corrispondere. Per evitare una virgoletta che corrisponde a quella utilizzata per dichiarare una stringa, includi quella virgoletta due volte. Questo è l'unico carattere di escape nelle AWS IoT SiteWise stringhe.

Esempi: 'active', "inactive", '{"temp": 52}', '{"temp": "high"}'

Usa gli operatori nelle espressioni delle formule

È possibile utilizzare i seguenti operatori comuni nelle espressioni di formule.

Operatore	Descrizione
+	Se entrambi gli operandi sono numeri, questo operatore aggiunge gli operandi sinistro e destro. Se uno degli operandi è una stringa, questo operatore concatena gli operandi sinistro e destro come stringhe. Ad esempio, l'espressione restituisce. <code>1 + 2 + " is three"</code> <code>"3 is three"</code> La stringa concatenata può contenere fino a 1024 caratteri. Se la stringa supera i 1024 caratteri, AWS IoT SiteWise non restituisce un punto dati per quel calcolo.
-	Sottrae l'operando destro dall'operando sinistro.

Operatore	Descrizione
	È possibile utilizzare questo operatore solo con operandi numerici.
/	Divide l'operando sinistro per l'operando destro. È possibile utilizzare questo operatore solo con operandi numerici.
*	Moltiplica gli operandi sinistro e destro. È possibile utilizzare questo operatore solo con operandi numerici.
^	Solleva l'operando di sinistra alla potenza dell'operando destro (esponenziazione). È possibile utilizzare questo operatore solo con operandi numerici.
%	Restituisce il resto della divisione dell'operando sinistro per l'operando destro. Il segno del risultato è identico a quello dell'operando sinistro. Questo comportamento è diverso dall'operazione del modulo. È possibile utilizzare questo operatore solo con operandi numerici.
$x < y$	Restituisce 1 se x è minore di y , altrimenti 0.
$x > y$	Restituisce 1 se x è maggiore di y , altrimenti 0.
$x \leq y$	Restituisce 1 se x è minore o uguale a y , altrimenti 0.
$x \geq y$	Restituisce 1 se x è maggiore o uguale a y , altrimenti 0.

Operatore	Descrizione
<code>x == y</code>	Restituisce 1 se x è uguale a y, altrimenti 0.
<code>x != y</code>	Restituisce 1 se non x è uguale a y, altrimenti 0.
<code>!x</code>	Restituisce 1 se x viene valutato come 0 (falso), altrimenti 0. x viene valutato falso se: • x è un operando numerico e viene valutato in. 0 • x viene valutato in una stringa vuota. • x viene valutato in un array vuoto. • x viene valutato in. None
<code>x and y</code>	Restituisce 0 se x viene valutato come 0 (false). Altrimenti, restituisce il risultato valutato di. y x o y viene valutato falso se: • x o y è un operando numerico e viene valutato come. 0 • x o y viene valutato in una stringa vuota. • x o y viene valutato in un array vuoto. • x o y viene valutato in. None

Operatore	Descrizione
<code>x or y</code>	Restituisce 1 se x viene valutato come 1 (true). Altrimenti, restituisce il risultato valutato di. y xo y viene valutato falso se: • xor y è un operando numerico e viene valutato come. 0 • xo y viene valutato in una stringa vuota. • xo y viene valutato in un array vuoto. • xo y viene valutato in. None
<code>not x</code>	Restituisce 1 se x viene valutato come 0 (false), altrimenti. 0 xviene valutato falso se: • xè un operando numerico e viene valutato in. 0 • xviene valutato in una stringa vuota. • xviene valutato in un array vuoto. • xviene valutato in. None
<code>[]</code> <code>s[index]</code>	Restituisce il carattere in corrispondenza di un indice <code>index</code> della stringas. È equivalente alla sintassi dell'indice in Python. Example Esempi • "Hello!"[1] restituisce e. • "Hello!"[-2] restituisce o.

Operatore	Descrizione
[] <code>s[start:end:step]</code>	Restituisce una parte della stringa. <code>s</code> È equivalente alla sintassi slice in Python. Questo operatore ha i seguenti argomenti: • <code>start</code>— (Facoltativo) L'indice iniziale inclusivo della fetta. L'impostazione predefinita è 0. • <code>end</code>— (Facoltativo) L'indice finale esclusivo della fetta. Il valore predefinito è la lunghezza della stringa. • <code>step</code>— (Facoltativo) Il numero da incrementare per ogni passaggio della fetta. Ad esempio, potete specificare di 2 restituire una sezione con ogni altro carattere o specificare di -1 invertire la sezione. L'impostazione predefinita è 1. È possibile omettere l'argomento <code>step</code> per utilizzarne il valore predefinito. Ad esempio, <code>s[1:4:1]</code> è uguale a <code>s[1:4]</code>. Gli argomenti devono essere numeri interi o la costante none. Se si specificano <code>none</code>, AWS IoT SiteWise utilizza il valore predefinito per quell'argomento. Example Esempi • <code>"Hello!"[1:4]</code> restituisce <code>"ell"</code>. • <code>"Hello!"[:2]</code> restituisce <code>"He"</code>. • <code>"Hello!"[3:]</code> restituisce <code>"lo!"</code>. • <code>"Hello!"[:-4]</code> restituisce <code>"He"</code>. • <code>"Hello!"[:2]</code> restituisce <code>"Hlo"</code>. • <code>"Hello!"[::-1]</code> restituisce <code>"!olleH"</code>.

Usa le costanti nelle espressioni delle formule

In AWS IoT SiteWise, è possibile utilizzare costanti nelle espressioni e nelle formule per rappresentare valori fissi o parametri predefiniti. Le costanti possono essere utilizzate in vari contesti, ad esempio trasformazioni di dati, condizioni di allarme o calcoli di visualizzazione. Utilizzando le costanti, è possibile semplificare le espressioni e renderle più leggibili e gestibili.

È possibile utilizzare le seguenti costanti matematiche comuni nelle espressioni. Tutte le costanti non fanno distinzione tra maiuscole e minuscole.

Note

Se si definisce una variabile con lo stesso nome di una costante, la variabile sostituisce la costante.

Costante	Descrizione
pi	Il numero pi (π): 3.141592653589793
e	Il numero e: 2.718281828459045
true	Equivalente al numero 1. Nel AWS IoT SiteWise, i booleani si convertono nei loro equivalenti numerici.
false	Equivalente al numero 0. In AWS IoT SiteWise, i booleani vengono convertiti nei loro equivalenti numerici.
none	Equivalente a nessun valore. È possibile utilizzare questa costante per non restituire nulla come risultato di un' espressione condizionale .

Usa le funzioni nelle espressioni delle formule

È possibile utilizzare le seguenti funzioni per operare sui dati nelle espressioni delle formule.

Le trasformazioni e le metriche supportano diverse funzioni. La tabella seguente indica quali tipi di funzioni sono compatibili con ogni tipo di proprietà della formula.

 Note

È possibile includere un massimo di 10 funzioni in un'espressione di formula.

Tipo di funzione	Trasformazioni	Metriche
Usa funzioni comuni nelle espressioni delle formule	 Sì	 Sì
Usa le funzioni di confronto nelle espressioni delle formule	 Sì	 Sì
Usa le funzioni condizionali nelle espressioni delle formule	 Sì	 Sì
Usa le funzioni di stringa nelle espressioni delle formule	 Sì	 Sì
Usa le funzioni di aggregazione nelle espressioni delle formule	 No	 Sì

Tipo di funzione	Trasformazioni	Metriche
Usa le funzioni temporali nelle espressioni delle formule	 Sì	 Sì
Utilizza le funzioni di data e ora nelle espressioni delle formule	 Sì	 Sì

Sintassi della funzione

È possibile utilizzare la seguente sintassi per creare funzioni:

Sintassi regolare

Con la sintassi normale, il nome della funzione è seguito da parentesi con zero o più argomenti.

function_name(argument1, argument2, argument3, ...). Ad esempio, le funzioni con la sintassi normale potrebbero essere simili a `log(x)` e `contains(s, substring)`.

Sintassi uniforme delle chiamate di funzione (UFCS)

UFCS consente di chiamare funzioni utilizzando la sintassi per le chiamate di metodo nella programmazione orientata agli oggetti. Con UFCS, il primo argomento è seguito da dot (`.`), quindi dal nome della funzione e dagli argomenti rimanenti (se presenti) tra parentesi.

argument1.function_name(argument2, argument3, ...). Ad esempio, le funzioni con UFCS potrebbero essere simili `x.log()` a e `s.contains(substring)`

È inoltre possibile utilizzare UFCS per concatenare le funzioni successive. AWS IoT SiteWise utilizza il risultato della valutazione della funzione corrente come primo argomento per la funzione successiva.

Ad esempio, è possibile utilizzare `message.jp('$ .status').lower().contains('fail')` invece di `contains(lower(jp(message, '$ .status')), 'fail')`.

Per ulteriori informazioni, visita il sito Web del [linguaggio di programmazione D](#).

 Note

È possibile utilizzare UFCS per tutte le AWS IoT SiteWise funzioni. AWS IoT SiteWise le funzioni non distinguono tra maiuscole e minuscole. Ad esempio, è possibile utilizzare `lower(s)` e in modo `Lower(s)` intercambiabile.

Usa funzioni comuni nelle espressioni delle formule

Nelle [trasformazioni](#) e nelle [metriche](#), è possibile utilizzare le seguenti funzioni per calcolare le funzioni matematiche comuni nelle trasformazioni e nelle metriche.

Funzione	Descrizione
<code>abs(x)</code>	Restituisce il valore assoluto di x.
<code>acos(x)</code>	Restituisce l'arccoseno di x.
<code>asin(x)</code>	Restituisce l'arcsine di x.
<code>atan(x)</code>	Restituisce l'arctangente di x.
<code>cbrt(x)</code>	Restituisce la radice cubica di x.
<code>ceil(x)</code>	Restituisce il numero intero più vicino maggiore di x.
<code>cos(x)</code>	Restituisce il coseno di x.
<code>cosh(x)</code>	Restituisce il coseno iperbolico di x.
<code>cot(x)</code>	Restituisce la cotangente di x.
<code>exp(x)</code>	Restituisce e al potere di x.
<code>expm1(x)</code>	Restituisce $\exp(x) - 1$. Utilizzate questa funzione $\exp(x) - 1$ per calcolare con maggiore precisione valori piccoli di x.

Funzione	Descrizione
<code>floor(x)</code>	Restituisce il numero intero più vicino inferiore a x .
<code>log(x)</code>	Restituisce $\log_e$ (base e) di x .
<code>log10(x)</code>	Restituisce $\log_{10}$ (base 10) di x .
<code>log1p(x)</code>	Restituisce $\log(1 + x)$. Utilizzare questa funzione $\log(1 + x)$ per calcolare con maggiore precisione valori piccoli di x .
<code>log2(x)</code>	Restituisce $\log_2$ (base 2) di x .
<code>pow(x, y)</code>	Restituisce x al potere di y . Questo è equivalente a x^y .
<code>signum(x)</code>	Restituisce il segno di x (-1 per input negativi, 0 per input pari a zero, +1 per input positivi).
<code>sin(x)</code>	Restituisce il seno di x .
<code>sinh(x)</code>	Restituisce il seno iperbolico di x .
<code>sqrt(x)</code>	Restituisce la radice quadrata di x .
<code>tan(x)</code>	Restituisce la tangente di x .
<code>tanh(x)</code>	Restituisce la tangente iperbolica di x .

Usa le funzioni di confronto nelle espressioni delle formule

Nelle [trasformazioni](#) e nelle [metriche](#), è possibile utilizzare le seguenti funzioni di confronto per confrontare due valori e generare 1 (vero) o 0 (falso). AWS IoT SiteWise [confronta le stringhe in base all'ordine lessicografico](#).

Funzione	Descrizione
<code>gt(x, y)</code>	Restituisce 1 se x è maggiore di y, altrimenti 0 ($x > y$). Questa funzione non restituisce un valore se x e y sono tipi incompatibili, come un numero e una stringa.
<code>gte(x, y)</code>	Restituisce 1 se x è maggiore o uguale a y, altrimenti 0 ($x \geq y$). AWS IoT SiteWise considera gli argomenti uguali se rientrano in una tolleranza relativa di $1E-9$. Si comporta in modo simile alla funzione isclose in Python. Questa funzione non restituisce un valore se x e y sono tipi incompatibili, come un numero e una stringa.
<code>eq(x, y)</code>	Restituisce 1 se x è uguale a y, altrimenti 0 ($x == y$). AWS IoT SiteWise considera gli argomenti uguali se rientrano in una tolleranza relativa di $1E-9$. Si comporta in modo simile alla funzione isclose in Python. Questa funzione non restituisce un valore se x e y sono tipi incompatibili, come un numero e una stringa.
<code>lt(x, y)</code>	Restituisce 1 se x è inferiore a y, altrimenti 0 ($x < y$). Questa funzione non restituisce un valore se x e y sono tipi incompatibili, come un numero e una stringa.

Funzione	Descrizione
<code>lte(x, y)</code>	Restituisce 1 se x è minore o uguale a y, altrimenti 0 ($x \leq y$). AWS IoT SiteWise considera gli argomenti uguali se rientrano in una tolleranza relativa di $1E-9$. Si comporta in modo simile alla funzione isclose in Python. Questa funzione non restituisce un valore se x e y sono tipi incompatibili, come un numero e una stringa.
<code>isnan(x)</code>	Restituisce 1 se x è uguale a NaN, altrimenti 0. Questa funzione non restituisce un valore se x è una stringa.

Usa le funzioni condizionali nelle espressioni delle formule

Nelle [trasformazioni](#) e nelle [metriche](#), è possibile utilizzare la seguente funzione per verificare una condizione e restituire risultati diversi, indipendentemente dal fatto che la condizione restituisca vero o falso.

Funzione	Descrizione
<code>if(condition, result_if_true, result_if_false)</code>	Valuta <code>condition</code> e restituisce <code>result_if_true</code> se la condizione restituisce vero o <code>result_if_false</code> se la condizione restituisce <code>false</code>. <code>condition</code> deve essere un numero. Questa funzione considera <code>0</code> una stringa vuota come <code>false</code> e tutto il resto (incluso NaN) come <code>true</code>. I booleani vengono convertiti in <code>0</code> (<code>false</code>) e <code>1</code> (<code>true</code>).

Funzione	Descrizione
	È possibile restituire la costante none da questa funzione per scartare l'output per una particolare condizione. Ciò significa che puoi filtrare i punti dati che non soddisfano una condizione. Per ulteriori informazioni, consulta Filtra i punti dati. Example Esempi • <code>if(0, x, y)</code> restituisce la variabile <code>y</code>. • <code>if(5, x, y)</code> restituisce la variabile <code>x</code>. • <code>if(gt(temp, 300), x, y)</code> restituisce la variabile <code>x</code> se la variabile <code>temp</code> è maggiore di 300. • <code>if(gt(temp, 300), temp, none)</code> restituisce la variabile <code>temp</code> se è maggiore o uguale a 300, oppure <code>none</code> (nessun valore) se <code>temp</code> è minore di 300. Si consiglia di utilizzare UFCS per funzioni condizionali annidate in cui uno o più argomenti sono funzioni condizionali. È possibile utilizzare <code>if(condition, result_if_true)</code> per valutare una condizione e <code>elif(condition, result_if_true, result_if_false)</code> valutare condizioni aggiuntive. Ad esempio, puoi usare <code>if(condition1, result1_if_true).elif(condition2, result2_if_true, result2_if_false)</code> invece di <code>if(condition1, result1_if_true, if(condition2, result2_if_true, result2_if_false))</code>.

Funzione	Descrizione
	È inoltre possibile concatenare funzioni condizionali intermedie aggiuntive. Ad esempio, è possibile utilizzare, <code>if(condition1, result1_if_true).elif(condition2, result2_if_true).elif(condition3, result3_if_true, result3_if_false)</code> anziché annidare, più <code>if</code> istruzioni, ad esempio. <code>if(condition1, result1_if_true, if(condition2, result2_if_true, if(condition3, result3_if_true result3_if_false)))</code>  Important È necessario utilizzare <code>elif(condition, result_if_true, result_if_false)</code> con UFCS.

Usa le funzioni di stringa nelle espressioni delle formule

Nelle [trasformazioni](#) e nelle [metriche](#), è possibile utilizzare le seguenti funzioni per operare sulle stringhe. Per ulteriori informazioni, consulta [Usa le stringhe nelle formule](#).

 Important

Le espressioni di formule possono generare solo valori doppi o stringhe. Le espressioni annidate possono generare altri tipi di dati, ad esempio stringhe, ma la formula nel suo insieme deve restituire un numero o una stringa. È possibile utilizzare la [funzione jp](#) per convertire una stringa in un numero. Il valore booleano deve essere 1 (vero) o 0 (falso). Per ulteriori informazioni, consulta [Valori indefiniti, infiniti e di overflow](#).

Funzione	Descrizione
<code>len(s)</code>	Restituisce la lunghezza della stringa. <code>s</code>
<code>find(s, substring)</code>	Restituisce l'indice della stringa <code>substring</code> nella stringa <code>s</code> .
<code>contains(s, substring)</code>	Restituisce 1 se la stringa <code>s</code> contiene la stringa <code>substring</code> , altrimenti 0.
<code>upper(s)</code>	Restituisce la stringa <code>s</code> in formato maiuscolo.
<code>lower(s)</code>	Restituisce la stringa <code>s</code> in formato minuscolo.
<code>jp(s, json_path)</code>	Valuta la stringa <code>s</code> con l'JsonPath espressione <code>json_path</code> e restituisce il risultato. Utilizzate questa funzione per effettuare le seguenti operazioni: • Estrai un valore, un array o un oggetto da una struttura JSON serializzata. • Converti una stringa in un numero. Ad esempio, la formula viene <code>jp('111', '\$')</code> restituita 111 come numero. Per estrarre un valore di stringa da una struttura JSON e restituirlo come numero, è necessario utilizzare più funzioni annidate <code>jp</code>. La <code>jp</code> funzione esterna estrae la stringa dalla struttura JSON e la <code>jp</code> funzione interna converte la stringa in un numero. La stringa <code>json_path</code> deve contenere una stringa letterale. Ciò significa che non <code>json_path</code> può essere un'espressione che restituisce una stringa.

Funzione	Descrizione
	Example Esempi • <code>jp('{ "status": "active", "value": 15 }', '\$.value')</code> restituisce 15. • <code>jp('{ "measurement": { "reading": 25, "confidence": 0.95 } }', '\$.measurement.reading')</code> restituisce 25. • <code>jp('[2,8,23]', '\$[2]')</code> restituisce 23. • <code>jp('{ "values": [3,6,7] }', '\$.values[1]')</code> restituisce 6. • <code>jp('111', '\$')</code> restituisce 111. • <code>jp(jp('{ "measurement": { "reading": 25, "confidence": "0.95" } }', '\$.measurement.confidence'), '\$')</code> restituisce 0.95.
<pre>join(s0, s1, s2, s3, ...)</pre>	Restituisce una stringa concatenata con un delimitatore. Questa funzione utilizza la prima stringa di input come delimitatore e unisce le stringhe di input rimanenti. Si comporta in modo simile alla funzione join (CharSequence delimiter, CharSequence... elements) in Java. Example Esempi • <code>join("-", "aa", "bb", "cc")</code> restituisce aa-bb-cc

Funzione	Descrizione
<code>format(expression: "format")</code> o <code>format("format", expression)</code>	Restituisce una stringa nel formato specificato. Questa funzione restituisce <code>expression</code> un valore e quindi restituisce il valore nel formato specificato. Si comporta in modo simile alla funzione format (String format, Object... args) in Java. Per ulteriori informazioni sui formati supportati, consulta Conversioni in Class Formatter nella piattaforma Java, Specifiche dell'API Standard Edition 7. Example Esempi • <code>format(100+1: "d")</code> restituisce una stringa,. 101 • <code>format("The result is %d", 100+1)</code> restituisce una stringa,The result is 101.

Funzione	Descrizione
f'expression'	Restituisce una stringa concatenata. Con questa funzione formattata, è possibile utilizzare un'espressione semplice per concatenare e formattare stringhe. Queste funzioni possono contenere espressioni annidate. È possibile utilizzare {} (parentesi graffe arricciate) per interpolare le espressioni. Si comporta in modo simile alle stringhe letterali formattate in Python. Example Esempi • f'abc{1+2: "f"}d' restituisce abc3.000000d . Per valutare questa espressione di esempio, procedi come segue: 1. format(1+2: "f") restituisce un numero in virgola mobile, 3.000000. 2. join(' ', "abc", 1+2, 'd') restituisce una stringa, abc3.000000d . Puoi anche scrivere l'espressione nel modo seguente: join(' ', "abc", format(1+2: "f"), 'd') .

Usa le funzioni di aggregazione nelle espressioni delle formule

Solo nelle [metriche](#), puoi utilizzare le seguenti funzioni che aggregano i valori di input in ogni intervallo di tempo e calcolano un singolo valore di output. Le funzioni di aggregazione possono aggregare i dati dagli asset associati.

Gli argomenti delle funzioni di aggregazione possono essere [variabili](#), [numeri letterali](#), [funzioni temporali](#), [espressioni annidate o funzioni](#) di aggregazione. La formula `max(latest(x), latest(y), latest(z))` utilizza una funzione di aggregazione come argomento e restituisce il valore corrente massimo delle proprietà, e. x y z

È possibile utilizzare espressioni annidate nelle funzioni di aggregazione. Quando si utilizzano espressioni nidificate, si applicano le seguenti regole:

- Ogni argomento può avere una sola variabile.

Example

Ad esempio, $\text{avg}(x * (x - 1))$ e $\text{sum}(x/2) / \text{avg}(y^2)$ sono supportati.

Ad esempio, $\text{min}(x/y)$ non è supportato.

- Ogni argomento può avere espressioni annidate a più livelli.

Example

Ad esempio, $\text{sum}(\text{avg}(x^2)/2)$ è supportato.

- Argomenti diversi possono avere variabili diverse.

Example

Ad esempio, $\text{sum}(x/2, y^2)$ è supportato.

Note

- Se le espressioni contengono misurazioni, AWS IoT SiteWise utilizza gli ultimi valori dell'intervallo di tempo corrente per le misurazioni per calcolare gli aggregati.
- Se le espressioni contengono attributi, AWS IoT SiteWise utilizza i valori più recenti per gli attributi per calcolare gli aggregati.

Funzione	Descrizione
$\text{avg}(x_0, \dots, x_n)$	Restituisce la media dei valori delle variabili date nell'intervallo di tempo specificato. Questa funzione restituisce un punto dati solo se le variabili specificate hanno almeno un punto dati nell'intervallo di tempo corrente.

Funzione	Descrizione
$\text{sum}(x_0, \dots, x_n)$	Restituisce la somma dei valori delle variabili date nell'intervallo di tempo specificato. Questa funzione emette un punto dati solo se le variabili date hanno almeno un punto dati nell'intervallo di tempo corrente.
$\text{min}(x_0, \dots, x_n)$	Restituisce il valore minimo delle variabili date nell'intervallo di tempo specificato. Questa funzione emette un punto dati solo se le variabili date hanno almeno un punto dati nell'intervallo di tempo corrente.
$\text{max}(x_0, \dots, x_n)$	Restituisce il valore massimo delle variabili date nell'intervallo di tempo specificato. Questa funzione emette un punto dati solo se le variabili date hanno almeno un punto dati nell'intervallo di tempo corrente.
$\text{count}(x_0, \dots, x_n)$	Restituisce il numero totale di punti di dati per le variabili date nell'intervallo di tempo corrente. Per ulteriori informazioni su come contare il numero di punti dati che soddisfano una condizione, consulta Conta i punti dati che corrispondono a una condizione. Questa funzione calcola un punto dati per ogni intervallo di tempo.

Funzione	Descrizione
<code>stdev(x₀, ..., x_n)</code>	Restituisce la deviazione standard dei valori delle variabili specificate nell'intervallo di tempo corrente. Questa funzione emette un punto dati solo se le variabili date hanno almeno un punto dati nell'intervallo di tempo corrente.

Usa le funzioni temporali nelle espressioni delle formule

Utilizza le funzioni temporali per restituire valori in base ai timestamp dei punti dati.

Usa le funzioni temporali nelle metriche

Solo nelle [metriche](#), puoi utilizzare le seguenti funzioni che restituiscono valori in base ai timestamp dei punti dati.

Gli argomenti delle funzioni temporali devono essere proprietà del modello di asset locale o espressioni annidate. Ciò significa che non è possibile utilizzare le proprietà dei modelli di asset secondari nelle funzioni temporali.

È possibile utilizzare espressioni annidate nelle funzioni temporali. Quando si utilizzano espressioni nidificate, si applicano le seguenti regole:

- Ogni argomento può avere una sola variabile.

Ad esempio, `latest( t*9/5 + 32 )` è supportato.

- Gli argomenti non possono essere funzioni di aggregazione.

Ad esempio, `first( sum(x) )` non è supportato.

Funzione	Descrizione
<code>first(x)</code>	Restituisce il valore con il primo timestamp delle variabili date nell'intervallo di tempo specificato.

Funzione	Descrizione
<code>last(x)</code>	Restituisce il valore con l'ultimo timestamp delle variabili date nell'intervallo di tempo specificato.
<code>earliest(x)</code>	Restituisce l'ultimo valore della variabile specificata prima dell'inizio dell'intervallo di tempo corrente. Questa funzione calcola un punto dati per ogni intervallo di tempo, se la proprietà di input ha almeno un punto dati nella cronologia. Per informazioni dettagliate, vedi time-range-definti on.
<code>latest(x)</code>	Restituisce l'ultimo valore della variabile specificata con l'ultimo timestamp prima della fine dell'intervallo di tempo corrente. Questa funzione calcola un punto dati per ogni intervallo di tempo, se la proprietà di input ha almeno un punto dati nella cronologia. Per informazioni dettagliate, vedi time-range-definti on.

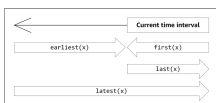
Funzione	Descrizione
<code>statetime(x)</code>	Restituisce la quantità di tempo in secondi in cui le variabili date sono risultate positive nell'intervallo di tempo specificato. È possibile utilizzare le funzioni di confronto per creare una proprietà di trasformazione da utilizzare per la <code>statetime</code> funzione. Ad esempio, se si dispone di una proprietà <code>Idle</code>, cioè 0 o 1, è possibile calcolare la durata dell'inattività nell'intervallo di tempo con questa espressione: <code>IdleTime = statetime (Idle)</code> . Per ulteriori informazioni, vedere lo scenario statetime di esempio. La funzione non supporta le proprietà dei parametri come variabili di input. Questa funzione calcola un punto dati per ogni intervallo di tempo, se la proprietà di input ha almeno un punto dati nella cronologia.

Funzione	Descrizione
<code>TimeWeightedAvg(x, [interpolation])</code>	Restituisce la media dei dati di input ponderata con gli intervalli di tempo tra i punti. Vedi Parametri delle funzioni ponderate nel tempo per i dettagli del calcolo e degli intervalli. L'argomento opzionale <code>interpolation</code> deve essere una costante di stringa: • <code>locf</code>— Questa è l'impostazione predefinita. Il calcolo utilizza l'algoritmo di calcolo Last Observed Carry Forward per gli intervalli tra i punti dati. In questo approccio, il punto dati viene calcolato come ultimo valore osservato fino al successivo timestamp del punto dati di input. Il valore dopo un buon punto dati viene estrapolato come valore fino al successivo timestamp del punto dati. • <code>linear</code>— Il calcolo utilizza l'algoritmo di calcolo dell'interpolazione lineare per gli intervalli tra i punti dati. Il valore tra due punti dati validi viene estrapolato come interpolazione lineare tra i valori di tali punti dati. Il valore tra punti dati buoni e cattivi o il valore dopo l'ultimo punto dati valido verrà estrapolato come punto dati valido.

Funzione	Descrizione
<code>TimeWeightedStDev(x, [algo])</code>	Restituisce la deviazione standard dei dati di input ponderata con gli intervalli di tempo tra i punti. Vedi Parametri delle funzioni ponderate nel tempo per i dettagli del calcolo e degli intervalli. Il calcolo utilizza l'algoritmo di calcolo Last Observed Carry Forward per gli intervalli tra i punti dati. In questo approccio, il punto dati viene calcolato come ultimo valore osservato fino al successivo timestamp del punto dati di input. Il peso viene calcolato come intervallo di tempo in secondi tra i punti dati o i confini della finestra. L'argomento opzionale <code>algo</code> deve essere una costante di stringa: • <code>f</code>— Questa è l'impostazione predefinita. Restituisce una varianza campionaria ponderata imparziale con pesi di frequenza, calcolata in <code>TimeWeight</code> secondi. Questo algoritmo viene generalmente assunto in base alla deviazione standard ed è noto come correzione di Bessel della deviazione standard per campioni ponderati. • <code>p</code>— Restituisce la varianza del campione ponderata e distorta, nota anche come varianza della popolazione. Le seguenti formule vengono utilizzate per il calcolo nei casi in cui: • S_p = deviazione standard della popolazione

Funzione	Descrizione
	• S_f = deviazione standard di frequenza • X_i = dati in entrata • ω_i = peso che equivale all'intervallo di tempo in secondi • μ^* = media ponderata dei dati in entrata Equazione per la deviazione standard della popolazione: $S_p^2 = \frac{\sum_{i=1}^N \omega_i (x_i - \mu^*)^2}{\sum_{i=1}^N \omega_i}$ Equazione per la deviazione standard di frequenza: $S_f^2 = \frac{\sum_{i=1}^N \omega_i (x_i - \mu^*)^2}{\sum_{i=1}^N \omega_i - 1}$

Il diagramma seguente mostra come AWS IoT SiteWise calcola le funzioni temporali, e `first` `last` `earliest` `latest`, rispetto all'intervallo di tempo corrente.



Note

- L'intervallo di tempo per `first(x)`, `last(x)` è (inizio finestra corrente, fine finestra corrente].
- L'intervallo di tempo per `latest(x)` è (inizio dell'ora, fine della finestra corrente].
- L'intervallo di tempo per `earliest(x)` è (inizio dell'ora, fine della finestra precedente].

Parametri delle funzioni ponderati nel tempo

Le funzioni ponderate in base al tempo calcolate per la finestra aggregata tengono conto di quanto segue:

- Punti dati all'interno della finestra
- Intervalli di tempo tra i punti dati
- Ultimo punto dati prima della finestra
- Primo punto dati dopo la finestra (per alcuni algoritmi)

Termini:

- Punto dati errato: qualsiasi punto dati con qualità o valore non numerico non buono. Questo non viene considerato nel calcolo dei risultati di una finestra.
- Intervallo errato: l'intervallo dopo un punto dati errato. Anche l'intervallo precedente al primo punto dati noto è considerato un intervallo errato.
- Buon punto dati: qualsiasi punto dati con buona qualità e valore numerico.

Note

- AWS IoT SiteWise consuma dati di GOOD qualità solo quando calcola trasformazioni e metriche. Ignora e fornisce dati UNCERTAIN. BAD
- L'intervallo precedente al primo punto dati noto è considerato un intervallo errato. Per ulteriori informazioni, consulta [the section called “Tutorial sulle espressioni di formule”](#).

L'intervallo dopo l'ultimo punto dati noto continua all'infinito e influisce su tutte le finestre successive. Quando arriva un nuovo punto dati, la funzione ricalcola l'intervallo.

Seguendo le regole precedenti, il risultato aggregato della finestra viene calcolato e limitato ai limiti delle finestre. Per impostazione predefinita, la funzione invia il risultato della finestra solo se l'intera finestra è un intervallo adeguato.

Se l'intervallo di validità della finestra è inferiore alla lunghezza della finestra, la funzione non invia la finestra.

Quando i punti dati che influiscono sul risultato della finestra cambiano, la funzione ricalcola la finestra, anche se i punti dati si trovano all'esterno della finestra.

Se la proprietà di input ha almeno un punto dati nella sua cronologia ed è stato avviato un calcolo, la funzione calcola le funzioni aggregate ponderate nel tempo per ogni intervallo di tempo.

Example Esempio di scenario statetime

Si consideri un esempio in cui si dispone di una risorsa con le seguenti proprietà:

- **Idle**— Una misurazione che è 0 o 1. Quando il valore è 1, la macchina è inattiva.
- **Idle Time**— Una metrica che utilizza la formula `statetime(Idle)` per calcolare la quantità di tempo in secondi in cui la macchina è inattiva, per intervallo di 1 minuto.

La proprietà **Idle** ha i seguenti punti dati.

Time stamp	2:00:00 PM	2:00:30 PM	2:01:15 PM	2:02:45 PM	2:04:00 PM
Idle	0	1	1	0	0

AWS IoT SiteWise calcola la **Idle Time** proprietà ogni minuto in base ai valori di **Idle**. Al termine di questo calcolo, la proprietà **Idle Time** ha i seguenti punti dati.

Time stamp	2:00:00 PM	2:01:00 PM	2:02:00 PM	2:03:00 PM	2:04:00 PM
Idle Time	N/D	30	60	45	0

AWS IoT SiteWise esegue i seguenti calcoli **Idle Time** alla fine di ogni minuto.

- Alle 2:00 PM (dall'1:59 PM alle 2:00 PM)
 - Non ci sono dati per **Idle** prima delle 2:00 PM, quindi nessun punto dati viene calcolato.
- Alle 2:01 PM (dalle 2:00 PM alle 2:01 PM)
 - Alle 2:00:00 PM, la macchina è attiva (**Idle** è 0).
 - Alle 2:00:30 PM, la macchina è inattiva (**Idle** è 1).
 - **Idle** non cambia di nuovo prima della fine dell'intervallo alle 2:01:00 PM, quindi **Idle Time** è 30 secondi.

- Alle 2:02 PM (da 2:01 PM alle 2:02 PM)
 - Alle 2:01:00 PM, la macchina è inattiva (con l'ultimo punto dati alle 2:00:30 PM).
 - Alle 2:01:15, la macchina è ancora inattiva.
 - Idle non cambia di nuovo prima della fine dell'intervallo alle 2:02:00 PM, quindi Idle Time è 60 secondi.
- Alle 2:03 PM (dalle 2:02 PM fino 2:03 PM)
 - Alle 2:02:00 PM, la macchina è inattiva (per l'ultimo punto dati alle 2:01:15 PM).
 - Alle 2:02:45 PM, la macchina è attiva.
 - Idle non cambia di nuovo prima della fine dell'intervallo alle 2:03:00 PM, quindi Idle Time è 45 secondi.
- Alle 2:04 PM (dalle 2:03 PM fino alle 2:04 PM)
 - Alle 2:03:00 PM, la macchina è attiva (per l'ultimo punto dati alle 2:02:45 PM).
 - Idle non cambia di nuovo prima della fine dell'intervallo alle 2:04:00 PM, quindi Idle Time è 0 secondi.

Example Esempio TimeWeightedAvg e TimeWeightedStDev scenario

Le tabelle seguenti forniscono input e output di esempio per queste metriche della finestra di un minuto: `Avg(x)`, `TimeWeightedAvg(x)`, `TimeWeightedAvg(x, "linear")`, `stDev(x)`, `timeWeightedStDev(x)`, `timeWeightedStDev(x, 'p')`

Esempio di input per una finestra aggregata di un minuto:

Note

Questi punti dati sono tutti di qualità. GOOD

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

Risultati aggregati in uscita:

 Note

Nessuno: risultato non prodotto per questa finestra.

Orario	Avg(x)	TimeWeigh tedAvg(x)	TimeWeigh tedAvg(X, "linear")	stDev(X)	timeWeigh tedStDev(x)	timeWeigh tedStDev(x, 'p')
3:00:00	4	Nessuno	Nessuno	0	Nessuno	Nessuno
3:01:00	2	4	3	0	0	0
3:02:00	14	9	13	6	5,4306 100415817 75	5,3851648 07134504
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	Nessuno	0	0	Nessuno	0	0

Usa le funzioni temporali nelle trasformazioni

Solo nelle [trasformazioni](#), è possibile utilizzare la `pretrigger()` funzione per recuperare il valore di GOOD qualità di una variabile prima dell'aggiornamento della proprietà che ha avviato il calcolo della trasformazione corrente.

Consideriamo un esempio in cui un produttore monitora AWS IoT SiteWise lo stato di una macchina. Il produttore utilizza le seguenti misurazioni e trasformazioni per rappresentare il processo:

- Una misurazione `current_state`, che può essere 0 o 1.
 - Se la macchina è in stato di pulizia, è `current_state` uguale a 1.
 - Se la macchina è in stato di produzione, è `current_state` uguale a 0.
- Una trasformazione, che `cleaning_state_duration` equivale a.


```
if(pretrigger(current_state) == 1, timestamp(current_state) -
timestamp(pretrigger(current_state)), none)
```

 Questa trasformazione restituisce per quanto tempo la macchina è rimasta nello stato di pulizia, in secondi, nel formato Unix epoch. Per ulteriori informazioni, vedere [Usa le funzioni condizionali nelle espressioni delle formule](#) e la funzione [timestamp\(\)](#).

Se la macchina rimane in stato di pulizia più a lungo del previsto, il produttore potrebbe esaminare la macchina.

È possibile utilizzare la `pretrigger()` funzione anche in trasformazioni multivariate. Ad esempio, avete due misurazioni denominate `x` and `y` una trasformazione uguale a $z = x + y + pretrigger(y)$. La tabella seguente mostra i valori per `x` e `z` dalle 9:00 alle 9:15. `y`

Note

- Questo esempio presuppone che i valori delle misurazioni arrivino in ordine cronologico. Ad esempio, il valore di `x` per 09:00 AM arriva prima del valore di `x` 09:05.
- Se i punti dati per le 9:05 arrivano prima dei punti dati per le 9:00, `z` non viene calcolato alle 9:05.
- Se il valore di `x` 9:05 AM arriva prima del valore di 09:00 AM e i valori di `y` arrivano in ordine cronologico, è uguale alle 9:05. $y = z - x = 22 - 20 = 2$

	09:00	09:05	09:10	09:15
x	10	20		30
y	1	2	3	
z = x + y + pretrigge r(y)	y non riceve alcun punto dati prima delle 09:00. Pertanto, z non viene calcolato alle 09:00.	23 = 20 + 2 + 1 pretrigge r(y) è uguale a 1.	25 = 20 + 3 + 2 x non riceve un nuovo punto dati. pretrigge r(y) è uguale a 2.	36 = 30 + 3 + 3 y non riceve un nuovo punto dati. Pertanto, è pretrigge r(y) uguale a 3 alle 09:15.

Utilizza le funzioni di data e ora nelle espressioni delle formule

Nelle [trasformazioni](#) e nelle [metriche](#), puoi utilizzare le funzioni di data e ora nei seguenti modi:

- Recupera il timestamp corrente di un punto dati in UTC o nel fuso orario locale.
- Costruisci timestamp con argomenti, come, e. year month day_of_month
- Estrai un periodo di tempo, ad esempio un anno o un mese, con l'argomento. unix_time

Funzione	Descrizione
now()	Restituisce la data e l'ora correnti, in secondi, nel formato Unix epoch.
timestamp()	• Nelle trasformazioni, la funzione restituisce il timestamp, in secondi, del messaggio di input nel formato Unix epoch. Solo nelle trasformazioni, puoi eseguire una delle seguenti operazioni: • Fornite una variabile come argomento della funzione. La timestamp (<i>variable-name</i>) funzione restituisce

Funzione	Descrizione
	il timestamp, in secondi, dell'ultimo valore di GOOD qualità per la variabile specificata nel formato Unix epoch. Ad esempio, se la risorsa ha una proprietà di trasformazione denominata <code>Temperature_F</code> che utilizza la $9/5 * \text{Temperature\_C}$ formula per convertire ogni punto di dati di temperatura da Celsius a Fahrenheit, potete utilizzare la <code>timestamp(Temperature_F)</code> funzione per ottenere il timestamp del valore di qualità più recente per la proprietà. <code>GOOD Temperature_F</code> • Utilizzate la <code>pretrigger()</code> funzione come argomento della funzione. La <code>timestamp(pretrigger(<i>variable-name</i>))</code> funzione restituisce il timestamp , in secondi, del valore di GOOD qualità per la variabile specificata prima dell'aggiornamento della proprietà che ha avviato il calcolo della trasformazione corrente nel formato Unix epoch. Per ulteriori informazioni, consulta Usa le funzioni temporali nelle trasformazioni. • Nelle metriche, la funzione restituisce il timestamp recuperato alla fine della finestra corrente, in secondi, nel formato Unix epoch.

Funzione	Descrizione
<code>mktime(time_zone, year, month, day_of_month, hour, minute, second)</code>	Restituisce il tempo di input in secondi, nel formato Unix epoch. Per l'utilizzo di questa funzione si applicano i seguenti requisiti: • L'argomento del fuso orario deve essere una stringa tra virgolette ('UTC '). Se non specificato, il fuso orario predefinito è UTC. L'argomento del fuso orario può essere il primo o l'ultimo argomento. • L'anno, il mese, il giorno del mese, l'ora, il minuto e il secondo argomento devono essere in ordine. • Gli argomenti anno, mese e data sono obbligatori. Per l'utilizzo di questa funzione si applicano i seguenti limiti: • <code>year</code>- I valori validi sono compresi tra 1970 e 2250. • <code>month</code>- I valori validi sono compresi tra 1 e 12. • <code>day-of-month</code> - I valori validi sono compresi tra 1 e 31. • <code>hour</code>- I valori validi sono compresi tra 0 e 23. • <code>minute</code>- I valori validi sono compresi tra 0 e 59. • <code>second</code>- I valori validi sono compresi tra 0 e 60. Può essere un numero in virgola mobile. Esempi:

Funzione	Descrizione
	• <code>mktime(2020, 2, 29)</code> • <code>mktime('UTC+3', 2021, 12, 31, 22)</code> • <code>mktime(2022, 10, 13, 2, 55, 13.68, 'PST')</code>

Funzione	Descrizione
<code>localtime(unix_time, time_zone)</code>	Restituisce l'anno, il giorno del mese, il giorno della settimana, il giorno dell'anno, l'ora, il minuto o il secondo nel fuso orario specificato dall'ora Unix. Per l'utilizzo di questa funzione si applicano i seguenti requisiti: • L'argomento del fuso orario deve essere una stringa tra virgolette ('UTC '). Se non specificato, il fuso orario predefinito è UTC. • L'argomento Unix time è l'ora in secondi, nel formato Unix epoch. L'intervallo valido è 1-31556889864403199. Può essere un numero a virgola mobile. Esempio di risposta: 2007-12-03T10:15:30+01:00[Europe/Paris] <code>localtime(unix_time, time_zone)</code> non è una funzione autonoma. Le <code>sec()</code> funzioni <code>year()</code><code>mon()</code><code>mday</code><code>wday()</code><code>yday()</code><code>hour()</code><code>minute()</code>, e prendono <code>localtime(unix_time, time_zone)</code> come argomento. Esempi: • <code>year(localtime('GMT', 1605898608.8113723))</code> • <code>now().localtime().year()</code> • <code>timestamp().localtime('PST').year()</code>

Funzione	Descrizione
	<code>localtime(1605289736, 'Europe/London').year()</code>
<code>year(localtime(unix_time, time_zone))</code>	Restituisce l'anno da <code>localtime(unix_time, time_zone)</code> .
<code>mon(localtime(unix_time, time_zone))</code>	Restituisce il mese da <code>localtime(unix_time, time_zone)</code> .
<code>mday(localtime(unix_time, time_zone))</code>	Restituisce il giorno del mese da <code>localtime(unix_time, time_zone)</code> .
<code>wday(localtime(unix_time, time_zone))</code>	Restituisce il giorno della settimana da <code>localtime(unix_time, time_zone)</code> .
<code>yday(localtime(unix_time, time_zone))</code>	Restituisce il giorno dell'anno da <code>localtime(unix_time, time_zone)</code> .
<code>hour(localtime(unix_time, time_zone))</code>	Restituisce l'ora da <code>localtime(unix_time, time_zone)</code> .
<code>minute(localtime(unix_time, time_zone))</code>	Restituisce il minuto da <code>localtime(unix_time, time_zone)</code> .
<code>sec(localtime(unix_time, time_zone))</code>	Restituisce il secondo da <code>localtime(unix_time, time_zone)</code> .

Formati di fuso orario supportati

È possibile specificare l'argomento del fuso orario nei seguenti modi:

- Scostamento del fuso orario: specifica 'Z' l'UTC o un offset ('+2'o). '-5'
- Offset IDs : combina un'abbreviazione del fuso orario e un offset. Ad esempio 'GMT+2' e 'UTC-01:00'. L'abbreviazione del fuso orario deve contenere solo tre lettere.
- Basato sulla regione IDs : ad esempio, 'Etc/GMT+12' e 'Pacific/Pago_Pago'.

Abbreviazioni dei fusi orari supportate

Le funzioni di data e ora supportano le seguenti abbreviazioni del fuso orario di tre lettere:

- EST - - 05:00
- SAB -- 10:00
- SAB - 07:00
- ACT - Australia/Darwin
- AET - Australia/Sydney
- AT - _Aires America/Argentina/Buenos
- ARTE - Africa/Cairo
- AST - America/Anchorage
- BET - America/San Paolo
- BST - Asia/Dacca
- CAT - Africa/Harare
- CET - Europa/Parigi
- CNT - America/St. Johns
- CST - America/Chicago
- CTT - Asia/Shanghai
- EAT - Africa/Addis_Ababa
- DIETA - America/Indiana/Indianapolis
- IST - Asia/Calcutta
- JST - Asia/Tokyo
- MIT - Pacifico/Apia
- NET - Asia/Erevan
- NST - Pacifico/Auckland
- PLT - Asia/Karachi
- PRT - America/Portorico
- PST - America/Los Angeles
- SST - Pacifico/Guadalcanal
- VST - Asia/Ho_Chi_Min

Supportato in base alla regione IDs

Le funzioni di data e ora supportano le seguenti funzioni basate sulla regione IDs, organizzate in base alla loro relazione con UTC+ 00:00:

- ETC/GMT+12 (UTC- 12:00)
- Pacifico/Pago_Pago (UTC- 11:00)
- Pacifico/Samoa (UTC- 11:00)
- Pacifico/Niue (UTC- 11:00)
- Stati Uniti/Samoa (UTC- 11:00)
- ETC/GMT+11 (UTC- 11:00)
- Pacifico/Midway (UTC- 11:00)
- Pacifico/Honolulu (UTC- 10:00)
- Pacifico/Rarotonga (UTC- 10:00)
- Pacifico/Tahiti (UTC- 10:00)
- Pacifico/Johnston (UTC- 10:00)
- Stati Uniti/Hawaii (UTC- 10:00)
- Sistema V/ HST1 0 (UTC- 10:00)
- ETC/GMT+10 (UTC- 10:00)
- Pacifico/Marchesi (UTC- 09:30)
- ETC/GMT+9 (UTC- 09:00)
- Pacifico/Gambier (UTC- 09:00)
- America/Atka (UTC- 09:00)
- Sistema V/ YST9 (UTC- 09:00)
- America/Adak (UTC- 09:00)
- Stati Uniti/Aleutine (UTC- 09:00)
- ETC/GMT+8 (UTC- 08:00)
- Stati Uniti/Alaska (UTC- 08:00)
- America/Juneau (UTC- 08:00)
- America/Metlakatla (UTC- 08:00)
- America/Yakutat (UTC- 08:00)

- Pacifico/Pitcairn (UTC- 08:00)
- America/Sitka (UTC- 08:00)
- America/Anchorage (UTC- 08:00)
- Sistema V/ PST8 (UTC- 08:00)
- America/Nome (UTC- 08:00)
- Sistema V/ YST9 YDT (UTC- 08:00)
- Canada/Yukon (UTC- 07:00)
- Stati Uniti/Pacifico - Nuovo (UTC- 07:00)
- ETC/GMT+7 (UTC- 07:00)
- Stati Uniti/Arizona (UTC- 07:00)
- America/Dawson_Creek (UTC- 07:00)
- Canada/Pacifico (UTC- 07:00)
- PST8PDT (UTC- 07:00)
- Sistema V/ MST7 (UTC- 07:00)
- America/Dawson (UTC- 07:00)
- Messico/ 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)
- Sistema V/ PST8 PDT (UTC- 07:00)
- America/Los Angeles (UTC- 07:00)
- Stati Uniti/Pacifico (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)
- Pacifico/Pasqua (UTC- 06:00)
- Messico/ BajaSur (UTC- 06:00)
- America/Regina (UTC- 06:00)
- America/Denver (UTC- 06:00)
- Pacifico/Galapagos (UTC- 06:00)
- America/Yellowknife (UTC- 06:00)
- America/Swift_Current (UTC- 06:00)
- America/Inuvik (UTC- 06:00)
- America/Mazatlán (UTC- 06:00)
- America/Boise (UTC- 06:00)
- America/Costa-Rica (UTC- 06:00)
- MST7MDT (UTC- 06:00)
- Sistema V/ CST6 (UTC- 06:00)
- America/Chihuahua (UTC- 06:00)
- America/Ojinaga (UTC- 06:00)
- Cile/ EasterIsland (UTC- 06:00)
- Stati Uniti/Mountain (UTC- 06:00)
- America/Edmonton (UTC- 06:00)
- Canada/Montagna (UTC- 06:00)
- America/Cambridge_Bay (UTC- 06:00)
- Navajo (UTC- 06:00)
- Sistema V/ MST7 MDT (UTC- 06:00)
- Canada/Saskatchewan (UTC- 06:00)
- America/Shiprock (UTC- 06:00)

- America/Panama (UTC- 05:00)
- America/Chicago (UTC- 05:00)
- America/Eirunepé (UTC- 05:00)
- ETC/GMT+5 (UTC- 05:00)
- Messico/Generale (UTC- 05:00)
- America/Porto_Acre (UTC- 05:00)
- America/Guayaquil (UTC- 05:00)
- America/Rankin_Inlet (UTC- 05:00)
- Stati Uniti/Central (UTC- 05:00)
- America/Rainy_River (UTC- 05:00)
- America/Indiana/Knox(UTC- 05:00)
- America/North_Dakota/Beulah(UTC- 05:00)
- America/Monterrey (UTC- 05:00)
- America/Giamaica (UTC- 05:00)
- America/Atikokan (UTC- 05:00)
- America/Coral_Harbour (UTC- 05:00)
- America/North_Dakota/Center(UTC- 05:00)
- America/Cayman (UTC- 05:00)
- America/Indiana/Tell_Città (UTC- 05:00)
- America/Città del Messico (UTC- 05:00)
- America/Matamoros (UTC- 05:00)
- CST6CDT (UTC- 05:00)
- America/Knox_in (UTC- 05:00)
- America/Bogotà (UTC- 05:00)
- America/Menominee (UTC- 05:00)
- America/Resolute (UTC- 05:00)
- Sistema V/ EST5 (UTC- 05:00)
- Canada/Central (UTC- 05:00)
- Brasile/Acre (UTC- 05:00)

- America/Cancún (UTC- 05:00)
- America/Lima (UTC- 05:00)
- America/Bahia Banderas (UTC- 05:00)
- Stati Uniti/Indiana-Starke (UTC- 05:00)
- America/Rio_Branco (UTC- 05:00)
- Sistema V/ CST6 CDT (UTC- 05:00)
- Giamaica (UTC- 05:00)
- America/Mérida (UTC- 05:00)
- America/North_Dakota/New_Salem (UTC- 05:00)
- America/Winnipeg (UTC- 05:00)
- America/Cuiabá (UTC- 04:00)
- America/Marigot (UTC- 04:00)
- America/Indiana/Petersburg(UTC- 04:00)
- Cile/Continentale (UTC- 04:00)
- America/Grand_Turk (UTC- 04:00)
- Cuba (UTC- 04:00)
- ETC/GMT+4 (UTC- 04:00)
- America/Manaus (UTC- 04:00)
- America/Fort_Wayne (UTC- 04:00)
- America/Saint Thomas (UTC- 04:00)
- America/Anguilla (UTC- 04:00)
- America/L'Avana (UTC- 04:00)
- Stati Uniti/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/Martinica (UTC- 04:00)
- America/Porto_Rico (UTC- 04:00)

- America/Port_of_Spain (UTC- 04:00)
- Sistema V/ 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/Saint Vincent (UTC- 04:00)
- America/Kentucky/Louisville(UTC- 04:00)
- America/Dominica (UTC- 04:00)
- America/Asunción (UTC- 04:00)
- EST5EDT (UTC- 04:00)
- America/Nassau (UTC- 04:00)
- America/Kentucky/Monticello(UTC- 04:00)
- Brasile/Ovest (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_Bay (UTC- 04:00)
- America/Indiana/Marengo(UTC- 04:00)
- America/Blanc-Sablon (UTC- 04:00)
- America/Santo Domingo (UTC- 04:00)
- Stati Uniti/Orientali (UTC- 04:00)
- Canada/Est (UTC- 04:00)
- America/ Port-au-Prince (UTC- 04:00)
- America/Saint Barthelemy (UTC- 04:00)
- America/Nipigon (UTC- 04:00)

- Stati Uniti/Indiana orientale (UTC- 04:00)
- America/Saint Lucia (UTC- 04:00)
- America/Montserrat (UTC- 04:00)
- America/Lower_Princes (UTC- 04:00)
- America/Detroit (UTC- 04:00)
- America/Tortola (UTC- 04:00)
- America/Porto_Velho (UTC- 04:00)
- America/Campo_Grande (UTC- 04:00)
- America/Isole Vergini (UTC- 04:00)
- America/Pangnirtung (UTC- 04:00)
- America/Montréal (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/Saint Kitts (UTC- 04:00)
- America/Caracas (UTC- 04:00)
- America/Guadalupa (UTC- 04:00)
- America/Toronto (UTC- 04:00)
- Sistema V/ EST5 EDT (UTC- 04:00)
- America/Argentina/Catamarca(UTC- 03:00)
- Canada/Atlantico (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)
- Brasile/Est (UTC- 03:00)
- America/Argentina/Mendoza(UTC- 03:00)
- America/Argentina/Rio_Gallegos (UTC- 03:00)

- America/Catamarca (UTC- 03:00)
- America/Córdoba (UTC- 03:00)
- America/San Paolo (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)
- Atlantico/Bermuda (UTC- 03:00)
- America/Maceió (UTC- 03:00)
- Atlantico/Stanley (UTC- 03:00)
- America/Halifax (UTC- 03:00)
- Antartide/Rothera (UTC- 03:00)
- America/Argentina/San_Luis (UTC- 03:00)
- America/Argentina/Ushuaia(UTC- 03:00)
- Antartide/Palmer (UTC- 03:00)
- America/Punta_Arenas (UTC- 03:00)
- America/Glace_Bay (UTC- 03:00)
- America/Fortaleza (UTC- 03:00)
- America/Thule (UTC- 03:00)
- America/Argentina/La_Rioja (UTC- 03:00)
- America/Belém (UTC- 03:00)
- America/Jujuy (UTC- 03:00)
- America/Bahia (UTC- 03:00)
- America/Goose_Bay (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)
- Sistema AST4 V/ADT (UTC- 03:00)
- America/Argentina/Buenos_Aires (UTC- 03:00)
- America/Saint Johns (UTC- 02:30)
- Canada/Terranova (UTC- 02:30)
- America/Miquelon (UTC- 02:00)
- ETC/GMT+2 (UTC- 02:00)
- America/Godthab (UTC- 02:00)
- America/Norfolk (UTC- 02:00)
- Brasile/ DeNoronha (UTC- 02:00)
- Atlantico/Georgia del Sud (UTC- 02:00)
- ETC/GMT+1 (UTC- 01:00)
- Atlantico/Capo Verde (UTC- 01:00)
- Pacifico/Kiritimati (UTC+ 14:00)
- ETC/GMT-14 (UTC+ 14:00)
- Pacifico/Fakaofu (UTC+ 13:00)
- Pacifico/Enderbury (UTC+ 13:00)
- Pacifico/Apia (UTC+ 13:00)
- Pacifico/Tongatapu (UTC+ 13:00)
- Etc/GMT-13 (UTC+ 13:00)
- NZ-CHAT (UTC+ 12:45)
- Pacifico/Chatham (UTC+ 12:45)
- Pacifico/Kwajalein (UTC+ 12:00)
- Antartide/ (UTC+ 12:00) McMurdo
- Pacifico/Wallis (UTC+ 12:00)
- Pacifico/Fiji (UTC+ 12:00)
- Pacifico/Funafuti (UTC+ 12:00)
- Pacifico/Nauru (UTC+ 12:00)
- Kwajalein (UTC+ 12:00)

- NZ (UTC+ 12:00)
- Pacifico/Wake (UTC+ 12:00)
- Antartide/South_Pole (UTC+ 12:00)
- Pacifico/Tarawa (UTC+ 12:00)
- Pacifico/Auckland (UTC+ 12:00)
- Asia/Kamchatka (UTC+ 12:00)
- Etc/GMT-12 (UTC+ 12:00)
- Asia/Anadyr (UTC+ 12:00)
- Pacifico/Majuro (UTC+ 12:00)
- Pacifico/Ponape (UTC+ 11:00)
- Pacifico/Bougainville (UTC+ 11:00)
- Antartide/Macquarie (UTC+ 11:00)
- Pacifico/Pohnpei (UTC+ 11:00)
- Pacifico/Efate (UTC+ 11:00)
- Pacifico/Norfolk (UTC+ 11:00)
- Asia/Magadan (UTC+ 11:00)
- Pacifico/Kosrae (UTC+ 11:00)
- Asia/Sakhalin (UTC+ 11:00)
- Pacifico/Noumea (UTC+ 11:00)
- ETC/GMT-11 (UTC+ 11:00)
- Asia/Srednekolymk (UTC+ 11:00)
- Pacifico/Guadalcanal (UTC+ 11:00)
- Australia/Lord Howe (UTC+ 10:30)
- Australia/LHI (UTC+ 10:30)
- Australia/Hobart (UTC+ 10:00)
- Pacifico/Yap (UTC+ 10:00)
- Australia/Tasmania (UTC+ 10:00)
- Pacifico/Port_Moresby (UTC+ 10:00)
- Australia/ACT (UTC+ 10:00)
- Australia/Victoria (UTC+ 10:00)

- Pacifico/Chuuk (UTC+ 10:00)
- Australia/Queensland (UTC+ 10:00)
- Australia/Canberra (UTC+ 10:00)
- Australia/Currie (UTC+ 10:00)
- Pacifico/Guam (UTC+ 10:00)
- Pacifico/Truk (UTC+ 10:00)
- Australia/Nuovo Galles del Sud (UTC+ 10:00)
- Asia/Vladivostok (UTC+ 10:00)
- Pacifico/Saipan (UTC+ 10:00)
- Antartide/Dumont (UTC+ 10:00)DUrville)
- 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/Nord (UTC+ 09:30)
- Australia/Yancowinna (UTC+ 09:30)
- Australia/Adelaide (UTC+ 09:30)
- Australia/Broken_Hill (UTC+ 09:30)
- Australia/Sud (UTC+ 09:30)
- Australia/Darwin (UTC+ 09:30)
- ETC/GMT-9 (UTC+ 09:00)
- Pacifico/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)
- MERCOLEDÌ (UTC+ 09:00)

- Asia/Seul (UTC+ 09:00)
- Asia/Khandyga (UTC+ 09:00)
- Giappone (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/Macao (UTC+ 08:00)
- Asia/Choibalsan (UTC+ 08:00)
- Asia/Shanghai (UTC+ 08:00)
- Antartide/Casey (UTC+ 08:00)
- Asia/Ulan_Bator (UTC+ 08:00)
- Asia/Chongqing (UTC+ 08:00)
- Asia/Ulan Bator (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)
- Singapore (UTC+ 08:00)
- Asia/Brunei (UTC+ 08:00)
- Australia/Ovest (UTC+ 08:00)
- Asia/Hong Kong (UTC+ 08:00)
- Asia/Makassar (UTC+ 08:00)
- Hong Kong (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)
- Antartide/Davis (UTC+ 07:00)
- Asia/Tomsk (UTC+ 07:00)
- Asia/Giacarta (UTC+ 07:00)
- Asia/Barnaul (UTC+ 07:00)
- Indiano/Natale (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/Novokuzneck (UTC+ 07:00)
- Asia/Krasnoyarsk (UTC+ 07:00)
- Asia/Saigon (UTC+ 07:00)
- Asia/Yangon (UTC+ 06:30)
- Asia/Yangon (UTC+ 06:30)
- Indiana/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/Dacca (UTC+ 06:00)
- Indiana/Chagos (UTC+ 06:00)
- Asia/Qyzylorda (UTC+ 06:00)
- Asia/Bishkek (UTC+ 06:00)
- Antartide/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/Kathmandu (UTC+ 05:45)
- Asia/Calcutta (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/Samarcanda (UTC+ 05:00)
- Asia/Karachi (UTC+ 05:00)
- Asia/Ekaterinburg (UTC+ 05:00)
- Asia/Dushanbe (UTC+ 05:00)
- Indiana/Maldives (UTC+ 05:00)
- Asia/Orale (UTC+ 05:00)
- Asia/Taskent (UTC+ 05:00)
- Antartide/Mawson (UTC+ 05:00)
- Asia/Aktobe (UTC+ 05:00)
- Asia/Ashkhabad (UTC+ 05:00)
- Asia/Ashgabat (UTC+ 05:00)
- Asia/Atyrau (UTC+ 05:00)
- Indiana/Kerguelen (UTC+ 05:00)
- Iran (UTC+ 04:30)
- Asia/Teheran (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)

- Indiana/Riunione (UTC+ 04:00)
- Europa/Saratov (UTC+ 04:00)
- Europa/Samara (UTC+ 04:00)
- Indiana/Mahé (UTC+ 04:00)
- Asia/Baku (UTC+ 04:00)
- Asia/Muscat (UTC+ 04:00)
- Europa/Volgograd (UTC+ 04:00)
- Europa/Astrakhan (UTC+ 04:00)
- Asia/Tbilisi (UTC+ 04:00)
- Europa/Ulyanovsk (UTC+ 04:00)
- Asia/Aden (UTC+ 03:00)
- Africa/Nairobi (UTC+ 03:00)
- Europa/Istanbul (UTC+ 03:00)
- ETC/GMT-3 (UTC+ 03:00)
- Europa/Zaporozhye (UTC+ 03:00)
- Israele (UTC+ 03:00)
- Indiana/Comore (UTC+ 03:00)
- Antartide/Syowa (UTC+ 03:00)
- Africa/Mogadiscio (UTC+ 03:00)
- Europa/Bucarest (UTC+ 03:00)
- Africa/Asmeria (UTC+ 03:00)
- Europa/Mariehamn (UTC+ 03:00)
- Asia/Istanbul (UTC+ 03:00)
- Europa/Tiraspol (UTC+ 03:00)
- Europa/Mosca (UTC+ 03:00)
- Europa/Chisinau (UTC+ 03:00)
- Europa/Helsinki (UTC+ 03:00)
- Asia/Beirut (UTC+ 03:00)
- Asia/Tel Aviv (UTC+ 03:00)
- Africa/Gibuti (UTC+ 03:00)

- Europa/Simferopol (UTC+ 03:00)
- Europa/Sofia (UTC+ 03:00)
- Asia/Gaza (UTC+ 03:00)
- Africa/Asmara (UTC+ 03:00)
- Europa/Riga (UTC+ 03:00)
- Asia/Bagdad (UTC+ 03:00)
- Asia/Damasco (UTC+ 03:00)
- Africa/Dar_es_Salaam (UTC+ 03:00)
- Africa/Addis_Abeba (UTC+ 03:00)
- Europa/Uzhgorod (UTC+ 03:00)
- Asia/Gerusalemme (UTC+ 03:00)
- Asia/Riyad (UTC+ 03:00)
- Asia/Kuwait (UTC+ 03:00)
- Europa/Kirov (UTC+ 03:00)
- Africa/Kampala (UTC+ 03:00)
- Europa/Minsk (UTC+ 03:00)
- Asia/Qatar (UTC+ 03:00)
- Europa/Kiev (UTC+ 03:00)
- Asia/Bahrein (UTC+ 03:00)
- Europa/Vilnius (UTC+ 03:00)
- Indiana/Antananarivo (UTC+ 03:00)
- Indiana/Mayotte (UTC+ 03:00)
- Europa/Tallinn (UTC+ 03:00)
- Turchia (UTC+ 03:00)
- Africa/Juba (UTC+ 03:00)
- Asia/Nicosia (UTC+ 03:00)
- Asia/Famagosta (UTC+ 03:00)
- S-DOM (UTC+ 03:00)
- EET (UTC+ 03:00)
- Asia/Hebron (UTC+ 03:00)

- Asia/Amman (UTC+ 03:00)
- Europa/Nicosia (UTC+ 03:00)
- Europa/Atene (UTC+ 03:00)
- Africa/Cairo (UTC+ 02:00)
- Africa/Mbabane (UTC+ 02:00)
- Europa/Bruxelles (UTC+ 02:00)
- Europa/Varsavia (UTC+ 02:00)
- CET (UTC+ 02:00)
- Europa/Lussemburgo (UTC+ 02:00)
- ETC/GMT-2 (UTC+ 02:00)
- Libia (UTC+ 02:00)
- Africa/Kigali (UTC+ 02:00)
- Africa/Tripoli (UTC+ 02:00)
- Europa/Kaliningrad (UTC+ 02:00)
- Africa/Windhoek (UTC+ 02:00)
- Europa/Malta (UTC+ 02:00)
- Europa/Busingen (UTC+ 02:00)
-
- Europa/Skopje (UTC+ 02:00)
- Europa/Sarajevo (UTC+ 02:00)
- Europa/Roma (UTC+ 02:00)
- Europa/Zurigo (UTC+ 02:00)
- Europa/Gibilterra (UTC+ 02:00)
- Africa/Lubumbashi (UTC+ 02:00)
- Europa/Vaduz (UTC+ 02:00)
- Europa/Lubiana (UTC+ 02:00)
- Europa/Berlino (UTC+ 02:00)
- Europa/Stoccolma (UTC+ 02:00)
- Europa/Budapest (UTC+ 02:00)
- Europa/Zagabria (UTC+ 02:00)

- Europa/Parigi (UTC+ 02:00)
- Africa/Ceuta (UTC+ 02:00)
- Europa/Praga (UTC+ 02:00)
- Antartide/Troll (UTC+ 02:00)
- Africa/Gaborone (UTC+ 02:00)
- Europa/Copenaghen (UTC+ 02:00)
- Europa/Vienna (UTC+ 02:00)
- Europa/Tirana (UTC+ 02:00)
- INCONTRATO (UTC+ 02:00)
- Europa/Amsterdam (UTC+ 02:00)
- Africa/Maputo (UTC+ 02:00)
- Europa/San_Marino (UTC+ 02:00)
- Polonia (UTC+ 02:00)
- Europa/Andorra (UTC+ 02:00)
- Europa/Oslo (UTC+ 02:00)
- Europa/Podgorica (UTC+ 02:00)
- Africa/Bujumbura (UTC+ 02:00)
- Atlantico/Jan_Mayen (UTC+ 02:00)
- Africa/Maseru (UTC+ 02:00)
- Europa/Madrid (UTC+ 02:00)
- Africa/Blantyre (UTC+ 02:00)
- Africa/Lusaka (UTC+ 02:00)
- Africa/Harare (UTC+ 02:00)
- Africa/Khartum (UTC+ 02:00)
- Africa/Johannesburg (UTC+ 02:00)
- Europa/Belgrado (UTC+ 02:00)
- Europa/Bratislava (UTC+ 02:00)
- Artico/Longyearbyen (UTC+ 02:00)
- Egitto (UTC+ 02:00)
- Europa/Vaticano (UTC+ 02:00)

- Europa/Monaco (UTC+ 02:00)
- Europa/Londra (UTC+ 01:00)
- Etc/GMT-1 (UTC+ 01:00)
- Europa/Jersey (UTC+ 01:00)
- Europa/Guernsey (UTC+ 01:00)
- Europa/Isle_of_Man (UTC+ 01:00)
- Africa/Tunisi (UTC+ 01:00)
- Africa/Malabo (UTC+ 01:00)
- GB-Irlanda (UTC+ 01:00)
- Africa/Lagos (UTC+ 01:00)
- Africa/Algeri (UTC+ 01:00)
- IT (UTC+ 01:00)
- Portogallo (UTC+ 01:00)
- Africa/Sao_Tomé (UTC+ 01:00)
- Africa/N' Djamena (UTC+ 01:00)
- Atlantico/Isole Faroe (UTC+ 01:00)
- Irlanda (UTC+ 01:00)
- Atlantico/Faroe (UTC+ 01:00)
- Europa/Dublino (UTC+ 01:00)
- Africa/Libreville (UTC+ 01:00)
- Africa/EI_Aaiun (UTC+ 01:00)
- Africa/EI_Aaiun (UTC+ 01:00)
- Africa/Douala (UTC+ 01:00)
- Africa/Brazzaville (UTC+ 01:00)
- Africa/Porto-Novo (UTC+ 01:00)
- Atlantico/Madeira (UTC+ 01:00)
- Europa/Lisbona (UTC+ 01:00)
- Atlantico/Canarie (UTC+ 01:00)
- Africa/Casablanca (UTC+ 01:00)

- Europa/Belfast (UTC+ 01:00)
- Africa/Luanda (UTC+ 01:00)
- Africa/Kinshasa (UTC+ 01:00)
- Africa/Bangui (UTC+ 01:00)
- UMIDO (UTC+ 01:00)
- Africa/Niamey (UTC+ 01:00)
- GMT (UTC+ 00:00)
- etc/GMT-0 (UTC+ 00:00)
- Atlantic/Sant' Elena (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)
- Universale (UTC+ 00:00)
- Africa/Nouakchott (UTC+ 00:00)
- UTC (UTC+ 00:00)
- Etc/Universal (UTC+ 00:00)
- Atlantico/Azzorre (UTC+ 00:00)
- Africa/Abidjan (UTC+ 00:00)
- Africa/Accra (UTC+ 00:00)
- ETC/UTC (UTC+ 00:00)
- GMT0 (UTC+ 00:00)
- Zulu (UTC+ 00:00) Zulu (UTC+ 00:00)
- Africa/Ouagadougou (UTC+ 00:00)
- Atlantico/Reykjavík (UTC+ 00:00)
- Etc/Zulu (UTC+ 00:00)
- Islanda (UTC+ 00:00)

- Africa/Lomé (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/Timbuctù (UTC+ 00:00)
- UTC (UTC+ 00:00)
- Africa/Monrovia (UTC+ 00:00)
- ETC/UTC (UTC+ 00:00)

Tutorial sulle espressioni di formule

Puoi seguire questi tutorial per utilizzare le espressioni delle formule in. AWS IoT SiteWise

Argomenti

- [Usa le stringhe nelle formule](#)
- [Filtra i punti dati](#)
- [Conta i punti dati che corrispondono a una condizione](#)
- [Dati tardivi nelle formule](#)
- [Qualità dei dati nelle formule](#)
- [Valori indefiniti, infiniti e di overflow](#)

Usa le stringhe nelle formule

È possibile utilizzare le stringhe nelle espressioni delle formule. È inoltre possibile inserire stringhe da variabili che fanno riferimento alle proprietà degli attributi e delle misurazioni.

Important

Le espressioni delle formule possono generare solo valori doppi o stringhe. Le espressioni annidate possono generare altri tipi di dati, ad esempio stringhe, ma la formula nel suo

insieme deve restituire un numero o una stringa. È possibile utilizzare la [funzione jp](#) per convertire una stringa in un numero. Il valore booleano deve essere 1 (vero) o 0 (falso). Per ulteriori informazioni, consulta [Valori indefiniti, infiniti e di overflow](#).

AWS IoT SiteWise fornisce le seguenti funzionalità di espressione delle formule che è possibile utilizzare per operare sulle stringhe:

- [Stringhe letterali](#)
- L'[operatore di indice](#) () s[index]
- L'[operatore slice](#) () s[start:end:step]
- [Funzioni di confronto, che è possibile utilizzare per confrontare le stringhe in base all'ordine lessicografico](#)
- [Funzioni di stringa](#), che includono la jp funzione in grado di analizzare oggetti JSON serializzati e convertire stringhe in numeri

Filtra i punti dati

Puoi utilizzare la [funzione if](#) per filtrare i punti dati che non soddisfano una condizione. La if funzione valuta una condizione e restituisce valori true e false risultati diversi. È possibile utilizzare la [costante none](#) come output per un caso di una if funzione per scartare il punto dati relativo a quel caso.

Per filtrare i punti dati che corrispondono a una condizione

- Crea una trasformazione che utilizzi la if funzione per definire una condizione che verifichi se una condizione è soddisfatta e la restituisca none come result_if_false valore result_if_true or.

Example Esempio: filtra i punti dati in cui l'acqua non bolle

Prendiamo in considerazione uno scenario in cui si esegue temp_c una misurazione che fornisce la temperatura (in gradi Celsius) dell'acqua in una macchina. Puoi definire la seguente trasformazione per filtrare i punti dati in cui l'acqua non bolle:

- Trasforma: boiling_temps = if(gte(temp_c, 100), temp_c, none) — Restituisce la temperatura se è maggiore o uguale a 100 gradi Celsius, altrimenti non restituisce alcun punto dati.

Conta i punti dati che corrispondono a una condizione

È possibile utilizzare [le funzioni di confronto](#) e [sum \(\)](#) per contare il numero di punti dati per i quali una condizione è vera.

Per contare i punti dati che corrispondono a una condizione

1. Crea una trasformazione che utilizza una funzione di confronto per definire una condizione di filtro su un'altra proprietà.
2. Creare un parametro che sommi i punti dati in cui tale condizione è soddisfatta.

Example Esempio: contare il numero di punti dati in cui l'acqua è in ebollizione.

Si consideri uno scenario in cui si esegue una misurazione che fornisce la temperatura (in gradi Celsius) dell'acqua in una macchina. `temp_c` È possibile definire le seguenti proprietà di trasformazione e parametro per contare il numero di punti dati in cui l'acqua è in ebollizione:

- Trasforma: `is_boiling = gte(temp_c, 100)` — Restituisce 1 se la temperatura è maggiore o uguale a 100 gradi Celsius, altrimenti restituisce. 0
- Metrica: `boiling_count = sum(is_boiling)` — Restituisce il numero di punti dati in cui l'acqua bolle.

Dati tardivi nelle formule

AWS IoT SiteWise supporta l'inserimento tardivo di dati risalenti fino a 7 giorni fa. Quando AWS IoT SiteWise riceve dati in ritardo, ricalcola i valori esistenti per qualsiasi metrica che inserisce i dati in ritardo in una finestra precedente. Questi ricalcoli comportano costi di elaborazione dei dati.

Note

Quando AWS IoT SiteWise calcola proprietà che immettono dati tardivi, utilizza l'espressione della formula corrente di ogni proprietà.

Dopo aver AWS IoT SiteWise ricalcolato una finestra precedente per una metrica, sostituisce il valore precedente per quella finestra. Se hai abilitato le notifiche per quella metrica, emette AWS IoT SiteWise anche una notifica del valore della proprietà. Ciò significa che è possibile ricevere una nuova notifica di aggiornamento del valore della proprietà per la stessa proprietà e timestamp per cui

è stata precedentemente ricevuta una notifica. Se le applicazioni o i data lake utilizzano notifiche sui valori delle proprietà, devi aggiornare il valore precedente con il nuovo valore in modo che i dati siano accurati.

Qualità dei dati nelle formule

In AWS IoT SiteWise, ogni punto dati ha un codice di qualità, che può essere uno dei seguenti:

- GOOD— I dati non sono interessati da alcun problema.
- BAD— I dati sono interessati da un problema come il guasto del sensore.
- UNCERTAIN— I dati sono influenzati da un problema come l'imprecisione del sensore.

AWS IoT SiteWise utilizza solo dati GOOD di qualità quando calcola trasformazioni e metriche. AWS IoT SiteWise restituisce solo dati di GOOD qualità per calcoli di successo. Se un calcolo non ha esito positivo, AWS IoT SiteWise non emette un punto dati per quel calcolo. Ciò può verificarsi se un calcolo genera un valore indefinito, infinito o di overflow.

Per ulteriori informazioni su come eseguire query sui dati e filtrare in base alla qualità dei dati, vedere [Interroga i dati da AWS IoT SiteWise](#).

Valori indefiniti, infiniti e di overflow

Alcune espressioni di formule (ad esempio $x / 0$, $\sqrt{-1}$, $\log(0)$) calcolano valori non definiti in un sistema numerico reale, infiniti o al di fuori dell'intervallo supportato da. AWS IoT SiteWise. Quando l'espressione di una proprietà di asset calcola un valore indefinito, infinito o di overflow, AWS IoT SiteWise non genera un punto dati per quel calcolo.

AWS IoT SiteWise inoltre non emette un punto dati se calcola un valore non numerico come risultato di un'espressione di formula. Ciò significa che se definisci una formula che calcola una stringa, una matrice o la [costante none](#), AWS IoT SiteWise non genera un punto dati per quel calcolo.

Example Esempi

Ciascuna delle seguenti espressioni di formula restituisce un valore che non AWS IoT SiteWise può essere rappresentato come numero. AWS IoT SiteWise non genera un punto dati quando calcola queste espressioni di formule.

- $x / 0$ non è definito.
- $\log(0)$ non è definito.
- $\sqrt{-1}$ non è definito in un sistema numerico reale.

- `"hello" + " world"` è una stringa.
- `jp({'values':[3,6,7]}, '$.values')` è un array.
- `if(gte(temp, 300), temp, none)` è `none` quando `temp` è inferiore a `300`.

Crea modelli compositi personalizzati (componenti)

I modelli compositi personalizzati, o componenti, se utilizzi la console, forniscono un altro livello di organizzazione per i modelli di asset e i modelli di componenti. Puoi usarli per strutturare i tuoi modelli raggruppando le proprietà o facendo riferimento ad altri modelli. Per ulteriori informazioni sull'utilizzo di modelli compositi personalizzati, consulta [Modelli compositi personalizzati \(componenti\)](#)

È possibile creare un modello composito personalizzato all'interno di un modello di asset o di componenti esistente. Esistono due tipi di modelli compositi personalizzati. Per raggruppare le proprietà correlate all'interno di un modello, è possibile creare un modello composito personalizzato in linea. Per fare riferimento a un modello di componente all'interno del vostro modello di asset o modello di componente, potete creare un modello composito component-model-based personalizzato.

Le sezioni seguenti descrivono come utilizzare l' AWS IoT SiteWise API per creare modelli compositi personalizzati.

Argomenti

- [Crea un componente in linea \(console\)](#)
- [Create un modello composito personalizzato in linea \(AWS CLI\)](#)
- [Crea un component-model-based componente \(console\)](#)
- [Crea un modello composito component-model-based personalizzato \(AWS CLI\)](#)

Crea un componente in linea (console)

È possibile utilizzare la AWS IoT SiteWise console per creare un componente in linea che definisce le proprie proprietà.

Note

Poiché si tratta di un componente in linea, queste proprietà si applicano solo al modello di asset corrente e non sono condivise altrove.

Se avete bisogno di produrre un modello riutilizzabile (ad esempio, per condividerlo tra più modelli di asset o per includere più istanze all'interno di un modello di asset), dovrete

invece creare un componente basato su un modello di componente. Per i dettagli, consulta la sezione seguente.

Per creare un componente (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione selezionare Models (Modelli).
3. Scegliete il modello di asset a cui desiderate aggiungere un componente.
4. Nella scheda Proprietà, scegliete Componenti.
5. Scegli Crea componente.
6. Nella pagina Crea componente, procedi come segue:
 - a. Immettete un nome per il componente, ad esempio **ServoMotor** o **ServoMotor Model**. Questo nome deve essere univoco per tutti i componenti del tuo account in questa regione.
 - b. (Facoltativo) Aggiungi le definizioni attributi per il modello. Gli attributi rappresentano informazioni che cambiano raramente. Per ulteriori informazioni, consulta [Definisci dati statici \(attributi\)](#).
 - c. (Facoltativo) Aggiungi le definizioni misurazione per il modello. Le misurazioni rappresentano flussi di dati provenienti dall'apparecchiatura. Per ulteriori informazioni, consulta [Definisci i flussi di dati provenienti dalle apparecchiature \(misurazioni\)](#).
 - d. (Facoltativo) Aggiungi le definizioni di trasformazione per il modello. Le trasformazioni sono formule che mappano i dati da un modulo all'altro. Per ulteriori informazioni, consulta [Trasforma i dati \(trasformazioni\)](#).
 - e. (Facoltativo) Aggiungi le definizioni parametro per il modello. Le metriche sono formule che aggregano i dati su intervalli di tempo. Le metriche possono inserire dati dalle risorse associate, in modo da poter calcolare valori che rappresentano l'operazione o un sottoinsieme dell'operazione. Per ulteriori informazioni, consulta [Aggrega i dati provenienti da proprietà e altre risorse \(metriche\)](#).
 - f. Scegliete Crea componente.

Create un modello composito personalizzato in linea (AWS CLI)

È possibile utilizzare il AWS Command Line Interface (AWS CLI) per creare un modello composito personalizzato in linea che definisce le proprie proprietà.

Utilizzate l'[CreateAssetModelCompositeModel](#) operazione per creare un modello in linea con proprietà. Questa operazione prevede un payload con la seguente struttura.

Note

Poiché si tratta di un modello composito in linea, queste proprietà si applicano solo al modello di asset corrente e non sono condivise altrove. Ciò che lo rende «in linea» è che non fornisce un valore per il `composedAssetModelId` campo.

Se avete bisogno di produrre un modello riutilizzabile (ad esempio, da condividere tra più modelli di asset o per includere più istanze all'interno di un unico modello di asset), dovrete invece creare un modello `component-model-based` composito. Per i dettagli, consultate la sezione seguente.

```
{
  "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"
    }
  ]
}
```

Crea un component-model-based componente (console)

È possibile utilizzare la AWS IoT SiteWise console per creare un componente basato su un modello di componente.

Per creare un component-model-based componente (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione selezionare Models (Modelli).
3. Scegliete il modello di asset a cui desiderate aggiungere un componente.
4. Nella scheda Proprietà, scegliete Componenti.
5. Scegli Crea componente.
6. Nella pagina Crea componente, procedi come segue:
 - a. Seleziona il modello di componente su cui vuoi basare il componente.
 - b. Immettete un nome per il componente, ad esempio **ServoMotor** o **ServoMotor Model**. Questo nome deve essere univoco per tutti i componenti del tuo account in questa regione.
 - c. Scegli Crea componente.

Crea un modello composito component-model-based personalizzato (AWS CLI)

Puoi utilizzarli AWS CLI per creare un modello composito component-model-based personalizzato all'interno del tuo modello di asset. Un modello composito component-model-based personalizzato è un riferimento a un modello di componente che hai già definito altrove.

Utilizzate l'[CreateAssetModelCompositeModel](#) operazione per creare un modello composito component-model-based personalizzato. Questa operazione prevede un payload con la seguente struttura.

Note

In questo esempio, il valore di `composedAssetModelId` è l'ID del modello di asset o l'ID esterno di un modello di componente esistente. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise . Per un esempio di come creare un modello di componente, vedete [Crea un modello di componente \(AWS CLI\)](#).

```
{
  "assetModelCompositeModelName": "CNCLathe_ServoMotorA",
  "assetModelCompositeModelType": "CUSTOM",
```

```
"composedAssetModelId": component model ID
]
```

Poiché si tratta solo di un riferimento, un modello composito component-model-based personalizzato non ha proprietà proprie, a parte un nome.

Se desiderate aggiungere più istanze dello stesso componente al vostro modello di asset (ad esempio, una macchina CNC con più servomotori), potete aggiungere più modelli compositi component-model-based personalizzati che hanno ciascuno il proprio nome ma che fanno tutti lo stesso riferimento. `composedAssetModelId`

È possibile annidare i componenti all'interno di altri componenti. A tale scopo, puoi aggiungere un modello component-model-based composito, come mostrato in questo esempio, a uno dei tuoi modelli di componenti.

Create risorse per i modelli di asset in AWS IoT SiteWise

È possibile creare un asset partendo da un modello. Devi avere un modello di asset prima di poter creare un asset. Se non hai creato nessun modello di asset, consulta [Crea modelli di asset in AWS IoT SiteWise](#).

Note

È possibile creare asset solo da modelli ACTIVE. Se lo stato del modello non fosse ACTIVE, bisognerà attendere qualche minuto per poter creare degli asset correlati. Per ulteriori informazioni, consulta [Stati di asset e modelli](#).

Argomenti

- [Creare una risorsa \(console\)](#)
- [Crea una risorsa \(AWS CLI\)](#)
- [Configura una nuova risorsa](#)

Creare una risorsa (console)

Puoi usare la AWS IoT SiteWise console per creare una risorsa.

Per creare un asset (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Asset.
3. Selezionare Create asset (Crea asset).
4. Nella pagina Crea asset, esegui le operazioni seguenti:
 - a. Per Modello, scegli il modello di asset da cui creare un asset.

Note

Se il modello non è ATTIVO, è necessario attendere che diventi attivo oppure risolvere i problemi se è NON RIUSCITO.

- b. Immetti un nome per l'asset.
- c. (Facoltativo) Aggiungi i tag per l'asset. Per ulteriori informazioni, consulta [Tagga le tue AWS IoT SiteWise risorse](#).
- d. Selezionare Create asset (Crea asset).

Quando crei una risorsa, la AWS IoT SiteWise console accede alla pagina della nuova risorsa. In questa pagina puoi vedere lo stato dell'asset che inizialmente è CREAZIONE IN CORSO. Questa pagina si aggiorna automaticamente, quindi attendi l'aggiornamento dello stato dell'asset.

Note

Il processo di creazione di un asset può richiedere fino a un minuto. Dopo che lo Stato è ATTIVO, potete eseguire operazioni di aggiornamento sulla risorsa. Per ulteriori informazioni, consulta [Stati di asset e modelli](#).

Dopo aver creato un asset, consulta [Configura una nuova risorsa](#).

Crea una risorsa (AWS CLI)

È possibile utilizzare AWS Command Line Interface (AWS CLI) per creare una risorsa da un modello di asset.

Per creare un asset, è necessario disporre di un `assetModelId`. Se avete creato un modello di asset, ma non lo conoscete `assetModelId`, utilizzate l'[ListAssetModels](#) API per visualizzare tutti i vostri modelli di asset.

Per creare una risorsa da un modello di asset, utilizzate l'[CreateAsset](#) API con i seguenti parametri:

- `assetName`— Il nome della nuova risorsa. Assegna un nome alla risorsa per aiutarti a identificarla.
- `assetModelId`— L'ID della risorsa. Questo è l'ID effettivo in formato UUID, o `externalId:myExternalId` se ne ha uno. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise .

Per creare una risorsa (AWS CLI)

- Esegui il comando seguente per creare un asset. Sostituiscilo *asset-name* con un nome per la risorsa e *asset-model-id* con l'ID o l'ID esterno del modello di asset.

```
aws iotsitewise create-asset \  
  --asset-name asset-name \  
  --asset-model-id asset-model-id
```

L'azione restituisce una risposta contenente lo stato e i dettagli del nuovo asset nel formato seguente.

```
{  
  "assetId": "String",  
  "assetArn": "String",  
  "assetStatus": {  
    "state": "String",  
    "error": {  
      "code": "String",  
      "message": "String"  
    }  
  }  
}
```

L'elemento `state` è `CREATING` fino a quando l'asset non viene creato.

 Note

Il processo di creazione di un asset può richiedere fino a un minuto. Per verificare lo stato della risorsa, utilizzate l'[DescribeAsset](#) operazione con l'ID della risorsa come `assetId` parametro. Dopo che la risorsa state è ACTIVE pronta, potete eseguire operazioni di aggiornamento sulla risorsa. Per ulteriori informazioni, consulta [Stati di asset e modelli](#).

Dopo aver creato un asset, consulta [Configura una nuova risorsa](#).

Configura una nuova risorsa

Dopo aver creato una risorsa in AWS IoT SiteWise, è possibile eseguire diversi passaggi successivi per utilizzare appieno la risorsa e i relativi dati. Questi passaggi possono includere la configurazione dei flussi di dati per l'acquisizione dei dati dalla risorsa, l'impostazione di allarmi e notifiche per monitorare le prestazioni della risorsa, la creazione di visualizzazioni e dashboard per visualizzare i dati della risorsa e l'integrazione della risorsa con altri AWS servizi o applicazioni di terze parti per ulteriori analisi o automazione.

Completate la configurazione della risorsa con le seguenti azioni opzionali:

- [Gestisci i flussi di dati per AWS IoT SiteWise](#) se l'asset ha proprietà di misurazione.
- [Aggiorna i valori degli attributi](#) se l'asset ha valori di attributo univoci.
- [Associare e dissociare gli asset](#) se l'asset è padre.

Cerca risorse su Console AWS IoT SiteWise

Utilizza la funzionalità Console AWS IoT SiteWise di ricerca per trovare risorse in base ai metadati e ai filtri dei valori delle proprietà in tempo reale.

Prerequisiti

AWS IoT SiteWise richiede autorizzazioni per l'integrazione per organizzare e AWS IoT TwinMaker modellare meglio i dati industriali. Se hai concesso le autorizzazioni a AWS IoT SiteWise, utilizza l'[ExecuteQuery](#) API. Se non hai concesso le autorizzazioni e hai bisogno di assistenza per AWS IoT SiteWise iniziare, consulta. [Integrazione AWS IoT SiteWise e AWS IoT TwinMaker](#)

Ricerca avanzata su Console AWS IoT SiteWise

Ricerca di metadati

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Ricerca avanzata in Risorse.
3. In Ricerca avanzata, scegliete l'opzione di ricerca dei metadati.
4. Compila i parametri. Compila il maggior numero possibile di campi per una ricerca efficiente.
 - a. Nome della risorsa: inserisci un nome completo della risorsa o un nome parziale per una ricerca ampia.
 - b. Nome della proprietà: inserisci il nome completo della proprietà o un nome parziale per una ricerca più ampia.
 - c. Operatore: scegli un operatore tra:
 - =
 - <
 - >
 - <=
 - >=
 - d. Valore della proprietà: questo valore viene confrontato con il valore più recente della proprietà.
 - e. Tipo di valore della proprietà: il tipo di dati della proprietà. Scegli tra le seguenti opzioni:
 - Doppio
 - Numero intero
 - Stringa
 - Booleano
5. Selezionare Search (Cerca).
6. Nella tabella dei risultati della ricerca, scegliete la risorsa dalla colonna Nome. Verrà visualizzata la pagina dettagliata della risorsa.

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

Ricerca parziale

Non è necessario fornire tutti i parametri per la ricerca di risorse. Ecco alcuni esempi di ricerche parziali che utilizzano l'opzione di ricerca Metadati:

- Trova le risorse in base al loro nome:
 - Inserisci un valore solo nel campo Nome risorsa.
 - I campi Nome della proprietà e Valore della proprietà sono vuoti.
- Trova risorse contenenti proprietà con un nome specifico:
 - Inserisci un valore solo nel campo Nome della proprietà.
 - I campi Nome della risorsa e Valore della proprietà sono vuoti.
- Trova le risorse in base ai valori più recenti delle loro proprietà:
 - Immettete i valori nei campi Nome della proprietà e Valore della proprietà.
 - Seleziona un tipo di valore dell'operatore e della proprietà.

Ricerca con Query Builder

1. Passare alla Console AWS IoT SiteWise.

2. Nel riquadro di navigazione, scegli Ricerca avanzata in Risorse.
3. In Ricerca avanzata scegli l'opzione Query builder.
4. Nel riquadro Query Builder, scrivi la tua query SQL per recuperare unasset_name, e.
asset_id asset_description
5. Selezionare Search (Cerca).
6. Dalla tabella dei risultati della ricerca, scegliete la risorsa dalla colonna Nome. Verrà visualizzata la pagina dettagliata della risorsa.

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' page is displayed. It includes a 'Create asset' button and a description of assets. The 'Advanced search' section has two tabs: 'Metadata search' and 'Query builder'. The 'Query builder' tab is active, showing 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 results (2)' section shows a table with two results.

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

- La SELECT clausola della query SQL deve includere i asset_id campi asset_name and per garantire una risorsa valida nella tabella dei risultati della ricerca.

- Il generatore di query visualizza solo il nome, l'ID della risorsa e la descrizione nella tabella dei risultati. L'aggiunta di altri campi alla SELECT clausola non aggiunge altre colonne alla tabella dei risultati

Aggiorna i valori degli attributi

Gli asset ereditano gli attributi del proprio modello con i relativi valori predefiniti. In alcuni casi, è consigliabile mantenere l'attributo predefinito del modello di asset; ad esempio, per la proprietà del produttore di un asset. In altri casi, invece, occorre aggiornare l'attributo ereditato; ad esempio, per la latitudine e la longitudine di un asset.

Updating an attribute value (console)

È possibile utilizzare la AWS IoT SiteWise console per aggiornare il valore di una proprietà di un asset attributo.

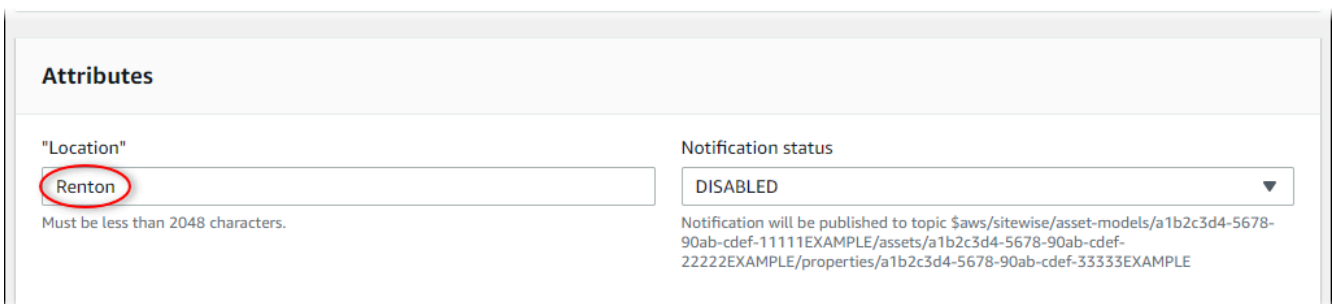
Per aggiornare il valore di un attributo (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Asset.
3. Scegli l'asset per cui vuoi aggiornare un attributo.

Tip

Puoi scegliere l'icona a forma di freccia per espandere una gerarchia di asset e trovare il tuo asset.

4. Scegli Modifica.
5. Individua l'attributo da aggiornare e quindi immetti il nuovo valore.



Attributes

"Location"

Renton

Must be less than 2048 characters.

Notification status

DISABLED

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

6. Scegli Save (Salva).

Updating an attribute value (AWS CLI)

È possibile utilizzare AWS Command Line Interface (AWS CLI) per aggiornare il valore di un attributo.

Per completare questa procedura, è necessario conoscere l'elemento `assetId` dell'asset e l'elemento `propertyId` della proprietà. Puoi anche usare l'ID esterno. Se hai creato una risorsa e non la conosci `assetId`, utilizza l'[ListAssets](#) API per elencare tutte le risorse per un modello specifico. Utilizzate l'[DescribeAsset](#) operazione per visualizzare le proprietà della risorsa, inclusa la proprietà IDs.

L'operazione [BatchPutAssetPropertyValue](#) consente di assegnare i valori degli attributi all'asset. È possibile utilizzare questa operazione per impostare più attributi contemporaneamente. Il payload di questa operazione include un elenco di voci, ciascuna delle quali contenente l'ID asset, l'ID proprietà e il valore dell'attributo.

Per aggiornare il valore di un attributo (AWS CLI)

1. Crea un file denominato `batch-put-payload.json` e copia il seguente oggetto JSON nel file. Questo esempio di payload mostra come impostare la latitudine e la longitudine di una turbina eolica. Aggiorna i valori IDs, e i timestamp per modificare il payload per il tuo caso d'uso.

```
{
  "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
      }
    }
  ]
}
```

- Ogni voce nel payload contiene un `entryId` che è possibile definire come una qualsiasi stringa univoca. Se una richiesta non riesce, ciascun errore conterrà l'`entryId` della richiesta corrispondente in modo che sia possibile sapere quale richiesta riprovare.
- Per impostare il valore di un attributo, è possibile includere una struttura `timestamp-quality-value` (TQV) nell'elenco di proprietà di `propertyValues` ogni attributo. Questa struttura deve contenere il nuovo `value` e il `timestamp` corrente.
 - `value`— Una struttura che contiene uno dei seguenti campi, a seconda del tipo di proprietà impostata:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`
 - `timestamp`— Una struttura che contiene l'ora attuale dell'epoca Unix in secondi,. `timeInSeconds` AWS IoT SiteWise rifiuta tutti i punti dati con `timestamp` che esistevano da più di 7 giorni nel passato o più recenti di 5 minuti nelle future.

Per ulteriori informazioni su come preparare un payload per [BatchPutAssetPropertyValue](#), consulta [Inserisci dati con AWS IoT SiteWise APIs](#).

2. Esegui il comando seguente per inviare i valori degli attributi a: AWS IoT SiteWise


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

Associare e dissociare gli asset

Se il modello di asset li prevede nelle proprie gerarchie, è possibile associare asset figlio all'asset padre. Gli asset padre possono accedere e aggregare i dati dagli asset associati. Per ulteriori informazioni sui modelli di asset gerarchici, consulta [Definite le gerarchie dei modelli di asset](#).

Argomenti

- [Associa e dissocia gli asset \(console\)](#)
- [Associa e dissocia le risorse \(\)AWS CLI](#)

Associa e dissocia gli asset (console)

Puoi utilizzare la AWS IoT SiteWise console per associare e dissociare le risorse.

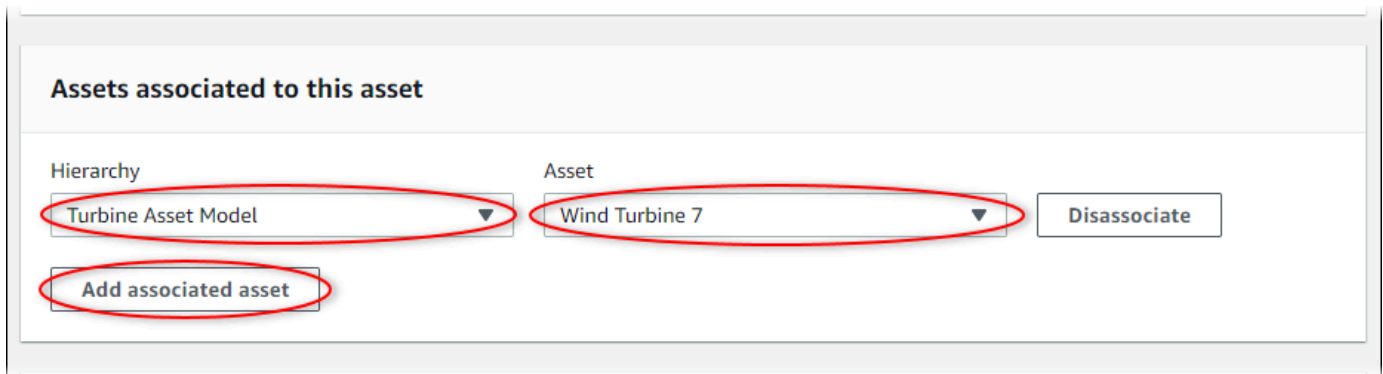
Per associare un asset (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Asset.
3. Scegli l'asset padre a cui vuoi associare un asset figlio.

Tip

Puoi scegliere l'icona a forma di freccia per espandere una gerarchia di asset e trovare il tuo asset.

4. Scegli Modifica.
5. In Asset associati a questo asset scegli Aggiungi asset associato.



Assets associated to this asset

Hierarchy: Turbine Asset Model ▼

Asset: Wind Turbine 7 ▼

Disassociate

Add associated asset

6. Per Gerarchia, scegli la gerarchia che definisce la relazione tra l'asset padre e l'asset figlio.
7. Per Asset, scegli l'asset figlio da associare.
8. Scegli Save (Salva).

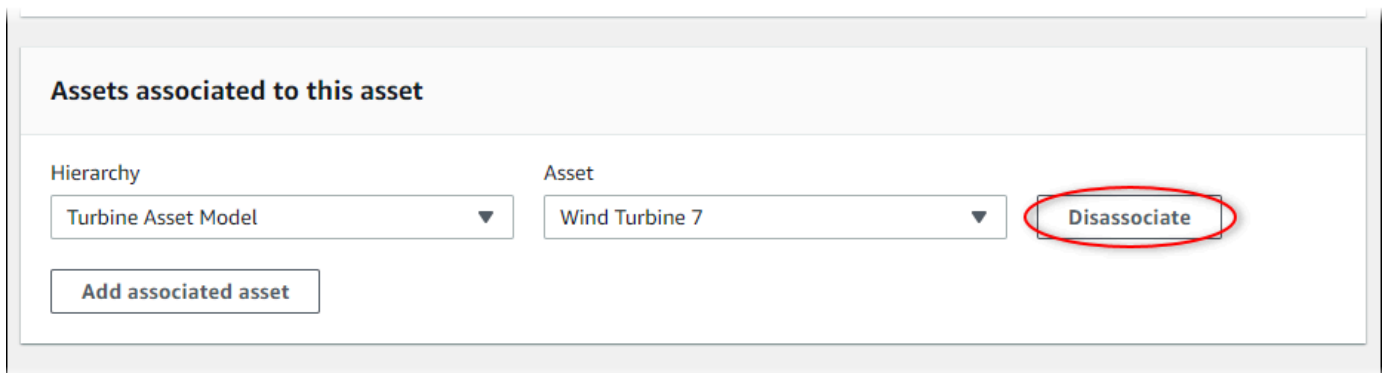
Per annullare l'associazione di un asset (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Asset.
3. Scegli l'asset padre per cui vuoi annullare l'associazione di un asset figlio.

Tip

Puoi scegliere l'icona a forma di freccia per espandere una gerarchia di asset e trovare il tuo asset.

4. Scegli Modifica.
5. In Asset associati a questo asset scegli Annulla associazione.



Assets associated to this asset

Hierarchy: Turbine Asset Model ▼

Asset: Wind Turbine 7 ▼

Disassociate

Add associated asset

6. Scegli Save (Salva).

Associa e dissocia le risorse (AWS CLI)

È possibile utilizzare AWS Command Line Interface (AWS CLI) per associare e dissociare gli asset.

Per questa procedura, devi conoscere l'ID della gerarchia (`hierarchyId`) presente nel modello di asset padre che definisce la relazione con il modello di asset figlio. Utilizzate l'[DescribeAsset](#) operazione per trovare l'ID della gerarchia nella risposta.

Per trovare un ID gerarchia

- Esegui il comando seguente per descrivere l'asset padre. Sostituisci *parent-asset-id* con l'ID o l'ID esterno della risorsa principale.

```
aws iotsitewise describe-asset --asset-id parent-asset-id
```

L'operazione restituisce una risposta contenente i dettagli dell'asset. La risposta contiene un `assetHierarchies` elenco con la seguente struttura:

```
{
  ...
  "assetHierarchies": [
    {
      "id": "String",
      "name": "String"
    }
  ],
  ...
}
```

L'ID gerarchia è il valore `id` di una gerarchia nell'elenco delle gerarchie di asset.

Una volta ottenuto l'ID gerarchia puoi associare o annullare l'associazione di un asset alla gerarchia.

Per associare una risorsa secondaria a una risorsa principale, utilizzate l'[AssociateAssets](#) operazione. Per dissociare una risorsa secondaria da una risorsa principale, utilizzate l'[DisassociateAssets](#) operazione. Specifica i seguenti parametri, che sono gli stessi per entrambe le operazioni:

- `assetId`— L'ID o l'ID esterno della risorsa principale.
- `hierarchyId`— L'ID della gerarchia o l'ID esterno nella risorsa principale.

- `childAssetId`— L'ID o l'ID esterno della risorsa secondaria.

Per associare una risorsa (AWS CLI)

- Esegui il comando seguente per associare un asset figlio a un asset padre. Sostituisci *parent-asset-id* con l'ID dell'asset padre, *hierarchy-id* con l'ID della gerarchia e *child-asset-id* con l'ID dell'asset figlio:

```
aws iotsitewise associate-assets \
  --asset-id parent-asset-id \
  --hierarchy-id hierarchy-id \
  --child-asset-id child-asset-id
```

Per dissociare una risorsa (AWS CLI)

- Esegui il comando seguente per annullare l'associazione di un asset figlio a un asset padre. Sostituisci *parent-asset-id* con l'ID dell'asset padre, *hierarchy-id* con l'ID della gerarchia e *child-asset-id* con l'ID dell'asset figlio:

```
aws iotsitewise disassociate-assets \
  --asset-id parent-asset-id \
  --hierarchy-id hierarchy-id \
  --child-asset-id child-asset-id
```

Aggiorna risorse e modelli

È possibile aggiornare gli asset, i modelli di asset e i modelli dei componenti AWS IoT SiteWise per modificarne i nomi e le definizioni. Queste operazioni di aggiornamento sono asincrone e richiedono tempo per essere propagate. AWS IoT SiteWise Controllate lo stato della risorsa o del modello prima di apportare ulteriori modifiche. È necessario attendere la propagazione delle modifiche prima di poter continuare a utilizzare l'asset o il modello aggiornato.

Argomenti

- [Aggiorna le risorse in AWS IoT SiteWise](#)
- [Aggiorna i modelli di asset e i modelli dei componenti](#)
- [Aggiorna modelli compositi personalizzati \(componenti\)](#)
- [Blocco ottimistico per le scritture dei modelli di asset](#)

Aggiorna le risorse in AWS IoT SiteWise

Puoi utilizzare la AWS IoT SiteWise console o l'API per aggiornare il nome di una risorsa.

Quando aggiorni una risorsa, lo stato della risorsa rimane invariato UPDATING fino alla propagazione delle modifiche. Per ulteriori informazioni, consulta [Stati di asset e modelli](#).

Argomenti

- [Aggiorna una risorsa \(console\)](#)
- [Aggiorna una risorsa \(AWS CLI\)](#)

Aggiorna una risorsa (console)

Puoi utilizzare la AWS IoT SiteWise console per aggiornare i dettagli della risorsa.

Per aggiornare un asset (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Asset.
3. Scegli l'asset da aggiornare.

Tip

Puoi scegliere l'icona a forma di freccia per espandere una gerarchia di asset e trovare il tuo asset.

4. Scegli Modifica.
5. Aggiorna il nome dell'asset.
6. (Facoltativo) In questa pagina aggiorna altre informazioni relative all'asset. Per ulteriori informazioni, consulta gli argomenti seguenti:
 - [Gestisci i flussi di dati per AWS IoT SiteWise](#)
 - [Aggiorna i valori degli attributi](#)
 - [Interagisci con altri AWS servizi](#)
7. Scegli Save (Salva).

Aggiorna una risorsa (AWS CLI)

Puoi usare il AWS Command Line Interface (AWS CLI) per aggiornare il nome di una risorsa.

Utilizzate l'[UpdateAsset](#) operazione per aggiornare una risorsa. Specifica i seguenti parametri:

- `assetId`— L'ID della risorsa. Questo è l'ID effettivo in formato UUID, o `externalId:myExternalId` se ne ha uno. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise .
- `assetName`— Il nuovo nome della risorsa.

Per aggiornare il nome di una risorsa (AWS CLI)

- Esegui il comando seguente per aggiornare il nome di un asset. Sostituisci *asset-id* con l'ID o l'ID esterno della risorsa. Aggiorna il *asset-name* con il nuovo nome della risorsa.

```
aws iotsitewise update-asset \  
  --asset-id asset-id \  
  --asset-name asset-name
```

Aggiorna i modelli di asset e i modelli dei componenti

Utilizza la AWS IoT SiteWise console o l'API per aggiornare un modello di asset o un modello di componente.

Non puoi modificare il tipo o il tipo di dati di una proprietà esistente o la finestra di una metrica esistente. Inoltre, non è possibile modificare il tipo di modello da modello di asset a modello di componenti o viceversa.

Important

- Se rimuovete una proprietà da un modello di asset o da un modello di componente, AWS IoT SiteWise elimina tutti i dati precedenti relativi a quella proprietà. Per quanto riguarda i modelli a componenti, ciò influisce su tutti i modelli di asset che utilizzano quel modello di componenti, quindi prestate particolare attenzione a capire in che misura la modifica può essere applicata.

- Se rimuovete una definizione di gerarchia da un modello di asset, AWS IoT SiteWise dissocia tutti gli asset in quella gerarchia.

Quando aggiorni un modello, anche gli asset basati su quel modello assimilano le modifiche apportate. Fino a quando le modifiche non si propagano, ogni asset ha lo stato `UPDATING`. È necessario attendere fino a quando gli asset ritornano nello stato `ACTIVE` prima di poter interagire con loro. In questo lasso di tempo, lo stato del modello di asset aggiornato sarà `PROPAGATING`.

Quando aggiornate un modello di componente, ogni modello di asset che incorpora quel modello di componente riflette le modifiche. Fino a quando le modifiche al modello di componente non si propagano, ogni modello di asset interessato presenta `UPDATING` lo stato, seguito dall'`PROPAGATING` aggiornamento degli asset associati, come descritto nel paragrafo precedente. È necessario attendere che tali modelli di asset tornino allo `ACTIVE` stato precedente prima di interagire con essi. Durante questo periodo, lo stato del modello di componente aggiornato sarà `PROPAGATING`.

Per ulteriori informazioni, consulta [Stati di asset e modelli](#).

Argomenti

- [Aggiornamento di un modello di asset o componente \(console\)](#)
- [Aggiornate un modello di asset o componente \(\)AWS CLI](#)

Aggiornamento di un modello di asset o componente (console)

È possibile utilizzare la AWS IoT SiteWise console per aggiornare un modello di asset o un modello di componente.

Per aggiornare un modello di asset o un modello di componente (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione selezionare Models (Modelli).
3. Scegliete il modello di asset o il modello di componente da aggiornare.
4. Scegli Modifica.
5. Nella pagina Modifica modello esegui una delle operazioni seguenti:
 - In Dettagli del modello modifica il nome del modello.
 - Modifica una delle definizioni degli attributi. Non è possibile modificare il tipo di dati degli attributi esistenti. Per ulteriori informazioni, consulta [Definisci dati statici \(attributi\)](#).

- Modifica una delle definizioni di misurazione. Non è possibile modificare il tipo di dati delle misurazioni esistenti. Per ulteriori informazioni, consulta [Definisci i flussi di dati provenienti dalle apparecchiature \(misurazioni\)](#).
- Modifica una delle definizioni di trasformazione. Per ulteriori informazioni, consulta [Trasforma i dati \(trasformazioni\)](#).
- Modifica una delle definizioni di parametro. Non è possibile modificare l'intervallo di tempo dei parametri esistenti. Per ulteriori informazioni, consulta [Aggrega i dati provenienti da proprietà e altre risorse \(metriche\)](#).
- (Solo modelli di asset) Modificate una qualsiasi definizione della gerarchia. Non è possibile modificare il modello di gerarchia delle gerarchie esistenti. Per ulteriori informazioni, consulta [Definite le gerarchie dei modelli di asset](#).

6. Scegli Save (Salva).

Note

Le richieste di aggiornamento effettuate nella console vengono rifiutate se un altro utente aggiorna correttamente il modello di asset dall'ultima apertura della pagina Modifica modello. La console richiede all'utente di aggiornare la pagina Modifica modello per recuperare il modello aggiornato. È necessario effettuare nuovamente gli aggiornamenti e riprovare a salvare. Per ulteriori dettagli, consulta [Blocco ottimistico per le scritture dei modelli di asset](#).

Aggiornate un modello di asset o componente ()AWS CLI

Utilizzate il AWS Command Line Interface (AWS CLI) per aggiornare un modello di asset o un modello di componente.

Utilizzate l'[UpdateAssetModel](#) API per aggiornare il nome, la descrizione e le proprietà di un modello di asset o di un modello di componente. Solo per i modelli di asset, puoi aggiornare le gerarchie. Specifica i seguenti parametri:

- `assetModelId`— L'ID della risorsa. Questo è l'ID effettivo in formato UUID, o `externalId:myExternalId` se ne ha uno. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise .

Specificate il modello aggiornato nel payload. Per maggiori informazioni sul formato previsto di un modello di asset o di un modello di componente, consulta [Crea modelli di asset in AWS IoT SiteWise](#).

 Warning

L'[UpdateAssetModel](#) API sovrascrive il modello esistente con il modello fornito nel payload. Per evitare di eliminare le proprietà o le gerarchie del modello, è necessario includere le relative definizioni nel payload IDs del modello aggiornato. Per informazioni su come interrogare la struttura esistente del modello, consultate l'operazione. [DescribeAssetModel](#)

 Note

La procedura seguente può aggiornare solo modelli composti di tipo AWS/ALARM. Se desideri aggiornare i modelli CUSTOM composti, usa [UpdateAssetModelCompositeModel](#) invece. Per ulteriori informazioni, consulta [Aggiorna modelli composti personalizzati \(componenti\)](#).

Per aggiornare un modello di asset o un modello di componente (AWS CLI)

1. Eseguite il comando seguente per recuperare la definizione del modello esistente. *asset-model-id* Sostituitelo con l'ID o l'ID esterno del modello di asset o del modello di componente da aggiornare.

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

Il comando precedente restituisce la definizione del modello corrispondente alla versione più recente del modello.

Per un caso d'uso in cui un modello di asset si trova in uno FAILED stato, recuperate la definizione di modello valida corrispondente alla sua versione attiva per creare la richiesta di aggiornamento. Per informazioni dettagliate, vedi [Versioni del modello di asset](#). Eseguite il comando seguente per recuperare la definizione attiva del modello:

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id --asset-model-version ACTIVE
```

L'operazione restituisce una risposta che contiene i dettagli del modello. La risposta ha la seguente struttura.

```
{
  "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"
}
```

Per maggiori informazioni, vedi l'operazione [DescribeAssetModel](#).

2. Crea un file denominato `update-asset-model.json` e copia la risposta del comando precedente nel file.
3. Rimuovi le seguenti coppie chiave-valore dall'oggetto JSON in `update-asset-model.json`:
 - `assetModelId`
 - `assetModelArn`
 - `assetModelCompositeModelSummaries`
 - `assetModelCreationDate`
 - `assetModelLastUpdateDate`
 - `assetModelStatus`
 - `assetModelType`

- `assetModelVersion`
- `eTag`

L'[UpdateAssetModel](#) operazione prevede un payload con la seguente struttura:

```
{
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition,
  "assetModelCompositeModels": Array of AssetModelCompositeModel
}
```

4. In `update-asset-model.json`, effettua una delle seguenti operazioni:
 - Modifica il nome del modello di asset (`assetModelName`).
 - Modifica, aggiungi o rimuovi la descrizione del modello di asset (`assetModelDescription`).
 - Modifica, aggiungi o rimuovi qualsiasi proprietà del modello di asset (`assetModelProperties`). Non è possibile modificare l'elemento `dataType` delle proprietà esistenti o l'elemento `window` dei parametri esistenti. Per ulteriori informazioni, consulta [Definite le proprietà dei dati](#).
 - Modifica, aggiungi o rimuovi una delle gerarchie del modello di asset (`assetModelHierarchies`). Non è possibile modificare l'elemento `childAssetModelId` delle gerarchie esistenti. Per ulteriori informazioni, consulta [Definite le gerarchie dei modelli di asset](#).
 - Modificate, aggiungete o rimuovete uno qualsiasi dei modelli composti di type `AWS/ALARM` (`assetModelCompositeModels`) del modello di asset. Gli allarmi monitorano altre proprietà in modo da poter identificare quando le apparecchiature o i processi richiedono attenzione. Ogni definizione di allarme è un modello composito che standardizza l'insieme di proprietà utilizzate dall'allarme. Per ulteriori informazioni, consultare [Monitora i dati con allarmi in AWS IoT SiteWise](#) e [Definisci gli allarmi sui modelli di asset in AWS IoT SiteWise](#).
5. Esegui il comando seguente per aggiornare il modello di asset con la definizione memorizzata in `update-asset-model.json`. Sostituisci `asset-model-id` con l'ID del modello di asset:

```
aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
```

```
--cli-input-json file://model-payload.json
```

Important

Quando più utenti aggiornano un modello di asset contemporaneamente, le modifiche di un utente possono essere inavvertitamente sovrascritte da un altro utente. Per evitare ciò, è necessario definire una richiesta di aggiornamento condizionale. Consultare [Blocco ottimistico per le scritture dei modelli di asset](#).

Aggiorna modelli compositi personalizzati (componenti)

È possibile utilizzare l' AWS IoT SiteWise API per aggiornare un modello composito personalizzato o la AWS IoT SiteWise console per aggiornare i componenti.

Argomenti

- [Aggiorna un componente \(console\)](#)
- [Aggiornate un modello composito personalizzato \(AWS CLI\)](#)

Aggiorna un componente (console)

È possibile utilizzare la AWS IoT SiteWise console per aggiornare un componente.

Per aggiornare un componente (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione selezionare Models (Modelli).
3. Scegliete il modello di asset in cui si trova il componente.
4. Nella scheda Proprietà, scegliete Componenti.
5. Scegliete il componente che desiderate aggiornare.
6. Scegli Modifica.
7. Nella pagina Modifica componente, effettuate una delle seguenti operazioni:
 - In Dettagli del modello modifica il nome del modello.
 - Modifica una delle definizioni degli attributi. Non è possibile modificare il tipo di dati degli attributi esistenti. Per ulteriori informazioni, consulta [Definisci dati statici \(attributi\)](#).

- Modifica una delle definizioni di misurazione. Non è possibile modificare il tipo di dati delle misurazioni esistenti. Per ulteriori informazioni, consulta [Definisci i flussi di dati provenienti dalle apparecchiature \(misurazioni\)](#).
- Modifica una delle definizioni di trasformazione. Per ulteriori informazioni, consulta [Trasforma i dati \(trasformazioni\)](#).
- Modifica una delle definizioni di parametro. Non è possibile modificare l'intervallo di tempo dei parametri esistenti. Per ulteriori informazioni, consulta [Aggrega i dati provenienti da proprietà e altre risorse \(metriche\)](#).

8. Scegli Save (Salva).

Aggiornate un modello composito personalizzato (AWS CLI)

Usa il AWS Command Line Interface (AWS CLI) per aggiornare un modello composito personalizzato.

Per aggiornare il nome o la descrizione, utilizzate l'[UpdateAssetModelCompositeModel](#) operazione. Solo per i modelli compositi personalizzati in linea, puoi anche aggiornare le proprietà. Non è possibile aggiornare le proprietà di un modello composito component-model-based personalizzato, poiché il modello di componente a cui fa riferimento fornisce le proprietà associate.

Important

Se rimuovete una proprietà da un modello composito personalizzato, AWS IoT SiteWise elimina tutti i dati precedenti relativi a quella proprietà. Non è possibile modificare il tipo o il tipo di dati di una proprietà esistente.

Per sostituire una proprietà esistente del modello composito con una nuova con la stessa proprietàname, procedi come segue:

1. Inviare una `UpdateAssetModelCompositeModel` richiesta con l'intera proprietà esistente rimossa.
2. Invia una seconda `UpdateAssetModelCompositeModel` richiesta che includa la nuova proprietà. La nuova proprietà dell'asset avrà la name stessa di quella precedente e AWS IoT SiteWise genererà una nuova proprietà univoca `id`.

Per aggiornare un modello composito personalizzato (AWS CLI)

1. Per recuperare la definizione del modello composito esistente, eseguite il comando seguente. Sostituitelo *composite-model-id* con l'ID o l'ID esterno del modello composito personalizzato da aggiornare e *asset-model-id* con il modello di asset a cui è associato il modello composito personalizzato. Per ulteriori informazioni, consulta la AWS IoT SiteWise Guida per l'utente di .

- a. Eseguite il comando seguente:

```
aws iotsitewise describe-asset-model-composite-model \  
--asset-model-composite-model-id composite-model-id \  
--asset-model-id asset-model-id
```

- b. Il comando precedente restituisce la definizione del modello composito corrispondente all'ultima versione del modello associato. Per un caso d'uso in cui un modello di asset si trova in uno FAILED stato, recuperate la definizione di modello valida corrispondente alla sua versione attiva per creare la richiesta di aggiornamento. Per informazioni dettagliate, vedi [Versioni del modello di asset](#).

- c. Eseguite il comando seguente per recuperare la definizione attiva del modello:

```
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. Per maggiori informazioni, vedi l'operazione [DescribeAssetModelCompositeModel](#).
2. Create un file chiamato `update-custom-composite-model.json`, quindi copiate la risposta del comando precedente nel file.
 3. Rimuovi ogni coppia chiave-valore dall'oggetto JSON ad `update-custom-composite-model.json` eccezione dei seguenti campi:
 - `assetModelCompositeModelName`
 - `assetModelCompositeModelDescription`(se presente)
 - `assetModelCompositeModelProperties`(se presente)
 4. In `update-custom-composite-model.json`, effettua una delle seguenti operazioni:
 - Modificare il valore di `assetModelCompositeModelName`.
 - Aggiungi o `assetModelCompositeModelDescription` rimuovi o modifichane il valore.

- Solo per i modelli compositi personalizzati in linea: modifica, aggiungi o rimuovi qualsiasi proprietà del modello di asset in `assetModelCompositeModelProperties`.

Per ulteriori informazioni sul formato richiesto per questo file, consultate la sintassi della richiesta per [UpdateAssetModelCompositeModel](#).

5. Eseguite il comando seguente per aggiornare il modello composito personalizzato con la definizione memorizzata in `update-custom-composite-model.json`. Sostituiscilo *composite-model-id* con l'ID del modello composito e *asset-model-id* con l'ID del modello di asset in cui si trova.

```
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

Quando più utenti aggiornano un modello di asset contemporaneamente, le modifiche di un utente possono essere inavvertitamente sovrascritte da un altro utente. Per evitare ciò, è necessario definire una richiesta di aggiornamento condizionale. Consultare [Blocco ottimistico per le scritture dei modelli di asset](#).

Blocco ottimistico per le scritture dei modelli di asset

Quando aggiorna un modello di asset, un utente esegue le seguenti operazioni:

1. Leggete la definizione attuale del modello di asset.
2. Modifica la definizione del modello di asset con le modifiche richieste.
3. Aggiorna il modello di asset con la nuova definizione.

In uno scenario con due utenti che aggiornano un modello, è possibile quanto segue:

- L'utente A legge la definizione X del modello di asset.
- L'utente B legge la definizione del modello di asset X e apporta le modifiche, modificando la definizione di X.

- L'utente A esegue il commit e sovrascrive la modifica apportata dall'utente B per il modello di asset X, senza verificare o incorporare le modifiche dell'utente B.

Il blocco ottimistico è un meccanismo utilizzato da per prevenire sovrascritture accidentali AWS IoT SiteWise come nello scenario precedente. Il blocco ottimistico è una strategia per garantire che la versione corrente del modello di asset venga aggiornata o eliminata e che corrisponda alla versione attuale. AWS IoT SiteWise Ciò protegge le scritture del modello di asset dalla sovrascrittura da aggiornamenti accidentali.

Segui questi passaggi per eseguire la scrittura del modello di asset con blocco ottimistico:

Argomenti

- [Esecuzione di scritture su modelli di asset con blocco ottimistico \(console\)](#)
- [Esecuzione di scritture su modelli di asset con optimistic lock \(\)AWS CLI](#)

Esecuzione di scritture su modelli di asset con blocco ottimistico (console)

La procedura riportata di seguito descrive come eseguire le scritture dei modelli di asset con un blocco ottimistico sulla versione attiva del modello di asset nella console.

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione selezionare Models (Modelli).
3. Scegliete il modello di asset o il modello di componente da aggiornare.
4. Scegli Modifica.
5. Apporta le modifiche nella pagina Modifica modello.
6. Scegli Save (Salva).

Note

A volte, uno o più aggiornamenti del modello sono stati eseguiti correttamente tra il momento in cui l'utente inizia a modificare il modello e il salvataggio delle modifiche apportate al modello.

Per garantire che l'utente non sovrascriva accidentalmente i nuovi aggiornamenti riusciti, la scrittura dell'utente viene rifiutata. La console disattiva il pulsante Salva e richiede all'utente di aggiornare la pagina Modifica modello. L'utente deve aggiornare

nuovamente la nuova versione attiva del modello. L'utente deve eseguire i seguenti passaggi aggiuntivi:

7. Scegliere Refresh (Aggiorna).
8. Segui nuovamente i passaggi 5 e 6.

Esecuzione di scritture su modelli di asset con optimistic lock ()AWS CLI

La procedura riportata di seguito descrive come eseguire la scrittura di modelli di asset con blocco ottimistico in AWS CLI

1. Recupera la definizione del modello ETag associata alla definizione corrente

ETag è un token unico generato per ogni nuova rappresentazione di un modello di asset. Chiama [DescribeAssetModel](#) l'API per recuperare la definizione corrente del modello di asset e la relativa definizione associata alla ETag risposta.

Durante gli aggiornamenti simultanei, gli utenti eseguono aggiornamenti con successo (modello in ACTIVE stato) o aggiornamenti non riusciti (modello in FAILED stato). Per evitare che un utente sovrascriva accidentalmente un aggiornamento riuscito, dovete recuperare la versione attiva del modello di asset da [Versioni del modello di asset](#) e ottenerne il valore. ETag

Esegui il comando seguente:

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id \
--asset-model-version ACTIVE
```

La risposta restituisce la seguente struttura:

```
{
  "assetModelId": "String",
  "assetModelArn": "String",
  "assetModelName": "String",
  ...
  "eTag": "String"
}
```

 Note

È necessario recuperare la versione più recente del modello di asset e la relativa ETag per non sovrascrivere gli aggiornamenti.

2. Esegui le operazioni UPDATE e DELETE con condizioni di scrittura

Il seguente modello di asset APIs supporta il blocco ottimistico:

- [UpdateAssetModel](#)
- [DeleteAssetModel](#)
- [CreateAssetModelCompositeModel](#)
- [UpdateAssetModelCompositeModel](#)
- [DeleteAssetModelCompositeModel](#)

 Note

Gli scenari seguenti utilizzano l'UpdateAssetModelAPI come riferimento. Le condizioni si applicano a tutte le operazioni sopra elencate.

Gli scenari seguenti descrivono le diverse condizioni di scrittura a seconda dei requisiti di controllo della concorrenza:

- Esegui il comando seguente per non sovrascrivere eventuali aggiornamenti riusciti. Non deve esistere una nuova versione attiva, a partire dall'ultima versione attiva letta. Sostituisci e - tag con quella ETag restituita nell'operazione API utilizzata nella lettura della versione attiva.

```
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
```

- Quando la creazione di un modello fallisce, non esiste ancora una versione attiva del modello, perché è in uno FAILED stato. È comunque possibile sovrascrivere una nuova versione

attiva presente, prima che le modifiche vengano confermate. Esegui il comando seguente per non sovrascrivere una nuova versione attiva, quando una versione attiva non esiste durante l'ultima lettura.

```
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
```

- Esegui il comando seguente per evitare di sovrascrivere eventuali aggiornamenti riusciti o meno. Questo comando definisce una condizione di scrittura che garantisce che non venga creata una versione più recente dopo l'ultima versione letta. Sostituisci `e-tag` con l'operazione API ETag restituita nella lettura della versione attiva.

```
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
```

Se la condizione di scrittura restituisce un risultato positivo `FALSE`, la richiesta di scrittura ha esito negativo e restituisce `PreconditionFailedException`.

Eliminare risorse e modelli in AWS IoT SiteWise

Puoi eliminare le tue risorse e i tuoi modelli una AWS IoT SiteWise volta che hai finito di utilizzarli. Le operazioni di eliminazione sono asincrone e richiedono tempo per essere propagate. AWS IoT SiteWise

Argomenti

- [Eliminare le risorse in AWS IoT SiteWise](#)
- [Eliminare i modelli di asset in AWS IoT SiteWise](#)

Eliminare le risorse in AWS IoT SiteWise

Puoi utilizzare la AWS IoT SiteWise console o l'API per eliminare una risorsa non più necessaria nel tuo ambiente. L'eliminazione di un modello di asset elimina anche tutti gli asset e i modelli di componenti associati. Tuttavia, è importante notare che l'eliminazione di una risorsa o di un modello è un'azione permanente e che vengono rimossi anche tutti i dati associati alle risorse eliminate. Prima di eliminare risorse o modelli, si consiglia di esaminare eventuali dipendenze o integrazioni che potrebbero essere influenzate e assicurarsi di disporre di un backup di tutti i dati importanti.

Prima di eliminare un asset, è necessario innanzitutto annullare l'associazione degli asset figlio all'asset padre. Per ulteriori informazioni, consulta [Associare e dissociare gli asset](#). Se utilizzate il AWS Command Line Interface (AWS CLI), potete utilizzare l'[ListAssociatedAssets](#) operazione per elencare i figli di una risorsa.

Quando elimini un asset, lo stato dell'asset è DELETING fino a quando le modifiche non vengono propagate. Per ulteriori informazioni, consulta [Stati di asset e modelli](#). Non è possibile eseguire query su un asset eliminato. Se lo facessi, l'API restituirebbe una risposta HTTP 404.

Important

AWS IoT SiteWise elimina tutti i dati delle proprietà degli asset eliminati.

Argomenti

- [Eliminare una risorsa \(console\)](#)
- [Eliminare una risorsa \(AWS CLI\)](#)

Eliminare una risorsa (console)

Puoi utilizzare la AWS IoT SiteWise console per eliminare una risorsa.

Per eliminare un asset (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Asset.
3. Scegli l'asset da eliminare.

 Tip

Puoi scegliere l'icona a forma di freccia per espandere una gerarchia di asset e trovare il tuo asset.

4. Se l'asset ha asset associati, elimina ogni asset associato. Puoi scegliere il nome di un asset per aprire la relativa pagina in cui è possibile eliminarlo.
5. Nella pagina dell'asset, scegli Elimina.
6. Nella finestra di dialogo Elimina risorsa, effettuate le seguenti operazioni:
 - a. Immetti **Delete** per confermare l'eliminazione.
 - b. Scegliere Delete (Elimina).

Eliminare una risorsa (AWS CLI)

È possibile utilizzare il AWS Command Line Interface (AWS CLI) per eliminare una risorsa.

Utilizzate l'[DeleteAsset](#) operazione per eliminare una risorsa. Specifica il parametro seguente:

- `assetId`— L'ID della risorsa. Questo è l'ID effettivo in formato UUID, o `externalId:myExternalId` se ne ha uno. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise .

Per eliminare una risorsa (AWS CLI)

1. Esegui il comando seguente per elencare le gerarchie dell'asset. Sostituisci *asset-id* con l'ID o l'ID esterno della risorsa:

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

L'operazione restituisce una risposta contenente i dettagli dell'asset. La risposta contiene un `assetHierarchies` elenco con la seguente struttura:

```
{
  ...
  "assetHierarchies": [
    {
```

```
    "id": "String",  
    "name": "String"  
  }  
],  
  ...  
}
```

Per maggiori informazioni, vedi l'operazione [DescribeAsset](#).

2. Per ogni gerarchia, esegui il comando seguente per elencare gli elementi figlio dell'asset associati alla gerarchia. Sostituisci *asset-id* con l'ID o l'ID esterno della risorsa e *hierarchy-id* con l'ID o l'ID esterno della gerarchia.

```
aws iotsitewise list-associated-assets \  
  --asset-id asset-id \  
  --hierarchy-id hierarchy-id
```

Per maggiori informazioni, vedi l'operazione [ListAssociatedAssets](#).

3. Esegui il comando seguente per eliminare ogni asset associato e quindi eliminare l'asset. *asset-id* Sostituiscilo con l'ID o l'ID esterno della risorsa.

```
aws iotsitewise delete-asset --asset-id asset-id
```

Eliminare i modelli di asset in AWS IoT SiteWise

È possibile utilizzare la AWS IoT SiteWise console o l'API per eliminare un modello di asset.

Prima di poter eliminare un modello di asset, è necessario eliminare tutte le risorse che sono state create dal modello di asset.

Quando elimini un modello di asset, lo stato del modello di asset è DELETING fino a quando le modifiche non vengono propagate. Per ulteriori informazioni, consulta [Stati di asset e modelli](#). Non è possibile eseguire query su un modello di asset eliminato. Se lo facessi, l'API restituirebbe una risposta HTTP 404.

Argomenti

- [Eliminare un modello di asset \(console\)](#)
- [Eliminare un modello di asset \(AWS CLI\)](#)

Eliminare un modello di asset (console)

È possibile utilizzare la AWS IoT SiteWise console per eliminare un modello di asset.

Per eliminare un modello di asset (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione selezionare Models (Modelli).
3. Scegli il modello di asset da eliminare.
4. Se il modello include asset, elimina ogni asset. Scegli il nome di un asset per aprire la relativa pagina in cui è possibile eliminarlo. Per ulteriori informazioni, consulta [Eliminare una risorsa \(console\)](#).
5. Nella pagina del modello, scegli Elimina.
6. Nella finestra di dialogo Elimina modello, effettuate le seguenti operazioni:
 - a. Immetti **Delete** per confermare l'eliminazione.
 - b. Scegliere Delete (Elimina).

Eliminare un modello di asset (AWS CLI)

È possibile utilizzare AWS Command Line Interface (AWS CLI) per eliminare un modello di asset.

Utilizzate l'[DeleteAssetModel](#) operazione per eliminare un modello di asset. Specifica il parametro seguente:

- `assetModelId`— L'ID della risorsa. Questo è l'ID effettivo in formato UUID, o `externalId:myExternalId` se ne ha uno. Per ulteriori informazioni, consulta [Oggetti di riferimento con oggetti esterni IDs](#) nella Guida per l'utente di AWS IoT SiteWise .

Per eliminare un modello di asset (AWS CLI)

1. Esegui il comando seguente per elencare tutti gli asset creati dal modello. `asset-model-id` Sostituilo con l'ID o l'ID esterno del modello di asset.

```
aws iotsitewise list-assets --asset-model-id asset-model-id
```

Per maggiori informazioni, vedi l'operazione [ListAssets](#).

2. Se il comando precedente restituisce asset dal modello, elimina ogni asset. Per ulteriori informazioni, consulta [Eliminare una risorsa \(AWS CLI\)](#).
3. Esegui il comando seguente per eliminare il modello di asset. `asset-model-id` Sostituiscilo con l'ID o l'ID esterno del modello di asset.

```
aws iotsitewise delete-asset-model --asset-model-id asset-model-id
```

Important

Per evitare di eliminare un modello di asset aggiornato contemporaneamente dall'ultima operazione di lettura, è necessario definire una richiesta di eliminazione condizionale. Consultare [Blocco ottimistico per le scritture dei modelli di asset](#).

Operazioni in blocco con asset e modelli

Per lavorare con un gran numero di risorse o modelli di asset, utilizzate le operazioni in blocco per importare ed esportare in blocco le risorse in una posizione diversa. Ad esempio, puoi creare un file di dati che definisce asset o modelli di asset in un bucket Amazon S3 e utilizzare l'importazione in blocco per crearli o aggiornarli. AWS IoT SiteWise In alternativa, se disponi di un gran numero di asset o modelli di asset AWS IoT SiteWise, puoi esportarli in Amazon S3.

Note

Puoi eseguire operazioni in blocco AWS IoT SiteWise richiamando le operazioni nell' AWS IoT TwinMaker API. Puoi farlo senza configurare AWS IoT TwinMaker o creare un' AWS IoT TwinMaker area di lavoro. Tutto ciò di cui hai bisogno è un bucket Amazon S3 in cui inserire i tuoi contenuti. AWS IoT SiteWise

Argomenti

- [Concetti e terminologia chiave](#)
- [Funzionalità supportate](#)
- [Prerequisiti per le operazioni in blocco](#)
- [Eseguire un processo di importazione in blocco](#)

- [Esegui un processo di esportazione in blocco](#)
- [Monitoraggio dell'avanzamento dei lavori e gestione degli errori](#)
- [Importa esempi di metadati](#)
- [Esporta esempi di metadati](#)
- [AWS IoT SiteWise schema del processo di trasferimento dei metadati](#)

Concetti e terminologia chiave

AWS IoT SiteWise le funzionalità di importazione ed esportazione in blocco si basano sui seguenti concetti e terminologia:

- **Importazione:** l'azione di spostare risorse o modelli di asset da un file in un bucket Amazon S3 a AWS IoT SiteWise
- **Esportazione:** l'azione di spostare risorse o modelli di asset AWS IoT SiteWise da un bucket Amazon S3.
- **Fonte:** la posizione di partenza da cui desideri spostare i contenuti.

Ad esempio, un bucket Amazon S3 è una fonte di importazione ed AWS IoT SiteWise è una fonte di esportazione.

- **Destinazione:** la posizione desiderata in cui desideri spostare i tuoi contenuti.

Ad esempio, un bucket Amazon S3 è una destinazione di esportazione ed AWS IoT SiteWise è una destinazione di importazione.

- **AWS IoT SiteWise Schema:** questo schema viene utilizzato per importare ed esportare metadati da AWS IoT SiteWise
- **Risorsa di primo livello:** una AWS IoT SiteWise risorsa che è possibile creare o aggiornare singolarmente, ad esempio una risorsa o un modello di risorsa.
- **Risorsa secondaria:** una risorsa annidata all'interno di una AWS IoT SiteWise risorsa di primo livello. Gli esempi includono proprietà, gerarchie e modelli compositi.
- **Metadati:** informazioni chiave necessarie per importare o esportare correttamente le risorse. Esempi di metadati sono le definizioni degli asset e i modelli di asset.
- **metadataTransferJob:** L'oggetto creato durante l'esecuzione `CreateMetadataTransferJob`.

Funzionalità supportate

Questo argomento spiega cosa è possibile fare quando si esegue un'operazione in blocco. Le operazioni in blocco supportano le seguenti funzionalità:

- Creazione di risorse di primo livello: quando importate una risorsa o un modello di asset che non definisce un ID o il cui ID non corrisponde a quello di uno esistente, verrà creato come nuova risorsa.
- Sostituzione di risorse di primo livello: quando importate una risorsa o un modello di risorsa il cui ID corrisponde a uno già esistente, sostituirà la risorsa esistente.
- Creazione, sostituzione o eliminazione di sottomodelli: quando l'importazione sostituisce una risorsa di primo livello come una risorsa o un modello di risorsa, la nuova definizione sostituisce tutte le risorse secondarie, come proprietà, gerarchie o modelli compositi.

Ad esempio, se aggiorni un modello di asset durante un'importazione in blocco e la versione aggiornata definisce una proprietà che non era presente nell'originale, viene creata una nuova proprietà. Se definisce una proprietà già esistente, la proprietà esistente verrà aggiornata. Se il modello di asset aggiornato omette una proprietà che era presente nell'originale, la proprietà viene eliminata.

- Nessuna eliminazione di risorse di primo livello: le operazioni in blocco non eliminano una risorsa o un modello di asset. Le operazioni in blocco si limitano a crearli o aggiornarli.

Prerequisiti per le operazioni in blocco

Questa sezione spiega i prerequisiti delle operazioni in blocco, incluse le autorizzazioni AWS Identity and Access Management (IAM) per lo scambio di risorse tra i AWS servizi e il computer locale. Prima di iniziare un'operazione in blocco, completa il seguente prerequisito:

- Crea un bucket Amazon S3 per archiviare le risorse. Per ulteriori informazioni sull'uso di Amazon S3, consulta [Che cos'è Amazon S3?](#)

Autorizzazioni IAM

Per eseguire operazioni in blocco, è necessario creare una policy AWS Identity and Access Management (IAM) con autorizzazioni che consentano lo scambio di AWS risorse tra Amazon S3 e il AWS IoT SiteWise computer locale. Per ulteriori informazioni sulla creazione di policy IAM, consulta la sezione relativa alla [creazione delle policy IAM](#).

Per eseguire operazioni in blocco, sono necessarie le seguenti politiche.

AWS IoT SiteWise politica

Questa policy consente l'accesso alle azioni AWS IoT SiteWise API richieste per le operazioni in blocco:

```
{
  "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 TwinMaker politica

Questa politica consente l'accesso alle operazioni AWS IoT TwinMaker API utilizzate per lavorare con operazioni di massa:

```
{
  "Sid": "MetadataTransferJobApiAccess",
  "Effect": "Allow",
  "Action": [
    "iottwinmaker:CreateMetadataTransferJob",
    "iottwinmaker:CancelMetadataTransferJob",
    "iottwinmaker:GetMetadataTransferJob",
    "iottwinmaker:ListMetadataTransferJobs"
  ],
  "Resource": "*"
}
```

Politica di Amazon S3

Questa policy fornisce l'accesso ai bucket Amazon S3 per il trasferimento di metadati per operazioni di massa.

For a specific Amazon S3 bucket

Se utilizzi un bucket specifico per lavorare con i metadati delle tue operazioni di massa, questa policy fornisce l'accesso a quel 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

Se utilizzerai molti bucket diversi per lavorare con i metadati delle operazioni di massa, questa policy fornisce l'accesso a qualsiasi bucket:

```
{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject",
    "s3:GetObject",
    "s3:GetBucketLocation",
    "s3:ListBucket",
    "s3:AbortMultipartUpload",
    "s3:ListBucketMultipartUploads",
    "s3:ListMultipartUploadParts"
  ],
  "Resource": "*"
}
```

Per informazioni sulla risoluzione dei problemi relativi alle operazioni di importazione ed esportazione, consulta [Risolvi i problemi di importazione ed esportazione in blocco](#)

Eseguire un processo di importazione in blocco

L'importazione in blocco è l'azione di spostare i metadati in un'area di lavoro. AWS IoT SiteWise Ad esempio, l'importazione in blocco può spostare i metadati da un file locale o da un file in un bucket Amazon S3 a un'area di lavoro. AWS IoT SiteWise

Passaggio 1: preparare il file da importare

Scaricate il file in formato AWS IoT SiteWise nativo per importare gli asset e i relativi modelli. Per ulteriori dettagli, consulta [AWS IoT SiteWise schema del processo di trasferimento dei metadati](#).

Passaggio 2: carica il file preparato su Amazon S3

Carica il file su Amazon S3. Per ulteriori informazioni, consulta [Caricamento di un file su Amazon S3](#) nella Guida per l'utente di Amazon Simple Storage Service.

Importa metadati (console)

Puoi utilizzare il Console AWS IoT SiteWise per importare in blocco i metadati. Segui [Passaggio 1: preparare il file da importare](#) e [Passaggio 2: carica il file preparato su Amazon S3](#) prepara un file pronto per essere importato.

Importa dati da Amazon S3 a Console AWS IoT SiteWise

1. Passare alla [console AWS IoT SiteWise](#).
2. Scegli Bulk operations New dal pannello di navigazione.
3. Scegli Nuova importazione per avviare il processo di importazione.
4. Nella pagina Importa metadati:
 - Scegli Browse Amazon S3 per visualizzare il bucket e i file Amazon S3.
 - Passa al bucket Amazon S3 che contiene il file di importazione preparato.
 - Seleziona il file da importare.
 - Esamine il file selezionato e scegliete Importa.
5. La pagina Operazioni in blocco sui SiteWise metadati di Console AWS IoT SiteWise mostra il processo di importazione appena creato nella tabella di avanzamento dei lavori.

Importa metadati (AWS CLI)

Per eseguire un'azione di importazione, utilizzate la procedura seguente:

Importa dati da Amazon S3 a AWS CLI

1. Crea un file di metadati che specifichi le risorse che desideri importare, seguendo il [AWS IoT SiteWise schema del processo di trasferimento dei metadati](#) Archivia questo file nel tuo bucket Amazon S3.

Per esempi di file di metadati da importare, consulta. [Importa esempi di metadati](#)

2. Ora create un file JSON con il corpo della richiesta. Il corpo della richiesta specifica l'origine e la destinazione del processo di trasferimento. Questo file è separato dal file del passaggio precedente. Assicurati di specificare il tuo bucket Amazon S3 come origine e iotsitewise come destinazione.

L'esempio seguente mostra il corpo della richiesta:

```
{
  "metadataTransferJobId": "your-transfer-job-Id",
  "sources": [{
    "type": "s3",
    "s3Configuration": {
      "location": "arn:aws:s3:::amzn-s3-demo-bucket/
your_import_metadata.json"
    }
  ]},
  "destination": {
    "type": "iotsitewise"
  }
}
```

3. Invoca il `CreateMetadataTransferJob` eseguendo il AWS CLI comando seguente. In questo esempio, viene denominato `createMetadataTransferJobExport.json` il file del corpo della richiesta del passaggio precedente.

```
aws iottwinmaker create-metadata-transfer-job --region us-east-1 \
  --cli-input-json file://createMetadataTransferJobImport.json
```

Questo creerà un processo di trasferimento dei metadati e inizierà il processo di trasferimento delle risorse selezionate.

Esegui un processo di esportazione in blocco

L'esportazione in blocco è l'azione di spostare i metadati da un' AWS IoT SiteWise area di lavoro a un bucket Amazon S3.

Quando esegui un'esportazione in blocco dei tuoi AWS IoT SiteWise contenuti su Amazon S3, puoi specificare filtri per limitare i modelli e gli asset specifici che desideri esportare.

I filtri devono essere specificati in una `iotSiteWiseConfiguration` sezione all'interno della sezione delle fonti della richiesta JSON.

Note

Puoi includere più filtri nella tua richiesta. L'operazione in blocco esporterà modelli di asset e asset che corrispondono a uno qualsiasi dei filtri.

Se non fornite alcun filtro, l'operazione in blocco esporta tutti i modelli e gli asset degli asset.

Example corpo della richiesta con filtri

```
{
  "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"
    }
  }
}
```


Esporta metadati (console)

La procedura seguente spiega l'azione di esportazione della console:

Creare un processo di esportazione in Console AWS IoT SiteWise

1. Passare alla [console AWS IoT SiteWise](#).
2. Scegli Operazioni in blocco Nuove dal pannello di navigazione.
3. Scegli Nuova esportazione per avviare il processo di esportazione.
4. Nella pagina Esporta metadati:
 - Immettete un nome per il processo di esportazione. Questo è il nome usato per il file esportato nel tuo bucket Amazon S3.
 - Scegliete le risorse da esportare, che imposta i filtri per il lavoro:
 - Esporta tutti gli asset e i modelli di asset. Utilizza i filtri sugli asset e sui modelli di asset.
 - Esporta risorse. Filtra in base alle tue risorse.
 - Seleziona la risorsa da utilizzare per il filtro di esportazione.
 - (Facoltativo) Aggiungete la progenie o il modello di asset associato.
 - Esporta modelli di asset. Filtra i tuoi modelli di asset.
 - Seleziona il modello di asset da utilizzare per il filtro di esportazione.
 - (Facoltativo) Aggiungi la progenie, l'asset associato o entrambi.
 - Scegli Next (Successivo).
 - Passa al bucket Amazon S3:
 - Scegli Browse Amazon S3 per visualizzare il bucket e i file Amazon S3.
 - Passa al bucket Amazon S3 in cui deve essere inserito il file.
 - Scegli Next (Successivo).
 - Controlla il processo di esportazione e scegli Esporta.
5. La pagina Operazioni in blocco sui SiteWise metadati di Console AWS IoT SiteWise mostra il processo di importazione appena creato nella tabella di avanzamento dei lavori.

Per i diversi modi di utilizzare i filtri durante l'esportazione dei metadati, consulta. [Esporta esempi di metadati](#)

Esporta metadati ()AWS CLI

La procedura seguente spiega l'azione di AWS CLI esportazione:

Esportazione di dati AWS IoT SiteWise da Amazon S3

1. Crea un file JSON con il corpo della richiesta. Il corpo della richiesta specifica l'origine e la destinazione del processo di trasferimento. L'esempio seguente mostra un esempio di corpo della richiesta:

```
{
  "metadataTransferJobId": "your-transfer-job-Id",
  "sources": [{
    "type": "iotsitewise"
  }],
  "destination": {
    "type": "s3",
    "s3Configuration": {
      "location": "arn:aws:s3:::amzn-s3-demo-bucket"
    }
  }
}
```

Assicurati di specificare il tuo bucket Amazon S3 come destinazione del processo di trasferimento dei metadati.

Note

Questo esempio esporterà tutti i tuoi modelli e asset di asset. Per limitare l'esportazione a modelli o asset specifici, potete includere filtri nel corpo della richiesta. Per ulteriori informazioni sull'applicazione dei filtri di esportazione, consulta [Esporta esempi di metadati](#).

2. Salva il file del corpo della richiesta da utilizzare nel passaggio successivo. In questo esempio, il file è denominato `createMetadataTransferJobExport.json`.
3. Invoca il `CreateMetadataTransferJob` eseguendo il seguente AWS CLI comando:

```
aws iottwinmaker create-metadata-transfer-job --region us-east-1 \
```

```
--cli-input-json file://createMetadataTransferJobExport.json
```

Sostituisci il file JSON di input `createMetadataTransferJobExport.json` con il tuo nome del file di trasferimento.

Monitoraggio dell'avanzamento dei lavori e gestione degli errori

L'elaborazione di un processo in blocco richiede tempo. Ogni processo viene elaborato nell'ordine di AWS IoT SiteWise ricezione della richiesta. Viene elaborato one-at-a-time per ogni account. Quando un lavoro viene completato, il successivo in coda avvia automaticamente l'elaborazione. AWS IoT SiteWise risolve i job in modo asincrono e aggiorna lo stato di ciascuno man mano che procede. Ogni processo ha un campo di stato che contiene lo stato della risorsa e un messaggio di errore, se applicabile.

Lo stato può avere uno dei seguenti valori:

- **VALIDATING**— Convalida del lavoro, incluso il formato del file inviato e il relativo contenuto.
- **PENDING**— Il lavoro è in coda. È possibile annullare i lavori in questo stato dalla AWS IoT SiteWise console, ma tutti gli altri stati continueranno fino alla fine.
- **RUNNING**— Elaborazione del lavoro. Sta creando e aggiornando risorse come definito dal file di importazione o esportando risorse in base ai filtri del processo di esportazione scelti. Se annullata, qualsiasi risorsa importata da questo lavoro non viene eliminata. Per ulteriori informazioni, consulta [Rivedi lo stato di avanzamento e i dettagli del lavoro \(console\)](#).
- **CANCELLING**— Il lavoro viene annullato attivamente.
- **ERROR**— Una o più risorse non sono state elaborate. Consulta il rapporto dettagliato sul lavoro per ulteriori informazioni. Per ulteriori informazioni, consulta [Controlla i dettagli dell'errore \(console\)](#).
- **COMPLETED**— Job completato senza errori.
- **CANCELLED**— Il lavoro viene annullato e non è in coda. Se hai annullato un **RUNNING** lavoro, le risorse già importate da questo lavoro al momento dell'annullamento non vengono eliminate da AWS IoT SiteWise.

Argomenti

- [Monitoraggio dei progressi dei lavori](#)
- [Ispeziona gli errori per AWS IoT SiteWise](#)

Monitoraggio dei progressi dei lavori

Rivedi lo stato di avanzamento e i dettagli del lavoro (console)

Visualizza [Importa metadati \(console\)](#) o [Esporta metadati \(console\)](#) avvia un lavoro collettivo.

Panoramica dello stato di avanzamento del lavoro nella AWS IoT SiteWise console:

1. Passare alla [console AWS IoT SiteWise](#).
2. Scegli Operazioni in blocco Nuove dal pannello di navigazione.
3. La tabella di avanzamento dei lavori nella AWS IoT SiteWise console mostra l'elenco dei lavori relativi alle operazioni in blocco.
4. La colonna Job type descrive se si tratta di un processo di esportazione o importazione. Le colonne Data di importazione mostrano la data di inizio del processo.
5. La colonna Stato mostra lo stato del lavoro. È possibile selezionare un lavoro per visualizzarne i dettagli.
6. Il lavoro selezionato mostra Successo in caso di successo o un elenco di errori se il lavoro non è riuscito. Per ogni tipo di risorsa viene inoltre visualizzata una descrizione dell'errore.

Panoramica dei dettagli del lavoro nella AWS IoT SiteWise console:

La tabella di avanzamento dei lavori nella AWS IoT SiteWise console mostra l'elenco dei lavori eseguiti in blocco.

1. Scegli un lavoro per visualizzare maggiori dettagli.
2. Per un processo di importazione, Data source ARN rappresenta la posizione Amazon S3 del file di importazione.
3. Per un processo di esportazione, Data destination ARN rappresenta la posizione Amazon S3 del file dopo l'esportazione.
4. Inoltre StatusStatus reason, fornisci ulteriori dettagli sul lavoro corrente. Per ulteriori dettagli, consulta [Monitoraggio dell'avanzamento dei lavori e gestione degli errori](#).
5. Queued positionRappresenta la posizione del lavoro nella coda del processo. I lavori vengono elaborati uno alla volta. Una posizione in coda pari a 1 indica che il lavoro verrà elaborato successivamente.
6. La pagina dei dettagli dei lavori mostra anche i conteggi relativi allo stato di avanzamento dei lavori.

- I tipi di conteggio dell'avanzamento dei lavori sono:
 - i. `Total resources`— Indica il numero totale di asset nel processo di trasferimento.
 - ii. `Succeeded`— Indica il numero di asset trasferiti con successo durante il processo.
 - iii. `Failed`— Indica il numero di asset che hanno avuto esito negativo durante il processo.
 - iv. `Skipped`— Indica il numero di risorse che sono state ignorate durante il processo.
- 7. Uno stato del lavoro pari a `PENDING` o `VALIDATING`, visualizza tutti i conteggi relativi all'avanzamento dei lavori. – Ciò indica che i conteggi relativi all'avanzamento dei lavori sono in fase di valutazione.
- 8. Uno stato del lavoro pari a `RUNNING` mostra il `Total resources` conteggio, il lavoro inviato per l'elaborazione. I conteggi dettagliati (`SucceededFailed`, e `Skipped`) si applicano alle risorse elaborate. La somma dei conteggi dettagliati è inferiore al `Total resources` conteggio, finché lo stato del lavoro non è `COMPLETED` o `ERROR`.
- 9. Se lo stato di un lavoro è `COMPLETED` o `ERROR`, il `Total resources` conteggio è uguale alla somma dei conteggi dettagliati (`SucceededFailed`, e). `Skipped`
- 10. Se lo stato di un lavoro è `ERROR`, consulta la tabella `Job failures` per i dettagli sugli errori e gli errori specifici. Per ulteriori dettagli, consulta [Controlla i dettagli dell'errore \(console\)](#).

Rivedi lo stato di avanzamento e i dettagli del lavoro (AWS CLI)

Dopo aver avviato un'operazione in blocco, puoi verificarne o aggiornarne lo stato utilizzando le seguenti azioni API:

- Per recuperare informazioni su un lavoro specifico, utilizza l'azione [GetMetadataTransferJobAPI](#).

Recupera informazioni con l'**GetMetadataTransferJobAPI**:

1. Crea ed esegui un processo di trasferimento. Chiamata dell'API `GetMetadataTransferJob`.

Example AWS CLI comando:

```
aws iottwinmaker get-metadata-transfer-job \  
  --metadata-transfer-job-id your_metadata_transfer_job_id \  
  --region your_region
```

2. L'GetMetadataTransferJobAPI restituisce un MetadataTransferJobProgress oggetto con i seguenti parametri:

- succeededCount: indica il numero di risorse trasferite correttamente durante il processo.
- FailedCount: indica il numero di asset che hanno avuto esito negativo durante il processo.
- SkippedCount: indica il conteggio degli asset che sono stati ignorati durante il processo.
- totalCount: indica il conteggio totale degli asset nel processo di trasferimento.

Questi parametri indicano lo stato di avanzamento del lavoro. Se lo stato èRUNNING, aiutano a tenere traccia del numero di risorse ancora da elaborare.

Se si verificano errori di convalida dello schema o se failedCount è maggiore o uguale a 1, lo stato di avanzamento del lavoro diventa. ERROR Un report di errore completo per il processo viene inserito nel tuo bucket Amazon S3. Per ulteriori dettagli, consulta [Ispeziona gli errori per AWS IoT SiteWise](#).

- Per elencare i lavori correnti, utilizza l'azione [ListMetadataTransferJobsAPI](#).

Utilizza un file JSON per filtrare i lavori restituiti in base al loro stato corrente. Consultate la procedura seguente:

1. Per specificare i filtri che desideri utilizzare, crea un file JSON AWS CLI di input. Vuoi usare:

```
{
  "sourceType": "s3",
  "destinationType": "iottwinmaker",
  "filters": [{
    "state": "COMPLETED"
  }]
}
```

Per un elenco di state valori validi, consulta la Guida [ListMetadataTransferJobsFilter](#) di riferimento dell'AWS IoT TwinMaker API.

2. Utilizzate il file JSON come argomento nel seguente comando di AWS CLI esempio:

```
aws iottwinmaker list-metadata-transfer-job --region your_region \
  --cli-input-json file://ListMetadataTransferJobsExample.json
```

- Per annullare un lavoro, utilizza l'azione [CancelMetadataTransferJob](#) API. Questa API annulla lo specifico processo di trasferimento dei metadati, senza influire sulle risorse già esportate o importate:

```
aws iottwinmaker cancel-metadata-transfer-job \  
  --region your_region \  
  --metadata-transfer-job-id job-to-cancel-id
```

Ispeziona gli errori per AWS IoT SiteWise

Controlla i dettagli dell'errore (console)

I dettagli dell'errore nella AWS IoT SiteWise console:

1. Passare alla [console AWS IoT SiteWise](#).
2. Consulta la tabella di avanzamento dei lavori in blocco Console AWS IoT SiteWise per un elenco dei lavori eseguiti in blocco.
3. Seleziona un lavoro per visualizzarne i dettagli.
4. Se lo stato di un lavoro è COMPLETED o ERROR, il `Total resources` conteggio è uguale alla somma dei conteggi dettagliati (SucceededFailed, eSkipped).
5. Se lo stato di un lavoro è ERROR, consulta la tabella Job failures per i dettagli sugli errori e gli errori specifici.
6. La tabella Job failures mostra il contenuto del rapporto Job. Il `Resource type` campo indica la posizione dell'errore o degli errori, ad esempio:
 - Ad esempio, un errore di convalida nel `Resource type` campo indica che il modello di importazione e il formato del file dello schema dei metadati non corrispondono. `Bulk operations template` Per ulteriori informazioni, consulta [AWS IoT SiteWise schema del processo di trasferimento dei metadati](#).
 - Un errore Asset nel `Resource type` campo indica che la risorsa non è stata creata a causa di un conflitto con un'altra risorsa. Per informazioni sugli [errori](#) e i conflitti relativi alle AWS IoT SiteWise risorse, consultate Errori comuni.

Controlla i dettagli dell'errore ()AWS CLI

Per gestire e diagnosticare gli errori prodotti durante un processo di trasferimento, vedete la seguente procedura sull'utilizzo dell'azione `GetMetadataTransferJob` API:

1. Dopo aver creato ed eseguito un processo di trasferimento, chiama [GetMetadataTransferJob](#):

```
aws iottwinmaker get-metadata-transfer-job \  
    --metadata-transfer-job-id your_metadata_transfer_job_id \  
    --region us-east-1
```

2. Una volta visualizzato lo stato del lavoro a cui rivolgiti `COMPLETED`, puoi iniziare a verificare i risultati del lavoro.
3. Quando si chiama `GetMetadataTransferJob`, restituisce un oggetto chiamato [MetadataTransferJobProgress](#).

L' `MetadataTransferJobProgress` oggetto contiene i seguenti parametri:

- `FailedCount`: indica il numero di asset che hanno avuto esito negativo durante il processo di trasferimento.
 - `SkippedCount`: indica il numero di asset che sono stati ignorati durante il processo di trasferimento.
 - `suceededCount`: indica il numero di asset che hanno avuto successo durante il processo di trasferimento.
 - `totalCount`: indica il numero totale di asset coinvolti nel processo di trasferimento.
4. Inoltre, la chiamata API restituisce un elemento `reportUrl` che contiene un URL predefinito. Se la tua operazione di trasferimento presenta problemi che devi approfondire, visita questo URL.

Importa esempi di metadati

Questa sezione mostra come creare file di metadati per importare modelli di asset e asset con un'unica operazione di importazione in blocco.

Esempio di importazione in blocco

È possibile importare molti modelli di asset e asset con un'unica operazione di importazione in blocco. L'esempio seguente mostra come creare un file di metadati a tale scopo.

In questo scenario di esempio, sono presenti diversi siti di lavoro che contengono robot industriali nelle celle di lavoro.

L'esempio definisce due modelli di asset:

- **RobotModel1**: Questo modello di asset rappresenta un particolare tipo di robot che avete nei vostri cantieri. Il robot ha una proprietà di misurazione, **Temperature**.
- **WorkCell1**: Questo modello di asset rappresenta una raccolta di robot all'interno di uno dei vostri siti di lavoro. Il modello di asset definisce una gerarchia **robotHierarchyOEM1**, per rappresentare la relazione tra robot e celle di lavoro.

L'esempio definisce anche alcune risorse:

- **WorkCell1**: una cella di lavoro all'interno del sito di Boston
- **RobotArm123456**: un robot all'interno di quella cella di lavoro
- **RobotArm987654**: un altro robot all'interno di quella cella di lavoro

Il seguente file di metadati JSON definisce questi modelli e asset. L'esecuzione di un'importazione in blocco con questi metadati crea i modelli di asset e le risorse al loro interno AWS IoT SiteWise, comprese le relative relazioni gerarchiche.

File di metadati per l'importazione

```
{
  "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": "workCellHierarchyWithOEM1Robot",
        "childAssetExternalId": "Robot.OEM1.3536.123456"
      },
      {
        "externalId": "workCellHierarchyWithOEM1Robot",
        "childAssetExternalId": "Robot.OEM1.3536.987654"
      }
    ]
  }
]

```

```
}  
  ]  
}
```

Esempio di onboarding iniziale di modelli e asset

In questo scenario di esempio, in un'azienda sono presenti diversi siti di lavoro che contengono robot industriali.

L'esempio definisce più modelli di asset:

- **Sample_Enterprise**— Questo modello di asset rappresenta la società di cui fanno parte i siti. Il modello di asset definisce una gerarchia per rappresentare la relazione tra i siti e l'azienda.
`Enterprise to Site`
- **Sample_Site**— Questo modello di asset rappresenta i siti di produzione all'interno dell'azienda. Il modello di asset definisce una gerarchia `Site to Line`, per rappresentare la relazione tra le linee e il sito.
- **Sample_Welding Line**— Questo modello di asset rappresenta una linea di assemblaggio all'interno dei siti di lavoro. Il modello di asset definisce una gerarchia `Line to Robot`, per rappresentare la relazione tra i robot e la linea.
- **Sample_Welding Robot**— Questo modello di asset rappresenta un particolare tipo di robot nei vostri cantieri.

L'esempio definisce anche gli asset in base ai modelli di asset.

- **Sample_AnyCompany Motor**— Questa risorsa viene creata a partire da un modello di `Sample_Enterprise` asset.
- **Sample_Chicago**— Questa risorsa viene creata a partire da un modello di `Sample_Site` asset.
- **Sample_Welding Line 1**— Questa risorsa viene creata a partire da un modello di `Sample_Welding Line` asset.
- **Sample_Welding Robot 1**— Questa risorsa viene creata a partire da un modello di `Sample_Welding Robot` asset.
- **Sample_Welding Robot 2**— Questa risorsa viene creata a partire da un modello di `Sample_Welding Robot` asset.

Il seguente file di metadati JSON definisce questi modelli e risorse di asset. L'esecuzione di un'importazione in blocco con questi metadati crea i modelli di asset e le risorse al loro interno AWS IoT SiteWise, comprese le relative relazioni gerarchiche.

File JSON per integrare risorse e modelli da importare

```
{
  "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"
      }
    ]
  }
]

```



```
}
    ]
  }
]
}
```

La schermata seguente mostra i modelli visualizzati nell'esempio di codice precedente Console AWS IoT SiteWise dopo l'esecuzione del precedente esempio di codice.

IoT SiteWise > Models

Models (4)

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 instances

< 1 >

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...

La schermata seguente mostra i modelli, gli asset e le gerarchie visualizzati Console AWS IoT SiteWise dopo l'esecuzione del precedente esempio di codice.

IoT SiteWise > Assets

Assets (1)

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_AnyEnterprise Motor		ACTIVE	November 10, 2023 at 11:23:06 (UTC-5:00)	November 10, 2023 at 11:23:06 (UTC-...
Sample_Chicago		ACTIVE	November 10, 2023 at 11:22:57 (UTC-5:00)	November 10, 2023 at 11:22:57 (UTC-...
Sample_Welding Line 1		ACTIVE	November 10, 2023 at 11:22:48 (UTC-5:00)	November 10, 2023 at 11:22:48 (UTC-...
Sample_Welding Robot 1		ACTIVE	November 10, 2023 at 11:22:39 (UTC-5:00)	November 10, 2023 at 11:22:39 (UTC-...
Sample_Welding Robot 2		ACTIVE	November 10, 2023 at 11:22:30 (UTC-5:00)	November 10, 2023 at 11:22:30 (UTC-...

Esempio di onboarding di risorse aggiuntive

Questo esempio definisce risorse aggiuntive da importare in un modello di asset esistente nel tuo account:

- Sample_Welding Line 2— Questa risorsa viene creata a partire da un modello di Sample_Welding Line asset.
- Sample_Welding Robot 3— Questa risorsa viene creata a partire da un modello di Sample_Welding Robot asset.
- Sample_Welding Robot 4— Questa risorsa viene creata a partire da un modello di Sample_Welding Robot asset.

Per creare le risorse iniziali per questo esempio, vedere [Esempio di onboarding iniziale di modelli e asset](#).

Il seguente file di metadati JSON definisce questi modelli e asset di asset. L'esecuzione di un'importazione in blocco con questi metadati crea i modelli di asset e le risorse al loro interno AWS IoT SiteWise, comprese le relative relazioni gerarchiche.

File JSON per l'onboarding di risorse aggiuntive

```
{
  "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"
        }
      ]
    }
  ]
}

```

La schermata seguente mostra modelli, risorse e gerarchie visualizzati Console AWS IoT SiteWise dopo l'esecuzione del precedente esempio di codice.

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_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)

Esempio di onboarding di nuove proprietà

Questo esempio definisce nuove proprietà sui modelli di asset esistenti. Guarda [Esempio di onboarding di risorse aggiuntive](#) come integrare risorse e modelli aggiuntivi.

- **Joint 1 Temperature**— Questa proprietà viene aggiunta al modello di `Sample_Welding Robot` asset. Questa nuova proprietà si propagherà anche a ogni risorsa creata dal modello di `Sample_Welding Robot` asset.

Per aggiungere una nuova proprietà a un modello di asset esistente, vedete il seguente esempio di file di metadati JSON. Come mostrato in JSON, l'intera definizione del modello di `Sample_Welding Robot` asset esistente deve essere fornita insieme alla nuova proprietà. Se non viene fornito l'intero elenco di proprietà della definizione esistente, AWS IoT SiteWise elimina le proprietà omesse.

File JSON per incorporare nuove proprietà

Questo esempio aggiunge una nuova proprietà `Joint 1 Temperature` al modello di asset.

```
{
  "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"
    }
]
}
]

```

```
}
```

Esempio di gestione dei flussi di dati

Questo esempio mostra due modi di gestire i flussi di dati associati a una proprietà di un asset. Quando si rinomina un alias di proprietà della risorsa, sono disponibili due opzioni per i dati storici attualmente archiviati nel flusso di dati della proprietà della risorsa.

- Opzione uno: mantiene il flusso di dati corrente e rinomina solo l'alias, in modo che i dati storici siano accessibili con il nuovo alias.

Nell'esempio del file di metadati JSON, la proprietà asset con ID `External_Id_Welding_Robot_Cycle_Count` cambia il suo alias in `AnyCompany/Chicago/Welding Line/S3000/Count-Updated`. I dati storici per questa proprietà dell'asset rimangono gli stessi dopo questa modifica.

- Opzione due: assegna un nuovo flusso di dati alla proprietà dell'asset, accessibile con il nuovo alias. Il vecchio flusso di dati e i relativi dati storici sono ancora accessibili con il vecchio alias, ma non sono associati ad alcuna proprietà dell'asset.

Nell'esempio del file di metadati JSON, la proprietà dell'asset con ID `External_Id_Welding_Robot_Joint_1_Current` cambia il suo alias in `AnyCompany/Chicago/Welding Line/S4999/1/Current`. Questa volta il valore aggiuntivo `retainDataOnAliasChange` è presente e impostato su `False`. Con questa impostazione, il flusso di dati originale viene dissociato dalla proprietà dell'asset e viene creato un nuovo flusso di dati senza dati storici.

Per accedere al vecchio flusso di dati con i dati storici originali, in AWS Console Home, vai alla pagina Data Streams e cerca il vecchio alias `AnyCompany/Chicago/Welding Line/S3000/1/Current`.

File JSON per aggiornare gli alias delle proprietà

```
{
  "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"
    }
]
}

```

Esporta esempi di metadati

Quando esegui un'esportazione in blocco dei tuoi AWS IoT SiteWise contenuti su Amazon S3, puoi specificare filtri per limitare i modelli e gli asset specifici che desideri esportare.

Specificate i filtri in una `iotSiteWiseConfiguration` sezione all'interno della `sources` sezione del corpo della richiesta.

Note

Puoi includere più filtri. L'operazione in blocco esporterà qualsiasi modello di asset o risorsa che corrisponde a uno qualsiasi dei filtri.

Se non fornite alcun filtro, l'operazione esporterà tutti i modelli e gli asset degli asset.

```

{
  "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"  
    }  
  }  
}
```

Filtra per modello di asset

Puoi filtrare un modello di asset specifico. Puoi anche includere tutti gli asset che utilizzano quel modello o tutti i modelli di asset all'interno della sua gerarchia. Non puoi includere sia gli asset che la gerarchia.

Per ulteriori informazioni sulle gerarchie, consultare [Definite le gerarchie dei modelli di asset](#).

Asset model

Questo filtro include il modello di asset specificato:

```
"filterByAssetModel": {  
  "assetModelId": "asset model ID"  
}
```

Asset model and its assets

Questo filtro include il modello di asset specificato, insieme a tutti gli asset che utilizzano quel modello di asset:

```
"filterByAssetModel": {  
  "assetModelId": "asset model ID",  
  "includeAssets": true  
}
```

Asset model and its hierarchy

Questo filtro include il modello di asset specificato, insieme a tutti i modelli di asset associati nella sua gerarchia:

```
"filterByAssetModel": {  
  "assetModelId": "asset model ID",  
  "includeOffspring": true  
}
```

Filtra per risorsa

Puoi filtrare una risorsa specifica. Puoi anche includere il suo modello di asset o tutte le risorse associate all'interno della sua gerarchia. Non è possibile includere sia il modello di asset che la gerarchia.

Per ulteriori informazioni sulle gerarchie, consultare [Definite le gerarchie dei modelli di asset](#).

Asset

Questo filtro include la risorsa specificata:

```
"filterByAsset": {  
  "assetId": "asset ID"  
}
```

Asset and its asset model

Questo filtro include l'asset specificato, insieme al modello di asset che utilizza:

```
"filterByAsset": {  
  "assetId": "asset ID",  
  "includeAssetModel": true  
}
```

Asset and its hierarchy

Questo filtro include la risorsa specificata, insieme a tutte le risorse associate nella sua gerarchia:

```
"filterByAsset": {  
  "assetId": "asset ID",  
  "includeOffspring": true  
}
```

AWS IoT SiteWise schema del processo di trasferimento dei metadati

Utilizza lo schema del processo di trasferimento AWS IoT SiteWise dei metadati come riferimento quando esegui le tue operazioni di importazione ed esportazione in blocco:

```
{  
  "$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"
    }
  }
}
}

```

Monitora i dati con allarmi in AWS IoT SiteWise

È possibile configurare allarmi per i dati per avvisare il team quando le apparecchiature o i processi funzionano in modo non ottimale. Prestazioni ottimali di una macchina o di un processo significa che i valori di determinati parametri devono rientrare in un intervallo di limiti superiore e inferiore. Quando questi parametri sono esterni al loro intervallo di funzionamento, gli operatori delle apparecchiature devono ricevere una notifica per poter risolvere il problema. Utilizza gli allarmi per identificare rapidamente i problemi e avvisare gli operatori per massimizzare le prestazioni delle apparecchiature e dei processi.

Argomenti

- [Tipi di allarmi](#)
- [Stati di allarme](#)
- [Proprietà dello stato di allarme](#)
- [Definisci gli allarmi sui modelli di asset in AWS IoT SiteWise](#)
- [Configura gli allarmi sugli asset in AWS IoT SiteWise](#)
- [Rispondi agli allarmi in AWS IoT SiteWise](#)
- [Inserisci uno stato di allarme esterno in AWS IoT SiteWise](#)

Tipi di allarmi

È possibile definire allarmi rilevati nel AWS Cloud e allarmi rilevati con processi esterni. AWS IoT SiteWise supporta i seguenti tipi di allarmi:

- AWS IoT Events allarmi

Note

Avviso di fine del supporto: il 20 maggio 2026, AWS terminerà il supporto per. AWS IoT Events Dopo il 20 maggio 2026, non potrai più accedere alla AWS IoT Events console o AWS IoT Events alle risorse. Per ulteriori informazioni, consulta [AWS IoT Events Fine del supporto](#).

AWS IoT Events gli allarmi sono allarmi che vengono rilevati in. AWS IoT Events AWS IoT SiteWise invia i valori delle proprietà degli asset a un modello di allarme in. AWS IoT Events Quindi, AWS IoT Events invia lo stato di allarme a AWS IoT SiteWise. È possibile configurare opzioni come quando viene rilevato l'allarme e a chi notificare quando lo stato dell'allarme cambia. È inoltre possibile definire le [AWS IoT Events azioni](#) che si verificano quando lo stato dell'allarme cambia.

Gli allarmi in AWS IoT Events sono esempi di modelli di allarme. Il modello di allarme specifica la soglia e la gravità dell'allarme, cosa fare quando lo stato dell'allarme cambia e altro ancora. Quando si configura ogni caratteristica del modello di allarme, si specifica una proprietà di attributo dal modello di asset monitorato dall'allarme. Tutte le risorse basate sul modello di asset utilizzano il valore dell'attributo per AWS IoT Events valutare quella caratteristica dell'allarme. Per ulteriori informazioni, consulta [Uso degli allarmi](#) nella Guida per gli AWS IoT Events sviluppatori.

È possibile rispondere a un AWS IoT Events allarme quando cambia stato. Ad esempio, puoi confermare o posticipare una sveglia quando diventa attiva. Puoi anche abilitare, disabilitare e ripristinare gli allarmi.

SiteWise Gli utenti di Monitor possono visualizzare, configurare e rispondere agli AWS IoT Events allarmi nei SiteWise portali Monitor. Per ulteriori informazioni, consulta [Monitoraggio con allarmi](#) nella Guida all'applicazione.AWS IoT SiteWise Monitor

Note

AWS IoT Events si applicano costi per valutare questi allarmi e trasferire dati tra AWS IoT SiteWise e. AWS IoT Events Per ulteriori informazioni, consulta [Prezzi di AWS IoT Events](#).

• Allarmi esterni

Gli allarmi esterni sono allarmi che vengono valutati al di fuori. AWS IoT SiteWise Usa gli allarmi esterni se disponi di una fonte di dati che riporta lo stato degli allarmi. L'allarme esterno contiene una proprietà di misurazione in cui si inseriscono i dati sullo stato dell'allarme.

Non è possibile confermare o posticipare un allarme esterno quando cambia stato.

SiteWise Gli utenti di Monitor possono vedere lo stato degli allarmi esterni nei portali SiteWise Monitor, ma non possono configurare o rispondere a questi allarmi.

AWS IoT SiteWise non valuta lo stato degli allarmi esterni.

Stati di allarme

Gli allarmi industriali includono informazioni sullo stato dell'apparecchiatura o del processo che monitorano e (opzionali) informazioni sulla risposta dell'operatore allo stato di allarme.

Quando si definisce un AWS IoT Events allarme, si specifica se abilitare o meno il flusso di conferma. Il flusso di conferma è abilitato per impostazione predefinita. Quando si abilita questa opzione, gli operatori possono confermare l'allarme e lasciare una nota con i dettagli sull'allarme o sulle azioni intraprese per risolverlo. Se un operatore non riconosce un allarme attivo prima che diventi inattivo, l'allarme si blocca. Lo stato bloccato indica che l'allarme è diventato attivo e non è stato riconosciuto, quindi l'operatore deve controllare l'apparecchiatura o il processo e confermare l'allarme interrotto.

Gli allarmi hanno i seguenti stati:

- **Normale (Normal):** l'allarme è abilitato ma inattivo. Il processo o l'attrezzatura industriale funzionano come previsto.
- **Attivo (Active):** l'allarme è attivo. Il processo o l'attrezzatura industriale non rientra nel suo intervallo operativo e richiede attenzione.
- **Riconosciuto (Acknowledged):** un operatore ha riconosciuto lo stato dell'allarme.

Questo stato si applica solo agli allarmi in cui è abilitato il flusso di conferma.

- **Latched (Latched):** l'allarme è tornato alla normalità ma era attivo e nessun operatore lo ha riconosciuto. Il processo o l'attrezzatura industriale richiedono l'attenzione di un operatore per ripristinare l'allarme alla normalità.

Questo stato si applica solo agli allarmi in cui è abilitato il flusso di conferma.

- **Snoozed (SnoozeDisabled):** l'allarme è disattivato perché un operatore ha posticipato l'allarme. L'operatore definisce la durata della sonorizzazione dell'allarme. Dopo tale durata, l'allarme torna allo stato normale.
- **Disabilitato (Disabled):** l'allarme è disabilitato e non viene rilevato.

Proprietà dello stato di allarme

AWS IoT SiteWise memorizza i dati sullo stato di allarme come oggetto JSON serializzato su una stringa. Questo oggetto contiene lo stato e informazioni aggiuntive sull'allarme, come le azioni di risposta dell'operatore e la regola valutata dall'allarme.

Si identifica la proprietà dello stato dell'allarme in base al nome e al tipo di struttura, `AWS/ALARM_STATE`. Per ulteriori informazioni, consulta [Definisci gli allarmi sui modelli di asset in AWS IoT SiteWise](#).

L'oggetto dati sullo stato dell'allarme contiene le seguenti informazioni:

`stateName`

Lo stato di un allarme. Per ulteriori informazioni, consulta [Stati di allarme](#).

Tipo di dati: `STRING`

`customerAction`

(Facoltativo) Un oggetto che contiene informazioni sulla risposta di un operatore all'allarme. Gli operatori possono abilitare, disabilitare, confermare e posticipare gli allarmi. Quando lo fanno, i dati sullo stato di allarme includono la loro risposta e la nota che possono lasciare quando rispondono. Questo oggetto contiene le seguenti informazioni:

`actionName`

Il nome dell'azione intrapresa dall'operatore per rispondere all'allarme. Questo valore contiene una delle seguenti stringhe:

- `ENABLE`
- `DISABLE`
- `SNOOZE`
- `ACKNOWLEDGE`
- `RESET`

Tipo di dati: `STRING`

`enable`

(Facoltativo) Un oggetto presente `customerAction` quando l'operatore attiva l'allarme. Quando un operatore attiva l'allarme, lo stato dell'allarme cambia in `Normal`. Questo oggetto contiene le seguenti informazioni:

note

(Facoltativo) La nota che il cliente lascia quando attiva l'allarme.

Tipo di dati: STRING

Lunghezza massima: 128 caratteri

disable

(Facoltativo) Un oggetto presente `customerAction` quando l'operatore disattiva l'allarme. Quando un operatore attiva l'allarme, lo stato dell'allarme cambia in `Disabled`. Questo oggetto contiene le seguenti informazioni:

note

(Facoltativo) La nota che il cliente lascia quando disattiva l'allarme.

Tipo di dati: STRING

Lunghezza massima: 128 caratteri

acknowledge

(Facoltativo) Un oggetto presente `customerAction` quando l'operatore riconosce l'allarme. Quando un operatore attiva l'allarme, lo stato dell'allarme cambia in `Acknowledged`. Questo oggetto contiene le seguenti informazioni:

note

(Facoltativo) La nota che il cliente lascia quando riconosce l'allarme.

Tipo di dati: STRING

Lunghezza massima: 128 caratteri

snooze

(Facoltativo) Un oggetto presente `customerAction` quando l'operatore inserisce l'allarme. Quando un operatore attiva l'allarme, lo stato dell'allarme cambia in `SnoozeDisabled`. Questo oggetto contiene le seguenti informazioni:

snoozeDuration

La durata in secondi durante la quale l'operatore posticipa l'allarme. L'allarme cambia allo `Normal` stato dopo questa durata.

Tipo di dati: INTEGER

note

(Facoltativo) La nota che il cliente lascia quando posticipa la sveglia.

Tipo di dati: STRING

Lunghezza massima: 128 caratteri

ruleEvaluation

(Facoltativo) Un oggetto che contiene informazioni sulla regola che valuta l'allarme. Questo oggetto contiene le seguenti informazioni:

simpleRule

Un oggetto che contiene informazioni su una regola semplice, che confronta il valore di una proprietà con un valore di soglia con un operatore di confronto. Questo oggetto contiene le seguenti informazioni:

inputProperty

Il valore della proprietà che questo allarme valuta.

Tipo di dati: DOUBLE

operator

L'operatore di confronto utilizzato da questo allarme per confrontare la proprietà con la soglia. Questo valore contiene una delle seguenti stringhe:

- <— Meno di
- <=— Minore o uguale
- ==— Uguale
- !=— Non uguale
- >=— Maggiore o uguale
- >— Maggiore di

Tipo di dati: STRING

threshold

Il valore di soglia con cui questo allarme confronta il valore della proprietà.

Tipo di dati: DOUBLE

Definisci gli allarmi sui modelli di asset in AWS IoT SiteWise

I modelli di asset favoriscono la standardizzazione dei dati e degli allarmi industriali. È possibile definire le definizioni degli allarmi sui modelli di asset per standardizzare gli allarmi per tutti gli asset in base a un modello di asset.

Utilizzate modelli di asset composti per definire allarmi sui modelli di asset. I modelli di asset composti sono modelli di asset che standardizzano un insieme specifico di proprietà su un altro modello di asset. I modelli di asset composti assicurano la presenza di determinate proprietà su un modello di asset. Gli allarmi hanno proprietà relative al tipo, allo stato e (facoltativo) all'origine, quindi il modello composito degli allarmi impone l'esistenza di tali proprietà.

Ogni modello di asset composito ha un tipo che definisce le proprietà di quel modello composito. I modelli composti di allarme definiscono le proprietà per il tipo di allarme, lo stato di allarme e (opzionale) la fonte di allarme. Quando create una risorsa da un modello di asset con modelli composti, la risorsa include le proprietà del modello composito insieme alle proprietà specificate nel modello di asset.

Ogni proprietà in un modello composito deve avere il nome che la identifica per il tipo di modello composito. Le proprietà del modello composito supportano proprietà con tipi di dati complessi. Queste proprietà hanno il tipo di STRUCT dati e una `dataTypeSpec` caratteristica che specifica il tipo di dati complesso della proprietà. Le proprietà dei tipi di dati complessi contengono dati JSON serializzati come stringhe.

I modelli composti di allarme hanno le seguenti proprietà. Ogni proprietà deve avere il nome che la identifica per questo tipo di modello composito.

Tipo di allarme

Il tipo di allarme. Specifica una delle seguenti proprietà:

- `IOT_EVENTS`— Un AWS IoT Events allarme. AWS IoT SiteWise invia dati AWS IoT Events per valutare lo stato di questo allarme. È necessario specificare la proprietà della fonte dell'allarme per definire il modello di AWS IoT Events allarme per questa definizione di allarme.

Note

Avviso di fine del supporto: il 20 maggio 2026, AWS terminerà il supporto per AWS IoT Events. Dopo il 20 maggio 2026, non potrai più accedere alla AWS IoT Events console

o AWS IoT Events alle risorse. Per ulteriori informazioni, consulta [AWS IoT Events Fine del supporto](#).

- EXTERNAL— Un allarme esterno. Si inserisce lo stato dell'allarme come misurazione.

Nome della proprietà: AWS/ALARM_TYPE

Tipo di proprietà: [attributo](#)

Tipo di dati: STRING

Stato di allarme

I dati delle serie temporali relativi allo stato dell'allarme. Si tratta di un oggetto serializzato come stringa che contiene lo stato e altre informazioni sull'allarme. Per ulteriori informazioni, consulta [Proprietà dello stato di allarme](#).

Nome della proprietà: AWS/ALARM_STATE

Tipo di proprietà: [misurazione](#)

Tipo di dati: STRUCT

Tipo di struttura dati: AWS/ALARM_STATE

Fonte di allarme

(Facoltativo) L'Amazon Resource Name (ARN) della risorsa che valuta lo stato dell'allarme. Per gli AWS IoT Events allarmi, questo è l'ARN del modello di allarme.

Nome della proprietà: AWS/ALARM_SOURCE

Tipo di proprietà: [attributo](#)

Tipo di dati: STRING

Example Esempio di modello composito di allarme

Il seguente modello di impianto rappresenta una caldaia dotata di un allarme per monitorarne la temperatura. AWS IoT SiteWise invia i dati di temperatura AWS IoT Events a per rilevare l'allarme.

```
{
  "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": {}
        }
      ]
    }
  ]
}
```

Argomenti

- [Requisiti per le notifiche di allarme in AWS IoT SiteWise](#)
- [Definisci gli AWS IoT Events allarmi per AWS IoT SiteWise](#)
- [Definisci gli allarmi esterni in AWS IoT SiteWise](#)

Requisiti per le notifiche di allarme in AWS IoT SiteWise

AWS IoT Events utilizza una AWS Lambda funzione del tuo AWS account per inviare notifiche di allarme. È necessario creare questa funzione Lambda nella stessa AWS regione degli allarmi per abilitare le notifiche di allarme. Questa funzione Lambda utilizza [Amazon Simple Notification Service \(Amazon SNS\)](#) per inviare notifiche di testo e [Amazon Simple Email Service \(Amazon SES\)](#) per inviare notifiche e-mail. Quando crei l'AWS IoT Events allarme, configuri i protocolli e le impostazioni che l'allarme utilizza per inviare le notifiche.

Note

Avviso di fine del supporto: il 20 maggio 2026, AWS terminerà il supporto per AWS IoT Events. Dopo il 20 maggio 2026, non potrai più accedere alla AWS IoT Events console o AWS IoT Events alle risorse. Per ulteriori informazioni, consulta [AWS IoT Events Fine del supporto](#).

AWS IoT Events fornisce un modello di AWS CloudFormation stack che puoi usare per creare questa funzione Lambda nel tuo account. Per ulteriori informazioni, consulta la [funzione Lambda di notifica degli allarmi nella Guida](#) per gli AWS IoT Events sviluppatori.

Definisci gli AWS IoT Events allarmi per AWS IoT SiteWise

Note

Avviso di fine del supporto: il 20 maggio 2026, AWS terminerà il supporto per AWS IoT Events. Dopo il 20 maggio 2026, non potrai più accedere alla AWS IoT Events console o AWS IoT Events alle risorse. Per ulteriori informazioni, consulta [AWS IoT Events Fine del supporto](#).

Quando crei un AWS IoT Events allarme, AWS IoT SiteWise invia i valori delle proprietà dell'asset AWS IoT Events per valutare lo stato dell'allarme. AWS IoT Events le definizioni degli allarmi dipendono dal modello di allarme in cui si definisce AWS IoT Events. Per definire un AWS IoT Events allarme su un modello di asset, si definisce un modello composito di allarme che specifica il modello di AWS IoT Events allarme come proprietà della fonte dell'allarme.

AWS IoT Events gli allarmi dipendono da input come le soglie di allarme e le impostazioni di notifica degli allarmi. Questi input vengono definiti come attributi nel modello di asset. È quindi possibile personalizzare questi input su ogni risorsa in base al modello. La AWS IoT SiteWise console può creare questi attributi per te. Se definisci gli allarmi con l'API AWS CLI o, devi definire manualmente questi attributi nel modello di asset.

Puoi anche definire altre azioni che si verificano quando viene rilevato un allarme, come azioni di notifica di allarme personalizzate. Ad esempio, puoi configurare un'azione che invia una notifica push a un argomento di Amazon SNS. Per ulteriori informazioni sulle azioni che puoi definire, consulta [Lavorare con altri AWS servizi](#) nella Guida per gli AWS IoT Events sviluppatori.

Quando aggiorni o elimini un modello di asset, AWS IoT SiteWise puoi verificare se un modello di allarme in AWS IoT Events sta monitorando una proprietà di asset associata a questo modello di asset. Ciò impedisce di eliminare una proprietà dell'asset attualmente utilizzata da un AWS IoT Events allarme. Per abilitare questa funzionalità AWS IoT SiteWise, è necessario disporre dell'`iotevents:ListInputRoutings` autorizzazione. Questa autorizzazione consente di AWS IoT SiteWise effettuare chiamate all'operazione [ListInputRoutings](#) API supportata da AWS IoT Events. Per ulteriori informazioni, consulta [ListInputRoutings Autorizzazione \(Facoltativa\)](#).

Note

La funzionalità di notifica degli allarmi non è disponibile nella regione Cina (Pechino).

Argomenti

- [Definisci un AWS IoT Events allarme \(AWS IoT SiteWise console\)](#)
- [Definisci un AWS IoT Events allarme \(AWS IoT Events console\)](#)
- [Definisci un AWS IoT Events allarme \(AWS CLI\)](#)

Definisci un AWS IoT Events allarme (AWS IoT SiteWise console)

È possibile utilizzare la AWS IoT SiteWise console per definire un AWS IoT Events allarme su un modello di asset esistente. Per definire un AWS IoT Events allarme su un nuovo modello di asset, create il modello di asset, quindi completate questi passaggi. Per ulteriori informazioni, consulta [Crea modelli di asset in AWS IoT SiteWise](#).

Important

Ogni allarme richiede un attributo che specifica il valore di soglia con cui confrontare l'allarme. È necessario definire l'attributo del valore di soglia nel modello di asset prima di poter definire un allarme.

Si consideri un esempio in cui si desidera definire un allarme che rilevi quando una turbina eolica supera la velocità massima del vento di 50 mph. Prima di definire l'allarme, è necessario definire un attributo (Velocità massima del vento) con un valore predefinito di. 50

Per definire un AWS IoT Events allarme su un modello di asset

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione selezionare Models (Modelli).
3. Scegliete il modello di asset per il quale definire un allarme.
4. Scegliete la scheda Allarme.
5. Scegli Aggiungi allarme.
6. Nella sezione Opzioni del tipo di allarme, scegli AWS IoT Events Allarme.
7. Nella sezione Dettagli dell'allarme, procedi come segue:
 - a. Immetti un nome per l'allarme.
 - b. (Facoltativo) Inserisci una descrizione per l'allarme.

8. Nella sezione Definizioni delle soglie, definisci quando viene rilevato l'allarme e la gravità dell'allarme. Esegui questa operazione:
 - a. Seleziona la proprietà in base alla quale viene rilevato l'allarme. Ogni volta che questa proprietà riceve un nuovo valore, AWS IoT SiteWise invia il valore AWS IoT Events a per valutare lo stato dell'allarme.
 - b. Seleziona l'operatore da utilizzare per confrontare la proprietà con il valore di soglia. Seleziona una delle opzioni seguenti:
 - < meno di
 - <= minore o uguale
 - == uguale
 - != non uguale
 - >= maggiore o uguale
 - > maggiore di
 - c. Per Valore, selezionate la proprietà dell'attributo da utilizzare come valore di soglia. AWS IoT Events confronta il valore della proprietà con il valore di questo attributo.
 - d. Inserisci la gravità dell'allarme. Usa un numero comprensibile al tuo team per indicare la gravità dell'allarme.
9. (Facoltativo) Nella sezione Impostazioni di notifica - opzionale, procedi come segue:

- a. Scegli Attivo.

 Note

Se scegli Inattivo, tu e il tuo team non riceverete alcuna notifica di allarme.

- b. Per Destinatario, scegli il destinatario.

 Important

È possibile inviare notifiche di allarme agli AWS IAM Identity Center utenti. Per utilizzare questa funzionalità, è necessario abilitare IAM Identity Center. Puoi abilitare IAM Identity Center solo in una AWS regione alla volta. Ciò significa che puoi definire le notifiche di allarme solo nella regione in cui abiliti IAM Identity Center.

Per ulteriori informazioni, consulta [Nozioni di base](#) nella Guida per l'utente di AWS IAM Identity Center .

- c. Per Protocollo, scegli tra le seguenti opzioni:
- Email e testo: l'allarme avvisa gli utenti di IAM Identity Center con un messaggio SMS e un messaggio e-mail.
 - E-mail: l'allarme avvisa gli utenti di IAM Identity Center con un messaggio e-mail.
 - Testo: l'allarme avvisa gli utenti di IAM Identity Center con un messaggio SMS.
- d. Per Mittente, scegli il mittente.

 Important

È necessario verificare l'indirizzo e-mail del mittente in Amazon Simple Email Service (Amazon SES). Per ulteriori informazioni, consulta la sezione [Verifica dell'identità di un indirizzo e-mail](#), nella Amazon Simple Email Service Developer Guide.

10. Nella sezione Stato predefinito degli asset, puoi impostare lo stato predefinito per gli allarmi creati da questo modello di asset.

 Note

Attivate o disattivate questo allarme per gli asset che create da questo modello di asset in una fase successiva.

11. Nella sezione Impostazioni avanzate, puoi configurare le autorizzazioni, le impostazioni di notifica aggiuntive, le azioni relative allo stato degli allarmi, il modello di allarme in SiteWise Monitor e il flusso di conferma.

 Note

AWS IoT Events gli allarmi richiedono i seguenti ruoli di servizio:

- Un ruolo che AWS IoT Events presuppone di inviare i valori dello stato di allarme a. AWS IoT SiteWise

- Un ruolo che AWS IoT Events presuppone l'invio di dati a Lambda. Questo ruolo è necessario solo se l'allarme invia notifiche.

Nella sezione Autorizzazioni, procedi come segue:

- a. Per AWS IoT Events il ruolo, utilizza un ruolo esistente o crea un ruolo con le autorizzazioni richieste. Questo ruolo richiede `iotsitewise:BatchPutAssetPropertyValue` autorizzazione e una relazione di fiducia che consenta a `iotevents.amazonaws.com` di assumere il ruolo.
- b. Per il ruolo AWS IoT Events Lambda, usa un ruolo esistente o crea un ruolo con le autorizzazioni richieste. Questo ruolo richiede le `sso-directory:DescribeUser` autorizzazioni `lambda:InvokeFunction` e e una relazione di fiducia che `iotevents.amazonaws.com` consenta di assumere il ruolo.

12. (Facoltativo) Nella sezione Impostazioni di notifica aggiuntive, procedi come segue:

- a. Per l'attributo Destinatario, si definisce un attributo il cui valore specifica il destinatario della notifica. Puoi scegliere gli utenti di IAM Identity Center come destinatari.

Puoi creare un attributo o utilizzare un attributo esistente sul modello di asset.

- Se scegli Crea un nuovo attributo destinatario, specifica il nome dell'attributo Destinatario e il valore predefinito del destinatario, facoltativo per l'attributo.
- Se scegli Usa un attributo destinatario esistente, scegli l'attributo in Nome attributo destinatario. L'avviso utilizza il valore predefinito dell'attributo scelto.

È possibile sovrascrivere il valore predefinito di ogni risorsa creata da questo modello di asset.

- b. Per l'attributo messaggio personalizzato, si definisce un attributo il cui valore specifica il messaggio personalizzato da inviare oltre al messaggio di modifica dello stato predefinito. Ad esempio, puoi specificare un messaggio che aiuti il tuo team a capire come rispondere a questo allarme.

È possibile scegliere di creare un attributo o utilizzare un attributo esistente nel modello di asset.

- Se scegli di creare un nuovo attributo personalizzato del messaggio, specifica il nome dell'attributo del messaggio personalizzato e il valore predefinito del messaggio personalizzato, facoltativo per l'attributo.
- Se scegli Usa un attributo di messaggio personalizzato esistente, scegli l'attributo in Nome attributo messaggio personalizzato. L'avviso utilizza il valore predefinito dell'attributo scelto.

È possibile sovrascrivere il valore predefinito di ogni risorsa creata da questo modello di asset.

c. Per Gestire la funzione Lambda, effettuate una delle seguenti operazioni:

- Per AWS IoT SiteWise creare una nuova funzione Lambda, scegli Crea una nuova lambda da un modello gestito da AWS.
- Per utilizzare una funzione Lambda esistente, scegli Usa una funzione Lambda esistente e scegli il nome della funzione.

Per ulteriori informazioni, consulta [Gestione delle notifiche di allarme nella Guida](#) per gli AWS IoT Events sviluppatori.

13. (Facoltativo) Nella sezione Imposta l'azione dello stato, procedi come segue:

- a. Scegliete Modifica azione.
- b. In Aggiungi azioni relative allo stato di allarme, aggiungi azioni, quindi scegli Salva.

Puoi aggiungere fino a 10 azioni.

AWS IoT Events può eseguire azioni quando l'allarme è attivo. È possibile definire azioni integrate per utilizzare un timer o impostare una variabile o inviare dati ad altre AWS risorse. Per ulteriori informazioni, consulta [Azioni supportate](#) nella Guida per AWS IoT Events gli sviluppatori.

14. (Facoltativo) In Gestisci il modello di allarme in SiteWise Monitor: facoltativo, scegli Attivo o Inattivo.

Usa questa opzione per aggiornare il modello di allarme in SiteWise Monitorss. Per impostazione predefinita, questa opzione è abilitata.

15. In Conferma flusso, scegli Attivo o Inattivo. Per ulteriori informazioni sul flusso di conferma, consulta [Stati di allarme](#).

16. Scegli Aggiungi allarme.

Note

La AWS IoT SiteWise console effettua più richieste API per aggiungere l'allarme al modello di asset. Quando scegli Aggiungi allarme, la console apre una finestra di dialogo che mostra lo stato di avanzamento di queste richieste API. Rimani su questa pagina fino a quando ogni richiesta API non ha esito positivo o fino a quando una richiesta API non riesce. Se una richiesta fallisce, chiudi la finestra di dialogo, risolvi il problema e scegli Aggiungi allarme per riprovare.

Definisci un AWS IoT Events allarme (AWS IoT Events console)

È possibile utilizzare la AWS IoT Events console per definire un AWS IoT Events allarme su un modello di asset esistente. Per definire un AWS IoT Events allarme su un nuovo modello di asset, create il modello di asset, quindi completate questi passaggi. Per ulteriori informazioni, consulta [Crea modelli di asset in AWS IoT SiteWise](#).

Important

Ogni allarme richiede un attributo che specifica il valore di soglia con cui confrontare l'allarme. È necessario definire l'attributo del valore di soglia nel modello di asset prima di poter definire un allarme.

Si consideri un esempio in cui si desidera definire un allarme che rilevi quando una turbina eolica supera la velocità massima del vento di 50 mph. Prima di definire l'allarme, è necessario definire un attributo (Velocità massima del vento) con un valore predefinito di 50.

Per definire un AWS IoT Events allarme su un modello di asset

1. Passare alla [console AWS IoT Events](#).
2. Nel pannello di navigazione, scegli Modelli di allarme.
3. Scegli Crea modello di allarme.
4. Immetti un nome per l'allarme.
5. (Facoltativo) Inserisci una descrizione per il tuo allarme.
6. Nella sezione Obiettivo dell'allarme, procedi come segue:

- a. Per le opzioni di Target, scegli la proprietà AWS IoT SiteWise dell'asset.
 - b. Scegli il modello di asset per il quale desideri aggiungere l'allarme.
7. Nella sezione Definizioni delle soglie, definisci quando viene rilevato l'allarme e la gravità dell'allarme. Esegui questa operazione:
 - a. Seleziona la proprietà in base alla quale viene rilevato l'allarme. Ogni volta che questa proprietà riceve un nuovo valore, AWS IoT SiteWise invia il valore AWS IoT Events a per valutare lo stato dell'allarme.
 - b. Seleziona l'operatore da utilizzare per confrontare la proprietà con il valore di soglia. Seleziona una delle opzioni seguenti:
 - < meno di
 - <= minore o uguale
 - == uguale
 - != non uguale
 - >= maggiore o uguale
 - > maggiore di
 - c. Per Valore, selezionate la proprietà dell'attributo da utilizzare come valore di soglia. AWS IoT Events confronta il valore della proprietà con il valore di questo attributo.
 - d. Inserisci la gravità dell'allarme. Usa un numero comprensibile al tuo team per indicare la gravità dell'allarme.
8. (Facoltativo) Nella sezione Impostazioni di notifica - opzionale, procedi come segue:
 - a. Per Protocollo, scegli tra le seguenti opzioni:
 - Email e testo: l'allarme avvisa gli utenti di IAM Identity Center con un messaggio SMS e un messaggio e-mail.
 - E-mail: l'allarme avvisa gli utenti di IAM Identity Center con un messaggio e-mail.
 - Testo: l'allarme avvisa gli utenti di IAM Identity Center con un messaggio SMS.
 - b. Per Mittente, scegli il mittente.

 Important

È necessario verificare l'indirizzo e-mail del mittente in Amazon Simple Email Service (Amazon SES). Per ulteriori informazioni, consulta la sezione [Verifica degli](#)

[indirizzi e-mail in Amazon SES](#), nella Amazon Simple Email Service Developer Guide.

- c. Scegli l'attributo nell'attributo Recipient (facoltativo). L'allarme utilizza il valore predefinito dell'attributo scelto.
 - d. Scegli l'attributo in Attributo messaggio personalizzato - opzionale. L'avviso utilizza il valore predefinito dell'attributo scelto.
9. Nella sezione Istanza, specifica lo stato predefinito per questo avviso. È possibile attivare o disattivare questo allarme per tutte le risorse create da questo modello di asset in un passaggio successivo.
10. Nelle impostazioni avanzate, è possibile configurare le autorizzazioni, le impostazioni di notifica aggiuntive, le azioni relative allo stato di allarme, il modello di allarme in SiteWise Monitor e il flusso di conferma.

 Note

AWS IoT Events gli allarmi richiedono i seguenti ruoli di servizio:

- Un ruolo che AWS IoT Events presuppone di inviare i valori dello stato di allarme a. AWS IoT SiteWise
- Un ruolo che AWS IoT Events presuppone l'invio di dati a Lambda. Questo ruolo è necessario solo se l'allarme invia notifiche.

- a. Nella sezione Conferma flusso, scegli Abilitato o Disabilitato. Per ulteriori informazioni sul flusso di conferma, consulta [Stati di allarme](#).
- b. Nella sezione Autorizzazioni, procedi come segue:
 - i. Per AWS IoT Events il ruolo, utilizza un ruolo esistente o crea un ruolo con le autorizzazioni richieste. Questo ruolo richiede `iotsitewise:BatchPutAssetPropertyValue` autorizzazione e una relazione di fiducia che consenta a `iotevents.amazonaws.com` di assumere il ruolo.
 - ii. Per il ruolo Lambda, usa un ruolo esistente o crea un ruolo con le autorizzazioni richieste. Questo ruolo richiede le `sso-directory:DescribeUser` autorizzazioni `lambda:InvokeFunction` e e una relazione di fiducia che `iotevents.amazonaws.com` consenta di assumere il ruolo.

c. (Facoltativo) Nel riquadro Impostazioni di notifica aggiuntive, procedi come segue:

- Per Gestire la funzione Lambda, effettuate una delle seguenti operazioni:
 - Per AWS IoT Events creare una nuova funzione Lambda, scegli Crea una nuova funzione Lambda.
 - Per utilizzare una funzione Lambda esistente, scegli Usa una funzione Lambda esistente e scegli il nome della funzione.

Per ulteriori informazioni, consulta [Gestione delle notifiche di allarme nella Guida](#) per gli AWS IoT Events sviluppatori.

d. (Facoltativo) Nella sezione Imposta l'azione dello stato - opzionale, procedi come segue:

- In Azioni relative allo stato di allarme, aggiungi azioni, quindi scegli Salva.

Puoi aggiungere fino a 10 azioni.

AWS IoT Events può eseguire azioni quando l'allarme è attivo. È possibile definire azioni integrate per utilizzare un timer o impostare una variabile o inviare dati ad altre AWS risorse. Per ulteriori informazioni, consulta [Azioni supportate](#) nella Guida per AWS IoT Events gli sviluppatori.

11. Scegli Create (Crea).

Note

La AWS IoT Events console effettua più richieste API per aggiungere l'allarme al modello di asset. Quando scegli Aggiungi allarme, la console apre una finestra di dialogo che mostra lo stato di avanzamento di queste richieste API. Rimani su questa pagina fino a quando ogni richiesta API non ha esito positivo o fino a quando una richiesta API non riesce. Se una richiesta fallisce, chiudi la finestra di dialogo, risolvi il problema e scegli Aggiungi allarme per riprovare.

Definisci un AWS IoT Events allarme (AWS CLI)

È possibile utilizzare il AWS Command Line Interface (AWS CLI) per definire un AWS IoT Events allarme che monitora la proprietà di un asset. È possibile definire l'allarme su un modello di asset

nuovo o esistente. Dopo aver definito l'allarme sul modello di asset, create un allarme AWS IoT Events e lo collegate al modello di asset. In questo processo, effettuate le seguenti operazioni:

Fasi

- [Fase 1: Definizione di un allarme su un modello di asset](#)
- [Fase 2: Definizione di un modello di allarme AWS IoT Events](#)
- [Passaggio 3: Abilita il flusso di dati tra e AWS IoT SiteWiseAWS IoT Events](#)

Fase 1: Definizione di un allarme su un modello di asset

Aggiungi una definizione di allarme e le proprietà associate a un modello di asset nuovo o esistente.

Per definire un allarme su un modello di asset (CLI)

1. Crea un file denominato `asset-model-payload.json`. Segui i passaggi in queste altre sezioni per aggiungere i dettagli del tuo modello di asset al file, ma non inviare la richiesta per creare o aggiornare il modello di asset. In questa sezione, aggiungi una definizione di allarme ai dettagli del modello di asset contenuti nel `asset-model-payload.json` file.
 - Per ulteriori informazioni su come creare un modello di asset, consulta [Crea un modello di asset \(AWS CLI\)](#).
 - Per ulteriori informazioni su come aggiornare un modello di asset esistente, consulta [Aggiornate un modello di asset o componente \(AWS CLI\)](#).

Note

Il modello di asset deve definire almeno una proprietà dell'asset, inclusa la proprietà dell'asset da monitorare con l'allarme.

2. Aggiungi un modello composito di allarme (`assetModelCompositeModels`) al modello di asset. Un modello composito di AWS IoT Events allarme specifica il `IOT_EVENTS` tipo e specifica una proprietà della fonte di allarme. Si aggiunge la proprietà della fonte di allarme dopo aver creato il modello di allarme in AWS IoT Events

Important

Il modello composito di allarme deve avere lo stesso nome del modello di AWS IoT Events allarme creato in seguito. I nomi dei modelli di allarme possono contenere solo

caratteri alfanumerici. Specificate un nome alfanumerico univoco in modo da poter utilizzare lo stesso nome per il modello di allarme.

```
{
  ...
  "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. Aggiungi un attributo di soglia di allarme al modello di asset. Specificate il valore predefinito da utilizzare per questa soglia. È possibile sovrascrivere questo valore predefinito su ogni risorsa basata su questo modello.

Note

L'attributo della soglia di allarme deve essere un INTEGER o unDOUBLE.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "Temperature Max Threshold",
      "dataType": "DOUBLE",
      "type": {
        "attribute": {
          "defaultValue": "105.0"
        }
      }
    }
  ]
}
```

4. (Facoltativo) Aggiungi gli attributi di notifica degli allarmi al modello di asset. Questi attributi specificano il destinatario di IAM Identity Center e altri input AWS IoT Events utilizzati per inviare notifiche quando l'allarme cambia stato. Puoi sovrascrivere queste impostazioni predefinite su ogni risorsa basata su questo modello.

 Important

È possibile inviare notifiche di allarme agli utenti. AWS IAM Identity Center Per utilizzare questa funzionalità, è necessario abilitare IAM Identity Center. Puoi abilitare IAM Identity Center solo in una AWS regione alla volta. Ciò significa che puoi definire le notifiche di allarme solo nella regione in cui abiliti IAM Identity Center. Per ulteriori informazioni, consulta [Nozioni di base](#) nella Guida per l'utente di AWS IAM Identity Center .

Esegui questa operazione:

- a. Aggiungi un attributo che specifica l'ID del tuo archivio di identità IAM Identity Center. Puoi utilizzare l'operazione [ListInstances](#) API IAM Identity Center per elencare i tuoi archivi di identità. Questa operazione funziona solo nella regione in cui è abilitato IAM Identity Center.

```
aws sso-admin list-instances
```

Quindi, specifica l'Identity Store ID (ad esempio, d-123EXAMPLE) come valore predefinito per l'attributo.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "identityStoreId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "d-123EXAMPLE"
        }
      }
    }
  ]
}
```

- b. Aggiungi un attributo che specifica l'ID dell'utente IAM Identity Center che riceve le notifiche. Per definire un destinatario di notifica predefinito, aggiungi un ID utente IAM Identity Center come valore predefinito. Effettua una delle seguenti operazioni per ottenere un ID utente IAM Identity Center:
 - i. Puoi utilizzare l'[ListUsers](#) API IAM Identity Center per ottenere l'ID di un utente di cui conosci il nome utente. Sostituiscilo *d-123EXAMPLE* con l'ID del tuo archivio di identità e sostituiscilo *Name* con il nome utente dell'utente.

```
aws identitystore list-users \
  --identity-store-id d-123EXAMPLE \
  --filters AttributePath=UserName,AttributeValue=Name
```

- ii. Utilizza la [console IAM Identity Center](#) per sfogliare gli utenti e trovare un ID utente.

Quindi, specifica l'ID utente (ad esempio 123EXAMPLE-a1b2c3d4-5678-90ab-cdef-3333EXAMPLE) come valore predefinito per l'attributo o definisci l'attributo senza un valore predefinito.

```
{
  ...
```

```

"assetModelProperties": [
  ...
  {
    "name": "userId",
    "dataType": "STRING",
    "type": {
      "attribute": {
        "defaultValue": "123EXAMPLE-a1b2c3d4-5678-90ab-cdef-33333EXAMPLE"
      }
    }
  }
]
}

```

- c. (Facoltativo) Aggiungi un attributo che specifica l'ID mittente predefinito per le notifiche tramite SMS (testo). L'ID mittente viene visualizzato come mittente dei messaggi inviati da Amazon Simple Notification Service (Amazon SNS). Per ulteriori informazioni, consulta [Richiedere un ID mittente AWS End User Messaging SMS nella Guida per l'utente.AWS End User Messaging SMS](#)

```

{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "senderId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "MyFactory"
        }
      }
    }
  ]
}

```

- d. (Facoltativo) Aggiungi un attributo che specifica l'indirizzo e-mail predefinito da utilizzare come indirizzo del mittente nelle notifiche e-mail.

```

{
  ...
  "assetModelProperties": [

```

```

...
{
  "name": "fromAddress",
  "dataType": "STRING",
  "type": {
    "attribute": {
      "defaultValue": "my.factory@example.com"
    }
  }
}
]
}

```

- e. (Facoltativo) Aggiungi un attributo che specifica l'oggetto predefinito da utilizzare nelle notifiche e-mail.

```

{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "emailSubject",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "[ALERT] High boiler temperature"
        }
      }
    }
  ]
}

```

- f. (Facoltativo) Aggiungi un attributo che specifica un messaggio aggiuntivo da includere nelle notifiche. Per impostazione predefinita, i messaggi di notifica includono informazioni sull'allarme. È inoltre possibile includere un messaggio aggiuntivo che fornisca all'utente ulteriori informazioni.

```

{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "additionalMessage",

```

```

    "dataType": "STRING",
    "type": {
      "attribute": {
        "defaultValue": "Turn off the power before you check the alarm."
      }
    }
  }
]
}

```

5. Crea il modello di asset o aggiorna il modello di asset esistente. Esegui una di queste operazioni:

- Per creare il modello di asset, esegui il comando seguente.

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-
payload.json
```

- Per aggiornare il modello di asset esistente, esegui il comando seguente. Sostituisci *asset-model-id* con l'ID del modello di asset.

```
aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --cli-input-json file://asset-model-payload.json
```

Dopo aver eseguito il comando, `assetModelId` annotatelo nella risposta.

Esempio: modello di asset per caldaie

Il seguente modello di asset rappresenta una caldaia che riporta i dati sulla temperatura. Questo modello di asset definisce un allarme che rileva il surriscaldamento della caldaia.

```

{
  "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": {}
        }
      }
    ]
  }
]

```

```

    }
  }
]
}
]
}

```

Fase 2: Definizione di un modello di allarme AWS IoT Events

Crea il modello di allarme in AWS IoT Events. In AWS IoT Events, si utilizzano le espressioni per specificare i valori nei modelli di allarme. È possibile utilizzare espressioni per specificare valori AWS IoT SiteWise da valutare e utilizzare come input per l'allarme. Quando AWS IoT SiteWise invia i valori delle proprietà dell'asset al modello di allarme, AWS IoT Events valuta l'espressione per ottenere il valore della proprietà o l'ID della risorsa. È possibile utilizzare le seguenti espressioni nel modello di allarme:

- Valori delle proprietà degli asset

Per ottenere il valore di una proprietà dell'asset, utilizzate la seguente espressione.

assetModelId Sostituítelo con l'ID del modello di asset e *propertyId* sostituítelo con l'ID della proprietà.

```
$sitewise.assetModel.`assetModelId`.`propertyId`.propertyValue.value
```

- Risorsa IDs

Per ottenere l'ID della risorsa, utilizzate la seguente espressione. *assetModelId* Sostituítelo con l'ID del modello di asset e *propertyId* sostituítelo con l'ID della proprietà.

```
$sitewise.assetModel.`assetModelId`.`propertyId`.assetId
```

Note

Quando create il modello di allarme, potete definire valori letterali anziché espressioni che restituiscono AWS IoT SiteWise valori. In questo modo è possibile ridurre il numero di attributi definiti nel modello di asset. Tuttavia, se definisci un valore come valore letterale, non puoi personalizzare quel valore sugli asset in base al modello di asset. Inoltre, gli AWS IoT

SiteWise Monitor utenti non possono personalizzare l'allarme, poiché possono configurare le impostazioni di allarme solo sugli asset.

Per creare un modello di AWS IoT Events allarme (CLI)

1. Quando si crea il modello di allarme in AWS IoT Events, è necessario specificare l'ID di ogni proprietà utilizzata dall'allarme, che include quanto segue:
 - La proprietà dello stato di allarme nel modello di asset composito
 - La proprietà monitorata dall'allarme
 - L'attributo threshold
 - (Facoltativo) L'attributo ID del negozio di identità di IAM Identity Center
 - (Facoltativo) L'attributo ID utente di IAM Identity Center
 - (Facoltativo) L'attributo ID mittente SMS
 - (Facoltativo) L'attributo email proveniente dall'indirizzo
 - (Facoltativo) L'attributo oggetto dell'email
 - (Facoltativo) L'attributo aggiuntivo del messaggio

Eseguite il comando seguente per recuperare queste proprietà sul modello IDs di asset. *asset-model-id* Sostituirelo con l'ID del modello di asset del passaggio precedente.

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

L'operazione restituisce una risposta contenente i dettagli del modello di asset. Annota l'ID di ogni proprietà utilizzata dall'allarme. Li usi IDs quando crei il modello di AWS IoT Events allarme nel passaggio successivo.

2. Crea il modello di allarme in AWS IoT Events. Esegui questa operazione:
 - a. Crea un file denominato `alarm-model-payload.json`.
 - b. Copia il seguente oggetto JSON nel file.
 - c. Inserisci un nome (`alarmModelName`), una descrizione (`alarmModelDescription`) e una gravità (`severity`) per l'allarme. Per quanto riguarda la gravità, specificate un numero intero che rifletta i livelli di gravità della vostra azienda.

⚠ Important

Il modello di allarme deve avere lo stesso nome del modello composito di allarmi definito in precedenza nel modello di asset.

I nomi dei modelli di allarme possono contenere solo caratteri alfanumerici.

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3
}
```

- d. Aggiungi la regola di confronto (alarmRule) all'allarme. Questa regola definisce la proprietà da monitorare (inputProperty), il valore di soglia da confrontare (threshold) e l'operatore di confronto da utilizzare (comparisonOperator).
- Sostituisci *assetModelId* con l'ID del modello di asset.
 - Sostituisci *alarmPropertyId* con l'ID della proprietà monitorata dall'allarme.
 - Sostituisci *thresholdAttributeId* con l'ID della proprietà dell'attributo di soglia.
 - Sostituisci *GREATER* con l'operatore da utilizzare per confrontare i valori delle proprietà con la soglia. Seleziona una delle opzioni seguenti:
 - 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. Aggiungi un'azione (alarmEventActions) per inviare lo stato di allarme al AWS IoT SiteWise momento in cui l'allarme cambia stato.

Note

Per la configurazione avanzata, è possibile definire azioni aggiuntive da eseguire quando l'allarme cambia stato. Ad esempio, è possibile richiamare una AWS Lambda funzione o pubblicare su un argomento MQTT. Per ulteriori informazioni, consulta [Lavorare con altri AWS servizi](#) nella Guida per gli AWS IoT Events sviluppatori.

- Sostituisci *assetModelId* con l'ID del modello di asset.
- Sostituisci *alarmPropertyId* con l'ID della proprietà monitorata dall'allarme.
- Sostituisci *alarmStatePropertyId* con l'ID della proprietà dello stato di allarme nel modello composito di allarme.

```

{
  "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. (Facoltativo) Configura le impostazioni di notifica degli allarmi. L'azione di notifica degli allarmi utilizza una funzione Lambda nel tuo account per inviare notifiche di allarme. Per ulteriori informazioni, consulta [Requisiti per le notifiche di allarme in AWS IoT SiteWise](#). Nelle impostazioni di notifica degli allarmi, puoi configurare le notifiche SMS ed e-mail da inviare agli utenti di IAM Identity Center. Esegui questa operazione:
- i. Aggiungi la configurazione della notifica di allarme (alarmNotification) al payload in alarm-model-payload.json.
 - Sostituisci *alarmNotificationFunctionArn* con l'ARN della funzione Lambda che gestisce le notifiche di allarme.

```

{
  "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. (Facoltativo) Configura le notifiche SMS (smsConfigurations) da inviare a un utente di IAM Identity Center quando l'allarme cambia stato.
 - Sostituisci *identityStoreIdAttributeId* con l'ID dell'attributo che contiene l'ID dell'archivio di identità di IAM Identity Center.
 - Sostituisci *userIdAttributeId* con l'ID dell'attributo che contiene l'ID dell'utente IAM Identity Center.
 - Sostituisci *senderIdAttributeId* con l'ID dell'attributo che contiene l'ID mittente di Amazon SNS o rimuovilo senderId dal payload.
 - Sostituiscilo *additionalMessageAttributeId* con l'ID dell'attributo che contiene il messaggio aggiuntivo o rimuovilo additionalMessage dal 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":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.va
                                    "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
                                }
                            }
                        ],
                        "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId`.propertyValue.value",
                        "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.
                    }
                ]
            }
        ]
    }
}

```



```
}
}
```

iii. (Facoltativo) Configura le notifiche e-mail (emailConfigurations) da inviare a un utente di IAM Identity Center quando l'allarme cambia stato.

- Sostituisci *identityStoreIdAttributeId* con l'ID della proprietà dell'attributo Identity Store ID di IAM Identity Center.
- Sostituisci *userIdAttributeId* con l'ID della proprietà dell'attributo ID utente di IAM Identity Center.
- Sostituiscila *fromAddressAttributeId* con l'ID della proprietà dell'attributo dell'indirizzo «from» o from rimuovila dal payload.
- Sostituisci *emailSubjectAttributeId* con l'ID della proprietà dell'attributo dell'oggetto dell'email o rimuovi subject dal payload.
- Sostituisci *additionalMessageAttributeId* con l'ID della proprietà aggiuntiva dell'attributo del messaggio o rimuovila additionalMessage dal 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. (Facoltativo) Aggiungi le funzionalità di allarme (alarmCapabilities) al payload in `alarm-model-payload.json`. In questo oggetto, è possibile specificare se il flusso di conferma è abilitato e lo stato di abilitazione predefinito per gli asset in base al modello di asset. Per ulteriori informazioni sul flusso di conferma, consulta [Stati di allarme](#).

```

{
  "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. Aggiungi il ruolo del servizio IAM (`roleArn`) a cui AWS IoT Events puoi presumere di inviare dati AWS IoT SiteWise. Questo ruolo richiede `iotsitewise:BatchPutAssetPropertyValue` autorizzazione e una relazione di fiducia che `iotevents.amazonaws.com` consenta di assumere il ruolo. Per inviare notifiche, questo ruolo richiede anche le `sso-directory:DescribeUser` autorizzazioni `lambda:InvokeFunction` e. Per ulteriori informazioni, consulta i [ruoli del servizio di allarme](#) nella Guida per gli AWS IoT Events sviluppatori.

- Sostituisci `roleArn` con l'ARN del ruolo che AWS IoT Events può assumere per eseguire queste azioni.

```

{
  "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. Esegui il comando seguente per creare il modello di AWS IoT Events allarme dal payload in `alarm-model-payload.json`

```
aws iotevents create-alarm-model --cli-input-json file://alarm-model-payload.json
```

- j. L'operazione restituisce una risposta che include l'ARN del modello di allarme, `alarmModelArn`. Copia questo ARN per impostare la definizione di allarme sul tuo modello di asset nel passaggio successivo.

Passaggio 3: Abilita il flusso di dati tra e AWS IoT SiteWiseAWS IoT Events

Dopo aver creato le risorse richieste in AWS IoT SiteWise e AWS IoT Events, puoi abilitare il flusso di dati tra le risorse per attivare l'allarme. In questa sezione, aggiorni la definizione di allarme nel modello di asset per utilizzare il modello di allarme creato nel passaggio precedente.

Per abilitare il flusso di dati tra AWS IoT SiteWise e AWS IoT Events (CLI)

- Imposta il modello di allarme come fonte dell'allarme nel modello di asset. Esegui questa operazione:
 - a. Esegui il comando seguente per recuperare la definizione del modello di asset esistente. Sostituisci *asset-model-id* con l'ID del modello di asset.

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

L'operazione restituisce una risposta contenente i dettagli del modello di asset.

- b. Crea un file denominato `update-asset-model-payload.json` e copia la risposta del comando precedente nel file.
- c. Rimuovete le seguenti coppie chiave-valore dal `update-asset-model-payload.json` file:
 - `assetModelId`
 - `assetModelArn`
 - `assetModelCreationDate`
 - `assetModelLastUpdateDate`
 - `assetModelStatus`
- d. Aggiungi la proprietà della fonte dell'allarme (`AWS/ALARM_SOURCE`) al modello composito di allarme che hai definito in precedenza. Sostituisci *alarmModelArn* con l'ARN del modello di allarme, che imposta il valore della proprietà della fonte di allarme.

```
{
  ...
  "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. Eseguite il comando seguente per aggiornare il modello di asset con la definizione memorizzata nel `update-asset-model-payload.json` file. Sostituisci *asset-model-id* con l'ID del modello di asset.

```

aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --cli-input-json file://update-asset-model-payload.json

```

Il tuo modello di asset ora definisce un allarme che rileva. AWS IoT Events L'allarme monitora la proprietà di destinazione in tutti gli asset in base a questo modello di asset. Puoi configurare l'allarme su ogni asset per personalizzare proprietà come la soglia o il destinatario IAM Identity Center per ogni asset. Per ulteriori informazioni, consulta [Configura gli allarmi sugli asset in AWS IoT SiteWise](#).

Definisci gli allarmi esterni in AWS IoT SiteWise

Gli allarmi esterni contengono lo stato di un allarme rilevato all'esterno. AWS IoT SiteWise

Definisci un allarme esterno (console)

È possibile utilizzare la AWS IoT SiteWise console per definire un allarme esterno su un modello di asset esistente. Per definire un allarme esterno su un nuovo modello di asset, create il modello di asset, quindi completate questi passaggi. Per ulteriori informazioni, consulta [Crea modelli di asset in AWS IoT SiteWise](#).

Per definire un allarme su un modello di asset

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione selezionare Models (Modelli).
3. Scegliete il modello di asset per il quale definire un allarme.
4. Scegliete la scheda Definizioni degli allarmi.
5. Scegli Aggiungi allarme.
6. Nelle opzioni del tipo di allarme, scegli Allarme esterno.
7. Immetti un nome per l'allarme.
8. (Facoltativo) Inserisci una descrizione per l'allarme.
9. Scegli Aggiungi allarme.

Definizione di un allarme esterno (CLI)

È possibile utilizzare il AWS CLI per definire un allarme esterno su un modello di asset nuovo o esistente.

Per aggiungere un allarme esterno a un modello di asset, è necessario aggiungere un modello composito di allarmi al modello di asset. Un modello composito di allarme esterno specifica il EXTERNAL tipo e non specifica una proprietà della fonte di allarme. L'esempio seguente di allarme composito definisce un allarme di temperatura esterna.

```
{
  ...
  "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": {}
          }
        }
      ]
    }
  ]
}
```

Per ulteriori informazioni su come aggiungere un modello composito a un modello di asset nuovo o esistente, consultate quanto segue:

- [Crea un modello di asset \(AWS CLI\)](#)
- [Aggiornate un modello di asset o componente \(AWS CLI\)](#)

Dopo aver definito l'allarme esterno, potete importare lo stato di allarme agli asset in base al modello di asset. Per ulteriori informazioni, consulta [Inserisci uno stato di allarme esterno in AWS IoT SiteWise](#).

Configura gli allarmi sugli asset in AWS IoT SiteWise

Dopo aver definito un AWS IoT Events allarme su un modello di asset, è possibile configurare l'allarme su ogni asset in base al modello di asset. È possibile modificare il valore di soglia e le impostazioni di notifica per l'allarme. Ciascuno di questi valori è un attributo della risorsa, quindi puoi aggiornare il valore predefinito dell'attributo per configurare questi valori.

Note

È possibile configurare questi valori per gli AWS IoT Events allarmi, ma non per gli allarmi esterni.

Argomenti

- [Configura un valore di soglia \(console\)](#)
- [Configura un valore di soglia \(AWS CLI\)](#)
- [Configura le impostazioni di notifica in AWS IoT SiteWise](#)

Configura un valore di soglia (console)

È possibile utilizzare la AWS IoT SiteWise console per aggiornare il valore dell'attributo che specifica il valore di soglia di un allarme.

Per aggiornare il valore di soglia di un allarme (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Asset.
3. Scegliete la risorsa per la quale desiderate aggiornare un valore di soglia di allarme.

Tip

Puoi scegliere l'icona a forma di freccia per espandere una gerarchia di asset e trovare il tuo asset.

4. Scegli Modifica.
5. Trova l'attributo che l'allarme utilizza per il suo valore di soglia, quindi inserisci il nuovo valore.
6. Scegli Save (Salva).

Configura un valore di soglia (AWS CLI)

È possibile utilizzare AWS Command Line Interface (AWS CLI) per aggiornare il valore dell'attributo che specifica il valore di soglia di un allarme.

Per completare questa procedura, è necessario conoscere l'elemento `assetId` dell'asset e l'elemento `propertyId` della proprietà. Puoi anche usare l'ID esterno. Se hai creato una risorsa e non la conosci `assetId`, utilizza l'[ListAssets](#) API per elencare tutte le risorse per un modello specifico. Utilizzate l'[DescribeAsset](#) operazione per visualizzare le proprietà della risorsa, inclusa la proprietà IDs.

L'operazione [BatchPutAssetPropertyValue](#) consente di assegnare i valori degli attributi all'asset. È possibile utilizzare questa operazione per impostare più attributi contemporaneamente. Il payload di questa operazione include un elenco di voci, ciascuna delle quali contenente l'ID asset, l'ID proprietà e il valore dell'attributo.

Per aggiornare il valore di un attributo (AWS CLI)

1. Crea un file denominato `batch-put-payload.json` e copia il seguente oggetto JSON nel file. Questo esempio di payload mostra come impostare la latitudine e la longitudine di una turbina eolica. Aggiorna i valori IDs, e i timestamp per modificare il payload per il tuo caso d'uso.

```
{
  "entries": [
    {
      "entryId": "windfarm3-turbine7-latitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-2222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-3333EXAMPLE",
      "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
        }
      }
    ]
  }
]
}

```

- Ogni voce nel payload contiene un `entryId` che è possibile definire come una qualsiasi stringa univoca. Se una richiesta non riesce, ciascun errore conterrà l'`entryId` della richiesta corrispondente in modo che sia possibile sapere quale richiesta riprovare.
- Per impostare il valore di un attributo, è possibile includere una struttura `timestamp-quality-value` (TQV) nell'elenco di proprietà di `propertyValues` ogni attributo. Questa struttura deve contenere il nuovo `value` e il `timestamp` corrente.
- `value`— Una struttura che contiene uno dei seguenti campi, a seconda del tipo di proprietà impostata:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`
- `timestamp`— Una struttura che contiene l'ora attuale dell'epoca Unix in secondi,. `timeInSeconds` AWS IoT SiteWise rifiuta tutti i punti dati con `timestamp` che esistevano da più di 7 giorni nel passato o più recenti di 5 minuti nelle future.

Per ulteriori informazioni su come preparare un payload per [BatchPutAssetPropertyValue](#), consulta [Inserisci dati con AWS IoT SiteWise APIs](#).

2. Esegui il comando seguente per inviare i valori degli attributi a: AWS IoT SiteWise

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

Configura le impostazioni di notifica in AWS IoT SiteWise

È possibile configurare le impostazioni di notifica degli allarmi utilizzando la AWS IoT SiteWise console o AWS Command Line Interface (AWS CLI).

Configurare le impostazioni di notifica (console)

È possibile utilizzare la AWS IoT SiteWise console per aggiornare il valore degli attributi che specificano le impostazioni di notifica per un allarme.

Per aggiornare le impostazioni di notifica di un avviso (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Asset.
3. Scegli la risorsa per la quale desideri aggiornare le impostazioni della sveglia.
4. Scegli Modifica.
5. Trova l'attributo utilizzato dall'allarme per l'impostazione di notifica che desideri modificare, quindi inserisci il nuovo valore.
6. Scegli Save (Salva).

Configurazione delle impostazioni di notifica (CLI)

È possibile utilizzare AWS Command Line Interface (AWS CLI) per aggiornare il valore dell'attributo che specifica le impostazioni di notifica per un avviso.

Per completare questa procedura, è necessario conoscere l'elemento `assetId` dell'asset e l'elemento `propertyId` della proprietà. Puoi anche usare l'ID esterno. Se hai creato una risorsa e non la conosci `assetId`, utilizza l'[ListAssets](#) API per elencare tutte le risorse per un modello specifico. Utilizzate l'[DescribeAsset](#) operazione per visualizzare le proprietà della risorsa, inclusa la proprietà IDs.

L'operazione [BatchPutAssetPropertyValue](#) consente di assegnare i valori degli attributi all'asset. È possibile utilizzare questa operazione per impostare più attributi contemporaneamente. Il payload di

questa operazione include un elenco di voci, ciascuna delle quali contenente l'ID asset, l'ID proprietà e il valore dell'attributo.

Per aggiornare il valore di un attributo (AWS CLI)

1. Crea un file denominato `batch-put-payload.json` e copia il seguente oggetto JSON nel file. Questo esempio di payload mostra come impostare la latitudine e la longitudine di una turbina eolica. Aggiorna i valori IDs, e i timestamp per modificare il payload per il tuo caso d'uso.

```
{
  "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
          }
        }
      ]
    }
  ]
}
```



```
}
```

- Ogni voce nel payload contiene un `entryId` che è possibile definire come una qualsiasi stringa univoca. Se una richiesta non riesce, ciascun errore conterrà l'`entryId` della richiesta corrispondente in modo che sia possibile sapere quale richiesta riprovare.
- Per impostare il valore di un attributo, è possibile includere una struttura `timestamp-quality-value` (TQV) nell'elenco di proprietà di `propertyValues` ogni attributo. Questa struttura deve contenere il nuovo `value` e il `timestamp` corrente.
- `value`— Una struttura che contiene uno dei seguenti campi, a seconda del tipo di proprietà impostata:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`
- `timestamp`— Una struttura che contiene l'ora attuale dell'epoca Unix in secondi, `timeInSeconds`. AWS IoT SiteWise rifiuta tutti i punti dati con `timestamp` che esistevano da più di 7 giorni nel passato o più recenti di 5 minuti nelle future.

Per ulteriori informazioni su come preparare un payload per [BatchPutAssetPropertyValue](#), consulta [Inserisci dati con AWS IoT SiteWise APIs](#).

2. Esegui il comando seguente per inviare i valori degli attributi a: AWS IoT SiteWise

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

Rispondi agli allarmi in AWS IoT SiteWise

Quando un AWS IoT Events allarme cambia stato, puoi fare quanto segue per rispondere all'allarme:

- Conferma un allarme per indicare che stai gestendo il problema.
- Posticipa una sveglia per disattivarla temporaneamente.
- Disattiva un allarme per disattivarlo in modo permanente fino a quando non lo riattivi.
- Abilita un allarme disabilitato per rilevare lo stato dell'allarme.

- Reimposta un allarme per cancellarne lo stato e l'ultimo valore.

Puoi utilizzare la AWS IoT SiteWise console o l' AWS IoT Events API per rispondere a un allarme.

Note

È possibile rispondere agli AWS IoT Events allarmi, ma non agli allarmi esterni.

Argomenti

- [Rispondere a un allarme \(console\)](#)
- [Rispondi a un allarme \(API\)](#)

Rispondere a un allarme (console)

È possibile utilizzare la AWS IoT SiteWise console per confermare, posticipare, disattivare o attivare un allarme.

Argomenti

- [Riconoscere un allarme \(console\)](#)
- [Posticipa una sveglia \(console\)](#)
- [Disattiva un allarme \(console\)](#)
- [Abilita un allarme \(console\)](#)
- [Reimpostazione di un allarme \(console\)](#)

Riconoscere un allarme (console)

Puoi confermare un allarme per indicare che stai gestendo il problema.

Note

È necessario abilitare il flusso di conferma sull'allarme in modo da poter confermare l'allarme. Questa opzione è abilitata di default se si definisce l'allarme dalla AWS IoT SiteWise console.

Per confermare un allarme (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Asset.
3. Scegliete la risorsa per la quale desiderate confermare un allarme.

Tip

Puoi scegliere l'icona a forma di freccia per espandere una gerarchia di asset e trovare il tuo asset.

4. Scegli la scheda Allarmi.
5. Seleziona l'allarme da confermare, quindi scegli Azioni per aprire il menu delle azioni di risposta.
6. Scegli Riconosci. Lo stato dell'allarme cambia in Riconosciuto.

Posticipa una sveglia (console)

È possibile posticipare una sveglia per disattivarla temporaneamente. Specificate la durata per cui posticipare la sveglia.

Per posticipare una sveglia (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Asset.
3. Scegliete la risorsa per la quale desiderate posticipare una sveglia.

Tip

Puoi scegliere l'icona a forma di freccia per espandere una gerarchia di asset e trovare il tuo asset.

4. Scegli la scheda Allarmi.
5. Seleziona la sveglia da posticipare, quindi scegli Azioni per aprire il menu delle azioni di risposta.
6. Scegli Snooze. Si apre un modello in cui si specifica la durata dello snooze.
7. Scegliete la lunghezza dello snooze o inserite una lunghezza dello snooze personalizzata.
8. Scegli Save (Salva). Lo stato della sveglia cambia in Snoozed.

Disattiva un allarme (console)

Puoi disabilitare un allarme in modo che non venga più rilevato. Dopo aver disattivato l'allarme, è necessario riattivarlo se si desidera che l'allarme venga rilevato.

Per disattivare un allarme (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Asset.
3. Scegliete la risorsa per cui desiderate disattivare un allarme.

Tip

Puoi scegliere l'icona a forma di freccia per espandere una gerarchia di asset e trovare il tuo asset.

4. Scegli la scheda Allarmi.
5. Seleziona l'allarme da disabilitare, quindi scegli Azioni per aprire il menu delle azioni di risposta.
6. Scegliere Disabilita. Lo stato dell'allarme cambia in Disabilitato.

Abilita un allarme (console)

È possibile abilitare nuovamente il rilevamento di un allarme dopo averlo disattivato o posticipato.

Per abilitare un allarme (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Asset.
3. Scegliete la risorsa per la quale desiderate attivare un allarme.

Tip

Puoi scegliere l'icona a forma di freccia per espandere una gerarchia di asset e trovare il tuo asset.

4. Scegli la scheda Allarmi.
5. Seleziona l'allarme da abilitare, quindi scegli Azioni per aprire il menu delle azioni di risposta.

6. Scegli Abilita . Lo stato dell'allarme passa a Normale.

Reimpostazione di un allarme (console)

È possibile reimpostare un allarme per cancellarne lo stato e l'ultimo valore.

Per resettare un allarme (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Asset.
3. Scegliete la risorsa per la quale desiderate reimpostare un allarme.

Tip

Puoi scegliere l'icona a forma di freccia per espandere una gerarchia di asset e trovare il tuo asset.

4. Scegli la scheda Allarmi.
5. Seleziona l'allarme da abilitare, quindi scegli Azioni per aprire il menu delle azioni di risposta.
6. Scegliere Reimposta. Lo stato dell'allarme passa a Normale.

Rispondi a un allarme (API)

Puoi utilizzare l' AWS IoT Events API per confermare, posticipare, disabilitare, abilitare o reimpostare un allarme. Per ulteriori informazioni, consulta le seguenti operazioni nell'AWS IoT Events API Reference:

- [BatchAcknowledgeAlarm](#)
- [BatchSnoozeAlarm](#)
- [BatchDisableAlarm](#)
- [BatchEnableAlarm](#)
- [BatchResetAlarm](#)

Per ulteriori informazioni, consulta [Rispondere agli allarmi nella Guida](#) per gli AWS IoT Events sviluppatori.

Inserisci uno stato di allarme esterno in AWS IoT SiteWise

Gli allarmi esterni sono allarmi che vengono valutati all'esterno. AWS IoT SiteWise. È possibile utilizzare allarmi esterni quando si dispone di una fonte di dati che riporta lo stato dell'allarme che si desidera importare. AWS IoT SiteWise

Le proprietà dello stato di allarme richiedono un formato specifico per i valori dei dati dello stato di allarme. Ogni valore di dati deve essere un oggetto JSON serializzato su una stringa. Quindi, si inserisce la stringa serializzata come valore di stringa. Per ulteriori informazioni, consulta [Proprietà dello stato di allarme](#).

Example Esempio di valore dei dati sullo stato di allarme (non serializzato)

```
{
  "stateName": "Active"
}
```

Example Esempio di valore dei dati sullo stato di allarme (serializzato)

```
{\"stateName\":\"Active\"}
```

Note

Se la tua fonte di dati non può riportare dati in questo formato o non puoi convertire i dati in questo formato prima di inserirli, puoi scegliere di non utilizzare una proprietà di allarme. Puoi invece importare i dati come proprietà di misurazione con il tipo di dati stringa, ad esempio. Per ulteriori informazioni, consultare [Definisci i flussi di dati provenienti dalle apparecchiature \(misurazioni\)](#) e [Inserisci dati in AWS IoT SiteWise](#).

Mappa i flussi di stato degli allarmi esterni in AWS IoT SiteWise

È possibile definire alias di proprietà per mappare i flussi di dati alle proprietà dello stato di allarme. Ciò consente di identificare facilmente una proprietà dello stato di allarme quando si inseriscono o recuperano dati. Per ulteriori informazioni sugli alias di proprietà, vedere. [Gestisci i flussi di dati per AWS IoT SiteWise](#)

Argomenti

- [Mappa i flussi di stato degli allarmi esterni \(console\)](#)

- [Mappa i flussi di stato degli allarmi esterni \(\)AWS CLI](#)

Mappa i flussi di stato degli allarmi esterni (console)

È possibile definire alias di proprietà per mappare i flussi di dati alle proprietà dello stato di allarme. Ciò consente di identificare facilmente una proprietà dello stato di allarme quando si inseriscono o recuperano dati. Per ulteriori informazioni sugli alias di proprietà, vedere. [Gestisci i flussi di dati per AWS IoT SiteWise](#)

È possibile utilizzare la AWS IoT SiteWise console per impostare un alias per una proprietà dello stato di allarme.

Per impostare un alias di proprietà per una proprietà dello stato di allarme (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Asset.
3. Scegli l'asset per cui vuoi impostare un alias di proprietà.

Tip

Puoi scegliere l'icona a forma di freccia per espandere una gerarchia di asset e trovare il tuo asset.

4. Scegli Modifica.
5. Scorri fino a Allarmi ed espandi la sezione.
6. In Allarmi esterni, inserisci l'alias in Alias di proprietà, facoltativo.
7. Scegli Save (Salva).

Mappa i flussi di stato degli allarmi esterni ()AWS CLI

È possibile definire alias di proprietà per mappare i flussi di dati alle proprietà dello stato di allarme. Ciò consente di identificare facilmente una proprietà dello stato di allarme quando si inseriscono o recuperano dati. Per ulteriori informazioni sugli alias di proprietà, vedere. [Gestisci i flussi di dati per AWS IoT SiteWise](#)

È possibile utilizzare AWS Command Line Interface (AWS CLI) per impostare un alias per una proprietà dello stato di allarme.

Per completare questa procedura, è necessario conoscere l'elemento `assetId` dell'asset e l'elemento `propertyId` della proprietà. È inoltre possibile utilizzare l'ID esterno. Se hai creato una risorsa e non la conosci `assetId`, utilizza l'[ListAssets](#) API per elencare tutte le risorse per un modello specifico. Utilizzate l'[DescribeAsset](#) operazione per visualizzare le proprietà della risorsa, inclusa la proprietà IDs.

Note

La [DescribeAsset](#) risposta include l'elenco di modelli di asset composti per l'asset. Ogni allarme è un modello composto. Per trovare il `propertyId`, trova il modello composto per l'allarme, quindi trova la `AWS/ALARM_STATE` proprietà in quel modello composto.

Per ulteriori informazioni su come impostare l'alias della proprietà, vedere [Aggiorna l'alias di proprietà di un asset](#).

Inserisci i dati sullo stato di allarme in AWS IoT SiteWise

Le proprietà dello stato di allarme prevedono che lo stato di allarme sia una stringa JSON serializzata. Per importare lo stato di allarme in un allarme esterno in AWS IoT SiteWise, si inserisce questa stringa serializzata come valore di stringa con data e ora. L'esempio seguente mostra un valore di dati di stato per un allarme attivo.

```
{\"stateName\":\"Active\"}
```

Per identificare una proprietà dello stato di allarme, puoi specificare una delle seguenti opzioni:

- La `assetId` fine `propertyId` della proprietà di allarme a cui stai inviando i dati.
- Il `propertyAlias`, che è un alias di flusso di dati (ad esempio, `/company/windfarm/3/turbine/7/temperature/high`). Per utilizzare questa opzione, devi prima impostare l'alias della proprietà di allarme. Per informazioni su come impostare gli alias di proprietà per le proprietà dello stato di allarme, vedi. [Mappa i flussi di stato degli allarmi esterni in AWS IoT SiteWise](#)

Il seguente esempio di [BatchPutAssetPropertyValue](#) API payload mostra come formattare lo stato di un allarme esterno. Questo allarme esterno segnala quando la lettura dei giri al minuto (RPM) di una turbina eolica è troppo alta.

Example Esempio di BatchPutAssetPropertyValue payload per i dati sullo stato di allarme

```
{
  "entries": [
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/temperature/high",
      "propertyValues": [
        {
          "value": {
            "stringValue": "{\"stateName\":\"Active\"}"
          },
          "timestamp": {
            "timeInSeconds": 1607550262
          }
        }
      ]
    }
  ]
}
```

Per ulteriori informazioni su come utilizzare l'BatchPutAssetPropertyValueAPI per importare dati, consulta. [Inserisci dati con AWS IoT SiteWise APIs](#)

Per ulteriori informazioni su altri modi di importare dati, consulta. [Inserisci dati in AWS IoT SiteWise](#)

AWS IoT SiteWise Assistente

L' AWS IoT SiteWise Assistente è un assistente generativo basato sull'intelligenza artificiale. Consente a utenti come i responsabili degli impianti, gli ingegneri della qualità e i tecnici di manutenzione di ottenere informazioni, risolvere problemi e intraprendere azioni direttamente dai propri dati operativi e aziendali.

L' AWS IoT SiteWise Assistente consolida le informazioni provenienti da AWS IoT dati, modelli di asset, manuali e documentazione in riepiloghi comprensibili degli eventi critici. Consente inoltre sessioni interattive di approfondimento con domande e risposte per facilitare la diagnosi, l'esplorazione delle cause alla radice e consigli guidati.

Argomenti

- [Configura l'Assistente AWS IoT SiteWise](#)
- [Creazione di un set di dati](#)
- [Modifica un set di dati](#)
- [Eliminare un set di dati](#)
- [AWS IoT SiteWise Domande dell'assistente](#)

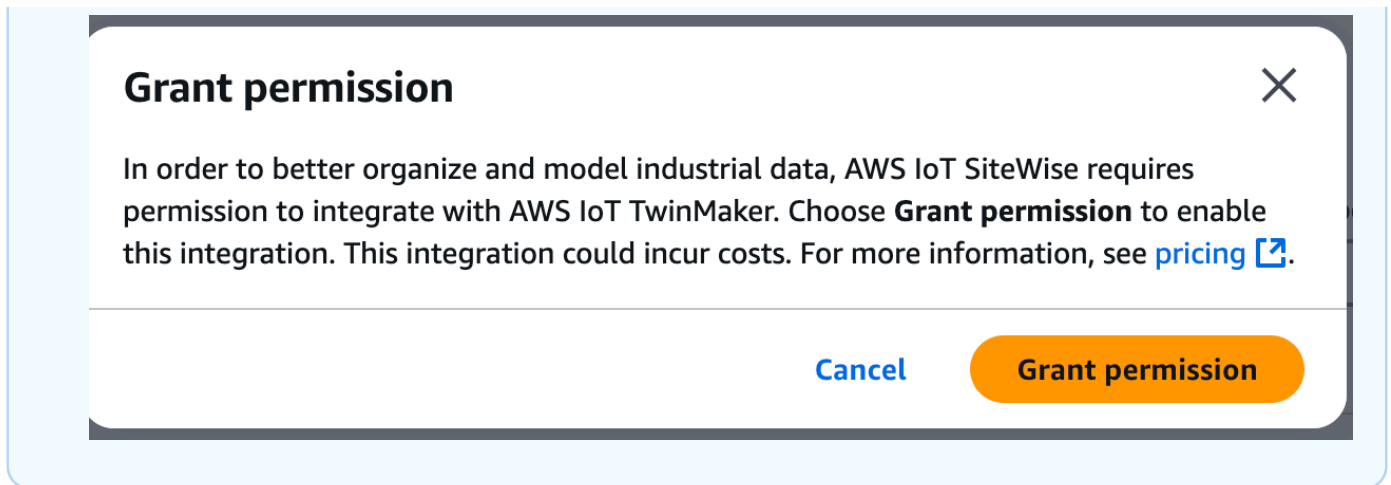
Configura l'Assistente AWS IoT SiteWise

AWS IoT SiteWise Configurazione dell'assistente

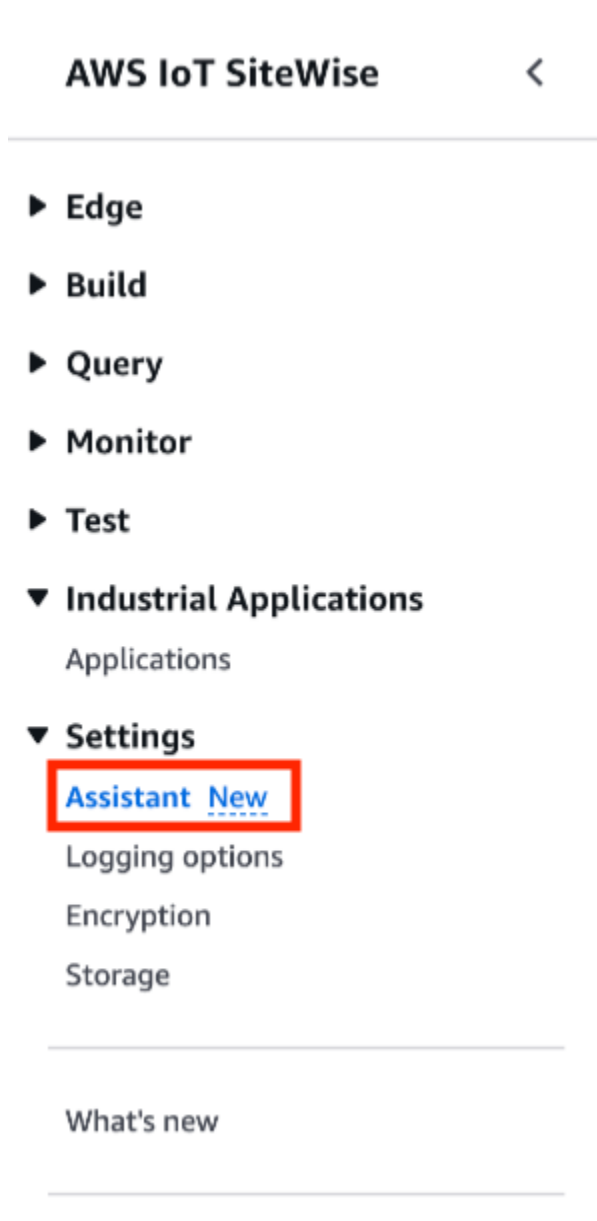
1. Accedi alla [console AWS IoT SiteWise](#).

Note

Concedi le autorizzazioni per abilitare l'integrazione con il AWS IoT TwinMaker servizio. Ciò è necessario per consentire all' AWS IoT SiteWise Assistente e al dashboard di eseguire le query SQL nelle AWS IoT SiteWise risorse. Vedi [Integrate AWS IoT SiteWise and. AWS IoT TwinMaker](#)



2. Scegli Assistente dal pannello di navigazione a sinistra.



Creazione di un set di dati

Note

L' AWS IoT SiteWise Assistente deve utilizzare un set di dati con un indice [Amazon Kendra](#) per conoscenze e indicazioni a livello aziendale. Se non disponi di un indice Amazon Kendra, [consulta Creazione di un indice per crearne uno](#). L'aggiunta di un [set](#) di dati migliora la qualità della risposta dell'Assistente e riduce al minimo le allucinazioni.

Console

Crea un set di dati nella console AWS IoT SiteWise

1. I set di dati vengono visualizzati nella sezione Datasets della pagina Assistente AWS IoT SiteWise .
2. Se non esistono set di dati, scegli Crea set di dati.
3. Nella pagina dei dettagli del set di dati, scegli un indice Kendra dal menu a discesa da associare al set di dati.
4. Il nome del set di dati viene compilato dall'indice Kendra selezionato nel passaggio 3. Modifica il nome se necessario.
5. (Facoltativo) La descrizione del set di dati viene compilata dall'indice Kendra selezionato nel passaggio 3. Modifica la descrizione se necessario.
6. Nella sezione Autorizzazioni, scegli una delle seguenti opzioni:
 - a. Scegli Crea e utilizza un nuovo ruolo di servizio. Per impostazione predefinita, crea AWS IoT SiteWise automaticamente un ruolo di servizio. Questo ruolo consente all' AWS IoT SiteWise Assistente di accedere ai tuoi indici Kendra.
 - b. Scegli Usa un ruolo di servizio esistente, quindi scegli il ruolo di destinazione.
7. Scegli Create (Crea) .

[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

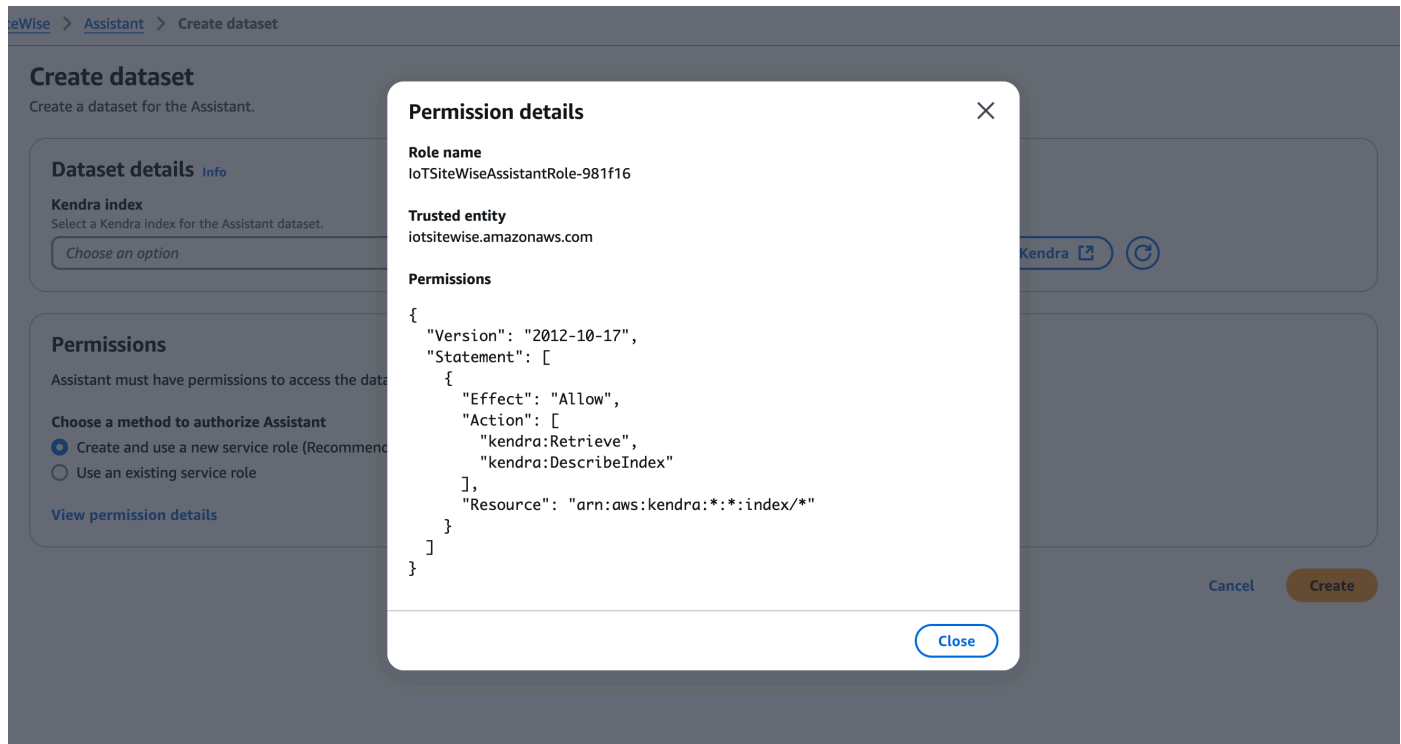
- ☒ Create and use a new service role - *recommended*
- ☐ Use an existing service role

[View permission details](#)

[Cancel](#)

[Create](#)

Il ruolo di servizio creato da AWS IoT SiteWise per l'utente, se l'utente ha scelto di creare e utilizzare un nuovo ruolo di servizio.



AWS CLI

Crea un set di dati in AWS CLI

1. Crea un ruolo IAM utilizzato per creare un set di dati. Utilizza la seguente politica di autorizzazioni:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kendra:Retrieve"
      ],
      "Resource": "arn:aws:kendra:*:*:index/*"
    }
  ]
}
```

Utilizza la seguente relazione di fiducia:

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "iotsitewise.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Crea un file `create-dataset.json` con il modello fornito nell'esempio. Compila e connettiti con questo set di `datasetId` `datikendra` `knowledgeBaseArn`. `roleArn`

```
{
  "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. Crea il set di dati con il seguente comando:

```
aws iotsitewise create-dataset --cli-input-json file://create-dataset.json --
region us-east-1
```


Modifica un set di dati

Console

Modifica un set di dati

1. I set di dati vengono visualizzati nella sezione Datasets della pagina Assistente. Scegli un set di dati da modificare. Scegli Modifica per iniziare a modificare.
2. Nella pagina dei dettagli del set di dati, scegli un indice Kendra dal menu a discesa da associare al set di dati.
3. Il nome del set di dati viene compilato dall'indice Kendra selezionato nel passaggio 2. Modifica il nome se necessario.
4. (Facoltativo) La descrizione del set di dati viene compilata dall'indice Kendra selezionato nel passaggio 2. Modifica la descrizione se necessario.
5. Nella sezione Autorizzazioni, scegli una delle seguenti opzioni:
 - a. Scegli Crea e utilizza un nuovo ruolo di servizio. Per impostazione predefinita, crea AWS IoT SiteWise automaticamente un ruolo di servizio. Questo ruolo consente all' AWS IoT SiteWise Assistente di accedere ai tuoi indici Kendra.
 - b. Scegli Usa un ruolo di servizio esistente, quindi scegli il ruolo di destinazione.
6. Scegli Salva modifiche per salvare la selezione.

[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

Modifica un set di dati in AWS CLI

1. Crea un file update-dataset.json con il modello fornito nell'esempio. Compila e connettiti con questo set di datasetId datikendra knowledgeBaseArn. roleArn

```
{
  "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. Aggiorna il set di dati con il seguente comando:

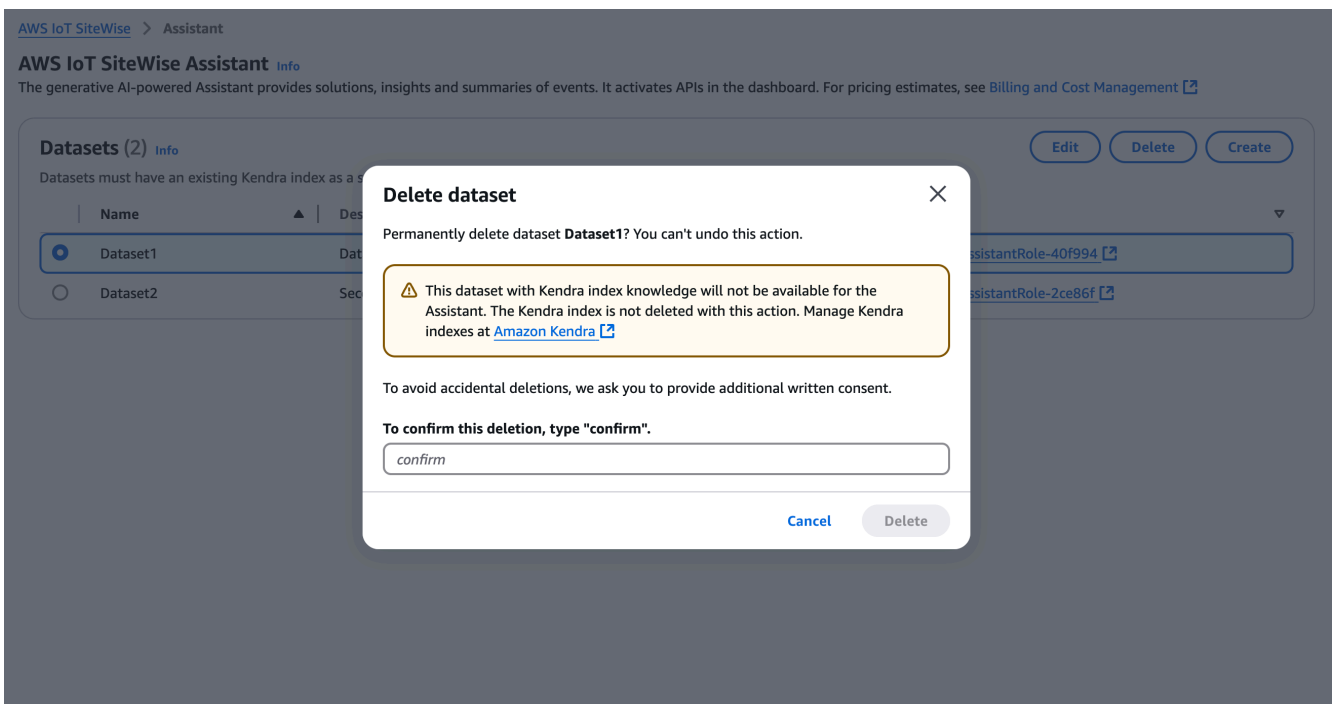
```
aws iotsitewise update-dataset --cli-input-json file://update-dataset.json --  
region us-east-1
```

Eliminare un set di dati

Console

Eliminare un set di dati

1. I set di dati vengono visualizzati nella sezione Datasets della pagina Assistente. Scegli un set di dati. Scegli Elimina.
2. Digita confirm nel popup per confermare l'eliminazione.



3. Scegli Elimina.

AWS CLI

Eliminare un set di dati

- Elimina il set di dati con. datasetId

```
aws iotsitewise delete-dataset --region us-east-1 --dataset-id <UUID>
```

AWS IoT SiteWise Domande dell'assistente

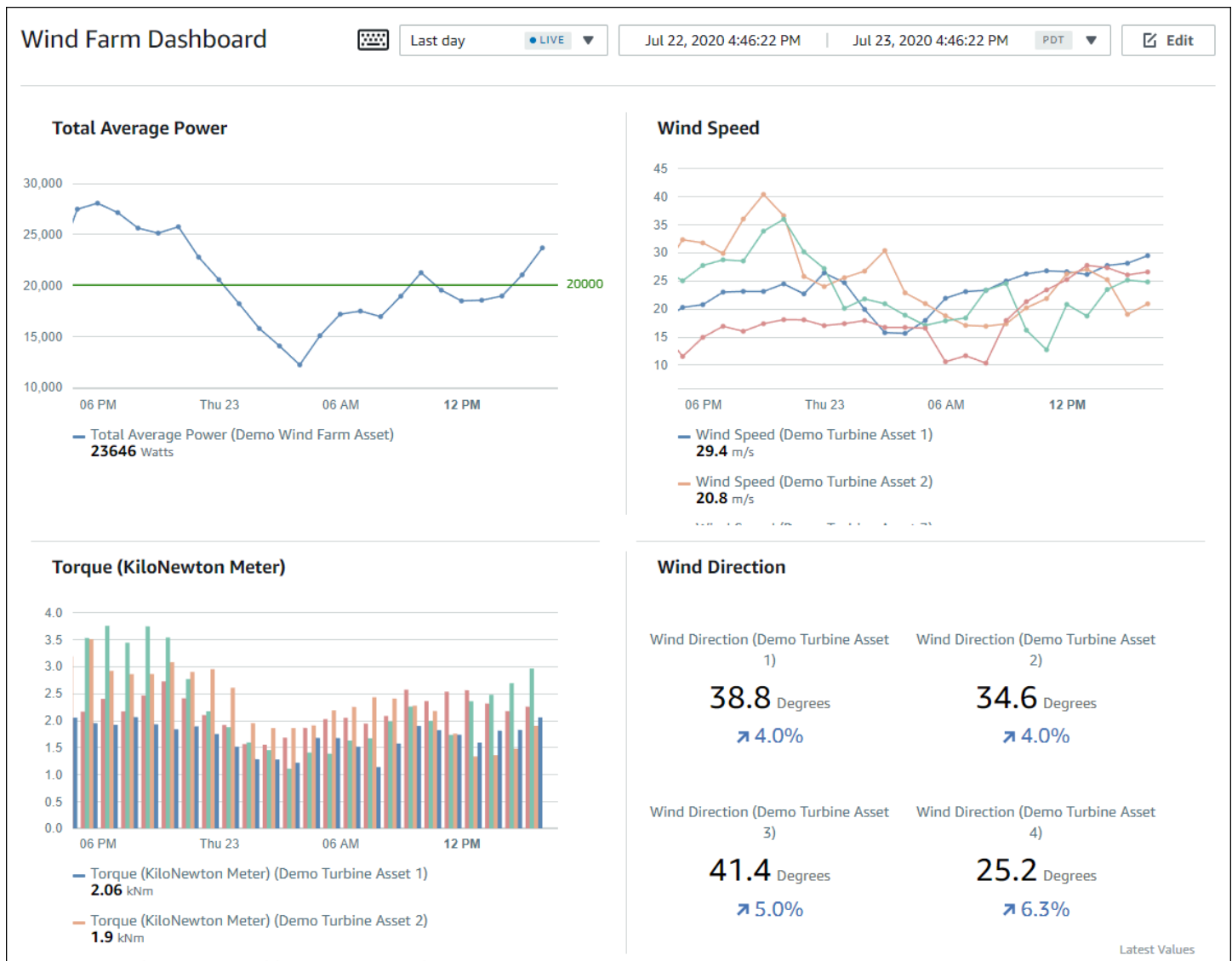
Vedi [Esempi di domande da porre all'Assistente AWS IoT SiteWise](#) per saperne di più su come interrogare AWS IoT SiteWise Assistant.

Monitora i dati con AWS IoT SiteWise Monitor

Puoi utilizzarli AWS IoT SiteWise per monitorare i dati provenienti da processi, dispositivi e apparecchiature creando portali web SiteWise Monitor. SiteWise Monitor è una funzionalità AWS IoT SiteWise che puoi utilizzare per creare portali sotto forma di un'applicazione web gestita, utili a visualizzare e condividere i propri dati operativi. Puoi creare dei progetti con i pannelli di controllo per visualizzare i dati di processi, dispositivi e apparecchiature connessi a AWS IoT.

I portali consentono agli esperti dei domini, quali gli ingegneri di processo, di reperire immediatamente informazioni sui dati operativi e comprendere il comportamento dei dispositivi e delle apparecchiature.

Di seguito è riportato un pannello di controllo esemplificativo che mostra i dati relativi a un parco eolico.



Poiché AWS IoT SiteWise acquisisce i dati nel tempo, puoi utilizzare SiteWise Monitor per visualizzare i dati operativi nel tempo o gli ultimi valori riportati in momenti specifici. e, di conseguenza, notare o reperire andamenti o informazioni altrimenti difficili da rilevare.

SiteWise Monitora i ruoli

Quattro ruoli interagiscono con SiteWise Monitor:

AWS amministratore

L' AWS amministratore utilizza la AWS IoT SiteWise console per creare portali. L'amministratore AWS può anche assegnare gli amministratori per il portale e aggiungere gli utenti del portale.

Gli amministratori del portale assegnano successivamente gli utenti del portale ai progetti come proprietari o visualizzatori. L' AWS amministratore lavora esclusivamente nella AWS console.

Amministratore del portale

Ogni portale SiteWise Monitor ha uno o più amministratori del portale. Gli amministratori del portale utilizzano il portale per creare progetti che contengono raccolte di asset e pannelli di controllo. L'amministratore del portale assegna quindi asset e proprietari a ciascun progetto. Controllando l'accesso al progetto, gli amministratori del portale specificano quali asset possono vedere i proprietari e i visualizzatori del progetto.

Proprietario del progetto

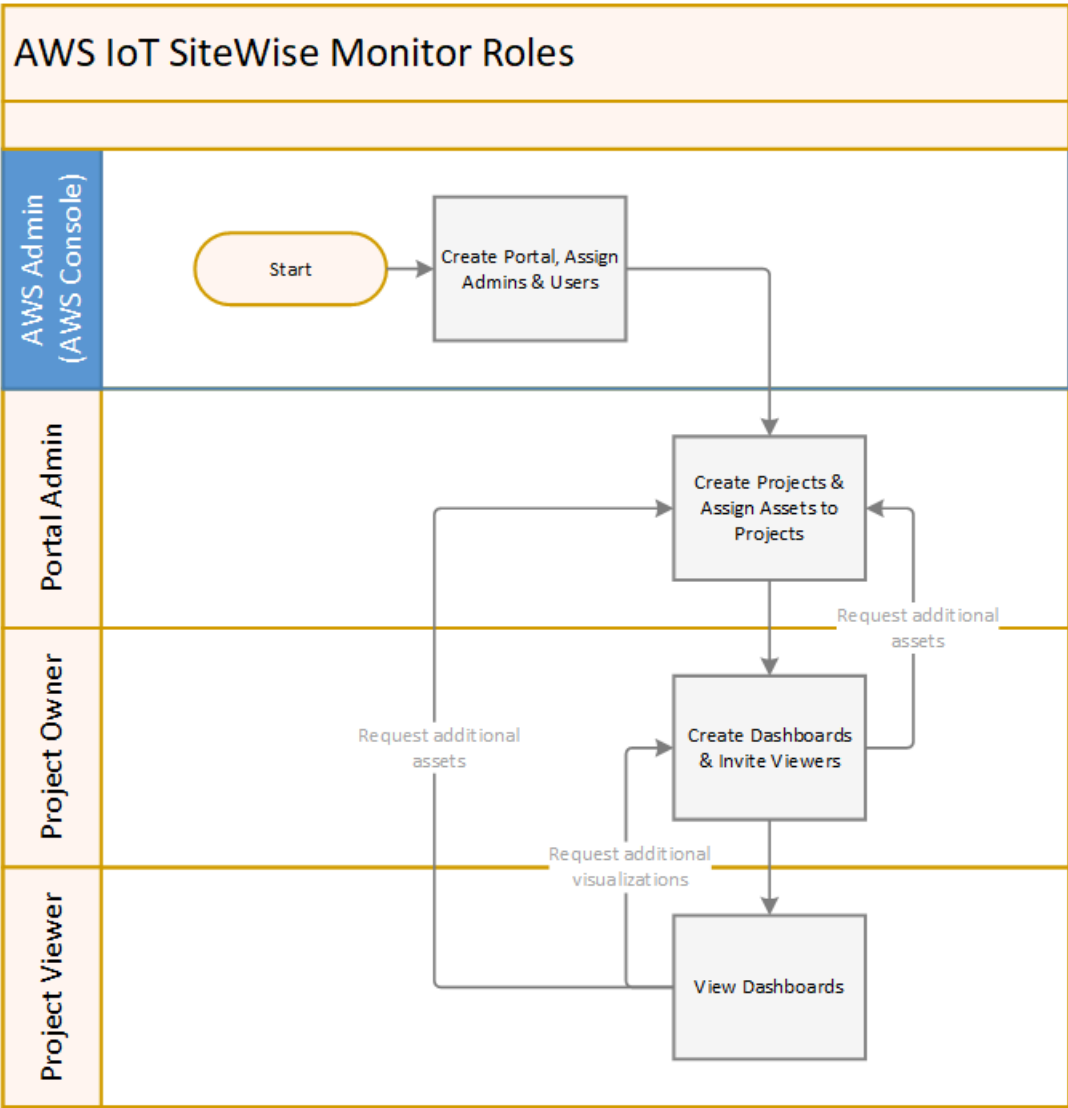
Ogni progetto SiteWise Monitor ha dei proprietari. I proprietari dei progetti creano visualizzazioni sotto forma di pannelli di controllo al fine di rappresentare i dati operativi in modo coerente. Una volta che i pannelli di controllo sono pronti per la condivisione, il proprietario del progetto può invitare i visualizzatori al progetto. I proprietari del progetto possono anche assegnare altri proprietari al progetto. I proprietari dei progetti possono configurare le soglie e le impostazioni di notifica per gli allarmi.

Visualizzatore del progetto

Ogni progetto SiteWise Monitor ha dei visualizzatori. I visualizzatori del progetto possono connettersi al portale per visualizzare i pannelli di controllo creati dai proprietari del progetto. In ogni dashboard, i visualizzatori dei progetti possono regolare l'intervallo di tempo per comprendere meglio i dati operativi. I visualizzatori del progetto possono visualizzare solo i pannelli di controllo nei progetti a cui hanno accesso. I visualizzatori del progetto possono confermare e posticipare gli allarmi.

A seconda dell'organizzazione, la stessa persona potrebbe svolgere più ruoli.

L'immagine seguente illustra come questi quattro ruoli interagiscono nel portale Monitor. SiteWise



Puoi gestire chi ha accesso ai tuoi dati utilizzando AWS IAM Identity Center o IAM. Gli utenti dei tuoi dati possono accedere a SiteWise Monitor da un browser desktop o mobile utilizzando le proprie credenziali IAM Identity Center o IAM.

Federazione SAML

IAM Identity Center e IAM supportano la federazione delle identità con [SAML \(Security Assertion Markup Language\) 2.0](#). SAML 2.0 è uno standard aperto utilizzato da molti provider di identità esterni (IdPs) per autenticare gli utenti e trasmettere le informazioni relative all'identità e alla sicurezza ai provider di servizi (). SPs SPs sono in genere applicazioni o servizi. La federazione SAML consente agli amministratori e agli utenti del portale SiteWise Monitor di accedere ai portali assegnati con credenziali esterne, come nomi utente e password aziendali.

Puoi configurare IAM Identity Center e IAM per utilizzare la federazione basata su SAML per l'accesso ai portali Monitor. SiteWise

Centro identità IAM

Gli amministratori e gli utenti del portale possono accedere al portale di AWS accesso con i loro nomi utente e password aziendali. Possono quindi accedere ai portali Monitor assegnati SiteWise . IAM Identity Center utilizza i certificati per impostare una relazione di fiducia SAML tra il tuo provider di identità e. AWS Per ulteriori informazioni, consulta il [profilo SCIM e l'implementazione SAML 2.0](#) nella Guida per l'AWS IAM Identity Center utente.

IAM

Gli amministratori e gli utenti del portale possono richiedere credenziali di sicurezza temporanee per accedere ai portali Monitor loro assegnati. SiteWise Crei un'identità di provider di identità SAML in IAM per impostare una relazione di fiducia tra il tuo provider di identità e. AWS Per ulteriori informazioni, consulta [Utilizzo della federazione basata su SAML per l'accesso alle API AWS](#), nella Guida per l'utente IAM.

Gli amministratori e gli utenti del portale possono accedere al portale aziendale e selezionare l'opzione per accedere alla AWS console di gestione. Possono quindi accedere ai portali SiteWise Monitor assegnati. Il portale della tua azienda gestisce lo scambio di fiducia tra il tuo provider di identità e AWS. Per ulteriori informazioni, consulta [Abilitare gli utenti federati SAML 2.0 ad accedere alla Console di AWS gestione nella Guida](#) per l'utente IAM.

Note

Quando aggiungi utenti o amministratori al portale, evita di creare policy IAM che limitino le autorizzazioni degli utenti, come un IP limitato. Qualsiasi policy allegata con autorizzazioni limitate non sarà in grado di connettersi al portale. AWS IoT SiteWise

SiteWise Monitora i concetti

Per utilizzare SiteWise Monitor, è necessario conoscere i seguenti concetti:

Portal

Un AWS IoT SiteWise Monitor portale è un'applicazione web che è possibile utilizzare per visualizzare e condividere AWS IoT SiteWise i dati. Un portale ha uno o più amministratori e contiene zero o più progetti.

Progetto

Ogni portale SiteWise Monitor contiene una serie di progetti. Ogni progetto ha un sottoinsieme di asset AWS IoT SiteWise associati. I proprietari del progetto creano uno o più pannelli di controllo per fornire un modo coerente per visualizzare i dati associati agli asset. I proprietari del progetto possono invitare i visualizzatori al progetto per consentire loro di visualizzare gli asset e i pannelli di controllo del progetto. Il progetto è l'unità di condivisione di base all'interno di SiteWise Monitor. I proprietari del progetto possono invitare gli utenti a cui l'AWS amministratore ha concesso l'accesso al portale. Un utente deve avere accesso a un portale prima che un progetto del portale possa essere condiviso con l'utente.

Asset

Quando i dati vengono importati AWS IoT SiteWise dalle apparecchiature industriali, i dispositivi, le apparecchiature e i processi vengono rappresentati ciascuno come risorse. A ogni risorsa sono associati proprietà e allarmi. L'amministratore del portale assegna i set di asset a ciascun progetto.

Proprietà

Le proprietà sono dati di serie temporali associati agli asset. Ad esempio, un pezzo di apparecchiatura può avere un numero di serie, una posizione, una marca e un modello e una data di installazione. Può anche avere valori di serie temporali per disponibilità, prestazioni, qualità, temperatura, pressione e così via.

Allarme

Gli allarmi monitorano le proprietà per identificare quando l'apparecchiatura si trova al di fuori del suo intervallo operativo. Ogni allarme definisce una soglia e una proprietà da monitorare. Quando la proprietà supera la soglia, l'allarme diventa attivo e indica che tu o qualcuno del tuo team dovete risolvere il problema. I proprietari del progetto possono personalizzare le soglie e le impostazioni di notifica per gli allarmi. I visualizzatori del progetto possono confermare e posticipare gli allarmi e possono lasciare un messaggio con i dettagli sull'allarme o sull'azione che hanno intrapreso per risolverlo.

Dashboard

Ogni progetto contiene un set di pannelli di controllo. I pannelli di controllo forniscono un set di visualizzazioni per i valori di un set di asset. I proprietari del progetto creano i pannelli di controllo e le visualizzazioni contenute. Quando il proprietario di un progetto è pronto a condividere il set di pannelli di controllo, il proprietario può invitare i visualizzatori al progetto, consentendo loro di accedere a tutti i pannelli di controllo del progetto. Se si desidera un set diverso di visualizzatori per pannelli di controllo diversi, è necessario dividere i pannelli di controllo tra i progetti. Quando gli spettatori guardano le dashboard, possono personalizzare l'intervallo di tempo per esaminare dati specifici.

Visualizzazione

In ogni dashboard, i proprietari del progetto decidono come visualizzare le proprietà e gli allarmi delle risorse associate al progetto. La disponibilità potrebbe essere rappresentata come un grafico a linee, mentre altri valori potrebbero essere visualizzati come grafici a barre o indicatori chiave di prestazione (KPIs). Gli allarmi vengono visualizzati al meglio come griglie di stato e linee temporali di stato. I proprietari del progetto personalizzano ogni visualizzazione per fornire la migliore comprensione dei dati per l'asset.

Inizia con AWS IoT SiteWise Monitor (Classic)

Se sei l'AWS amministratore della tua organizzazione, crei portali dalla AWS IoT SiteWise console. Completa i seguenti passaggi per creare un portale in modo che i membri della tua organizzazione possano visualizzare AWS IoT SiteWise i tuoi dati:

1. Configurare e creare un portale
2. Aggiungere amministratori del portale e inviare e-mail di invito
3. Aggiungere utenti del portale

Dopo aver creato un portale, l'amministratore del portale può visualizzare le AWS IoT SiteWise risorse e assegnarle ai progetti del portale. A questo punto, i proprietari di progetto possono creare pannelli di controllo per la visualizzazione delle proprietà degli asset che consentano ai visualizzatori di progetto di appurare le prestazioni di dispositivi, processi e apparecchiature.

 Note

Quando aggiungete utenti o amministratori al portale, evitate di creare policy AWS Identity and Access Management (IAM) che limitino le autorizzazioni degli utenti, come l'IP limitato. Qualsiasi policy allegata con autorizzazioni limitate non sarà in grado di connettersi al portale. AWS IoT SiteWise

Puoi seguire un tutorial che illustra i passaggi necessari per configurare un portale con un progetto, pannelli di controllo e diversi utenti per uno scenario specifico utilizzando i dati della centrale eolica. Per ulteriori informazioni, consulta [Visualizza e condividi i dati dei parchi eolici in Monitor SiteWise](#).

Argomenti

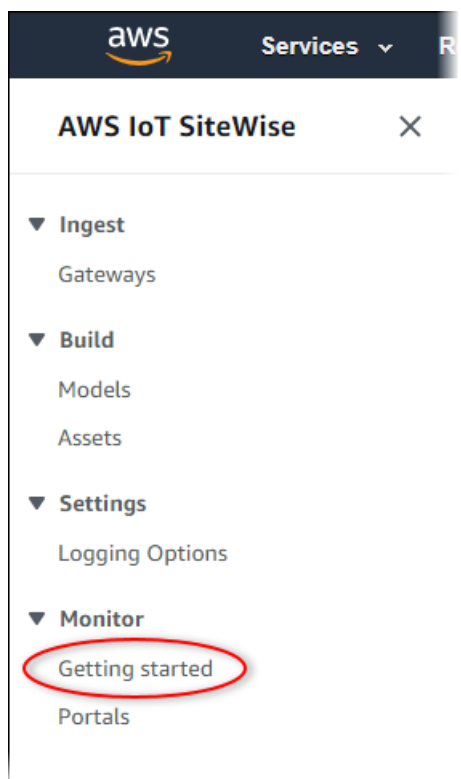
- [Crea un portale in Monitor SiteWise](#)
- [Configura il tuo portale in SiteWise Monitor](#)
- [Invita gli amministratori in Monitor SiteWise](#)
- [Aggiungere utenti del portale in Monitor SiteWise](#)
- [Creare AWS IoT SiteWise dashboard \(\)AWS CLI](#)
- [Attiva gli allarmi per i tuoi portali in AWS IoT SiteWise](#)
- [Attivazione del AWS IoT SiteWise portale all'edge](#)
- [Amministra i tuoi portali SiteWise Monitor](#)

Crea un portale in Monitor SiteWise

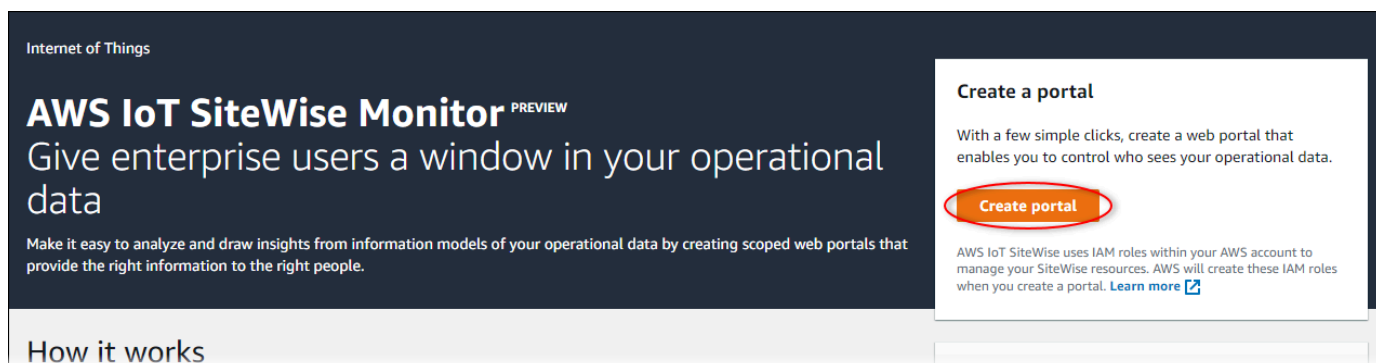
Crei un portale SiteWise Monitor nella AWS IoT SiteWise console.

Per creare un portale

1. Accedere alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, selezionare Monitor (Monitoraggio), Getting started (Operazioni di base).



3. Selezionare Create Portal (Crea portale).



Successivamente, bisogna fornire alcune informazioni di base per la configurazione del portale.

Configura il tuo portale in SiteWise Monitor

Gli utenti utilizzano i portali per visualizzare i tuoi dati. Puoi personalizzare il nome, la descrizione, il marchio, l'autenticazione dell'utente, l'email di contatto dell'assistenza e le autorizzazioni di un portale.

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.

Name should be 1-128 characters and only contain A-Z a-z 0-9 _ and -.

Description - optional

Create a description of your portal

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.

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.

☒ 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

Tags

This resource doesn't have any tags.

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](#)

Per configurare un portale

1. Immettere un nome da assegnare al portale.
2. (Facoltativo) Inserire una descrizione del portale. Se si dispone di più portali, occorre utilizzare descrizioni significative per tenere traccia dei contenuti di ciascuno.
3. (Facoltativo) Caricare un'immagine per visualizzare il marchio nel portale. Scegliere un'immagine PNG quadrata. Se si carica un'immagine non quadrata, il portale ridimensiona l'immagine in un quadrato.
4. Selezionare una delle seguenti opzioni:
 - Scegli IAM Identity Center se gli utenti del tuo portale accedono a questo portale con i loro nomi utente e password aziendali.

Se non hai abilitato IAM Identity Center nel tuo account, procedi come segue:

- a. Selezionare Create user (Crea utente).
- b. Nella pagina Crea utente, per creare il primo portale, inserisci l'indirizzo e-mail, il nome e il cognome dell'utente, quindi scegli Crea utente.

Note

- AWS abilita automaticamente IAM Identity Center nel tuo account quando crei il primo utente del portale.
- Puoi configurare IAM Identity Center in una sola regione alla volta. SiteWise Monitor si connette alla regione che hai configurato per IAM Identity Center.

Ciò significa che utilizzi una regione per l'accesso a IAM Identity Center, ma puoi creare portali in qualsiasi regione.

- Scegli IAM se gli utenti del portale accedono a questo portale con le proprie credenziali IAM.

 Important

Gli utenti o i ruoli devono disporre dell'`iot:DescribePortal` autorizzazione per accedere al portale.

5. Immettere un indirizzo e-mail che gli utenti del portale possano contattare, in caso di dubbi o problemi, per trovare una soluzione.
6. (Facoltativo) Aggiungere tag per il portale. Per ulteriori informazioni, consulta [Tagga le tue AWS IoT SiteWise risorse](#).
7. Selezionare una delle seguenti opzioni:
 - Scegli Crea e utilizza un nuovo ruolo di servizio. Per impostazione predefinita, SiteWise Monitor crea automaticamente un ruolo di servizio per ogni portale. Questo ruolo consente agli utenti del portale di accedere alle AWS IoT SiteWise risorse dell'utente. Per ulteriori informazioni, consulta [Utilizza i ruoli di servizio per AWS IoT SiteWise Monitor](#).
 - Scegli Usa un ruolo di servizio esistente, quindi scegli il ruolo di destinazione.
8. Seleziona Next (Successivo).
9. (Facoltativo) Abilita gli allarmi per il tuo portale. Per ulteriori informazioni, consulta [Attiva gli allarmi per i tuoi portali in AWS IoT SiteWise](#).
10. Scegli Crea. AWS IoT SiteWise creerà il tuo portale.

 Note

Se si chiude la console, è possibile completare la procedura di configurazione aggiungendo amministratori e utenti. Per ulteriori informazioni, consulta [Aggiungi o rimuovi gli amministratori del portale in AWS IoT SiteWise](#). Se non desideri conservare questo portale, eliminalo in modo che non utilizzi risorse. Per ulteriori informazioni, consulta [Eliminare un portale in AWS IoT SiteWise](#).

La colonna Status può avere uno dei seguenti valori.

- **CREAZIONE:** AWS IoT SiteWise sta elaborando la richiesta di creazione del portale. Il completamento di questo processo può richiedere alcuni minuti.
- **AGGIORNAMENTO:** AWS IoT SiteWise sta elaborando la richiesta di aggiornamento del portale. Il completamento di questo processo può richiedere alcuni minuti.
- **IN SOSPEO:** AWS IoT SiteWise è in attesa del termine della propagazione dei record DNS. Il completamento di questo processo può richiedere diversi minuti. È possibile eliminare il portale mentre lo stato è IN SOSPEO.
- **ELIMINAZIONE:** AWS IoT SiteWise sta elaborando la richiesta di eliminazione del portale. Il completamento di questo processo può richiedere alcuni minuti.
- **ATTIVO:** quando il portale diventa attivo, gli utenti del portale possono accedervi.
- **FALLITO:** AWS IoT SiteWise impossibile elaborare la richiesta di creazione, aggiornamento o eliminazione del portale. Se hai abilitato AWS IoT SiteWise l'invio di log ad Amazon CloudWatch Logs, puoi utilizzare questi log per risolvere i problemi. [Per ulteriori informazioni, consulta *Monitoring with Logs. AWS IoT SiteWise CloudWatch*](#)

Una volta creato il portale, viene visualizzato un messaggio.

A green banner with a white checkmark icon on the left and a white 'X' icon on the right. The text in the center reads: "Successfully created portal URL at https://a1b2c3d4-5678-90ab-cdef-1111EXAMPLE.app.iotsitewise.aws".

Successfully created portal URL at https://a1b2c3d4-5678-90ab-cdef-1111EXAMPLE.app.iotsitewise.aws

Successivamente, bisogna invitare al portale uno o più utenti che fungano da amministratori. A questo punto, è stato creato un portale al quale nessuno, però, può accedere.

Invita gli amministratori in Monitor SiteWise

Per iniziare a usare il nuovo portale, è necessario assegnare un amministratore del portale. L'amministratore del portale crea i progetti, ne sceglie i proprietari e vi assegna gli asset necessari. Gli amministratori del portale possono vedere tutte le tue AWS IoT SiteWise risorse.

In base al servizio di autenticazione utente, scegli una delle seguenti opzioni:

IAM Identity Center

Se utilizzi SiteWise Monitor per la prima volta, puoi scegliere l'utente che hai creato in precedenza come amministratore del portale. Se desideri aggiungere un altro utente come amministratore del portale, puoi creare un utente IAM Identity Center da questa pagina. In alternativa, puoi

connettere un provider di identità esterno a IAM Identity Center. Per ulteriori informazioni, consulta la [Guida per l'utente AWS IAM Identity Center](#).

Come invitare gli amministratori

1. Selezionare le caselle di controllo riferite agli utenti da nominare amministratori del portale. Questo aggiunge gli utenti all'elenco degli amministratori del portale.

 Note

Se utilizzi IAM Identity Center come archivio di identità e hai effettuato l'accesso al tuo account di AWS Organizations gestione, puoi scegliere Crea utente per creare un utente IAM Identity Center. IAM Identity Center invia al nuovo utente un'e-mail per consentirgli di impostare la password. È quindi possibile assegnare l'utente al portale come amministratore. Per ulteriori informazioni, consulta la sezione [Gestione delle identità in IAM Identity Center](#).

2. Facoltativamente, è possibile avvalersi del comando Send invite to selected users (Manda un invito agli utenti selezionati). Viene visualizzato il client di posta elettronica e nel corpo del messaggio viene popolato un invito.

È possibile personalizzare il messaggio e-mail prima di inviarlo agli amministratori del portale. È inoltre possibile inviare l'e-mail agli amministratori del portale in un secondo momento. Se stai provando SiteWise Monitor per la prima volta e aggiungi il tuo nuovo utente o ruolo IAM Identity Center o IAM come amministratore del portale, non devi inviarti un'e-mail.

3. Se si aggiunge un utente indesiderato come amministratore, per risolvere basta deselectionarne la casella di controllo.
4. Una volta invitati gli amministratori del portale, bisogna selezionare Next (Avanti).

IAM

Puoi scegliere un utente o un ruolo come amministratore del portale. Se desideri aggiungere un altro utente o ruolo come amministratore del portale, puoi creare un utente o un ruolo nella console IAM. Per ulteriori informazioni, consulta [Creazione di un utente IAM nel tuo AWS account](#) e [Creazione di ruoli IAM](#) nella Guida per l'utente IAM.

Come invitare gli amministratori

1. Esegui questa operazione:
 - Scegli gli utenti IAM per aggiungere un utente IAM come amministratore del portale.
 - Scegli i ruoli IAM per aggiungere un ruolo IAM come amministratore del portale.
2. Seleziona le caselle di controllo per gli utenti o i ruoli che desideri come amministratori del portale. In questo modo gli utenti o i ruoli vengono aggiunti all'elenco degli amministratori del portale.
3. Se aggiungi un utente o un ruolo che non desideri utilizzare come amministratore, deseleziona la casella di controllo relativa a quell'utente o ruolo.
4. Una volta invitati gli amministratori del portale, bisogna selezionare Next (Avanti).

Important

Gli utenti o i ruoli devono disporre dell'`iotsitewise:DescribePortalautorizzazione` per accedere al portale.

Note

Se utilizzi IAM Identity Center come archivio di identità e hai effettuato l'accesso al tuo account di AWS Organizations gestione, puoi scegliere Crea utente per creare un utente IAM Identity Center. IAM Identity Center invia al nuovo utente un'e-mail per consentirgli di impostare la password. È quindi possibile assegnare l'utente al portale come amministratore. Per ulteriori informazioni, consulta la sezione [Gestione delle identità in IAM Identity Center](#).

Sarà possibile modificare l'elenco degli amministratori del portale, in un secondo momento. Per ulteriori informazioni, consulta [Aggiungi o rimuovi gli amministratori del portale in AWS IoT SiteWise](#).

Note

Poiché solo un amministratore del portale può creare progetti e assegnare loro risorse, è necessario specificare almeno un amministratore del portale.

Come ultimo passaggio, è possibile aggiungere utenti in grado di accedere al nuovo portale.

Aggiungere utenti del portale in Monitor SiteWise

È possibile controllare quali utenti hanno accesso ai portali. In ogni portale, gli amministratori del portale creano uno o più progetti e assegnano gli utenti del portale come proprietari o visualizzatori per ciascun progetto. Ogni proprietario del progetto può invitare altri utenti del portale a diventare proprietari o visualizzatori del progetto.

In base al servizio di autenticazione utente, scegli una delle seguenti opzioni:

IAM Identity Center

Se desideri aggiungere un utente all'elenco Utenti, completa i seguenti passaggi.

Come aggiungere utenti del portale

1. Scegli gli utenti dall'elenco Utenti da aggiungere al portale. In questo modo gli utenti vengono aggiunti all'elenco degli utenti del portale. Se utilizzi SiteWise Monitor per la prima volta, non è necessario aggiungere l'amministratore del portale come utente del portale.

Note

Se utilizzi IAM Identity Center come archivio di identità e hai effettuato l'accesso al tuo account di AWS Organizations gestione, puoi scegliere Crea utente per creare un utente IAM Identity Center. IAM Identity Center invia al nuovo utente un'e-mail per consentirgli di impostare la password. È quindi possibile assegnare l'utente al portale come utente. Per ulteriori informazioni, consulta la sezione [Gestione delle identità in IAM Identity Center](#).

2. Se si aggiunge e concede l'accesso a un utente indesiderato, per risolvere basta deselectionarne la casella di controllo.
3. Quando hai finito di selezionare gli utenti, scegli Assegna utenti.

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

Find resources

	Display name	Email
☐	Jane Doe	janedoe@example.com
☒	John Doe	johndoe@example.com

Selected users (1)

Cancel Previous **Assign users**

IAM

Se vedi l'utente o il ruolo che desideri aggiungere nell'elenco degli utenti o dei ruoli IAM, completa i seguenti passaggi.

Come aggiungere utenti del portale

1. Esegui le seguenti opzioni:
 - Scegli utenti IAM per aggiungere un utente IAM come utente del portale.
 - Scegli i ruoli IAM per aggiungere un ruolo IAM come utente del portale.

Se utilizzi SiteWise Monitor per la prima volta, non è necessario aggiungere l'amministratore del portale come utente del portale.

2. Seleziona le caselle di controllo relative agli utenti o ai ruoli che desideri utilizzare come utenti del portale. In questo modo gli utenti o i ruoli vengono aggiunti all'elenco degli utenti del portale.
3. Se si aggiunge e concede l'accesso a un utente indesiderato, per risolvere basta deselezionare la casella di controllo.
4. Quando hai finito di selezionare gli utenti, scegli Assegna utenti.

⚠ Important

Gli utenti o i ruoli devono disporre dell'`iotsitewise:DescribePortal` autorizzazione per accedere al portale.

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

☒	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_GTnd0O4Wr	03-16-2021
☐	AWSIoTSiteWiseMonitorServiceRole_rHINLNCS-	03-11-2021
☐	AWSIoTSiteWiseMonitorServiceRole_XB330QUIO	03-10-2021
☐	[REDACTED]	
☐	[REDACTED]	
☐	[REDACTED]	

► Portal users (2) [Remove](#)

Cancel Previous **Assign users**

Complimenti! Hai creato con successo un portale, hai assegnato gli amministratori del portale e gli utenti assegnati che possono utilizzare quel portale quando invitati a farlo. Gli amministratori del tuo portale possono ora creare progetti e aggiungervi asset. A loro volta, i proprietari di progetto sono in condizione di creare i pannelli di controllo per la visualizzazione dei dati riferiti agli asset di ogni progetto.

È possibile modificare l'elenco degli utenti del portale in un secondo momento. Per ulteriori informazioni, consulta [Aggiungere o rimuovere utenti del portale in AWS IoT SiteWise](#).

Per, all'occorrenza, apportare modifiche al portale, consulta [Amministra i tuoi portali SiteWise Monitor](#).

Per iniziare a usare il portale, vedi Guida [introduttiva](#) nella SiteWise Monitor Application Guide.

Creare AWS IoT SiteWise dashboard ()AWS CLI

Quando si definiscono visualizzazioni (o widget) nei dashboard utilizzando il AWS CLI, è necessario specificare le seguenti informazioni nel documento JSON. `dashboardDefinition` Questa definizione è un parametro delle operazioni `and`. [CreateDashboardUpdateDashboard](#)

`widgets`

Un elenco delle strutture di definizione di widget ciascuna contenente le seguenti informazioni:

`type`

Il tipo di widget. AWS IoT SiteWise fornisce i seguenti tipi di widget:

- `sc-line-chart`— Un grafico a linee. Per ulteriori informazioni, vedere [Grafici a linee](#) nella Guida AWS IoT SiteWise Monitor all'applicazione.
- `sc-scatter-chart`— Un grafico a dispersione. Per ulteriori informazioni, vedere [Grafici a dispersione](#) nella Guida dell'AWS IoT SiteWise Monitor applicazione.
- `sc-bar-chart`— Un grafico a barre. Per ulteriori informazioni, vedere [Grafici a barre](#) nella Guida AWS IoT SiteWise Monitor all'applicazione.
- `sc-status-grid`— Un widget di stato che mostra il valore più recente delle proprietà degli asset sotto forma di griglia. Per ulteriori informazioni, consultate [Status widgets](#) nella Guida all'AWS IoT SiteWise Monitor applicazione.
- `sc-status-timeline`— Un widget di stato che mostra i valori storici delle proprietà degli asset sotto forma di sequenza temporale. Per ulteriori informazioni, consultate [Status widgets](#) nella Guida all'AWS IoT SiteWise Monitor applicazione.
- `sc-kpi`— Una visualizzazione degli indicatori chiave di prestazione (KPI). Per ulteriori informazioni, consulta [i widget KPI](#) nella Guida all'applicazione.AWS IoT SiteWise Monitor
- `sc-table`— Un widget per tabelle. Per ulteriori informazioni, consultate [Table widgets](#) nella Guida all'AWS IoT SiteWise Monitor applicazione.

`title`

Il titolo del widget.

x

La posizione orizzontale del widget, a partire dal lato sinistro della griglia. Questo valore si riferisce alla posizione del widget nella griglia del pannello di controllo.

y

La posizione verticale del widget, a partire dalla parte superiore della griglia. Questo valore si riferisce alla posizione del widget nella griglia del pannello di controllo.

width

La larghezza del widget, espressa in numero di spazi sulla griglia del pannello di controllo.

height

L'altezza del widget, espressa in numero di spazi sulla griglia del pannello di controllo.

metrics

Un elenco di strutture di parametri ciascuna delle quali definisce un flusso di dati per questo widget. Ogni struttura dell'elenco deve contenere le seguenti informazioni:

label

Un'etichetta da visualizzare per questo parametro.

type

Il tipo di origine dati per questa metrica. AWS IoT SiteWise fornisce i seguenti tipi di metriche:

- **iotsitewise**— La dashboard recupera i dati relativi alla proprietà di un asset in AWS IoT SiteWise. Se scegli questa opzione, devi definire `assetId` e `propertyId` per questo parametro.

assetId

(Facoltativo) L'ID di un asset in AWS IoT SiteWise.

Questo campo è obbligatorio se scegli **iotsitewise** per `type` in questo parametro.

propertyId

(Facoltativo) L'ID di una proprietà di asset in AWS IoT SiteWise.

Questo campo è obbligatorio se scegli **iotsitewise** per `type` in questo parametro.

analysis

(Facoltativo) Una struttura che definisce l'analisi, ad esempio le linee di tendenza, da visualizzare per il widget. Per ulteriori informazioni, consulta [Configurazione delle linee di tendenza](#) nella Guida all'AWS IoT SiteWise Monitor applicazione. È possibile aggiungere una linea di tendenza per ogni proprietà nel widget. La struttura di analisi contiene le seguenti informazioni:

trends

(Facoltativo) Un elenco di strutture di tendenza, ciascuna delle quali definisce un'analisi delle tendenze per questo widget. Ogni struttura dell'elenco contiene le seguenti informazioni:

type

Il tipo di linea di tendenza. Scegliete la seguente opzione:

- `linear-regression`— Visualizza una linea di regressione lineare. SiteWise Monitor utilizza il metodo dei [minimi quadrati](#) per calcolare la regressione lineare.

annotations

(Facoltativo) Una struttura di annotazioni che definisce le soglie per il widget. Per ulteriori informazioni, consulta [Configurazione delle soglie nella Guida](#) all'applicazione AWS IoT SiteWise Monitor. È possibile aggiungere fino a sei annotazioni per widget. La struttura delle annotazioni contiene le seguenti informazioni:

y

(Facoltativo) Un elenco di strutture di annotazioni, ciascuna delle quali definisce una soglia orizzontale per questo widget. Ogni struttura dell'elenco contiene le seguenti informazioni:

comparisonOperator

L'operatore di confronto per la soglia. Seleziona una delle seguenti opzioni:

- `LT`— Evidenzia le proprietà che hanno almeno un punto dati inferiore a `value`.
- `GT`— Evidenzia le proprietà che hanno almeno un punto dati maggiore di `value`.
- `LTE`— Evidenzia le proprietà che hanno almeno un punto dati inferiore o uguale a `value`.
- `GTE`— Evidenzia le proprietà che hanno almeno un punto dati maggiore o uguale a `value`.
- `EQ`— Evidenzia le proprietà che hanno almeno un punto dati uguale a `value`.

value

Il valore di soglia per confrontare i punti dati con `comparisonOperator`.

color

(Facoltativo) Il codice esadecimale a 6 cifre del colore di soglia. La visualizzazione mostra le legende delle proprietà in questo colore per le proprietà con almeno un punto dati che soddisfa la regola della soglia. Il valore predefinito è `black ()`. `#000000`

showValue

(Facoltativo) Se mostrare o meno il valore della soglia nei margini del widget. L'impostazione predefinita è `true`.

properties

(Facoltativo) Un dizionario semplice di proprietà per il widget. I membri di questa struttura dipendono dal contesto. AWS IoT SiteWise fornisce i seguenti widget che utilizzano:

properties

- [I grafici a linee](#), [i grafici a dispersione](#) e [i grafici a barre](#) hanno le seguenti proprietà:

colorDataAcrossThresholds

(Facoltativo) Se modificare o meno il colore dei dati che superano le soglie in questo widget. Quando abiliti questa opzione, i dati che superano una soglia vengono visualizzati nel colore che scegli. L'impostazione predefinita è `true`.

- [Le griglie di stato hanno le](#) seguenti proprietà:

labels

(Facoltativo) Una struttura che definisce le etichette da visualizzare nella griglia di stato. La struttura delle etichette contiene le seguenti informazioni:

showValue

(Facoltativo) Se visualizzare o meno l'unità e il valore per ogni proprietà dell'asset in questo widget. L'impostazione predefinita è `true`.

Example Esempio di definizione del pannello di controllo

L'esempio seguente definisce un pannello di controllo da un payload archiviato in un file JSON.

```
aws iotsitewise create-dashboard \
```

```
--project-id a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE \  
--dashboard-name "Wind Farm Dashboard" \  
--dashboard-definition file://dashboard-definition.json
```

L'esempio JSON seguente per `dashboard-definition.json` definisce il pannello di controllo con i seguenti widget di visualizzazione:

- Un grafico a linee che visualizza la potenza totale del parco eolico in alto a sinistra del pannello di controllo. Questo grafico a linee include una soglia che indica quando il parco eolico produce meno energia della potenza minima prevista. Questo grafico a linee include anche una linea di tendenza alla regressione lineare.
- Un grafico a barre che visualizza la velocità del vento per quattro turbine in alto a destra del pannello di controllo.

Note

Questo esempio rappresenta le visualizzazioni di grafici a linee e a barre su un pannello di controllo. Questo pannello di controllo è simile al [pannello di controllo del parco eolico di esempio](#).

```
{  
  "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"
  }
]
}
```

Attiva gli allarmi per i tuoi portali in AWS IoT SiteWise

È possibile abilitare la funzionalità di allarme supportata da AWS IoT Events for your portals in modo che gli amministratori del portale possano creare, modificare ed eliminare modelli di AWS IoT Events allarme nei portali Monitor. SiteWise I proprietari dei progetti possono configurare gli allarmi. I visualizzatori del progetto possono visualizzare i dettagli degli allarmi. Questa sezione spiega come utilizzare la AWS IoT SiteWise console per abilitare la funzionalità di allarme per i portali.

Important

- Non puoi creare allarmi esterni nei tuoi portali.
- Se desideri inviare notifiche di allarme, devi scegliere IAM Identity Center per il servizio di autenticazione degli utenti.
- La funzione di notifica degli allarmi non è disponibile in Cina (Pechino) Regione AWS.

Quando configuri e crei un portale, puoi abilitare gli allarmi e le notifiche di allarme nella Fase 2 Funzionalità aggiuntive. In base al servizio di autenticazione utente, scegli una delle seguenti opzioni:

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](#)

Per abilitare gli allarmi per un portale

1. (Facoltativo) Scegli Abilita allarmi.
 - Per il ruolo di AWS IoT SiteWise accesso, utilizza un ruolo esistente o crea un ruolo con le autorizzazioni richieste. Questo ruolo richiede `iotevents:BatchPutMessage` autorizzazione e una relazione di fiducia che `iotevents.amazonaws.com` consenta `iot.amazonaws.com` e assuma il ruolo.
2. (Facoltativo) Scegli Abilita notifiche di allarme.
 - a. Per Mittente, scegli il mittente.

⚠ Important

È necessario verificare l'indirizzo e-mail del mittente in Amazon SES. Per ulteriori informazioni, consulta la sezione [Verifica degli indirizzi e-mail in Amazon SES](#), nella Amazon Simple Email Service Developer Guide.

- b. Per AWS Lambda il ruolo, usa un ruolo esistente o crea un ruolo con le autorizzazioni richieste. Questo ruolo richiede le `sso-directory:DescribeUser` autorizzazioni `lambda:InvokeFunction` e una relazione di fiducia che `lambda.amazonaws.com` consenta `iotevents.amazonaws.com` e assuma il ruolo.
- c. Per quanto riguarda AWS Lambda le funzioni, scegli una funzione Lambda esistente o crea una funzione che gestisca le notifiche di allarme. Per ulteriori informazioni, consulta [Gestione delle notifiche di allarme](#) nella Guida per gli AWS IoT Events sviluppatori.

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**

Per abilitare gli allarmi per un portale

- (Facoltativo) Scegli Abilita allarmi.

- Per il ruolo di AWS IoT SiteWise accesso, utilizza un ruolo esistente o crea un ruolo con le autorizzazioni richieste. Questo ruolo richiede `iotevents:BatchPutMessage` autorizzazione e una relazione di fiducia che `iotevents.amazonaws.com` consenta `iot.amazonaws.com` e assuma il ruolo.

Per ulteriori informazioni sugli allarmi in SiteWise Monitor, consulta [Monitoraggio con allarmi](#) nella Guida all'AWS IoT SiteWise applicazione.

Attivazione del AWS IoT SiteWise portale all'edge

Dopo aver abilitato il portale sull'edge, questo portale è disponibile su tutti i gateway SiteWise Edge con il pacchetto di elaborazione dati abilitato nell'account.

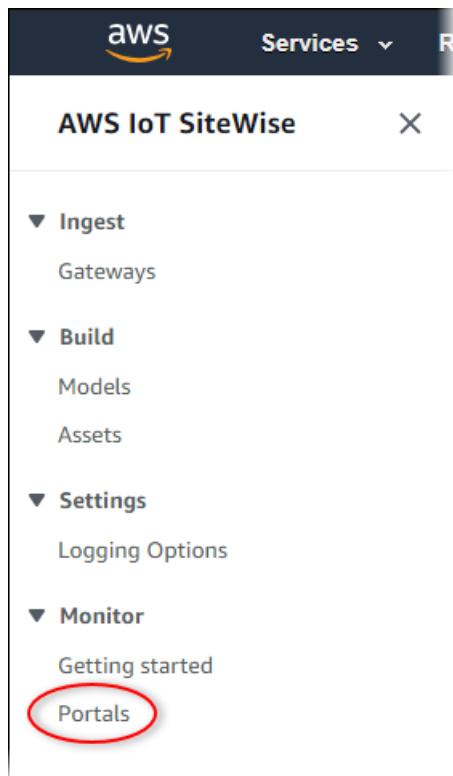
Per abilitare il portale nella periferia

1. Nella sezione Configurazione Edge, attiva Abilita questo portale all'edge.
2. Scegli Create (Crea) .

Amministra i tuoi portali SiteWise Monitor

Hai la possibilità di gestire e configurare vari aspetti del portale. Ciò include l'aggiunta e la rimozione di utenti o amministratori, l'impostazione delle autorizzazioni e dei ruoli degli utenti, la personalizzazione dell'URL e del nome del portale, l'impostazione delle informazioni di contatto dell'assistenza e l'invio di inviti via e-mail agli amministratori del portale.

1. Accedi alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, si deve selezionare Monitor (Monitoraggio), Portals (Portali).



3. Bisogna, infine, scegliere un portale e selezionare View details (Visualizza dettagli) (o scegliere Portal name (Nome portale)).
4. È possibile, a questo punto, svolgere le seguenti attività amministrative:
 - [Modifica i dettagli del portale in AWS IoT SiteWise](#)
 - [Aggiungi o rimuovi gli amministratori del portale in AWS IoT SiteWise](#)
 - [Inviare inviti via e-mail agli amministratori del portale](#)
 - [Aggiungere o rimuovere utenti del portale in AWS IoT SiteWise](#)
 - [Eliminare un portale in AWS IoT SiteWise](#)

Per informazioni su come creare un portale, consultare [Inizia con AWS IoT SiteWise Monitor \(Classic\)](#).

Argomenti

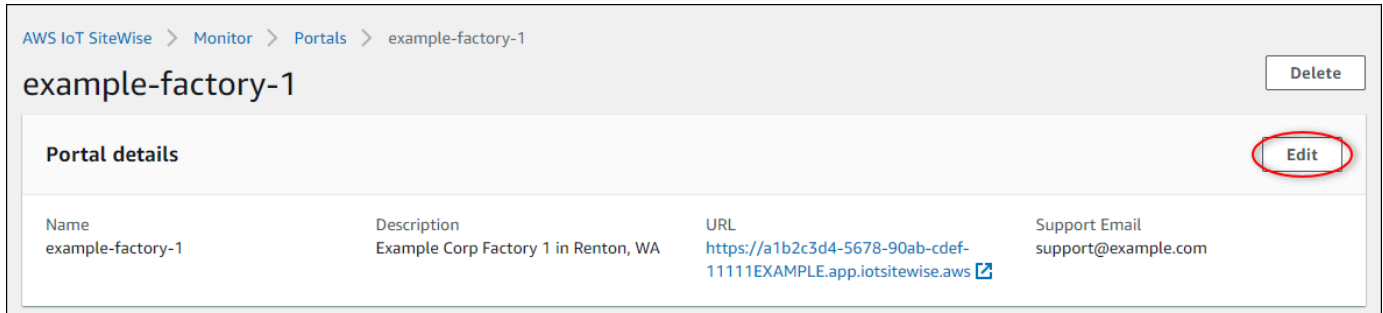
- [Modifica i dettagli del portale in AWS IoT SiteWise](#)
- [Aggiungi o rimuovi gli amministratori del portale in AWS IoT SiteWise](#)
- [Inviare inviti via e-mail agli amministratori del portale](#)
- [Aggiungere o rimuovere utenti del portale in AWS IoT SiteWise](#)

- [Eliminare un portale in AWS IoT SiteWise](#)

Modifica i dettagli del portale in AWS IoT SiteWise

Puoi modificare il nome, la descrizione, l'e-mail di supporto, il marchio e le autorizzazioni di un portale.

1. Alla pagina con i dettagli del portale, nella sezione Portal details (Dettagli portale), selezionare Edit (Modifica).

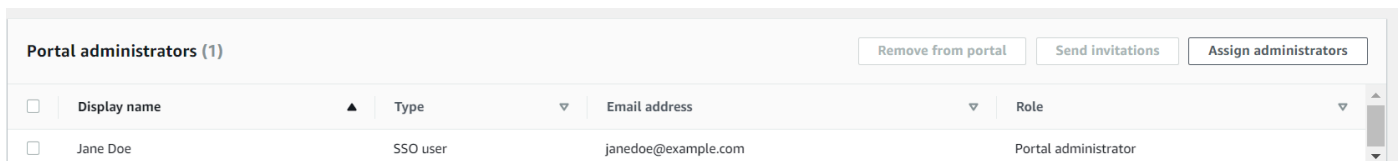


2. Aggiornare i campi Name (Nome), Description (Descrizione), Portal branding (Marchio del portale), Support contact email E-mail di contatto di supporto) o Permissions (Autorizzazioni).
3. Al termine, selezionare Save (Salva).

Aggiungi o rimuovi gli amministratori del portale in AWS IoT SiteWise

In pochi passaggi, è possibile aggiungere o rimuovere gli utenti nominati amministratori di un portale. In base al servizio di autenticazione utente, scegli una delle seguenti opzioni.

IAM Identity Center

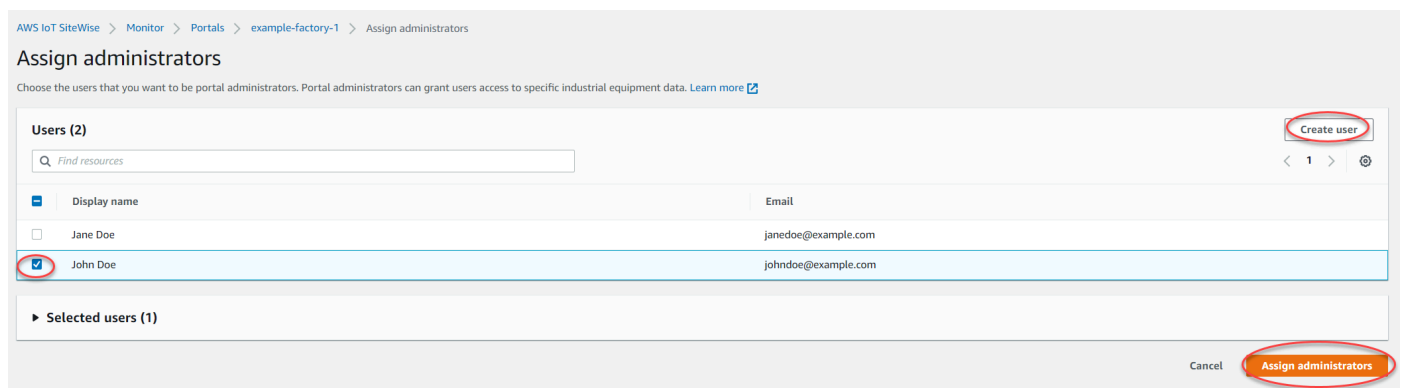


Come aggiungere amministratori del portale

1. Nella pagina dei dettagli del portale, nella sezione Amministratori del portale, scegli Assegna amministratori.
2. Nella pagina Assegna amministratori, seleziona le caselle di controllo relative agli utenti da aggiungere al portale come amministratori.

Note

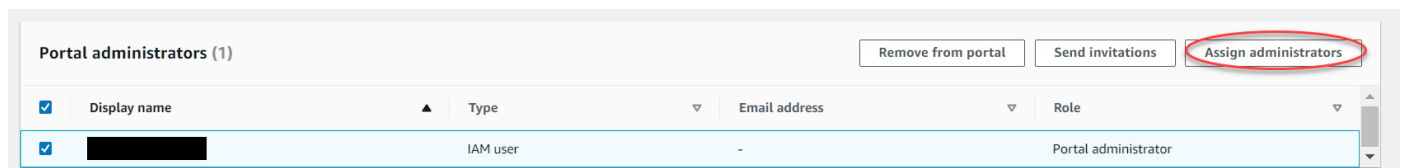
Se utilizzi IAM Identity Center come archivio di identità e hai effettuato l'accesso al tuo account di AWS Organizations gestione, puoi scegliere Crea utente per creare un utente IAM Identity Center. IAM Identity Center invia al nuovo utente un'e-mail per consentirgli di impostare la password. È quindi possibile assegnare l'utente al portale come amministratore. Per ulteriori informazioni, consulta la sezione [Gestione delle identità in IAM Identity Center](#).

3. Scegli Assegna amministratori.**Come rimuovere amministratori del portale**

- Alla pagina con i dettagli del portale, nella sezione Portal administrators (Amministratori portale), occorre spuntare la casella di controllo di ogni utente da rimuovere, per poi selezionare Remove from portal (Rimuovi dal portale).

Note

Ti consigliamo di selezionare almeno un amministratore del portale.

IAM

Come aggiungere amministratori del portale

1. Nella pagina dei dettagli del portale, nella sezione Amministratori del portale, scegli Assegna amministratori.
2. Nella pagina Assegna amministratori, procedi come segue:
 - Scegli utenti IAM, se desideri aggiungere un utente IAM come amministratore del portale.
 - Scegli i ruoli IAM, se desideri aggiungere un ruolo IAM come amministratore del portale.
3. Seleziona le caselle di controllo relative agli utenti o ai ruoli che desideri utilizzare come amministratori del portale. In questo modo gli utenti o i ruoli vengono aggiunti all'elenco degli amministratori del portale.
4. Scegli Assegna amministratori.

Important

Gli utenti o i ruoli devono disporre dell'`iotsitewise:DescribePortal` autorizzazione per accedere al portale.

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 `iotsitewise: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]	
☐	[REDACTED]	

► Portal administrators (2) [Remove](#)

[Cancel](#) [Assign administrators](#)

Come rimuovere amministratori del portale

- Alla pagina con i dettagli del portale, nella sezione Portal administrators (Amministratori portale), occorre spuntare la casella di controllo di ogni utente da rimuovere, per poi selezionare Remove from portal (Rimuovi dal portale).

Note

Non è consigliabile lasciare un portale senza amministratori.

Inviare inviti via e-mail agli amministratori del portale

È possibile inviare inviti via e-mail agli amministratori del portale.

- Nella pagina con i dettagli del portale, selezionare le caselle di controllo per gli amministratori del portale nella sezione Portal administrators (Amministratori portale).

Portal administrators (1) [Remove from portal](#) [Send invitations](#) [Assign users](#)

☒	Display name	Email address	Role
☒	John Doe	john.doe@example.com	Portal administrator

2. Bisogna poi selezionare **Send invitations (Manda inviti)**. Viene visualizzato il client di posta elettronica e nel corpo del messaggio viene popolato un invito.

È possibile personalizzare il messaggio e-mail prima di inviarlo agli amministratori del portale.

Aggiungere o rimuovere utenti del portale in AWS IoT SiteWise

Puoi scegliere quali utenti hanno accesso al portale. Gli utenti del portale vengono visualizzati nell'elenco degli utenti all'interno di un portale SiteWise Monitor. Da questo elenco, gli amministratori del portale possono aggiungere proprietari di progetti e i proprietari di progetti possono aggiungere visualizzatori di progetti.

Note

Gli amministratori del portale e gli utenti del portale possono contattarti tramite l'e-mail di supporto di un portale se hanno bisogno di aggiungere o rimuovere un utente.

In base al servizio di autenticazione utente, scegli una delle seguenti opzioni.

IAM Identity Center

Portal users (1)					Remove from portal	Assign users
☐	Display name	Type	Email address	Role		
☐	John Doe	SSO user	johndoe@example.com	Portal viewer		

Come aggiungere utenti del portale

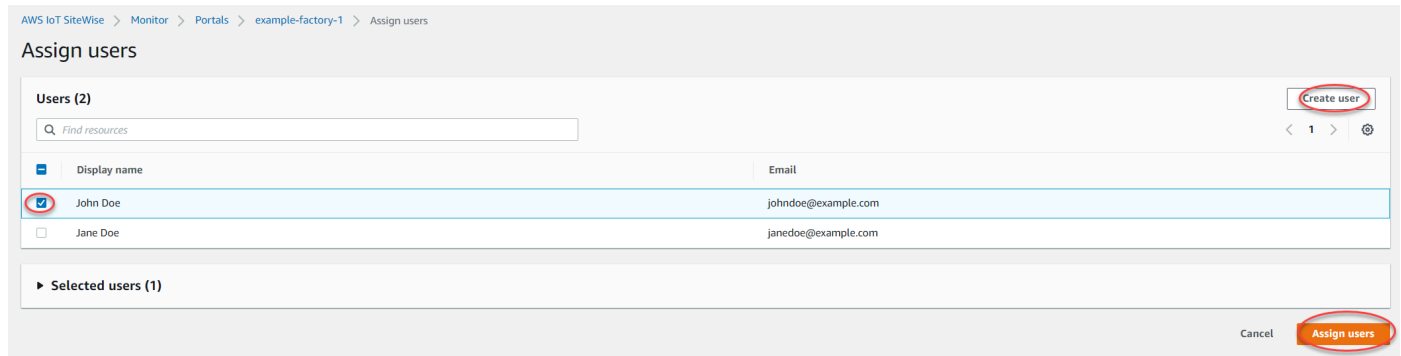
1. Nella pagina dei dettagli del portale, nella sezione **Utenti del portale**, scegli **Assegna utenti**.
2. Nella pagina **Assegna utenti**, seleziona la casella di controllo relativa agli utenti da aggiungere al portale.

Note

Se utilizzi IAM Identity Center come archivio di identità e hai effettuato l'accesso al tuo account di AWS Organizations gestione, puoi scegliere **Crea utente** per creare un utente IAM Identity Center. IAM Identity Center invia al nuovo utente un'e-mail per consentirgli di impostare la password. È quindi possibile assegnare l'utente al portale.

come utente. Per ulteriori informazioni, consulta la sezione [Gestione delle identità in IAM Identity Center](#).

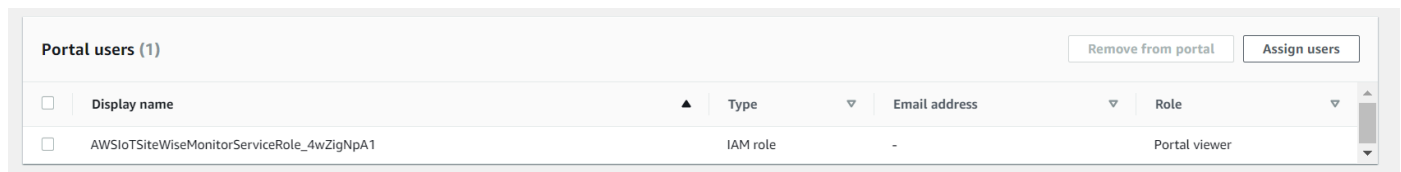
3. Scegliere Assign users (Assegna utenti).



Come rimuovere utenti del portale

- Nella pagina dei dettagli del portale, nella sezione Utenti del portale, seleziona la casella di controllo relativa agli utenti da rimuovere dal portale, quindi scegli Rimuovi dal portale.

IAM



Come aggiungere utenti del portale

- Nella pagina dei dettagli del portale, nella sezione Utenti del portale, scegli Assegna utenti.
- Nella pagina Assegna utenti, procedi come segue:
 - Scegli utenti IAM per aggiungere un utente IAM come utente del portale.
 - Scegli i ruoli IAM per aggiungere un ruolo IAM come utente del portale.
- Seleziona le caselle di controllo relative agli utenti o ai ruoli che desideri aggiungere come utenti del portale. In questo modo gli utenti o i ruoli vengono aggiunti all'elenco degli utenti del portale.
- Scegliere Assign users (Assegna utenti).

AWS IoT SiteWise > Monitor > Portals > example-factory-2 > Assign users

Assign users

Users Roles

IAM users (1) [Manage users in IAM console](#)

Find user name

☒	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) [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_RHNLNCS-	03-11-2021
☐	AWSIoTSiteWiseMonitorServiceRole_XB330QUIO	03-10-2021
☐	[REDACTED]	
☐	[REDACTED]	
☐	[REDACTED]	

► Portal users (2) [Remove](#)

Cancel [Assign users](#)

Come rimuovere utenti del portale

- Nella pagina dei dettagli del portale, nella sezione Utenti del portale, seleziona la casella di controllo relativa agli utenti da rimuovere dal portale, quindi scegli Rimuovi dal portale.

Important

Gli utenti o i ruoli devono disporre dell'`iotsitewise:DescribePortalautorizzazione` per accedere al portale.

Eliminare un portale in AWS IoT SiteWise

Puoi eliminare un portale se lo hai creato a scopo di test o se è un duplicato di un portale già esistente.

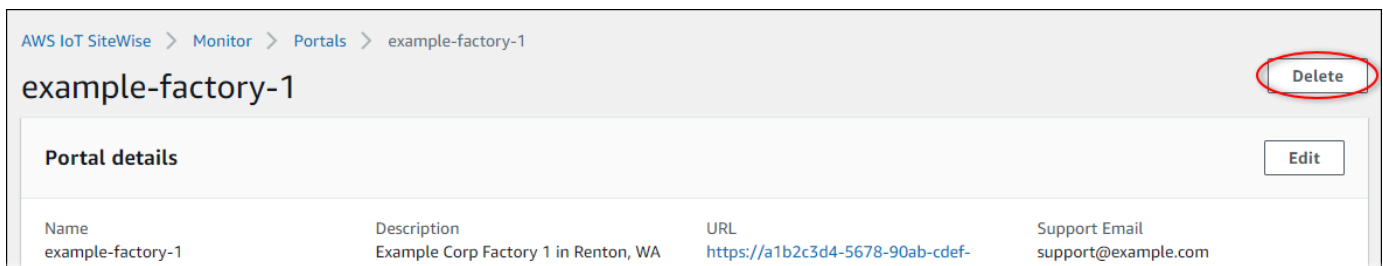
Note

Prima di poter eliminare un portale, è necessario eliminare manualmente tutti i pannelli di controllo e i progetti che contiene. Per ulteriori informazioni, consulta [Eliminazione di progetti ed Eliminazione di dashboard](#) nella SiteWise Monitor Application Guide.

1. Nella pagina con i dettagli del portale, selezionare Delete (Elimina).

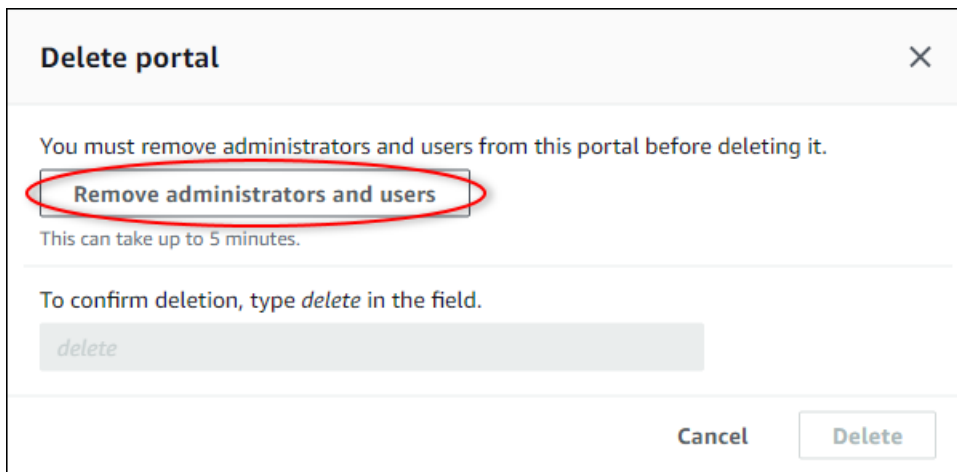
Important

Quando si elimina un portale, si perdono tutti i progetti in esso contenuti, con i relativi pannelli di controllo. Questa operazione non può essere annullata. I dati degli asset non subiscono variazioni.

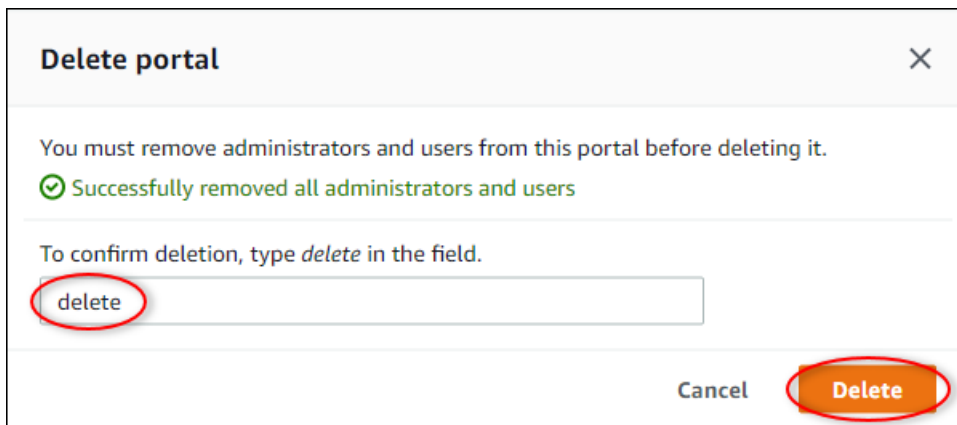


2. Nella finestra di dialogo Elimina portali, scegli Rimuovi amministratori e utenti.

Per poter eliminare definitivamente un portale, bisogna innanzitutto rimuoverne tutti gli amministratori e utenti. Se il portale non dispone di amministratori o utenti, il pulsante non viene visualizzato ed è possibile passare direttamente alla fase successiva.



3. Se si è certi di voler eliminare l'intero portale, immettere **delete** nel campo per confermare l'eliminazione.



4. Scegli Delete (Elimina).

Guida introduttiva a AWS IoT SiteWise Monitor (AI-Aware) - anteprima

In qualità di AWS amministratore dell'organizzazione, puoi creare portali dalla AWS IoT SiteWise console per consentire ai membri dell'organizzazione di visualizzare i tuoi dati. AWS IoT SiteWise Completa i seguenti passaggi per iniziare.

1. Configura e crea un portale.
2. Aggiungi amministratori del portale e invia e-mail di invito.
3. Aggiungi utenti del portale.

Dopo aver creato un portale, l'amministratore del portale può creare progetti e aggiungere utenti al progetto. I membri del progetto creano quindi dashboard su AWS IoT SiteWise cui visualizzare i dati connessi e consentire loro di monitorare le prestazioni dei dispositivi, dei processi e delle apparecchiature connessi.

Note

Quando aggiungi utenti o amministratori al portale, evita di creare policy AWS Identity and Access Management (IAM) che limitino le autorizzazioni degli utenti, come l'IP limitato. Qualsiasi policy allegata con autorizzazioni limitate non sarà in grado di connettersi al portale. AWS IoT SiteWise

Crea progetti da condividere con i tuoi team. I proprietari dei progetti possono quindi creare dashboard per visualizzare le proprietà delle risorse e aiutare gli utenti a comprendere le prestazioni di dispositivi, processi e apparecchiature. Fornisce inoltre una visione coerente delle operazioni ai team.

Le dashboard aiutano a visualizzare e comprendere i dati del progetto. Aiuta le aziende e gli utenti delle applicazioni a tenere traccia dei propri AWS IoT dispositivi e dati. Scegli il tipo di visualizzazione che meglio mostra i tuoi dati in base alle tue esigenze. Riorganizza e ridimensiona le visualizzazioni per creare un layout adatto al tuo team. Esplora le risorse e i dati dei dispositivi, dei processi e delle apparecchiature, identifica rapidamente i problemi e migliora l'efficienza operativa.

Argomenti

- [Crea un portale](#)
- [Configura il tuo portale](#)
- [Amministra i tuoi portali](#)
- [Eliminare un portale](#)
- [Crea dashboard con AWS CLI](#)
- [Accesso al portale](#)
- [Crea un progetto](#)
- [Aggiorna un progetto](#)
- [Elimina un progetto](#)
- [Creazione di un pannello di controllo](#)
- [Aggiornare una dashboard](#)

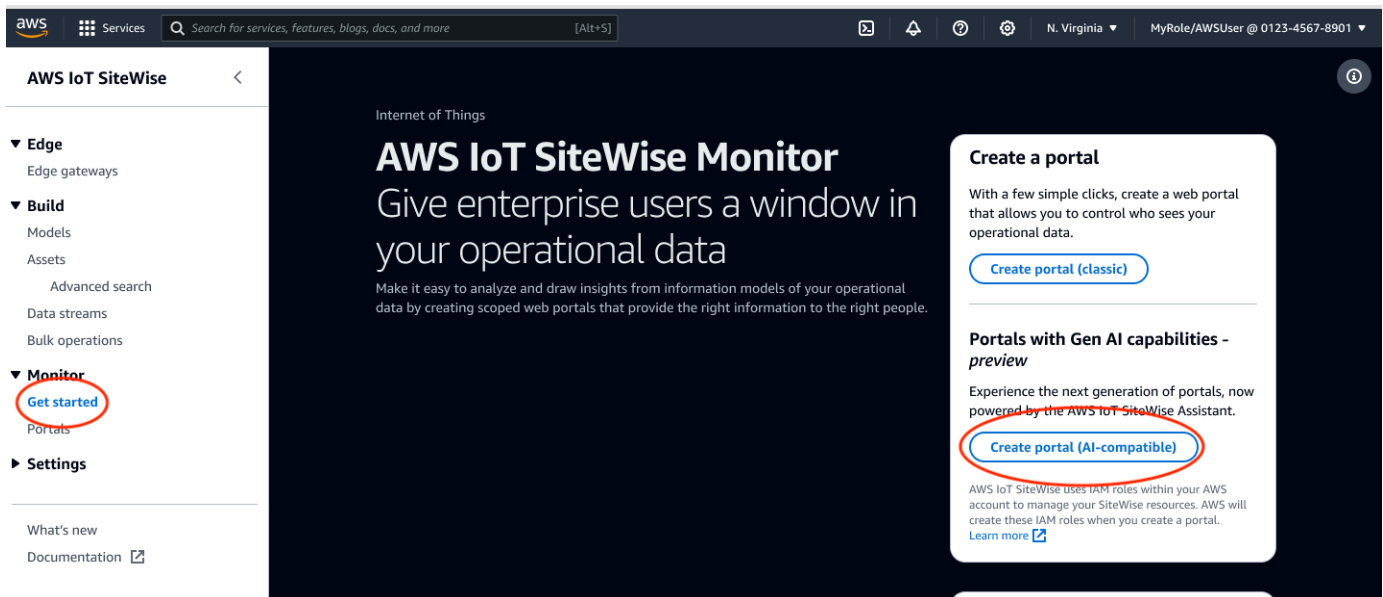
- [Eliminare una dashboard](#)
- [Configurare la dashboard](#)

Crea un portale

Crei un portale SiteWise Monitor nella AWS IoT SiteWise console.

Per creare un portale

1. Accedere alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Monitor, Inizia.
3. Scegli Crea portale (AI-Aware).



Successivamente, bisogna fornire alcune informazioni di base per la configurazione del portale.

Configura il tuo portale

Gli utenti utilizzano i portali per visualizzare i tuoi dati. Puoi personalizzare il nome, la descrizione, il marchio, l'autenticazione dell'utente, l'email di contatto dell'assistenza e le autorizzazioni di un portale.

Passaggi per configurare un portale:

1. Immettere un nome da assegnare al portale.

2. (Facoltativo) Inserire una descrizione del portale. Se si dispone di più portali, occorre utilizzare descrizioni significative per tenere traccia dei contenuti di ciascuno.
3. (Facoltativo) Caricare un'immagine per visualizzare il marchio nel portale. Scegliere un'immagine PNG quadrata. Se si carica un'immagine non quadrata, il portale ridimensiona l'immagine in un quadrato.
4. Inserisci un indirizzo e-mail nella casella e-mail di contatto del supporto per problemi di supporto.
5. Nella casella Autenticazione utente, scegli la seguente opzione:
 - Scegli IAM Identity Center se gli utenti del portale accedono a questo portale con i loro nomi utente e password aziendali.

Se non hai abilitato IAM Identity Center nel tuo account, procedi come segue:

- a. Selezionare Create user (Crea utente).
- b. Nella pagina Crea utente, per creare il primo portale, inserisci l'indirizzo e-mail, il nome e il cognome dell'utente, quindi scegli Crea utente.

 Note

Il supporto per le credenziali IAM sarà presto disponibile.

6. Scegli una delle seguenti opzioni nella sezione Accesso al servizio:
 - Scegli Crea e utilizza un nuovo ruolo di servizio. Per impostazione predefinita, SiteWise Monitor crea automaticamente un ruolo di servizio per ogni portale. Questo ruolo consente agli utenti del portale di accedere alle AWS IoT SiteWise risorse dell'utente. Per ulteriori informazioni, consulta [Utilizza i ruoli di servizio per AWS IoT SiteWise Monitor](#).
 - Scegli Usa un ruolo di servizio esistente, quindi scegli il ruolo di destinazione.
7. Scegli di abilitare l' AWS IoT SiteWise Assistente per questo portale. L' AWS IoT SiteWise Assistente fornisce analisi rapide dei dati, approfondimenti in tempo reale e consigli guidati.

 Note

L'attivazione dell' AWS IoT SiteWise Assistente comporterà dei costi. Per utilizzare soluzioni e linee guida di livello aziendale, devi disporre di un set di dati associato all'indice Amazon Kendra.

8. (Facoltativo) Aggiungere tag per il portale. Per ulteriori informazioni, consulta [Tagga le tue AWS IoT SiteWise risorse](#).
9. Scegli Crea portale. AWS IoT SiteWise creerà il tuo portale.

Note

Se si chiude la console, è possibile completare la procedura di configurazione aggiungendo amministratori e utenti. Per ulteriori informazioni, consulta [Aggiungere o rimuovere gli amministratori del portale](#). Se non desideri conservare questo portale, eliminalo in modo che non utilizzi risorse. Per ulteriori informazioni, consulta [Eliminare un portale](#).

Una volta creato il portale, viene visualizzato un messaggio.

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)	

Una volta creato, il portale viene elencato nella sezione Portali. La sezione Dettagli del portale elenca il nome, la descrizione, l'ID, l'URL, lo stato, le date dell'ultimo aggiornamento e creazione, il marchio del portale e l'e-mail di assistenza per ogni portale.

La colonna Status può avere uno dei seguenti valori.

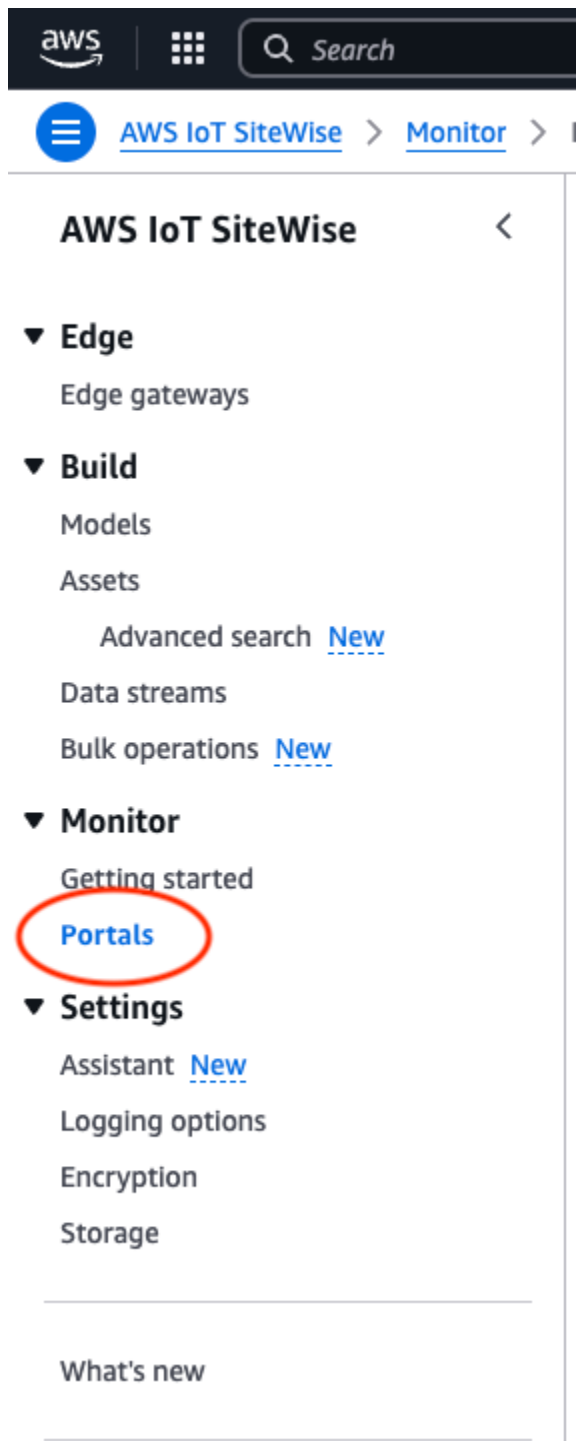
- **CREATING** - AWS IoT SiteWise sta elaborando la richiesta di creazione del portale. Il completamento di questo processo può richiedere diversi minuti.
- **AGGIORNAMENTO** - AWS IoT SiteWise sta elaborando la richiesta di aggiornamento del portale. Il completamento di questo processo può richiedere alcuni minuti.

- **PENDING** - AWS IoT SiteWise è in attesa del termine della propagazione dei record DNS. Il completamento di questo processo può richiedere diversi minuti. È possibile eliminare il portale mentre lo stato è IN SOSPEO.
- **ELIMINAZIONE** - AWS IoT SiteWise sta elaborando la richiesta di eliminazione del portale. Il completamento di questo processo può richiedere diversi minuti.
- **ATTIVO** - Quando il portale diventa attivo, gli utenti del portale possono accedervi.
- **FALLITO** - AWS IoT SiteWise impossibile elaborare la richiesta di creazione, aggiornamento o eliminazione del portale. Se hai abilitato AWS IoT SiteWise l'invio di log ad Amazon CloudWatch Logs, puoi utilizzare questi log per risolvere i problemi. [Per ulteriori informazioni, consulta Monitoring with Logs. AWS IoT SiteWise CloudWatch](#)

Amministra i tuoi portali

Hai la possibilità di gestire e configurare vari aspetti del portale. Ciò include l'aggiunta e la rimozione di amministratori, l'impostazione di autorizzazioni e ruoli, la personalizzazione del nome e della descrizione, la configurazione dell'e-mail di supporto e l'invio di inviti agli amministratori del portale.

1. Accedi alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, si deve selezionare Monitor (Monitoraggio), Portals (Portali).



3. Scegli un portale, quindi scegli Apri portale (o scegli il nome del portale).
4. È possibile, a questo punto, svolgere le seguenti attività amministrative:
 - [Modifica gli attributi del portale](#)
 - [Aggiungere o rimuovere gli amministratori del portale](#)

- [Inviare inviti via e-mail agli amministratori del portale](#)
- [Eliminare un portale in AWS IoT SiteWise](#)

Modifica gli attributi del portale

Puoi modificare il nome, la descrizione, il marchio, l'email di supporto e l'accesso al servizio di un portale.

1. Alla pagina con i dettagli del portale, nella sezione Portal details (Dettagli portale), selezionare Edit (Modifica).

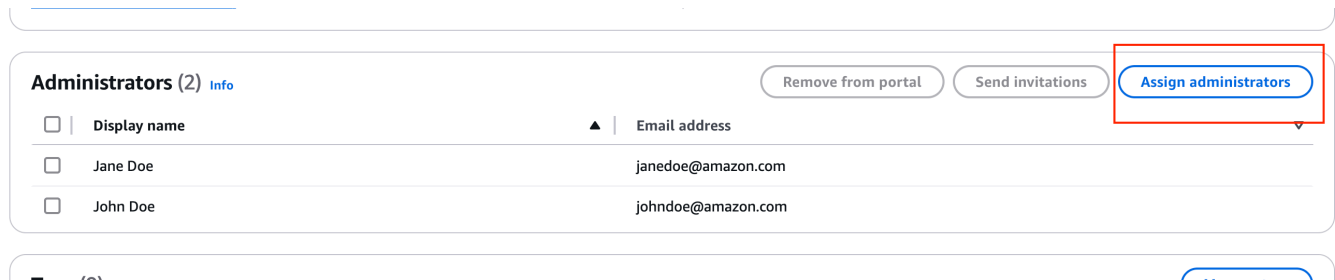
Portal details		
Name example portal	Status Active	URL https://p-jfnlf2d8.gamma.iotsitewise.aws
Type AI-compatible	AWS IoT SiteWise Assistant Disabled	Portal branding -
Description -	Last updated November 1, 2024, 15:37 (UTC-07:00)	Support contact email myemail@mycompany.com
ID f5fc93a1-011c-4c5a-81a6-e001b50d2547	Date created November 1, 2024, 15:37 (UTC-07:00)	

2. Aggiorna il nome, la descrizione, il marchio del portale, l'email di contatto del supporto, l'AWS IoT SiteWise assistente o l'accesso al servizio.
3. Al termine, scegli Salva modifiche.

Aggiungere o rimuovere gli amministratori del portale

In pochi passaggi, è possibile aggiungere o rimuovere gli utenti nominati amministratori di un portale. In base al servizio di autenticazione utente, scegli una delle seguenti opzioni.

IAM Identity Center



Come aggiungere amministratori del portale

1. Nella pagina dei dettagli del portale, nella sezione Amministratori, scegli Assegna amministratori.
2. Nella pagina Assegna amministratori, selezionare gli utenti da aggiungere al portale come amministratori.

Note

Se utilizzi IAM Identity Center come archivio di identità e hai effettuato l'accesso al tuo account di AWS Organizations gestione, puoi scegliere Crea utente per creare un utente IAM Identity Center. IAM Identity Center invia al nuovo utente un'e-mail per consentirgli di impostare la password. È quindi possibile assegnare l'utente al portale come amministratore. Per ulteriori informazioni, consulta la sezione [Gestione delle identità in IAM Identity Center](#).

3. Scegli Assegna amministratori.

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

2 matches

☐	Name	Email
☒	johndoe@amazon.com	johndoe@amazon.com
☐	janedoe@amazon.com	janedoe@amazon.com

[Cancel](#)[Assign administrators](#)

Come rimuovere amministratori del portale

- Alla pagina con i dettagli del portale, nella sezione Portal administrators (Amministratori portale), occorre spuntare la casella di controllo di ogni utente da rimuovere, per poi selezionare Remove from portal (Rimuovi dal portale).

Note

Il campo Amministratori (#) elenca il numero di amministratori del portale. È possibile aggiungere più amministratori del portale per gestire e lavorare sui progetti.

Inviare inviti via e-mail agli amministratori del portale

È possibile inviare inviti via e-mail agli amministratori del portale.

1. Nella pagina dei dettagli del portale, nella sezione Amministratori, seleziona le caselle di controllo relative agli amministratori del portale.
2. Bisogna poi selezionare Send invitations (Manda inviti). Viene visualizzato il client di posta elettronica e nel corpo del messaggio viene popolato un invito.

È possibile personalizzare il messaggio e-mail prima di inviarlo agli amministratori del portale.

Eliminare un portale

Puoi eliminare un portale se lo hai creato a scopo di test o se è un duplicato di un portale già esistente.

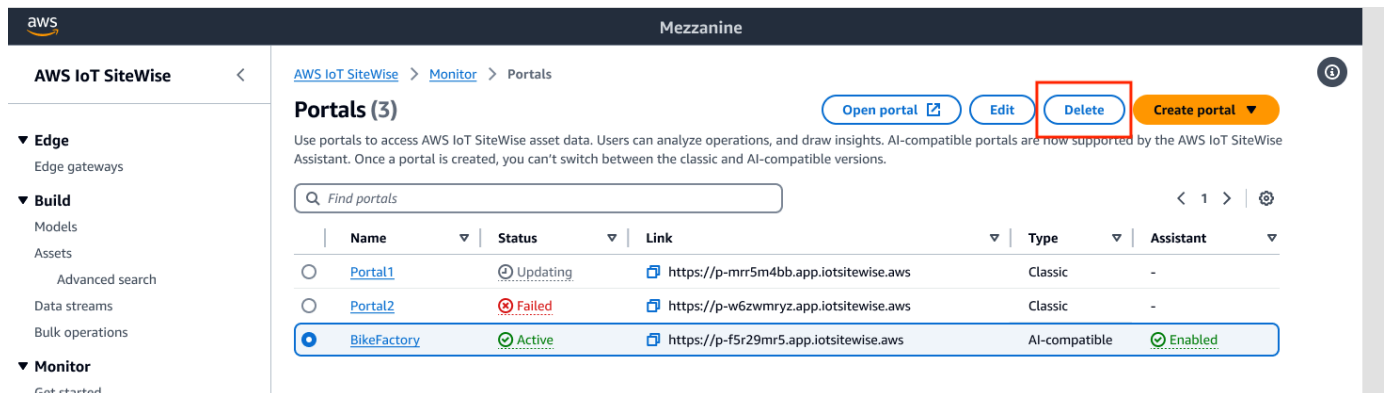
Note

Prima di poter eliminare un portale, è necessario eliminare manualmente tutti i pannelli di controllo e i progetti che contiene.

1. Nella pagina con i dettagli del portale, selezionare Delete (Elimina).

⚠ Important

Quando si elimina un portale, si perdono tutti i progetti in esso contenuti, con i relativi pannelli di controllo. Questa operazione non può essere annullata. I dati degli asset non subiscono variazioni.



2. Nella finestra di dialogo Elimina portale, scegli Rimuovi amministratori e utenti.

Per poter eliminare definitivamente un portale, bisogna innanzitutto rimuoverne tutti gli amministratori e utenti. Se il portale non dispone di amministratori o utenti, il pulsante non viene visualizzato ed è possibile passare direttamente alla fase successiva.

3. Se si è certi di voler eliminare l'intero portale, immettere **confirm** nel campo per confermare l'eliminazione.
4. Scegli Elimina.

Crea dashboard con AWS CLI

Quando si definiscono visualizzazioni (o widget) nei dashboard utilizzando il AWS CLI, è necessario specificare le seguenti informazioni nel documento JSON. `dashboardDefinition` Questa definizione è un parametro delle operazioni `and`. [CreateDashboardUpdateDashboard](#)

`displaySettings`

Le impostazioni dello schermo con i seguenti parametri:

- `numRows`— Numero di righe nel layout della dashboard. Ogni riga ha la larghezza di `CellSize`.

- `numColumns`— Numero di colonne nel layout del pannello di controllo. Ogni colonna ha una larghezza di `CellSize`.
- `cellSize`— (Facoltativo) La dimensione di una cella nel layout in pixel. Deve essere un numero positivo. L'impostazione predefinita è 10.
- `significantDigits`— (Facoltativo) Numero di cifre significative da visualizzare nella dashboard. L'impostazione predefinita è 4.

`querySettings`

Le informazioni sulla query con il seguente parametro:

- `refreshRate`— (Facoltativo) La velocità con cui i dati si aggiornano, in millisecondi. Accetta i seguenti valori: 1000, 5000, 10000, 60000, 300000.

`defaultViewport`

Se non viene fornito, il valore predefinito è quello degli ultimi cinque minuti. Contiene i seguenti parametri:

- `duration`— (Facoltativo) Determina a che punto del passato interrogare i dati a partire dal tempo presente.
- `start`— (Facoltativo) È di tipo `Date`. L'intervallo di tempo di inizio dell'interrogazione dei dati. Richiede una `end data` specificata.
- `end`— (Facoltativo) È di tipo `Data`. L'intervallo di tempo di fine dell'interrogazione dei dati. Richiede una `start data` specificata.

`widgets`

Un elenco di strutture di definizione dei widget che contengono le seguenti informazioni:

`type`

Il tipo di widget. AWS IoT SiteWise fornisce i seguenti tipi di widget:

- `xy-plot`— Un grafico a linee o un grafico a dispersione a seconda della configurazione.
- `bar-chart`— Un grafico a barre.
- `kpi-chart`— Un grafico degli indicatori chiave delle prestazioni.
-

`status-timeline`— Un widget di stato che visualizza e naviga tra i dati delle serie temporali provenienti da una o più fonti di dati.

- `text`— Un widget di testo.
- `table`— Un widget da tabella.

`id`

Un identificatore univoco per il widget.

`x`

La posizione orizzontale del widget, a partire dalla sinistra della dashboard. Questo valore si riferisce alla posizione del widget nella griglia del pannello di controllo.

`y`

La posizione verticale del widget, a partire dalla parte superiore della dashboard. Questo valore si riferisce alla posizione del widget nella griglia del pannello di controllo.

`z`

L'ordine relativo dei widget. Un widget con valore Z più grande viene visualizzato davanti al widget con valore Z inferiore, se si sovrappongono.

`width`

La larghezza del widget, espressa in numero di celle sulla dashboard.

`height`

L'altezza del widget, espressa in numero di celle sulla dashboard.

`properties`

Un elenco di proprietà del widget. Varia in base al tipo di widget. Vedi [IoT App Kit](#) per i dettagli.

Example Esempio di definizione del pannello di controllo

L'esempio seguente definisce un pannello di controllo da un payload archiviato in un file JSON.

```
aws iotsitewise create-dashboard \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE \  
  --dashboard-name "Example Dashboard" \  
  --properties '[{"name": "Example Property"}']
```

```
--dashboard-definition file://dashboard-definition.json
```

L'esempio JSON seguente per `dashboard-definition.json` definisce il pannello di controllo con i seguenti widget di visualizzazione:

```
{
  "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"
                    }
                ]
            }
        }
    }
}

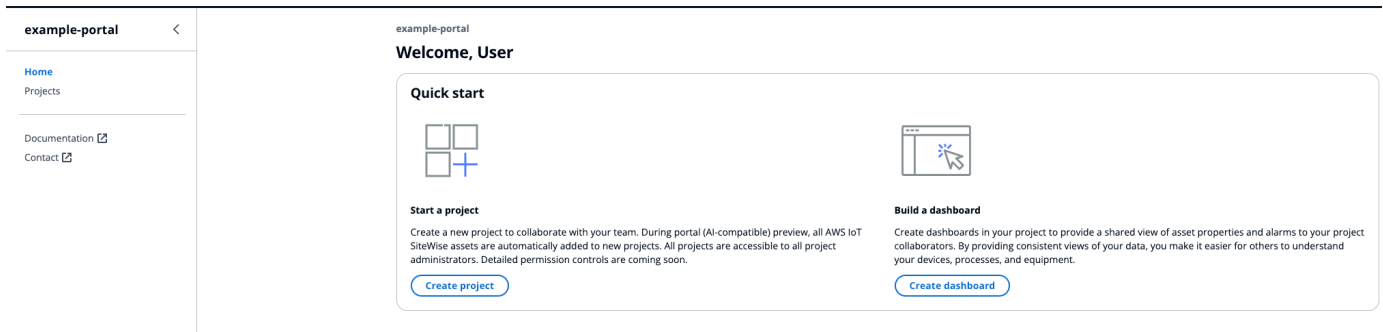
```

```
        }
      ],
      "properties": [],
      "assetModels": [],
      "alarms": [],
      "alarmModels": [],
      "requestSettings": {
        "aggregation": "AVERAGE"
      }
    },
    "axis": {
      "showX": true,
      "showY": true
    },
    "styleSettings": {
      "2960b958-2034-4d6e-bcc2-360f1f02e505": {
        "color": "#7d2105"
      }
    }
  },
  "querySettings": {
    "refreshRate": 5000
  }
}
```

Accesso al portale

Login utente

1. Nel browser, inserisci l'URL dell'applicazione.
2. Inserisci il nome utente e la password e fai clic sul pulsante Accedi.
3. Ora hai effettuato l'accesso all'applicazione.



Crea un progetto

Crea progetto

1. Un progetto viene creato in due modi:
 - a. Nella home page, nella sezione Benvenuto sotto Avvio rapido, scegli Crea progetto.
 - b. Dal riquadro di navigazione a sinistra, scegli Progetti. Scegli Crea nell'angolo in alto a destra per creare un progetto.
2. Nella sezione Crea progetti, inserisci un nome per il progetto e fornisci una descrizione opzionale.
3. Scegli Create (Crea) .

Aggiorna un progetto

Modifica progetto

1. Scegli il pulsante Aggiorna nell'angolo in alto a destra della pagina del progetto per modificare i dettagli del progetto.
2. Cambia il nome del progetto modificando il nome del progetto.
3. Modifica la descrizione del progetto modificando i dettagli della descrizione.
4. Seleziona Aggiorna per salvare le modifiche.

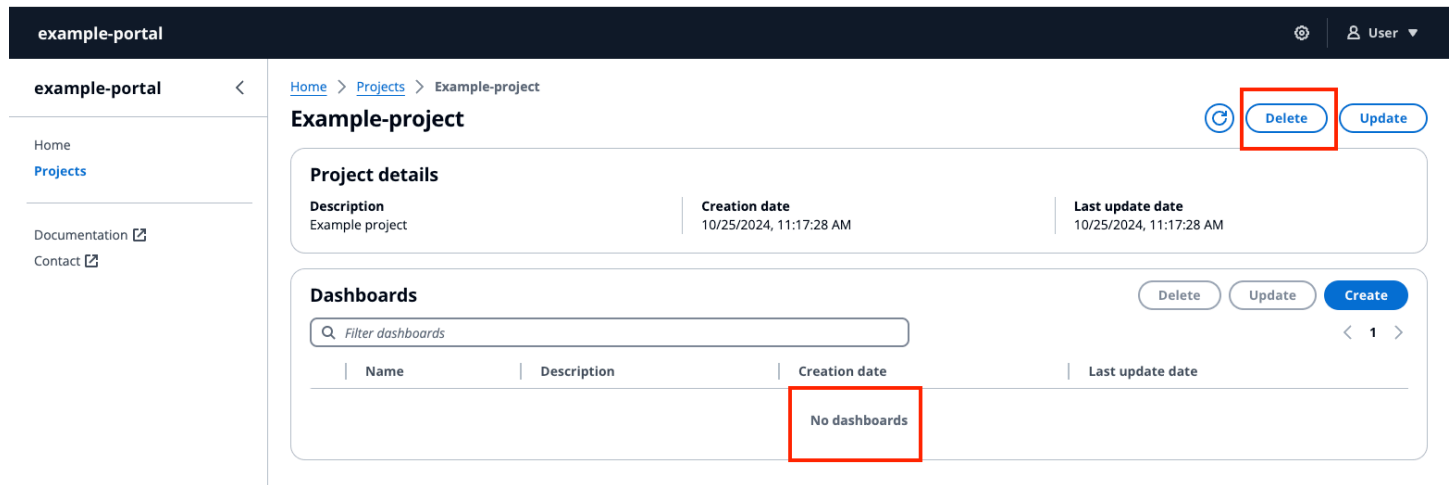
The screenshot shows the 'Example-project' page in the AWS IoT SiteWise portal. The page has a dark header with 'example-portal' and a user profile. A sidebar on the left contains links for 'Home', 'Projects', 'Documentation', and 'Contact'. The main content area is titled 'Example-project' and includes a breadcrumb trail 'Home > Projects > Example-project'. In the top right corner of the main area, there are three buttons: a refresh icon, 'Delete', and 'Update'. The 'Update' button is highlighted with a red rectangular box. Below the buttons, the 'Project details' section shows the project name 'Example project', its creation date '10/25/2024, 11:17:28 AM', and its last update date '10/25/2024, 11:17:28 AM'. The 'Dashboards' section features a search bar, a table with one dashboard entry, and buttons for 'Delete', 'Update', and 'Create'. The table has columns for Name, Description, Creation date, and Last update date.

Name	Description	Creation date	Last update date
example dashboard	dashboard	10/25/2024, 11:17:52 AM	10/25/2024, 11:17:52 AM

Elimina un progetto

Eliminare il progetto

1. È possibile eliminare il progetto solo dopo aver eliminato tutte le dashboard del progetto.
2. Seleziona il pulsante Elimina in alto a destra nella pagina del progetto.
3. Conferma nuovamente che desideri eliminare il progetto.
4. Seleziona Elimina per eliminare il progetto.

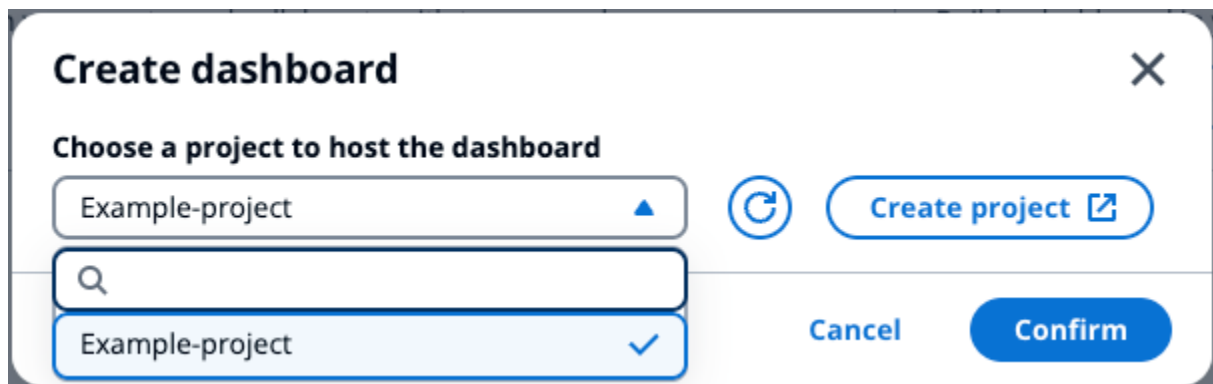


Creazione di un pannello di controllo

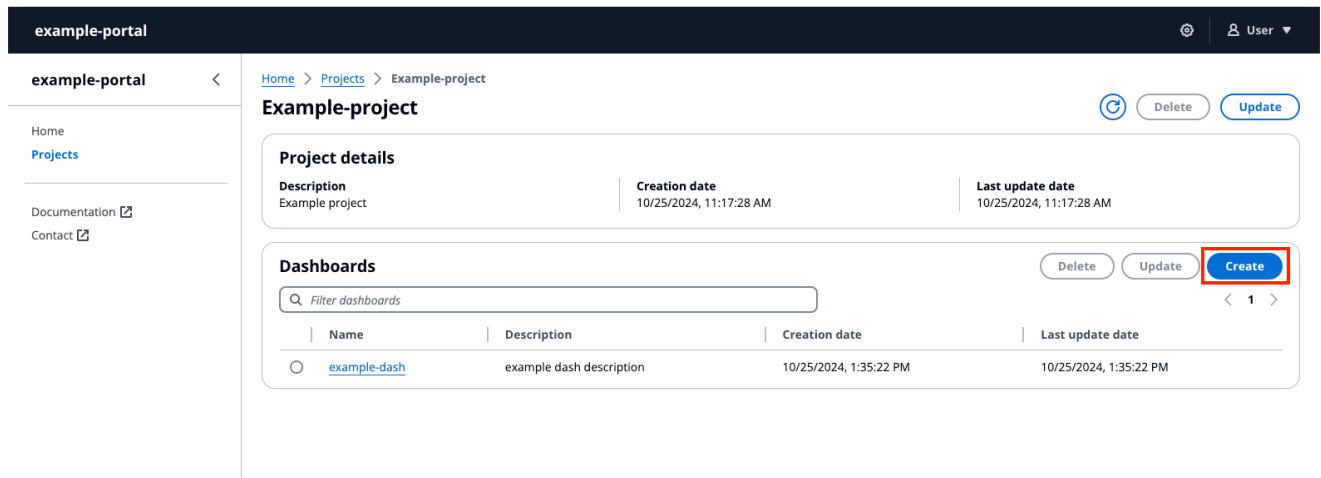
Creazione di un pannello di controllo

1. Crea una dashboard in due modi:

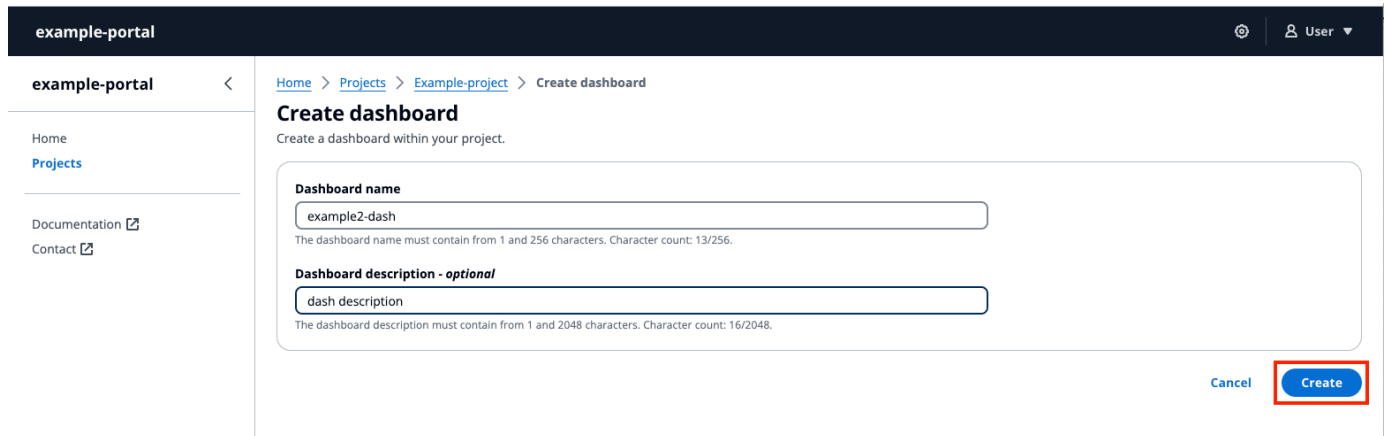
- a. Crea una dashboard da Crea una dashboard nella home page.
 - i. Per creare la dashboard in un progetto esistente, scegli il nome del progetto dal menu a discesa in Scegli un progetto per ospitare la dashboard.
 - ii. Se non hai un progetto, scegli Crea progetto e seleziona Conferma.



- b. Crea una dashboard da un progetto nella sezione Progetti, in Dashboard.



2. Seleziona Crea nell'angolo in alto a destra.
3. Inserisci un nome per la dashboard e fornisci una descrizione facoltativa della dashboard.
4. Seleziona Crea.



5. Configura la dashboard appena creata.

Aggiornare una dashboard

La sezione Dashboard elenca le dashboard del progetto. Seleziona una dashboard dall'elenco.

Aggiornare una dashboard

1. Seleziona una dashboard da aggiornare.

example-portal

Home > Projects > Example-project

Example-project

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

2. Aggiorna il nome della dashboard e, facoltativamente, la descrizione della dashboard. Seleziona **Aggiorna** per salvare le modifiche.

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**

Eliminare una dashboard

La sezione Dashboards elenca i dashboard del progetto. Seleziona una dashboard dall'elenco.

Eliminare una dashboard

1. Seleziona una dashboard da eliminare.

The screenshot shows the 'Example-project' page in the AWS IoT SiteWise portal. The page has a dark header with 'example-portal' and a user profile. Below the header, there's a breadcrumb trail: Home > Projects > Example-project. The main content area is titled 'Example-project' and contains two sections: 'Project details' and 'Dashboards'. The 'Project details' section shows the project description 'Example project', creation date '10/25/2024, 11:17:28 AM', and last update date '10/25/2024, 11:17:28 AM'. The 'Dashboards' section has a search bar 'Filter dashboards' and a table of dashboards. The table has columns: Name, Description, Creation date, and Last update date. The first row is 'example-dash' with description 'example dash description', creation date '10/25/2024, 1:35:22 PM', and last update date '10/25/2024, 1:35:22 PM'. There are 'Delete', 'Update', and 'Create' buttons for the dashboards.

2. Seleziona Elimina per eliminare la dashboard. Questa operazione non può essere annullata.

Configurare la dashboard

La sezione Dashboards elenca le dashboard del progetto. Seleziona una dashboard dall'elenco. La modalità Modifica consente di configurare la dashboard aggiungendo widget e configurandoli. Il pulsante Anteprima consente di visualizzare le modifiche.

The screenshot shows the 'example-dash' dashboard configuration page. The page has a dark header with 'example-portal' and a user profile. Below the header, there's a breadcrumb trail: Home > Projects > Example-project > example-dash. The main content area is titled 'example-dash' and contains a 'Time range' dropdown set to 'Last 5 minutes', a 'Refresh rate' dropdown set to '5s', and buttons for 'Save' and 'Edit'. There is also an 'AI Assistant' button.

Passaggi per configurare la dashboard:

- Trascina e rilascia diversi tipi di widget di dati nell'area di disegno della dashboard per la visualizzazione dei dati.
- Aggiungi dati ai widget desiderati, dal Resource Explorer a sinistra. Il Resource explorer è composto dalle sezioni Modeled, Unmodeled e Dynamic Assets. Ricerca per nome della risorsa o nome della proprietà. Seleziona la proprietà da aggiungere e scegli Aggiungi.
- Ottimizza il layout e lo stile modificando le configurazioni sui widget. Configura i componenti tra cui titolo, soglie e altre specifiche di configurazione.
- Configura l'intervallo di tempo in cui vengono visualizzati i dati.

- Scegli l'intervallo di tempo in cui vengono visualizzati i dati. Scegli un intervallo di tempo e una frequenza di aggiornamento dall'angolo in alto a destra e personalizza l'intervallo. Scegli la frequenza con cui i dati devono essere aggiornati dal menu.
- Seleziona l'intervallo di tempo su un widget utilizzando la rotellina di scorrimento del mouse trackball o facendo clic con il pulsante destro del mouse. Questo sposta l'intervallo di tempo di visualizzazione.
- Seleziona Salva.

Argomenti

- [Esploratore di risorse](#)
- [Widget](#)
- [Configura i widget](#)
- [Usa i widget](#)
- [Allarmi nei widget](#)
- [AWS IoT SiteWise Utilizzo dell'assistente nei widget](#)
- [Esempi di domande da porre all'Assistente AWS IoT SiteWise](#)

Esploratore di risorse

Questa sezione descrive gli asset modellati, non modellati e dinamici. Scegli le risorse tra le tre, aggiungile ai tuoi widget e visualizzale.

Argomenti

- [Modellato](#)
- [Non modellato](#)
- [Risorse dinamiche](#)

Modellato

Questa sezione descrive il processo di selezione e visualizzazione degli asset modellati.

Selezione degli asset

Le risorse possono essere interrogate come segue:

- Cerca il nome di una risorsa. Usa un jolly. * Ad esempio, Wind* restituisce i nomi delle risorse che iniziano con il testoWind. È necessario effettuare l'[integrazione con AWS IoT TwinMaker](#) per usufruire di questa funzionalità.
- Tutte le risorse sono elencate per impostazione predefinita.

Dagli asset elencati, filtra per nome, descrizione, ID o ID del modello di asset. Seleziona una risorsa per elencarne le proprietà (flussi di dati) e gli allarmi.

Selezione del flusso di dati

I flussi di dati sono elencati sotto il menu Data Streams. Filtra i flussi di dati elencati dai metadati delle [proprietà](#) nella versione più recente//. <https://docs.aws.amazon.com/iot-sitewise/APIReference>
Seleziona uno o più flussi di dati a seconda del widget selezionato.

- KPI e Gauge supportano solo un singolo flusso di dati.
- I widget rimanenti supportano più flussi di dati con selezione multipla.

Selezione degli allarmi

AWS IoT SiteWise gli allarmi sono elencati sotto il menu Alarm Data Streams. Filtra i flussi di dati degli allarmi elencati in base ai metadati degli allarmi. nome, proprietà di input e ID del modello composito sono alcuni dei metadati utilizzati per il filtraggio. Seleziona uno o più flussi di dati a seconda del widget selezionato.

- KPI e Gauge supportano solo un singolo allarme.
- I widget rimanenti supportano più allarmi con selezione multipla.

Visualizzazione delle risorse modellate

1. Trascina il widget sull'area di disegno. Seleziona le proprietà di ogni pannello di widget per creare una dashboard.
2. L'opzione Filtro filtra le risorse per scegliere la risorsa da visualizzare. Il filtraggio viene eseguito per testo, proprietà o valore. Il filtraggio riguarda le risorse caricate nel browser e non il filtro di backend.
3. Cerca per elencare una risorsa da aggiungere al widget.
4. Aggiungi la risorsa al widget nell'area di disegno.

5. Scegliete Reimposta per selezionare un'altra risorsa o apportare modifiche alla risorsa scelta.
6. Salva la dashboard. In modalità Anteprima, scegliete risorse diverse dal menu a discesa per monitorare le proprietà di ciascuna risorsa senza ricostruire i pannelli di dati.

 Note

La rotellina delle impostazioni di configurazione sul lato destro mostra le preferenze che l'utente può scegliere, come le preferenze relative alle dimensioni della pagina, alle prime colonne permanenti, alle ultime colonne permanenti e alle preferenze delle colonne. Personalizza le tue preferenze e scegli Conferma per applicare le modifiche.

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.

Non modellato

Questa sezione descrive la ricerca di flussi di dati non modellati e l'aggiunta ai widget da visualizzare.

Configura la dashboard

683

Visualizzazione di flussi di dati non modellati

1. Trascina il widget sull'area di disegno. Seleziona le proprietà di ogni pannello di widget per creare una dashboard.
2. I flussi di dati non modellati sono elencati nella sezione Serie temporali. Hanno proprietà personalizzabili.
3. L'opzione Filtro filtra i flussi di dati da visualizzare. Il filtraggio riguarda i flussi di dati caricati nel browser e non il filtraggio back-end.
4. Aggiungi il flusso di dati al widget nell'area di disegno.
5. Scegli Ripristina per deselezionare il flusso di dati.
6. Salva la dashboard. In modalità Anteprima, scegliete risorse diverse dal menu a discesa per monitorare le proprietà di ciascuna risorsa senza ricostruire i pannelli di dati.

Note

La rotellina delle impostazioni di configurazione sul lato destro mostra le preferenze che l'utente può scegliere, come le preferenze relative alle dimensioni della pagina, alle prime colonne permanenti, alle ultime colonne permanenti e alle preferenze delle colonne. Personalizza le tue preferenze e scegli Conferma per applicare le modifiche.

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...

Risorse dinamiche

Il nuovo SiteWise Monitor consente ai clienti di cambiare dinamicamente gli asset per un modello di asset selezionato. È possibile visualizzare le proprietà di diverse risorse selezionandole da un menu a discesa.

Visualizzazione dinamica delle risorse

1. Scegli la scheda Risorse dinamiche nell'esploratore delle risorse.
2. Seleziona un modello di asset per cui elencare gli asset dal menu a discesa.
3. Seleziona la risorsa predefinita dal menu a discesa.
4. Scegliete Imposta modello di asset per selezionare il modello di asset.
5. Salva la dashboard. In modalità Anteprima, scegliete risorse diverse dal menu a discesa per monitorare le proprietà di ciascuna risorsa, senza ricostruire i pannelli di dati.

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. ✓

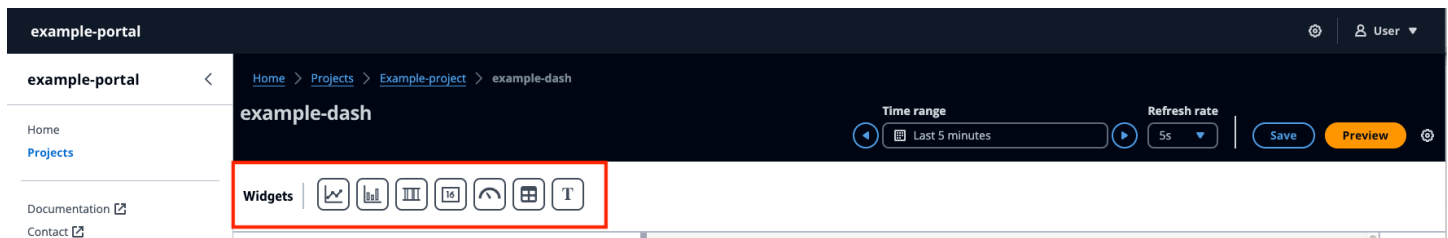
Finished loading asset models.

Set asset model

Widget

Widgets supporta un'ampia gamma di funzionalità, tra cui allarmi, live streaming ad alte prestazioni e sincronizzazione fluida con altri componenti dell'App Kit IoT. La dashboard supporta i seguenti widget:

- **Linea:** il widget Linea è un widget di visualizzazione che mostra tendenze e cambiamenti nel tempo. È costituito da una serie di punti dati, ciascuno rappresentato da un punto o un indicatore, collegati da segmenti di linea retta per creare un grafico a linee. Supporta un'ampia gamma di funzionalità, tra cui allarmi, soglie, live streaming ad alte prestazioni e sincronizzazione fluida con altri componenti dell'App Kit IoT. Questo widget è personalizzabile per comunicare dati complessi in modo chiaro e conciso.
- **Grafico a barre:** il grafico a barre è un potente strumento di visualizzazione che mostra dati di serie temporali. Supporta un'ampia gamma di funzionalità, tra cui allarmi, live streaming ad alte prestazioni e sincronizzazione fluida con altri componenti dell'App Kit IoT.
- **Cronologia:** il widget Timeline offre un modo per visualizzare e navigare tra i dati delle serie temporali provenienti da fonti di dati. È unico per la visualizzazione dei valori del flusso di dati con colori distinti sulla timeline. Supporta un ricco set di funzionalità tra cui allarmi, live streaming ad alte prestazioni e sincronizzazione fluida tra altri componenti dell'App Kit IoT. È utilizzato al meglio per visualizzare tipi di dati non numerici/
- **KPI:** il componente Key Performance Indicator (KPI) fornisce una rappresentazione compatta di una panoramica delle proprietà degli asset. Supporta allarmi e soglie. Questa panoramica fornisce informazioni fondamentali sulle prestazioni complessive di dispositivi, apparecchiature e processi. KPI supporta solo un singolo flusso di dati o allarme e non più flussi di dati.
- **Gauge:** il componente Gauge fornisce una rappresentazione compatta di una panoramica delle proprietà degli asset. Viene utilizzato per visualizzare informazioni fondamentali sulle prestazioni complessive dei dispositivi, delle apparecchiature o dei processi. È funzionalmente uguale ai KPI, ma visivamente diverso. Gauge visualizza il valore del flusso di dati, la soglia e l'intervallo di valori. Puoi interagire con AWS IoT i dati provenienti da una o più fonti di dati con Gauge.
- **Tabella:** il componente Table fornisce una forma compatta per la visualizzazione di uno o più flussi di dati da una o più fonti di dati di serie temporali. Visualizza le risorse con Property, Latest value e Unit in forma tabellare. Supporta gli AWS IoT SiteWise allarmi.
- **Testo:** il widget Testo aiuta a scrivere testo con vari colori e caratteri. Puoi creare un link associando un testo a un URL. I campi Proprietà e Soglie non sono abilitati per questo widget.

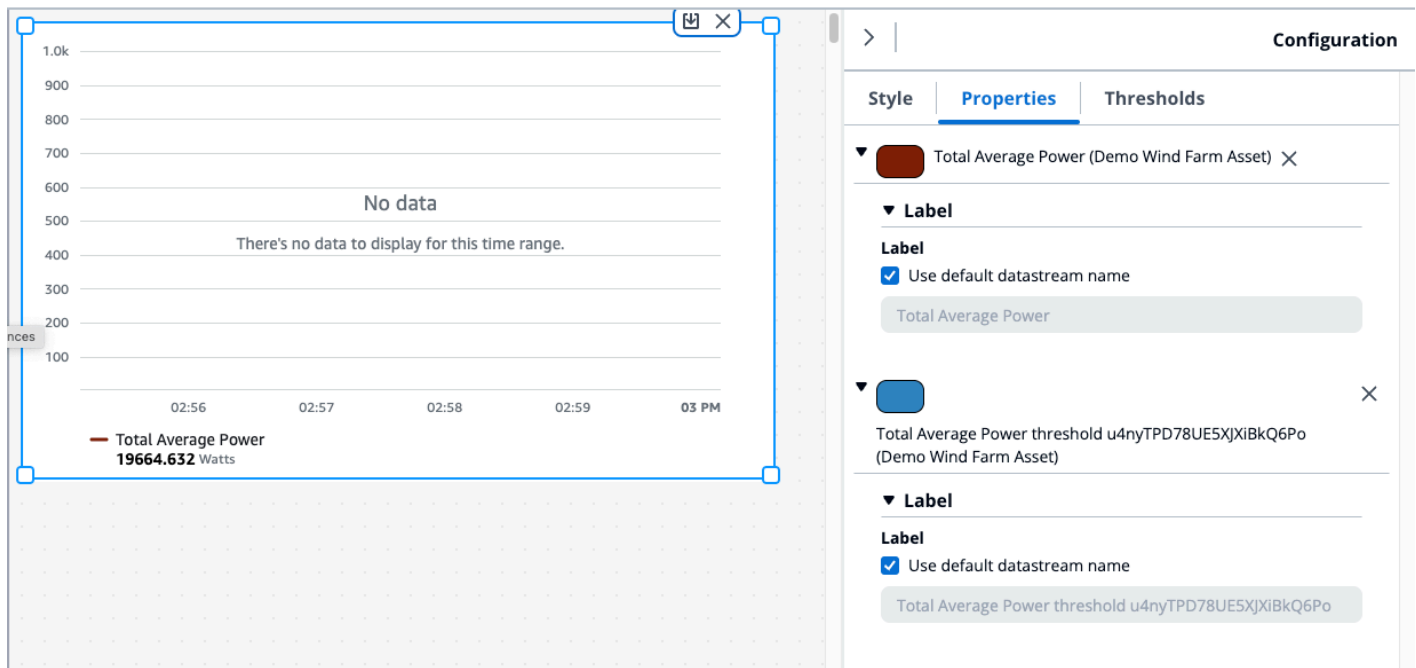


Configura i widget

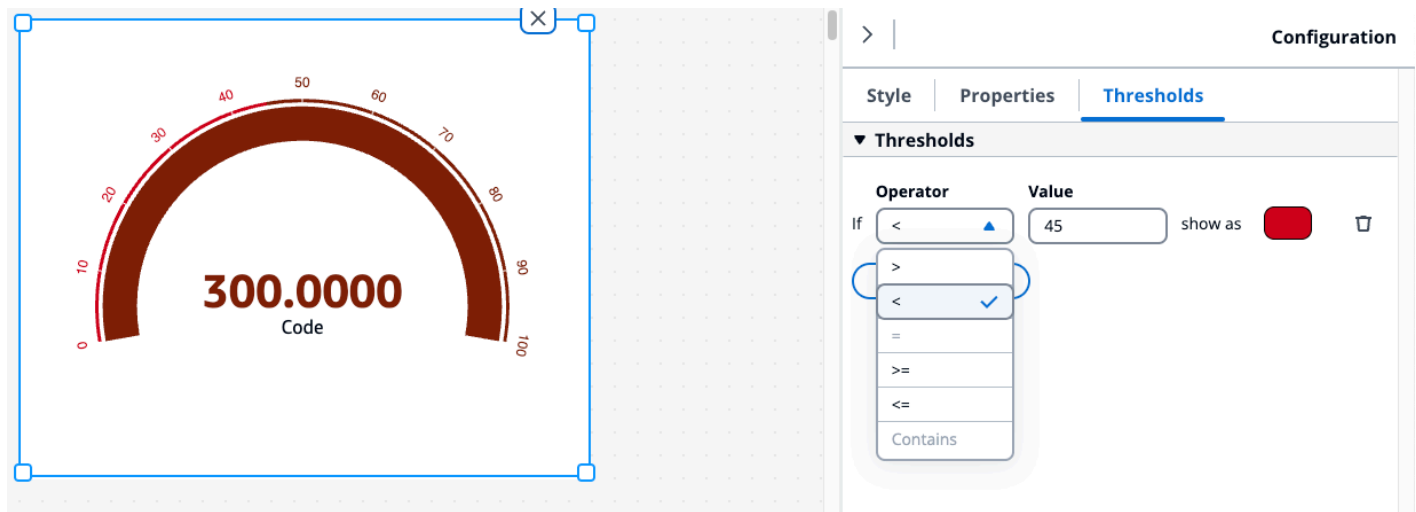
Una volta aggiunto il widget alla dashboard, puoi configurarlo scegliendo l'icona Configurazione nel pannello di destra.

- **Stile:** aggiungi un titolo nel titolo del widget. Widget diversi hanno configurazioni diverse. Di seguito sono elencati alcuni esempi.
 - **Widget della barra:**
 - **Risoluzione e aggregazione:** imposta qui i valori per la risoluzione e l'aggregazione.
 - **Formattazione dei dati:** imposta le posizioni decimali sul numero di decimali da visualizzare.
 - **Stile di visualizzazione:** seleziona i valori da visualizzare.
 - **Asse:** scegli di visualizzare l'asse.
 - **Widget di linea:**
 - **Risoluzione e aggregazione:** imposta qui i valori per la risoluzione e l'aggregazione.
 - **Formattazione dei dati:** imposta le posizioni decimali sul numero di decimali da visualizzare.
 - **Asse Y :** aggiungi un'etichetta e i valori Min e Max.
 - **Stile del widget:** seleziona i valori del tipo di linea, dello stile della linea, dello spessore della linea e della forma del punto dati.
 - **Legenda:** scegli Allineamento e Visualizza.
 - **Widget del misuratore:**
 - **Risoluzione e aggregazione:** imposta qui i valori per la risoluzione e l'aggregazione.
 - **Formattazione dei dati:** imposta le posizioni decimali sul numero di decimali da visualizzare.
 - **Stile di visualizzazione:** seleziona i valori da visualizzare.
 - **Asse Y:** aggiunge un'etichetta e i valori minimo e massimo.
 - **Font:** seleziona i valori della dimensione del carattere, della dimensione del carattere dell'unità e della dimensione del carattere dell'etichetta.

- **Proprietà:** tutte le proprietà dei widget sono elencate in questa sezione. Widget diversi hanno proprietà diverse. Di seguito sono elencati alcuni esempi.
 - **Widget di linea:**
 - **Etichetta:** scegli di utilizzare il nome predefinito del flusso di dati o di assegnare un nuovo nome.
 - **Stile:** imposta il tipo di linea, lo stile della linea sul numero di decimali da visualizzare.
 - **Asse Y:** seleziona i valori sullo stile predefinito, mostra i controlli dell'asse Y e imposta i valori minimo e massimo.
 - **Widget della tabella:**
 - **Etichetta:** scegli di utilizzare il nome predefinito del flusso di dati o di assegnare un nuovo nome.
 - **Widget della tabella:**
 - **Etichetta:** scegli di utilizzare il nome predefinito del flusso di dati o di assegnare un nuovo nome.



- Soglie: aggiungi una soglia per un widget. Widget diversi hanno configurazioni diverse. Di seguito sono elencati alcuni esempi.
 - Widget grafico a barre:
 - Scegli Aggiungi una soglia da aggiungere al widget.
 - Scegli Operatore e inserisci un valore per la soglia. Personalizza la soglia con un colore dalla tavolozza dei colori.
 - Puoi scegliere di applicare la soglia a tutti i dati.
 - Widget di linea:
 - Scegli Aggiungi una soglia da aggiungere al widget.
 - Scegli Operatore e inserisci un valore per la soglia. Personalizza la soglia con un colore dalla tavolozza dei colori.
 - Scegli come mostrare le soglie dal menu a discesa.
 - Widget indicatore:
 - Scegli Aggiungi una soglia da aggiungere al widget.
 - Scegli Operatore e inserisci un valore per la soglia. Personalizza la soglia con un colore dalla tavolozza dei colori.



Usa i widget

Puoi utilizzare i widget nella dashboard singolarmente o selezionandoli più volte.

Modifica i widget nella dashboard

Scegli un singolo widget e modificalo. Per modificare più widget nella dashboard, fai clic con Shift + con il tasto sinistro del mouse e seleziona tutti i widget nella dashboard. Una volta selezionati, gli utenti possono aggiungere nuovi flussi di dati e modificare il titolo del widget nelle impostazioni di configurazione dello stile. Il titolo viene modificato per tutti i widget nella dashboard.

Fate clic con il pulsante destro del mouse sull'area di disegno ed effettuate le seguenti operazioni:

- Copia: aggiungi una copia del widget all'area di disegno.
- Elimina: elimina il widget.
- Porta in primo piano: porta il widget selezionato in primo piano nell'area di disegno.
- Torna indietro: invia il widget selezionato sul retro dell'area di disegno.

Ridimensiona i widget

Ridimensiona i widget singolarmente o in gruppo selezionando più widget nella dashboard.

Per modificare le dimensioni dei widget:

- Per modificare le dimensioni di un singolo widget, selezionate il widget e trascinatelo per un angolo per modificarne le dimensioni.

- Per modificare le dimensioni di più widget, selezionateli premendo Shift + facendo clic con il tasto sinistro del mouse e trascinateli di lato per modificarne le dimensioni.

Elimina i widget nella dashboard

Elimina i widget singolarmente o in gruppo selezionando più widget nella dashboard.

Per eliminare i widget:

- Per eliminare un singolo widget, selezionatelo, fate clic con il pulsante destro del mouse e scegliete Elimina. Puoi anche selezionare e fare clic su X nell'angolo in alto a destra per eliminare il widget.
- Per eliminare più widget, selezionali premendo Shift + facendo clic con il pulsante sinistro del mouse, quindi fai clic con il pulsante destro del mouse e scegli Elimina.

Allarmi nei widget

Gli allarmi avvisano te e il tuo team quando le apparecchiature o i processi funzionano in modo non ottimale. Prestazioni ottimali di una macchina o di un processo significa che i valori di determinati parametri devono rientrare in un intervallo di limiti superiore e inferiore. Quando queste metriche non rientrano nell'intervallo operativo, gli operatori delle apparecchiature devono essere avvisati in modo che possano risolvere il problema. Gli allarmi consentono di identificare rapidamente i problemi e avvisano gli operatori per massimizzare le prestazioni delle apparecchiature e dei processi.

È possibile trovare un allarme associato a una risorsa nella scheda Modeled di Resource Explorer.

- Cercate e selezionate una risorsa.
- Scorri verso il basso oltre la tabella Data Streams fino alla sezione Alarm Data Streams ed espandi la tabella.
- Seleziona un allarme nella tabella Allarmi e scegli Aggiungi.

Argomenti

- [Allarmi in diversi widget](#)

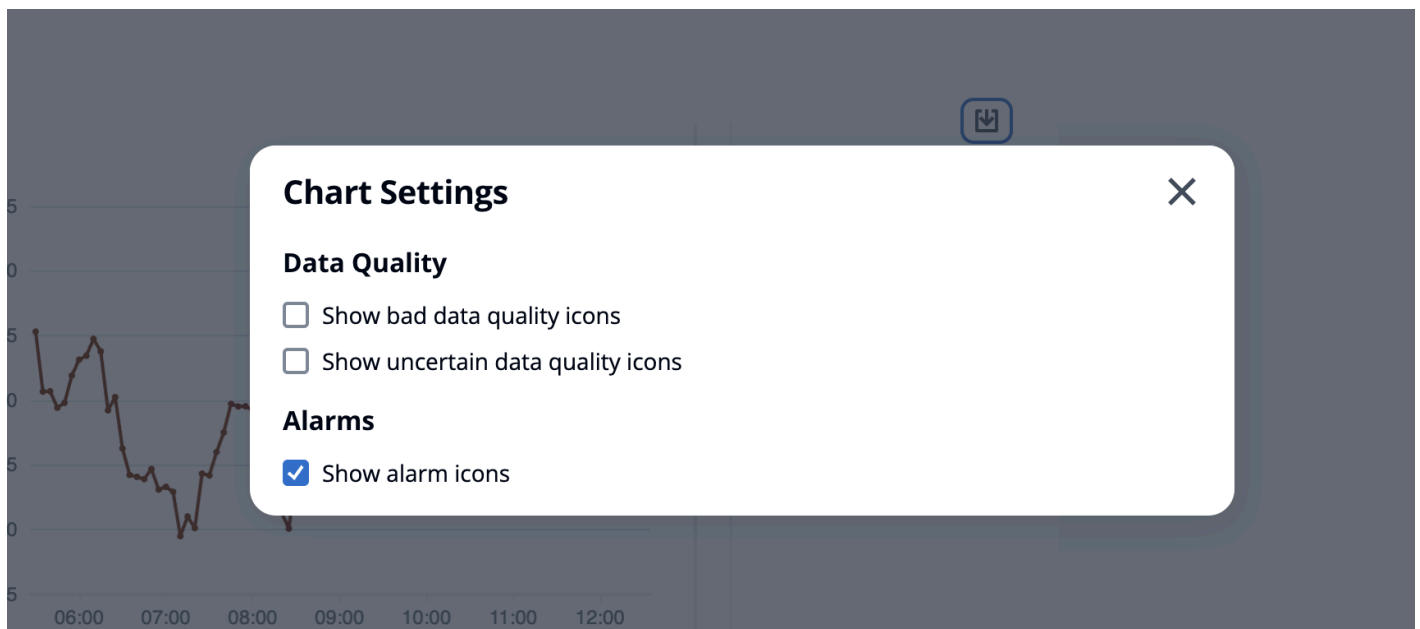
Allarmi in diversi widget

Per tutti i widget:

- Le impostazioni delle proprietà del flusso di dati dipendono dal tipo di proprietà aggiunta a un widget. Le proprietà del flusso di dati supportano completamente l'impostazione delle proprietà, mentre le proprietà degli allarmi attualmente non consentono configurazioni di impostazione delle proprietà.
- Se si aggiunge un flusso di dati di allarme, al grafico viene aggiunto anche il flusso di dati della proprietà di input associato. Se si rimuove il flusso di dati di allarme, viene rimossa anche la relativa proprietà di input.
- Per controllare singolarmente il flusso di dati delle proprietà di input di un allarme, è necessario aggiungerli entrambi separatamente.

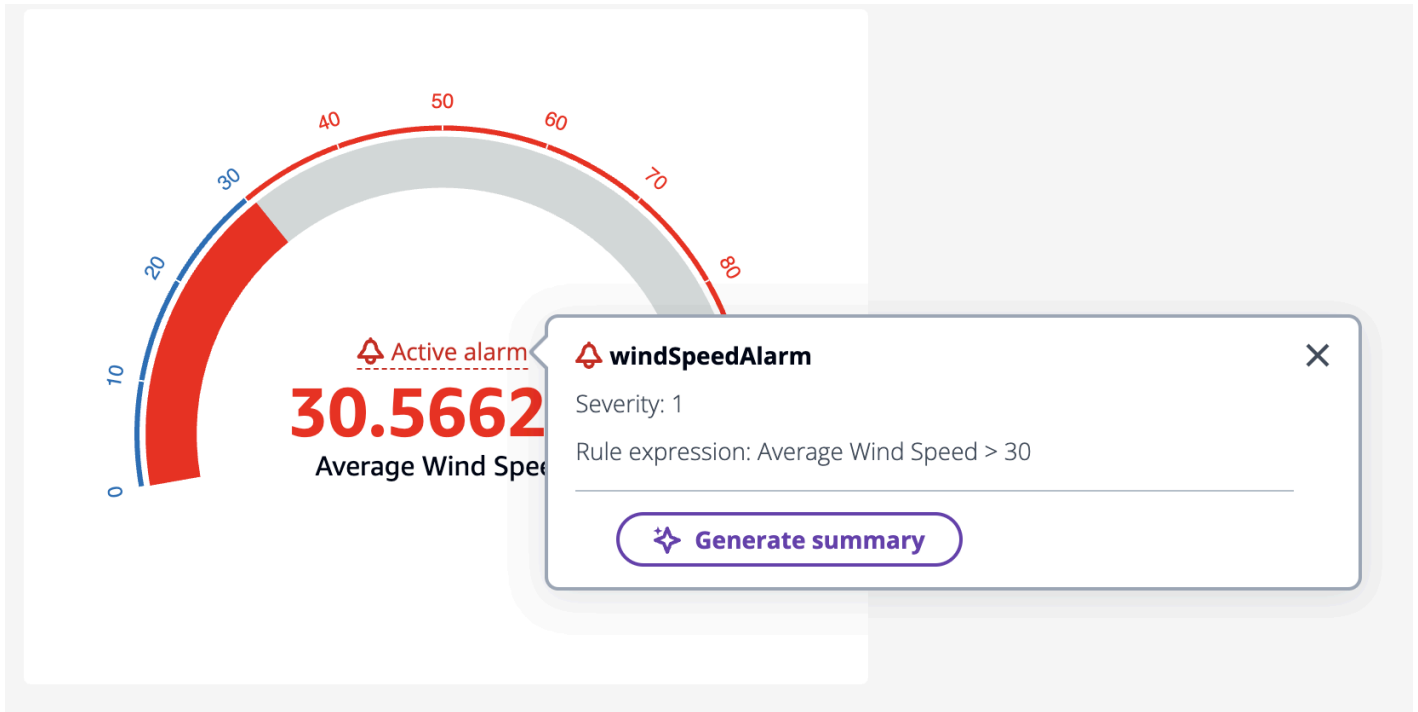
Gli esempi seguenti mostrano come alcuni widget utilizzano gli allarmi.

- Grafico a linee
 - L'allarme e il relativo flusso di dati delle proprietà di input vengono aggiunti al grafico.
 - È possibile visualizzare lo stato dell'allarme nella legenda del grafico e sotto forma di icone che si spostano sul flusso di dati quando l'allarme cambia stato.
 - È possibile disattivare le icone degli allarmi dalle impostazioni del grafico.

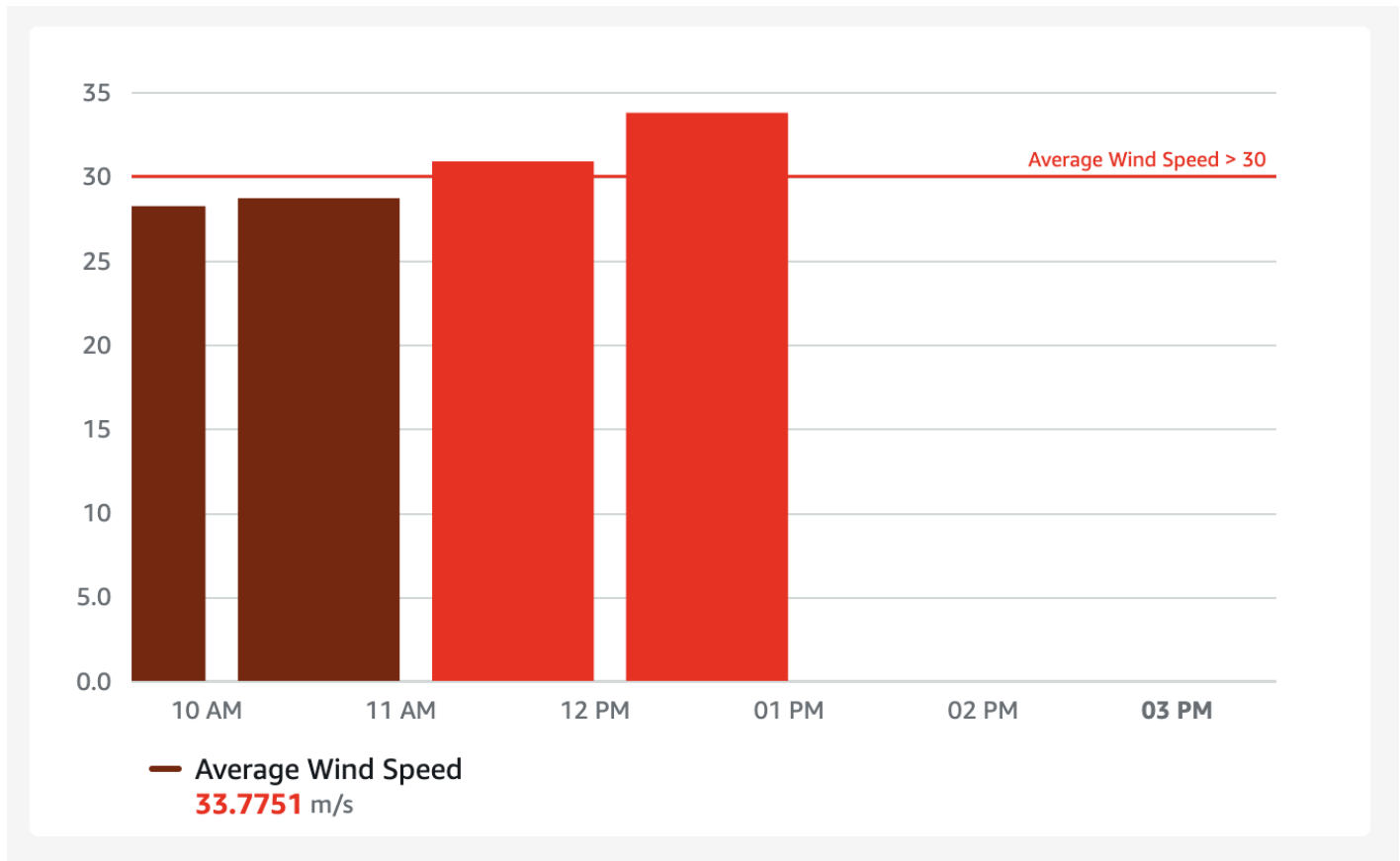


- KPI e indicatore
 - L'allarme e il relativo flusso di dati delle proprietà di input vengono aggiunti al widget scelto.
 - La soglia di allarme viene aggiunta al widget, che cambia colore in base alla sua configurazione.

- È possibile selezionare lo stato dell'allarme sul widget, visualizzare i dettagli dell'allarme e fare clic su Genera riepilogo per richiamare il comando AWS IoT SiteWise per ottenere un riepilogo dell'allarme.



- Tabella
 - L'allarme e la relativa proprietà di input vengono aggiunti come riga nella tabella.
- Grafico a barre
 - L'allarme viene aggiunto come soglia al grafico, che cambia il colore di qualsiasi flusso di dati che supera la soglia.
 - È possibile aggiungere separatamente tutti i flussi di dati associati.
 - Non è possibile interagire con l' AWS IoT SiteWise Assistente dal widget.

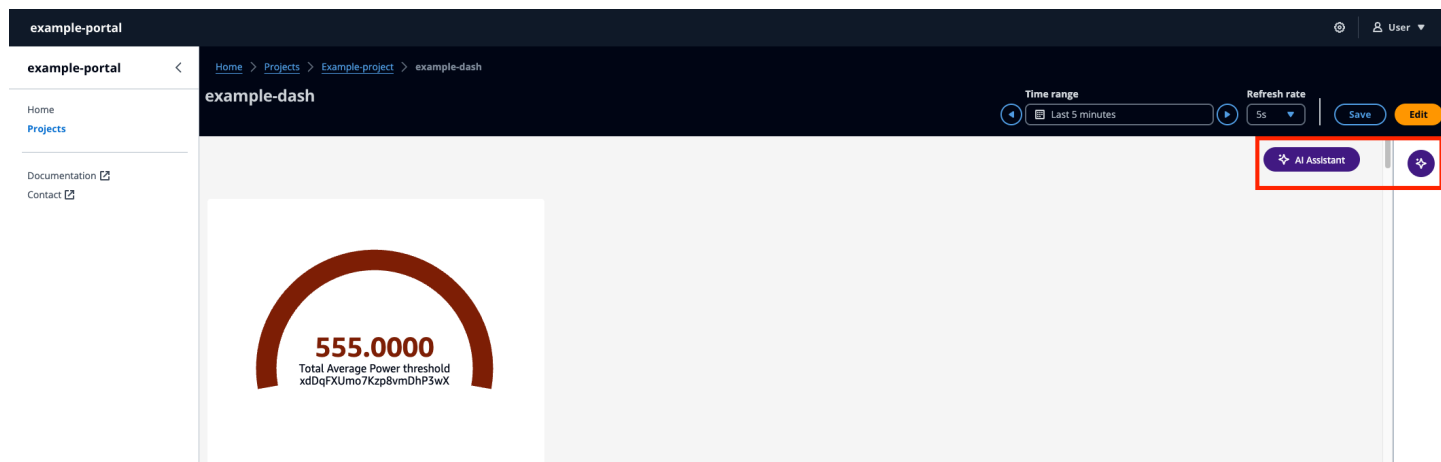


- Cronologia dello stato
 - L'allarme viene aggiunto come soglia alla timeline.
 - L'aggiunta dello stato dell'allarme e dei dati delle relative proprietà di input alla timeline è in corso.
 - Non è possibile interagire con l' AWS IoT SiteWise Assistente dal widget.

AWS IoT SiteWise Utilizzo dell'assistente nei widget

L' AWS IoT SiteWise Assistente è un assistente generativo basato sull'intelligenza artificiale. Consente a utenti come i responsabili degli impianti, gli ingegneri della qualità e i tecnici di manutenzione di ottenere informazioni, risolvere problemi e intraprendere azioni direttamente dai propri dati operativi e aziendali. L' AWS IoT SiteWise Assistente consolida le informazioni provenienti da AWS IoT dati, modelli di asset, manuali e documentazione in riepiloghi comprensibili degli eventi critici. Consente inoltre sessioni interattive di approfondimento con domande e risposte per facilitare la diagnosi, l'esplorazione delle cause alla radice e consigli guidati.

Il pulsante AWS IoT SiteWise Assistente si trova nell'angolo in alto a destra della dashboard. Fai clic su di esso per attivare l'Assistente. Può essere utilizzato solo con la modalità Anteprima del pannello di controllo.



Usa l' AWS IoT SiteWise Assistente nei seguenti scenari:

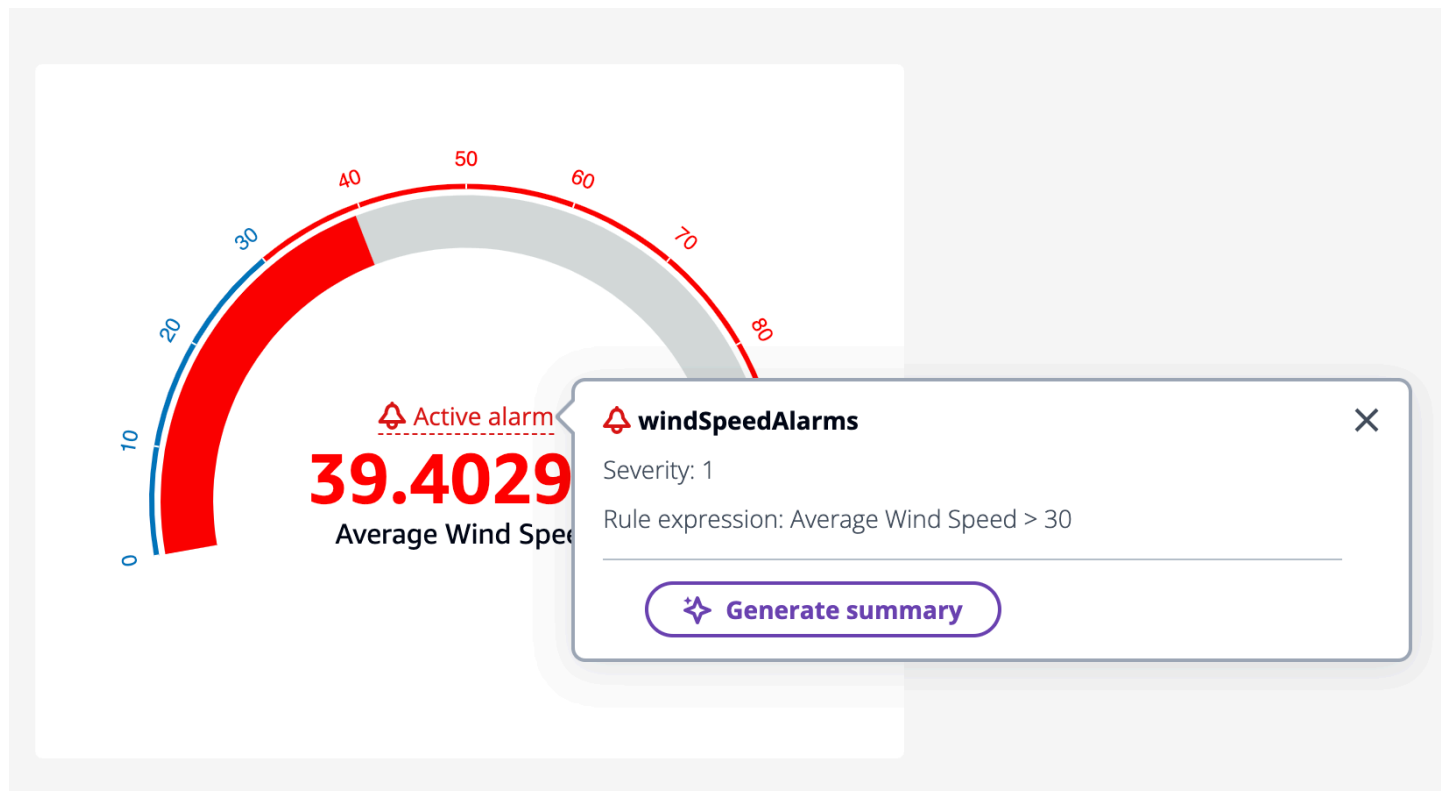
Argomenti

- [Caso d'uso: riepiloghi degli allarmi](#)
- [Caso d'uso: riepiloghi situazionali](#)
- [Caso d'uso: riepiloghi approfonditi](#)

Caso d'uso: riepiloghi degli allarmi

Riepiloga l'allarme corrente per un pannello selezionato sulla dashboard. Gli allarmi sono supportati dai widget Line, KPI, Gauge e Table. Scegli un widget con un allarme e riepilogalo.

- Seleziona Allarme attivo sul widget.
- L'espressione Severità e Regola viene visualizzata per l'allarme.
- Scegli Genera riepilogo per generare un riepilogo.



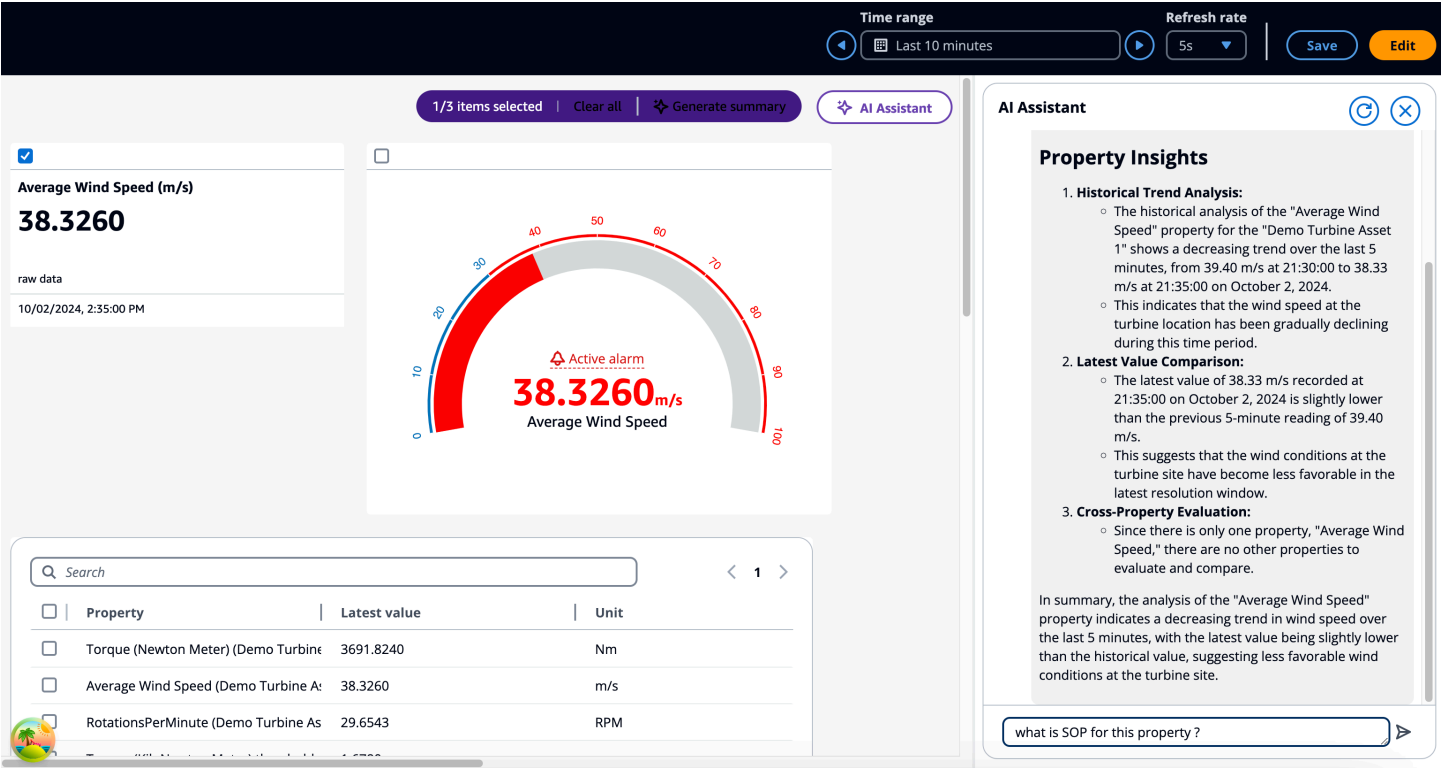
Caso d'uso: riepiloghi situazionali

Seleziona fino a tre widget da riepilogare. Possono essere una combinazione di widget e proprietà. Se ne sono selezionati più di tre, l'Assistente restituisce un errore.

Genera un riepilogo della situazione con AWS IoT SiteWise Assistant

1. Fai clic su AI Assistant. Visualizza un menu con tre opzioni.
 - a. Elementi selezionati: selezionane solo tre. Non è possibile selezionarne più di tre.
 - b. Cancella tutto: cancella la selezione.
 - c. Genera riepilogo: genera un riepilogo sugli elementi selezionati.
2. Scegli Genera riepilogo per generare il riepilogo sugli elementi selezionati.

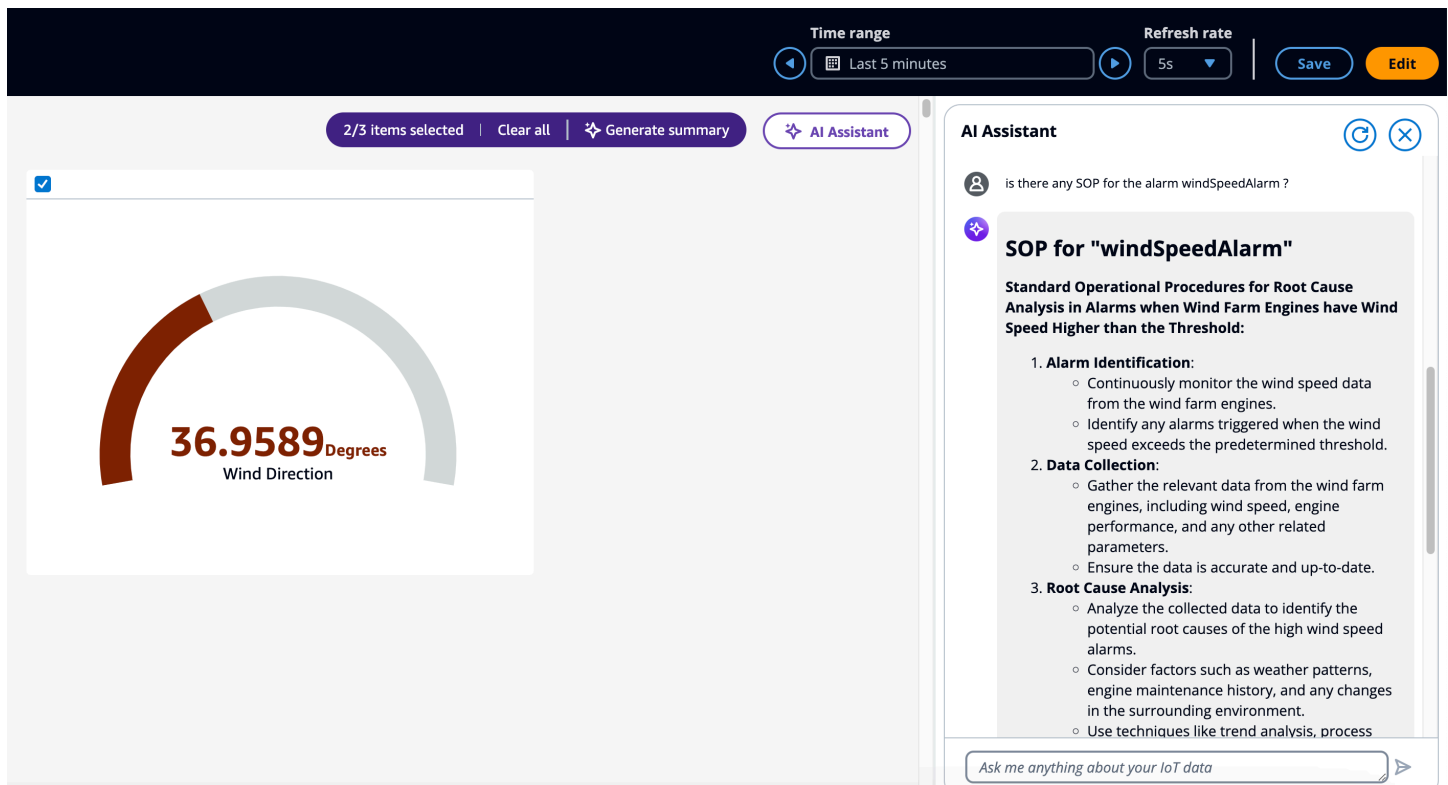
L'immagine seguente ha un widget selezionato e un riepilogo dell' AWS IoT SiteWise Assistente.



Caso d'uso: riepiloghi approfonditi

Questo è il caso d'uso in cui l'utente può eseguire un'analisi approfondita e accedere SOPs (Procedura operativa standard), ai manuali, alla documentazione e prendere in considerazione le fasi successive da intraprendere. Per l'esempio della sezione precedente, se l'utente sceglie di saperne di più sulla SOP di questa proprietà, chiedi all'Assistente informazioni sulla SOP per questa proprietà. Questo mostra all'utente le informazioni approfondite relative alla SOP.

L'esempio seguente mostra la risposta alla domanda "Esiste una SOP per l'allarme? windSpeedAlarm »



Esempi di domande da porre all'Assistente AWS IoT SiteWise

Note

- L' AWS IoT SiteWise Assistente deve utilizzare un set di dati con un indice [Amazon Kendra](#) per conoscenze e indicazioni a livello aziendale. Se non disponi di un indice Amazon Kendra, [consulta Creazione di un indice per crearne uno](#). L'aggiunta di un [set](#) di dati migliora la qualità della risposta dell'Assistente. Per ulteriori informazioni, consulta [Creare un set](#) di dati.
- Alcune domande richiedono AWS IoT TwinMaker l'integrazione. Vedi [Integrate AWS IoT TwinMaker e AWS IoT SiteWise](#) per i dettagli.

Alcune domande di follow-up da porre all'Assistente dopo aver visualizzato un riepilogo degli allarmi nella dashboard, nell'ambito della stessa conversazione.

- Mostra i dettagli della risorsa riportati nel riepilogo riportato sopra?
- Qual è il percorso gerarchico dalla radice alla risorsa menzionata?
- Quali sono i beni discendenti dipendenti dalla risorsa menzionata?

- Quali sono gli asset dipendenti dalla risorsa menzionata che dispongono di allarmi attivi?
- Trova tutti gli asset con allarmi attivi.

Alcune domande di follow-up da porre all'Assistente dopo aver visualizzato un riepilogo della proprietà nella dashboard, nell'ambito della stessa conversazione.

- Esegui la stessa analisi nelle ultime 24 ore.
- Trova la documentazione relativa alle proprietà sopra menzionate.
- Fornisci i dettagli dell'ID dell'asset 1da67d28-14f8-4f71-a06a-386f0425a21d/nome dell'asset Demo Turbine Asset 1.

AWS IoT SiteWise Richiama l'Assistente dall'API.

- Genera un riepilogo degli allarmi per il nome dell'allarme windSpeedAlarmnell'asset id. d591e153-e5cf-4206-96bb-ce3c119d9d2d
- Genera un riepilogo degli allarmi delle ultime 12 ore/2 giorni/1 settimana per il nome windSpeedAlarmdell'allarme nell'asset id. d591e153-e5cf-4206-96bb-ce3c119d9d2d
- Genera un riepilogo delle proprietà per l'id della proprietà in asset id ab187fb7-d74b-44d9-bd9b-f2f19a9137cc d591e153-e5cf-4206-96bb-ce3c119d9d2d
- Genera un riepilogo della proprietà per le ultime 12 ore/2 giorni/1 settimana per l'ID della proprietà nell'asset idab187fb7-d74b-44d9-bd9b-f2f19a9137cc. d591e153-e5cf-4206-96bb-ce3c119d9d2d
- Trova gli asset con il nome dell'asset Turbine.
- Dammi i valori correnti delle proprietà id 5356168c-3390-456f-802c-9f6e047810d4 in asset idd591e153-e5cf-4206-96bb-ce3c119d9d2d,3cbb084e-1ded-4b08-9f21-1b47b2fb86fd.
- Qual è la relazione tra asset id d591e153-e5cf-4206-96bb-ce3c119d9d2d e asset id3cbb084e-1ded-4b08-9f21-1b47b2fb86fd.
- Trova la documentazione su come risolvere il problema dei bassi regimi di rotazione delle turbine eoliche.
- Genera un riepilogo delle proprietà per l'alias della proprietà. WindSpeed
- Quali sono i controlli pre-operativi in base alla mia knowledge base?

Interroga i dati da AWS IoT SiteWise

Puoi utilizzare le operazioni AWS IoT SiteWise API per interrogare i valori correnti, i valori storici e gli aggregati delle proprietà degli asset su intervalli di tempo specifici.

Utilizza queste funzionalità per ottenere informazioni dettagliate sui tuoi dati. Ad esempio, scopri tutte le tue risorse con un determinato valore di proprietà o crea una rappresentazione personalizzata dei tuoi dati. Puoi anche utilizzare le operazioni API per sviluppare soluzioni software che si integrano con i dati industriali archiviati nelle tue AWS IoT SiteWise risorse. È inoltre possibile esplorare i dati degli asset in tempo reale in AWS IoT SiteWise Monitor. Per informazioni su come configurare SiteWise Monitor, consulta [Monitora i dati con AWS IoT SiteWise Monitor](#).

Le operazioni descritte in questa sezione restituiscono oggetti di valore delle proprietà che contengono strutture di timestamp, quality, value (TQV):

- `timestamp` contiene l'orario UTC (epoca (Unix epoch)) corrente in secondi, con offset in nanosecondi.
- `quality` contiene una delle seguenti stringhe che indicano la qualità del punto dati:
 - `GOOD`— I dati non sono interessati da alcun problema.
 - `BAD`— I dati sono interessati da un problema come il guasto del sensore.
 - `UNCERTAIN`— I dati sono influenzati da un problema come l'imprecisione del sensore.
- `value` contiene uno dei seguenti campi, a seconda del tipo di proprietà:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`

Argomenti

- [Interroga i valori correnti delle proprietà degli asset in AWS IoT SiteWise](#)
- [Interroga i valori storici delle proprietà degli asset in AWS IoT SiteWise](#)
- [Interroga gli aggregati delle proprietà degli asset in AWS IoT SiteWise](#)
- [AWS IoT SiteWise linguaggio di interrogazione](#)

Interroga i valori correnti delle proprietà degli asset in AWS IoT SiteWise

Questo tutorial mostra due modi per ottenere il valore corrente di una proprietà di un asset. È possibile utilizzare la AWS IoT SiteWise console o utilizzare l'API in AWS Command Line Interface (AWS CLI).

Argomenti

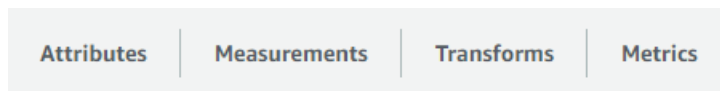
- [Interroga il valore corrente di una proprietà dell'asset \(console\)](#)
- [Interroga il valore corrente di una proprietà dell'asset \(\)AWS CLI](#)

Interroga il valore corrente di una proprietà dell'asset (console)

È possibile utilizzare la AWS IoT SiteWise console per visualizzare il valore corrente di una proprietà di un asset.

Per ottenere il valore corrente di una proprietà di asset (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Asset.
3. Scegli l'asset con la proprietà su cui eseguire la query.
4. Scegliete l'icona a forma di freccia per espandere una gerarchia di risorse e trovare la risorsa desiderata.
5. Scegli la scheda per il tipo di proprietà. Ad esempio, scegli Misurazioni per visualizzare il valore corrente di una proprietà di misurazione.



6. Trova la proprietà da visualizzare. Il valore corrente viene visualizzato nella colonna Valore più recente.

Interroga il valore corrente di una proprietà dell'asset ()AWS CLI

È possibile utilizzare AWS Command Line Interface (AWS CLI) per interrogare il valore corrente di una proprietà dell'asset.

Utilizzate l'[GetAssetPropertyValue](#) operazione per interrogare il valore corrente di una proprietà dell'asset.

Per identificare la proprietà di un asset, specificate una delle seguenti opzioni:

- La proprietà `assetId` e `propertyId` della risorsa a cui vengono inviati i dati.
- `propertyAlias`, che è un alias del flusso di dati (ad esempio, `/company/windfarm/3/turbine/7/temperature`). Per utilizzare questa opzione, è necessario prima impostare l'alias della proprietà dell'asset. Per impostare gli alias delle proprietà, vedere. [Gestisci i flussi di dati per AWS IoT SiteWise](#)

Per ottenere il valore corrente di una proprietà di un asset ()AWS CLI

- Esegui il comando seguente per ottenere il valore corrente della proprietà di asset. Sostituisci *asset-id* con l'ID della risorsa e *property-id* con l'ID della proprietà.

```
aws iotsitewise get-asset-property-value \  
  --asset-id asset-id \  
  --property-id property-id
```

L'operazione restituisce una risposta contenente il TQV attuale della proprietà nel formato seguente.

```
{  
  "propertyValue": {  
    "value": {  
      "booleanValue": Boolean,  
      "doubleValue": Number,  
      "integerValue": Number,  
      "stringValue": "String",  
      "nullValue": {  
        "valueType": "String"  
      }  
    },  
    "timestamp": {  
      "timeInSeconds": Number,  
      "offsetInNanos": Number  
    },  
    "quality": "String"  
  }  
}
```

```
}
```

Interroga i valori storici delle proprietà degli asset in AWS IoT SiteWise

È possibile utilizzare l'[GetAssetPropertyValueHistory](#) operazione AWS IoT SiteWise API per interrogare i valori storici di una proprietà dell'asset.

Per identificare una proprietà di un asset, specificate una delle seguenti opzioni:

- La proprietà `assetId` e `propertyId` della risorsa a cui vengono inviati i dati.
- `propertyAlias`, che è un alias del flusso di dati (ad esempio, `/company/windfarm/3/turbine/7/temperature`). Per utilizzare questa opzione, è necessario prima impostare l'alias della proprietà dell'asset. Per impostare gli alias delle proprietà, vedere. [Gestisci i flussi di dati per AWS IoT SiteWise](#)

Passa i seguenti parametri per affinare i risultati:

- `startDate`— L'inizio esclusivo dell'intervallo da cui interrogare i dati storici, espresso in secondi nell'epoca Unix.
- `endDate`— La fine inclusiva dell'intervallo da cui interrogare i dati storici, espressa in secondi nell'epoca Unix.
- `maxResults`— Il numero massimo di risultati da restituire in una richiesta. I valori predefiniti sono i risultati. 20
- `nextToken`— Un token di impaginazione restituito da una precedente chiamata di questa operazione.
- `timeOrdering`— L'ordine da applicare ai valori restituiti: ASCENDING o. DESCENDING
- `qualities`— La qualità per filtrare i risultati in base a: GOODBAD, oUNCERTAIN.

Per interrogare la cronologia dei valori di una proprietà dell'asset (AWS CLI)

1. Esegui il comando seguente per ottenere la cronologia dei valori per la proprietà di asset. Questo comando esegue una query per la cronologia della proprietà su un intervallo di 10 minuti. Sostituisci *asset-id* con l'ID della risorsa e *property-id* con l'ID della proprietà. Sostituisci i parametri di data con l'intervallo per la query.


```
aws iotsitewise get-asset-property-value-history \  
  --asset-id asset-id \  
  --property-id property-id \  
  --start-date 1575216000 \  
  --end-date 1575216600
```

L'operazione restituisce una risposta che contiene lo storico TQVs della proprietà nel seguente formato:

```
{  
  "assetPropertyValueHistory": [  
    {  
      "value": {  
        "booleanValue": Boolean,  
        "doubleValue": Number,  
        "integerValue": Number,  
        "stringValue": "String",  
        "nullValue": {  
          "valueType": "String"  
        }  
      },  
      "timestamp": {  
        "timeInSeconds": Number,  
        "offsetInNanos": Number  
      },  
      "quality": "String"  
    },  
    {  
      "value": {  
        "booleanValue": Boolean,  
        "doubleValue": Number,  
        "integerValue": Number,  
        "stringValue": "String",  
        "nullValue": {  
          "valueType": "String"  
        }  
      },  
      "timestamp": {  
        "timeInSeconds": Number,  
        "offsetInNanos": Number  
      },  
      "quality": "String"  
    },  
    {  
      "value": {  
        "booleanValue": Boolean,  
        "doubleValue": Number,  
        "integerValue": Number,  
        "stringValue": "String",  
        "nullValue": {  
          "valueType": "String"  
        }  
      },  
      "timestamp": {  
        "timeInSeconds": Number,  
        "offsetInNanos": Number  
      },  
      "quality": "String"  
    }  
  ],  
  "nextToken": "String"  
}
```

2. Se esistono più valori, è possibile passare il token di impaginazione dal `nextToken` campo a una successiva chiamata all'[GetAssetPropertyValueHistory](#) operazione.

Interroga gli aggregati delle proprietà degli asset in AWS IoT SiteWise

AWS IoT SiteWise calcola automaticamente i valori aggregati delle proprietà degli asset, che sono un insieme di metriche di base calcolate su più intervalli di tempo. AWS IoT SiteWise calcola i seguenti aggregati ogni minuto, ora e giorno per le proprietà degli asset:

- **media:** la media (media) dei valori di una proprietà in un intervallo di tempo.
- **count:** il numero di punti dati per una proprietà in un intervallo di tempo.
- **maximum** — Il massimo dei valori di una proprietà in un intervallo di tempo.
- **minimo** — Il minimo dei valori di una proprietà in un intervallo di tempo.
- **deviazione standard:** la deviazione standard dei valori di una proprietà in un intervallo di tempo.
- **sum** — La somma dei valori di una proprietà in un intervallo di tempo.

Per le proprietà non numeriche, come stringhe e valori booleani, calcola solo l'aggregato di conteggio. AWS IoT SiteWise

Puoi anche calcolare parametri personalizzati per i dati di asset. Con le proprietà metriche, definisci aggregazioni specifiche per la tua operazione. Le proprietà metriche offrono funzioni di aggregazione e intervalli di tempo aggiuntivi che non sono precalcolati per l'API. AWS IoT SiteWise Per ulteriori informazioni, consulta [Aggrega i dati provenienti da proprietà e altre risorse \(metriche\)](#).

Argomenti

- [Aggregati per una proprietà di asset \(API\)](#)
- [Aggregati per una proprietà di asset \(\)AWS CLI](#)

Aggregati per una proprietà di asset (API)

Utilizza l' AWS IoT SiteWise API per ottenere aggregati per una proprietà di asset.

Utilizzate l'[GetAssetPropertyAggregates](#) operazione per interrogare gli aggregati di una proprietà di un asset.

Per identificare la proprietà di un asset, specificate una delle seguenti opzioni:

- La proprietà `assetId` e `propertyId` della risorsa a cui vengono inviati i dati.

- `IlpropertyAlias`, che è un alias del flusso di dati (ad esempio, `/company/windfarm/3/turbine/7/temperature`). Per utilizzare questa opzione, è necessario prima impostare l'alias della proprietà dell'asset. Per impostare gli alias delle proprietà, vedere [Gestisci i flussi di dati per AWS IoT SiteWise](#)

È necessario passare i seguenti parametri obbligatori:

- `aggregateTypes`— L'elenco degli aggregati da recuperare. Puoi specificare uno qualsiasi di AVERAGE, COUNT, MAXIMUM, MINIMUM, STANDARD_DEVIATION e SUM.
- `resolution`— L'intervallo di tempo per il quale recuperare la metrica: 1m (1 minuto), (15 minuti), 15m (1 ora) o 1h 1d (1 giorno).
- `startDate`— L'inizio esclusivo dell'intervallo da cui interrogare i dati storici, espresso in secondi nell'epoca Unix.
- `endDate`— La fine inclusiva dell'intervallo da cui interrogare i dati storici, espressa in secondi nell'epoca Unix.

È anche possibile passare uno dei seguenti parametri per perfezionare i risultati:

- `maxResults`— Il numero massimo di risultati da restituire in una richiesta. I valori predefiniti sono i risultati. 20
- `nextToken`— Un token di impaginazione restituito da una precedente chiamata di questa operazione.
- `timeOrdering`— L'ordine da applicare ai valori restituiti: ASCENDING o. DESCENDING
- `qualities`— La qualità per filtrare i risultati in base a: GOODBAD, oUNCERTAIN.

Note

L'[GetAssetPropertyAggregates](#) operazione restituisce un TQV con un formato diverso rispetto alle altre operazioni descritte in questa sezione. La struttura di `value` annovera un campo per ogni `aggregateTypes` incluso nella richiesta. `timestamp` contiene l'orario in cui si è verificata l'aggregazione, in secondi nel formato epoca (Unix epoch).

Aggregati per una proprietà di asset (AWS CLI)

Per interrogare gli aggregati per una proprietà di un asset (AWS CLI)

1. Esegui il comando seguente per ottenere aggregati per la proprietà di asset. Questo comando esegue la query per la media e la somma con una risoluzione di 1 ora per uno specifico intervallo di 1 ora. Sostituisci *asset-id* con l'ID della risorsa e *property-id* con l'ID della proprietà. Sostituisci i parametri con gli aggregati e l'intervallo per la query.

```
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
```

L'operazione restituisce una risposta che contiene lo storico TQVs della proprietà nel formato seguente. La risposta include solo gli aggregati richiesti.

```
{
  "aggregatedValues": [
    {
      "timestamp": Number,
      "quality": "String",
      "value": {
        "average": Number,
        "count": Number,
        "maximum": Number,
        "minimum": Number,
        "standardDeviation": Number,
        "sum": Number
      }
    }
  ],
  "nextToken": "String"
}
```

2. Se esistono più valori, è possibile passare il token di impaginazione dal `nextToken` campo a una successiva chiamata all'[GetAssetPropertyAggregates](#) operazione.

 Note

Se l'intervallo di query contiene un `null` valore TQVs, consulta [AssetPropertyValueAPI](#). Tutte le statistiche, tranne il conteggio, danno come risultato una `null` risposta simile alle statistiche di String TQVs. Se l'intervallo di query contiene `Double.NaN` un tipo doppio TQVs, tutti i calcoli tranne il conteggio daranno come risultato un `Double.NaN`.

AWS IoT SiteWise linguaggio di interrogazione

Con l'operazione dell'[ExecuteQuery](#) API di recupero AWS IoT SiteWise dei dati, è possibile recuperare informazioni sulle definizioni strutturali dichiarative e sui dati delle serie temporali ad esse associati, da quanto segue:

- modelli
- risorse
- misurazioni
- metriche
- trasforma
- aggregati

Questo può essere fatto con istruzioni di query simili a SQL, in un'unica richiesta API.

 Note

Questa funzionalità è disponibile in tutte le regioni in cui entrambe AWS IoT SiteWise AWS IoT TwinMaker sono disponibili, ad eccezione di AWS GovCloud (Stati Uniti occidentali).

Argomenti

- [Prerequisiti](#)
- [Riferimento al linguaggio di interrogazione per AWS IoT SiteWise](#)

Prerequisiti

AWS IoT SiteWise richiede le autorizzazioni per l'integrazione in AWS IoT TwinMaker modo da poter organizzare e modellare i dati industriali.

Prima di poter recuperare informazioni su modelli, risorse, misurazioni, metriche, trasformazioni e aggregati, assicuratevi che siano soddisfatti i seguenti prerequisiti:

- Ruoli collegati ai servizi per entrambi e configurazione nel tuo account. AWS IoT SiteWise AWS IoT TwinMaker AWS Per ulteriori informazioni sui ruoli collegati ai servizi, consulta [Create a service-linked role nella](#) IAM User Guide.
- Un' AWS IoT SiteWise integrazione abilitata per il tuo ruolo IAM. Per ulteriori informazioni, consulta [Integrazione AWS IoT SiteWise e AWS IoT TwinMaker](#).
- Un' AWS IoT TwinMaker area di lavoro con ID `IoTSiteWiseDefaultWorkspace` nel tuo account nella regione. Per ulteriori informazioni, consulta [Using the IoTSiteWiseDefaultWorkspace in the AWS IoT TwinMaker](#) User Guide.
- Sono abilitate le modalità di prezzo in bundle standard o a più livelli. AWS IoT TwinMaker Per ulteriori informazioni, consulta le [modalità di AWS IoT TwinMaker prezzo di Switch](#) nella Guida per l'AWS IoT TwinMaker utente.

Riferimento al linguaggio di interrogazione per AWS IoT SiteWise

AWS IoT SiteWise supporta un linguaggio di interrogazione avanzato per lavorare con i dati. I tipi di dati, gli operatori, le funzioni e i costrutti disponibili sono descritti nei seguenti argomenti.

Vedere [Query di esempio](#) per scrivere interrogazioni con il linguaggio di AWS IoT SiteWise interrogazione.

Argomenti

- [Comprendi le viste di riferimento delle query](#)
- [Tipi di dati supportati](#)
- [Recupera i dati con un'istruzione SELECT](#)
- [Operatori logici](#)
- [Operatori di confronto](#)
- [Query di esempio](#)

Comprendi le viste di riferimento delle query

Questa sezione fornisce informazioni per aiutarti a comprendere le visualizzazioni AWS IoT SiteWise, ad esempio i metadati dei processi e i dati di telemetria.

Le tabelle seguenti forniscono i nomi e le descrizioni delle viste.

Modello di dati

Nome della vista	Descrizione della vista
asset	Contiene informazioni sulla derivazione dell'asset e del modello.
asset_property	Contiene informazioni sulla struttura della proprietà dell'asset.
raw_time_series	Contiene i dati storici delle serie temporali.
latest_value_time_series	Contiene il valore più recente della serie temporale.
precomputed_aggregates	Contiene i valori delle proprietà aggregati degli asset calcolati automaticamente. Sono un insieme di metriche di base calcolate su più intervalli di tempo.

Le viste seguenti elencano i nomi delle colonne per le query insieme ai dati di esempio.

Visualizza: risorsa

asset_id	nome_risorsa	descrizione_risorsa	asset_model_id
88898498-0b8b-42b5-bf57-16180bc3d3a0	WindTurbine A	WindTurbine Risorsa A	17847250-5bf0-4f74-b775-cc03f05e7cb8
17847250-5bf0-4f74-b775-cc03f05e7cb8	Modello di investimento per turbine eoliche	Rappresenta una turbina in un parco eolico.	

Visualizza: asset_property

property_id	id_risorsa	nome_proprietà	property_alias	asset_composite_model_id	
b29be434-b000-4d74-b809-75287d83bcd6	88898498-0b8b-42b5-bf57-16180bc3d3a0	temperatura del motore	Rochester2/44///Line-5/Bus-2/Machine-5/Temperature		
3b458f00-24e7-458a-b4e8-c6026eff654a	88898498-0b8b-42b5-bf57-16180bc3d3a0	direzione del vento	/company/windfarm/3/turbine/7/winddirection	2f458n00-56e7-458h-b4e8-c6026eff985g	

Visualizza: raw_time_series

asset_id	id_proprietà	property_alias	timestamp dell'evento	qualità	valore_bonoleano	valore_int	valore_doppio	valore_stringa
888984980b8b-42b5-bf57-16180bc3d3a0	b29be434-b000-4d74-b809-75287d83bcd6	Rochester2/44///Line-5/Bus-2/Machine-5/Temperature	157521960	BUONO			115.0	
888984980b8b-42b5-bf57-16180bc3d3a0	3b458f00-24e7-458a-b4e8-c6026eff654a	/company/windfarm/3/turbine/7/winddirection	157521967	BUONO			348,75	

asset_id	id_proprietà	property_alias	timestamp dell'evento	qualità	valore_booleano	valore_int	valore_double	valore_stringa
-bf57-16180bc3d3a	-b4e8-c6026eff654a	windfarm3/turbine/7/winddirection						

Note

È necessario includere una clausola di filtro `event_timestamp` nella colonna per interrogare la vista. `raw_time_series` Si tratta di un filtro obbligatorio e senza di esso l'interrogazione avrà esito negativo.

Example query

```
SELECT event_timestamp, double_value FROM raw_time_series WHERE event_timestamp > 1234567890
```

Visualizza: `latest_value_time_series`

asset_id	id_proprietà	property_alias	timestamp dell'evento	qualità	valore_booleano	valore_int	valore_double	valore_stringa
888984980b8b-42b1-bf57-16180bc3d3a	3b458f00-24e7-458a-b4e8-c6026eff654a	/company, windfarm3/turbine/7/winddirection	157521960	BUONO			355,39	

Visualizza: precomputed_aggregates

asset_id	id_proprietà	property alias	timestamp dell'evento	qualità	risoluzione	somma	valore_integgero	valore_dio	valore_ssimo	valore_nimo	stdev_valore
8889840b8b-4	b29be4b000-4	Roche: 2/44/, 0	157521	BUON	15 min	1105,4	15	73,4	80,6	68	3,64
-	-	Li									
bf57-16	b809-7	ne-5/									
80bc3d	87d83t	Bus-2/									
		Machi									
		-5/									
		Tempe:									
		ature									

Tipi di dati supportati

AWS IoT SiteWise il linguaggio di interrogazione supporta i seguenti tipi di dati.

Valore scalare

Tipo di dati	Descrizione
STRING	Una stringa di lunghezza massima di 1024 byte.
INTEGER	Un numero intero con segno a 32 bit con un intervallo da $-2,147,483,648$ to $2,147,483,647$
DOUBLE	Un numero a virgola mobile con intervallo da -10^{100} to 10^{100} IEEE 754 precisione doppia.
BOOLEAN	true o false.

`Null` valore: un valore booleano `true` che indica la mancanza di dati definiti.

Note

I dati a doppia precisione non sono esatti. Alcuni valori non vengono convertiti esattamente e non rappresenteranno tutti i numeri reali a causa della precisione limitata. I dati a virgola mobile nella query potrebbero non corrispondere allo stesso valore rappresentato internamente. Il valore viene arrotondato se la precisione di un numero di input è troppo elevata.

Recupera i dati con un'istruzione SELECT

L'istruzione `SELECT` viene utilizzata per recuperare dati da una o più viste. AWS IoT SiteWise supporta un implicito `JOIN` delle visualizzazioni. È possibile elencare le viste da unire (nella `FROM` clausola dell'istruzione `SELECT`), utilizzando le virgole per separarle.

Example

Usa la seguente dichiarazione: `SELECT`

```
SELECT select_expr [, ...]  
[ FROM from_item [, ...] ]  
[ WHERE [LIKE condition ESCAPE condition] ]
```

Nell'esempio precedente, la `LIKE` clausola specifica le condizioni di ricerca e filtraggio utilizzando wild card. AWS IoT SiteWise supporta percentage (%) come carattere wild card.

Example da usare % in una condizione:

```
Prefix search: String%  
Infix search: %String%  
Suffix search: %String
```

Example per cercare una risorsa:

```
SELECT asset_name, asset_description FROM asset WHERE asset_name LIKE 'Wind%'
```

Example per cercare una risorsa utilizzando una condizione ESCAPE:

```
SELECT asset_name, asset_description FROM asset WHERE asset_name LIKE 'room\%' ESCAPE  
'\'
```

Operatori logici

AWS IoT SiteWise supporta i seguenti operatori logici.

Operatori logici

Operatore	Descrizione	Esempio
AND	TRUEse entrambi i valori sono veri	a AND b

Se a o b èFALSE, l'espressione precedente restituisce false. ANDAffinché un operatore restituisca vero, sia a che b devono essere veri.

Example

```
SELECT a.asset_name  
FROM asset as a, latest_value_time_series as t  
WHERE t.int_value > 30 AND t.event_timestamp > 1234567890
```

Operatori di confronto

AWS IoT SiteWise supporta i seguenti operatori di confronto. Tutte le operazioni di confronto sono disponibili per i tipi di dati incorporati e vengono valutate in modo booleano.

Operatori logici

Operatore	Descrizione
<	Minore di
>	Maggiore di
<=	Minore o uguale a

Operatore	Descrizione
>=	Maggiore o uguale a
=	Equals
!=	Non uguale

Tabella di verità delle operazioni di confronto per valori non numerici

Tipo	Digitare >= x	Tipo <= x	Tipo > x	Tipo < x	Tipo = x	Digitale = x
NaN	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
NULL	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE

Ci sono alcuni predicati che si comportano come operatori ma hanno una sintassi speciale. Vedi sotto per alcuni esempi.

Predicati di confronto

Operatore	Descrizione
IS NULL	Verifica se un valore è NULL.
IS NOT NULL	Verifica se un valore non lo è NULL.
IS NaN	Verifica se un valore è NaN.
IS NOT NaN	Verifica se un valore non lo è NaN.

Query di esempio

filtraggio dei metadati

L'esempio seguente riguarda il filtraggio dei metadati con un'SELECTistruzione con il linguaggio di interrogazione: AWS IoT SiteWise

```
SELECT a.asset_name, p.property_name
FROM asset a, asset_property p
WHERE a.asset_id = p.asset_id AND a.asset_name LIKE '%windmill%'
```

Filtraggio dei valori

Di seguito è riportato un esempio di filtraggio dei valori mediante un'SELECTistruzione con il linguaggio di AWS IoT SiteWise interrogazione:

```
SELECT a.asset_name FROM asset a, raw_time_series r
WHERE a.asset_id = r.asset_id AND r.int_value > 30 AND r.event_timestamp > 1234567890
AND r.event_timestamp < 1234567891
```

Interagisci con altri AWS servizi

AWS IoT SiteWise può pubblicare i dati degli asset sul broker di messaggi di pubblicazione e sottoscrizione AWS IoT MQTT, in modo da poter interagire con i dati degli asset provenienti da altri servizi. AWS IoT SiteWise assegna a ciascuna proprietà degli asset un argomento MQTT univoco che potete utilizzare per indirizzare i dati degli asset ad altri AWS servizi utilizzando le regole di base. AWS IoT Ad esempio, puoi configurare le regole di AWS IoT base per eseguire le seguenti attività:

- Identificare un guasto nell'apparecchiatura e informare il personale appropriato, inviando i dati ad [AWS IoT Events](#).
- Storizza determinati dati di asset per utilizzarli in soluzioni software esterne inviando dati ad [Amazon DynamoDB](#).
- Generare rapporti settimanali, attivando una funzione [AWS Lambda](#).

Puoi seguire un tutorial che illustra i passaggi necessari per configurare una regola che memorizza i valori delle proprietà in DynamoDB. Per ulteriori informazioni, consulta [Pubblica aggiornamenti dei valori delle proprietà su Amazon DynamoDB](#).

Per ulteriori informazioni su come configurare una regola, consulta [Rules](#) nella AWS IoT Developer Guide.

Puoi anche riutilizzare i dati di altri AWS servizi in AWS IoT SiteWise. Per importare dati tramite l'azione della AWS IoT SiteWise regola, vedere [Inserisci dati per AWS IoT SiteWise utilizzare le regole AWS IoT Core](#).

Argomenti

- [Comprendi le proprietà degli asset negli argomenti MQTT](#)
- [Attiva le notifiche sulle proprietà degli asset in AWS IoT SiteWise](#)
- [Interroga le notifiche relative alle proprietà degli asset in AWS IoT SiteWise](#)
- [Esporta i dati su Amazon S3 con notifiche sulle proprietà degli asset](#)
- [Integrazione AWS IoT SiteWise con Grafana](#)
- [Integrazione AWS IoT SiteWise e AWS IoT TwinMaker](#)
- [Rileva le anomalie con Lookout for Equipment](#)

Comprendi le proprietà degli asset negli argomenti MQTT

A ogni proprietà di asset è assegnato un percorso di argomento MQTT univoco nel formato seguente.

```
$aws/sitewise/asset-models/assetModelId/assets/assetId/properties/propertyId
```

Note

AWS IoT SiteWise non supporta il carattere jolly del filtro degli argomenti # (a più livelli) nel motore di regole AWS IoT Core. È invece possibile utilizzare il carattere jolly + (a livello singolo). Puoi, ad esempio, avvalerti del filtro di argomenti che segue per recuperare tutti gli aggiornamenti relativi a un particolare modello di asset.

```
$aws/sitewise/asset-models/assetModelId/assets/+ /properties/+
```

Per ulteriori informazioni sui caratteri jolly dei filtri per argomenti, consulta [gli argomenti nella Guida AWS IoT](#) principale per gli sviluppatori.

Attiva le notifiche sulle proprietà degli asset in AWS IoT SiteWise

È possibile abilitare le notifiche sulle proprietà su cui pubblicare AWS IoT Core gli aggiornamenti dei dati degli asset e quindi eseguire query sui dati. Con le notifiche sulle proprietà degli asset, AWS IoT SiteWise fornisce un AWS CloudFormation modello che puoi utilizzare per esportare AWS IoT SiteWise i dati in Amazon S3.

Note

I dati degli asset vengono inviati AWS IoT Core ogni volta che vengono ricevuti da AWS IoT SiteWise, indipendentemente dal fatto che il valore sia cambiato.

Argomenti

- [Attiva le notifiche sulle proprietà degli asset \(console\)](#)
- [Attiva le notifiche sulle proprietà degli asset \(\)AWS CLI](#)

Attiva le notifiche sulle proprietà degli asset (console)

Per impostazione predefinita, AWS IoT SiteWise non pubblica gli aggiornamenti dei valori delle proprietà. Puoi utilizzare la AWS IoT SiteWise console per abilitare le notifiche per una proprietà della risorsa.

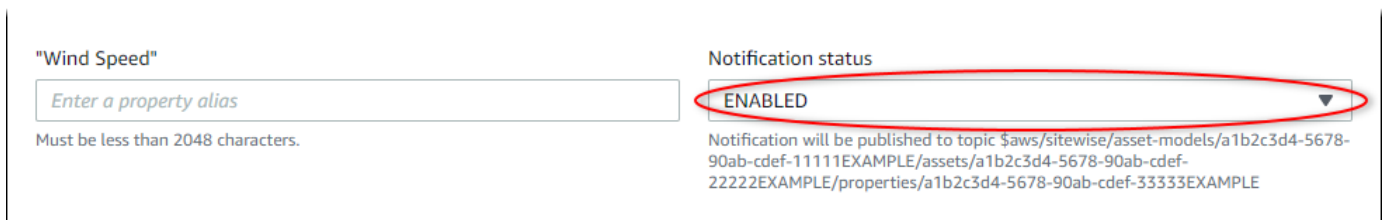
Per abilitare o disabilitare le notifiche per una proprietà di asset (console)

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Asset.
3. Scegli l'asset per abilitare le notifiche di una proprietà.

Tip

Puoi scegliere l'icona a forma di freccia per espandere una gerarchia di asset e trovare il tuo asset.

4. Scegli Modifica.
5. Per lo stato di notifica della proprietà di asset scegli ABILITATO.



The screenshot shows the AWS IoT SiteWise console interface. On the left, there is a text input field labeled '"Wind Speed"' with a placeholder text 'Enter a property alias' and a note 'Must be less than 2048 characters.' To the right of this field is a dropdown menu titled 'Notification status'. The dropdown is currently set to 'ENABLED' and is circled in red. Below the dropdown, there is a detailed notification topic string: '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'.

Puoi inoltre scegliere DISABILITATO per disabilitare le notifiche per la proprietà di asset.

6. Seleziona Salva.

Attiva le notifiche sulle proprietà degli asset (AWS CLI)

Per impostazione predefinita, AWS IoT SiteWise non pubblica gli aggiornamenti dei valori delle proprietà. È possibile utilizzare AWS Command Line Interface (AWS CLI) per abilitare o disabilitare le notifiche per una proprietà di un asset.

Per completare questa procedura, è necessario conoscere l'elemento `assetId` dell'asset e l'elemento `propertyId` della proprietà. Puoi anche usare l'ID esterno. Se hai creato una risorsa e non la conosci `assetId`, utilizza l'[ListAssetsAPI](#) per elencare tutte le risorse per un modello specifico.

Utilizzate l'[DescribeAsset](#) operazione per visualizzare le proprietà della risorsa, inclusa la proprietà IDs.

Utilizzate l'[UpdateAssetProperty](#) operazione per abilitare o disabilitare le notifiche per la proprietà di una risorsa. Specifica i seguenti parametri:

- `assetId`— L'ID della risorsa.
- `propertyId`— L'ID della proprietà dell'asset.
- `propertyNotificationState`— Lo stato di notifica del valore della proprietà: `ENABLED` o `DISABLED`.
- `propertyAlias`— L'alias della proprietà. Specificare l'alias esistente della proprietà quando si aggiorna lo stato di notifica. Se si omette questo parametro, l'alias esistente della proprietà viene rimosso.

Per abilitare o disabilitare le notifiche per una proprietà di asset (CLI)

1. Esegui il comando seguente per recuperare l'alias della proprietà di asset. Sostituisci *asset-id* con l'ID della risorsa e *property-id* con l'ID della proprietà.

```
aws iotsitewise describe-asset-property \  
  --asset-id asset-id \  
  --property-id property-id
```

L'operazione restituisce una risposta contenente i dettagli della proprietà di asset nel formato seguente. L'alias della proprietà si trova in `assetProperty.alias` nell'oggetto JSON.

```
{  
  "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. Esegui il comando seguente per abilitare le notifiche per la proprietà di asset. *property-alias* Sostituiscilo con l'alias della proprietà riportato nella risposta del comando precedente oppure omettete di `--property-alias` aggiornare la proprietà senza un alias.

```

aws iotsitewise update-asset-property \
  --asset-id asset-id \
  --property-id property-id \
  --property-notification-state ENABLED \
  --property-alias property-alias

```

Puoi inoltre passare `--property-notification-state DISABLED` per disabilitare le notifiche per la proprietà di asset.

Interroga le notifiche relative alle proprietà degli asset in AWS IoT SiteWise

Per interrogare le notifiche sulle proprietà degli asset, crea AWS IoT Core regole costituite da istruzioni SQL.

AWS IoT SiteWise pubblica gli aggiornamenti dei dati delle proprietà degli asset su AWS IoT Core nel seguente formato.

```

{
  "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"
        }
    }
}
]
}
}

```

Ogni struttura nell'`values` elenco è una struttura timestamp-quality-value (TQV).

- `timestamp` contiene l'orario UTC (epoca (Unix epoch)) corrente in secondi, con offset in nanosecondi.
- `quality` contiene una delle seguenti stringhe che indicano la qualità del punto dati:
 - GOOD— I dati non sono interessati da alcun problema.
 - BAD— I dati sono interessati da un problema come il guasto del sensore.
 - UNCERTAIN— I dati sono influenzati da un problema come l'imprecisione del sensore.
- `value` contiene uno dei seguenti campi, a seconda del tipo di proprietà:
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`

`nullValue`— Una struttura con il seguente campo che indica il tipo di valore della proprietà con valore Null e qualità di or. BAD UNCERTAIN

- `valueType`— Enumerazione di {"B», «D», «S», «I"}

Per analizzare i valori al di fuori dell'array `values`, è necessario utilizzare query complesse su oggetti nidificati nelle istruzioni SQL delle regole. Per ulteriori informazioni, consulta le [interrogazioni sugli oggetti annidati](#) nella Guida per gli AWS IoT sviluppatori o consulta il [Pubblica aggiornamenti dei valori delle proprietà su Amazon DynamoDB](#) tutorial per un esempio specifico di analisi dei messaggi di notifica delle proprietà degli asset.

Example Query di esempio per estrarre l'array dei valori

Nell'istruzione seguente viene illustrato come eseguire una query sull'array dei valori delle proprietà aggiornati per una proprietà di tipo doppio specifica su tutti gli asset con tale proprietà.

```
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'
```

L'istruzione della query della regola precedente restituisce i dati nel formato seguente.

```
{
  "windspeed": [
    26.32020195042838,
    26.282584572975477,
    26.352566977372508,
    26.283084346171442,
    26.571883739599322,
    26.60684140743005,
    26.628738636715045,
    26.273486932802125,
    26.436379105473964,
    26.600590095377303
  ]
}
```

Example Query di esempio per estrarre un singolo valore

Nell'istruzione seguente viene illustrato come eseguire una query del primo valore dall'array dei valori delle proprietà per una proprietà di tipo doppio specifica su tutti gli asset con tale proprietà.

```
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'
```

L'istruzione della query della regola precedente restituisce i dati nel formato seguente.

```
{
  "windspeed": 26.32020195042838
}
```

Important

Questa istruzione di query regola ignora gli aggiornamenti di valore diversi dal primo in ogni batch. Ogni lotto può contenere fino a 10 valori. Se è necessario includere i valori rimanenti, è necessario impostare una soluzione più complessa per l'output dei valori delle proprietà delle risorse ad altri servizi. Ad esempio, potete impostare una regola con un' AWS Lambda azione per ripubblicare ogni valore dell'array su un altro argomento e impostare un'altra regola per interrogare quell'argomento e pubblicare ogni valore nell'azione della regola desiderata.

Esporta i dati su Amazon S3 con notifiche sulle proprietà degli asset

Puoi esportare i dati in entrata AWS IoT SiteWise da un bucket Amazon S3 nel tuo account. Puoi eseguire il backup dei dati in un formato che puoi utilizzare per creare report storici o per analizzare i dati con metodi complessi.

Per esportare i dati delle serie temporali da AWS IoT SiteWise, abilita la funzionalità cold tier per archiviare i dati in un bucket Amazon S3. Per ulteriori dettagli, consulta [Gestire l'archiviazione dei dati in AWS IoT SiteWise](#).

Per esportare modelli di asset e metadati di asset da AWS IoT SiteWise, utilizza la funzionalità di operazioni in blocco per esportare i metadati in un bucket Amazon S3. Per ulteriori dettagli, consulta [Operazioni in blocco con risorse](#) e modelli.

Integrazione AWS IoT SiteWise con Grafana

Grafana è una piattaforma di visualizzazione dei dati utilizzata per visualizzare e monitorare i dati nei dashboard. Nella versione 10.4.0 e successive di Grafana, utilizza il AWS IoT SiteWise plug-in per visualizzare i AWS IoT SiteWise dati delle risorse nelle dashboard Grafana. Gli utenti possono visualizzare i dati da più AWS fonti (come AWS IoT SiteWise Amazon Timestream CloudWatch e Amazon) e altre fonti di dati con un'unica dashboard Grafana.

Hai due opzioni per utilizzare il plugin: AWS IoT SiteWise

- Server Grafana locali

Puoi configurare il AWS IoT SiteWise plugin su un server Grafana che gestisci. Per ulteriori informazioni su come aggiungere e utilizzare il plug-in, consulta il file [README di AWS IoT SiteWise Datasource](#) sul sito Web. GitHub

- AWS Managed Service for Grafana

Puoi utilizzare il AWS IoT SiteWise plug-in nel AWS Managed Service for Grafana (AMG). AMG gestisce i server Grafana per te in modo che tu possa visualizzare i tuoi dati senza dover creare, impacchettare o implementare alcun hardware o altra infrastruttura Grafana. Per ulteriori informazioni, consulta i seguenti argomenti nella AWS Managed Service for Grafana User Guide:

- [Cos'è Amazon Managed Service for Grafana \(AMG\)?](#)
- [Utilizzo della fonte di AWS IoT SiteWise dati](#)

Example Esempio di dashboard Grafana

[La seguente dashboard Grafana visualizza il parco eolico dimostrativo.](#) Puoi accedere a questa dashboard demo sul sito web di [Grafana Play](#).



Integrazione AWS IoT SiteWise e AWS IoT TwinMaker

L'AWS IoT TwinMaker integrazione con consente l'accesso a funzionalità affidabili AWS IoT SiteWise, come l'ExecuteQueryAPI per il recupero AWS IoT SiteWise dei dati e la ricerca avanzata delle risorse nella console. AWS IoT SiteWise Per integrare i servizi e utilizzare queste funzionalità, devi prima abilitare l'integrazione.

Argomenti

- [Abilitazione dell'integrazione](#)
- [Integrazione e AWS IoT SiteWiseAWS IoT TwinMaker](#)

Abilitazione dell'integrazione

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni. L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Per ulteriori informazioni sulle azioni AWS IoT SiteWise supportate, vedere [Azioni definite da AWS IoT SiteWise](#) nel Service Authorization Reference.

Per ulteriori informazioni sul ruolo AWS IoT TwinMaker collegato al servizio, consulta la sezione [Ruoli collegati ai servizi AWS IoT TwinMaker nella Guida per l'utente](#). AWS IoT TwinMaker

Prima di poter effettuare l'integrazione con AWS IoT SiteWise e AWS IoT TwinMaker, è necessario concedere le seguenti autorizzazioni che consentono l'integrazione con un'area di AWS IoT SiteWise lavoro collegata: AWS IoT TwinMaker

- `iotsitewise:EnableSiteWiseIntegration`— Consente AWS IoT SiteWise l'integrazione con uno spazio di lavoro collegato AWS IoT TwinMaker . Questa integrazione consente di AWS IoT TwinMaker leggere tutte le informazioni di modellazione AWS IoT SiteWise tramite un ruolo collegato al AWS IoT TwinMaker servizio. Per abilitare questa autorizzazione, aggiungi la seguente policy al tuo ruolo IAM:

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

Integrazione e AWS IoT SiteWiseAWS IoT TwinMaker

Per integrare AWS IoT SiteWise e AWS IoT TwinMaker, è necessario disporre di quanto segue:

- AWS IoT SiteWise ruolo collegato al servizio impostato nel tuo account

- AWS IoT TwinMaker ruolo collegato al servizio impostato nel tuo account
- AWS IoT TwinMaker spazio di lavoro con ID `IoTSiteWiseDefaultWorkspace` nel tuo account nella regione.

Da integrare utilizzando la console AWS IoT SiteWise

Quando vedi il AWS IoT TwinMaker banner Integrazione con nella console, scegli Concedi l'autorizzazione. I prerequisiti vengono creati nel tuo account.

Da integrare utilizzando il AWS CLI

Per integrarlo AWS IoT SiteWise e AWS IoT TwinMaker utilizzarlo AWS CLI, inserisci i seguenti comandi:

1. Chiama `CreateServiceLinkedRole` con un `AWSServiceName` `diotsitewise.amazonaws.com`.

```
aws iam create-service-linked-role --aws-service-name iotsitewise.amazonaws.com
```

2. Chiama `CreateServiceLinkedRole` con un `AWSServiceName` di `iottwinmaker.amazonaws.com`.

```
aws iam create-service-linked-role --aws-service-name iottwinmaker.amazonaws.com
```

3. Chiama `CreateWorkspace` con un ID di `IoTSiteWiseDefaultWorkspace`.

```
aws iottwinmaker create-workspace --workspace-id IoTSiteWiseDefaultWorkspace
```

Rileva le anomalie con Lookout for Equipment

Note

Il rilevamento delle anomalie è disponibile solo nelle regioni in cui è disponibile Amazon Lookout for Equipment.

Puoi integrarti AWS IoT SiteWise con Amazon Lookout for Equipment per ottenere informazioni dettagliate sulle tue apparecchiature industriali attraverso il rilevamento delle anomalie e la

manutenzione predittiva delle apparecchiature industriali. Lookout for Equipment è un servizio di machine learning (ML) per il monitoraggio delle apparecchiature industriali che rileva il comportamento anomalo delle apparecchiature e identifica potenziali guasti. Con Lookout for Equipment, è possibile implementare programmi di manutenzione predittiva e identificare i processi delle apparecchiature non ottimali. Per ulteriori informazioni su Lookout for Equipment, [consulta Cos'è Amazon Lookout for Equipment?](#) nella Guida per l'utente di Amazon Lookout for Equipment.

Quando crei una previsione per addestrare un modello ML a rilevare il comportamento anomalo delle apparecchiature, AWS IoT SiteWise invia i valori delle proprietà degli asset a Lookout for Equipment per addestrare un modello ML per rilevare il comportamento anomalo delle apparecchiature. Per definire una definizione di previsione su un modello di asset, specifichi i ruoli IAM necessari a Lookout for Equipment per accedere ai tuoi dati e alle proprietà da inviare a Lookout for Equipment e inviare i dati elaborati ad Amazon S3. Per ulteriori informazioni, consulta [Crea modelli di asset in AWS IoT SiteWise](#).

Per integrare AWS IoT SiteWise e Lookout for Equipment, dovrai eseguire i seguenti passaggi di alto livello:

- Aggiungi una definizione di previsione su un modello di asset che delinei le proprietà che desideri monitorare. La definizione di previsione è una raccolta riutilizzabile di misurazioni, trasformazioni e metriche utilizzata per creare previsioni sugli asset basati su quel modello di asset.
- Addestra la previsione in base ai dati storici che fornisci.
- Pianifica l'inferenza, che indica la AWS IoT SiteWise frequenza con cui eseguire una previsione specifica.

Una volta pianificata l'inferenza, il modello Lookout for Equipment monitora i dati ricevuti dalle apparecchiature e cerca anomalie nel comportamento delle apparecchiature. È possibile visualizzare e analizzare i risultati in SiteWise Monitor, utilizzando le operazioni dell'API AWS IoT SiteWise GET o la console Lookout for Equipment. Puoi anche creare allarmi utilizzando i rilevatori di allarme del modello Asset per avvisarti del comportamento anomalo delle apparecchiature.

Argomenti

- [Aggiungi una definizione di previsione \(console\)](#)
- [Addestra una previsione \(console\)](#)
- [Avvia o interrompi l'inferenza su una previsione \(console\)](#)
- [Aggiungere una definizione di previsione \(CLI\)](#)

- [Addestra una previsione e un'inferenza iniziale \(CLI\)](#)
- [Addestra una previsione \(CLI\)](#)
- [Avvia o interrompi l'inferenza su una previsione \(CLI\)](#)

Aggiungi una definizione di previsione (console)

Per iniziare a inviare i dati raccolti da AWS IoT SiteWise a Lookout for Equipment, è necessario aggiungere AWS IoT SiteWise una definizione di previsione a un modello di asset.

Per aggiungere una definizione di previsione a un modello di asset AWS IoT SiteWise

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegliete Modelli e selezionate il modello di asset a cui desiderate aggiungere la definizione di previsione.
3. Scegliete Previsioni.
4. Scegli Aggiungi definizione di previsione.
5. Definisci i dettagli sulla definizione della previsione.
 - a. Inserisci un nome e una descrizione univoci per la definizione della previsione. Scegli il nome con attenzione perché dopo aver creato la definizione di previsione, non puoi cambiarne il nome.
 - b. Crea o seleziona un ruolo di autorizzazione IAM che AWS IoT SiteWise consenta di condividere i dati delle tue risorse con Amazon Lookout for Equipment. Il ruolo deve avere le seguenti politiche IAM e trust. Per informazioni sulla creazione del ruolo, consulta [Creazione di un ruolo utilizzando politiche di fiducia personalizzate \(console\)](#).

Policy IAM

```
{
  "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:Region:Account_ID:inference-
scheduler/IoTSiteWise_*",
        "arn:aws:lookoutequipment:Region:Account_ID:model/
IoTSiteWise_*",
        "arn:aws:lookoutequipment:Region:Account_ID: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::Account_ID:role/Role_name"
}

```

```

    }
  ]
}

```

Policy di trust

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "Service": "iotsitewise.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "Account_ID"
      },
      "ArnEquals": {
        "aws:SourceArn":
"arn:aws:iotsitewise:Region:Account_ID:asset/*"
      }
    },
  },
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "lookoutequipment.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "Account_ID"
      },
      "ArnEquals": {
        "aws:SourceArn":
"arn:aws:lookoutequipment:Region:Account_ID:*"
      }
    },
  }
]
}

```

- c. Scegli Next (Successivo).
6. Seleziona gli attributi dei dati (misurazioni, trasformazioni e metriche) che desideri inviare a Lookout for Equipment.
 - a. (Facoltativo) Seleziona le misurazioni.
 - b. (Facoltativo) Seleziona le trasformazioni.
 - c. (Facoltativo) Seleziona le metriche.
 - d. Scegli Next (Successivo).
7. Rivedi le tue selezioni. Per aggiungere la definizione di previsione al modello di asset, nella pagina di riepilogo, scegli Aggiungi definizione di previsione.

Puoi anche modificare o eliminare una definizione di previsione esistente a cui sono allegate previsioni attive.

Addestra una previsione (console)

Dopo aver aggiunto una definizione di previsione a un modello di asset, puoi addestrare le previsioni relative ai tuoi asset.

Per addestrare una previsione in AWS IoT SiteWise

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Risorse e seleziona la risorsa che desideri monitorare.
3. Scegliete Previsioni.
4. Seleziona i pronostici che desideri addestrare.
5. In Azioni, scegli Inizia allenamento ed esegui le seguenti operazioni:
 - a. In Dettagli di previsione, seleziona un ruolo di autorizzazioni IAM che AWS IoT SiteWise consenta di condividere i dati degli asset con Lookout for Equipment. Se devi creare un nuovo ruolo, scegli Crea un nuovo ruolo.
 - b. Per le impostazioni dei dati di allenamento, inserisci un intervallo temporale dei dati di allenamento per selezionare quali dati utilizzare per addestrare la previsione.
 - c. (Facoltativo) Seleziona la frequenza di campionamento dei dati dopo la post-elaborazione.

- d. (Facoltativo) Per le etichette dati, fornisci un bucket Amazon S3 e un prefisso che contenga i dati di etichettatura. Per ulteriori informazioni sui dati di etichettatura, consulta la sezione [Etichettatura dei dati](#) nella Amazon Lookout for Equipment User Guide.
 - e. Scegli Next (Successivo).
6. (Facoltativo) Se desideri che la previsione sia attiva non appena ha completato l'allenamento, in Impostazioni avanzate, seleziona Attiva automaticamente la previsione dopo l'allenamento, quindi procedi come segue:
 - a. In Dati di input, per Frequenza di caricamento dei dati, definisci la frequenza di caricamento dei dati e per il tempo di ritardo di Offset, definisci la quantità di buffer da utilizzare.
 - b. Scegli Next (Successivo).
7. Controlla i dettagli del pronostico e scegli Salva e inizia.

Avvia o interrompi l'inferenza su una previsione (console)

Note

I costi di Lookout for Equipment si applicano alle inferenze programmate con i dati trasferiti AWS IoT SiteWise tra e Lookout for Equipment. Per ulteriori informazioni, consulta i prezzi di [Amazon Lookout for Equipment](#).

Se hai aggiunto la previsione `lookoutequipment:CreateDataset` ma non hai scelto di attivarla dopo l'allenamento, devi attivarla per iniziare a monitorare le tue risorse.

Per avviare l'inferenza per una previsione

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Risorse e seleziona la risorsa a cui viene aggiunta la previsione.
3. Scegliete Previsioni.
4. Seleziona le previsioni che desideri attivare.
5. In Azioni, scegli Avvia inferenza ed esegui le seguenti operazioni:
 - a. In Dati di input, per Frequenza di caricamento dei dati, definisci la frequenza di caricamento dei dati e per il tempo di ritardo di Offset, definisci la quantità di buffer da utilizzare.

- b. Scegli Salva e inizia.

Per interrompere l'inferenza per una previsione

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Risorse e seleziona la risorsa a cui viene aggiunta la previsione.
3. Scegliete Previsioni.
4. Seleziona i pronostici che desideri interrompere.
5. In Azioni, scegli Stop inference.

Aggiungere una definizione di previsione (CLI)

Per definire una definizione di previsione su un modello di asset nuovo o esistente, puoi utilizzare (). AWS Command Line Interface AWS CLI Dopo aver definito la definizione di previsione sul modello di asset, addestrate e pianificate l'inferenza per una previsione su un asset per eseguire il rilevamento delle anomalie con AWS IoT SiteWise Lookout for Equipment.

Prerequisiti

Per completare questi passaggi, è necessario disporre di un modello di asset e creare almeno una risorsa. Per ulteriori informazioni, consulta [Crea un modello di asset \(AWS CLI\)](#) e [Crea una risorsa \(AWS CLI\)](#).

Se non lo sapete AWS IoT SiteWise, dovete chiamare l'operazione `CreateBulkImportJob` API per importare i valori delle proprietà dell'asset AWS IoT SiteWise, che verranno utilizzati per addestrare il modello. Per ulteriori informazioni, consulta [Crea un processo di importazione in blocco AWS IoT SiteWise \(\)AWS CLI](#).

Per aggiungere una definizione di previsione

1. Crea un file denominato `asset-model-payload.json`. Segui i passaggi descritti in queste altre sezioni per aggiungere i dettagli del tuo modello di asset al file, ma non inviare la richiesta per creare o aggiornare il modello di asset.
 - Per ulteriori informazioni su come creare un modello di asset, consulta [Crea un modello di asset \(AWS CLI\)](#)

- Per ulteriori informazioni su come aggiornare un modello di asset esistente, consulta [Aggiornate un modello di asset o componente \(AWS CLI\)](#)
2. Aggiungete un modello composito Lookout for Equipment `assetModelCompositeModels` () al modello di asset aggiungendo il codice seguente.
- Sostituirelo *Property* con l'ID delle proprietà che desiderate includere. Per averli IDs, chiama [DescribeAssetModel](#).
 - Sostituiscilo *RoleARN* con l'ARN di un ruolo IAM che consente a Lookout for Equipment di accedere ai tuoi dati. AWS IoT SiteWise

```
{
  ...
  "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. Crea il modello di asset o aggiorna il modello di asset esistente. Esegui una di queste operazioni:

- Per creare il modello di asset, esegui il seguente comando:

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

- Per aggiornare il modello di asset esistente, esegui il comando seguente. Sostituiscilo *asset-model-id* con l'ID del modello di asset che desiderate aggiornare.

```
aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --cli-input-json file://asset-model-payload.json
```

Dopo aver eseguito il comando, `assetModelId` annotatelo nella risposta.

Addestra una previsione e un'inferenza iniziale (CLI)

Ora che la definizione di previsione è stata definita, puoi addestrare gli asset in base ad essa e avviare l'inferenza. Se vuoi addestrare la tua previsione ma non iniziare l'inferenza, passa a [Addestra una previsione \(CLI\)](#). Per addestrare la previsione e iniziare l'inferenza sulla risorsa, avrai bisogno della `assetId` risorsa di destinazione.

Per addestrare e avviare l'inferenza della previsione

1. Esegui il seguente comando per trovare quanto segue `assetModelCompositeModelId`. `assetModelCompositeModelSummaries` *asset-model-id* Sostituiscilo con l'ID del modello di asset in cui avete creato [Aggiornate un modello di asset o componente \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \
  --asset-model-id asset-model-id \
```

2. Eseguite il comando seguente per trovare `actionDefinitionId` l'`TrainingWithInference` azione. Sostituisci *asset-model-id* con l'ID utilizzato nel passaggio precedente e sostituisci *asset-model-composite-model-id* con l'ID restituito nel passaggio precedente.

```
aws iotsitewise describe-asset-model-composite-model \
  --asset-model-id asset-model-id \
  --asset-model-composite-model-id asset-model-composite-model-id \
```

3. Create un file chiamato `train-start-inference-prediction.json` e aggiungete il codice seguente, sostituendo il seguente:

- *asset-id* con l'ID della risorsa di destinazione
- *action-definition-id* con l'ID dell' `TrainingWithInference` azione
- *StartTime* con l'inizio dei dati di allenamento, forniti in secondi d'epoca
- *EndTime* con i dati di fine addestramento, forniti in secondi d'epoca
- *TargetSamplingRate* con la frequenza di campionamento dei dati dopo la post-elaborazione da parte di Lookout for Equipment. I valori consentiti sono: 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. Eseguì il seguente comando per avviare l'addestramento e l'inferenza:

```
aws iotsitewise execute-action --cli-input-json file://train-start-inference-prediction.json
```

Addestra una previsione (CLI)

Ora che la definizione di previsione è stata definita, puoi addestrare gli asset in base ad essa. Per addestrare la previsione sull'asset, avrai bisogno della risorsa `assetId` di destinazione.

Per addestrare la previsione

1. Esegui il seguente comando per trovare quanto segue `assetModelCompositeModelId`.
`assetModelCompositeModelSummaries` *asset-model-id* Sostituilo con l'ID del modello di asset in cui avete creato [Aggiornate un modello di asset o componente \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \  
  --asset-model-id asset-model-id \
```

2. Eseguite il comando seguente per trovare `actionDefinitionId` l'Trainingazione. Sostituisci *asset-model-id* con l'ID utilizzato nel passaggio precedente e sostituisci *asset-model-composite-model-id* con l'ID restituito nel passaggio precedente.

```
aws iotsitewise describe-asset-model-composite-model \  
  --asset-model-id asset-model-id \  
  --asset-model-composite-model-id asset-model-composite-model-id \
```

3. Create un file chiamato `train-prediction.json` e aggiungete il codice seguente, sostituendo il seguente:
 - *asset-id* con l'ID della risorsa di destinazione
 - *action-definition-id* con l'ID dell'azione formativa
 - *StartTime* con i dati di inizio dell'allenamento, forniti in secondi epocali
 - *EndTime* con i dati di fine addestramento, forniti in secondi d'epoca
 - (Facoltativo) *BucketName* con il nome del bucket Amazon S3 che contiene i dati dell'etichetta
 - (Facoltativo) *Prefix* con il prefisso associato al bucket Amazon S3.
 - *TargetSamplingRate* con la frequenza di campionamento dei dati dopo la post-elaborazione da parte di Lookout for Equipment. I valori consentiti sono: PT1S | PT5S | PT10S | PT15S | PT30S | PT1M | PT5M | PT10M | PT15M | PT30M | PT1H

 Note

Includi sia il nome che il prefisso del bucket o nessuno dei due.

```
{
  "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. Esegui il seguente comando per iniziare l'allenamento:

```
aws iotsitewise execute-action --cli-input-json file://train-prediction.json
```

Prima di iniziare l'inferenza, è necessario completare l'addestramento. Per verificare lo stato della formazione, effettuate una delle seguenti operazioni:

- Dalla console, accedi alla risorsa su cui si basa la previsione.
- Da AWS CLI, chiama `BatchGetAssetPropertyValue` utilizzando l'indirizzo `propertyId` della `trainingStatus` proprietà.

Avvia o interrompi l'inferenza su una previsione (CLI)

Una volta addestrata la previsione, puoi avviare l'inferenza per dire a Lookout for Equipment di iniziare a monitorare le tue risorse. Per avviare o interrompere l'inferenza, avrai bisogno della risorsa `assetId` di destinazione.

Per iniziare l'inferenza

1. Esegui il seguente comando per trovare il `assetModelCompositeModelId` sotto `assetModelCompositeModelSummaries`. *asset-model-id* Sostituilo con l'ID del modello di asset in cui avete creato [Aggiornate un modello di asset o componente \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \  
  --asset-model-id asset-model-id \
```

2. Eseguite il comando seguente per trovare `actionDefinitionId` l'Inferenza. Sostituisci *asset-model-id* con l'ID utilizzato nel passaggio precedente e sostituisci *asset-model-composite-model-id* con l'ID restituito nel passaggio precedente.

```
aws iotsitewise describe-asset-model-composite-model \  
  --asset-model-id asset-model-id \  
  --asset-model-composite-model-id asset-model-composite-model-id \
```

3. Create un file chiamato `start-inference.json` e aggiungete il codice seguente, sostituendo il seguente:

- *asset-id* con l'ID della risorsa di destinazione
- *action-definition-id* con l'ID dell'azione di inferenza iniziale
- *Offset* con la quantità di buffer da usare
- *Frequency* con la frequenza con cui vengono caricati i dati

```
{  
  "targetResource": {  
    "assetId": "asset-id"  
  },  
  "actionDefinitionId": "action-definition-id",  
  "actionPayload": { "stringValue": "{ \"l4EInference\": { \"inferenceMode\": \"START\", \"dataDelayOffsetInMinutes\": Offset, \"dataUploadFrequency\": \"Frequency\" } }" }  
}
```

4. Esegui il seguente comando per avviare l'inferenza:

```
aws iotsitewise execute-action --cli-input-json file://start-inference.json
```


Per interrompere l'inferenza

1. Esegui il seguente comando per trovare il `assetModelCompositeModelId` sotto `assetModelCompositeModelSummaries`. *asset-model-id* Sostituilo con l'ID del modello di asset in cui avete creato [Aggiornate un modello di asset o componente \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \  
  --asset-model-id asset-model-id \
```

2. Eseguite il comando seguente per trovare `actionDefinitionId` l'Inferenzaazione. Sostituisci *asset-model-id* con l'ID utilizzato nel passaggio precedente e sostituisci *asset-model-composite-model-id* con l'ID restituito nel passaggio precedente.

```
aws iotsitewise describe-asset-model-composite-model \  
  --asset-model-id asset-model-id \  
  --asset-model-composite-model-id asset-model-composite-model-id \
```

3. Create un file chiamato `stop-inference.json` e aggiungete il codice seguente, sostituendo il seguente:

- *asset-id* con l'ID della risorsa di destinazione
- *action-definition-id* con l'ID dell'azione di inferenza iniziale

```
{  
  "targetResource": {  
    "assetId": "asset-id"  
  },  
  "actionDefinitionId": "action-definition-id",  
  "actionPayload": { "stringValue": "{\"l4EIInference\":{\"inferenceMode\":\"STOP\"}}"}  
}
```

4. Esegui il comando seguente per interrompere l'inferenza:

```
aws iotsitewise execute-action --cli-input-json file://stop-inference.json
```

Gestisci l'archiviazione dei dati in AWS IoT SiteWise

È possibile AWS IoT SiteWise configurare il salvataggio dei dati nei seguenti livelli di storage:

Livello più elevato

L'hot storage tier è uno storage AWS IoT SiteWise gestito in serie temporali. Hot tier è più efficace per i dati a cui si accede di frequente, con bassa write-to-read latenza. I dati archiviati nell'hot tier vengono utilizzati dalle applicazioni industriali che richiedono un accesso rapido ai valori più recenti delle misurazioni delle apparecchiature. Ciò include applicazioni che visualizzano metriche in tempo reale con una dashboard interattiva o applicazioni che monitorano le operazioni e lanciano allarmi per identificare problemi di prestazioni.

Per impostazione predefinita, i dati inseriti vengono archiviati nel livello più AWS IoT SiteWise elevato. È possibile definire un periodo di conservazione per il livello caldo, dopodiché AWS IoT SiteWise spostare i dati nel livello più elevato in uno storage di livello caldo o freddo, in base alla configurazione. Per ottimizzare le prestazioni e l'efficienza in termini di costi, imposta il periodo di conservazione dei livelli più elevati in modo che sia più lungo del tempo necessario per recuperare spesso i dati. Viene utilizzato per metriche, allarmi e scenari di monitoraggio in tempo reale. Se non è impostato un periodo di conservazione, i dati vengono archiviati a tempo indeterminato nel livello più elevato.

Livello caldo

Il warm storage tier è un livello AWS IoT SiteWise gestito efficace per l'archiviazione economica dei dati storici. È ideale per recuperare grandi volumi di dati con caratteristiche di latenza media write-to-read. Utilizza il livello «warm» per archiviare i dati storici necessari per carichi di lavoro di grandi dimensioni. Ad esempio, viene utilizzato per il recupero dei dati per analisi, applicazioni di business intelligence (BI), strumenti di reporting e formazione di modelli di machine learning (ML). Se si abilita il livello di conservazione a freddo, è possibile definire un periodo di conservazione del livello caldo. Al termine del periodo di conservazione, AWS IoT SiteWise elimina i dati dal livello caldo.

Livello freddo

Il livello di cold storage utilizza un bucket Amazon S3 per archiviare dati usati raramente. Con il cold tier abilitato, AWS IoT SiteWise replica le serie temporali, incluse misurazioni, metriche, trasformazioni e aggregazioni e le definizioni dei modelli di asset ogni 6 ore. Cold tier viene utilizzato per archiviare dati che tollerano un'elevata latenza di lettura per report cronologici e backup.

Argomenti

- [Configurare le impostazioni di archiviazione in AWS IoT SiteWise](#)
- [Risolvere i problemi relativi alle impostazioni di archiviazione per AWS IoT SiteWise](#)
- [Percorsi dei file e schemi di dati salvati nella fase fredda](#)

Configurare le impostazioni di archiviazione in AWS IoT SiteWise

È possibile configurare le impostazioni di archiviazione per attivare lo storage warm tier gestito dal servizio e anche per replicare i dati sul piano freddo. Per ulteriori informazioni sul periodo di conservazione per i livelli caldo e caldo, consulta [Impatto sulla conservazione dei dati](#). Durante la configurazione delle impostazioni di archiviazione, procedi come segue:

- Conservazione a livello elevato: imposta un periodo di conservazione per quanto tempo i dati vengono archiviati nel livello più elevato prima che vengano eliminati e spostati nello storage a livello caldo o a livello freddo gestito dal servizio in base alle impostazioni di archiviazione. AWS IoT SiteWise eliminerà tutti i dati del livello più elevato che esistevano prima della fine del periodo di conservazione. Se non imposti un periodo di conservazione, i tuoi dati vengono archiviati a tempo indeterminato nel livello più elevato.
- Conservazione a livello caldo: imposta un periodo di conservazione per quanto tempo i dati vengono archiviati nel livello caldo prima che vengano eliminati dallo AWS IoT SiteWise storage e trasferiti nello storage cold tier gestito dal cliente. AWS IoT SiteWise elimina tutti i dati dal livello caldo che esisteva prima della fine del periodo di conservazione. Se non è impostato un periodo di conservazione, i dati vengono archiviati a tempo indeterminato nel livello caldo.

Note

Per migliorare le prestazioni delle query, imposta un periodo di conservazione di livello elevato con lo storage di livello caldo.

Impatto della conservazione dei dati nello storage di livello caldo e caldo

- Quando si riduce il periodo di conservazione dello storage hot tier, i dati vengono trasferiti in modo permanente dal livello caldo a quello caldo o freddo. Quando si riduce il periodo di conservazione

del livello caldo, i dati vengono spostati sul livello freddo ed eliminati definitivamente dal livello caldo.

- Quando si aumenta il periodo di conservazione dello storage di livello caldo o caldo, la modifica influisce sui dati inviati AWS IoT SiteWise da quel momento in poi. AWS IoT SiteWise non recupera i dati dallo storage a caldo o a freddo per inserirli nel livello più caldo. Ad esempio, se il periodo di conservazione dello storage hot tier è inizialmente impostato su 30 giorni e poi aumentato a 60 giorni, occorrono 30 giorni perché lo storage hot tier contenga 60 giorni di dati.

Argomenti

- [Configura le impostazioni di archiviazione per il livello caldo \(console\)](#)
- [Configura le impostazioni di archiviazione per warm tier \(AWS CLI\)](#)
- [Configura le impostazioni di archiviazione per il livello freddo \(console\)](#)
- [Configura le impostazioni di archiviazione per il livello freddo \(AWS CLI\)](#)

Configura le impostazioni di archiviazione per il livello caldo (console)

La procedura seguente mostra come configurare le impostazioni di archiviazione per replicare i dati sul livello warm della AWS IoT SiteWise console.

Per configurare le impostazioni di archiviazione nella console

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, in Impostazioni, scegli Archiviazione.
3. Nell'angolo in alto a destra, scegliere Edit (Modifica).
4. Nella pagina Modifica archiviazione, procedi come segue:
5. Per le impostazioni Hot tier, procedi come segue:
 - Se desideri impostare un periodo di conservazione per il periodo di archiviazione dei dati nel livello più elevato prima di essere eliminati e spostati nello storage di livello caldo gestito dal servizio, scegli Abilita periodo di conservazione.
 - Per configurare un periodo di conservazione, inserisci un numero intero e scegli un'unità. Il periodo di conservazione deve essere maggiore o uguale a 30 giorni.

AWS IoT SiteWise elimina tutti i dati del livello più importante che sono più vecchi del periodo di conservazione. Se non imposti un periodo di conservazione, i dati vengono archiviati a tempo indeterminato.

6. (Consigliato) Per le impostazioni del livello Warm, procedi come segue:

- Per attivare l'archiviazione di livello caldo, seleziona Confermo e attiva l'opzione di archiviazione di livello caldo per attivare l'archiviazione di livello caldo.
- (Facoltativo) Per configurare un periodo di conservazione, inserisci un numero intero e scegli un'unità. Il periodo di conservazione deve essere maggiore o uguale a 365 giorni.

AWS IoT SiteWise elimina i dati del livello «warm» che esistevano prima del periodo di conservazione. Se non imposti un periodo di conservazione, i dati vengono archiviati a tempo indeterminato.

 Note

- Quando si opta per il livello caldo, la configurazione viene visualizzata una sola volta.
- Per impostare la conservazione a caldo, è necessario disporre di un livello di archiviazione a caldo o a freddo. Per l'efficienza in termini di costi e il recupero dei dati storici, AWS IoT SiteWise consiglia di archiviare i dati a lungo termine nel livello caldo.
- Per impostare la conservazione del livello caldo, è necessario disporre di un livello di conservazione a freddo.

7. Scegli Salva per salvare le impostazioni di archiviazione.

Nella sezione AWS IoT SiteWise Archiviazione, lo storage Warm tier si trova in uno di questi stati:

- Abilitato: se i dati esistevano prima del periodo di conservazione del livello più elevato, li AWS IoT SiteWise sposta nel livello caldo».
- Disabilitato: lo storage warm tier è disabilitato.

Configura le impostazioni di archiviazione per warm tier (AWS CLI)

È possibile configurare le impostazioni di archiviazione per spostare i dati al livello caldo utilizzando AWS CLI i comandi seguenti.

Per evitare di sovrascrivere la configurazione esistente, recupera le informazioni sulla configurazione di archiviazione corrente eseguendo il comando seguente:

```
aws iotsitewise describe-storage-configuration
```

Example risposta senza una configurazione di livello freddo esistente

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "disassociatedDataStorage": "ENABLED",
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2021-10-14T15:53:35-07:00",
  "warmTier": "DISABLED"
}
```

Example risposta con la configurazione Cold Tier esistente

```
{
  "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"
}
```

Configura le impostazioni di archiviazione per il piano caldo con AWS CLI

Esegui il comando seguente per configurare le impostazioni di archiviazione. Sostituisci `file-name` con il nome del file che contiene la configurazione AWS IoT SiteWise di archiviazione.

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

Example AWS IoT SiteWise configurazione con livello caldo e caldo

```
{
    "storageType": "SITEWISE_DEFAULT_STORAGE",
    "disassociatedDataStorage": "ENABLED",
    "warmTier": "ENABLED",
    "retentionPeriod": {
        "numberOfDays": hot-tier-retention-in-days
    }
}
```

`hot-tier-retention-in-days` deve essere un numero intero maggiore o uguale a 30 giorni.

Example response

```
{
    "storageType": "SITEWISE_DEFAULT_STORAGE",
    "configurationStatus": {
        "state": "UPDATE_IN_PROGRESS"
    }
}
```

Se hai abilitato lo storage a livello freddo, vedi [Configura le impostazioni di archiviazione con AWS CLI un piano freddo esistente](#).

Configura le impostazioni di archiviazione con AWS CLI un piano freddo esistente

Configura le impostazioni di archiviazione utilizzando AWS CLI lo storage a livello freddo esistente

- Esegui il comando seguente per configurare le impostazioni di archiviazione. Sostituisci *file-name* con il nome del file che contiene la configurazione AWS IoT SiteWise di archiviazione.

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

Example AWS IoT SiteWise configurazione di archiviazione

- *amzn-s3-demo-bucket* Sostituiscilo con il nome del tuo bucket Amazon S3.
- *prefix* Sostituiscilo con il tuo prefisso Amazon S3.
- Sostituiscilo *aws-account-id* con l'ID del tuo AWS account.
- Sostituisci *role-name* con il nome del ruolo di accesso di Amazon S3 che consente di inviare dati AWS IoT SiteWise ad Amazon S3.
- Sostituisci *hot-tier-retention-in-days* con un numero intero maggiore o uguale a 30 giorni.
- Sostituire *warm-tier-retention-in-days* con un numero intero maggiore o uguale a 365 giorni.

Note

AWS IoT SiteWise eliminerà tutti i dati del livello caldo più vecchi del periodo di conservazione del livello freddo. Se non imposti un periodo di conservazione, i dati vengono archiviati a tempo indeterminato.

```
{
  "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"
  }
}
```

Configura le impostazioni di archiviazione per il livello freddo (console)

La procedura seguente mostra come configurare le impostazioni di archiviazione per replicare i dati sul livello freddo della AWS IoT SiteWise console.

Per configurare le impostazioni di archiviazione nella console

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, in Impostazioni, scegli Archiviazione.
3. Nell'angolo in alto a destra, scegliere Edit (Modifica).
4. Nella pagina Modifica archiviazione, procedi come segue:
 - a. Per le impostazioni di archiviazione, scegli Abilita archiviazione a freddo. La memorizzazione a livello freddo è disabilitata per impostazione predefinita.
 - b. Per la posizione del bucket S3, inserisci il nome di un bucket Amazon S3 esistente e un prefisso.

Note

- Amazon S3 utilizza il prefisso come nome di cartella nel bucket Amazon S3. Il prefisso deve contenere da 1 a 255 caratteri e terminare con una barra (/). AWS IoT SiteWise I dati vengono salvati in questa cartella.
- Se non disponi di un bucket Amazon S3, scegli Visualizza, quindi creane uno nella console Amazon S3. Per ulteriori informazioni, consulta [Crea il tuo primo bucket S3](#) nella Amazon S3 User Guide.

- c. Per il ruolo di accesso S3, esegui una delle seguenti operazioni:

- Scegli Crea un ruolo da un modello AWS gestito, crea AWS automaticamente un ruolo IAM che consente di AWS IoT SiteWise inviare dati ad Amazon S3.
- Scegli Usa un ruolo esistente, quindi scegli il ruolo che hai creato dall'elenco.

Note

- È necessario utilizzare lo stesso nome di bucket Amazon S3 per la posizione del bucket S3 utilizzato nel passaggio precedente e nella policy IAM.
- Assicurati che il tuo ruolo disponga delle autorizzazioni mostrate nell'esempio seguente.

Example politica delle autorizzazioni:

```
{
  "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/*"
      ]
    }
  ]
}
```

Sostituisci amzn-s3-demo-bucket con il nome del tuo bucket Amazon S3.

- Se il bucket Amazon S3 è crittografato utilizzando una chiave KMS gestita dal cliente, la chiave KMS deve avere una politica di accesso con un ruolo e operazioni IAM. kms:Decrypt kms:GenerateDataKey

- d. Per configurare l'hot tier, consulta la Fase 5 di [Configura le impostazioni di archiviazione per il livello caldo \(console\)](#)
- e. (Facoltativo) Per AWS IoT Analytics l'integrazione, procedi come segue.

 Note

Avviso di fine del supporto: il 15 dicembre 2025, AWS terminerà il supporto per AWS IoT Analytics. Dopo il 15 dicembre 2025, non potrai più accedere alla AWS IoT Analytics console o AWS IoT Analytics alle risorse. Per ulteriori informazioni, consulta [AWS IoT Analytics Fine del supporto](#).

- i. Se desideri utilizzarli AWS IoT Analytics per interrogare i tuoi dati, scegli Archivio AWS IoT Analytics dati abilitato.
- ii. AWS IoT SiteWise genera un nome per il tuo data store oppure puoi inserire un nome diverso.

AWS IoT SiteWise crea automaticamente un data store in AWS IoT Analytics cui salvare i dati. Per interrogare i dati, puoi usarli AWS IoT Analytics per creare set di dati. Per ulteriori informazioni, consulta [Lavorare con AWS IoT SiteWise i dati](#) nella Guida per l'AWS IoT Analytics utente.

- f. Scegli Save (Salva).

Nella sezione AWS IoT SiteWise Archiviazione, lo storage Cold tier può avere uno dei seguenti valori:

- Abilitato: AWS IoT SiteWise replica i dati nel bucket Amazon S3 specificato.
- Abilitazione: AWS IoT SiteWise sta elaborando la richiesta per abilitare lo storage a freddo. Il completamento di questo processo può richiedere diversi minuti.
- Enable_Failed: AWS IoT SiteWise impossibile elaborare la tua richiesta di abilitare lo storage a freddo. Se hai abilitato AWS IoT SiteWise l'invio di log ad Amazon CloudWatch Logs, puoi utilizzare questi log per risolvere i problemi. Per ulteriori informazioni, consulta [Monitoraggio con Amazon CloudWatch Logs](#).
- Disabilitato: lo storage a livello freddo è disabilitato.

Configura le impostazioni di archiviazione per il livello freddo (AWS CLI)

La procedura seguente mostra come configurare le impostazioni di archiviazione per replicare i dati nel livello freddo utilizzando AWS CLI.

Per configurare le impostazioni di archiviazione utilizzando AWS CLI

1. Per esportare i dati in un bucket Amazon S3 nel tuo account, esegui il seguente comando per configurare le impostazioni di archiviazione. Sostituisci *file-name* con il nome del file che contiene la configurazione AWS IoT SiteWise di storage.

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

Example AWS IoT SiteWise configurazione di archiviazione

- *amzn-s3-demo-bucket* Sostituiscilo con il nome del tuo bucket Amazon S3.
- *prefix* Sostituiscilo con il tuo prefisso Amazon S3.
- Sostituiscilo *aws-account-id* con l'ID del tuo AWS account.
- Sostituisci *role-name* con il nome del ruolo di accesso di Amazon S3 che consente di inviare dati AWS IoT SiteWise ad Amazon S3.
- Sostituisci *retention-in-days* con un numero intero maggiore o uguale a 30 giorni.

```
{
  "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

- È necessario utilizzare lo stesso nome di bucket Amazon S3 nella configurazione AWS IoT SiteWise dello storage e nella policy IAM.
- Assicurati che il tuo ruolo disponga delle autorizzazioni mostrate nell'esempio seguente.

Example politica delle autorizzazioni:

```
{
  "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/*"
      ]
    }
  ]
}
```

Sostituisci amzn-s3-demo-bucket con il nome del tuo bucket Amazon S3.

- Se il bucket Amazon S3 è crittografato utilizzando una chiave KMS gestita dal cliente, la chiave KMS deve avere una politica di accesso con un ruolo e operazioni IAM.
kms:Decrypt kms:GenerateDataKey

Example response

```
{
  "storageType": "MULTI_LAYER_STORAGE",
```

```
"retentionPeriod": {
  "numberOfDays": 100,
  "unlimited": false
},
"configurationStatus": {
  "state": "UPDATE_IN_PROGRESS"
}
}
```

Note

L'aggiornamento della configurazione di storage può richiedere alcuni minuti. AWS IoT SiteWise

2. Per recuperare le informazioni sulla configurazione dello storage, esegui il comando seguente.

```
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. Per interrompere l'esportazione dei dati nel bucket Amazon S3, esegui il seguente comando per configurare le impostazioni di archiviazione.

```
aws iotsitewise put-storage-configuration --storage-type SITEWISE_DEFAULT_STORAGE
```

Note

Per impostazione predefinita, i tuoi dati vengono archiviati solo nel livello più elevato di AWS IoT SiteWise

Example response

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "configurationStatus": {
    "state": "UPDATE_IN_PROGRESS"
  }
}
```

4. Per recuperare le informazioni sulla configurazione dello storage, esegui il comando seguente.

```
aws iotsitewise describe-storage-configuration
```

Example response

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2021-03-30T15:57:14-07:00"
}
```

(Facoltativo) Creare un archivio AWS IoT Analytics dati (AWS CLI)

Note

Avviso di fine del supporto: il 15 dicembre 2025, AWS terminerà il supporto per AWS IoT Analytics. Dopo il 15 dicembre 2025, non potrai più accedere alla AWS IoT Analytics console

o AWS IoT Analytics alle risorse. Per ulteriori informazioni, consulta [AWS IoT Analytics Fine del supporto](#).

Un AWS IoT Analytics data store è un repository scalabile e interrogabile che riceve e archivia dati. È possibile utilizzare la AWS IoT SiteWise console o AWS IoT Analytics APIs creare un archivio dati per salvare AWS IoT Analytics i dati. AWS IoT SiteWise Per interrogare i dati, si creano set di dati utilizzando AWS IoT Analytics. Per ulteriori informazioni, consulta [Lavorare con AWS IoT SiteWise i dati](#) nella Guida per l'AWS IoT Analytics utente.

I passaggi seguenti consentono AWS CLI di creare un archivio dati in AWS IoT Analytics.

Per creare un archivio dati, esegui il comando seguente. Sostituisci *file-name* con il nome del file che contiene la configurazione del data store.

```
aws iotanalytics create-datastore --cli-input-json file://file-name.json
```

Note

- È necessario specificare il nome di un bucket Amazon S3 esistente. Se non disponi di un bucket Amazon S3, creane prima uno. Per ulteriori informazioni, consulta [Crea il tuo primo bucket S3](#) nella Amazon S3 User Guide.
- È necessario utilizzare lo stesso nome di bucket Amazon S3 nella configurazione AWS IoT SiteWise dello storage, nella policy IAM e nella configurazione del AWS IoT Analytics data store.

Example AWS IoT Analytics configurazione del data store

Sostituisci *data-store-name* e *amzn-s3-demo-bucket* con il nome del tuo AWS IoT Analytics data store e il nome del bucket Amazon S3.

```
{
  "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  
  }  
}
```

Risolvere i problemi relativi alle impostazioni di archiviazione per AWS IoT SiteWise

Utilizza le seguenti informazioni per individuare e risolvere i problemi relativi alla configurazione dello storage.

Problemi

- [Errore: il bucket non esiste](#)
- [Errore: accesso negato al percorso Amazon S3](#)
- [Errore: non è possibile assumere il ruolo ARN](#)
- [Errore: accesso al bucket Amazon S3 interregionale non riuscito](#)

Errore: il bucket non esiste

Soluzione: AWS IoT SiteWise non è stato possibile trovare il bucket Amazon S3. Assicurati di inserire il nome di un bucket Amazon S3 esistente nella regione corrente.

Errore: accesso negato al percorso Amazon S3

Soluzione: AWS IoT SiteWise impossibile accedere al tuo bucket Amazon S3. Esegui questa operazione:

- Assicurati di utilizzare lo stesso bucket Amazon S3 specificato nella policy IAM.
- Assicurati che il tuo ruolo disponga delle autorizzazioni mostrate nell'esempio seguente.

Example policy di autorizzazioni

```
{
  "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/*"
      ]
    }
  ]
}
```

Sostituisci amzn-s3-demo-bucket con il nome del tuo bucket Amazon S3.

Errore: non è possibile assumere il ruolo ARN

Soluzione: non AWS IoT SiteWise potresti assumere il ruolo IAM per tuo conto. Assicurati che il tuo ruolo si affidi al seguente servizio: `iotsitewise.amazonaws.com`. Per ulteriori informazioni, consulta [Non posso assumere un ruolo](#), consulta la Guida per l'utente IAM.

Errore: accesso al bucket Amazon S3 interregionale non riuscito

Soluzione: il bucket Amazon S3 che hai specificato si trova in una regione diversa. AWS Assicurati che il bucket e gli AWS IoT SiteWise asset Amazon S3 si trovino nella stessa regione.

Percorsi dei file e schemi di dati salvati nella fase fredda

AWS IoT SiteWise archivia i dati nella fase fredda replicando serie temporali, tra cui misurazioni, metriche, trasformazioni e aggregazioni, nonché definizioni di asset e modelli di asset. Di seguito vengono descritti i percorsi dei file e gli schemi dei dati inviati al livello freddo.

Argomenti

- [Dati dell'attrezzatura \(misurazioni\)](#)
- [Metriche, trasformazioni e aggregazioni](#)
- [Metadati delle risorse](#)
- [metadati della gerarchia degli asset](#)
- [File di indice dei dati di archiviazione](#)

Dati dell'attrezzatura (misurazioni)

AWS IoT SiteWise esporta i dati dell'apparecchiatura (misurazioni) nella zona fredda una volta ogni sei ore. I dati grezzi vengono salvati nel livello freddo nel formato [Apache AVRO](#) (.avro).

Percorso del file

AWS IoT SiteWise memorizza i dati dell'apparecchiatura (misurazioni) nel livello freddo utilizzando il seguente modello.

```
{keyPrefix}/raw/startYear={startYear}/startMonth={startMonth}/startDay={startDay}/  
seriesBucket={seriesBucket}/raw_{timeseriesId}_{startTimestamp}_{quality}.avro
```

Ogni percorso di file verso i dati grezzi in Amazon S3 contiene i seguenti componenti.

Componente del percorso	Descrizione
keyPrefix	Il prefisso Amazon S3 che hai specificato nella configurazione dello AWS IoT SiteWise

Componente del percorso	Descrizione
	storage. Amazon S3 utilizza il prefisso come nome di cartella nel bucket.
<code>raw</code>	La cartella che memorizza i dati delle serie temporali provenienti dall'apparecchiatura (misurazioni). La <code>raw</code> cartella viene salvata nella cartella dei prefissi.
<code>seriesBucket</code>	Un numero esadecimale compreso tra 00 e ff. Questo numero è derivato da <code>timeSeriesId</code>. Questa partizione viene utilizzata per aumentare la velocità effettiva durante le operazioni di AWS IoT SiteWise scrittura sul livello freddo. Quando usi Amazon Athena per eseguire query, puoi utilizzare la partizione e per il partizionamento fine per migliorare le prestazioni delle query. <code>seriesBucket</code> e <code>timeSeriesBucket</code> nei metadati degli asset c'è lo stesso numero.
<code>startYear</code>	L'anno dell'ora di inizio esclusiva associata ai dati delle serie temporali.
<code>startMonth</code>	Il mese dell'ora di inizio esclusiva associata ai dati delle serie temporali.
<code>startDay</code>	Il giorno del mese dell'ora di inizio esclusiva associata ai dati delle serie temporali.

Componente del percorso	Descrizione
<code>fileName</code>	Il nome del file utilizza il carattere di sottolineatura (<code>_</code>) come delimitatore per separare quanto segue: • Il prefisso. <code>raw</code> • Il <code>timeSeriesId</code> valore. • Il timestamp dell'epoca dell'ora di inizio esclusiva associata ai dati delle serie temporali. • La qualità dei dati. Valori validi: <code>GOODBAD</code>, <code>eUNCERTAIN</code> . Per ulteriori informazioni, consulta AssetPropertyValue nella documentazione di riferimento dell'API AWS IoT SiteWise . Il file viene salvato nel <code>.avro</code> formato utilizzando la compressione Snappy.

Example percorso del file verso i dati grezzi nella fase fredda

```
keyPrefix/raw/startYear=2021/startMonth=1/startDay=2/seriesBucket=a2/
raw_7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_95e63da7-d34e-43e1-
bc6f-1b490154b07a_1609577700_GOOD.avro
```

Campi

Lo schema dei dati grezzi esportati nel livello freddo contiene i seguenti campi.

AWS IoT SiteWise consiglia ai clienti di implementare il supporto per l'evoluzione dello schema sui sistemi che leggono i dati grezzi dal livello freddo, poiché in futuro potrebbero essere introdotti campi aggiuntivi.

I dati nulli sono rappresentati come se tutti i campi di valore fossero nulli. Tuttavia, i clienti continueranno a ricevere il tipo di dati corretto quando effettuano query con AWS IoT SiteWise APIs

Nome del campo	Tipi supportati	Tipo di predefinito	Descrizione
<code>seriesId</code>	<code>string</code>	N/D	L'ID che identifica i dati delle serie temporali provenienti dall'apparecchiatura (misurazioni). È possibile utilizzare questo campo per unire dati grezzi e metadati delle risorse nelle query.
<code>timeInSeconds</code>	<code>long</code>	N/D	La data e l'ora, in secondi, nel formato Unix epoch. I dati frazionari in nanosecondi sono forniti da <code>offsetInNanos</code>
<code>offsetInNanos</code>	<code>long</code>	N/D	L'offset in nanosecondi da <code>timeInSeconds</code>
<code>quality</code>	<code>string</code>	N/D	La qualità del valore delle serie temporali.
<code>doubleValue</code>	<code>double</code> o <code>null</code>	<code>null</code>	Dati di serie temporali di tipo <code>double</code> (numero in virgola mobile).
<code>stringValue</code>	<code>string</code> o <code>null</code>	<code>null</code>	Dati di serie temporali di tipo <code>string</code> (sequenza di caratteri).

Nome del campo	Tipi supportati	Tipo di predefinito	Descrizione
<code>integerValue</code>	<code>int</code> o <code>null</code>	<code>null</code>	Dati di serie temporali di tipo intero (numero intero).
<code>booleanValue</code>	<code>boolean</code> o <code>null</code>	<code>null</code>	Dati di serie temporali di tipo booleano (vero o falso).
<code>jsonValue</code>	<code>string</code> o <code>null</code>	<code>null</code>	Dati di serie temporali di tipo JSON (tipi di dati complessi memorizzati come stringa).
<code>recordVersion</code>	<code>long</code> o <code>null</code>	<code>null</code>	Il numero di versione del record. È possibile utilizzare il numero di versione per selezionare il record più recente. I record più recenti hanno numeri di versione più grandi.

Example dati grezzi nella fase fredda

```
{
  "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": {
    "double": 0.92,
    "stringValue": null,
    "integerValue": null,
    "booleanValue": null,
    "jsonValue": null,
    "re": null
  }
}
```

Metriche, trasformazioni e aggregazioni

AWS IoT SiteWise esporta metriche, trasforma e aggrega nel livello freddo una volta ogni sei ore.

[Le metriche, le trasformazioni e gli aggregati vengono salvati nel livello freddo nel formato Apache AVRO \(\).](#) .avro

Percorso del file

AWS IoT SiteWise archivia metriche, trasformazioni e aggregazioni nel livello freddo utilizzando il seguente modello.

```
{keyPrefix}/agg/startYear={startYear}/startMonth={startMonth}/startDay={startDay}/
seriesBucket={seriesBucket}/agg_{timeseriesId}_{startTimestamp}_{quality}.avro
```

Ogni percorso di file verso metriche, trasformazioni e aggregazioni in Amazon S3 contiene i seguenti componenti.

Componente del percorso	Descrizione
keyPrefix	Il prefisso Amazon S3 che hai specificato nella configurazione dello AWS IoT SiteWise storage. Amazon S3 utilizza il prefisso come nome di cartella nel bucket.
agg	La cartella che memorizza i dati delle serie temporali ricavati dalle metriche. La agg cartella viene salvata nella cartella dei prefissi.
seriesBucket	Un numero esadecimale compreso tra 00 e ff. Questo numero è derivato da. timeSeriesId Questa partizione viene utilizzata per aumentare la velocità effettiva durante le

Componente del percorso	Descrizione
	operazioni di AWS IoT SiteWise scrittura sul livello freddo. Quando usi Amazon Athena per eseguire query, puoi utilizzare la partizion e per il partizionamento fine per migliorare le prestazioni delle query. <code>seriesBucket</code> e <code>timeSeriesBucket</code> nei metadati degli asset c'è lo stesso numero.
<code>startYear</code>	L'anno dell'ora di inizio esclusiva associata ai dati delle serie temporali.
<code>startMonth</code>	Il mese dell'ora di inizio esclusiva associata ai dati delle serie temporali.
<code>startDay</code>	Il giorno del mese dell'ora di inizio esclusiva associata ai dati delle serie temporali.
<code>fileName</code>	Il nome del file utilizza il carattere di sottolineatura (<code>_</code>) come delimitatore per separare quanto segue: • Il prefisso. <code>raw</code> • Il <code>timeSeriesId</code> valore. • Il timestamp dell'epoca dell'ora di inizio esclusiva associata ai dati delle serie temporali. • La qualità dei dati. Valori validi: <code>GOODBAD</code>, <code>eUNCERTAIN</code> . Per ulteriori informazioni, consulta AssetPropertyValue nella documentazione di riferimento dell'API AWS IoT SiteWise . Il file viene salvato nel <code>.avro</code> formato utilizzando la compressione Snappy.

Example percorso del file verso le metriche nella fase fredda

```
keyPrefix/agg/startYear=2021/startMonth=1/startDay=2/seriesBucket=a2/  
agg_7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_95e63da7-d34e-43e1-  
bc6f-1b490154b07a_1609577700_G00D.avro
```

Campi

Lo schema di metriche, trasformazioni e aggregati esportati nel livello freddo contiene i seguenti campi.

Nome del campo	Tipi supportati	Tipo di predefinito	Descrizione
seriesId	string	N/D	L'ID che identifica i dati delle serie temporali provenienti da apparecchiature, metriche o trasformazioni. È possibile utilizzare questo campo per unire dati non elaborati e metadati delle risorse nelle query.
timeInSeconds	long	N/D	La data e l'ora, in secondi, nel formato Unix epoch. I dati frazionari in nanosecondi sono forniti da. offsetInNanos
offsetInNanos	long	N/D	L'offset in nanosecondi da. timeInSeconds

Nome del campo	Tipi supportati	Tipo di predefinito	Descrizione
quality	string	N/D	La qualità con cui filtrare i dati degli asset.
resolution	string	N/D	L'intervallo di tempo in cui aggregare i dati.
count	double o null	null	Il numero totale di punti dati per le variabili specificate nell'intervallo di tempo corrente.
average	double o null	null	La media dei valori delle variabili specificate nell'intervallo di tempo corrente.
min	double o null	null	Il minimo dei valori delle variabili specificate nell'intervallo di tempo corrente.
max	boolean o null	null	Il massimo dei valori delle variabili specificate nell'intervallo di tempo corrente.
sum	string o null	null	La somma dei valori delle variabili specificate nell'intervallo di tempo corrente.

Nome del campo	Tipi supportati	Tipo di predefinito	Descrizione
recordVersion	long o null	null	Il numero di versione del record. È possibile utilizzare il numero di versione per selezionare il record più recente. I record più recenti hanno numeri di versione più grandi.

Example Dati metrici nella fase fredda

```
{
  "seriesId": "f74c2828-5317-4df3-ba16-6d41b5bcb531",
  "timeInSeconds": 1637334060,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "resolution": "1s",
  "value": {
    "double": 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": "1s",
  "value": {
    "double": 46.0,
    "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": "1s",
  "value": {
    "double": 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": "1s",
  "value": {
    "double": 46.0,
    "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": "1s",
  "value": {
    "double": 16.0,
    "min": {
      "double": 1.0
    },
    "max": {
      "double": 31.0
    },
    "sum": {
      "double": 496.0
    },
    "recordVersion": null
  }
}
```

Metadati delle risorse

Quando abiliti AWS IoT SiteWise l'esportazione dei dati nel livello freddo per la prima volta, i metadati degli asset vengono esportati nel livello freddo. Dopo la configurazione iniziale, AWS IoT SiteWise

esporta i metadati degli asset nel livello solo quando modificate le definizioni dei modelli di asset o le definizioni degli asset. I metadati delle risorse vengono salvati nel livello freddo nel formato JSON () delimitato da nuova riga. .ndjson

Percorso del file

AWS IoT SiteWise archivia i metadati delle risorse nel livello freddo utilizzando il seguente modello.

```
{keyPrefix}/asset_metadata/asset_{assetId}.ndjson
```

Ogni percorso di file verso i metadati delle risorse nel livello freddo contiene i seguenti componenti.

Componente del percorso	Descrizione
keyPrefix	Il prefisso Amazon S3 che hai specificato nella configurazione di storage AWS IoT SiteWise s. Amazon S3 utilizza il prefisso come nome di cartella nel bucket.
asset_metadata	La cartella in cui sono archiviati i metadati delle risorse. La asset_metadata cartella viene salvata nella cartella dei prefissi.
fileName	Il nome del file utilizza il carattere di sottolineatura (_) come delimitatore per separare quanto segue: • Il prefisso. asset • Il assetId valore. Il file viene salvato nel .ndjson formato.

Example percorso del file ai metadati delle risorse nel livello più freddo

```
keyPrefix/asset_metadata/asset_35901915-d476-4dca-8637-d9ed4df939ed.ndjson
```

Campi

Lo schema dei metadati delle risorse che viene esportato nel livello freddo contiene i seguenti campi.

Nome campo	Descrizione
assetId	L'ID dell'asset .
assetName	Il nome della risorsa.
assetExternalId	L'ID esterno della risorsa.
assetModelId	L'ID del modello di asset utilizzato per creare questa risorsa.
assetModelName	Il nome del modello di asset.
assetModelExternalId	L'ID esterno del modello di asset.
assetPropertyId	L'ID della proprietà dell'asset.
assetPropertyName	Il nome della proprietà dell'asset.
assetPropertyExternalId	L'ID esterno della proprietà dell'asset.
assetPropertyDataType	Il tipo di dati della proprietà dell'asset.
assetPropertyUnit	L'unità della proprietà dell'asset (ad esempio, Newtons eRPM).
assetPropertyAlias	L'alias che identifica la proprietà dell'asset, ad esempio il percorso del flusso di dati del server OPC UA (ad esempio,). /company/windfarm/3/turbine/7/temperature
timeSeriesId	L'ID che identifica i dati delle serie temporali provenienti da apparecchiature, metriche o trasformazioni. È possibile utilizzare questo campo per unire dati non elaborati e metadati delle risorse nelle query.
timeSeriesBucket	Un numero esadecimale compreso tra 00 e ff. Questo numero è derivato da. timeSerie

Nome campo	Descrizione
	<code>sId</code> Questa partizione viene utilizzata per aumentare la velocità effettiva durante le operazioni di AWS IoT SiteWise scrittura sul livello freddo. Quando usi Amazon Athena per eseguire query, puoi utilizzare la partizion e per il partizionamento fine per migliorare le prestazioni delle query. <code>timeSeriesBucket</code> e <code>seriesBucket</code> nel percorso del file i dati grezzi sono presenti gli stessi numeri.
<code>assetCompositeModelId</code>	L'ID del modello composito.
<code>assetCompositeModelExternalId</code>	L'ID esterno del modello composito.
<code>assetCompositeModelDescription</code>	La descrizione del modello composito.
<code>assetCompositeModelName</code>	Il nome del modello composito.
<code>assetCompositeModelType</code>	Il tipo del modello composito. Per i modelli compositi di allarme, questo tipo è <code>AWS/ALARM</code> .
<code>assetCreationDate</code>	La data di creazione della risorsa, nell'epoca di Unix.
<code>assetLastUpdateDate</code>	La data dell'ultimo aggiornamento della risorsa, espressa in Unix Epoch Time.
<code>assetStatusErrorCode</code>	Il codice di errore.
<code>assetStatusErrorMessage</code>	Messaggio di errore.
<code>assetStatusState</code>	Lo stato attuale della risorsa.

Example metadati degli asset nella fase fredda

```
{ "assetId": "7020c8e2-e6db-40fa-9845-ed0ddd4c77d", "assetExternalId": null, "assetName": "Wind Turbine Asset Model", "assetModelId": "ec1d924f-f07d-444f-b072-e2994c165d35", "assetModelExternalId": null, "assetModelName": "Wind Turbine Asset Model", "assetPropertyId": "95e63da7-d34e-43e1-bc6f-1b490154b07a", "assetPropertyExternalId": null, "assetPropertyName": "Temperature", "assetPropertyData": { "timeSeriesId": "7020c8e2-e6db-40fa-9845-ed0ddd4c77d_95e63da7-d34e-43e1-bc6f-1b490154b07a", "timeSeriesBucket": "f6", "assetArn": null, "assetCompositeModelDescription": null }, "assetPropertyUnit": "Celsius", "assetPropertyAlias": null }, { "assetId": "7020c8e2-e6db-40fa-9845-ed0ddd4c77d", "assetExternalId": null, "assetName": "Wind Turbine Asset Model", "assetModelId": "ec1d924f-f07d-444f-b072-e2994c165d35", "assetModelExternalId": null, "assetModelName": "Wind Turbine Asset Model", "assetPropertyId": "c706d54d-4c11-42dc-9a01-63662fc697b4", "assetPropertyExternalId": null, "assetPropertyName": "Pressure", "assetPropertyData": { "timeSeriesId": "7020c8e2-e6db-40fa-9845-ed0ddd4c77d_c706d54d-4c11-42dc-9a01-63662fc697b4", "timeSeriesBucket": "1e", "assetArn": null, "assetCompositeModelDescription": null }, "assetPropertyUnit": "Pascals", "assetPropertyAlias": null }, { "assetId": "7020c8e2-e6db-40fa-9845-ed0ddd4c77d", "assetExternalId": null, "assetName": "Wind Turbine Asset Model", "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 }, { "assetId": "7020c8e2-e6db-40fa-9845-ed0ddd4c77d_8cf1162f-dead-4fbe-b468-c8e24cde9f50", "timeSeriesBucket": "d7", "assetArn": null, "assetCompositeModelDescription": null, "assetPropertyUnit": "Celsius", "assetPropertyAlias": null }, { "assetId": "3a5f2a22-3b37-4332-9c1c-404ea1d73fab", "assetExternalId": null, "assetName": "BatchAssetModelDouble", "assetModelId": "ebc75e75e827", "assetModelExternalId": null, "assetModelName": "FlashTestAssetModelDouble", "assetPropertyId": "ab401a9176ed", "assetPropertyExternalId": null, "assetPropertyName": "measurementProperty", "assetPropertyData": { "timeSeriesId": "ff316f5ff8aa", "timeSeriesBucket": "af", "assetArn": null, "assetCompositeModelDescription": null, "assetPropertyUnit": "Celsius", "assetPropertyAlias": null } }
```

metadati della gerarchia degli asset

Quando AWS IoT SiteWise abilita il salvataggio dei dati nel livello freddo per la prima volta, i metadati della gerarchia degli asset vengono esportati nel livello freddo. Dopo la configurazione iniziale, AWS IoT SiteWise esporta i metadati della gerarchia degli asset nel livello freddo solo quando si apportano

modifiche al modello degli asset o alle definizioni degli asset. I metadati della gerarchia degli asset vengono salvati nel livello freddo nel formato JSON () delimitato da nuova riga. .ndjson

Un identificatore esterno per la gerarchia, la risorsa di destinazione o la risorsa di origine viene recuperato chiamando l'API. [DescribeAsset](#)

Percorso del file

AWS IoT SiteWise archivia i metadati della gerarchia degli asset nel livello freddo utilizzando il seguente modello.

```
{keyPrefix}/asset_hierarchy_metadata/{parentAssetId}_{hierarchyId}.ndjson
```

Ogni percorso di file verso i metadati della gerarchia degli asset nel livello freddo contiene i seguenti componenti.

Componente del percorso	Descrizione
keyPrefix	Il prefisso Amazon S3 che hai specificato nella configurazione dello AWS IoT SiteWise storage. Amazon S3 utilizza il prefisso come nome di cartella nel bucket.
asset_hierarchy_metadata	La cartella che memorizza i metadati della gerarchia degli asset. La asset_hierarchy_metadata cartella viene salvata nella cartella dei prefissi.
fileName	Il nome del file utilizza il carattere di sottolineatura (_) come delimitatore per separare quanto segue: • Il valore. parentAssetId • Il hierarchyId valore. Il file viene salvato nel .ndjson formato.

Example percorso del file ai metadati della gerarchia degli asset nel livello freddo

keyPrefix/asset_hierarchy_metadata/35901915-d476-4dca-8637-d9ed4df939ed_c5b3ced8-589a-48c7-9998-cdcccfc9747a0.ndjson

Campi

Lo schema dei metadati della gerarchia degli asset che viene esportato nel livello freddo contiene i seguenti campi.

Nome campo	Descrizione
sourceAssetId	L'ID della risorsa di origine in questa relazione tra asset.
targetAssetId	L'ID della risorsa di destinazione in questa relazione tra asset.
hierarchyId	L'ID della gerarchia.
associationType	Il tipo di associazione di questa relazione tra asset. Il valore deve essereCHILD. La risorsa di destinazione è una risorsa secondaria della risorsa di origine.

Example metadati della gerarchia degli asset nel livello freddo

```
{ "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" }
```

Per visualizzare i dati nella fase fredda

1. Accedi alla console [Amazon S3](#).
2. Nel pannello di navigazione, scegli Bucket, quindi scegli il tuo bucket Amazon S3.
3. Passa alla cartella che contiene i dati grezzi, i metadati degli asset o i metadati della gerarchia degli asset.
4. Seleziona i file, quindi da Azioni scegli Scarica.

File di indice dei dati di archiviazione

AWS IoT SiteWise utilizza questi file per ottimizzare le prestazioni delle query di dati. Vengono visualizzati nel bucket Amazon S3, ma non è necessario utilizzarli.

Percorso del file

AWS IoT SiteWise archivia i file di indice dei dati nella fase fredda utilizzando il seguente modello.

```
keyPrefix/index/series=timeseriesId/startYear=startYear/startMonth=startMonth/  
startDay=startDay/index_timeseriesId_startTimestamp_quality
```

Example percorso del file del file di indice di archiviazione dei dati

```
keyPrefix/index/series=7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_95e63da7-  
d34e-43e1-bc6f-1b490154b07a/startYear=2022/startMonth=02/startDay=03/  
index_7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_95e63da7-d34e-43e1-  
bc6f-1b490154b07a_1643846400_G00D
```

Esempi di codice per AWS IoT SiteWise l'utilizzo AWS SDKs

I seguenti esempi di codice mostrano come utilizzarlo AWS IoT SiteWise con un kit di sviluppo AWS software (SDK).

Le nozioni di base sono esempi di codice che mostrano come eseguire le operazioni essenziali all'interno di un servizio.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Nozioni di base

Salve AWS IoT SiteWise

L'esempio di codice seguente mostra come iniziare a utilizzare AWS IoT SiteWise.

Java

SDK per Java 2.x

Note

C'è altro da fare GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
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();
    }
}

```

- Per i dettagli sull'API, consulta la [ListAssetModels](#) sezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

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();
}
```

- Per i dettagli sull'API, consulta la [ListAssetModels](#) sezione AWS SDK per JavaScript API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
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"))
```

- Per i dettagli sull'API, consulta [ListAssetModels AWS SDK for Python \(Boto3\) API Reference](#).

Esempi di codice

- [Esempi di base per l'utilizzo AWS IoT SiteWiseAWS SDKs](#)
 - [Salve AWS IoT SiteWise](#)
 - [Scopri le nozioni di base di AWS IoT SiteWise con un SDK AWS](#)
 - [Azioni per l'utilizzo AWS IoT SiteWiseAWS SDKs](#)
 - [Utilizzo BatchPutAssetPropertyValue con un AWS SDK o una CLI](#)
 - [Utilizzo CreateAsset con un AWS SDK o una CLI](#)
 - [Utilizzo CreateAssetModel con un AWS SDK o una CLI](#)
 - [Utilizzo CreateGateway con un AWS SDK o una CLI](#)
 - [Utilizzo CreatePortal con un AWS SDK o una CLI](#)
 - [Utilizzo DeleteAsset con un AWS SDK o una CLI](#)
 - [Utilizzo DeleteAssetModel con un AWS SDK o una CLI](#)
 - [Utilizzo DeleteGateway con un AWS SDK o una CLI](#)
 - [Utilizzo DeletePortal con un AWS SDK o una CLI](#)
 - [Utilizzo DescribeAssetModel con un AWS SDK o una CLI](#)
 - [Utilizzo DescribeGateway con un AWS SDK o una CLI](#)
 - [Utilizzo DescribePortal con un AWS SDK o una CLI](#)
 - [Utilizzo GetAssetPropertyValue con un AWS SDK o una CLI](#)
 - [Utilizzo ListAssetModels con un AWS SDK o una CLI](#)

Esempi di base per l'utilizzo AWS IoT SiteWiseAWS SDKs

I seguenti esempi di codice mostrano come utilizzare le nozioni di base di AWS IoT SiteWise with. AWS SDKs

Esempi

- [Salve AWS IoT SiteWise](#)
- [Scopri le nozioni di base di AWS IoT SiteWise con un SDK AWS](#)
- [Azioni per l'utilizzo AWS IoT SiteWiseAWS SDKs](#)

- [Utilizzo BatchPutAssetPropertyValue con un AWS SDK o una CLI](#)
- [Utilizzo CreateAsset con un AWS SDK o una CLI](#)
- [Utilizzo CreateAssetModel con un AWS SDK o una CLI](#)
- [Utilizzo CreateGateway con un AWS SDK o una CLI](#)
- [Utilizzo CreatePortal con un AWS SDK o una CLI](#)
- [Utilizzo DeleteAsset con un AWS SDK o una CLI](#)
- [Utilizzo DeleteAssetModel con un AWS SDK o una CLI](#)
- [Utilizzo DeleteGateway con un AWS SDK o una CLI](#)
- [Utilizzo DeletePortal con un AWS SDK o una CLI](#)
- [Utilizzo DescribeAssetModel con un AWS SDK o una CLI](#)
- [Utilizzo DescribeGateway con un AWS SDK o una CLI](#)
- [Utilizzo DescribePortal con un AWS SDK o una CLI](#)
- [Utilizzo GetAssetPropertyValue con un AWS SDK o una CLI](#)
- [Utilizzo ListAssetModels con un AWS SDK o una CLI](#)

Salve AWS IoT SiteWise

L'esempio di codice seguente mostra come iniziare a utilizzare AWS IoT SiteWise.

Java

SDK per Java 2.x

Note

C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
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();
}
}

```

- Per i dettagli sull'API, consulta la [ListAssetModels](#) sezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
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();
}
```

- Per i dettagli sull'API, consulta la [ListAssetModels](#) sezione AWS SDK per JavaScript API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
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"))
```

- Per i dettagli sull'API, consulta [ListAssetModels AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scopri le nozioni di base di AWS IoT SiteWise con un SDK AWS

Gli esempi di codice seguenti mostrano come:

- Crea un modello di AWS IoT SiteWise asset.
- Crea una AWS IoT SiteWise risorsa.
- Recupera i valori dell'ID della proprietà.
- Invia dati a una AWS IoT SiteWise risorsa.
- Recupera il valore della proprietà AWS IoT SiteWise Asset.
- Crea un AWS IoT SiteWise portale.
- Crea un AWS IoT SiteWise gateway.
- Descrivi il AWS IoT SiteWise Gateway.
- Eliminare gli AWS IoT SiteWise asset.

Java

SDK per Java 2.x

Note

C'è altro su [GitHub](#). Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Esegui uno scenario interattivo che dimostri le AWS IoT SiteWise funzionalità.

```
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 Portal");
        logger.info("""
            An IoT SiteWise Portal allows you to aggregate data from multiple
            industrial sources,
            such as sensors, equipment, and control systems, into a centralized
            platform.

            """);
        waitForInputToContinue(scanner);
        String portalId;
        try {

```

```

        portalId = sitewiseActions.createPortalAsync(portalName, iamRole,
contactEmail).join();
        logger.info("Portal created successfully. Portal ID {}", portalId);
    } 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;
    }
    waitForInputToContinue(scanner);
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("7. Describe the Portal");
    logger.info(""""
        In this step, we get a description of the portal and display the
portal URL.
        """);
    waitForInputToContinue(scanner);
    try {
        String portalUrl =
sitewiseActions.describePortalAsync(portalId).join();
        logger.info("Portal URL: {}", portalUrl);
    } 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());
        }
        return;
    }
    waitForInputToContinue(scanner);

```

```

        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("8. 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("9. 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("10. 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.");
        waitForInputToContinue(scanner);
        try {
            sitewiseActions.deletePortalAsync(portalId).join();
            logger.info("Portal {} was deleted successfully.", portalId);

        } 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.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!");
}
}

```

Una classe wrapper per i metodi SDK. AWS IoT SiteWise

```

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());

```



```

        }
    });
}

/**
 * Creates a new IoT SiteWise portal.
 *
 * @param portalName the name of the portal to create.
 * @param iamRole the IAM role ARN to use for the portal.
 * @param contactEmail the email address of the portal contact.
 * @return a {@link CompletableFuture} that represents a {@link String}
result of the portal 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> createPortalAsync(String portalName, String
iamRole, String contactEmail) {
    CreatePortalRequest createPortalRequest = CreatePortalRequest.builder()
        .portalName(portalName)
        .portalDescription("This is my custom IoT SiteWise portal.")
        .portalContactEmail(contactEmail)
        .roleArn(iamRole)
        .build();

    return getAsyncClient().createPortal(createPortalRequest)
        .handle((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to create portal: {} ",
exception.getCause().getMessage());
                throw (CompletionException) exception;
            }
            return response.portalId();
        });
}

/**

```

```

    * Deletes a portal.
    *
    * @param portalId the ID of the portal to be deleted.
    * @return a {@link CompletableFuture} that represents a {@link
DeletePortalResponse}. 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<DeletePortalResponse> deletePortalAsync(String
portalId) {
        DeletePortalRequest deletePortalRequest = DeletePortalRequest.builder()
            .portalId(portalId)
            .build();

        return getAsyncClient().deletePortal(deletePortalRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("Failed to delete portal with ID: {}. Error:
{}", portalId, exception.getCause().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;
            });
    }

/**
 * Retrieves a portal's description.
 *
 * @param portalId the ID of the portal to describe.
 * @return a {@link CompletableFuture} that represents a {@link String}
result of the portal's start URL
 *
 * (see: {@link DescribePortalResponse#portalStartUrl()}). 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> describePortalAsync(String portalId) {
    DescribePortalRequest request = DescribePortalRequest.builder()
        .portalId(portalId)
        .build();

    return getAsyncClient().describePortal(request)
        .handle((response, exception) -> {
            if (exception != null) {
                logger.error("An exception occurred retrieving the portal
description: {}", exception.getCause().getMessage());
                throw (CompletionException) exception;
            }
            return response.portalStartUrl();
        });
}

/**
 * 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;
}
}
```

JavaScript

SDK per (v3) JavaScript

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
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 ListAssetModelPropertiesCommand({
          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 });
}
```

Python

SDK per Python (Boto3)

Note

C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

Esegui uno scenario interattivo al prompt dei comandi.

```
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

```

Classe IoTSitewise Wrapper che racchiude le azioni. 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

```

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Azioni per l'utilizzo AWS IoT SiteWiseAWS SDKs

I seguenti esempi di codice mostrano come eseguire singole AWS IoT SiteWise azioni con AWS SDKs. Ogni esempio include un collegamento a GitHub, dove sono disponibili le istruzioni per la configurazione e l'esecuzione del codice.

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per un elenco completo, consulta la [Documentazione di riferimento delle API AWS IoT SiteWise](#).

Esempi

- [Utilizzo BatchPutAssetPropertyValue con un AWS SDK o una CLI](#)
- [Utilizzo CreateAsset con un AWS SDK o una CLI](#)
- [Utilizzo CreateAssetModel con un AWS SDK o una CLI](#)
- [Utilizzo CreateGateway con un AWS SDK o una CLI](#)
- [Utilizzo CreatePortal con un AWS SDK o una CLI](#)
- [Utilizzo DeleteAsset con un AWS SDK o una CLI](#)
- [Utilizzo DeleteAssetModel con un AWS SDK o una CLI](#)
- [Utilizzo DeleteGateway con un AWS SDK o una CLI](#)
- [Utilizzo DeletePortal con un AWS SDK o una CLI](#)
- [Utilizzo DescribeAssetModel con un AWS SDK o una CLI](#)
- [Utilizzo DescribeGateway con un AWS SDK o una CLI](#)
- [Utilizzo DescribePortal con un AWS SDK o una CLI](#)
- [Utilizzo GetAssetPropertyValue con un AWS SDK o una CLI](#)
- [Utilizzo ListAssetModels con un AWS SDK o una CLI](#)

Utilizzo **BatchPutAssetPropertyValue** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare BatchPutAssetPropertyValue.

CLI

AWS CLI

Per inviare dati alle proprietà degli asset

L'batch-put-asset-property-valueesempio seguente invia i dati di alimentazione e temperatura alle proprietà dell'asset identificate dagli alias di proprietà.

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

Contenuto di 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
          }
        }
      ]
    }
  ]
}
```

Output:

```
{
  "errorEntries": []
}
```

Per ulteriori informazioni, consulta [Ingestione di dati utilizzando l' SiteWise API IoT nella AWS IoT SiteWise User Guide](#).

- Per i dettagli sull'API, consulta AWS CLI Command [BatchPutAssetPropertyValue](#)Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * 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());
        }
    });
}

```


- Per i dettagli sull'API, consulta la [BatchPutAssetPropertyValue](#) sezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
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;
    }
  }
};
```

- Per i dettagli sull'API, consulta la [BatchPutAssetPropertyValue](#) sezione AWS SDK per JavaScript API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
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

```

Una funzione di supporto per generare il parametro entries da un elenco di valori.

```

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

```

Ecco un esempio di elenco di valori da passare alla funzione di supporto.

```

values = [
    {

```

```

        "propertyId": humidity_property_id,
        "valueType": "doubleValue",
        "value": 65.0,
      },
      {
        "propertyId": temperature_property_id,
        "valueType": "doubleValue",
        "value": 23.5,
      },
    ]

```

- Per i dettagli sull'API, consulta [BatchPutAssetPropertyValue AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **CreateAsset** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare CreateAsset.

CLI

AWS CLI

Per creare una risorsa

L'create-assetesempio seguente crea un asset di turbine eoliche a partire da un modello di asset di turbine eoliche.

```

aws iotsitewise create-asset \
  --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
  --asset-name "Wind Turbine 1"

```

Output:

```

{
  "assetId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
  "assetArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",

```

```

    "assetStatus": {
        "state": "CREATING"
    }
}

```

Per ulteriori informazioni, consulta [Creazione di risorse](#) nella Guida per SiteWise l'utente AWS IoT.

- Per i dettagli sull'API, consulta [CreateAsset AWS CLI Command Reference](#).

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

/**
 * 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());
            }
        });
}

```

- Per i dettagli sull'API, consulta la [CreateAsset](#) sezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

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;
  }
}
};

```

- Per i dettagli sull'API, consulta la [CreateAsset](#) sezione AWS SDK per JavaScript API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

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

```

- Per i dettagli sull'API, consulta [CreateAsset AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **CreateAssetModel** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `CreateAssetModel`.

CLI

AWS CLI

Per creare un modello di asset

L'`create-asset-model` seguente crea un modello di asset che definisce una turbina eolica con le seguenti proprietà:

Numero di serie - Il numero di serie di una turbina eolica
 Potenza generata - Il flusso di dati sulla potenza generata da una turbina eolica
 Temperatura C - Il flusso di dati della temperatura da una turbina eolica in Celsius
 Temperatura F - I dati di temperatura mappati da Celsius a Fahrenheit

```

aws iotsitewise create-asset-model \
    --cli-input-json file://create-wind-turbine-model.json

```

Contenuto di `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"
                    }
                }
            }
        }
    ]
}

```

Output:

```

{
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "assetModelStatus": {
        "state": "CREATING"
    }
}

```

Per ulteriori informazioni, consulta [Definizione dei modelli di asset](#) nella AWS IoT SiteWise User Guide.

- Per i dettagli sull'API, consulta [CreateAssetModel AWS CLI](#) Command Reference.

Java

SDK per Java 2.x

 Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * 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());
            }
        });
    }
```

- Per i dettagli sull'API, consulta la [CreateAssetModel](#) sezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
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;
    }
  }
};
```

- Per i dettagli sull'API, consulta la [CreateAssetModel](#) sezione AWS SDK per JavaScript API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
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

```

Ecco un esempio di elenco di proprietà da passare alla funzione.

```

properties = [
    {
        "name": temperature_property_name,
        "dataType": "DOUBLE",
        "type": {
            "measurement": {},
        },
    },
    {
        "name": humidity_property_name,
        "dataType": "DOUBLE",
    },
]

```

```

        "type": {
            "measurement": {},
        },
    },
]

```

- Per i dettagli sull'API, consulta [CreateAssetModel AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **CreateGateway** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare CreateGateway.

CLI

AWS CLI

Per creare un gateway

L'create-gatewayesempio seguente crea un gateway che funziona su AWS IoT Greengrass.

```

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}

```

Output:

```

{
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
  "gatewayArn": "arn:aws:iotsitewise:us-west-2:123456789012:gateway/a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
}

```

Per ulteriori informazioni, consulta [Configurazione di un gateway](#) nella Guida per l'utente AWS IoT.

- Per i dettagli sull'API, consulta [CreateGateway AWS CLI Command Reference](#).

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * 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();
        });
}

```

- Per i dettagli sull'API, consulta la [CreateGateway](#) sezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

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;
    }
  }
};
```

- Per i dettagli sull'API, consulta la [CreateGateway](#) sezione AWS SDK per JavaScript API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

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

```

- Per i dettagli sull'API, consulta [CreateGateway AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **CreatePortal** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `CreatePortal`.

CLI

AWS CLI

Per creare un portale

L'`create-portal` seguente crea un portale web per un'azienda di parchi eolici. È possibile creare portali solo nella stessa regione in cui è stato abilitato il AWS Single Sign-On.

```

aws iotsitewise create-portal \
  --portal-name WindFarmPortal \
  --portal-description "A portal that contains wind farm projects for Example Corp." \
  --portal-contact-email support@example.com \
  --role-arn arn:aws:iam::123456789012:role/service-role/MySiteWiseMonitorServiceRole

```

Output:

```
{
  "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalArn": "arn:aws:iotsitewise:us-west-2:123456789012:portal/a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",
  "portalStatus": {
    "state": "CREATING"
  },
  "ssoApplicationId": "ins-a1b2c3d4-EXAMPLE"
}
```

Per ulteriori informazioni, consulta la sezione [Introduzione a AWS IoT SiteWise Monitor](#) nella AWS IoT SiteWise User Guide e [Enabling AWS SSO](#) nella AWS IoT SiteWise User Guide.

- Per i dettagli sull'API, consulta [CreatePortal AWS CLI Command Reference](#).

Java**SDK per Java 2.x****Note**

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * Creates a new IoT SiteWise portal.
 *
 * @param portalName the name of the portal to create.
 * @param iamRole the IAM role ARN to use for the portal.
 * @param contactEmail the email address of the portal contact.
 * @return a {@link CompletableFuture} that represents a {@link String}
 * result of the portal 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> createPortalAsync(String portalName, String
iamRole, String contactEmail) {
        CreatePortalRequest createPortalRequest = CreatePortalRequest.builder()
            .portalName(portalName)
            .portalDescription("This is my custom IoT SiteWise portal.")
            .portalContactEmail(contactEmail)
            .roleArn(iamRole)
            .build();

        return getAsyncClient().createPortal(createPortalRequest)
            .handle((response, exception) -> {
                if (exception != null) {
                    logger.error("Failed to create portal: {} ",
exception.getCause().getMessage());
                    throw (CompletionException) exception;
                }
                return response.portalId();
            });
    }
}

```

- Per i dettagli sull'API, consulta la [CreatePortal](#) sezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
import {
  CreatePortalCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create a Portal.
 * @param {{ portalName: string, portalContactEmail: string, roleArn: string }}
 */
export const main = async ({ portalName, portalContactEmail, roleArn }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new CreatePortalCommand({
        portalName: portalName, // The name to give the created Portal.
        portalContactEmail: portalContactEmail, // A valid contact email.
        roleArn: roleArn, // The ARN of a service role that allows the portal's
        // users to access the portal's resources.
      }),
    );
    console.log("Portal created successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "IoTSiteWiseError") {
      console.warn(
        `${caught.message}. There was a problem creating the Portal.`,
      );
    } else {
      throw caught;
    }
  }
};
```

- Per i dettagli sull'API, consulta la [CreatePortal](#) sezione AWS SDK per JavaScript API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
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_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

```

- Per i dettagli sull'API, consulta [CreatePortal AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DeleteAsset** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare DeleteAsset.

CLI

AWS CLI

Per eliminare una risorsa

L'`delete-asset` seguente elimina un asset di una turbina eolica.

```
aws iotsitewise delete-asset \  
  --asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE
```

Output:

```
{  
  "assetStatus": {  
    "state": "DELETING"  
  }  
}
```

Per ulteriori informazioni, consulta [Eliminazione delle risorse](#) nella Guida per l'utente AWS IoT.

- Per i dettagli sull'API, consulta [DeleteAsset AWS CLI Command Reference](#).

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**  
 * 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);
                }
            });
    }
}

```

- Per i dettagli sull'API, consulta la [DeleteAsset](#) sezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

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;
    }
  }
};
```

- Per i dettagli sull'API, consulta la [DeleteAsset](#) sezione AWS SDK per JavaScript API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

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

```


- Per i dettagli sull'API, consulta [DeleteAsset AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DeleteAssetModel** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare DeleteAssetModel.

CLI

AWS CLI

Per eliminare un modello di asset

L'`delete-asset-model` seguente elimina un modello di asset per turbine eoliche.

```
aws iotsitewise delete-asset-model \
  --asset-model-id a1b2c3d4-5678-90ab-cdef-1111EXAMPLE
```

Output:

```
{
  "assetModelStatus": {
    "state": "DELETING"
  }
}
```

Per ulteriori informazioni, consulta [Eliminazione dei modelli di asset](#) nella Guida per l'utente AWS IoT SiteWise.

- Per i dettagli sull'API, consulta [DeleteAssetModel AWS CLI Command Reference](#).

Java

SDK per Java 2.x

 Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * 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());
            }
        });
}
```

- Per i dettagli sull'API, consulta la [DeleteAssetModel](#) sezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
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;
    }
}
};

```

- Per i dettagli sull'API, consulta la [DeleteAssetModel](#) sezione AWS SDK per JavaScript API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

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

```

- Per i dettagli sull'API, consulta [DeleteAssetModel AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DeleteGateway** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare DeleteGateway.

CLI

AWS CLI

Per eliminare un gateway

L'`delete-gateway` seguente elimina un gateway.

```
aws iotsitewise delete-gateway \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta [Ingestione di dati utilizzando un gateway](#) nella AWS IoT SiteWise User Guide.

- Per i dettagli sull'API, consulta AWS CLI Command [DeleteGateway](#) Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * 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());
        }
    });
}

```

- Per i dettagli sull'API, consulta la [DeleteGateway](#) sezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

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;
    }
}
};

```

- Per i dettagli sull'API, consulta la [DeleteGateway](#) sezione AWS SDK per JavaScript API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

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

```

- Per i dettagli sull'API, consulta [DeleteGateway AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DeletePortal** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `DeletePortal`.

CLI

AWS CLI

Per eliminare un portale

L'`delete-portal` seguente elimina un portale Web per una società di parchi eolici.

```
aws iotsitewise delete-portal \
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaEXAMPLE
```

Output:

```
{
  "portalStatus": {
    "state": "DELETING"
  }
}
```

Per ulteriori informazioni, consulta [Eliminazione di un portale](#) nella Guida per l'utente SiteWise di AWS IoT.

- Per i dettagli sull'API, consulta [DeletePortal AWS CLI](#) Command Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

/**
 * Deletes a portal.
 *
 * @param portalId the ID of the portal to be deleted.
 * @return a {@link CompletableFuture} that represents a {@link
DeletePortalResponse}. 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<DeletePortalResponse> deletePortalAsync(String
portalId) {
    DeletePortalRequest deletePortalRequest = DeletePortalRequest.builder()
        .portalId(portalId)
        .build();

    return getAsyncClient().deletePortal(deletePortalRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to delete portal with ID: {}. Error:
{}", portalId, exception.getCause().getMessage());
            }
        });
}

```

- Per i dettagli sull'API, consulta la [DeletePortal](#) sezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
import {
  DeletePortalCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * List asset models.
 * @param {{ portalId : string }}
 */
export const main = async ({ portalId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    await client.send(
      new DeletePortalCommand({
        portalId: portalId, // The id of the portal.
      }),
    );
    console.log("Portal deleted successfully.");
    return { portalDeleted: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. There was a problem deleting the portal. Please check the portal id.`,
      );
    } else {
      throw caught;
    }
  }
};
```

- Per i dettagli sull'API, consulta la [DeletePortal](#) sezione AWS SDK per JavaScript API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
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_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
```

- Per i dettagli sull'API, consulta [DeletePortal AWS](#) SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DescribeAssetModel** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `DescribeAssetModel`.

CLI

AWS CLI

Per descrivere un modello di asset

L'`describe-asset-model` esempio seguente descrive un modello di asset per un parco eolico.

```
aws iotsitewise describe-asset-model \
    --asset-model-id a1b2c3d4-5678-90ab-cdef-2222EXAMPLE
```

Output:

```
{
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
  "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"
}
}

```

Per ulteriori informazioni, consulta [Descrivere un modello di asset specifico](#) nella AWS IoT SiteWise User Guide.

- Per i dettagli sull'API, consulta [DescribeAssetModel AWS CLI](#) Command Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

/**
 * 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;
                }
            });
    }
}

```

- Per i dettagli sull'API, consulta la [DescribeAssetModel](#) sezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
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;
    }
  }
};
```

- Per i dettagli sull'API, consulta la [DescribeAssetModel](#) sezione AWS SDK per JavaScript API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DescribeGateway** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare DescribeGateway.

CLI

AWS CLI

Per descrivere un gateway

L'`describe-gateway` esempio seguente descrive un gateway.

```
aws iotsitewise describe-gateway \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Output:

```
{
  "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
}
```

Per ulteriori informazioni, consulta [Ingestione di dati utilizzando un gateway](#) nella AWS IoT SiteWise User Guide.

- Per i dettagli sull'API, consulta AWS CLI Command [DescribeGateway](#) Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * 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());
            }
        });
}
```

- Per i dettagli sull'API, consulta la [DescribeGateway](#) sezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
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;
    }
  }
}
```

```

    }
  }
};

```

- Per i dettagli sull'API, consulta la [DescribeGateway](#) sezione AWS SDK per JavaScript API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

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

```

- Per i dettagli sull'API, consulta [DescribeGateway AWS](#) SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **DescribePortal** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `DescribePortal`.

CLI

AWS CLI

Per descrivere un portale

L'`describe-portal` seguente descrive un portale web per un'azienda di parchi eolici.

```
aws iotsitewise describe-portal \
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

Output:

```
{
  "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalArn": "arn:aws:iotsitewise:us-west-2:123456789012:portal/a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalName": "WindFarmPortal",
  "portalDescription": "A portal that contains wind farm projects for Example Corp.",
  "portalClientId": "E-a1b2c3d4e5f6_a1b2c3d4e5f6EXAMPLE",
  "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",
  "portalContactEmail": "support@example.com",
  "portalStatus": {
    "state": "ACTIVE"
  },
  "portalCreationDate": "2020-02-04T23:01:52.90248068Z",
  "portalLastUpdateDate": "2020-02-04T23:01:52.90248078Z",
  "roleArn": "arn:aws:iam::123456789012:role/MySiteWiseMonitorServiceRole"
}
```

Per ulteriori informazioni, consulta [Amministrare i portali nella Guida](#) per SiteWise l'utente AWS IoT.

- Per i dettagli sull'API, consulta AWS CLI Command [DescribePortal](#) Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * Retrieves a portal's description.
 *
 * @param portalId the ID of the portal to describe.
 * @return a {@link CompletableFuture} that represents a {@link String}
 * result of the portal's start URL
 *      (see: {@link DescribePortalResponse#portalStartUrl()}). 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> describePortalAsync(String portalId) {
    DescribePortalRequest request = DescribePortalRequest.builder()
        .portalId(portalId)
        .build();

    return getAsyncClient().describePortal(request)
        .handle((response, exception) -> {
            if (exception != null) {
                logger.error("An exception occurred retrieving the portal
description: {}", exception.getCause().getMessage());
                throw (CompletionException) exception;
            }
            return response.portalStartUrl();
        });
}
```

```
}
```

- Per i dettagli sull'API, consulta la [DescribePortal](#) sezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
import {
  DescribePortalCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Describe a portal.
 * @param {{ portalId: string }}
 */
export const main = async ({ portalId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new DescribePortalCommand({
        portalId: portalId, // The ID of the Gateway to describe.
      }),
    );
    console.log("Portal information retrieved successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. The Portal could not be found. Please check the Portal Id.`
      );
    }
  }
}
```

```

    );
  } else {
    throw caught;
  }
}
};

```

- Per i dettagli sull'API, consulta la [DescribePortal](#) sezione AWS SDK per JavaScript API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

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

```

- Per i dettagli sull'API, consulta [DescribePortal AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **GetAssetPropertyValue** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `GetAssetPropertyValue`.

CLI

AWS CLI

Per recuperare il valore corrente di una proprietà di asset

L'`get-asset-property-value` seguente recupera la potenza totale corrente di una turbina eolica.

```
aws iotsitewise get-asset-property-value \
  --asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE \
  --property-id a1b2c3d4-5678-90ab-cdef-6666EXAMPLE
```

Output:

```
{
  "propertyValue": {
    "value": {
      "doubleValue": 6890.8677520453875
    },
    "timestamp": {
      "timeInSeconds": 1580853000,
      "offsetInNanos": 0
    },
    "quality": "GOOD"
  }
}
```

Per ulteriori informazioni, consulta [Interrogazione dei valori delle proprietà degli asset correnti](#) nella AWS IoT SiteWise User Guide.

- Per i dettagli sull'API, consulta [GetAssetPropertyValue AWS CLI](#) Command Reference.

Java

SDK per Java 2.x

 Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```
/**
 * 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();
    });
}

```

- Per i dettagli sull'API, consulta la [GetAssetPropertyValue](#) sezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

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;
    }
}
};

```

- Per i dettagli sull'API, consulta la [GetAssetPropertyValue](#) sezione AWS SDK per JavaScript API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

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

```

- Per i dettagli sull'API, consulta [GetAssetPropertyValue AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **ListAssetModels** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `ListAssetModels`.

CLI

AWS CLI

Per elencare tutti i modelli di asset

L'`list-asset-models` esempio seguente elenca tutti i modelli di asset definiti nel vostro AWS account nella regione corrente.

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

Output:

```
{
  "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"
        }
    }
]
}

```

Per ulteriori informazioni, consulta [Elenco di tutti i modelli di asset](#) nella AWS IoT SiteWise User Guide.

- Per i dettagli sull'API, consulta [ListAssetModels AWS CLI](#) Command Reference.

Java

SDK per Java 2.x

Note

C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

/**
 * 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;
        });
}

```

- Per i dettagli sull'API, consulta la [ListAssetModels](#) sezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

Note

C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

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;
    }
  }
};

```

- Per i dettagli sull'API, consulta la [ListAssetModels](#) sezione AWS SDK per JavaScript API Reference.

Python

SDK per Python (Boto3)

Note

C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel [Repository di esempi di codice AWS](#).

```

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
```

- Per i dettagli sull'API, consulta [ListAssetModels AWS](#) SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Sicurezza in AWS IoT SiteWise

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS e te. Il [modello di responsabilità condivisa](#) descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- **Sicurezza del cloud:** AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. Per maggiori informazioni sui programmi di conformità applicabili a AWS IoT SiteWise, consulta la sezione [AWS Servizi rientranti nell'ambito dei programmi di conformità](#), .
- **Sicurezza nel cloud:** la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa durante l'utilizzo AWS IoT SiteWise. I seguenti argomenti mostrano come eseguire la configurazione AWS IoT SiteWise per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche a utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere AWS IoT SiteWise le tue risorse.

Argomenti

- [Protezione dei dati in AWS IoT SiteWise](#)
- [Crittografia dei dati in AWS IoT SiteWise](#)
- [Gestione delle identità e degli accessi per AWS IoT SiteWise](#)
- [Convalida della conformità per AWS IoT SiteWise](#)
- [Resilienza in AWS IoT SiteWise](#)
- [Sicurezza dell'infrastruttura in AWS IoT SiteWise](#)
- [Analisi della configurazione e delle vulnerabilità in AWS IoT SiteWise](#)
- [Endpoint VPC per AWS IoT SiteWise](#)
- [Best practice di sicurezza per AWS IoT SiteWise](#)

Protezione dei dati in AWS IoT SiteWise

Il modello di [responsabilità AWS condivisa modello](#) si applica alla protezione dei dati in AWS IoT SiteWise. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le [Domande frequenti sulla privacy dei dati](#). Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al [Modello di responsabilità condivisa AWS e GDPR](#) nel Blog sulla sicurezza AWS .

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta [Lavorare con i CloudTrail percorsi](#) nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-3](#).

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori AWS IoT SiteWise o Servizi AWS utilizzi la console, l'API o. AWS CLI AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo

vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

Argomenti

- [Privacy del traffico su Internet per AWS IoT SiteWise](#)
- [AWS IoT SiteWise Assistente al miglioramento del servizio aziendale](#)

Privacy del traffico su Internet per AWS IoT SiteWise

Le connessioni tra AWS IoT SiteWise e applicazioni locali, come i gateway SiteWise Edge, sono protette tramite connessioni Transport Layer Security (TLS). Per ulteriori informazioni, consulta [Crittografia dei dati in transito per AWS IoT SiteWise](#).

AWS IoT SiteWise non supporta connessioni tra zone di disponibilità all'interno di una AWS regione o connessioni tra account. AWS

Puoi configurare IAM Identity Center in una sola regione alla volta. SiteWise Monitor si connette alla regione che hai configurato per IAM Identity Center. Ciò significa che utilizzi una regione per l'accesso a IAM Identity Center, ma puoi creare portali in qualsiasi regione.

AWS IoT SiteWise Assistente al miglioramento del servizio aziendale

AWS IoT SiteWise L'Assistente non utilizza i dati dei clienti per migliorare il servizio o per migliorare i dati sottostanti LLMs.

Crittografia dei dati in AWS IoT SiteWise

La crittografia dei dati si riferisce alla protezione dei dati durante il transito (mentre viaggiano da AWS IoT SiteWise e verso e tra gateway SiteWise Edge e server) e quando sono inattivi (mentre sono archiviati su dispositivi locali o in AWS servizi). Puoi proteggere i dati in transito utilizzando Transport Layer Security (TLS) o i dati inattivi utilizzando la crittografia lato client.

Note

AWS IoT SiteWise esposizioni APIs di elaborazione edge ospitate all'interno di gateway SiteWise Edge e accessibili tramite la rete locale. Queste APIs sono esposte tramite una connessione TLS supportata da un certificato server di proprietà del connettore Edge. AWS IoT SiteWise Per l'autenticazione del client, APIs utilizzano una password di controllo degli

accessi. La chiave privata del certificato del server e la password di controllo degli accessi sono entrambe archiviate su disco. AWS IoT SiteWise l'elaborazione perimetrale si basa sulla crittografia del file system per la sicurezza di queste credenziali archiviate.

Per ulteriori informazioni sulla crittografia lato server e la crittografia lato client, esamina gli argomenti elencati di seguito.

Argomenti

- [Crittografia inattiva in AWS IoT SiteWise](#)
- [Crittografia dei dati in transito per AWS IoT SiteWise](#)
- [Gestione delle chiavi in AWS IoT SiteWise](#)

Crittografia inattiva in AWS IoT SiteWise

AWS IoT SiteWise archivia i tuoi dati nel AWS cloud e sui gateway AWS IoT SiteWise Edge.

Dati inattivi nel cloud AWS

AWS IoT SiteWise archivia i dati in altri AWS servizi che, per impostazione predefinita, crittografano i dati inattivi. Encryption at rest si integra con AWS Key Management Service (AWS KMS) per la gestione della chiave di crittografia utilizzata per crittografare i valori delle proprietà degli asset e i valori aggregati. AWS IoT SiteWise Puoi scegliere di utilizzare una chiave gestita dal cliente per crittografare i valori delle proprietà degli asset e aggregarli. AWS IoT SiteWise Puoi creare, gestire e visualizzare la tua chiave di crittografia tramite. AWS KMS

Puoi scegliere di Chiave di proprietà di AWS crittografare i tuoi dati o scegliere una chiave gestita dal cliente per crittografare i valori delle proprietà degli asset e i valori aggregati:

Come funziona

Encryption at rest si integra con la gestione della chiave di crittografia utilizzata AWS KMS per crittografare i dati.

- Chiave di proprietà di AWS — Chiave di crittografia predefinita. AWS IoT SiteWise possiede questa chiave. Non puoi visualizzare questa chiave nel tuo AWS account. Inoltre, non puoi visualizzare le operazioni sulla chiave nei AWS CloudTrail registri. È possibile utilizzare questa chiave senza costi aggiuntivi.

- Chiave gestita dal cliente: la chiave viene memorizzata nel tuo account, che crei, possiedi e gestisci. Hai il pieno controllo sulla chiave KMS. AWS KMS Si applicano costi aggiuntivi.

Chiavi di proprietà di AWS

Chiavi di proprietà di AWS non sono archiviate nel tuo account. Fanno parte di una raccolta di chiavi KMS che AWS possiede e gestisce per l'utilizzo in più AWS account. AWS i servizi possono essere utilizzati Chiavi di proprietà di AWS per proteggere i tuoi dati.

Non puoi visualizzare, gestire Chiavi di proprietà di AWS, utilizzare o controllare il loro utilizzo. Tuttavia, non è necessario eseguire alcuna operazione o modificare alcun programma per proteggere le chiavi che crittografano i dati.

Se li utilizzi, non ti viene addebitato alcun canone mensile o un canone di utilizzo Chiavi di proprietà di AWS e non vengono conteggiati nelle AWS KMS quote per il tuo account.

Chiavi gestite dal cliente

Le chiavi gestite dal cliente sono chiavi KMS nel tuo account create da te, di tua proprietà e gestite da te. Hai il pieno controllo su queste chiavi KMS, come le seguenti:

- Stabilire e mantenere le proprie politiche chiave, le politiche IAM e le sovvenzioni
- Abilitarli e disabilitarli
- Ruotando il loro materiale crittografico
- Aggiungere tag
- Creazione di alias che si riferiscono ad essi
- Pianificazione della loro eliminazione

Puoi anche utilizzare CloudTrail Amazon CloudWatch Logs per tenere traccia delle richieste AWS IoT SiteWise inviate a per tuo AWS KMS conto.

Se utilizzi chiavi gestite dai clienti, devi concedere AWS IoT SiteWise l'accesso alla chiave KMS memorizzata nel tuo account. AWS IoT SiteWise utilizza la crittografia a busta e la gerarchia delle chiavi per crittografare i dati. La chiave di AWS KMS crittografia viene utilizzata per crittografare la chiave principale di questa gerarchia di chiavi. Per ulteriori informazioni, consulta [Crittografia envelope](#) nella Guida per gli sviluppatori di AWS Key Management Service .

La seguente politica di esempio concede AWS IoT SiteWise le autorizzazioni per creare una chiave gestita dal cliente per conto dell'utente. Quando crei la tua chiave, devi consentire le azioni `kms:CreateGrant` e `kms:DescribeKey`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1603902045292",
      "Action": [
        "kms:CreateGrant",
        "kms:DescribeKey"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Il contesto di crittografia per la concessione creata utilizza il tuo ID `aws:iotsitewise:subscriberId` e quello dell'account.

Dati inattivi sui gateway SiteWise Edge

AWS IoT SiteWise i gateway archiviano i seguenti dati nel file system locale:

- informazioni sulla configurazione del codice sorgente OPC UA
- L'insieme di percorsi di flusso di dati OPC UA provenienti da fonti OPC UA collegate
- Dati industriali memorizzati nella cache quando il gateway SiteWise Edge perde la connessione a Internet

SiteWise I gateway Edge funzionano su. AWS IoT Greengrass AWS IoT Greengrass si affida alle autorizzazioni di file Unix e alla crittografia dell'intero disco (se abilitata) per proteggere i dati archiviati sul core. È tua responsabilità proteggere il file system e il dispositivo.

Tuttavia, AWS IoT Greengrass crittografa le copie locali dei segreti del server OPC UA recuperati da Secrets Manager. Per ulteriori informazioni, consulta [Secrets encryption](#) nella Developer Guide.AWS IoT Greengrass Version 1

Per ulteriori informazioni sulla crittografia a riposo sui AWS IoT Greengrass core, consulta [Encryption at rest](#) nella AWS IoT Greengrass Version 1 Developer Guide.

Crittografia dei dati in transito per AWS IoT SiteWise

AWS IoT SiteWise utilizza la crittografia in transito per proteggere i dati trasmessi tra i dispositivi, i gateway e il AWS cloud. La comunicazione con AWS IoT SiteWise è crittografata tramite HTTPS e TLS 1.2, garantendo che i dati rimangano riservati e protetti da accessi o intercettazioni non autorizzati.

Esistono tre modalità di comunicazione in cui i dati sono in transito:

- [Tramite Internet: la](#) comunicazione tra dispositivi locali (inclusi i gateway SiteWise Edge) AWS IoT SiteWise è crittografata.
- [Tramite la rete locale:](#) la comunicazione tra OpsHub for SiteWise application e i gateway SiteWise Edge è sempre crittografata. La comunicazione tra l'applicazione di SiteWise monitoraggio in esecuzione nel browser e i gateway SiteWise Edge è sempre crittografata. La comunicazione tra i gateway SiteWise Edge e le sorgenti OPC UA può essere crittografata.
- [Tra i componenti sui gateway SiteWise Edge:](#) la comunicazione tra AWS IoT Greengrass i componenti sui gateway SiteWise Edge non è crittografata.

Argomenti

- [Dati in transito su Internet](#)
- [Dati in transito sulla rete locale](#)
- [Dati in transito tra componenti locali su SiteWise Edge](#)

Dati in transito su Internet

AWS IoT SiteWise utilizza Transport Layer Security (TLS) per crittografare tutte le comunicazioni su Internet. Tutti i dati inviati al AWS Cloud vengono inviati tramite una connessione TLS utilizzando i protocolli MQTT o HTTPS, quindi sono sicuri per impostazione predefinita. SiteWise I gateway edge, che funzionano su AWS IoT Greengrass, e le notifiche sui valori delle proprietà utilizzano il modello di sicurezza del AWS IoT trasporto. Per ulteriori informazioni, consulta l'argomento relativo alla [sicurezza del trasporto](#) nella Guida per gli sviluppatori AWS IoT .

Dati in transito sulla rete locale

SiteWise Gli edge gateway seguono le specifiche OPC UA per la comunicazione con le sorgenti OPC UA locali. È tua responsabilità configurare le origini in modo da utilizzare una modalità di sicurezza dei messaggi che esegue la crittografia dei dati in transito.

Se si sceglie una modalità di sicurezza dei messaggi di firma, i dati in transito tra i gateway SiteWise Edge e le sorgenti vengono firmati ma non crittografati. Se si sceglie una modalità di sicurezza dei messaggi di firma e crittografia, i dati in transito tra i gateway SiteWise Edge e le fonti vengono firmati e crittografati. Per ulteriori informazioni sulla configurazione delle origini, consulta [Aggiungi fonti di dati al tuo gateway AWS IoT SiteWise Edge](#).

La comunicazione tra l'applicazione della console Edge e i gateway SiteWise Edge è sempre crittografata tramite TLS. Il connettore SiteWise Edge sul gateway SiteWise Edge genera e archivia un certificato autofirmato per poter stabilire una connessione TLS con la console edge per l'applicazione. AWS IoT SiteWise È necessario copiare questo certificato dal gateway SiteWise Edge alla console edge per l' AWS IoT SiteWise applicazione prima di connettere l'applicazione al gateway SiteWise Edge. Ciò garantisce che la console perimetrale AWS IoT SiteWise dell'applicazione sia in grado di verificare che sia connessa al gateway SiteWise Edge affidabile.

Oltre a TLS per la segretezza e l'autenticità del server, SiteWise Edge utilizza il protocollo SigV4 per stabilire l'autenticità dell'applicazione della console perimetrale. Il connettore SiteWise Edge sul gateway SiteWise Edge accetta e archivia una password per poter verificare le connessioni in entrata dall'applicazione della console edge, dall'applicazione SiteWise Monitor in esecuzione nei browser e da altri client basati sull'SDK. AWS IoT SiteWise

Per ulteriori informazioni sulla generazione della password e del certificato del server, consulta [the section called “Gestisci i gateway”](#)

Dati in transito tra componenti locali su SiteWise Edge

SiteWise Gli edge gateway funzionano AWS IoT Greengrass, il che non crittografa i dati scambiati localmente sul AWS IoT Greengrass core perché i dati non escono dal dispositivo. Ciò include la comunicazione tra AWS IoT Greengrass componenti come il connettore. AWS IoT SiteWise Per ulteriori informazioni, consulta [Data on the core device](#) nella AWS IoT Greengrass Version 1 Developer Guide.

Gestione delle chiavi in AWS IoT SiteWise

AWS IoT SiteWise gestione delle chiavi nel cloud

Per impostazione predefinita, AWS IoT SiteWise vengono utilizzati Chiavi gestite da AWS per proteggere i dati nel AWS cloud. Puoi aggiornare le impostazioni per utilizzare una chiave gestita dal cliente per crittografare alcuni dati. AWS IoT SiteWise È possibile creare, gestire e visualizzare la chiave di crittografia tramite AWS Key Management Service (AWS KMS).

AWS IoT SiteWise supporta la crittografia lato server con chiavi gestite dal cliente archiviate AWS KMS per crittografare i seguenti dati:

- valori delle proprietà degli asset
- Valori aggregati

Note

Altri dati e risorse vengono crittografati utilizzando la crittografia predefinita con chiavi gestite da AWS IoT SiteWise. Questa chiave è memorizzata nell' AWS IoT SiteWise account.

Per ulteriori informazioni, vedi [Cos'è AWS Key Management Service?](#) nella Guida per gli AWS Key Management Service sviluppatori.

Abilita la crittografia utilizzando chiavi gestite dal cliente

Per utilizzare le chiavi gestite dal cliente con AWS IoT SiteWise, devi aggiornare AWS IoT SiteWise le impostazioni.

Per abilitare la crittografia utilizzando le chiavi KMS

1. Passare alla [console AWS IoT SiteWise](#).
2. Scegli Impostazioni account e scegli Modifica per aprire la pagina Modifica impostazioni account.
3. Per Tipo di chiave di crittografia, scegli Scegli una AWS KMS chiave diversa. Ciò consente la crittografia con chiavi gestite dal cliente archiviate in AWS KMS.

 Note

Attualmente, è possibile utilizzare la crittografia a chiave gestita dal cliente solo per i valori delle proprietà degli asset e i valori aggregati.

4. Scegli la tua chiave KMS con una delle seguenti opzioni:

- Per utilizzare una chiave KMS esistente: scegli l'alias della tua chiave KMS dall'elenco.
- Per creare una nuova chiave KMS, scegli Crea una chiave. AWS KMS

 Note

In questo modo si apre AWS KMS Pannello di controllo. Per ulteriori informazioni sulla creazione di una chiave KMS, consulta [Creating keys](#) nella AWS Key Management Service Developer Guide.

5. Scegli Salva per aggiornare le impostazioni.

SiteWise Gestione delle chiavi del gateway Edge

SiteWise I gateway edge funzionano su AWS IoT Greengrass e i dispositivi AWS IoT Greengrass principali utilizzano chiavi pubbliche e private per autenticarsi con il AWS cloud e crittografare i segreti locali, come i segreti di autenticazione OPC UA. Per ulteriori informazioni, consulta [Gestione delle chiavi](#) nella Guida per gli sviluppatori.AWS IoT Greengrass Version 1

Gestione delle identità e degli accessi per AWS IoT SiteWise

AWS Identity and Access Management (IAM) è un software Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse. AWS IoT SiteWise IAM è uno Servizio AWS strumento che puoi utilizzare senza costi aggiuntivi.

Argomenti

- [Pubblico per la AWS IoT SiteWise sicurezza](#)
- [Effettua l'autenticazione con identità in AWS IoT SiteWise](#)

- [Come AWS IoT SiteWise funziona con IAM](#)
- [AWS politiche gestite per AWS IoT SiteWise](#)
- [Usa ruoli collegati ai servizi per AWS IoT SiteWise](#)
- [Imposta le autorizzazioni per gli allarmi relativi agli eventi in AWS IoT SiteWise](#)
- [Prevenzione sostitutiva confusa tra diversi servizi in AWS IoT SiteWise](#)
- [Risolvi i problemi relativi all' AWS IoT SiteWise identità e all'accesso](#)

Pubblico per la AWS IoT SiteWise sicurezza

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto. AWS IoT SiteWise

Utente del servizio: se utilizzi il AWS IoT SiteWise servizio per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più AWS IoT SiteWise funzionalità per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità di AWS IoT SiteWise, consulta [Risolvi i problemi relativi all' AWS IoT SiteWise identità e all'accesso](#).

Amministratore del servizio: se sei responsabile delle AWS IoT SiteWise risorse della tua azienda, probabilmente hai pieno accesso a AWS IoT SiteWise. È tuo compito determinare a quali AWS IoT SiteWise funzionalità e risorse devono accedere gli utenti del servizio. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per saperne di più su come la tua azienda può utilizzare IAM con AWS IoT SiteWise, consulta [Come AWS IoT SiteWise funziona con IAM](#).

Amministratore IAM: un amministratore IAM potrebbe essere interessato a ottenere dei dettagli su come scrivere policy per gestire l'accesso a AWS IoT SiteWise. Per visualizzare esempi di policy AWS IoT SiteWise basate sull'identità che puoi utilizzare in IAM, consulta. [AWS IoT SiteWise esempi di politiche basate sull'identità](#)

Effettua l'autenticazione con identità in AWS IoT SiteWise

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi [Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente](#).

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sul metodo consigliato per la firma delle richieste, consulta [Signature Version 4 AWS per le richieste API](#) nella Guida per l'utente IAM.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta [Autenticazione a più fattori](#) nella Guida per l'utente di AWS IAM Identity Center e [Utilizzo dell'autenticazione a più fattori \(MFA\)AWS in IAM](#) nella Guida per l'utente IAM.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'utente IAM.

Utenti e gruppi IAM

Un [utente IAM](#) è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, se si hanno casi d'uso specifici che richiedono credenziali a lungo termine con

utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina [Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine](#) nella Guida per l'utente IAM.

Un [gruppo IAM](#) è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, potresti avere un gruppo denominato IAMAdminse concedere a quel gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta [Casi d'uso per utenti IAM](#) nella Guida per l'utente IAM.

Ruoli IAM

Un [ruolo IAM](#) è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Per assumere temporaneamente un ruolo IAM in AWS Management Console, puoi [passare da un ruolo utente a un ruolo IAM \(console\)](#). Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta [Utilizzo di ruoli IAM](#) nella Guida per l'utente IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- **Accesso utente federato:** per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta [Create a role for a third-party identity provider \(federation\)](#) nella Guida per l'utente IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta [Set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center.
- **Autorizzazioni utente IAM temporanee:** un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.

- **Accesso multi-account:** è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.
- **Accesso a più servizi:** alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è normale che quel servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
- **Sessioni di accesso inoltrato (FAS):** quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama un Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta [Forward access sessions](#).
- **Ruolo di servizio:** un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione [Create a role to delegate permissions to an Servizio AWS](#) nella Guida per l'utente IAM.
- **Ruolo collegato al servizio:** un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.
- **Applicazioni in esecuzione su Amazon EC2:** puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un' EC2 istanza e che AWS CLI effettuano richieste AWS API. Questa soluzione è preferibile alla memorizzazione delle chiavi di accesso all'interno dell' EC2 istanza. Per assegnare un AWS ruolo a un' EC2 istanza e renderlo disponibile per tutte le sue applicazioni, crea un profilo di istanza collegato all'istanza. Un profilo di istanza contiene il ruolo e consente ai programmi in esecuzione sull' EC2 istanza di ottenere

credenziali temporanee. Per ulteriori informazioni, consulta [Utilizzare un ruolo IAM per concedere le autorizzazioni alle applicazioni in esecuzione su EC2 istanze Amazon](#) nella IAM User Guide.

Come AWS IoT SiteWise funziona con IAM

Prima di utilizzare AWS Identity and Access Management (IAM) per gestire l'accesso a AWS IoT SiteWise, è necessario comprendere con quali funzionalità IAM è possibile utilizzare AWS IoT SiteWise.

Funzionalità IAM	Supporto da AWS IoT SiteWise
Policy basate sull'identità con autorizzazioni a livello di risorse	Sì
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione delle policy	Sì
Policy basate sulle risorse	No
Elenchi di controllo degli accessi (ACLs)	No
Autorizzazione basata su tag (ABAC)	Sì
Credenziali temporanee	Sì
Sessioni di accesso diretto (FAS)	Sì
Ruoli collegati al servizio	Sì
Ruoli di servizio	Sì

Per avere una visione di alto livello di come AWS IoT SiteWise e altri AWS servizi funzionano con IAM, consulta [AWS i servizi che funzionano con IAM nella IAM User Guide](#).

Indice

- [AWS IoT SiteWise Ruoli IAM](#)
 - [Usa credenziali temporanee con AWS IoT SiteWise](#)
 - [Sessioni di accesso diretto \(FAS\) per AWS IoT SiteWise](#)
 - [Ruoli collegati ai servizi](#)
 - [Ruoli dei servizi](#)
 - [Scegli un ruolo IAM in AWS IoT SiteWise](#)
- [Autorizzazione basata su tag AWS IoT SiteWise](#)
- [AWS IoT SiteWise politiche basate sull'identità](#)
 - [Azioni di policy](#)
 - [BatchPutAssetPropertyValue autorizzazione](#)
 - [Risorse relative alle policy](#)
 - [Chiavi di condizione delle policy](#)
 - [Esempi](#)
- [AWS IoT SiteWise esempi di politiche basate sull'identità](#)
 - [Best practice delle policy](#)
 - [Utilizzo della console AWS IoT SiteWise](#)
 - [Consentire agli utenti di visualizzare le loro autorizzazioni](#)
 - [Consenti agli utenti di inserire dati nelle risorse in un'unica gerarchia](#)
 - [Visualizza le AWS IoT SiteWise risorse in base ai tag](#)
- [Gestisci l'accesso utilizzando le politiche in AWS IoT SiteWise](#)
 - [Policy basate sull'identità](#)
 - [Policy basate sulle risorse](#)
 - [Elenchi di controllo degli accessi \(\) ACLs](#)
 - [Altri tipi di policy](#)
 - [Più tipi di policy](#)

AWS IoT SiteWise Ruoli IAM

Un [ruolo IAM](#) è un'entità all'interno dell'account AWS che dispone di autorizzazioni specifiche.

Usa credenziali temporanee con AWS IoT SiteWise

È possibile utilizzare credenziali temporanee per effettuare l'accesso con la federazione, assumere un ruolo IAM o un ruolo multi-account. È possibile ottenere credenziali di sicurezza temporanee chiamando operazioni AWS STS API come [AssumeRole](#) o [GetFederationToken](#).

AWS IoT SiteWise supporta l'utilizzo di credenziali temporanee.

SiteWise Monitor supporta gli utenti federati per accedere ai portali. Gli utenti del portale si autenticano con le proprie credenziali IAM Identity Center o IAM.

Important

Gli utenti o i ruoli devono disporre dell'`iotsitewise:DescribePortal` autorizzazione per accedere al portale.

Quando un utente accede a un portale, SiteWise Monitor genera una politica di sessione che fornisce le seguenti autorizzazioni:

- Accesso in sola lettura alle risorse e ai dati delle risorse presenti AWS IoT SiteWise nel tuo account a cui il ruolo del portale fornisce l'accesso.
- Accesso ai progetti in tale portale per il quale l'utente dispone dell'accesso amministratore (proprietario del progetto) o di sola lettura (visualizzatore progetto).

Per ulteriori informazioni sulle autorizzazioni utente del portale federato, consulta [Utilizza i ruoli di servizio per AWS IoT SiteWise Monitor](#).

Sessioni di accesso diretto (FAS) per AWS IoT SiteWise

Supporta l'inoltro delle sessioni di accesso (FAS): sì

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama un Servizio AWS,

in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta [Forward access sessions](#).

Ruoli collegati ai servizi

[I ruoli collegati](#) ai AWS servizi consentono ai servizi di accedere alle risorse di altri servizi per completare un'azione per conto dell'utente. I ruoli collegati ai servizi vengono visualizzati nell' AWS account dell'utente e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non può modificarle.

AWS IoT SiteWise supporta ruoli collegati ai servizi. Per ulteriori informazioni su come creare e gestire i ruoli collegati ai servizi AWS IoT SiteWise , consulta [Usa ruoli collegati ai servizi per AWS IoT SiteWise](#).

Ruoli dei servizi

Questa caratteristica consente a un servizio di assumere un [ruolo di servizio](#) per conto dell'utente. Questo ruolo consente al servizio di accedere alle risorse in altri servizi per completare un'azione per conto dell'utente. I ruoli di servizio vengono visualizzati nell' AWS account e sono di proprietà dell'account. Ciò significa che un amministratore IAM può modificare le autorizzazioni per questo ruolo. Tuttavia, questo potrebbe pregiudicare la funzionalità del servizio.

AWS IoT SiteWise utilizza un ruolo di servizio per consentire agli utenti del portale SiteWise Monitor di accedere ad alcune AWS IoT SiteWise risorse per conto dell'utente. Per ulteriori informazioni, consulta [Utilizza i ruoli di servizio per AWS IoT SiteWise Monitor](#).

È necessario disporre delle autorizzazioni necessarie prima di poter creare modelli di AWS IoT Events allarme in AWS IoT SiteWise. Per ulteriori informazioni, consulta [Imposta le autorizzazioni per gli allarmi relativi agli eventi in AWS IoT SiteWise](#).

Scegli un ruolo IAM in AWS IoT SiteWise

Quando crei una portal risorsa in AWS IoT SiteWise, devi scegliere un ruolo per consentire agli utenti federati del tuo portale SiteWise Monitor di accedere per tuo AWS IoT SiteWise conto. Se in precedenza hai creato un ruolo di servizio, ti AWS IoT SiteWise fornisce un elenco di ruoli tra cui scegliere. In caso contrario, puoi creare un ruolo con le autorizzazioni richieste quando crei un

portale. È importante scegliere un ruolo che consenta l'accesso agli asset e ai dati degli asset. Per ulteriori informazioni, consulta [Utilizza i ruoli di servizio per AWS IoT SiteWise Monitor](#).

Autorizzazione basata su tag AWS IoT SiteWise

È possibile allegare tag alle AWS IoT SiteWise risorse o passare tag in una richiesta a AWS IoT SiteWise. Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'[elemento condizione](#) di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`. Per ulteriori informazioni sul tagging delle risorse di AWS IoT SiteWise, consulta [Tagga le tue AWS IoT SiteWise risorse](#).

Per visualizzare una policy basata sulle identità di esempio per limitare l'accesso a una risorsa basata su tag su tale risorsa, consulta [Visualizza le AWS IoT SiteWise risorse in base ai tag](#).

AWS IoT SiteWise politiche basate sull'identità

Le policy IAM ti consentono di controllare chi può fare cosa AWS IoT SiteWise. Puoi decidere quali azioni sono consentite o meno e impostare condizioni specifiche per tali azioni. Ad esempio, puoi stabilire delle regole su chi può visualizzare o modificare le informazioni AWS IoT SiteWise. AWS IoT SiteWise supporta azioni, risorse e chiavi di condizione specifiche. Per informazioni su tutti gli elementi utilizzati in una policy JSON, consulta [Documentazione di riferimento degli elementi delle policy JSON IAM](#) nella Guida per l'utente IAM.

Azioni di policy

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le operazioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Le azioni politiche AWS IoT SiteWise utilizzano il seguente prefisso prima dell'azione: `iotsitewise:`. Ad esempio, per concedere a qualcuno il permesso di caricare i dati delle proprietà degli asset AWS IoT SiteWise con l'operazione `BatchPutAssetPropertyValue` API, includi `iotsitewise:BatchPutAssetPropertyValue` azione nella sua politica. Le

dichiarazioni politiche devono includere un `NotAction` elemento `Action` or. AWS IoT SiteWise definisce il proprio set di azioni che descrivono le attività che è possibile eseguire con questo servizio.

Per specificare più operazioni in una singola istruzione, separarle con una virgola come mostrato di seguito.

```
"Action": [  
  "iotsitewise:action1",  
  "iotsitewise:action2"  
]
```

Puoi specificare più operazioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le operazioni che iniziano con la parola `Describe`, includi la seguente operazione.

```
"Action": "iotsitewise:Describe*"
```

Per visualizzare un elenco di AWS IoT SiteWise azioni, consulta [Actions defined by AWS IoT SiteWise](#) nella IAM User Guide.

BatchPutAssetPropertyValue autorizzazione

AWS IoT SiteWise autorizza l'accesso all'[BatchPutAssetPropertyValue](#) azione in un modo insolito. Per la maggior parte delle azioni, quando consenti o neghi l'accesso, tale azione restituisce un errore se le autorizzazioni non vengono concesse. Con `BatchPutAssetPropertyValue`, puoi inviare più immissioni di dati a diverse risorse e proprietà delle risorse in un'unica richiesta API. AWS IoT SiteWise autorizza ogni immissione di dati in modo indipendente. Per ogni singola immissione non autorizzata nella richiesta, AWS IoT SiteWise include un errore `AccessDeniedException` nell'elenco restituito. AWS IoT SiteWise riceve i dati relativi a qualsiasi immissione autorizzata e che ha esito positivo, anche se un'altra immissione nella stessa richiesta ha esito negativo.

Important

Prima di importare dati in un flusso di dati, procedi come segue:

- Autorizzate la `time-series` risorsa se utilizzate un alias di proprietà per identificare il flusso di dati.
- Autorizzate la `asset` risorsa se utilizzate un ID di risorsa per identificare la risorsa che contiene la proprietà della risorsa associata.

Risorse relative alle policy

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo [nome della risorsa Amazon \(ARN\)](#). È possibile eseguire questa operazione per operazioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*" 
```

Ogni dichiarazione di policy IAM si applica alle risorse specificate utilizzando le loro ARNs. Di seguito è riportata la sintassi generale di un ARN.

```
arn:${Partition}:${Service}:${Region}:${Account}:${ResourceType}/${ResourcePath}
```

Per ulteriori informazioni sul formato di ARNs, consulta [Identificare AWS le risorse con Amazon Resource Names \(ARNs\)](#).

Ad esempio, per specificare l'asset con ID `a1b2c3d4-5678-90ab-cdef-2222EXAMPLE` nell'istruzione, utilizza il seguente ARN.

```
"Resource": "arn:aws:iotsitewise:region:123456789012:asset/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE" 
```

Per specificare tutti i flussi di dati che appartengono a un account specifico, usa il carattere jolly (*):

```
"Resource": "arn:aws:iotsitewise:region:123456789012:time-series/*" 
```

Per specificare tutti gli asset che appartengono a un account specifico, utilizza il carattere jolly (*):

```
"Resource": "arn:aws:iotsitewise:region:123456789012:asset/*" 
```

Alcune AWS IoT SiteWise azioni, come quelle per la creazione di risorse, non possono essere eseguite su una risorsa specifica. In questi casi, è necessario utilizzare il carattere jolly (*).

```
"Resource": "*"
```

Per specificare più risorse in un'unica istruzione, separale ARNs con virgole.

```
"Resource": [  
  "resource1",  
  "resource2"  
]
```

Per visualizzare un elenco dei tipi di AWS IoT SiteWise risorse e relativi ARNs, consulta [Tipi di risorse definiti da AWS IoT SiteWise](#) nella Guida per l'utente IAM. Per informazioni sulle operazioni con cui è possibile specificare l'ARN di ogni risorsa, consulta la sezione [Operazioni definite da AWS IoT SiteWise](#).

Chiavi di condizione delle policy

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento Condition (o blocco Condition) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento Condition è facoltativo. È possibile compilare espressioni condizionali che utilizzano [operatori di condizione](#), ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi Condition in un'istruzione o più chiavi in un singolo elemento Condition, questi vengono valutati da AWS utilizzando un'operazione AND logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. OR Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

È possibile anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, è possibile autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta [Elementi delle policy IAM: variabili e tag](#) nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di [contesto delle condizioni AWS globali nella Guida](#) per l'utente IAM.

 Important

Molte chiavi di condizione sono specifiche di una risorsa e alcune operazioni API utilizzano più risorse. Se scrivi una dichiarazione di policy con una chiave di condizione, utilizza l'elemento `Resource` della dichiarazione per specificare la risorsa a cui viene applicata la chiave di condizione. In caso contrario, la policy potrebbe impedire agli utenti di eseguire operazioni perché il controllo della condizione ha esito negativo per le risorse alle quali non viene applicata la chiave di condizione. Se non desideri specificare una risorsa oppure se hai scritto l'elemento `Action` della policy in modo da includere più operazioni API, devi utilizzare il tipo di condizione `...IfExists` per assicurarti che la chiave di condizione venga ignorata per le risorse che non la utilizzano. Per ulteriori informazioni, consulta... [IfExists](#) condizioni nella Guida per l'utente IAM.

AWS IoT SiteWise definisce il proprio set di chiavi di condizione e supporta anche l'utilizzo di alcune chiavi di condizione globali. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le [chiavi di contesto delle condizioni AWS globali](#) nella Guida per l'utente IAM.

AWS IoT SiteWise chiavi di condizione

Chiave di condizione	Descrizione	Tipi
<code>iotsitewise:isAssociatedWithAssetProperty</code>	Se i flussi di dati sono associati a una proprietà dell'asset. Utilizzate questa chiave di condizione per definire le autorizzazioni in base all'esistenza di una proprietà di asset associata per i flussi di dati. Valore di esempio: <code>true</code>	Stringa
<code>iotsitewise:assetHierarchyPath</code>	Il percorso gerarchico della risorsa, che è una stringa di	Stringa

Chiave di condizione	Descrizione	Tipi
	risorse separate IDs ciascuna da una barra. Utilizzare questa chiave di condizione per definire le autorizzazioni in base a un sottoinsieme della gerarchia di tutti gli asset nell'account. Valore di esempio: / a1b2c3d4-5678-90ab- cdef-2222EXAMPLE/ a1b2c3d4-5678-90ab- cdef-6666EXAMPLE	
<code>iotsitewise:propertyId</code>	L'ID di una proprietà di asset. Utilizza questa chiave di condizione per definire le autorizzazioni in base a una proprietà specificata di un modello di asset. Questa chiave di condizione si applica a tutti gli asset di tale modello. Valore di esempio: a1b2c3d4-5678-90ab- cdef-3333EXAMPLE	Stringa

Chiave di condizione	Descrizione	Tipi
<code>iotsitewise:childAssetId</code>	L'ID di un asset associato come un figlio a un altro asset. Utilizza questa chiave di condizione per definire le autorizzazioni in base agli asset figlio. Per definire le autorizzazioni in base agli asset padre, utilizza la sezione delle risorse di un'istruzione policy. Valore di esempio: a1b2c3d4-5678-90ab-cdef-6666EXAMPLE	Stringa
<code>iotsitewise:iam</code>	L'ARN di un'identità IAM quando si elencano le politiche di accesso. Utilizza questa chiave di condizione per definire le autorizzazioni delle politiche di accesso per un'identità IAM. Valore di esempio: arn:aws:iam::123456789012:user/JohnDoe	Stringa, Null
<code>iotsitewise:propertyAlias</code>	L'alias che identifica una proprietà o un flusso di dati di una risorsa. Utilizzate questa chiave di condizione per definire le autorizzazioni in base all'alias.	Stringa

Chiave di condizione	Descrizione	Tipi
<code>iotsitewise:user</code>	L'ID di un utente di IAM Identity Center quando elenca le politiche di accesso. Utilizza questa chiave di condizione per definire le autorizzazioni delle policy di accesso per un utente IAM Identity Center. Valore di esempio: a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-aaaaEXAMPLE	Stringa, Null
<code>iotsitewise:group</code>	L'ID di un gruppo IAM Identity Center quando si elencano le politiche di accesso. Utilizza questa chiave di condizione per definire le autorizzazioni delle policy di accesso per un gruppo IAM Identity Center. Valore di esempio: a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE	Stringa, Null
<code>iotsitewise:portal</code>	ID di un portale in una policy di accesso. Utilizza questa chiave di condizione per definire le autorizzazioni delle policy di accesso in base a un portale. Valore di esempio: a1b2c3d4-5678-90ab-cdef-77777EXAMPLE	Stringa, Null

Chiave di condizione	Descrizione	Tipi
<code>iotsitewise:project</code>	L'ID di un progetto in una policy di accesso o l'ID di un progetto per un pannello di controllo. Utilizza questa chiave di condizione per definire il pannello di controllo o le autorizzazioni delle policy di accesso in base a un progetto. Valore di esempio: a1b2c3d4-5678-90ab-cdef-88888EXAMPLE	Stringa, Null

Per sapere con quali azioni e risorse puoi utilizzare una chiave di condizione, consulta [Azioni definite da AWS IoT SiteWise](#).

Esempi

Per visualizzare esempi di politiche AWS IoT SiteWise basate sull'identità, vedere. [AWS IoT SiteWise esempi di politiche basate sull'identità](#)

AWS IoT SiteWise esempi di politiche basate sull'identità

Per impostazione predefinita, le entità (utenti e ruoli) non sono autorizzate a creare o modificare AWS IoT SiteWise risorse. Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS l'API. Per modificare le autorizzazioni, un amministratore AWS Identity and Access Management (IAM) deve effettuare le seguenti operazioni:

1. Crea policy IAM che concedano a utenti e ruoli il permesso di eseguire operazioni API specifiche sulle risorse di cui hanno bisogno.
2. Allega tali policy agli utenti o ai gruppi che richiedono tali autorizzazioni.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consultare [Creazione di policy nella scheda JSON](#) nella Guida per l'utente di IAM.

Argomenti

- [Best practice delle policy](#)
- [Utilizzo della console AWS IoT SiteWise](#)
- [Consentire agli utenti di visualizzare le loro autorizzazioni](#)
- [Consenti agli utenti di inserire dati nelle risorse in un'unica gerarchia](#)
- [Visualizza le AWS IoT SiteWise risorse in base ai tag](#)

Best practice delle policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare AWS IoT SiteWise risorse nel tuo account. Queste azioni possono comportare costi aggiuntivi per l'Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni. AWS Sono disponibili nel tuo Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta [Policy gestite da AWS](#) o [Policy gestite da AWS per le funzioni dei processi](#) nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegio minimo: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta [Policy e autorizzazioni in IAM](#) nella Guida per l'utente IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse è possibile aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente IAM.

- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta [Convalida delle policy per il Sistema di analisi degli accessi IAM](#) nella Guida per l'utente IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta [Protezione dell'accesso API con MFA](#) nella Guida per l'utente IAM.

Per maggiori informazioni sulle best practice in IAM, consulta [Best practice di sicurezza in IAM](#) nella Guida per l'utente di IAM.

Utilizzo della console AWS IoT SiteWise

Per accedere alla AWS IoT SiteWise console, è necessario un set di autorizzazioni di base. Queste autorizzazioni ti consentono di visualizzare e gestire i dettagli sulle AWS IoT SiteWise risorse del tuo AWS account.

Se definisci una politica troppo restrittiva, la console potrebbe non funzionare come previsto per gli utenti o i ruoli (entità) che prevedono tale politica. Per garantire che tali entità possano ancora utilizzare la AWS IoT SiteWise console, allega loro la policy [AWSIoTSiteWiseConsoleFullAccess](#) gestita o definisci autorizzazioni equivalenti per tali entità. Per ulteriori informazioni, consulta [Aggiunta di autorizzazioni a un utente](#) nella Guida per l'utente IAM.

Se le entità utilizzano solo la AWS Command Line Interface (CLI) o l' AWS IoT SiteWise API e non la console, non necessitano di queste autorizzazioni minime. In tal caso, è sufficiente consentire loro di accedere alle azioni specifiche di cui hanno bisogno per le loro attività API.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o a livello di codice. AWS CLI 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": "*"
  }
]
}

```

Consenti agli utenti di inserire dati nelle risorse in un'unica gerarchia

In questo esempio, volete concedere a un utente del vostro AWS account l'accesso per scrivere dati su tutte le proprietà delle risorse in una gerarchia specifica di risorse, a partire dalla risorsa principale. `a1b2c3d4-5678-90ab-cdef-2222EXAMPLE` La policy concede l'autorizzazione `iotsitewise:BatchPutAssetPropertyValue` all'utente. Questa policy utilizza la chiave di condizione `iotsitewise:assetHierarchyPath` per limitare l'accesso agli asset il cui percorso della gerarchia corrisponde all'asset o ai relativi discendenti.

```

{
  "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/*"
        ]
      }
    }
  }
]
}

```

Visualizza le AWS IoT SiteWise risorse in base ai tag

Utilizza le condizioni della tua politica basata sull'identità per controllare l'accesso alle AWS IoT SiteWise risorse in base ai tag. Questo esempio mostra come creare una politica che consenta la visualizzazione delle risorse. Tuttavia, l'autorizzazione viene concessa solo se il valore del tag di asset Owner corrisponde al nome utente dell'utente. Questo criterio concede inoltre l'autorizzazione a completare questa azione sulla console.

```

{
  "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}"
      }
    }
  ]
}
```

Allega questa politica agli utenti del tuo account. Se un utente denominato `richard-roe` tenta di visualizzare una AWS IoT SiteWise risorsa, la risorsa deve essere contrassegnata con `Owner=richard-roe` o `owner=richard-roe`. In caso contrario, a Richard viene negato l'accesso. I nomi delle chiavi dei tag di condizione non fanno distinzione tra maiuscole e minuscole. Quindi, `Owner` corrisponde a entrambi `Owner` e `owner`. Per ulteriori informazioni, consulta la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente di IAM.

Gestisci l'accesso utilizzando le politiche in AWS IoT SiteWise

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta [Panoramica delle policy JSON](#) nella Guida per l'utente IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'operazione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall'AWS Management Console AWS CLI, dall'AWS CLI o dall'AWS API.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente](#) nella Guida per l'utente IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta [Scelta fra policy gestite e policy inline](#) nella Guida per l'utente IAM.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario [specificare un principale](#) in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Elenchi di controllo degli accessi (ACLs)

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioni ACLs, consulta la [panoramica della lista di controllo degli accessi \(ACL\)](#) nella Amazon Simple Storage Service Developer Guide.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- **Limiti delle autorizzazioni:** un limite delle autorizzazioni è una funzionalità avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal` sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente IAM.
- **Politiche di controllo del servizio (SCPs):** SCPs sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in AWS Organizations. AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più di proprietà dell'Account AWS azienda. Se abiliti tutte le funzionalità di un'organizzazione, puoi applicare le politiche di controllo del servizio (SCPs) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità presenti negli account dei membri, inclusa ciascuna di esse. Utente root dell'account AWS Per ulteriori informazioni su Organizations and SCPs, consulta [le politiche di controllo dei servizi](#) nella Guida AWS Organizations per l'utente.
- **Politiche di controllo delle risorse (RCPs):** RCPs sono politiche JSON che puoi utilizzare per impostare le autorizzazioni massime disponibili per le risorse nei tuoi account senza aggiornare le politiche IAM allegate a ciascuna risorsa di tua proprietà. L'RCP limita le autorizzazioni per le risorse negli account dei membri e può influire sulle autorizzazioni effettive per le identità, incluse le Utente root dell'account AWS, indipendentemente dal fatto che appartengano o meno all'organizzazione. Per ulteriori informazioni su Organizations e RCPs, incluso un elenco di Servizi AWS tale supporto RCPs, vedere [Resource control policies \(RCPs\)](#) nella Guida per l'AWS Organizations utente.
- **Policy di sessione:** le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta [Policy di sessione](#) nella Guida per l'utente IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta la [logica di valutazione delle policy](#) nella IAM User Guide.

AWS politiche gestite per AWS IoT SiteWise

Semplifica l'aggiunta di autorizzazioni a utenti, gruppi e ruoli utilizzando policy AWS gestite anziché scrivere policy da soli. Ci vogliono tempo ed esperienza per [creare policy gestite dai clienti IAM](#) che forniscano autorizzazioni precise al tuo team. Per una configurazione più rapida, prendi in considerazione l'utilizzo delle nostre policy AWS gestite per casi d'uso comuni. Trova le politiche AWS gestite nel tuo AWS account. Per ulteriori informazioni sulle policy gestite da AWS , consulta [Policy gestite da AWS](#) nella Guida per l'utente di IAM.

AWS i servizi si occupano dell'aggiornamento e della manutenzione delle politiche AWS gestite, il che significa che non puoi modificare le autorizzazioni di queste politiche. Occasionalmente, AWS IoT SiteWise può aggiungere autorizzazioni per integrare nuove funzionalità, con un impatto su tutte le identità con la politica allegata. Tali aggiornamenti sono comuni con l'introduzione di nuovi servizi o funzionalità. Tuttavia, le autorizzazioni non vengono mai rimosse, garantendo che le impostazioni rimangano intatte.

Inoltre, AWS supporta politiche gestite per le funzioni lavorative che si estendono su più servizi. Ad esempio, la policy ReadOnlyAccess AWS gestita fornisce l'accesso in sola lettura a tutti i AWS servizi e le risorse. Quando un servizio lancia una nuova funzionalità, AWS aggiunge autorizzazioni di sola lettura per nuove operazioni e risorse. Per un elenco con le descrizioni delle politiche relative alle funzioni lavorative, consulta le [politiche AWS gestite per le funzioni lavorative nella Guida per l'utente IAM](#).

AWS politica gestita: AWSIoT Site WiseReadOnlyAccess

Utilizza la policy AWSIoT SiteWiseReadOnlyAccess AWS gestita per consentire l'accesso in sola lettura a. AWS IoT SiteWise

È possibile allegare la policy AWSIoT SiteWiseReadOnlyAccess alle identità IAM.

Autorizzazioni a livello di servizio

Questa politica fornisce l'accesso in sola lettura a. AWS IoT SiteWise In questa policy non sono incluse altre autorizzazioni di servizio.

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

AWS politica gestita: Wise AWSService RoleForIo TSite

Il AWSServiceRoleForIoTSiteWise ruolo utilizza la AWSServiceRoleForIoTSiteWise politica con le seguenti autorizzazioni. Questa politica:

- Consente di AWS IoT SiteWise implementare gateway SiteWise Edge (che funzionano suAWS IoT Greengrass).
- Consente di eseguire AWS IoT SiteWise la registrazione.
- Consente di AWS IoT SiteWise eseguire una query di ricerca di metadati nel AWS IoT TwinMaker database.

Se utilizzi AWS IoT SiteWise un solo account utente, il AWSServiceRoleForIoTSiteWise ruolo crea la AWSServiceRoleForIoTSiteWise policy nel tuo account IAM e la associa ai ruoli collegati al AWSServiceRoleForIoTSiteWise [servizio](#) per. AWS IoT SiteWise

```
{
  "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"
      ]
    }
  }
}
]

```

}

AWS IoT SiteWise aggiornamenti alle politiche gestite AWS

È possibile visualizzare i dettagli sugli aggiornamenti delle politiche AWS gestite per AWS IoT SiteWise, a partire da quando questo servizio ha iniziato a tenere traccia delle modifiche. Per ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al feed RSS nella pagina della cronologia dei AWS IoT SiteWise documenti.

Modifica	Descrizione	Data
AWSServiceRoleForIoTSiteWise : aggiornamento a una politica esistente	AWS IoT SiteWise ora può eseguire una query di ricerca di metadati sul AWS IoT TwinMaker database.	6 novembre 2023
AWSIoTSiteWiseReadOnlyAccess : aggiornamento a una policy esistente	AWS IoT SiteWise ha aggiunto un nuovo prefisso di policyBatchGet* , che consente di eseguire operazioni di lettura in batch.	16 settembre 2022
AWSIoTSiteWiseReadOnlyAccess : nuova policy	AWS IoT SiteWise ha aggiunto una nuova politica a cui concedere l'accesso in sola lettura. AWS IoT SiteWise	24 novembre 2021
AWS IoT SiteWise ha iniziato a tenere traccia delle modifiche	AWS IoT SiteWise ha iniziato a tenere traccia delle modifiche per le sue politiche AWS gestite.	24 novembre 2021

Usa ruoli collegati ai servizi per AWS IoT SiteWise

AWS IoT SiteWise utilizza ruoli [collegati ai servizi AWS Identity and Access Management](#) (IAM). Un ruolo collegato ai servizi è un tipo unico di ruolo IAM a cui è collegato direttamente AWS

IoT SiteWise. I ruoli collegati ai servizi sono predefiniti AWS IoT SiteWise e includono tutte le autorizzazioni necessarie al servizio per chiamare altri servizi per tuo conto. AWS

I ruoli collegati ai servizi semplificano la configurazione di includendo automaticamente tutte le autorizzazioni necessarie. AWS IoT SiteWise AWS IoT SiteWise definisce le autorizzazioni dei suoi ruoli collegati ai servizi e, se non diversamente definito, solo può assumerne i ruoli. AWS IoT SiteWise Le autorizzazioni definite includono policy di attendibilità e di autorizzazioni. E quella politica di autorizzazione non può essere associata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato ai servizi solo dopo aver eliminato le risorse correlate. Questo protegge AWS IoT SiteWise le tue risorse perché non puoi rimuovere inavvertitamente l'autorizzazione ad accedere alle risorse.

Per informazioni su altri servizi che supportano i ruoli collegati ai servizi, consulta [AWS i servizi che funzionano con IAM](#) e cerca i servizi con Sì nella colonna Ruolo collegato ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Argomenti

- [Autorizzazioni del ruolo collegato ai servizi per AWS IoT SiteWise](#)
- [Creare un ruolo collegato al servizio per AWS IoT SiteWise](#)
- [Aggiorna un ruolo collegato al servizio per AWS IoT SiteWise](#)
- [Elimina un ruolo collegato al servizio per AWS IoT SiteWise](#)
- [Regioni supportate per i ruoli AWS IoT SiteWise collegati ai servizi](#)
- [Utilizza i ruoli di servizio per AWS IoT SiteWise Monitor](#)

Autorizzazioni del ruolo collegato ai servizi per AWS IoT SiteWise

AWS IoT SiteWise utilizza il ruolo collegato al servizio denominato AWSService RoleForIoTSiteWise. AWS IoT SiteWise utilizza questo ruolo collegato al servizio per distribuire i gateway SiteWise Edge (che funzionano su) ed eseguire la registrazione. AWS IoT Greengrass

Il ruolo AWSServiceRoleForIoTSiteWise collegato al servizio utilizza la policy con le seguenti autorizzazioni. AWSServiceRoleForIoTSiteWise Questa politica:

- Consente di AWS IoT SiteWise implementare gateway SiteWise Edge (che funzionano suAWS IoT Greengrass).

- Consente di eseguire AWS IoT SiteWise la registrazione.
- Consente di AWS IoT SiteWise eseguire una query di ricerca di metadati nel AWS IoT TwinMaker database.

Per ulteriori informazioni sulle azioni consentite in `AWSServiceRoleForIoTSiteWise`, consulta le [politiche AWS gestite per AWS IoT SiteWise](#).

```
{
  "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"
    ]
  }
}
```

È possibile utilizzare i log per monitorare e risolvere i problemi dei gateway Edge. SiteWise Per ulteriori informazioni, consulta [Monitora i log del gateway SiteWise Edge](#).

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato al servizio, configura innanzitutto le autorizzazioni. Per ulteriori informazioni, consulta [Autorizzazioni del ruolo collegato ai servizi](#) nella Guida per l'utente di IAM.

Creare un ruolo collegato al servizio per AWS IoT SiteWise

AWS IoT SiteWise richiede un ruolo collegato al servizio per eseguire determinate azioni e accedere alle risorse per conto dell'utente. Un ruolo collegato al servizio è un tipo unico di ruolo AWS Identity and Access Management (IAM) a cui è collegato direttamente. AWS IoT SiteWise Creando questo ruolo, AWS IoT SiteWise concede le autorizzazioni necessarie per accedere ad altri AWS servizi e risorse necessari per il suo funzionamento, come Amazon S3 per l'archiviazione dei dati AWS IoT o per la comunicazione tra dispositivi.

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando esegui le seguenti operazioni nella AWS IoT SiteWise console, AWS IoT SiteWise crea automaticamente il ruolo collegato al servizio.

- Crea un gateway Greengrass V1.
- Configura l'opzione di registrazione.
- Scegliendo il pulsante di attivazione nel banner di esecuzione della query.

Se elimini questo ruolo collegato ai servizi, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando esegui un'operazione nella AWS IoT SiteWise console, AWS IoT SiteWise crea nuovamente il ruolo collegato al servizio.

Puoi anche utilizzare la console o l'API IAM per creare un ruolo collegato al servizio per AWS IoT SiteWise

- Per farlo nella console IAM, crea un ruolo con la policy `AWSServiceRoleForIoTSiteWise` e instaura una relazione di fiducia con `iotsitewise.amazonaws.com`
- Per farlo utilizzando la AWS CLI nostra API IAM, crea un ruolo con la `arn:aws:iam::aws:policy/aws-service-role/AWSServiceRoleForIoTSiteWise` policy e instaura una relazione di fiducia con `iotsitewise.amazonaws.com`.

Per ulteriori informazioni, consulta [Create a service linked role](#) nella IAM User Guide.

Se elimini il ruolo collegato ai servizi, è possibile utilizzare lo stesso processo per crearlo nuovamente.

Aggiorna un ruolo collegato al servizio per AWS IoT SiteWise

AWS IoT SiteWise non consente di modificare il ruolo collegato al servizio `AWSServiceRoleForIoTSiteWise`. Dopo aver creato un ruolo collegato al servizio, non è possibile modificarne il nome, perché potrebbero farvi riferimento diverse entità. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per maggiori informazioni, consulta [Aggiornare un ruolo collegato al servizio nella IAM](#) User Guide.

Elimina un ruolo collegato al servizio per AWS IoT SiteWise

Se una funzionalità o un servizio che richiede un ruolo collegato al servizio non è più in uso, è consigliabile eliminare il ruolo associato. Questo per evitare di avere un'entità inattiva che non viene monitorata o gestita. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato al servizio prima di poterlo eliminare manualmente.

Note

Se il AWS IoT SiteWise servizio utilizza il ruolo quando si tenta di eliminare le risorse, l'eliminazione potrebbe non riuscire. In questo caso, attendi alcuni minuti e riprova.

Per eliminare AWS IoT SiteWise le risorse utilizzate da AWSService RoleForlo TSite Wise

1. Disabilita la registrazione per AWS IoT SiteWise. Per ulteriori informazioni, consulta [Cambia il tuo livello di registrazione](#)
2. Eliminare tutti i gateway SiteWise Edge attivi.

Per eliminare manualmente il ruolo collegato ai servizi mediante IAM

Usa la console IAM AWS CLI, o l' AWS API per eliminare il ruolo collegato al servizio AWSService RoleForlo TSite Wise. Per ulteriori informazioni, consulta [Eliminare ruoli o profili di istanza nella Guida](#) per l'utente IAM.

Regioni supportate per i ruoli AWS IoT SiteWise collegati ai servizi

AWS IoT SiteWise supporta l'utilizzo di ruoli collegati al servizio in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta [Endpoint e quote per AWS IoT SiteWise](#).

Utilizza i ruoli di servizio per AWS IoT SiteWise Monitor

Un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione [Create a role to delegate permissions to an Servizio AWS](#) nella Guida per l'utente IAM.

Per consentire agli utenti federati del portale SiteWise Monitor di accedere AWS IoT SiteWisealle AWS IAM Identity Center proprie risorse, è necessario assegnare un ruolo di servizio a ciascun portale creato. Il ruolo di servizio deve specificare SiteWise Monitor come entità attendibile e includere la policy [AWSIoTSiteWiseMonitorPortalAccess](#)gestita o definire [autorizzazioni equivalenti](#). Questa policy è gestita AWS e definisce il set di autorizzazioni che SiteWise Monitor utilizza per accedere alle tue risorse AWS IoT SiteWise e a quelle di IAM Identity Center.

Quando crei un portale SiteWise Monitor, devi scegliere un ruolo che consenta agli utenti di quel portale di accedere alle tue risorse AWS IoT SiteWisee a quelle di IAM Identity Center. La AWS IoT SiteWise console può creare e configurare il ruolo per te. Puoi modificare il ruolo in IAM in un secondo momento. Gli utenti del tuo portale avranno problemi a utilizzare i loro portali SiteWise Monitor se rimuovi le autorizzazioni richieste dal ruolo o elimini il ruolo.

 Note

I portali creati prima del 29 aprile 2020 non richiedevano ruoli del servizio. Se hai creato portali prima di tale data, devi collegare ruoli del servizio per continuare a utilizzarli. Per farlo, vai alla pagina Portali nella [AWS IoT SiteWise console](#), quindi scegli Migra tutti i portali per utilizzare i ruoli IAM.

Le sezioni seguenti descrivono come creare e gestire il ruolo del servizio SiteWise Monitor in o in. AWS Management Console AWS Command Line Interface

Indice

- [Autorizzazioni del ruolo di servizio per SiteWise Monitor \(Classic\)](#)
- [Autorizzazioni per ruoli di servizio per SiteWise Monitor \(AI-Aware\)](#)
- [Gestisci il ruolo del servizio SiteWise Monitor \(console\)](#)
 - [Trova il ruolo di servizio di un portale \(console\)](#)
 - [Crea un ruolo SiteWise del servizio Monitor \(AWS IoT SiteWise console\)](#)
 - [Crea un ruolo SiteWise del servizio Monitor \(console IAM\)](#)
 - [Modificare il ruolo di servizio di un portale \(console\)](#)
- [Gestire il ruolo del servizio SiteWise Monitor \(CLI\)](#)
 - [Trova il ruolo di servizio \(CLI\) di un portale](#)
 - [Creare il ruolo del servizio SiteWise Monitor \(CLI\)](#)
- [SiteWise Monitora gli aggiornamenti di AWSIoTSiteWiseMonitorServiceRole](#)

Autorizzazioni del ruolo di servizio per SiteWise Monitor (Classic)

Quando si crea un portale, AWS IoT SiteWise consente di creare un ruolo il cui nome inizia con AWSIoTSiteWiseMonitorServiceRole. Questo ruolo consente agli utenti federati di SiteWise Monitor di accedere alla configurazione del portale, agli asset, ai dati degli asset e ai dati di configurazione di IAM Identity Center.

Ai fini dell'assunzione del ruolo, il ruolo considera attendibile il seguente servizio:

- `monitor.iotsitewise.amazonaws.com`

Il ruolo utilizza la seguente politica di autorizzazioni, che inizia con `AWSIoTSiteWiseMonitorServicePortalPolicy`, per consentire agli utenti di SiteWise Monitor di completare azioni sulle risorse del tuo account. La politica [AWSIoTSiteWiseMonitorPortalAccess](#) gestita definisce autorizzazioni equivalenti.

```
{
  "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"
          ]
        }
      }
    }
  ]
}

```

Per ulteriori informazioni sulle autorizzazioni richieste per gli allarmi, consulta [Imposta le autorizzazioni per gli allarmi relativi agli eventi in AWS IoT SiteWise](#)

Quando un utente del portale accede, SiteWise Monitor crea una [politica di sessione](#) basata sull'intersezione tra il ruolo del servizio e le politiche di accesso dell'utente. Le policy di accesso definiscono il livello di accesso delle identità ai portali e ai progetti. Per ulteriori informazioni sulle autorizzazioni del portale e sulle politiche di accesso, consulta [Amministra i tuoi portali SiteWise Monitor](#) e [CreateAccessPolicy](#)

Autorizzazioni per ruoli di servizio per SiteWise Monitor (AI-Aware)

Quando si crea un portale, AWS IoT SiteWise consente di creare un ruolo il cui nome inizia con `IoTSiteWisePortalRole`. Questo ruolo consente agli utenti federati di SiteWise Monitor di accedere alla configurazione del portale, agli asset, ai dati degli asset e ai dati di configurazione di IAM Identity Center.

Warning

I ruoli di proprietario del progetto e visualizzatore del progetto non sono supportati per SiteWise Monitor (AI-Aware).

Ai fini dell'assunzione del ruolo, il ruolo considera attendibile il seguente servizio:

- `monitor.iotsitewise.amazonaws.com`

Il ruolo utilizza la seguente politica di autorizzazioni, che inizia con lo TSite Wise AIPortal AccessPolicy, per consentire agli utenti di SiteWise Monitor di completare azioni sulle risorse del proprio account.

```
{
  "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": "*"
}
]
}

```

Quando un utente del portale accede, SiteWise Monitor crea una [politica di sessione](#) basata sull'intersezione tra il ruolo del servizio e le politiche di accesso di quell'utente.

Gestisci il ruolo del servizio SiteWise Monitor (console)

Console AWS IoT SiteWise Facilita la gestione del ruolo del servizio SiteWise Monitor per i portali. Al momento della creazione di un portale, la console verifica la presenza di ruoli esistenti adatti all'allegato. Se non ce ne sono disponibili, la console può creare e configurare automaticamente un ruolo di servizio. Per ulteriori informazioni, consulta [Crea un portale in Monitor SiteWise](#).

Argomenti

- [Trova il ruolo di servizio di un portale \(console\)](#)
- [Crea un ruolo SiteWise del servizio Monitor \(AWS IoT SiteWise console\)](#)
- [Crea un ruolo SiteWise del servizio Monitor \(console IAM\)](#)
- [Modificare il ruolo di servizio di un portale \(console\)](#)

Trova il ruolo di servizio di un portale (console)

Utilizza i seguenti passaggi per trovare il ruolo di servizio associato a un portale SiteWise Monitor.

Per individuare il ruolo del servizio di un portale

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione a sinistra scegliere Portals (Portali).
3. Scegliere il portale per il quale si desidera individuare il ruolo del servizio.

Il ruolo collegato al portale viene visualizzato in Permissions (Autorizzazioni), Service role (Ruolo del servizio).

Crea un ruolo SiteWise del servizio Monitor (AWS IoT SiteWise console)

Quando crei un portale SiteWise Monitor, puoi creare un ruolo di servizio per il tuo portale. Per ulteriori informazioni, consulta [Crea un portale in Monitor SiteWise](#).

È inoltre possibile creare un ruolo di servizio per un portale esistente nella AWS IoT SiteWise console. Questo sostituisce il ruolo di servizio esistente del portale.

Per creare un ruolo del servizio per un portale esistente

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione scegli Portali.
3. Scegliere il portale per il quale si desidera creare un nuovo ruolo del servizio.
4. In Portal details (Dettagli portale), scegliere Edit (Modifica).
5. In Permissions (Autorizzazioni), scegliere Create and use a new service role (Crea e utilizza un nuovo ruolo del servizio) dall'elenco.
6. Immettere un nome per il nuovo ruolo.
7. Scegli Save (Salva).

Crea un ruolo SiteWise del servizio Monitor (console IAM)

Puoi creare un ruolo di servizio dal modello di ruolo di servizio nella console IAM. Questo modello di ruolo include la policy [AWSIoTSiteWiseMonitorPortalAccess](#) gestita e specifica SiteWise Monitor come entità affidabile.

Per creare un ruolo di servizio dal modello di ruolo di servizio del portale

1. Passare alla [IAM console](#) (Console IAM).

2. Nel pannello di navigazione, seleziona Roles (Ruoli).
3. Selezionare Create role (Crea ruolo).
4. In Scegli un caso d'uso, scegli IoT SiteWise.
5. In Seleziona il tuo caso d'uso, scegli IoT SiteWise Monitor - Portal.
6. Scegliere Next: Permissions (Successivo: Autorizzazioni).
7. Scegliere Next: Tags (Successivo: Tag).
8. Scegliere Next:Review (Successivo: Rivedi).
9. Immettete un nome di ruolo per il nuovo ruolo di servizio.
10. Scegliere Crea ruolo.

Modificare il ruolo di servizio di un portale (console)

Utilizzare la procedura seguente per scegliere un ruolo del servizio di SiteWise monitoraggio diverso per un portale.

Per modificare il ruolo del servizio di un portale

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione scegli Portali.
3. Scegliere il portale per il quale si desidera modificare il ruolo del servizio.
4. In Portal details (Dettagli portale), scegliere Edit (Modifica).
5. In Permissions (Autorizzazioni), scegliere Use an existing role (Utilizza un ruolo esistente).
6. Scegliere un ruolo esistente da collegare al portale.
7. Scegli Save (Salva).

Gestire il ruolo del servizio SiteWise Monitor (CLI)

È possibile utilizzare il AWS CLI per le seguenti attività di gestione dei ruoli del servizio del portale:

Argomenti

- [Trova il ruolo di servizio \(CLI\) di un portale](#)
- [Creare il ruolo del servizio SiteWise Monitor \(CLI\)](#)

Trova il ruolo di servizio (CLI) di un portale

Per trovare il ruolo di servizio associato a un portale di SiteWise monitoraggio, esegui il comando seguente per elencare tutti i portali nella regione corrente.

```
aws iotsitewise list-portals
```

L'operazione restituisce una risposta contenente i riepiloghi del portale nel formato seguente.

```
{
  "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"
    }
  ]
}
```

Puoi anche utilizzare l'[DescribePortal](#) operazione per trovare il ruolo del tuo portale se conosci l'ID del tuo portale.

Creare il ruolo del servizio SiteWise Monitor (CLI)

Utilizza i passaggi seguenti per creare un nuovo ruolo del servizio SiteWise Monitor.

Per creare un ruolo SiteWise del servizio Monitor

1. Crea un ruolo con una politica di fiducia che consenta a SiteWise Monitor di assumere il ruolo. In questo esempio viene creato un ruolo denominato **MySiteWiseMonitorPortalRole** da una policy di attendibilità archiviata in una stringa JSON.

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. Copiare il ruolo ARN dai metadati del ruolo nell'output. Quando si crea un portale, utilizzare questo ARN per associare il ruolo al portale. Per ulteriori informazioni sulla creazione di un portale, [CreatePortal](#) consulta l'AWS IoT SiteWise API Reference.
3. a. Per il SiteWise monitor (versione classica): allega la `AWSIoTSiteWiseMonitorPortalAccess` policy al ruolo o allega una policy che definisca autorizzazioni equivalenti.

```
aws iam attach-role-policy --role-name MySiteWiseMonitorPortalRole --policy-arn arn:aws:iam::aws:policy/service-role/AWSIoTSiteWiseMonitorPortalAccess
```

- b. Per il SiteWise monitor (compatibile con l'intelligenza artificiale): collega la `IoTSiteWiseAIPortalAccessPolicy` policy al ruolo o allega una policy che definisca autorizzazioni equivalenti. Ad esempio, crea una policy con autorizzazioni di accesso al portale. L'esempio seguente crea una politica denominata `MySiteWiseMonitorPortalAccess`.

```
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": ""
```

```
    }
  ]
}'
```

Per collegare un ruolo del servizio a un portale esistente

1. Per recuperare i dettagli esistenti del portale, eseguire il comando seguente. Sostituisci *portal-id* con l'ID del portale.

```
aws iotsitewise describe-portal --portal-id portal-id
```

L'operazione restituisce una risposta contenente i dettagli del portale nel formato seguente.

```
{
  "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. Per collegare un ruolo del servizio a un portale, eseguire il comando seguente. Sostituisci *role-arn* con il ruolo di servizio ARN e sostituisci i parametri rimanenti con i valori esistenti del portale.

```
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 Monitora gli aggiornamenti di AWS IoT SiteWise MonitorServiceRole

È possibile visualizzare i dettagli sugli aggiornamenti di AWS IoT SiteWise MonitorServiceRole per SiteWise Monitor, a partire da quando questo servizio ha iniziato a tenere traccia delle modifiche. Per ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al feed RSS nella pagina della cronologia dei AWS IoT SiteWise documenti.

Modifica	Descrizione	Data
AWS IoT SiteWise MonitorPortalAccess — Politica aggiornata	AWS IoT SiteWise ha aggiornato la politica di AWS IoT SiteWise MonitorPortalAccess gestione per la funzionalità degli allarmi.	27 maggio 2021
AWS IoT SiteWise ha iniziato a tenere traccia delle modifiche	AWS IoT SiteWise ha iniziato a tenere traccia delle modifiche relative al suo ruolo di servizio.	15 dicembre 2020

Imposta le autorizzazioni per gli allarmi relativi agli eventi in AWS IoT SiteWise

Quando utilizzi un modello di AWS IoT Events allarme per monitorare la proprietà di un AWS IoT SiteWise asset, devi disporre delle seguenti autorizzazioni IAM:

- Un ruolo AWS IoT Events di servizio che consente di AWS IoT Events inviare dati a AWS IoT SiteWise. Per ulteriori informazioni, consulta la sezione [Gestione delle identità e degli accessi AWS IoT Events nella Guida per](#) gli AWS IoT Events sviluppatori.
- È necessario disporre delle seguenti AWS IoT SiteWise autorizzazioni di azione:
`ioticsitewise:DescribeAssetModel`

`iotsitewise:UpdateAssetModelPropertyRouting`. Queste autorizzazioni consentono di AWS IoT SiteWise inviare i valori delle proprietà degli asset ai modelli di AWS IoT Events allarme.

Per ulteriori informazioni, consulta le [politiche basate sulle risorse](#) nella Guida per l'utente IAM.

Autorizzazioni di azione richieste

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni. L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio.

Prima di definire un modello di AWS IoT Events allarme, è necessario concedere le seguenti autorizzazioni che consentono AWS IoT SiteWise di inviare i valori delle proprietà degli asset al modello di allarme.

- `iotsitewise:DescribeAssetModel`, `iotsitewise:ListAssetModels` — Consente di AWS IoT Events verificare se esiste una proprietà dell'asset.
- `iotsitewise:UpdateAssetModelPropertyRouting` — Consente di AWS IoT SiteWise creare automaticamente abbonamenti che consentono AWS IoT SiteWise di inviare dati a AWS IoT Events.

Per ulteriori informazioni sulle azioni AWS IoT SiteWise supportate, vedere [Azioni definite da AWS IoT SiteWise](#) nel Service Authorization Reference.

Example Esempio di politica delle autorizzazioni 1

La seguente politica consente di AWS IoT SiteWise inviare i valori delle proprietà degli asset a qualsiasi modello di AWS IoT Events allarme.

```
{
  "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 Esempio di politica di autorizzazione 2

La seguente politica consente di AWS IoT SiteWise inviare i valori di una proprietà di asset specificata a un modello di AWS IoT Events allarme specificato.

```

{
  "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",
          "iotevents:alarmModelArn": "arn:aws:iotevents:us-
east-1:123456789012:alarmModel/MyAlarmModel"
        }
      }
    }
  ]
}

```

ListInputRoutings Autorizzazione (Facoltativa)

Quando aggiorni o elimini un modello di asset, AWS IoT SiteWise puoi verificare se un modello di allarme in esecuzione AWS IoT Events sta monitorando una proprietà di asset associata a questo modello di asset. Ciò impedisce di eliminare una proprietà dell'asset attualmente utilizzata da un AWS IoT Events allarme. Per abilitare questa funzionalità AWS IoT SiteWise, è necessario disporre dell'`iotevents:ListInputRoutings` autorizzazione. Questa autorizzazione consente di AWS IoT SiteWise effettuare chiamate all'operazione [ListInputRoutings](#) API supportata da AWS IoT Events.

Note

Ti consigliamo vivamente di aggiungere l'`ListInputRoutings` autorizzazione.

Example Esempio di politica sulle autorizzazioni

La seguente politica consente di aggiornare ed eliminare i modelli di asset e di utilizzare l'`ListInputRoutings` API in AWS IoT SiteWise.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:UpdateAssetModel",
        "iotsitewise>DeleteAssetModel",
        "iotevents:ListInputRoutings"
      ]
    }
  ]
}

```

```

    ],
    "Resource": "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/*"
  }
]
}

```

Autorizzazioni richieste per Monitor SiteWise

Se desideri utilizzare la funzionalità di allarme nei portali SiteWise Monitor, devi aggiornare il [ruolo del servizio SiteWise Monitor](#) con la seguente politica:

```

{
  "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"
            ]
        }
    }
}
]
}

```

Prevenzione sostitutiva confusa tra diversi servizi in AWS IoT SiteWise

Con "confused deputy" si intende un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire una certa operazione può costringere un'entità con più privilegi a eseguire tale operazione. Nel AWS, l'impersonificazione intersettoriale può portare al confuso problema del vice. La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua una chiamata a un altro servizio (il servizio chiamato). Il servizio chiamante può essere manipolato per utilizzare le proprie autorizzazioni e agire sulle risorse di un altro cliente, a cui normalmente non avrebbe accesso. Per evitare ciò, AWS fornisce alcuni strumenti che consentono di proteggere i dati per tutti i servizi che dispongono di principali del servizio a cui è stato consentito l'accesso alle risorse del tuo account.

Si consiglia di utilizzare [aws:SourceArn](#) le chiavi di contesto della condizione [aws:SourceAccount](#) globale nelle politiche delle risorse per limitare le autorizzazioni che AWS

IoT SiteWise forniscono un altro servizio alla risorsa. Se il valore `aws:SourceArn` non contiene l'ID account, ad esempio il nome della risorsa Amazon (ARN) di un bucket Amazon S3, è necessario utilizzare entrambe le chiavi di contesto delle condizioni globali per limitare le autorizzazioni. Se si utilizzano entrambe le chiavi di contesto delle condizioni globali e il valore `aws:SourceArn` contiene l'ID account, il valore `aws:SourceAccount` e l'account nel valore `aws:SourceArn` deve utilizzare lo stesso ID account nella stessa dichiarazione di policy.

- Utilizzare `aws:SourceArn` se si desidera consentire l'associazione di una sola risorsa all'accesso tra servizi.
- Utilizza `aws:SourceAccount` se desideri consentire l'associazione di qualsiasi risorsa in tale account all'uso tra servizi.

Il valore di `aws:SourceArn` deve essere la risorsa AWS IoT SiteWise del cliente associata alla `sts:AssumeRole` richiesta.

Il modo più efficace per proteggersi dal problema "confused deputy" è quello di usare la chiave di contesto della condizione globale `aws:SourceArn` con l'ARN completo della risorsa. Se non si conosce l'ARN completo della risorsa o se si sta specificando più risorse, utilizzare la chiave di condizione del contesto globale `aws:SourceArn` con caratteri speciali (*) per le parti sconosciute dell'ARN. Ad esempio, `arn:aws:servicename::123456789012:*`.

Example — Vice Prevenzione confusa

L'esempio seguente mostra come utilizzare le chiavi di contesto `aws:SourceArn` e `aws:SourceAccount` global condition AWS IoT SiteWise per prevenire il problema del confuso vice.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "iotsitewise.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Resource": [
      "arn:aws:iotsitewise::ResourceName/*"
    ],
    "Condition": {
      "ArnLike": {
```

```
    "aws:SourceArn": "arn:aws:iotsitewise:*:123456789012:*"
  },
  "StringEquals": {
    "aws:SourceAccount": "123456789012"
  }
}
```

Risolvi i problemi relativi all' AWS IoT SiteWise identità e all'accesso

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con AWS IoT SiteWise and AWS Identity and Access Management (IAM).

Argomenti

- [Non sono autorizzato a eseguire alcuna azione in AWS IoT SiteWise](#)
- [Non sono autorizzato a eseguire iam:PassRole](#)
- [Voglio consentire a persone esterne al mio AWS account di accedere alle mie AWS IoT SiteWise risorse](#)

Non sono autorizzato a eseguire alcuna azione in AWS IoT SiteWise

Se ti AWS Management Console dice che non sei autorizzato a eseguire un'azione, devi contattare l'amministratore per ricevere assistenza. L'amministratore è la persona da cui si sono ricevuti il nome utente e la password.

L'errore di esempio seguente si verifica quando l'utente mateojackson IAM tenta di utilizzare la console per visualizzare i dettagli su una risorsa ma non dispone `iotsitewise:DescribeAsset` delle autorizzazioni.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
    iotsitewise:DescribeAsset on resource: a1b2c3d4-5678-90ab-cdef-2222EXAMPLE
```

In questo caso, Mateo chiede al suo amministratore di aggiornare le sue policy per poter accedere alla risorsa asset con ID `a1b2c3d4-5678-90ab-cdef-2222EXAMPLE` mediante l'operazione `iotsitewise:DescribeAsset`.

Non sono autorizzato a eseguire **iam:PassRole**

Se ricevi un errore che indica che non sei autorizzato a eseguire l'operazione `iam:PassRole`, le tue policy devono essere aggiornate per poter passare un ruolo a AWS IoT SiteWise.

Alcuni Servizi AWS consentono di passare un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un utente IAM denominato `marymajor` cerca di utilizzare la console per eseguire un'operazione in AWS IoT SiteWise. Tuttavia, l'operazione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire a persone esterne al mio AWS account di accedere alle mie AWS IoT SiteWise risorse

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se AWS IoT SiteWise supporta queste funzionalità, consulta. [Come AWS IoT SiteWise funziona con IAM](#)
- Per scoprire come fornire l'accesso alle tue risorse attraverso Account AWS le risorse di tua proprietà, consulta [Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà](#) nella IAM User Guide.

- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta [Fornire l'accesso a soggetti Account AWS di proprietà di terze parti](#) nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta [Fornire l'accesso a utenti autenticati esternamente \(Federazione delle identità\)](#) nella Guida per l'utente IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

Convalida della conformità per AWS IoT SiteWise

AWS IoT SiteWise non rientra nell'ambito di alcun programma di AWS conformità.

Per un elenco dei AWS servizi che rientrano nell'ambito di specifici programmi di conformità, vedere [AWS Servizi compresi nell'ambito del programma di conformitàAWS](#) . Per informazioni generali, vedere Programmi di [AWS conformità Programmi](#) di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta [Scaricamento dei report in AWS Artifact](#) .

La vostra responsabilità di conformità durante l'utilizzo AWS IoT SiteWise è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- [Guide rapide su sicurezza e conformità](#) [Guide introduttive](#) implementazione illustrano considerazioni sull'architettura e forniscono passaggi per implementare ambienti di base incentrati sulla sicurezza e la conformità. AWS
- Whitepaper [sull'architettura per la sicurezza e la conformità HIPAA: questo white paper](#) descrive in che modo le aziende possono utilizzare per creare applicazioni conformi all'HIPAA. AWS
- AWS Risorse per [la conformità](#) [Risorse per la conformità](#): questa raccolta di potrebbe riguardare il settore e la località in cui operate.
- [Valutazione delle risorse in base alle regole contenute](#) nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- [AWS Security Hub](#)— Questo AWS servizio offre una visione completa dello stato di sicurezza dell'utente, AWS che consente di verificare la conformità agli standard e alle best practice del settore della sicurezza.

- [Dieci regole d'oro di sicurezza per le soluzioni Industrial IoT](#): questo post sul blog introduce dieci regole d'oro che aiutano a proteggere i sistemi di controllo industriale (ICS), l'Internet of Things (IIoT) industriale e gli ambienti cloud.
- [Best practice di sicurezza per la produzione OT](#): questo white paper descrive le migliori pratiche di sicurezza per progettare, implementare e progettare questi carichi di lavoro di produzione ibrida on-premise per il cloud. AWS

Resilienza in AWS IoT SiteWise

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

AWS IoT SiteWise è completamente gestito e utilizza AWS servizi durevoli e ad alta disponibilità, come Amazon S3 e Amazon EC2. Per garantire la disponibilità in caso di interruzione della zona di disponibilità, AWS IoT SiteWise opera su più zone di disponibilità.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, vedere [AWS Global Infrastructure](#).

Oltre all'infrastruttura AWS globale, AWS IoT SiteWise offre diverse funzionalità per supportare le esigenze di resilienza e backup dei dati:

- È possibile pubblicare gli aggiornamenti dei valori delle proprietà AWS IoT Core tramite messaggi MQTT, quindi configurare le regole per agire su tali dati. Con questa funzionalità, puoi eseguire il backup dei dati in altri AWS servizi come Amazon S3 e Amazon DynamoDB. Per ulteriori informazioni, consultare [Interagisci con altri AWS servizi](#) e [Esporta i dati su Amazon S3 con notifiche sulle proprietà degli asset](#).
- Puoi utilizzarla AWS IoT SiteWise Get * APIs per recuperare e fare il backup dei dati storici sulle proprietà degli asset. Per ulteriori informazioni, consulta [Interroga i valori storici delle proprietà degli asset in AWS IoT SiteWise](#).
- È possibile utilizzare il AWS IoT SiteWise Describe* APIs per recuperare le definizioni delle risorse, ad esempio risorse e modelli. È possibile eseguire il backup di queste definizioni e utilizzarle in seguito per ricreare le risorse. Per ulteriori informazioni, consulta la [Documentazione di riferimento delle API di AWS IoT SiteWise](#).

Sicurezza dell'infrastruttura in AWS IoT SiteWise

In quanto servizio gestito, AWS IoT SiteWise è protetto dalla sicurezza della rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta [AWS Cloud Security](#). Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Utilizzate chiamate API AWS pubblicate per accedere AWS IoT SiteWise attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare [AWS Security Token Service](#) (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

SiteWise I gateway edge, che funzionano su AWS IoT Greengrass, utilizzano certificati X.509 e chiavi crittografiche per connettersi e autenticarsi al cloud. AWS Per ulteriori informazioni, consulta [Autenticazione e autorizzazione del dispositivo nella Guida per gli AWS IoT Greengrass sviluppatori](#).AWS IoT Greengrass Version 1

Analisi della configurazione e delle vulnerabilità in AWS IoT SiteWise

I parchi istanze IoT possono essere costituiti da un numero elevato di dispositivi con funzionalità diverse, usati per lunghi periodi di tempo e distribuiti in varie aree geografiche. Queste caratteristiche rendono la configurazione di un parco istanze complessa e soggetta a errori. Poiché i dispositivi di solito hanno potenza di elaborazione, memoria e spazio di archiviazione limitati, non sempre possono supportare la crittografia e altre misure di sicurezza. I dispositivi, inoltre, usano spesso software con vulnerabilità note. La combinazione di questi fattori rende i parchi istanze IoT un facile bersaglio per gli hacker e rende difficile la protezione continuativa di un parco istanze di dispositivi.

AWS IoT Device Defender affronta queste sfide fornendo strumenti per identificare i problemi di sicurezza e le deviazioni dalle migliori pratiche. AWS IoT Device Defender Utilizzatelo per analizzare,

controllare e monitorare i dispositivi connessi per rilevare comportamenti anomali e mitigare i rischi per la sicurezza. AWS IoT Device Defender può controllare le flotte di dispositivi per garantire che aderiscano alle migliori pratiche di sicurezza e rilevi comportamenti anomali sui dispositivi. In questo modo è possibile applicare politiche di sicurezza coerenti in tutto il parco AWS IoT dispositivi e rispondere rapidamente quando i dispositivi vengono compromessi. Per ulteriori informazioni, consulta [Che cos'è AWS IoT Device Defender?](#) nella Guida per gli sviluppatori di AWS IoT Device Defender .

Se utilizzi i gateway SiteWise Edge per importare dati nel servizio, è tua responsabilità configurare e gestire l'ambiente del gateway SiteWise Edge. Questa responsabilità include l'aggiornamento alle versioni più recenti del software di sistema, AWS IoT Greengrass del software e del connettore del gateway SiteWise Edge. AWS IoT SiteWise Per ulteriori informazioni, consulta [Configurare il AWS IoT Greengrass core](#) nella Guida per gli AWS IoT Greengrass Version 1 sviluppatori e. [Gestione dei gateway SiteWise Edge](#)

Endpoint VPC per AWS IoT SiteWise

Un endpoint VPC di interfaccia stabilisce una connessione privata tra il tuo cloud privato virtuale (VPC) e. AWS IoT SiteWise [AWS PrivateLink](#) alimenta gli endpoint di interfaccia, abilitando l'accesso privato alle operazioni delle API. AWS IoT SiteWise È possibile ignorare la necessità di un gateway Internet, un dispositivo NAT, una connessione VPN o. AWS Direct Connect Le istanze nel tuo VPC non necessitano di indirizzi IP pubblici per comunicare AWS IoT SiteWise con le operazioni API. Il traffico tra il tuo VPC e AWS IoT SiteWise non esce dalla AWS rete.

Ogni endpoint di interfaccia è rappresentato da una o più [interfacce di rete elastiche](#) nelle sottoreti.

Prima di configurare un endpoint VPC di interfaccia per AWS IoT SiteWise, consulta la sezione [Accedere a un AWS servizio utilizzando un endpoint VPC di interfaccia](#) nella Guida. AWS PrivateLink

Operazioni API per endpoint VPC in AWS IoT SiteWise

AWS IoT SiteWise supporta l'effettuazione di chiamate alle seguenti operazioni AWS IoT SiteWise API dal tuo VPC:

- Per tutte le operazioni dell'API del piano dati, utilizza il seguente endpoint: Sostituisci *region* con il tuo Regione AWS

```
data.iotsitewise.region.amazonaws.com
```

Le operazioni dell'API del piano dati includono quanto segue:

- [BatchGetAssetPropertyValue](#)
 - [BatchGetAssetPropertyValueHistory](#)
 - [BatchPutAssetPropertyValue](#)
 - [GetAssetPropertyAggregates](#)
 - [GetAssetPropertyValue](#)
 - [GetAssetPropertyValueHistory](#)
 - [GetInterpolatedAssetPropertyValues](#)
- Per le operazioni API del piano di controllo utilizzate per gestire i modelli di asset, gli asset, i gateway SiteWise Edge, i tag e le configurazioni degli account, utilizzate il seguente endpoint. Sostituisci *region* con il tuo Regione AWS.

```
api.iotsitewise.region.amazonaws.com
```

Le operazioni API del piano di controllo supportate includono quanto segue:

- [AssociateAssets](#)
- [CreateAsset](#)
- [CreateAssetModel](#)
- [DeleteAsset](#)
- [DeleteAssetModel](#)
- [DeleteDashboard](#)
- [DescribeAsset](#)
- [DescribeAssetModel](#)
- [DescribeAssetProperty](#)
- [DescribeDashboard](#)
- [DescribeLoggingOptions](#)
- [DisassociateAssets](#)
- [ListAssetModels](#)
- [ListAssetRelationships](#)
- [ListAssets](#)

- [PutLoggingOptions](#)
- [UpdateAsset](#)
- [UpdateAssetModel](#)
- [UpdateAssetProperty](#)
- [CreateGateway](#)
- [DeleteGateway](#)
- [DescribeDefaultEncryptionConfiguration](#)
- [DescribeGateway](#)
- [DescribeGatewayCapabilityConfiguration](#)
- [DescribeStorageConfiguration](#)
- [ListGateways](#)
- [ListTagsForResource](#)
- [UpdateGateway](#)
- [UpdateGatewayCapabilityConfiguration](#)
- [PutDefaultEncryptionConfiguration](#)
- [PutStorageConfiguration](#)
- [TagResource](#)
- [UntagResource](#)

 Note

L'endpoint VPC dell'interfaccia per le operazioni dell'API del piano di controllo attualmente non supporta l'effettuazione di chiamate alle seguenti operazioni dell'API SiteWise Monitor:

- [BatchAssociateProjectAssets](#)
- [BatchDisassociateProjectAssets](#)
- [CreateAccessPolicy](#)
- [CreateDashboard](#)
- [CreatePortal](#)
- [CreateProject](#)
- [DeleteAccessPolicy](#)

- [DeleteProject](#)
- [DescribeAccessPolicy](#)
- [DescribePortal](#)
- [DescribeProject](#)
- [ListAccessPolicies](#)
- [ListDashboards](#)
- [ListPortals](#)
- [ListProjects](#)
- [ListProjectAssets](#)
- [UpdateAccessPolicy](#)
- [UpdateDashboard](#)
- [UpdatePortal](#)
- [UpdateProject](#)

Creazione di un endpoint VPC dell'interfaccia per AWS IoT SiteWise

Per creare un endpoint VPC per il AWS IoT SiteWise servizio, usa la console Amazon VPC o (). AWS Command Line Interface AWS CLIPer ulteriori informazioni, consulta [Accedere a un AWS servizio utilizzando un endpoint VPC di interfaccia nella Guida](#).AWS PrivateLink

Crea un endpoint VPC per AWS IoT SiteWise utilizzando uno dei seguenti nomi di servizio:

- Per le operazioni dell'API del piano dati, utilizza il seguente nome di servizio:

```
com.amazonaws.region.iotsitewise.data
```

- Per le operazioni dell'API del piano di controllo, utilizzare il seguente nome di servizio:

```
com.amazonaws.region.iotsitewise.api
```

Accesso AWS IoT SiteWise tramite un endpoint VPC di interfaccia

Quando crei un endpoint di interfaccia, generiamo nomi host DNS specifici per endpoint con cui puoi comunicare. AWS IoT SiteWise L'opzione DNS privato è abilitata per impostazione predefinita. Per ulteriori informazioni, consulta [Using private hosted zones](#) nella Amazon VPC User Guide.

Se abiliti il DNS privato per l'endpoint, puoi effettuare richieste API AWS IoT SiteWise tramite uno dei seguenti endpoint VPC.

- Per le operazioni dell'API del piano dati, utilizza il seguente endpoint: Sostituisci con il tuo. *region* Regione AWS

```
data.iotsitewise.region.amazonaws.com
```

- Per le operazioni dell'API del piano di controllo, utilizza il seguente endpoint: Sostituisci *region* con il tuo. Regione AWS

```
api.iotsitewise.region.amazonaws.com
```

Se disabiliti il DNS privato per l'endpoint, devi fare quanto segue per accedere AWS IoT SiteWise tramite l'endpoint:

1. Specificare l'URL dell'endpoint VPC nelle richieste API.

- Per le operazioni dell'API del piano dati, utilizza il seguente URL dell'endpoint. Sostituisci *vpc-endpoint-id* e *region* con l'ID e la regione dell'endpoint VPC.

```
vpc-endpoint-id.data.iotsitewise.region.vpce.amazonaws.com
```

- Per le operazioni dell'API del piano di controllo, utilizza il seguente URL dell'endpoint. Sostituisci *vpc-endpoint-id* e *region* con l'ID e la regione dell'endpoint VPC.

```
vpc-endpoint-id.api.iotsitewise.region.vpce.amazonaws.com
```

2. Disabilita l'iniezione del prefisso dell'host. Aggiungi AWS CLI e AWS SDKs anteposti all'endpoint del servizio vari prefissi host quando chiami ciascuna operazione API. Questa funzionalità causa la generazione AWS SDKs di AWS CLI e URLs che non sono validi per AWS IoT SiteWise quando si specifica un endpoint VPC.

 Important

Non è possibile disabilitare l'iniezione del prefisso host in o in. AWS CLI AWS Strumenti per PowerShell Ciò significa che se disabiliti il DNS privato, non puoi utilizzare questi strumenti per accedere AWS IoT SiteWise tramite l'endpoint VPC. Abilita il DNS privato per utilizzare AWS CLI o per accedere tramite AWS IoT SiteWise l' AWS Strumenti per PowerShell endpoint.

Per ulteriori informazioni su come disabilitare l'iniezione del prefisso host in AWS SDKs, consulta le seguenti sezioni della documentazione per ogni SDK:

- [AWS SDK per C++](#)
- [AWS SDK per Go](#)
- [AWS SDK per Go v2](#)
- [AWS SDK per Java](#)
- [AWS SDK for Java 2.x](#)
- [AWS SDK per JavaScript](#)
- [AWS SDK per .NET](#)
- [AWS SDK per PHP](#)
- [AWS SDK per Python \(Boto3\)](#)
- [AWS SDK per Ruby](#)

Per ulteriori informazioni, consulta [Accedere a un AWS servizio utilizzando un endpoint VPC di interfaccia nella Guida](#).AWS PrivateLink

Crea una policy per gli endpoint VPC per AWS IoT SiteWise

È possibile allegare un criterio all'endpoint VPC che controlla l'accesso all' AWS IoT SiteWise. Questa policy specifica le informazioni riportate di seguito:

- Il principale che può eseguire operazioni.
- Le operazioni che possono essere eseguite.
- Le risorse su cui è possibile eseguire le operazioni.

Per ulteriori informazioni, consulta [Controllo degli accessi agli endpoint VPC tramite le policy degli endpoint](#) nella Guida per l'utente di Amazon VPC.

Esempio: policy degli endpoint VPC per le azioni AWS IoT SiteWise

Di seguito è riportato un esempio di policy sugli endpoint per AWS IoT SiteWise. Se associata a un endpoint, questa policy concede l'accesso alle AWS IoT SiteWise azioni elencate all'utente *iotsitewiseadmin* nell'AWS account *123456789012* sulla risorsa specificata.

```
{
  "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"
        ]
      }
    }
  ]
}
```

Best practice di sicurezza per AWS IoT SiteWise

Questo argomento contiene le best practice di sicurezza per AWS IoT SiteWise.

Usa le credenziali di autenticazione sui tuoi server OPC UA

Richiedi le credenziali di autenticazione per connetterti ai tuoi server OPC UA. A questo scopo, consulta la documentazione relativa ai server. Quindi, per consentire al gateway SiteWise Edge di connettersi ai server OPC UA, aggiungete i segreti di autenticazione del server SiteWise al gateway Edge. Per ulteriori informazioni, consulta [Configurare l'autenticazione dell'origine dati per SiteWise Edge](#).

Utilizzate modalità di comunicazione crittografate per i vostri server OPC UA

Scegliete una modalità di sicurezza dei messaggi crittografata e non obsoleta quando configurate le sorgenti OPC UA per il vostro gateway Edge. SiteWise Questo aiuta a proteggere i dati industriali durante il trasferimento dai server OPC UA al gateway Edge. SiteWise Per ulteriori informazioni, consultare [Dati in transito sulla rete locale](#) e [Configura una sorgente OPC UA in SiteWise Edge](#).

Mantieni aggiornati i tuoi componenti

Se utilizzi i gateway SiteWise Edge per importare dati nel servizio, è tua responsabilità configurare e mantenere l'ambiente del gateway Edge. SiteWise Questa responsabilità include l'aggiornamento alle versioni più recenti del software di sistema, del software e dei connettori del gateway. AWS IoT Greengrass

Note

Il connettore AWS IoT SiteWise Edge memorizza segreti sul file system. Questi segreti controllano chi può visualizzare i dati memorizzati nella cache all'interno del gateway SiteWise Edge. Si consiglia vivamente di attivare la crittografia del disco o del file system per il sistema su cui è in esecuzione il SiteWise gateway Edge.

Per informazioni su come aggiornare i componenti della AWS IoT SiteWise console, consulta.

[Cambia la versione dei pacchetti di componenti del gateway SiteWise Edge](#)

Crittografa il file system del tuo gateway SiteWise Edge

Crittografa e proteggi il tuo gateway SiteWise Edge, in modo che i tuoi dati industriali siano al sicuro mentre si spostano attraverso il gateway SiteWise Edge. Se il gateway SiteWise Edge dispone di un modulo di sicurezza hardware, è possibile configurarlo AWS IoT Greengrass per proteggere il gateway SiteWise Edge. Per ulteriori informazioni, consulta [Integrazione della sicurezza hardware](#) nella Guida per gli AWS IoT Greengrass Version 1 sviluppatori. In caso contrario, consulta la documentazione del sistema operativo in uso per informazioni su come crittografare e proteggere il file system.

Accesso sicuro alla configurazione edge

Non condividere la password dell'applicazione Edge Console o la password dell'applicazione SiteWise Monitor. Non inserire questa password in luoghi in cui chiunque possa vederla. Implementa

una corretta politica di rotazione delle password configurando una scadenza appropriata per la password.

Protezione dei dati su Siemens Industrial Edge Management

I dati del dispositivo che scegli di condividere con AWS IoT SiteWise Edge sono determinati negli argomenti Siemens IEM Databus di configurazione. Selezionando gli argomenti da condividere con SiteWise Edge, si condividono i dati a livello di argomento. AWS IoT SiteWise Siemens Industrial Edge Marketplace è un mercato indipendente, separato da AWS. Per proteggere i dati condivisi, l'applicazione SiteWise Edge non verrà eseguita a meno che non venga utilizzata. Siemens Secured Storage Per ulteriori informazioni, consulta [Secure Storage](#), nella Siemens documentazione.

Concedi agli utenti di SiteWise Monitor le autorizzazioni minime possibili

Segui il principio del privilegio minimo utilizzando il set minimo di autorizzazioni relative ai criteri di accesso per gli utenti del portale.

- Quando si crea un portale, definire un ruolo che consenta il set minimo di asset necessario per tale portale. Per ulteriori informazioni, consulta [Utilizza i ruoli di servizio per AWS IoT SiteWise Monitor](#).
- Quando l'utente e gli amministratori del portale creano e condividono progetti, utilizzare il set minimo di asset richiesto per tale progetto.
- Quando un'identità non ha più bisogno di accedere a un portale o a un progetto, rimuovila da quella risorsa. Se tale identità non è più applicabile alla tua organizzazione, eliminala dal tuo archivio di identità.

La best practice basata sul principio minimo si applica anche ai ruoli IAM. Per ulteriori informazioni, consulta [Best practice delle policy](#).

Non esporre informazioni riservate

È necessario impedire la registrazione delle credenziali e di altre informazioni riservate, come le informazioni personali (PII). Ti consigliamo di implementare le seguenti misure di sicurezza anche se l'accesso ai log locali su un gateway SiteWise Edge richiede i privilegi di root e l'accesso ai CloudWatch log richiede le autorizzazioni IAM.

- Non utilizzare informazioni riservate in nomi, descrizioni o proprietà di asset o modelli.
- Non utilizzare informazioni sensibili nel gateway SiteWise Edge o nei nomi delle sorgenti.

- Non utilizzare informazioni riservate in nomi o descrizioni di portali, progetti o pannelli di controllo.

Segui le migliori pratiche di AWS IoT Greengrass sicurezza

Segui le best practice AWS IoT Greengrass di sicurezza per il tuo gateway SiteWise Edge.

Per ulteriori informazioni, consulta le [migliori pratiche di sicurezza](#) nella Guida per gli AWS IoT Greengrass Version 1 sviluppatori.

Consulta anche

- [Le migliori pratiche di sicurezza](#) nella Guida per AWS IoT gli sviluppatori
- [Dieci regole d'oro di sicurezza per le soluzioni Industrial IoT](#)

Accedi e monitora AWS IoT SiteWise

Il monitoraggio è un elemento importante per mantenere l'affidabilità, la disponibilità e le prestazioni delle AWS IoT SiteWise e altre AWS soluzioni. AWS IoT SiteWise supporta i seguenti strumenti di monitoraggio per monitorare il servizio, segnalare quando qualcosa non va e intraprendere azioni automatiche se necessario:

- Amazon CloudWatch monitora AWS le tue risorse e le applicazioni su cui esegui AWS in tempo reale. Raccogli e monitora i parametri, crea dashboard personalizzati e imposta allarmi che ti avvisano o intraprendono azioni quando una determinata metrica raggiunge una determinata soglia. Ad esempio, puoi tenere CloudWatch traccia dell'utilizzo della CPU o di altri parametri delle tue EC2 istanze Amazon e avviare automaticamente nuove istanze quando necessario. Per ulteriori informazioni, consulta la [Amazon CloudWatch User Guide](#).
- Amazon CloudWatch Logs monitora, archivia e accede ai tuoi file di registro da gateway SiteWise Edge e altre fonti CloudTrail. CloudWatch Logs possono monitorare le informazioni nei file di registro e avvisarti quando vengono raggiunte determinate soglie. Puoi inoltre archiviare i dati del log in storage estremamente durevole. Per ulteriori informazioni, consulta la [Amazon CloudWatch Logs User Guide](#).
- AWS CloudTrail acquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo AWS account. Quindi CloudTrail invia i file di log a un bucket Amazon S3 da te specificato. Puoi identificare quali utenti e account hanno effettuato le chiamate AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute le chiamate. Per ulteriori informazioni, consulta la [Guida per l'utente AWS CloudTrail](#).

Argomenti

- [Monitoraggio con Amazon CloudWatch Logs](#)
- [Monitora i log del gateway SiteWise Edge](#)
- [Monitoraggio AWS IoT SiteWise con i CloudWatch parametri di Amazon](#)
- [Registra le chiamate AWS IoT SiteWise API con AWS CloudTrail](#)

Monitoraggio con Amazon CloudWatch Logs

Configura AWS IoT SiteWise per registrare le informazioni in CloudWatch Logs per monitorare e risolvere i problemi del servizio.

Quando usi la AWS IoT SiteWise console, AWS IoT SiteWise crea un ruolo collegato al servizio che consente al servizio di registrare le informazioni per tuo conto. Se non utilizzi la AWS IoT SiteWise console, devi creare manualmente un ruolo collegato al servizio per ricevere i log. Per ulteriori informazioni, consulta [Creare un ruolo collegato al servizio per AWS IoT SiteWise](#).

È necessario disporre di una politica delle risorse che AWS IoT SiteWise consenta di inserire gli eventi di registro negli stream. CloudWatch Per creare e aggiornare una politica delle risorse per CloudWatch Logs, esegui il comando seguente. Sostituisci *logging-policy-name* con il nome della politica da creare.

```
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 supporta anche le chiavi di contesto [aws: SourceArn](#) e [aws: SourceAccount](#) condition. Queste chiavi di contesto delle condizioni sono opzionali.

Per creare o aggiornare una politica delle risorse che AWS IoT SiteWise consenta di inserire solo i log associati alla AWS IoT SiteWise risorsa specificata negli CloudWatch stream, esegui il comando ed esegui quanto segue:

- Sostituisci *logging-policy-name* con il nome della politica da creare.
- Sostituisci *source-ARN* con l'ARN della tua AWS IoT SiteWise risorsa, ad esempio un modello o un asset di asset. Per trovare l'ARN per ogni tipo di AWS IoT SiteWise risorsa, vedere [Tipi di risorse definiti da AWS IoT SiteWise](#) nel Service Authorization Reference.
- Sostituisci *account-ID* con l'ID AWS dell'account associato alla AWS IoT SiteWise risorsa specificata.

```
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\"] } } } ] }"
```

Per impostazione predefinita, AWS IoT SiteWise non registra le informazioni CloudWatch nei registri. Per attivare la registrazione, scegliete un livello di registrazione diverso da Disabilitato (). OFF AWS IoT SiteWise supporta i seguenti livelli di registrazione:

- OFF— La registrazione è disattivata.
- ERROR— Gli errori vengono registrati.
- INFO— Gli errori e i messaggi informativi vengono registrati.

È possibile configurare i gateway SiteWise Edge per registrare le informazioni su Logs through. CloudWatch AWS IoT Greengrass Per ulteriori informazioni, consulta [Monitora i log del gateway SiteWise Edge](#).

È inoltre possibile configurare AWS IoT Core la registrazione delle informazioni nei CloudWatch registri se si sta risolvendo un'azione relativa a una AWS IoT SiteWise regola. Per ulteriori informazioni, consulta [Risolvere i problemi relativi a un' AWS IoT SiteWise azione delle regole](#).

Indice

- [Gestisci l'accesso AWS IoT SiteWise](#)
 - [Trova il tuo livello di registrazione](#)
 - [Cambia il tuo livello di registrazione](#)
- [Esempio: voci dei file di AWS IoT SiteWise registro](#)

Gestisci l'accesso AWS IoT SiteWise

Utilizza la AWS IoT SiteWise console o AWS CLI per le seguenti attività di configurazione della registrazione.

Trova il tuo livello di registrazione

Console

Utilizza la procedura seguente per trovare il livello di registrazione corrente nella console AWS IoT SiteWise .

Per trovare il tuo attuale livello di AWS IoT SiteWise registrazione

1. Passare alla [console AWS IoT SiteWise](#).

2. Nel riquadro di navigazione a sinistra, scegliere Logging options (Opzioni di registrazione).

Lo stato di registrazione corrente viene visualizzato in Logging status (Stato di registrazione). Se la registrazione è attivata, il livello di registrazione corrente viene visualizzato in Livello di dettaglio.

AWS CLI

Esegui il comando seguente per trovare il tuo attuale livello di AWS IoT SiteWise registrazione con. AWS CLI

```
aws iotsitewise describe-logging-options
```

L'operazione restituisce una risposta contenente il livello di registrazione nel formato seguente.

```
{
  "loggingOptions": {
    "level": "String"
  }
}
```

Cambia il tuo livello di registrazione

Utilizzare la procedura seguente per modificare il livello di registrazione nella AWS IoT SiteWise console o in uso. AWS CLI

Console

Per modificare il livello di AWS IoT SiteWise registrazione

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione a sinistra, scegliere Logging options (Opzioni di registrazione).
3. Scegli Modifica.
4. Scegli il livello di verbosità da attivare.
5. Seleziona Salva.

AWS CLI

Esegui il AWS CLI comando seguente per modificare il livello di AWS IoT SiteWise registrazione. Sostituisci *logging-level* con il livello di registrazione desiderato.

```
aws iotsitewise put-logging-options --logging-options level=logging-level
```

Esempio: voci dei file di AWS IoT SiteWise registro

Ogni voce di AWS IoT SiteWise registro include informazioni sull'evento e risorse pertinenti per quell'evento, in modo da poter comprendere e analizzare i dati di registro.

L'esempio seguente mostra una voce CloudWatch Logs che AWS IoT SiteWise registra quando si crea correttamente un modello di asset.

```
{
  "eventTime": "2020-05-05T00:10:22.902Z",
  "logLevel": "INFO",
  "eventType": "AssetModelCreationSuccess",
  "message": "Successfully created asset model.",
  "resources": {
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
  }
}
```

Monitora i log del gateway SiteWise Edge

Puoi configurare il tuo gateway AWS IoT SiteWise Edge per registrare le informazioni su Amazon CloudWatch Logs o sul file system locale.

Argomenti

- [Usa Amazon CloudWatch Logs](#)
- [Usa i log in del servizio AWS IoT SiteWise](#)

Usa Amazon CloudWatch Logs

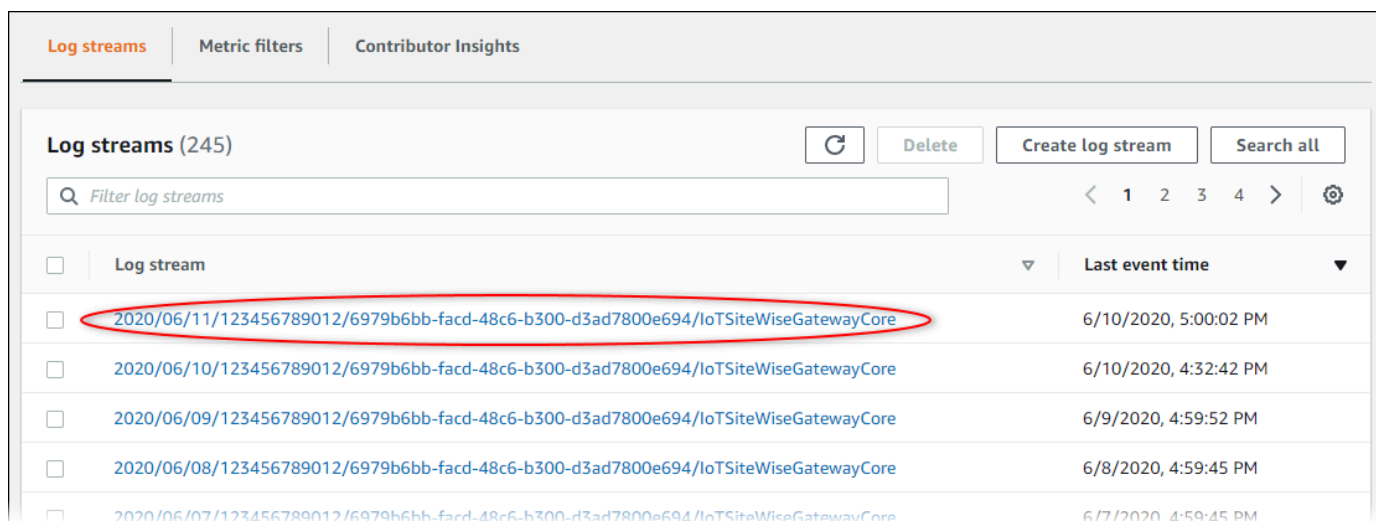
È possibile configurare il gateway SiteWise Edge per inviare i log a CloudWatch Logs. Per ulteriori informazioni, consulta [Abilitare la registrazione per CloudWatch i registri nella Guida per gli sviluppatori.AWS IoT Greengrass Version 2](#)

Per configurare e accedere ai CloudWatch registri (console)

1. Passare alla [console CloudWatch](#).
2. Nel pannello di navigazione, selezionare Log groups (Gruppi di log).
3. È possibile trovare i registri AWS IoT SiteWise dei componenti nei seguenti gruppi di log:
 - `/aws/greengrass/UserComponent/region/aws.iot.SiteWiseEdgeCollector0pcua`— I log del componente del gateway SiteWise Edge che raccoglie i dati dalle fonti OPC UA del gateway SiteWise Edge.
 - `/aws/greengrass/UserComponent/region/aws.iot.SiteWiseEdgePublisher`— I registri del componente del gateway SiteWise Edge su cui pubblica i flussi di dati OPC UA.

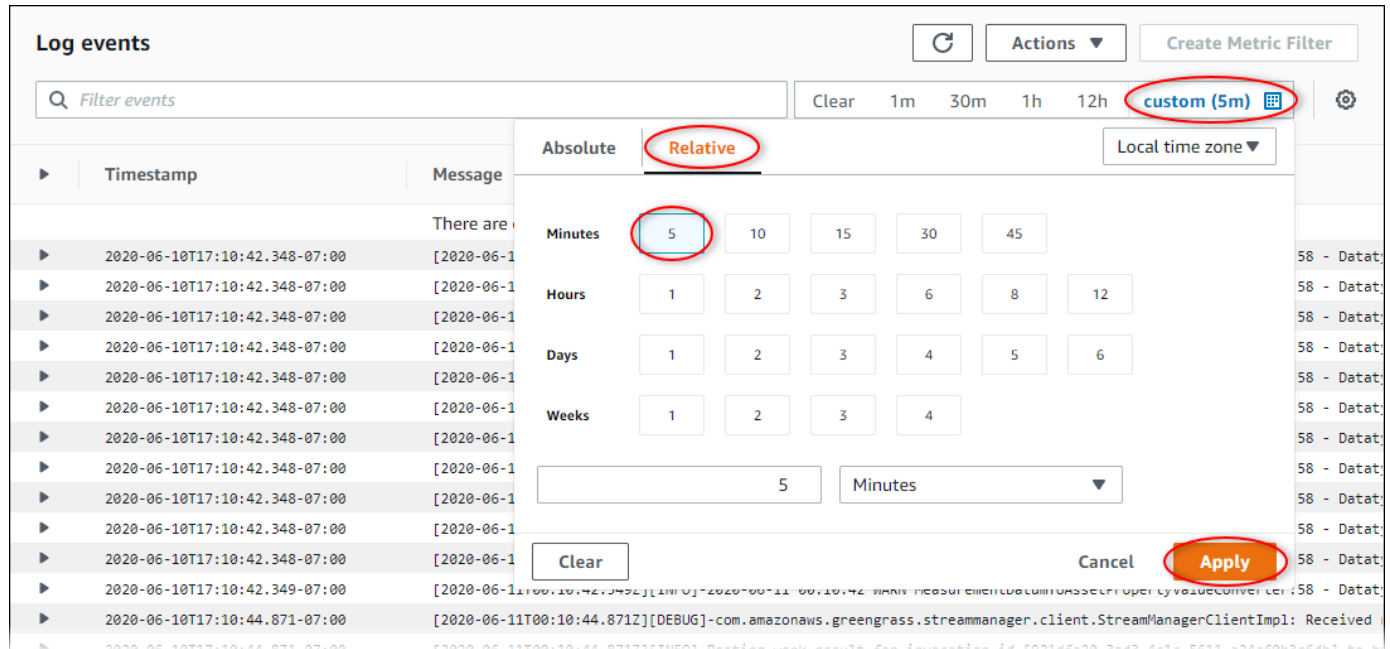
Scegliere il gruppo di log per il debug della funzione.

4. Scegliete un flusso di log il cui nome termini con il nome del gruppo. AWS IoT Greengrass Per impostazione predefinita, CloudWatch visualizza per primo il flusso di log più recente.



5. Per visualizzare i log degli ultimi 5 minuti, effettua le seguenti operazioni:
 - a. Scegli custom (personalizzato) nell'angolo in alto a destra.

- b. Scegli Relative (Relativo).
- c. Scegli 5 minuti.
- d. Scegli Applica.



- 6. (Facoltativo) Per visualizzare meno log, è possibile scegliere 1m nell'angolo in alto a destra.
- 7. Scorrere verso il basso l'elenco delle voci di log per visualizzare le più recenti.

Usa i log in del servizio AWS IoT SiteWise

SiteWise I dispositivi Edge Gateway includono file di registro del servizio per facilitare il debug dei problemi. Le seguenti sezioni ti aiuteranno a trovare e utilizzare i file di registro dei servizi per i componenti AWS IoT SiteWise OPC UA Collector e Publisher. AWS IoT SiteWise

AWS IoT SiteWise File di registro del servizio OPC UA Collector

Il componente AWS IoT SiteWise OPC UA Collector utilizza il seguente file di registro.

Linux

```
/greengrass/v2/logs/aws.iot.SiteWiseEdgeCollectorOpcua.log
```

Windows

```
C:\greengrass\v2\logs\aws.iot.SiteWiseEdgeCollectorOpcua.log
```

Per visualizzare i log di questo componente

- Esegui il seguente comando sul dispositivo principale per visualizzare il file di registro di questo componente in tempo reale. Sostituisci */greengrass/v2* o *C:\greengrass\v2* con il percorso della cartella AWS IoT Greengrass principale.

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 File di registro del servizio Publisher

Il componente AWS IoT SiteWise Publisher utilizza il seguente file di registro.

Linux

```
/greengrass/v2/logs/aws.iot.SiteWiseEdgePublisher.log
```

Windows

```
C:\greengrass\v2\logs\aws.iot.SiteWiseEdgePublisher.log
```

Per visualizzare i registri di questo componente

- Esegui il seguente comando sul dispositivo principale per visualizzare il file di registro di questo componente in tempo reale. Sostituisci */greengrass/v2* o *C:\greengrass\v2* con il percorso della cartella AWS IoT Greengrass principale.

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
```

Monitoraggio AWS IoT SiteWise con i CloudWatch parametri di Amazon

È possibile monitorare AWS IoT SiteWise l'utilizzo CloudWatch, che raccoglie dati grezzi e li elabora in metriche leggibili e quasi in tempo reale. Queste statistiche vengono conservate per un periodo di 15 mesi, per permettere l'accesso alle informazioni storiche e offrire una prospettiva migliore sulle prestazioni del servizio o dell'applicazione web. È anche possibile impostare allarmi che controllano determinate soglie e inviare notifiche o intraprendere azioni quando queste soglie vengono raggiunte. Per ulteriori informazioni, consulta la [Amazon CloudWatch User Guide](#).

AWS IoT SiteWise pubblica le metriche e le dimensioni elencate nelle sezioni seguenti nel namespace. AWS/IoTSiteWise

Tip

AWS IoT SiteWise pubblica le metriche a intervalli di un minuto. Quando visualizzi queste metriche nei grafici nella CloudWatch console, ti consigliamo di scegliere un periodo di 1 minuto. In questo modo puoi visualizzare la risoluzione massima disponibile dei dati di parametro.

Argomenti

- [AWS IoT Greengrass Version 2 metriche del gateway](#)

AWS IoT Greengrass Version 2 metriche del gateway

AWS IoT SiteWise pubblica le metriche del gateway per stream Classic, gateway V2 e gateway V3 compatibili con MQTT. Salvo diversa indicazione, ogni metrica è applicabile a entrambe le versioni di gateway self-hosted. Tutte le metriche del gateway SiteWise Edge vengono pubblicate a intervalli di un minuto.

SiteWise Metriche del gateway Edge

Parametro	Descrizione
<code>Gateway.AvailableMemory</code>	La memoria disponibile di un gateway SiteWise Edge. Unità: byte Dimensione: nessuna
<code>Gateway.AvailableDiskSpace</code>	Lo spazio su disco disponibile di un gateway SiteWise Edge. Unità: byte Dimensione: nessuna
<code>Gateway.CloudConnectivity</code>	Lo stato della connettività cloud di un gateway SiteWise Edge. Unità: nessuna Dimensione: GatewayId
<code>Gateway.CpuUsage</code>	L'utilizzo della CPU di un gateway SiteWise Edge. Unità: percentuale Dimensione: nessuna
<code>Gateway.TotalDiskSpace</code>	Lo spazio totale su disco di un gateway SiteWise Edge.

Parametro	Descrizione
	Unità: byte Dimensione: nessuna
Gateway.TotalMemory	La memoria totale di un gateway SiteWise Edge. Unità: byte Dimensione: nessuna
Gateway.UsedDiskSpace	Lo spazio su disco utilizzato di un gateway SiteWise Edge. Unità: byte Dimensione: nessuna
Gateway.UsedMemory	La memoria utilizzata di un gateway SiteWise Edge. Unità: byte Dimensione: nessuna
Gateway.UsedPercentageDiskSpace	La percentuale di spazio su disco utilizzata di un gateway SiteWise Edge. Unità: byte Dimensione: nessuna
Gateway.UsedPercentageMemory	La percentuale di memoria utilizzata di un gateway SiteWise Edge. Unità: byte Dimensione: nessuna

AWS IoT SiteWise metriche degli editori

Parametro	Descrizione
<code>IoTSiteWisePublisher.ComponentBuildVersion</code>	Questa metrica indica la versione di build del componente SiteWise editore IoT in esecuzione sul gateway. Il valore 1 indica che sul gateway è in esecuzione una versione dell'editore corrispondente alla <code>ComponentBuildVersion</code> dimensione. Unità: 1 Dimensioni: GatewayId, ComponentBuildVersion
<code>IoTSiteWisePublisher.DroppedCount</code>	Il numero di punti dati che vengono eliminati da un gateway SiteWise Edge (GatewayId) e non pubblicati sul cloud, generati ogni minuto. Unità: numero Dimensioni: GatewayId
<code>IoTSiteWisePublisher.Heartbeat</code>	Generato ogni minuto dal gateway Publisher in the SiteWise Edge. Unità: 1 (1 che rappresenta Publisher è in esecuzione e manca il punto dati che rappresenta Publisher non è in esecuzione). Dimensioni: GatewayId
<code>IoTSiteWisePublisher.IsConnectedToMqttBroker</code>	Generato ogni minuto dal gateway Publisher in the SiteWise Edge. Unità: 1 (1 che rappresenta l'editore è collegato a un broker MQTT). Dimensioni: GatewayId

Parametro	Descrizione
<code>IoTSiteWisePublisher.MessageCheckpointPersistenceErrorCount</code>	La metrica indica che il gateway ha rilevato un problema con il file di checkpoint utilizzato per tenere traccia dei dati elaborati dall'editore. Il valore di 1 indica che si è verificato un errore. Unità: nessuna Dimensioni: AccountId, GatewayId
<code>IoTSiteWisePublisher.MqttMessageReceivedSuccessCount</code>	Il numero di messaggi ricevuti con successo dall'editore dal broker MQTT, generati ogni minuto. Unità: numero Dimensioni: GatewayId
<code>IoTSiteWisePublisher.MqttReceivedSuccessBytes</code>	Il numero di byte di dati dei messaggi ricevuti con successo dall'editore dal broker MQTT, generati ogni minuto. Unità: numero Dimensioni: GatewayId
<code>IoTSiteWisePublisher.NumberOfSubscriptionsToMqttBroker</code>	Il numero di argomenti sottoscritti dal Publisher al broker MQTT, generato ogni minuto. Un argomento wild card multilivello viene conteggiato come 1. Unità: numero Dimensioni: GatewayId

Parametro	Descrizione
<code>IoTSiteWisePublisher.NumberOfUniqueMqttTopicsReceived</code>	Il numero di argomenti unici ricevuti dall'editore dal broker MQTT, generati ogni minuto. Unità: numero Dimensioni: GatewayId
<code>IoTSiteWisePublisher.PublisherFailureCount</code>	Il numero di punti dati che un gateway SiteWise Edge (GatewayId) non è riuscito a pubblicare, generato ogni minuto. Unità: numero Dimensioni: GatewayId
<code>IoTSiteWisePublisher.PublisherRejectedCount</code>	Il numero di punti dati che un gateway SiteWise Edge (GatewayId) ha rifiutato dal lato cloud, generato ogni minuto. Unità: numero Dimensioni: GatewayId
<code>IoTSiteWisePublisher.PublisherSuccessCount</code>	Il numero di punti dati che un gateway SiteWise Edge (GatewayId) ha pubblicato con successo sul cloud, generato ogni minuto. Unità: numero Dimensioni: GatewayId
<code>IoTSiteWisePublisher.PublisherToS3FailureCount</code>	Il numero di punti dati che un gateway (GatewayId) non è riuscito a pubblicare su un bucket Amazon S3. Unità: numero Dimensioni: GatewayId

Parametro	Descrizione
<code>IoTSiteWisePublisher.PublisherToS3SuccessCount</code>	Il numero di punti dati che un gateway (GatewayId) ha pubblicato con successo in un bucket Amazon S3. Unità: numero Dimensioni: GatewayId

Metriche del collettore OPC UA

Parametro	Descrizione
<code>OpcUaCollector.ActiveDataStreamCount</code>	Il numero di flussi di dati a cui un gateway SiteWise Edge (gatewayId) si è abbonato per una fonte OPC UA (). sourceName Unità: numero Dimensioni:,, GatewayId SourceName PropertyGroup
<code>OpcUaCollector.ComponentBuildVersion</code> (non disponibile su stream classici, gateway V2)	Questa metrica indica la versione di build del componente del collettore IoT SiteWise OPC UA in esecuzione sul gateway. Il valore 1 indica che il gateway sta eseguendo una versione del collettore corrispondente alla dimensione. ComponentBuildVersion Unità: 1 Dimensioni: GatewayId, ComponentBuildVersion
<code>OpcUaCollector.ConversionErrors</code>	Il numero di punti dati che un gateway SiteWise Edge (gatewayId) ha ricevuto per una fonte OPC UA (sourceName) che ha provocato errori di conversione durante l'invio dei dati

Parametro	Descrizione
	a AWS IoT SiteWise. Questi punti dati non verranno acquisiti da OPC UA Collector. Unità: numero Dimensioni: GatewayId SourceName
<code>OpcUaCollector.Heartbeat</code>	Generato ogni minuto per ogni sorgente OPC UA (sourceName) connessa a un gateway SiteWise Edge (gatewayId). Unità: Numero (1 che rappresenta la sorgente è connessa e 0 che rappresenta la sorgente è disconnessa). Dimensioni: GatewayId, SourceName
<code>OpcUaCollector.IncomingValuesCount</code>	Il numero di punti dati che un gateway SiteWise Edge (gatewayId) ha ricevuto per una fonte OPC UA (sourceName), generato ogni minuto. Unità: numero Dimensioni: GatewayId,, SourceName PropertyGroup
<code>OpcUaCollector.IncomingValuesErrors</code>	Il numero di punti dati che un gateway SiteWise Edge (gatewayId) riceve da una fonte OPC UA (sourceName) che non sono valori validi. Questi punti dati non vengono acquisiti dall'OPC UA Collector, generati ogni minuto. Unità: numero Dimensioni: GatewayId SourceName PropertyGroup

Parametro	Descrizione
<code>OpcUaCollector.IsConnectedToMqttBroker</code> (non disponibile su stream classici, gateway V2)	Generato ogni minuto dal componente di raccolta IoT SiteWise OPC UA nel gateway SiteWise Edge. Unità: 1 (1 che rappresenta il componente del collettore IoT SiteWise OPC UA è collegato a un broker MQTT) Dimensioni: GatewayId
<code>OpcUaCollector.MqttMessagesDroppedCount</code> (non disponibile su stream classici, gateway V2)	Il numero di messaggi MQTT rilasciati dal componente di raccolta IoT SiteWise OPC UA. Unità: numero Dimensioni: GatewayId, SourceName
<code>OpcUaCollector.MqttMessagesPublishedBytes</code> (non disponibile su stream classici, gateway V2)	Il numero di byte di dati dei messaggi MQTT pubblicati con successo dal componente di raccolta IoT SiteWise OPC UA sul broker MQTT. Unità: numero Dimensioni: GatewayId, SourceName
<code>OpcUaCollector.MqttMessagesPublishedCount</code> (non disponibile su stream classici, gateway V2)	Il numero di messaggi MQTT pubblicati con successo dal componente di raccolta IoT SiteWise OPC UA sul broker MQTT. Unità: numero Dimensioni: GatewayId, SourceName

Parametro	Descrizione
<code>OpcUaCollector.NullValueCount</code> (non disponibile su stream classici, gateway V2)	Il numero di valori nulli ricevuti dal componente del collettore IoT SiteWise OPC UA dal server OPC UA. Unità: numero Dimensioni:,, GatewayId SourceName PropertyGroup
<code>OpcUaCollector.NumberOfUniqueMqttTopicsPublished</code> (non disponibile su stream classici, gateway V2)	Il numero di argomenti MQTT unici pubblicati dal raccoglitore IoT SiteWise OPC UA al broker MQTT. Unità: numero Dimensioni:,, GatewayId SourceName

AWS IoT SiteWise metriche del processore

Parametro	Descrizione
<code>Gateway.DataProcessor.IngestionThrottled</code> (non disponibile sui gateway V3 dotati di MQTT)	Il numero di punti dati che sono stati limitati, generati ogni minuto. Unità: numero Dimensioni: ThrottledAt
<code>Gateway.DataProcessor.MeasurementRejected</code> (non disponibile sui gateway V3 abilitati per MQTT)	Il numero di misurazioni che sono state rifiutate, generate ogni minuto. Unità: numero Dimensioni: motivo
<code>Gateway.DataProcessor.MessagesRemaining</code> (non disponibile sui gateway V3 abilitati per MQTT)	Il numero di messaggi rimanenti in uno stream, generato ogni minuto.

Parametro	Descrizione
	Unità: numero
	Dimensioni: StreamName
Gateway.DataProcessor.ProcessingError (non disponibile sui gateway V3 abilitati per MQTT)	Il numero di errori di elaborazione, generati ogni minuto.
	Unità: numero
	Dimensioni: motivo

Registra le chiamate AWS IoT SiteWise API con AWS CloudTrail

AWS IoT SiteWise è integrato con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un utente, un ruolo o un AWS servizio in AWS IoT SiteWise. CloudTrail acquisisce le chiamate API AWS IoT SiteWise come eventi. Le chiamate acquisite includono chiamate dalla AWS IoT SiteWise console e chiamate di codice alle operazioni AWS IoT SiteWise API. Se crei un trail, puoi attivare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per. AWS IoT SiteWise Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare a quale richiesta è stata inviata AWS IoT SiteWise, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni in merito CloudTrail, consulta la [Guida AWS CloudTrail per l'utente](#).

AWS IoT SiteWise informazioni in CloudTrail

CloudTrail viene attivato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività di evento supportata in AWS IoT SiteWise, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi AWS di servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta [Visualizzazione degli eventi con cronologia degli CloudTrail eventi](#).

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi di AWS IoT SiteWise, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando si crea un trail nella console, il trail sarà valido in tutte

le Regioni AWS . Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Panoramica della creazione di un percorso](#)
- [CloudTrail servizi e integrazioni supportati](#)
- [Configurazione delle notifiche Amazon SNS per CloudTrail](#)
- [Ricezione di file di CloudTrail registro da più regioni](#) e [ricezione di file di CloudTrail registro da più account](#)

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente root o AWS Identity and Access Management (IAM).
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta [Elemento CloudTrail userIdentity](#).

AWS IoT SiteWise eventi di dati in CloudTrail

Gli [eventi di dati](#) forniscono informazioni sulle operazioni delle risorse eseguite su o in una risorsa (ad esempio, lettura o scrittura su un oggetto Amazon S3). Queste operazioni sono definite anche operazioni del piano dei dati. Gli eventi di dati sono spesso attività che interessano volumi elevati di dati. Per impostazione predefinita, CloudTrail non registra gli eventi relativi ai dati. La cronologia CloudTrail degli eventi non registra gli eventi relativi ai dati.

Per gli eventi di dati sono previsti costi aggiuntivi. Per ulteriori informazioni sui CloudTrail prezzi, consulta la sezione [AWS CloudTrail Prezzi](#).

Puoi registrare gli eventi relativi ai dati per i tipi di AWS IoT SiteWise risorse utilizzando la CloudTrail console o AWS CLI le operazioni CloudTrail dell'API. La [tabella](#) in questa sezione mostra i tipi di risorse disponibili per AWS IoT SiteWise.

- Per registrare gli eventi relativi ai dati utilizzando la CloudTrail console, crea un [percorso](#) o un [data store di eventi](#) per registrare gli eventi di dati oppure [aggiorna un trail o un data store di eventi esistente](#) per registrare gli eventi di dati.
 1. Scegli Data events per registrare gli eventi relativi ai dati.
 2. Dall'elenco Tipo di evento Data, scegli il tipo di risorsa per il quale desideri registrare gli eventi relativi ai dati.
 3. Scegli il modello di selettore di registro che desideri utilizzare. Puoi registrare tutti gli eventi relativi ai dati per il tipo di risorsa, registrare tutti `readOnly` gli eventi, registrare tutti `writeOnly` gli eventi o creare un modello di selettore di registro personalizzato per filtrare i `readOnly` `campieventName`, `resources.ARN`.
- Per registrare gli eventi relativi ai dati utilizzando il AWS CLI, configura il `--advanced-event-selectors` parametro in modo che il `eventCategory` campo sia uguale Data e il `resources.type` campo uguale al valore del tipo di risorsa (vedi [tabella](#)). È possibile aggiungere condizioni per filtrare i valori dei `resources.ARN` campi `readOnlyeventName`, e.
 - Per configurare un percorso per registrare gli eventi relativi ai dati, esegui il [AWS CloudTrail `put-event-selectors`](#) comando. Per ulteriori informazioni, vedere [Registrazione degli eventi relativi ai dati per i AWS CLI percorsi con](#).
 - Per configurare un Event Data Store per registrare gli eventi di dati, esegui il [AWS CloudTrail `create-event-data-store`](#) comando per creare un nuovo Event Data Store per registrare gli eventi di dati oppure esegui il [AWS CloudTrail `update-event-data-store`](#) comando per aggiornare un Event Data Store esistente. Per ulteriori informazioni, vedere [Registrazione degli eventi di dati per i data store di eventi con](#). AWS CLI

La tabella seguente elenca i tipi di AWS IoT SiteWise risorse. La colonna Data event type (console) mostra il valore da scegliere dall'elenco Data event type sulla CloudTrail console. La colonna del valore `resources.type` mostra il `resources.type` valore, da specificare durante la configurazione dei selettori di eventi avanzati utilizzando o. AWS CLI CloudTrail APIs La CloudTrail colonna Dati APIs registrati mostra le chiamate API registrate per il tipo di risorsa. CloudTrail

Tipo di evento di dati (console)	valore <code>resources.type</code>	Dati APIs registrati su* CloudTrail
AWS IoT SiteWise asset	<code>AWS::IoTSiteWise::Asset</code>	• BatchPutAssetProperty

Tipo di evento di dati (console)	valore resources.type	Dati APIs registrati su* CloudTrail
		• GetAssetPropertyValue • GetAssetPropertyValueHistory • GetAssetPropertyAggregates • GetInterpolatedAssetPropertyValues • BatchGetAssetPropertyValue • BatchGetAssetPropertyValueHistory • BatchGetAssetPropertyAggregates
AWS IoT SiteWise serie temporali	AWS::IoTSiteWise::TimeSeries	• BatchPutAssetPropertyValue • GetAssetPropertyValue • GetAssetPropertyValueHistory • GetAssetPropertyAggregates • GetInterpolatedAssetPropertyValues • BatchGetAssetPropertyValue • BatchGetAssetPropertyValueHistory • BatchGetAssetPropertyAggregates

Tipo di evento di dati (console)	valore resources.type	Dati APIs registrati su* CloudTrail
AWS IoT SiteWise Assistente	AWS::SitewiseAssis tant::Conversation	InvokeAssistant

Note

Il resources.type registrato nell'evento Cloudtrail dipende dall'identificatore utilizzato nella richiesta API. Se nella richiesta viene specificato un id di risorsa, viene registrato Asset resources.type, altrimenti viene registrato resources.type. TimeSeries

*Puoi configurare selettori di eventi avanzati per filtrare e resources . ARN campi per registrare solo gli eventName eventi che ritieni readOnlY importanti. Per ulteriori informazioni su questi campi, vedere [AdvancedFieldSelector](#).

AWS IoT SiteWise eventi di gestione in CloudTrail

[Gli eventi di gestione della registrazione](#) forniscono informazioni sulle operazioni di gestione eseguite sulle risorse AWS dell'account. Queste operazioni sono definite anche operazioni del piano di controllo (control-plane). Per impostazione predefinita, CloudTrail registra gli eventi di gestione.

AWS IoT SiteWise registra tutte le operazioni AWS IoT SiteWise del piano di controllo come eventi di gestione. Per un elenco delle operazioni del piano di AWS IoT SiteWise controllo a cui si AWS IoT SiteWise effettua l'accesso CloudTrail, consulta l'[AWS IoT SiteWise API](#) Reference.

Esempio: voci dei file di AWS IoT SiteWise registro

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'operazione richiesta, la data e l'ora dell'operazione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia stack ordinata delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che dimostra l>CreateAssetoperazione.

```

{
  "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"
}

```

Tagga le tue AWS IoT SiteWise risorse

L'etichettatura AWS IoT SiteWise delle risorse offre un modo efficace per classificare, gestire e recuperare le risorse organizzative in modo efficiente. Assegnando tag, che consistono in coppie chiave-valore, puoi allegare metadati descrittivi alle tue risorse. I metadati dei tag possono essere utilizzati per semplificare le operazioni. Ad esempio, in uno scenario di parco eolico, i tag consentono di etichettare le turbine con attributi specifici come posizione, capacità e stato operativo, consentendo una rapida identificazione e gestione all'interno. AWS IoT SiteWise

L'integrazione dei tag con le policy AWS Identity and Access Management (IAM) migliora la sicurezza e il controllo operativo definendo regole di accesso condizionale. Ciò significa che puoi specificare che siano ammessi solo gli utenti con determinati tag. Ad esempio, solo le persone etichettate con un determinato ruolo o reparto possono accedere o modificare risorse particolari.

Usa i tag in AWS IoT SiteWise

Utilizza i tag per classificare le AWS IoT SiteWise risorse in base allo scopo, al proprietario, all'ambiente o a qualsiasi altra classificazione per il tuo caso d'uso. In presenza di un numero elevato di risorse, puoi individuare rapidamente una risorsa specifica in base ai suoi tag.

Ogni tag è composto da una chiave e da un valore opzionale specificato dall'utente. Ad esempio, potete stabilire una serie di tag per i vostri modelli di asset per tracciarli in base ai processi industriali che supportano. Si consiglia di sviluppare un set personalizzato di chiavi di tag per ogni tipo di risorsa gestita. L'utilizzo di un set coerente di chiavi di tag può semplificare la gestione delle risorse.

Tagga con AWS Management Console

Il Tag Editor in the AWS Management Console offre un modo centralizzato e unificato per creare e gestire i tag per le risorse di tutti i AWS servizi. Per ulteriori informazioni, consulta [Guida introduttiva a Tag Editor](#) nella Guida per l'utente di Tagging AWS Resources and Tag Editor.

Tagga con l'API AWS IoT SiteWise

L' AWS IoT SiteWise API utilizza anche i tag. Prima di creare i tag, tenere presente le limitazioni relative al tagging. Per ulteriori informazioni, consulta la sezione relativa alle [convenzioni di denominazione e utilizzo dei tag](#) nella Riferimenti generali di AWS.

- Per aggiungere i tag durante la creazione di una risorsa, definirli nella proprietà tags della risorsa.

- Per aggiungere tag a una risorsa esistente o aggiornare i valori dei tag, utilizzate l'[TagResource](#) operazione.
- Per rimuovere i tag da una risorsa, usa l'[UntagResource](#) operazione.
- Per recuperare i tag associati a una risorsa, utilizzate l'[ListTagsForResource](#) operazione o descrivete la risorsa e controllatene le tags proprietà.

La tabella seguente elenca le risorse a cui è possibile etichettare utilizzando l' AWS IoT SiteWise API e le relative Create operazioni. Describe

Risorse taggabili AWS IoT SiteWise

Risorsa	Creare l'operazione	Descrivere l'operazione
Modello di asset o modello di componente	CreateAssetModel	DescribeAssetModel
Asset	CreateAsset	DescribeAsset
SiteWise Gateway Edge	CreateGateway	DescribeGateway
Portal	CreatePortal	DescribePortal
Progetto	CreateProject	DescribeProject
Dashboard	CreateDashboard	DescribeDashboard
Policy di accesso	CreateAccessPolicy	DescribeAccessPolicy
Serie temporali	BatchPutAssetPropertyValue	DescribeTimeSeries

È [BatchPutAssetPropertyValue](#) infatti possibile configurare le fonti di dati a cui inviare dati industriali AWS IoT SiteWise prima di creare modelli e asset di asset. AWS IoT SiteWise crea automaticamente flussi di dati per ricevere flussi di dati grezzi dalle apparecchiature. Per ulteriori informazioni, vedere [Gestione dell'ingestione dei dati](#).

Utilizza le operazioni seguenti per elencare e gestire i tag per le risorse che supportano il tagging:

- [TagResource](#)— Aggiunge tag a una risorsa o aggiorna il valore di un tag esistente.
- [ListTagsForResource](#)— Elenca i tag di una risorsa.

- [UntagResource](#)— Rimuove i tag da una risorsa.

Aggiungi o rimuovi tag da una risorsa in qualsiasi momento. Per aggiornare il valore di una chiave di tag esistente, aggiungi un nuovo tag con la stessa chiave e il nuovo valore desiderato alla risorsa. Questa azione sostituisce il vecchio valore con quello nuovo. Sebbene sia possibile assegnare una stringa vuota come valore di tag, non è possibile assegnare un valore nullo.

L'eliminazione di una risorsa rimuove anche tutti i tag ad essa collegati.

Usa i tag con le policy IAM

Utilizza i tag delle risorse nelle tue policy IAM per controllare l'accesso e le autorizzazioni degli utenti. Ad esempio, le policy possono consentire agli utenti di creare solo risorse a cui è associato un tag specifico. Le policy possono anche limitare gli utenti nella creazione o nella modifica di risorse con determinati tag.

Note

Se utilizzi tag per consentire o rifiutare agli utenti di accedere alle risorse, devi negare agli utenti la possibilità di aggiungere o rimuovere tali tag dalle stesse risorse. Altrimenti, un utente potrebbe aggirare le tue restrizioni e accedere a una risorsa modificandone i tag.

Puoi utilizzare le chiavi di contesto della condizione e i valori riportati di seguito nell'elemento `Condition` (denominato anche blocco `Condition`) di una dichiarazione di policy.

`aws:ResourceTag/tag-key: tag-value`

Consentire o negare agli utenti operazioni su risorse con tag specifici.

`aws:RequestTag/tag-key: tag-value`

Richiedere che un tag specifico venga utilizzato (o non utilizzato) durante la creazione o la modifica di una risorsa compatibile con l'applicazione dei tag.

`aws:TagKeys: [tag-key, ...]`

Richiedere che un set di chiavi di tag specifico venga utilizzato (o non utilizzato) durante la creazione o la modifica di una risorsa compatibile con l'applicazione dei tag.

 Note

Le chiavi e i valori del contesto della condizione in una policy IAM si applicano solo alle azioni che hanno una risorsa taggabile come parametro obbligatorio. Ad esempio, puoi impostare l'accesso condizionale basato su tag per [ListAssets](#). Non puoi attivare l'accesso condizionale basato su tag [PutLoggingOptions](#) perché nella richiesta non viene fatto riferimento a nessuna risorsa taggabile.

Per ulteriori informazioni, consulta [Controllare l'accesso alle AWS risorse utilizzando i tag di risorsa](#) e il [riferimento alla policy IAM JSON](#) nella IAM User Guide.

Esempi di politiche IAM che utilizzano i tag

- [Visualizza le AWS IoT SiteWise risorse in base ai tag](#)

Risoluzione dei problemi AWS IoT SiteWise

Utilizza le seguenti informazioni per risolvere i problemi relativi a. AWS IoT SiteWise

Argomenti

- [Risoluzione dei problemi di un gateway SiteWise Edge](#)
- [Risolvere i problemi relativi a un portale AWS IoT SiteWise](#)
- [Risolvere i problemi relativi a un' AWS IoT SiteWise azione delle regole](#)
- [Risoluzione dei problemi relativi alle operazioni di importazione ed esportazione in blocco](#)

Risoluzione dei problemi di un gateway SiteWise Edge

Risolvi i problemi comuni del gateway AWS IoT SiteWise Edge esplorando gli argomenti pertinenti.

Puoi anche visualizzare le CloudWatch metriche riportate dai tuoi gateway SiteWise Edge per risolvere problemi di connettività o flussi di dati. Per ulteriori informazioni, consulta [Monitoraggio AWS IoT SiteWise con i CloudWatch parametri di Amazon](#).

Argomenti

- [Configura e accedi ai log del gateway Edge SiteWise](#)
- [Risoluzione dei problemi relativi SiteWise al gateway Edge](#)
- [Risoluzione dei problemi relativi all'applicazione AWS IoT SiteWise Edge su Siemens Industrial Edge](#)
- [Risoluzione dei problemi di integrazioni open source su Edge](#)
- [Risoluzione dei problemi AWS IoT Greengrass](#)

Configura e accedi ai log del gateway Edge SiteWise

Prima di poter visualizzare i log del gateway SiteWise Edge, devi configurare il gateway SiteWise Edge per inviare i log ad Amazon CloudWatch Logs o archiviare i log sul file system locale.

- Usa CloudWatch Logs se desideri utilizzare il per visualizzare i file di log del AWS Management Console tuo gateway SiteWise Edge. Per ulteriori informazioni, consulta [Usa Amazon CloudWatch Logs](#).

- Utilizza i log del file system locale se desideri utilizzare la riga di comando o il software locale per visualizzare i file di registro del gateway SiteWise Edge. Per ulteriori informazioni, consulta [Usa i log in del servizio AWS IoT SiteWise](#).

Risoluzione dei problemi relativi SiteWise al gateway Edge

Utilizza le seguenti informazioni per risolvere i problemi relativi ai gateway SiteWise Edge.

Problemi

- [Impossibile distribuire i pacchetti sui gateway Edge SiteWise](#)
- [AWS IoT SiteWise non riceve dati dai server OPC UA](#)
- [Nessun dato viene visualizzato nella dashboard](#)
- [«Impossibile trovare o caricare la classe principale» visualizzato in aws.iot. SiteWiseEdgePublisher registra in/error greengrass/v2/logs](#)
- [Vedo 'SESSION_TAKEN_OVER' o 'com.aws.greengrass.mqttclient. MqttClient: Impossibile pubblicare il messaggio tramite Spooler e riproverò. ' nei log](#)
- [Vedo 'com.aws.greengrass.deployment. lotJobsHelper: Nessun processo di distribuzione trovato. ' o 'Risultato della distribuzione già segnalato'. nei registri](#)
- [Vedo lo stato «SYNC_FAILED» quando tento di configurare l'impostazione del timestamp in un gruppo di proprietà su una fonte di dati OPC UA](#)
- [I tipi di dati convertiti non sono inclusi](#)
- [Problemi relativi al Trust Store](#)
- [Problemi di installazione con proxy](#)

Impossibile distribuire i pacchetti sui gateway Edge SiteWise

Se il componente AWS IoT Greengrass nucleus (`aws.greengrass.Nucleus`) non è aggiornato, potresti non essere in grado di distribuire i pacchetti sul tuo SiteWise gateway Edge. Puoi usare la AWS IoT Greengrass V2 console per aggiornare il componente AWS IoT Greengrass nucleus.

Per aggiornare il componente AWS IoT Greengrass nucleus (console)

1. Passare alla [console AWS IoT Greengrass](#).
2. Nel riquadro di navigazione, sotto AWS IoT Greengrass, scegli Distribuzioni.
3. Nell'elenco Distribuzioni, seleziona la distribuzione che desideri modificare.

4. Scegli Rivedi.
5. Nella pagina Specificare la destinazione, scegli Avanti.
6. Nella pagina Seleziona componenti, in Componenti pubblici, nella casella di ricerca **aws.greengrass.Nucleus**, inserisci e quindi seleziona **AWS.GreenGrass.Nucleus**.
7. Scegli Next (Successivo).
8. Nella pagina Configura componenti, scegli Avanti.
9. Nella pagina Configura impostazioni avanzate, scegli Avanti.
10. Nella pagina Review (Verifica), scegli Deploy (Distribuisci).

AWS IoT SiteWise non riceve dati dai server OPC UA

Se le tue AWS IoT SiteWise risorse non ricevono i dati inviati dai tuoi server OPC UA, puoi cercare nei log del tuo gateway SiteWise Edge per risolvere i problemi. Cerca i `swPublisher` log a livello di informazioni che contengono il seguente messaggio.

```
Emitting diagnostic name=PublishError.SomeException
```

SomeException In base al tipo di registro, utilizza i seguenti tipi di eccezioni e i problemi corrispondenti per risolvere i problemi del gateway Edge: SiteWise

- **ResourceNotFoundException**— I vostri server OPC UA inviano dati che non corrispondono a un alias di proprietà per nessuna risorsa. Questa eccezione può verificarsi in due casi:
 - Gli alias delle proprietà non corrispondono esattamente alle variabili OPC UA, incluso il prefisso sorgente che hai definito. Verificare che gli alias delle proprietà e i prefissi di origine siano corretti.
 - Non hai mappato le variabili OPC UA alle proprietà degli asset. Per ulteriori informazioni, consulta [Gestisci i flussi di dati per AWS IoT SiteWise](#).

Se hai già mappato tutte le variabili OPC UA che desideri inserire AWS IoT SiteWise, puoi filtrare le variabili OPC UA inviate dal gateway Edge. SiteWise Per ulteriori informazioni, consulta [Usa i filtri dei nodi OPC UA in Edge SiteWise](#).

- **InvalidRequestException**— I tipi di dati delle variabili OPC UA non corrispondono ai tipi di dati relativi alle proprietà degli asset. Ad esempio, se una variabile OPC UA ha un tipo di dati intero, la proprietà dell'asset corrispondente deve essere di tipo intero. Una proprietà di asset di tipo doppio non può ricevere valori interi OPC UA. Per risolvere questo problema, definire nuove proprietà con i tipi di dati corretti.

- **TimestampOutOfRangeException**— Il gateway SiteWise Edge invia dati che non rientrano nell'intervallo consentito. AWS IoT SiteWise rifiuta tutti i punti dati con timestamp precedenti a 7 giorni o più recenti di 5 minuti nel futuro. Se il gateway SiteWise Edge ha perso l'alimentazione o la connessione al AWS Cloud, potrebbe essere necessario cancellare la cache del gateway SiteWise Edge.
- **ThrottlingException** oppure **LimitExceededException**: la richiesta ha superato una quota di AWS IoT SiteWise servizio, ad esempio la velocità di acquisizione dei punti dati o la frequenza di richiesta per le operazioni API relative ai dati relativi alle proprietà degli asset. Verificare che la configurazione non superi il valore [AWS IoT SiteWise quote](#).

Nessun dato viene visualizzato nella dashboard

Se nella dashboard non vengono visualizzati dati, la configurazione di Publisher e l'origine dati del gateway SiteWise Edge potrebbero non essere sincronizzate. Se non sono sincronizzati, l'aggiornamento del nome dell'origine dati può accelerare la sincronizzazione dal cloud all'edge, correggendo l'errore Out of sync.

Per aggiornare il nome di una fonte di dati

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Edge gateway.
3. Seleziona il gateway SiteWise Edge connesso alla dashboard.
4. In Origini dati, seleziona Modifica.
5. Seleziona un nuovo nome di origine e seleziona Salva per confermare la modifica.
6. Verifica le modifiche confermando che il nome dell'origine dati è stato aggiornato nella tabella Origini dati.

«Impossibile trovare o caricare la classe principale» visualizzato in aws.iot.

SiteWiseEdgePublisher registra in/error greengrass/v2/logs

Se viene visualizzato questo errore, potrebbe essere necessario aggiornare la versione java del gateway Edge. SiteWise

- Da un terminale, esegui il comando seguente:

```
java -version
```

La versione di java con cui è in esecuzione il gateway SiteWise Edge verrà visualizzata sotto `OpenJDK Runtime Environment`. Verrà visualizzata una risposta simile alla seguente:

```
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 mode)
```

Se si utilizza la versione Java 11.0.20.8.1, è necessario aggiornare il pacchetto IoT SiteWise Publisher alla versione 2.4.1 o successiva. È interessata solo la versione java 11.0.20.8.1, gli ambienti con altre versioni di Java possono continuare a utilizzare versioni precedenti del componente SiteWise IoT Publisher. Per ulteriori informazioni sull'aggiornamento di un pacchetto di componenti, vedere. [Cambia la versione dei pacchetti di componenti del gateway SiteWise Edge](#)

Vedo 'SESSION_TAKEN_OVER' o 'com.aws.greengrass.mqttclient. MqttClient: Impossibile pubblicare il messaggio tramite Spooler e riproverò. ' nei log

Se viene visualizzato un avviso che include SESSION_TAKEN_OVER o un errore incluso `com.aws.greengrass.mqttclient.MqttClient: Failed to publish the message via Spooler and will retry.` nei registri all'indirizzo `/greengrass/v2/logs/greengrass.log`, è possibile che si stia tentando di utilizzare lo stesso file di configurazione per più gateway SiteWise Edge su più dispositivi. Ogni gateway SiteWise Edge necessita di un file di configurazione unico per connettersi all'account AWS .

Vedo 'com.aws.greengrass.deployment. IotJobsHelper: Nessun processo di distribuzione trovato. ' o 'Risultato della distribuzione già segnalato'. nei registri

Se vedi `com.aws.greengrass.deployment.IotJobsHelper: No deployment job found. o Deployment result already reported.` nei tuoi log all'indirizzo `/greengrass/v2/logs/greengrass.log`, potresti provare a riutilizzare lo stesso file di configurazione.

Esistono diverse soluzioni:

- Se desideri riutilizzare il file di configurazione, procedi come segue:
 1. Passare alla [console AWS IoT SiteWise](#).
 2. Nel riquadro di navigazione, scegli Edge gateway.
 3. Scegli il gateway SiteWise Edge che desideri riutilizzare.

4. Scegli la scheda Aggiornamenti.
5. Seleziona una versione diversa di Publisher e scegli Distribuisci.

Segui i passaggi indicati [Crea un gateway per Siemens Industrial Edge](#) per creare un nuovo file di configurazione.

Vedo lo stato «SYNC_FAILED» quando tento di configurare l'impostazione del timestamp in un gruppo di proprietà su una fonte di dati OPC UA

Quando è AWS IoT SiteWise stato aggiornato il componente OPC UA Collector per AWS IoT Greengrass la versione 2.5.0, abbiamo introdotto una nuova opzione di configurazione del timestamp. Puoi utilizzare il timestamp del tuo dispositivo o il timestamp del server. Le versioni precedenti del componente OPC UA Collector non supportano questa opzione e non riescono a sincronizzarsi.

Esistono due modi per risolvere un errore di sincronizzazione dell'origine dati. Il metodo consigliato è quello di aggiornare il componente IoT SiteWise OPC UA collector alla versione 2.5.0 o successiva. In alternativa, puoi continuare a utilizzare la versione precedente del componente OPC UA Collector, se hai impostato il timestamp su. Source Per informazioni su come aggiornare il componente IoT SiteWise OPC UA Collector, vedere. [Aggiorna la versione di un componente AWS IoT SiteWise](#) Si consiglia di utilizzare le versioni più recenti di tutti i componenti.

Note

Non si verifica alcuna interruzione dei dati quando lo stato di sincronizzazione di un'origine dati fallisce. I dati di origine continuano a AWS IoT SiteWise fluire. La configurazione semplicemente non si sincronizza con il componente IoT SiteWise OPC UA collector durante l'implementazione. AWS IoT Greengrass V2

Per modificare la configurazione del timestamp per un gruppo di proprietà

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel pannello di navigazione, scegli Edge gateway.
3. Seleziona il gateway da modificare.
4. Nella sezione Origini dati, seleziona l'origine dati con lo stato di sincronizzazione non riuscita e scegli Modifica.
5. Espandi Configurazione avanzata, quindi espandi Impostazioni di gruppo.

6. In Timestamp, seleziona Sorgente. La selezione di Source rimuove la `timestampToReturn` proprietà dalla configurazione. Questa impostazione consente la raccolta del timestamp dell'origine dati dal dispositivo per impostazione predefinita, consentendo la sincronizzazione dell'origine dati con il componente di raccolta IoT SiteWise OPC UA.
7. Scegli Save (Salva).

I tipi di dati convertiti non sono inclusi

Se visualizzi un errore durante la conversione di tipi di dati OPC UA non supportati in stringhe AWS IoT SiteWise, ci sono alcuni possibili motivi:

- Il tipo di dati che stai tentando di convertire è un tipo di dati complesso. I tipi di dati complessi non sono supportati.
- Quando si utilizza Destinations as AWS IoT SiteWise Buffered con Amazon S3, il valore completo della stringa viene conservato nei file inviati a un bucket Amazon S3. Quando successivamente si inseriscono dati in un secondo momento AWS IoT SiteWise, i valori di stringa completi più lunghi di 1024 byte vengono rifiutati.

Problemi relativi al Trust Store

Se riscontri problemi relativi ai trust stores in SiteWise Edge, prendi in considerazione i seguenti passaggi per la risoluzione dei problemi:

- Verifica che il certificato CA AWS IoT Greengrass principale sia presente e formattato correttamente negli archivi attendibili appropriati
- Assicurati che la KeyStore password Java sia impostata correttamente e accessibile ai componenti SiteWise Edge
- Verifica che tutti i certificati personalizzati (ad esempio per i proxy HTTPS) siano nel formato corretto (in genere PEM) e importati correttamente negli archivi affidabili
- Verifica che gli archivi attendibili dispongano delle autorizzazioni di file corrette e siano accessibili ai processi Edge SiteWise
- Esamina i log di SiteWise Edge per eventuali errori SSL/TLS correlati, che potrebbero indicare problemi di trust store
- Testa SSL/TLS le connessioni in modo indipendente utilizzando strumenti come `openssl` la verifica della funzionalità del trust store

Problemi di installazione con proxy

Se riscontri problemi durante il processo di configurazione del proxy, considera i seguenti passaggi per la risoluzione dei problemi:

- Verifica che l'URL del proxy sia formattato correttamente e includa lo schema corretto (`http://` o `https://`)
- Assicurati che tutte le credenziali del proxy siano codificate come URL se contengono caratteri speciali
- Verifica che l'elenco senza proxy includa tutti gli indirizzi locali e gli endpoint di servizio necessari AWS
- Per i proxy HTTPS, verifica che il certificato CA fornito sia in formato PEM
- Esamina i log di installazione per eventuali messaggi di errore specifici che potrebbero indicare l'origine del problema
- Verifica la connessione proxy in modo indipendente per assicurarti che funzioni correttamente

Risoluzione dei problemi relativi all'applicazione AWS IoT SiteWise Edge su Siemens Industrial Edge

Per risolvere i problemi relativi AWS IoT SiteWise all'applicazione Edge sul Siemens Industrial Edge dispositivo, è possibile accedere ai registri dell'applicazione tramite i portali Siemens Industrial Edge Management o Siemens Industrial Edge Device (IED). Per ulteriori informazioni, consulta la sezione [Download dei registri nella documentazione](#) di Siemens.

I miei dati non vengono visualizzati in AWS IoT SiteWise

- Assicurati che non vi siano problemi con Databus gli utenti e che l'icona del segno di spunta per Databus_Configuration sia verde anziché grigia.
- È possibile che non sia in esecuzione una versione Siemens Industrial Edge Management che contiene Secure Storage. Aggiorna la tua versione del sistema operativo Siemens. Per ulteriori informazioni, consulta [Siemens Secure Storage e l'applicazione AWS IoT SiteWise Edge](#).

Vedo «File di configurazione mancante AWS_REGION» nei log

Se vedi `Config file missing AWS_REGION` nei log Siemens, il codice JSON del file di configurazione è danneggiato. Dovrai creare un nuovo file di configurazione. Segui i passaggi indicati [Crea un gateway per Siemens Industrial Edge](#) per creare un nuovo file di configurazione.

Viene visualizzato un messaggio di errore «Non sincronizzato» nella configurazione del gateway Edge

Se viene visualizzato un messaggio `Out of sync` di errore sul Siemens Industrial Edge gateway dopo il completamento dell'implementazione, significa che il componente IoT SiteWise Publisher non è sincronizzato con il gateway. Il componente IoT SiteWise Publisher funziona in background sui Siemens Industrial Edge gateway per fornire funzionalità tematiche MQTT. Abbiamo aggiornato i Siemens Industrial Edge gateway per utilizzare lo spazio dei nomi delle funzionalità anziché `iotsitewise:publisher:3` `iotsitewise:publisher:2`. Puoi eseguire l'aggiornamento alla versione più recente dell'editore per risolvere questo problema.

Per eseguire l'aggiornamento alla versione più recente di IoT SiteWise Publisher

1. Passare alla [console AWS IoT SiteWise](#).
2. Nel riquadro di navigazione, scegli Edge gateway.
3. Seleziona il Siemens Industrial Edge gateway da modificare.
4. Nella sezione Funzionalità di Edge, seleziona Visualizza versioni del software.
5. Seleziona l'ultima versione dell' SiteWise editore IoT nel menu a discesa Publisher.
6. Seleziona Fatto.

Risoluzione dei problemi di integrazioni open source su Edge

Questa sezione fornisce soluzioni ai problemi più comuni che potresti riscontrare durante l'integrazione di strumenti open source con Edge. SiteWise

Note

Node-RED®, InfluxDB® e Grafana® non sono fornitori o fornitori di Edge. SiteWise

Problemi di connessione

Node-RED non può connettersi al broker MQTT

Verifica che il broker MQTT sia in esecuzione e accessibile sulla porta specificata. Controlla la configurazione della rete e assicurati che l'indirizzo del broker sia corretto.

Per verificare lo stato del broker MQTT, esegui:

```
docker ps | grep emqx
```

Errori di connessione InfluxDB

Assicurati che il tuo token di autenticazione sia valido e di aver specificato l'organizzazione e i nomi dei bucket corretti. Verifica che InfluxDB sia funzionante e accessibile.

Per verificare lo stato di InfluxDB, esegui:

```
curl -I http://localhost:8086
```

Grafana non riesce a connettersi a InfluxDB

Verifica che la configurazione dell'origine dati InfluxDB in Grafana sia corretta, inclusi l'URL, il token di autenticazione, l'organizzazione e il bucket.

Problemi relativi al flusso di dati

Nessun dato visualizzato in AWS IoT SiteWise

Verifica che l'alias della tua proprietà nel flusso Node-RED corrisponda al formato previsto. Verificate che la struttura degli argomenti MQTT sia corretta e che il gateway SiteWise Edge sia configurato correttamente per ricevere dati dal broker MQTT.

Nessun dato SiteWise Edge memorizzato in InfluxDB

Verifica che il flusso di ritenzione Node-RED sia configurato correttamente e che il nodo writer InfluxDB abbia il bucket e le impostazioni di misurazione corretti. Controlla l'output di debug di Node-RED per eventuali errori.

Errori di formato dei dati

Assicurati che le funzioni di trasformazione dei dati convertano correttamente i dati tra i formati. Usa i nodi di debug Node-RED per ispezionare i dati in ogni fase del flusso.

Problemi di prestazioni

Utilizzo elevato della CPU o della memoria

Monitora l'utilizzo delle risorse e regola la configurazione dei componenti in base alle esigenze. Prendi in considerazione la possibilità di ridurre la frequenza di raccolta dei dati o di implementare il filtraggio dei dati per ridurre il carico di elaborazione.

Per monitorare l'utilizzo delle risorse, esegui:

```
docker stats
```

Caricamento lento della dashboard Grafana

Ottimizza le tue query su InfluxDB e prendi in considerazione l'aggiunta di limiti di intervallo di tempo ai pannelli della dashboard. Riduci il numero di punti dati visualizzati utilizzando le funzioni di aggregazione appropriate.

Registrazione e diagnostica

Per risolvere i problemi, controlla i log di ogni componente:

Registri Node-RED

Visualizza i log nella console Node-RED o esegui:

```
docker logs node-red
```

Registri di InfluxDB

Accedi ai log eseguendo:

```
docker logs influxdb
```

Tronchi Grafana

Visualizza i log eseguendo:

```
docker logs grafana
```

SiteWise Registri Edge

Controllate i log del gateway SiteWise Edge per eventuali problemi di connessione MQTT e di elaborazione dei dati. Per ulteriori informazioni, consulta [Risoluzione dei problemi di un gateway SiteWise Edge](#).

Risoluzione dei problemi AWS IoT Greengrass

Per trovare soluzioni a molti problemi relativi alla configurazione o all'implementazione del gateway SiteWise Edge AWS IoT Greengrass, consulta [Risoluzione dei problemi AWS IoT Greengrass](#) nella Guida per gli sviluppatori.AWS IoT Greengrass

Risolvere i problemi relativi a un portale AWS IoT SiteWise

Risolvi i problemi più comuni con i tuoi AWS IoT SiteWise portali.

Gli utenti e gli amministratori non possono accedere al portale AWS IoT SiteWise

Se gli utenti o gli amministratori non possono accedere AWS IoT SiteWise al tuo portale, potresti avere autorizzazioni limitate nelle policy allegate AWS Identity and Access Management (IAM) che impediscono il successo degli accessi.

Guarda i seguenti esempi di politiche IAM che comporteranno errori di accesso:

Note

Qualsiasi policy IAM allegata che includa un "Condition" elemento causerà un errore di accesso.

Esempio 1: la condizione qui è un IP limitato e ciò causerà un errore di accesso.

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

```

    "Effect": "Allow",
    "Action": [
        "iotsitewise:DescribePortal"
    ],
    "Resource": "*",
    "Condition": {
        "IpAddress": {
            "aws:SourceIp": [
                "REPLACESAMPLEIP"
            ]
        }
    }
}

```

Esempio 2: la condizione qui è un tag incluso e ciò causerà un errore di accesso.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/project": "*"
        }
      }
    }
  ]
}

```

Quando aggiungi utenti o amministratori al portale, evita di creare policy IAM che limitino le autorizzazioni degli utenti, come l'IP limitato. Qualsiasi policy allegata con autorizzazioni limitate non sarà in grado di connettersi al portale. AWS IoT SiteWise

Risolvere i problemi relativi a un' AWS IoT SiteWise azione delle regole

Per risolvere i problemi relativi all'azione della AWS IoT SiteWise regola in AWS IoT Core, è possibile eseguire una delle seguenti procedure:

- Configurazione di Amazon CloudWatch Logs
- Configurare un'azione di ripubblicazione dell'errore per la regola

Quindi, confrontare i messaggi di errore con gli errori presenti in questo argomento per risolvere il problema.

Argomenti

- [Configura i log AWS IoT Core](#)
- [Configura un'azione di errore di ripubblicazione](#)
- [Risolvi i problemi relativi alle regole](#)
- [Risolvere i problemi relativi a una regola \(\)AWS IoT SiteWise](#)
- [Risoluzione dei problemi relativi a una regola \(DynamoDB\)](#)

Configura i log AWS IoT Core

È possibile AWS IoT configurare la registrazione di vari livelli di informazioni nei CloudWatch registri.

Per configurare e accedere ai registri CloudWatch

1. Per configurare la registrazione per AWS IoT Core, consulta [Monitoring with CloudWatch Logs](#) nella Developer Guide.AWS IoT
2. Passare alla [console CloudWatch](#) .
3. Nel pannello di navigazione, selezionare Log groups (Gruppi di log).
4. Scegli il gruppo AWSIoTLogs.
5. Scegliere un flusso di log recente. Per impostazione predefinita, CloudWatch visualizza per primo il flusso di log più recente.
6. Scegliere una voce di log per espandere il messaggio di log. La voce di log potrebbe essere simile allo screenshot seguente.

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. Confrontare i messaggi di errore con gli errori in questo argomento per risolvere il problema.

Configura un'azione di errore di ripubblicazione

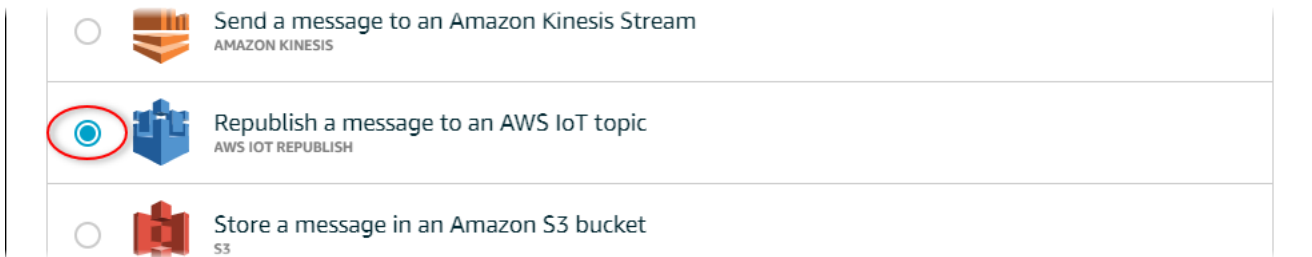
Puoi configurare un'azione di errore sulla regola per gestire i messaggi di errore. In questa procedura, è possibile configurare l'azione regola di ripubblicazione come azione di errore per visualizzare i messaggi di errore nel client di test MQTT.

Note

L'azione di ripubblicazione dell'errore genera solo un output equivalente ai log a livello di ERROR. [Se desideri registri più dettagliati, devi configurare i registri. CloudWatch](#)

Per aggiungere un'azione di errore di ripubblicazione a una regola

1. Passare alla [console AWS IoT](#).
2. Nel riquadro di navigazione sinistro scegliere Atti e quindi Regole.
3. Scegliere la regola.
4. In Azione errore, scegliere Aggiungi azione.
5. Scegli Ripubblica un messaggio su un argomento. AWS IoT



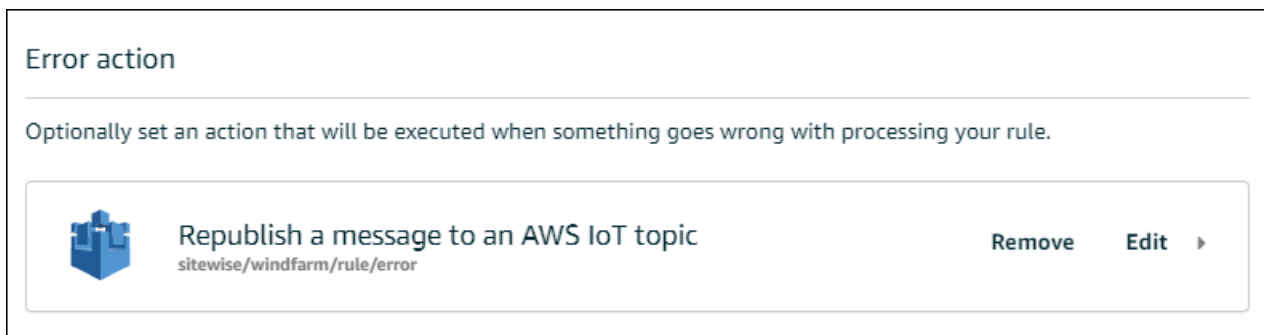
Send a message to an Amazon Kinesis Stream
AMAZON KINESIS

Republish a message to an AWS IoT topic
AWS IOT REPUBLISH

Store a message in an Amazon S3 bucket
S3

6. Scegliere Configura azione nella parte inferiore della pagina.
7. In Argomento, inserisci un argomento unico (ad esempio, **sitewise/windfarm/rule/error**). AWS IoT Core ripubblicherà i messaggi di errore relativi a questo argomento.
8. Scegli Seleziona per concedere AWS IoT Core l'accesso per eseguire l'azione di errore.
9. Scegliere Select (Seleziona) accanto al ruolo creato per la regola.
10. Scegliere Aggiorna ruolo per aggiungere le autorizzazioni aggiuntive al ruolo.
11. Selezionare Add action (Aggiungi operazione).

L'azione di errore della regola è simile allo screenshot seguente.



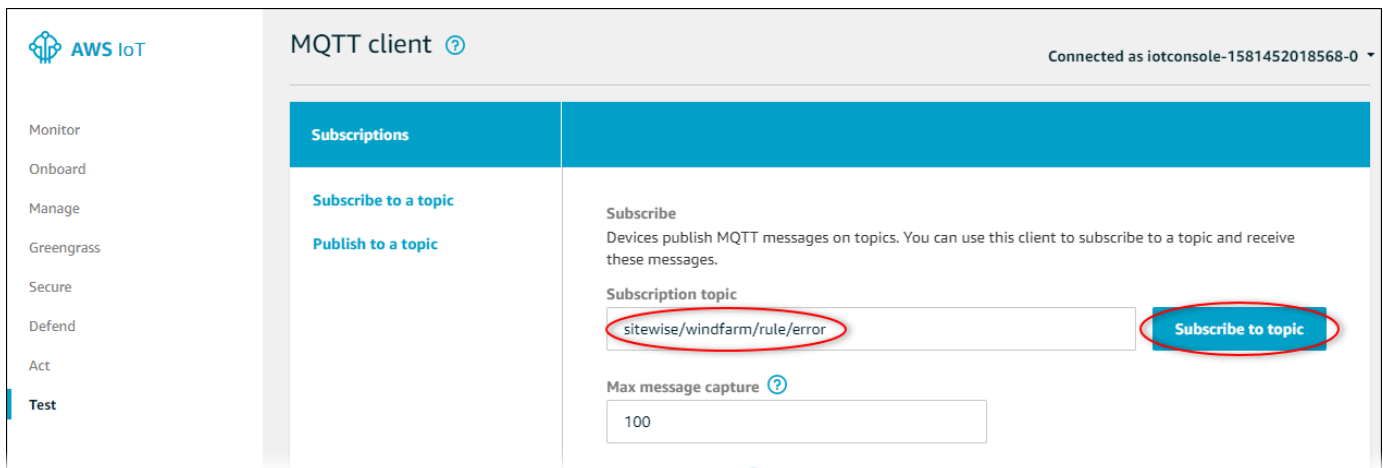
12. Scegli la freccia rivolta verso il basso nella parte superiore sinistra della console per tornare alla home page della AWS IoT console.

Dopo aver impostato l'azione di ripubblicazione dell'errore, è possibile visualizzare i messaggi di errore nel client di test MQTT in AWS IoT Core.

Nella procedura seguente, si sottoscrive l'argomento di errore nel client di test MQTT. Nel client di test MQTT, è possibile ricevere i messaggi di errore della regola per risolvere il problema.

Per sottoscrivere l'argomento dell'azione di errore

1. Passare alla [console AWS IoT](#).
2. Nel riquadro di navigazione a sinistra, scegliere Test per aprire il client di test MQTT.
3. Nel campo Subscription topic (Argomento sottoscrizione) immettere l'argomento di errore configurato in precedenza (ad esempio **sitewise/windfarm/rule/error**) e scegliere Subscribe to topic (Effettua sottoscrizione all'argomento).



4. Controllare se vengono visualizzati messaggi di errore e quindi espandere l'array `failures` in qualsiasi messaggio di errore.

Confronta i messaggi di errore con gli errori presenti in questo argomento per risolvere il problema.

Risolvi i problemi relativi alle regole

Utilizzare le seguenti informazioni per risolvere i problemi relativi alle regole.

Problemi

- [Errore: il membro deve trovarsi entro 604800 secondi prima e 300 secondi dopo il timestamp corrente](#)
- [Errore: il valore della proprietà non corrisponde al tipo di dati <type>](#)
- [Errore: Utente: <role-arn>non autorizzato a eseguire: iotsitewise: on resource BatchPutAssetPropertyValue](#)
- [Errore: iot.amazonaws.com non è in grado di eseguire: sts: on resource: AssumeRole <role-arn>](#)
- [Informazioni: non è stata inviata alcuna richiesta. PutAssetPropertyValueEntries era vuoto dopo aver eseguito i modelli sostitutivi.](#)

Errore: il membro deve trovarsi entro 604800 secondi prima e 300 secondi dopo il timestamp corrente

Il timestamp è più vecchio di 7 giorni o più recente di 5 minuti, rispetto all'ora attuale di Unix. Eseguire quanto segue:

- Controllare che il timestamp sia nell'ora in formato epoch Unix (UTC). Se si fornisce un timestamp con un fuso orario diverso, si riceve questo errore.
- Verifica che il timestamp sia espresso in secondi. AWS IoT SiteWise prevede che i timestamp siano suddivisi in secondi (in epoca Unix) e offset in nanosecondi.
- Verifica di caricare dati con data e ora non più tardi di 7 giorni nel passato.

Errore: il valore della proprietà non corrisponde al tipo di dati <type>

Una voce nell'azione di regola ha un tipo di dati diverso dalla proprietà dell'asset di destinazione. Ad esempio, la proprietà dell'asset di destinazione è di tipo `DOUBLE` ed è stato selezionato il tipo di dati `Integer` o è stato passato il valore `integerValue`. Eseguire quanto segue:

- Se configuri la regola dalla AWS IoT console, verifica di aver scelto il tipo di dati corretto per ogni immissione.
- Se configuri la regola dall'API o AWS Command Line Interface (AWS CLI), verifica che l'oggetto `value` utilizzi il campo di tipo corretto (ad esempio, `doubleValue` per una `DOUBLE` proprietà).

Errore: Utente: <role-arn>non autorizzato a eseguire: iotsitewise: on resource BatchPutAssetPropertyValue

La regola non è autorizzata ad accedere alla proprietà dell'asset di destinazione oppure la proprietà dell'asset di destinazione non esiste. Eseguire quanto segue:

- Verificare che l'alias della proprietà sia corretto e che si disponga di una proprietà dell'asset con l'alias di proprietà specificato. Per ulteriori informazioni, consulta [Gestisci i flussi di dati per AWS IoT SiteWise](#).
- Verificare che la regola abbia un ruolo e che il ruolo conceda l'autorizzazione `iotsitewise:BatchPutAssetPropertyValue` alla proprietà dell'asset di destinazione, ad esempio tramite la gerarchia dell'asset di destinazione. Per ulteriori informazioni, consulta [Concedi AWS IoT l'accesso richiesto](#).

Errore: iot.amazonaws.com non è in grado di eseguire: sts: on resource: AssumeRole <role-arn>

Il tuo utente non è autorizzato ad assumere il ruolo previsto dalla tua regola in (IAM). AWS Identity and Access Management

Verifica che al tuo utente sia concessa l'iam:PassRole autorizzazione per il ruolo in base alla tua regola. Per ulteriori informazioni, consulta [Pass role permissions](#) nella AWS IoT Developer Guide.

Informazioni: non è stata inviata alcuna richiesta. PutAssetPropertyValueEntries era vuoto dopo aver eseguito i modelli sostitutivi.

 Note

Questo messaggio è un log a livello di INFO.

La richiesta deve includere almeno una voce con tutti i parametri richiesti.

Verificare che i parametri della regola, inclusi i modelli di sostituzione, restituiscano valori non vuoti. I modelli di sostituzione non possono accedere ai valori definiti nelle clausole AS dell'istruzione della query della regola. Per ulteriori informazioni, consulta [Modelli sostitutivi](#) nella Guida per gli sviluppatori.AWS IoT

Risolvere i problemi relativi a una regola ()AWS IoT SiteWise

Segui i passaggi di questa procedura per risolvere la regola se i dati sull'utilizzo della CPU e della memoria non vengono visualizzati come previsto. AWS IoT SiteWise In questa procedura, è possibile configurare l'azione regola di ripubblicazione come azione di errore per visualizzare i messaggi di errore nel client di test MQTT. Puoi anche configurare la registrazione su Logs per la risoluzione dei problemi. CloudWatch Per ulteriori informazioni, consulta [Risolvere i problemi relativi a un' AWS IoT SiteWise azione delle regole](#).

Per aggiungere un'azione di errore di ripubblicazione a una regola

1. Passare alla [console AWS IoT](#).
2. Nel riquadro di navigazione a sinistra, scegli Routing dei messaggi, quindi scegli Regole.
3. Scegli la regola che hai creato in precedenza e scegli Modifica.

4. In Azione di errore - facoltativa, scegli Aggiungi azione di errore.
5. Scegli Ripubblica un messaggio su un AWS IoT argomento.
6. In Argomento, inserisci il percorso dell'errore (ad esempio, **sitewise/rule/tutorial/error**). AWS IoT Core ripubblicherà i messaggi di errore relativi a questo argomento.
7. Scegliete il ruolo che avete creato in precedenza (ad esempio, SiteWiseTutorialDeviceRuleRole).
8. Scegli Aggiorna.

Dopo aver impostato l'azione di ripubblicazione dell'errore, è possibile visualizzare i messaggi di errore nel client di test MQTT in AWS IoT Core.

Nella procedura seguente, si sottoscrive l'argomento di errore nel client di test MQTT.

Per sottoscrivere l'argomento dell'azione di errore

1. Passare alla [console AWS IoT](#).
2. Nella pagina di navigazione a sinistra, scegliete MQTT test client per aprire il client di test MQTT.
3. Nel campo Filtro argomento, inserisci **sitewise/rule/tutorial/error** e scegli Iscriviti.

Quando vengono visualizzati messaggi di errore, visualizzare l'array `failures` in qualsiasi messaggio di errore per diagnosticare i problemi. Per ulteriori informazioni sui possibili problemi e su come risolverli, consultare [Risolvere i problemi relativi a un' AWS IoT SiteWise azione delle regole](#).

Se non vengono visualizzati errori, verificare che la regola sia abilitata e che sia stato sottoscritto lo stesso argomento configurato nell'azione errore di ripubblicazione. Se dopo questa operazione ancora non vengono visualizzati gli errori, verificare che lo script del dispositivo sia in esecuzione e aggiornare correttamente la shadow del dispositivo.

Note

Puoi anche iscriverti all'argomento Shadow Update del tuo dispositivo per visualizzare il payload analizzato dalla tua AWS IoT SiteWise azione. A tale scopo, eseguire la sottoscrizione al seguente argomento.

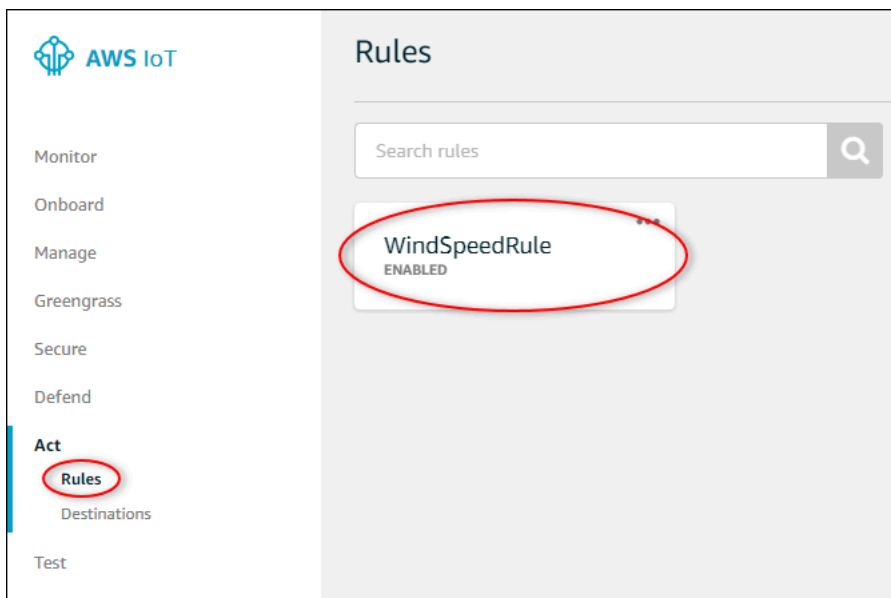
```
$aws/things/+/shadow/update/accepted
```

Risoluzione dei problemi relativi a una regola (DynamoDB)

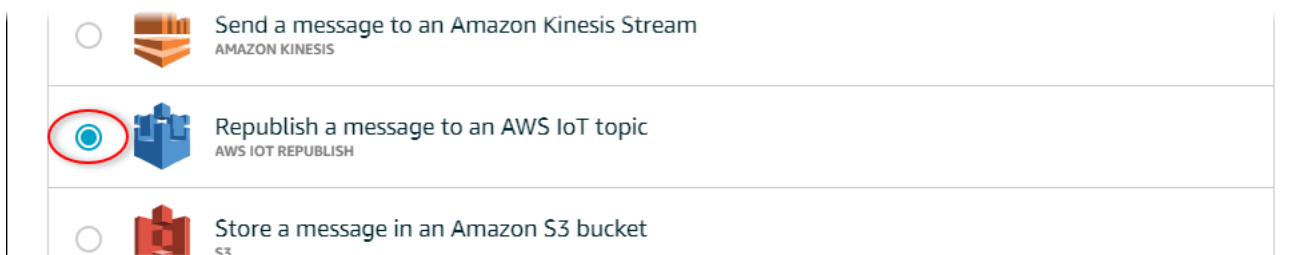
Segui i passaggi di questa procedura per risolvere i problemi relativi alla regola se i dati dell'asset demo non vengono visualizzati nella tabella DynamoDB come previsto. In questa procedura, è possibile configurare l'azione regola di ripubblicazione come azione di errore per visualizzare i messaggi di errore nel client di test MQTT. Puoi anche configurare la registrazione su Logs per risolvere i problemi. CloudWatch Per ulteriori informazioni, consulta la sezione [Monitoraggio con i registri CloudWatch](#) nella Guida per gli sviluppatori di AWS IoT .

Per aggiungere un'azione di errore di ripubblicazione a una regola

1. Passare alla [console AWS IoT](#).
2. Nel riquadro di navigazione sinistro scegliere Atti e quindi Regole.
3. Scegliere la regola creata in precedenza.



4. In Azione errore, scegliere Aggiungi azione.
5. Scegli Ripubblica un messaggio su un argomento. AWS IoT



6. Scegliere Configura azione nella parte inferiore della pagina.

7. In Topic (Argomento), inserisci **windspeed/error**. AWS IoT Core ripubblicherà i messaggi di errore su questo argomento.

The screenshot shows the 'Configure action' dialog for the 'Republish a message to an AWS IoT topic' action. The dialog has a blue header with the title 'Configure action'. Below the header, there is a section with the AWS IoT Republish icon and the title 'Republish a message to an AWS IoT topic'. A description states: 'This action will republish the message to another AWS IoT topic.' The configuration section includes a '*Topic' field with a help icon, where 'windspeed/error' is entered and circled in red. Below this is the 'Quality of Service' section with a help icon, showing two radio button options: '0 - The message is delivered zero or more times.' (selected) and '1 - The message is delivered one or more times.' Below the configuration section, there is a section titled 'Choose or create a role to grant AWS IoT access to perform this action.' It contains a 'No role selected' button, a 'Create Role' button, and a 'Select' button, which is circled in red. At the bottom of the dialog, there are 'Cancel' and 'Add action' buttons.

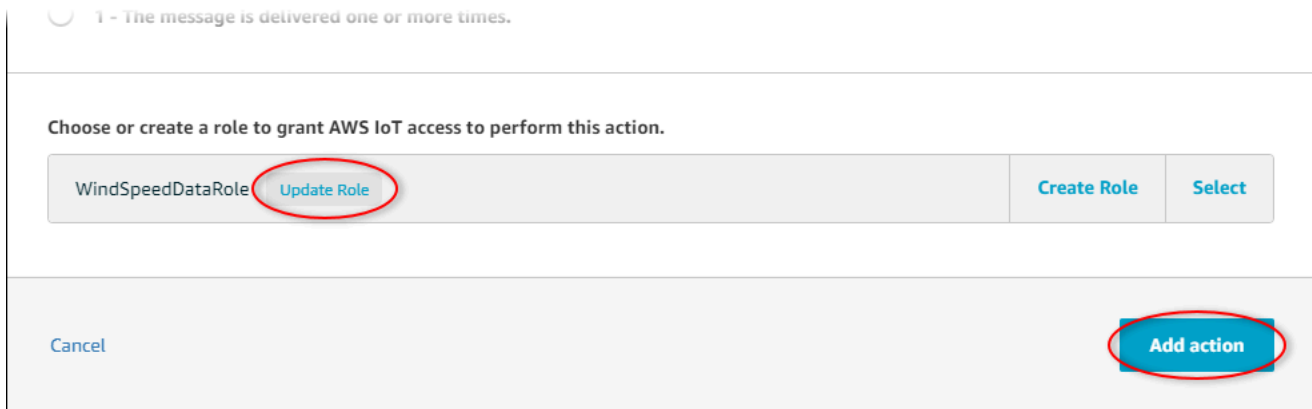
8. Scegli Seleziona per concedere a AWS IoT Core l'accesso per eseguire l'azione di errore utilizzando il ruolo che hai creato in precedenza.
9. Scegli Seleziona accanto al tuo ruolo.

The screenshot shows the role selection dialog. It has a header with the title 'Choose or create a role to grant AWS IoT access to perform this action.' Below the header, there is a table with the following structure:

No role selected	Refresh	Create Role	Close
Search for IAM roles			
WindSpeedDataRole			Select

The 'Select' button in the last row is circled in red.

10. Scegliere Aggiorna ruolo per aggiungere le autorizzazioni aggiuntive al ruolo.



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

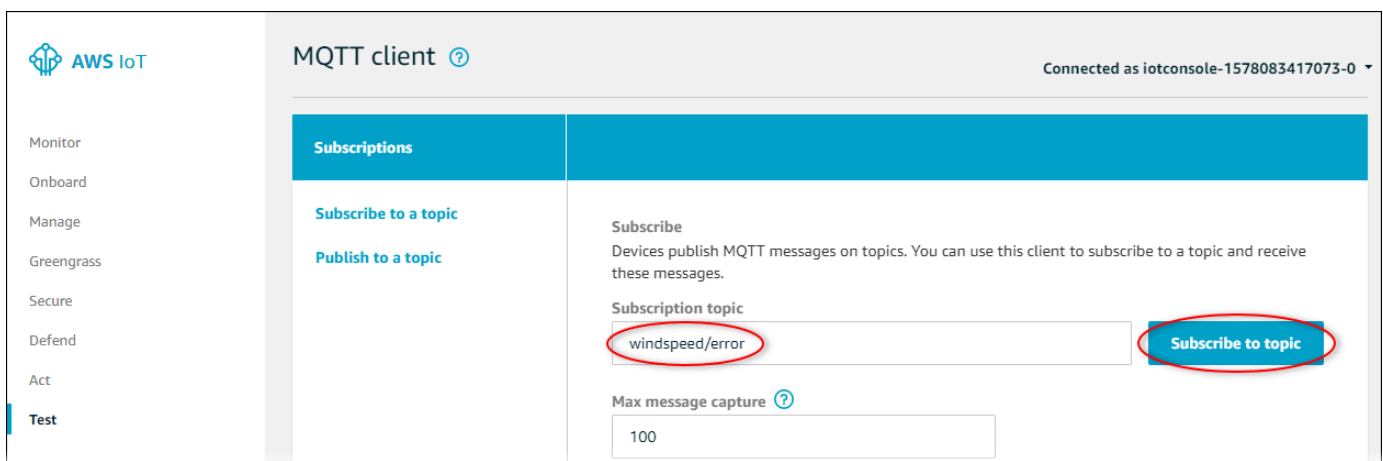
11. Scegliere Aggiungi azione per completare l'aggiunta dell'azione di errore.
12. Scegli la freccia indietro nella parte superiore sinistra della console per tornare alla home page della console AWS IoT Core.

Dopo aver impostato l'azione di errore di ripubblicazione, è possibile visualizzare i messaggi di errore nel client di test MQTT in AWS IoT Core.

Nella procedura seguente, si sottoscrive l'argomento di errore nel client di test MQTT.

Per sottoscrivere l'argomento dell'azione di errore

1. Nella pagina di navigazione a sinistra della console AWS IoT Core, scegli Test.
2. Nel campo Argomento della sottoscrizione immettere **windspeed/error** e scegliere Iscriviti all'argomento.



AWS IoT

Monitor

Onboard

Manage

Greengrass

Secure

Defend

Act

Test

MQTT client ?

Connected as iotconsole-1578083417073-0

Subscriptions

Subscribe to a topic

Publish to a topic

Subscribe

Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

Subscription topic

windspeed/error

Subscribe to topic

Max message capture ?

100

3. Controlla la visualizzazione dei messaggi di errore ed esplora l'array `failures` in un messaggio di errore per diagnosticare i seguenti problemi comuni:
 - Errori di battitura nell'istruzione query regola

- Autorizzazioni ruolo insufficienti

Se non vengono visualizzati errori, verificare che la regola sia abilitata e che sia stato sottoscritto lo stesso argomento configurato nell'azione errore di ripubblicazione. Se ancora non vengono visualizzati errori, verificare che le risorse del parco eolico dimostrativo esistano ancora e che siano state abilitate le notifiche sulle proprietà della velocità del vento. Se le tue risorse demo sono scadute e sono scomparse da AWS IoT SiteWise, puoi creare una nuova demo e aggiornare l'istruzione di interrogazione delle regole in modo che rifletta il modello e la proprietà dell'asset aggiornati. IDs

Risoluzione dei problemi relativi alle operazioni di importazione ed esportazione in blocco

Per gestire e diagnosticare gli errori prodotti durante un processo di trasferimento, consulta l' AWS IoT TwinMaker GetMetadataTransferJobAPI:

1. Dopo aver creato ed eseguito un processo di trasferimento, chiama l'GetMetadataTransferJobAPI:

```
aws iottwinmaker get-metadata-transfer-job \  
--metadata-transfer-job-id your_metadata_transfer_job_id \  
--region us-east-1
```

2. Lo stato del lavoro cambia in uno dei seguenti stati:
 - COMPLETED
 - CANCELLED
 - ERRORE
3. L'GetMetadataTransferJobAPI restituisce un [MetadataTransferJobProgress](#)oggetto.
4. L'MetadataTransferJobProgressoggetto contiene i seguenti parametri:
 - FailedCount: indica il numero di asset che hanno avuto esito negativo durante il processo di trasferimento.
 - SkippedCount: indica il numero di asset che sono stati ignorati durante il processo di trasferimento.

- `successededCount`: indica il numero di asset che hanno avuto successo durante il processo di trasferimento.
 - `totalCount`: indica il numero totale di asset coinvolti nel processo di trasferimento.
5. Inoltre, la chiamata API restituisce un elemento `reportURL`, che contiene un URL prefirmato. Se il processo di trasferimento presenta errori che richiedono un'analisi, puoi scaricare un rapporto di errore completo a questo URL.

AWS IoT SiteWise endpoint e quote

Le sezioni seguenti descrivono gli endpoint e le quote per AWS IoT SiteWise

Argomenti

- [AWS IoT SiteWise punti finali](#)
- [AWS IoT SiteWise quote](#)

AWS IoT SiteWise punti finali

La Riferimenti generali di AWS Guida elenca gli AWS IoT SiteWise endpoint di un Account AWS. Per ulteriori informazioni, consulta [AWS IoT SiteWise endpoint e quote nella Guida](#). Riferimenti generali di AWS

AWS IoT SiteWise quote

Le tabelle seguenti descrivono le quote in AWS IoT SiteWise. Per ulteriori informazioni sulle quote e su come richiedere aumenti delle quote, vedere le [quote AWS di servizio](#) nel Riferimenti generali di AWS. Per ulteriori informazioni sulle AWS IoT SiteWise quote, vedere [AWS IoT SiteWise Service Quotas](#) in Riferimenti generali di AWS.

Quote per AWS IoT SiteWise asset e modelli di asset

Risorsa	Descrizione	Quota	Regolabile
Numero di modelli di asset in ciascuno Regione AWS per ciascuno Account AWS	Il numero massimo di modelli di asset che è possibile creare in un anno Regione AWS per un Account AWS.	1000	Sì
Numero di asset in ogni modello di asset	Il numero massimo di asset che è possibile creare per ogni modello di asset.	10.000	Sì

Risorsa	Descrizione	Quota	Regolabile
Numero di risorse secondarie in ogni risorsa principale	Il numero massimo di risorse secondari e che è possibile associare a una risorsa principale.	2000	Sì
Profondità della struttura gerarchica del modello di asset	La profondità massima dell'albero della gerarchia degli asset per un modello di asset.	30	Sì
Numero di definizioni gerarchiche in ogni modello di asset	Il numero massimo di definizioni gerarchiche che è possibile avere in un modello di asset.		Sì
Numero di proprietà nel livello principale di ciascun modello di asset	Il numero massimo di <code>assetModeProperties</code> per ogni modello di asset. Questo conteggio non include <code>compositeModelProperties</code> . Questa quota si applica anche a qualsiasi risorsa unica creata da questo modello di asset.	500	Sì

Risorsa	Descrizione	Quota	Regolabile
Numero di proprietà in un modello di asset	Il numero massimo di proprietà di un modello di asset di tipo ASSET_MODEL o COMPONENT_MODEL . Questo numero viene determinato combinando le proprietà del modello di asset principale e di qualsiasi modello composito incluso component-model-based o in linea. Questa quota si applica anche a qualsiasi risorsa unica creata da questo modello di asset.	5000	Sì
Numero di proprietà in ogni modello composito	Il numero massimo di proprietà consentite per i modelli compositi . Inoltre, il numero massimo di proprietà consentite per un tipo di modello di asset COMPONENT_MODEL .	100	Sì

Risorsa	Descrizione	Quota	Regolabile
Profondità dell'albero delle proprietà in un modello di asset	Un modello con una proprietà di trasformazione C che consuma una proprietà di trasformazione B che consuma, a sua volta, una proprietà di misurazione A, ad esempio, ha una profondità di 3.	10	No
Numero di modelli di asset in ogni albero gerarchico	Il numero massimo di modelli di asset che è possibile includere in un singolo albero gerarchico.		Sì

Risorsa	Descrizione	Quota	Regolabile
Numero di proprietà direttamente dipendenti da un modello di asset	Questa quota limita il numero di proprietà che possono dipendere direttamente da una singola proprietà, come definito nelle espressioni formula di proprietà. Il numero di proprietà dipendenti deve essere maggiore del numero di proprietà direttamente dipendenti per un modello di asset. Richiedi un aumento per entrambe le quote se esistono più proprietà direttamente dipendenti rispetto a quelle dipendenti per un modello di asset.	20	Sì
Numero di proprietà dipendenti in un modello di asset	Questa quota limita il numero di proprietà che possono dipendere direttamente o indirettamente da una singola proprietà, come definito nelle espressioni formula di proprietà.	30	No

Risorsa	Descrizione	Quota	Regolabile
Numero di modelli compositi in un modello di asset	Il numero massimo di modelli compositi che è possibile avere su un singolo modello di asset.	50	Sì
Profondità del modello composito	La profondità massima dell'albero del modello composito in ogni modello di asset, inclusi i modelli in linea e component-model-based compositi.	2	Sì
Numero di modelli di asset unici che utilizzano lo stesso modello di component i	Il numero massimo di modelli di asset unici che dispongono di almeno un modello component-model-based composito che fa riferimento direttamente a un modello di asset specifico di tipo COMPONENT_MODEL.	20	Sì

Risorsa	Descrizione	Quota	Regolabile
Numero di variabili di proprietà in un'espressione di formula di proprietà	Ad esempio, ci sono due variabili di proprietà <code>power</code> e <code>temp</code> , nell'espressione <code>avg(power) + max(temp)</code> . Ciò vale anche per i risultati del calcolo delle trasformazioni.	10	No
Numero di funzioni in un'espressione di formula di proprietà	Ad esempio, nell'espressione sono presenti due funzioni <code>avg</code> e <code>max</code> , <code>avg(power) + max(temp)</code> .	10	No

Quote per i dati relativi alle proprietà AWS IoT SiteWise degli asset

Risorsa	Descrizione	Quota	Regolabile
Tasso di richiesta per le operazioni API dei dati delle proprietà asset	Il numero massimo di richieste API relative ai dati delle proprietà degli asset ogni secondo che puoi eseguire Regione AWS in ciascuna Account AWS di esse. Questa quota si applica alle operazioni API, quali <code>GetAssetPropertyValue</code> e <code>BatchPutAssets</code> .	1000	Sì

Risorsa	Descrizione	Quota	Regolabile
	<code>setPropertyValue</code> .		
Numero di punti dati al secondo per ogni qualità dei dati per ogni proprietà dell'asset	Questa quota si applica al numero massimo di punti dati timestamp-quality-value (TQV) con lo stesso timestamp in secondi per ogni qualità dei dati per ciascuna proprietà dell'asset. Puoi memorizzare questo numero di punti dati di buona qualità, qualità incerta e cattiva qualità per un dato secondo per ogni proprietà di risorsa.	10	No
Numero di BatchPutAssetPropertyValue voci inserite ogni secondo in ciascuna proprietà della risorsa per ciascuna di una. Regione AWS Account AWS	Il numero massimo di voci in ciascuna proprietà dell'asset BatchPutAssetPropertyValue da tutte le fonti, inclusi gateway SiteWise Edge, AWS IoT Core regole e chiamate API.	10	No

Risorsa	Descrizione	Quota	Regolabile
Percentuale di punti dati ingeriti	Il numero massimo di punti dati timestamp -quality-value (TQV) acquisiti al secondo in ciascuno per un. Regione AWS Account AWS	5000	Sì
Richiedi una tariffa per BatchGetAssetsPropertyAggregates	Il numero massimo di BatchGetAssetsPropertyAggregates richieste al secondo che è possibile eseguire Regione AWS in ognuna di esse Account AWS.	200	Sì
Tariffa di richiesta per BatchGetAssetsPropertyValue	Il numero massimo di BatchGetAssetsPropertyValue richieste al secondo che è possibile eseguire Regione AWS in ognuna di esse Account AWS.	500	Sì
Tariffa di richiesta per BatchGetAssetsPropertyValueHistory	Il numero massimo di BatchGetAssetsPropertyValueHistory richieste al secondo che puoi eseguire.	200	Sì

Risorsa	Descrizione	Quota	Regolabile
Numero di BatchPutAssetPropertyValues voci inserite ogni secondo per ogni risorsa, proprietà per ciascuna Regione AWS in un Account AWS	Questa quota si applica alle voci in ogni proprietà dell'asset BatchPutAssetPropertyValues provenienti da tutte le fonti, inclusi gateway SiteWise Edge, AWS IoT Core regole e chiamate API.	10	No
Frequenza delle GetAssetPropertyAggregates richieste e delle interrogazioni di BatchGetAssetPropertyAggregates immissione per ciascuna proprietà dell'asset	Il numero massimo di GetAssetPropertyAggregates richieste e BatchGetAssetPropertyAggregates immissioni totali per ogni proprietà dell'asset al secondo Regione AWS in ciascuna Account AWS di esse.	50	No

Risorsa	Descrizione	Quota	Regolabile
Frequenza delle GetAssetPropertyValue richieste e delle interrogazioni di BatchGetAssetPropertyValue immissione per ciascuna proprietà dell'asset	Il numero massimo di GetAssetPropertyValue richieste e BatchGetAssetPropertyValue immissioni totali per ogni proprietà dell'asset ogni secondo Regione AWS in ciascuna Account AWS di esse.	500	No
Frequenza delle GetAssetPropertyValueHistory richieste e delle interrogazioni di BatchGetAssetPropertyValueHistory immissione per ciascuna proprietà dell'asset	Il numero massimo di GetAssetPropertyValueHistory richieste e BatchGetAssetPropertyValueHistory immissioni totali per ogni proprietà dell'asset ogni secondo Regione AWS in ciascuna Account AWS di esse.	30	No

Risorsa	Descrizione	Quota	Regolabile
Frequenza delle GetInterp olatedAss etPropert yValues richieste	Il numero massimo di GetInterp olatedAss etPropert yValues richieste al secondo che è possibile eseguire Regione AWS in ognuna di esse Account AWS.	500	Sì
Numero di risultati in ogni GetInterp olatedAss etPropert yValues richiesta	Il numero massimo di risultati da restituire per una richiesta impaginat a. GetInterp olatedAss etPropert yValues	10	Sì

Risorsa	Descrizione	Quota	Regolabile
Velocità di punti dati recuperati da <code>GetAssetPropertyValueHistory</code> e <code>BatchGetAssetPropertyValues</code>	La velocità massima in byte (MB/secondo) dei punti dati recuperati ogni secondo per ciascuno di essi. Regione AWS Account AWS <code>GetAssetPropertyValueHistory</code> e <code>BatchGetAssetPropertyValues</code> Il payload di risposta valutato per questa quota utilizza campi Timestamp-Quality-Value (TQV) per ogni datapoint e arrotonda la dimensione in byte per ogni richiesta API al successivo incremento di 4 KB. Timestamp-quality-value I datapoint (TQV) recuperati ogni secondo variano a seconda del tipo di dati: Numero intero: fino a 5 milioni di TQV al secondo	100	Sì

Risorsa	Descrizione	Quota	Regolabile
	- Doppio: fino a 4 milioni di TQV al secondo - Booleano: fino a 6 milioni di TQV al secondo - Stringa: varia in base alla dimensione e del valore di ogni stringa.		

Quote per i gateway SiteWise Edge

Risorsa	Descrizione	Quota	Regolabile
Numero di gateway SiteWise Edge in ciascuno per un Regione AWS Account AWS	Il numero massimo di gateway SiteWise Edge che è possibile creare in un Regione AWS . Account AWS	100	Sì
Numero di sorgenti OPC UA in un SiteWise gateway Edge	Il numero massimo di sorgenti OPC UA che è possibile configurare in un gateway SiteWise Edge.	100	No
Numero totale di destinazioni in un gateway SiteWise Edge	Il numero massimo di destinazioni che è possibile configurare in un gateway SiteWise Edge.	100	No

Quote per AWS IoT SiteWise Monitor

Risorsa	Descrizione	Quota	Regolabile
Numero di portali in ciascuno Regione AWS per un Account AWS	Il numero massimo di portali SiteWise Monitor che è possibile creare in un Regione AWS . Account AWS	100	Sì
Numero di progetti in un portale	Il numero massimo di progetti che è possibile creare all'interno di un portale SiteWise Monitor.	100	Sì
Numero di dashboard in un progetto	Il numero massimo di dashboard che è possibile creare all'interno di un progetto in SiteWise Monitor.	100	Sì
Numero di risorse principali in un progetto	Il numero massimo di risorse di primo livello che è possibile aggiungere a un progetto in SiteWise Monitor.	1	No
Numero di visualizzazioni in una dashboard	Il numero massimo di elementi visivi (come diagrammi, grafici o tabelle) che puoi aggiungere a una	10	Sì

Risorsa	Descrizione	Quota	Regolabile
	dashboard in Monitor. SiteWise		
Numero di metriche in ogni visualizzazione del dashboard	Il numero massimo di metriche o punti dati che puoi visualizzare in una singola visualizzazione su una dashboard in Monitor. SiteWise	5	Sì
Numero di soglie per ogni visualizzazione del dashboard	Il numero massimo di livelli di soglia che è possibile impostare per ogni visualizzazione su una dashboard in Monitor. SiteWise	12	No

Quote per l'importazione e l'esportazione in AWS IoT SiteWise blocco di metadati

Risorsa	Descrizione	Quota	Regolabile
Numero di processi di trasferimento di metadati in coda	Il numero massimo di processi di trasferimento di PENDING metadati in coda.	10	Sì
Dimensione del file di importazione del processo di trasferimento dei metadati	La dimensione massima del file importato (in MB).	100	Sì
Numero di risorse di AWS IoT SiteWise	Il numero massimo di risorse di AWS IoT SiteWise importazione	5000	Sì

Risorsa	Descrizione	Quota	Regolabile
importazione in un lavoro	in un singolo lavoro. Una risorsa include risorse e modelli di asset.		
Numero di risorse di AWS IoT SiteWise esportazione in un lavoro	Il numero massimo di risorse di AWS IoT SiteWise esportazione in un singolo lavoro. Una risorsa include risorse e modelli di asset.	5000	Sì

Quote per l'importazione in AWS IoT SiteWise blocco di dati

Risorsa	Descrizione	Quota	Regolabile
Numero di processi di importazione in blocco in esecuzione	Il numero massimo di processi di importazione in blocco che possono essere eseguiti contemporaneamente.	100	No
Dimensione del file CSV	La dimensione massima del file CSV (in GB) in un processo di importazione in blocco.	10	No
Dimensione del file parquet non compresso	La dimensione massima del file (in MB) per un file parquet non compresso in un	256 MB	No

Risorsa	Descrizione	Quota	Regolabile
	processo di importazione in blocco.		
Dimensione del file CSV per l'inserimento nel buffer	La dimensione massima del file CSV (in MB) quando si utilizza l'ingestione bufferizzata in un processo di importazione di massa.	256 MB	No
Dimensione del gruppo di file di parquet non compresso	La dimensione massima di un gruppo di file di parquet non compresso.	64 MB	No
Numero di misurazioni uniche in un file per parquet	Il numero massimo di misurazioni uniche in una lima per parquet.	10000	No
Numero di giorni tra il timestamp del passato e quello odierno per l'ingestione bufferizzata	Il numero massimo di giorni tra un timestamp precedente e la data odierna quando si utilizza l'ingestione bufferizzata.	30	Sì
Richiedi la tariffa per ciascuno CreateBulkImportJobs Regione AWS Account AWS		10	Sì

Risorsa	Descrizione	Quota	Regolabile
Richiedi la tariffa <code>ListBulkImportJobs</code> per ciascuno Regione AWS di essi Account AWS		50	Sì
Richiedi la tariffa <code>DescribeBulkImportJobs</code> per ciascuno Regione AWS di essi Account AWS		50	Sì

Quote per la limitazione dell'API AWS IoT SiteWise Assistant

Quote per i limiti di limitazione dell'API AWS IoT SiteWise Assistant

Risorsa	Descrizione	Quota	Regolabile
Richiedi la tariffa di funzionamento <code>InvokeAssistant</code>	Il numero massimo di transazioni al minuto (TPM) che è possibile effettuare e sull' AWS IoT SiteWise <code>InvokeAssistant</code> API in un Account AWS. I limiti TPM si applicano a tutte le regioni supportate e sono regolabili in alcune aree.	10	No

Quote per il rilevamento delle anomalie

Le quote per il rilevamento delle anomalie sono condivise tra AWS IoT SiteWise Amazon Lookout for Equipment e Amazon. Per ulteriori informazioni, vedere [Quote per l'utilizzo di Lookout for Equipment](#).

Cronologia dei documenti per la Guida per AWS IoT SiteWise l'utente

La tabella seguente descrive la documentazione per questa versione di AWS IoT SiteWise.

- Versione API: 02-12-2019

Modifica	Descrizione	Data
Connect applicazioni esterne al broker EMQX con nomi utente e password	È stato aggiunto il supporto per la connessione di applicazioni esterne utilizzando nomi utente e password al broker AWS IoT Greengrass EMQX, che viene distribuito per impostazione predefinita durante la creazione di un nuovo gateway V3 abilitato per MQTT.	1 maggio 2025
Support per gateway V3 compatibili con MQTT su Edge SiteWise	Sono state aggiunte nuove funzionalità e sono stati rimossi i contenuti obsoleti • È stato aggiunto il supporto per i gateway V3 compatibili con MQTT. Configurazione della destinazione migliorata utilizzando filtri di percorso per sottoscrivere argomenti MQTT, tra cui l'inserimento di dati in tempo reale direttamente AWS IoT SiteWise o l'inserimento di dati nel buffer tramite Amazon S3.	26 febbraio 2025

- Rilasciata la versione 3.0.0 del collettore IoT SiteWise OPC UA e la versione 4.0.0 del componente editore IoT per. SiteWise AWS IoT Greengrass V2
- La versione precedente dei gateway SiteWise Edge ospitati autonomamente è stata ribattezzata Classic Streams, V2 gateway.
- Sono stati rimossi i riferimenti alla documentazione AWS IoT Greengrass V1 di SiteWise Edge perché non è più supportato per l'uso con. AWS IoT SiteWise

[Supporto per AWS IoT SiteWise Assistant](#)

È stato aggiunto il supporto per l' AWS IoT SiteWise Assistente, un assistente generativo basato sull'intelligenza artificiale.

18 novembre 2024

[Aggiunti timeout di sessione configurabili per Edge SiteWise APIs](#)

Sono state aggiunte impostazioni configurabili di timeout della sessione per gestire i periodi di inattività per End Edge. AWS OpsHub SiteWise APIs

31 ottobre 2024

[Aggiunte impostazioni proxy configurabili per Edge SiteWise APIs](#)

È stata aggiunta la gestione delle informazioni di trust store per abilitare il supporto proxy HTTPS per i gateway SiteWise Edge.

31 ottobre 2024

<u>Abilita CORS on for Edge SiteWise APIs</u>	È stato aggiunto il supporto CORS per SiteWise Edge APIs per consentire l'accesso sicuro alle applicazioni Web tra domini.	30 settembre 2024
<u>Support per CloudRail fonti di dati dei Litmus Edge partner</u>	È stato aggiunto il supporto per entrambe le fonti di dati CloudRail e Litmus Edge come partner.	5 settembre 2024
<u>Disponibilità generale per l'esecuzione di SiteWise Edge su Siemens Industrial Edge</u>	AWS IoT SiteWise ora supporta la disponibilità generale di SiteWise Edge in esecuzione sui dispositivi Siemens Industrial Edge.	24 luglio 2024
<u>Aggiunto il supporto per la configurazione del timestamp sulle sorgenti dati OPC UA</u>	AWS IoT SiteWise ora supporta la configurazione del timestamp per le fonti di dati OPC UA.	24 luglio 2024
<u>È stato aggiunto il supporto per la conversione dei tipi di dati su fonti di dati OPC UA</u>	AWS IoT SiteWise ora supporta la conversione dei tipi di dati per i tipi di dati OPC UA non supportati.	24 luglio 2024
<u>È stato aggiunto il supporto per l'esecuzione di un'anteprima di SiteWise Edge su Siemens Industrial Edge</u>	AWS IoT SiteWise ora supporta l'esecuzione di un'anteprima di SiteWise Edge sui dispositivi Siemens Industrial Edge.	26 novembre 2023

<u>È stato aggiunto il supporto per lo storage a livello caldo</u>	AWS IoT SiteWise ora supporta il warm storage, un livello di storage completamente gestito che consente ai clienti di archiviare e accedere in modo sicuro ai dati industriali.	15 novembre 2023
<u>È stato aggiunto il supporto per identificatori univoci definiti dall'utente</u>	AWS IoT SiteWise ora supporta l'uso di identificatori univoci definiti dall'utente per asset, modelli di asset, proprietà e gerarchie.	15 novembre 2023
<u>È stato aggiunto il supporto per il rilevamento di anomalie multivariate degli asset industriali</u>	AWS IoT SiteWise ora supporta il rilevamento multivariato delle anomalie degli asset industriali mediante l'integrazione di dati storici e in tempo reale sulle apparecchiature con Amazon Lookout for Equipment.	15 novembre 2023
<u>È stato aggiunto il supporto per l'inserimento conveniente e scalabile di dati di serie temporali in AWS IoT SiteWise</u>	AWS IoT SiteWise ora supporta l'inserimento conveniente e scalabile dei dati di serie temporali necessari per i casi d'uso analitici.	15 novembre 2023
<u>È stato aggiunto il supporto per l'importazione, l'esportazione e l'aggiornamento in blocco</u>	AWS IoT SiteWise ora supporta l'importazione, l'esportazione e l'aggiornamento in blocco dei metadati delle apparecchiature industriali.	15 novembre 2023

<u>È stato aggiunto il supporto per i componenti del modello di asset</u>	AWS IoT SiteWise ora supporta i componenti del modello Asset per aiutare i clienti industriali a creare componenti riutilizzabili.	15 novembre 2023
<u>Aggiunto supporto per l'applicazione dashboard IoT</u>	AWS IoT SiteWise ora supporta un'applicazione dashboard open source in cui è possibile visualizzare e interagire con i dati operativi.	15 novembre 2023
<u>Sono stati aggiornati i ruoli collegati ai servizi per AWS IoT SiteWise</u>	AWS IoT SiteWise dispone di nuovi ruoli collegati ai servizi e può eseguire una query di ricerca di metadati sul database. AWS IoT TwinMaker	6 novembre 2023
<u>Etichettatura aggiornata per le risorse del flusso di dati AWS IoT SiteWise</u>	È stato aggiunto il supporto per l'etichettatura delle risorse del flusso di dati.	18 agosto 2022
<u>Gateway SiteWise Edge aggiornati</u>	Ora puoi configurare l'editore per controllare quali dati vengono inviati dall'edge al cloud e l'ordine in cui vengono inviati al cloud.	12 gennaio 2022
<u>È stata aggiornata la AWS IoT SiteWise demo</u>	Ora puoi usare la demo per creare un portale SiteWise Monitor.	10 gennaio 2022
<u>Gestione aggiornata dello storage</u>	Ora puoi definire un periodo di conservazione per controllare per quanto tempo i dati vengono conservati nel livello più elevato.	29 novembre 2021

È stato aggiunto il supporto per la gestione del flusso di dati	Ora puoi inserire i dati AWS IoT SiteWise prima di creare modelli e asset di asset.	24 novembre 2021
Gerarchie di modelli di asset aggiornate	Un modello di asset figlio ora può essere associato a più modelli di asset principali.	28 ottobre 2021
Avvio della regione	Lanciato AWS IoT SiteWise in AWS GovCloud (Stati Uniti occidentali).	29 settembre 2021
Funzioni aggiornate	Sono state aggiunte le seguenti funzionalità • Nelle metriche, è possibile utilizzare espressioni annidate nelle funzioni di aggregazione e nelle funzioni temporali. • Nelle trasformazioni, potete utilizzare la funzione pretrigger () per recuperare e il valore di una variabile prima dell'aggiornamento della proprietà che ha attivato il calcolo della trasformazione corrente.	10 agosto 2021
Intervallo di tempo metrico personalizzato	È stato aggiunto il supporto per intervalli di tempo e offset personalizzati nelle metriche.	3 agosto 2021
Utilizzo AWS IoT SiteWise sul bordo	La funzione di elaborazione dei bordi è ora disponibile a livello generale.	29 luglio 2021

Esportazione di dati su Amazon S3	AWS IoT SiteWise ora può esportare dati su Amazon S3.	27 luglio 2021
Endpoint VPC ()AWS PrivateLink	L'endpoint VPC dell'interfaccia per le operazioni dell'API del piano di controllo è ora generalmente disponibile.	15 luglio 2021
Trasformazioni	Le trasformazioni ora possono inserire più variabili di proprietà degli asset.	8 luglio 2021
È stata aggiornata la funzione timestamp ()	Nelle trasformazioni, ora puoi fornire una variabile come argomento della funzione. <code>timestamp()</code>	16 giugno 2021
Disponibilità generale degli allarmi	La funzionalità degli allarmi è ora disponibile a livello generale.	27 maggio 2021
È stata rilasciata la versione 2 dell'adattatore di protocollo Modbus-TCP	È disponibile la versione 2 del connettore Modbus-TCP Protocol Adapter. Questa versione ha aggiunto il supporto per ASCII e le stringhe sorgente UTF8 codificate. ISO8859	24 maggio 2021

[Quote di servizio aggiornate](#)

Sono state aggiunte le seguenti quote per l'[GetInterpolatedAssetPropertyValues](#) API: frequenza delle GetInterpolatedAssetPropertyValues richieste, numero di risultati per GetInterpolatedAssetPropertyValues richiesta e numero di giorni tra la data di inizio passata e quella odierna di. GetInterpolatedAssetPropertyValues

29 aprile 2021

[Espressioni di formule aggiornate](#)

Sono stati aggiunti i seguenti operatori e funzioni:

22 aprile 2021

- Sono stati aggiunti i seguenti [operatori](#): < > <=>=,==,!=,!,and,or,enot.
- È stata aggiunta la seguente [funzione di confronto](#): neq(x, y).
- Sono state aggiunte le seguenti [funzioni di stringa](#): join(),format(),ef' '.

[Endpoint VPC \(AWS PrivateLink\)](#)

Sono state aggiunte informazioni su come stabilire una connessione privata tra il cloud privato virtuale (VPC) e il piano di AWS IoT SiteWise controllo APIs creando un endpoint VPC di interfaccia.

16 marzo 2021

Federazione IAM	Gli amministratori e gli utenti del portale SiteWise Monitor possono ora accedere ai portali assegnati con le proprie credenziali IAM.	16 marzo 2021
Lancio della regione	Lanciato AWS IoT SiteWise in Cina (Pechino).	3 febbraio 2021
Rilasciata la versione 10 del SiteWise connettore IoT	È disponibile la versione 10 del SiteWise connettore e IoT. Questa versione è configurata StreamManager per migliorare la gestione quando la connessione di origine viene persa e ristabilita. Questa versione accetta anche valori OPC UA con un ServerTimestamp quando non è SourceTimestamp disponibile.	22 gennaio 2021
Funzioni di data e ora	AWS IoT SiteWise ora supporta le funzioni di data e ora.	21 gennaio 2021
Sintassi della funzione	È ora possibile utilizzare la sintassi UFCS (Uniform Function Call Syntax) per le funzioni. AWS IoT SiteWise	11 gennaio 2021
Integrazione con Grafana	Sono state aggiunte informazioni su come visualizzare AWS IoT SiteWise i dati nelle dashboard di Grafana.	15 dicembre 2020

[AWS IoT SiteWise versione di funzionalità](#)

15 dicembre 2020

Ora puoi monitorare i tuoi dati con allarmi, elaborare i dati industriali sull'edge, utilizzare e Modbus TCP e le Ethernet/IP sorgenti del tuo gateway SiteWise Edge, filtrare i dati in entrata con bande morte e altro ancora.

- È stata aggiunta la sezione [Dati di monitoraggio con allarmi](#) che è possibile utilizzare per definire, configurare e rispondere agli allarmi. AWS IoT SiteWise
- È stata aggiunta la sezione [Elaborazione perimetrale](#) che è possibile utilizzare per configurare l'elaborazione dei dati industriali sui dispositivi periferici.
- Sono state aggiunte le sezioni [Modbus TCP ed EtherNet/IP](#) alla documentazione di origine del gateway Edge. SiteWise
- È stata aggiunta la sezione [di destinazione dell'origine](#) che è possibile utilizzare per personalizzare la destinazione di invio dei dati industriali in entrata.
- È stata aggiunta la sezione di [filtro OPC UA](#) che è possibile utilizzare per controllare la frequenza e il

	tipo di dati inviati al gateway SiteWise Edge dal server locale industriale.	
AWS IoT SiteWise ora supporta la gestione dei clienti CMKs	AWS IoT SiteWise ora supporta la crittografia con gestione dei clienti CMKs.	24 novembre 2020
Rilasciata la versione 8 del SiteWise connettore IoT	È disponibile la versione 8 del SiteWise connettore IoT. Questa versione migliora la stabilità quando il connettore presenta una connettività di rete intermittente.	19 novembre 2020
Utilizzo di stringhe e condizioni nelle espressioni delle formule	Sono state aggiunte informazioni sull'utilizzo di stringhe e funzioni condizionali nelle espressioni di formule per trasformazioni e metriche.	16 Novembre 2020
Acquisizione di dati tramite stream manager AWS IoT Greengrass	Sono state aggiunte informazioni su come importare dati IoT ad alto volume da fonti di dati locali utilizzando un dispositivo AWS IoT Greengrass perimetrale.	16 settembre 2020
Endpoint VPC (AWS PrivateLink)	Sono state aggiunte informazioni su come stabilire una connessione privata tra il cloud privato virtuale (VPC) e i AWS IoT SiteWise dati APIs creando un endpoint VPC di interfaccia.	4 settembre 2020

[Rilasciata la versione 7 del SiteWise connettore IoT](#)

È disponibile la versione 7 del SiteWise connettore IoT. Questa versione corregge un problema relativo alle metriche del gateway SiteWise Edge.

14 agosto 2020

[Creazione di utenti IAM Identity Center dalla console AWS IoT SiteWise](#)

Sono state aggiunte informazioni su come creare utenti IAM Identity Center nella AWS IoT SiteWise console. Ora puoi creare utenti IAM Identity Center quando assegni gli utenti a un portale nuovo o esistente. È stato aggiornato il tutorial sulla [visualizzazione e la condivisione dei dati dei parchi eolici](#) per utilizzare questa funzionalità. Questa modifica riduce il numero di passaggi del tutorial.

4 agosto 2020

[Risoluzione dei problemi migliorata del gateway SiteWise Edge](#)

Sono state aggiunte ulteriori informazioni su come risolvere i problemi di un gateway SiteWise Edge e su come [esportare il certificato client OPC UA](#) per un'origine.

18 giugno 2020

[Documentazione sulle attività della console](#)

Aggiunta della documentazione delle attività della console per [la modellazione degli asset industriali](#), [l'esecuzione di query sui dati delle proprietà di asset](#) e [l'interazione con altri servizi](#). Puoi seguire queste istruzioni per completare le attività nella console AWS IoT SiteWise .

11 giugno 2020

[Tutorial sull'analisi dei dati esportati](#)

[È stato aggiunto un tutorial che puoi seguire per imparare a usare Amazon Athena per analizzare i dati degli asset che hai esportato in Amazon S3 con il modello di funzionalità di esportazione. AWS CloudFormation](#)

27 maggio 2020

[Migliorato utilizzando espressioni di formule](#)

Sono state aggiunte informazioni dettagliate sul comportamento delle proprietà delle AWS IoT SiteWise formule e aggiunto un esempio di come contare i punti dati filtrati.

18 maggio 2020

[Rilasciata la versione 6 del SiteWise connettore IoT](#)

È disponibile la versione 6 del SiteWise connettore IoT. Questa versione aggiunge il supporto per le CloudWatch metriche e l'individuazione automatica di nuovi tag OPC UA. Ciò significa che non è necessario riavviare il gateway SiteWise Edge quando i tag cambiano per le sorgenti OPC UA. Questa versione del connettore richiede lo stream manager e il software AWS IoT Greengrass Core v1.10.0 o superiore.

29 aprile 2020

[AWS IoT SiteWise versione di funzionalità](#)

AWS IoT SiteWise versione di funzionalità. Ora puoi gestire i gateway SiteWise Edge con l'API, aggiungere il tuo logo ai portali, visualizzare le metriche dei gateway SiteWise Edge e altro ancora.

29 aprile 2020

- È stata aggiunta la sezione [Esportazione di dati in Amazon S3](#) con AWS CloudFormation un modello che puoi utilizzare per esportare nuovi valori di dati in un bucket Amazon S3.
- È stata aggiunta la sezione [Configurazione delle fonti di dati](#) che migliora la documentazione sui sorgenti del gateway SiteWise Edge e include il nuovo gateway Edge. SiteWise APIs
- È stata aggiunta la sezione relativa alle metriche del gateway SiteWise Edge che descrive le CloudWatch metriche pubblicate dai gateway SiteWise Edge.
- È stata aggiunta la sezione Configurazione di un gateway SiteWise Edge su Amazon con un AWS CloudFormation modello che è possibile utilizzare per configurare rapidamente le dipendenze del gateway

SiteWise Edge su un'istanza Amazon EC2.

- È stata aggiunta la sezione [sui ruoli del servizio di portale](#) che descrive la nuova funzionalità di autorizzazione dei SiteWise portali Monitor.
- È stata aggiornata la [documentazione del portale](#) per ruoli del servizio del portale e loghi del portale.
- È stata aggiunta la sezione [Taggare le risorse AWS IoT SiteWise](#).
- È stata aggiornata la sezione [Creazione di pannelli di controllo \(CLI\)](#) per la nuova struttura di definizione del pannello di controllo.
- È stata aggiunta la sezione relativa alla [sicurezza](#).

[Acquisizione di dati da AWS IoT Events](#)

Sono state aggiunte informazioni su come importare dati da AWS IoT Events quando si verifica un evento.

20 aprile 2020

[Visualizzazione e condivisione dei dati dei parchi eolici nel tutorial Monitor SiteWise](#)

È stato aggiunto un tutorial che puoi seguire per imparare a utilizzare AWS IoT SiteWise Monitor per visualizzare e condividere i dati degli asset.

12 marzo 2020

AWS IoT SiteWise concetti	È stato aggiunto un glossario di AWS IoT SiteWise concetti che è possibile utilizzare per conoscere il servizio e i termini più comuni.	5 marzo 2020
Istruzioni di AWS IoT Greengrass installazione rimosse	Sono state rimosse le istruzioni di installazione del software di AWS IoT Greengrass base dalla Guida AWS IoT SiteWise per l'utente. La AWS IoT Greengrass Developer Guide offre uno script di configurazione del dispositivo e istruzioni per la configurazione AWS IoT Greengrass su altre piattaforme come Amazon EC2 e Docker.	14 febbraio 2020
Migliore acquisizione dei dati tramite regole AWS IoT Core	Sono state aggiunte informazioni dettagliate sull' utilizzo e sulla risoluzione dei problemi dell'azione della AWS IoT SiteWise regola, che è possibile utilizzare per importare dati dai messaggi MQTT. AWS IoT Core	14 febbraio 2020
Rilasciata la versione 5 del SiteWise connettore IoT	È disponibile la versione 5 del SiteWise connettore IoT. Questa versione corregge un problema di compatibilità con il software AWS IoT Greengrass Core v1.9.4.	12 febbraio 2020

[Rilasciata la versione 4 del SiteWise connettore IoT](#)

È disponibile la versione 4 del SiteWise connettore IoT. Questa versione corregge un problema relativo alla riconnessione del server OPC UA.

7 febbraio 2020

[Asset industriali di modellazione ristrutturati](#)

È stata ristrutturata la sezione Aggiornamento di asset e modelli in più argomenti relativi alla modellazione degli asset industriali.

4 febbraio 2020

- [Stati di asset e modelli](#)
- [Gestisci i flussi di dati per AWS IoT SiteWise](#)
- [Aggiorna i valori degli attributi](#)
- [Associare e dissociare gli asset](#)
- [Aggiorna risorse e modelli](#)
- [Eliminare risorse e modelli in AWS IoT SiteWise](#)

[Tutorial sull'acquisizione di dati da oggetti AWS IoT](#)

È stato aggiunto un tutorial che puoi seguire per imparare a configurare un'azione della AWS IoT SiteWise regola per importare dati da una flotta di cose nuova o esistente. AWS IoT

4 febbraio 2020

Recupero di dati ristrutturato da AWS IoT SiteWise	Ha ristrutturato la sezione Recupero dei dati in due sezioni di primo livello: interrogazione dei valori e degli aggregati delle proprietà degli asset e interazione con altri servizi. AWS	21 gennaio 2020
Tutorial sulla pubblicazione degli aggiornamenti dei valori delle proprietà in Amazon DynamoDB	È stato aggiunto un tutorial che puoi seguire per imparare a utilizzare le notifiche sui valori delle proprietà per archiviare i dati degli asset in DynamoDB.	8 gennaio 2020
Usare le espressioni delle formule	Aggiunto il riferimento all'espressione di formula per organizzare le costanti e le funzioni disponibili per l'uso nelle proprietà di trasformazione e dei parametri. Sono state ristrutturate le proprietà dell'asset in argomenti separati per ogni tipo di proprietà.	7 gennaio 2020
Utilizzo dei filtri dei nodi OPC UA	Sono state aggiunte informazioni su come utilizzare i filtri dei nodi OPC UA per migliorare le prestazioni del gateway SiteWise Edge durante l'aggiunta di sorgenti gateway SiteWise Edge.	3 gennaio 2020

Aggiornamento di un connettore e	Sono state aggiunte informazioni su come aggiornare un gateway SiteWise Edge quando viene rilasciata una nuova versione del connettore.	30 dicembre 2019
Rilasciata la versione 3 del SiteWise connettore IoT	È disponibile la versione 3 del SiteWise connettore IoT. Questa versione rimuove il requisito di autorizzazione iot: *.	17 dicembre 2019
Rilasciata la versione 2 del SiteWise connettore IoT	È disponibile la versione 2 del SiteWise connettore IoT. Questa versione aggiunge il supporto per più risorse segrete OPC UA.	10 dicembre 2019
Creazione di dashboard (AWS CLI)	Sono state aggiunte informazioni su come creare una dashboard AWS IoT SiteWise Monitor utilizzando AWS CLI	6 dicembre 2019

[AWS IoT SiteWise è stata rilasciata la versione 2](#)

2 dicembre 2019

Anteprima rilasciata per la versione 2 di AWS IoT SiteWise. Ora puoi importare dati su OPC UA, MQTT e HTTP, modellare i dati in gerarchie di asset e visualizzare i tuoi dati con Monitor. SiteWise

- Riscritta la sezione dedicata alla [modellazione degli asset](#) per le modifiche apportate agli asset con i propri modelli e le relative gerarchie.
- È stata aggiornata la sezione relativa all'inserimento [dei dati per includere le fasi del connettore e le sezioni relative](#) all'inserimento dei dati non relative al gateway AWS IoT Greengrass .
- È stata aggiunta la [AWS IoT SiteWise Monitor](#) sezione e una [guida all'applicazione separata](#) che mostra come utilizzare l'applicazione web Monitor. SiteWise
- Aggiunte le sezioni [Interrogare i dati da AWS IoT SiteWise](#) e [Interagisci con altri AWS servizi](#).
- Riscritta la sezione [Nozioni di base](#) in conformità

all'aggiornamento dell'esperienza demo.

[AWS IoT SiteWise è stata rilasciata la versione 1](#)

Rilasciata l'anteprima iniziale per la versione 1 di AWS IoT SiteWise.

25 febbraio 2019

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