



Guida per gli sviluppatori

Amazon WorkDocs



Amazon WorkDocs: Guida per gli sviluppatori

Copyright © 2024 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 Amazon WorkDocs?	1
Accedere ad Amazon WorkDocs	1
Prezzi	1
Risorse	1
Nozioni di base	3
Connettersi ad Amazon WorkDocs con le credenziali utente IAM	3
Connettersi ad Amazon WorkDocs assumendo un ruolo	5
Caricamento di un documento	8
Download di un documento	9
Configurazione delle notifiche	10
Creazione di un utente	13
Concessione delle autorizzazioni agli utenti per una risorsa	14
Autenticazione e controllo degli accessi per le applicazioni amministrative	15
Concessione delle autorizzazioni agli sviluppatori per l' WorkDocs API Amazon	15
Concedere agli sviluppatori di terze parti l'autorizzazione per le WorkDocs API di Amazon	16
Concedere agli utenti il permesso di assumere un ruolo IAM	18
Limitazione dell'accesso a una WorkDocs istanza Amazon specifica	18
Autenticazione e controllo degli accessi per le applicazioni utente	20
Concessione delle autorizzazioni per chiamare Amazon WorkDocs API	20
Utilizzo degli ID delle cartelle nelle chiamate API	22
Creazione di un'applicazione	23
Ambiti delle applicazioni	23
Autorizzazione	24
Richiamo Amazon WorkDocs API	25
Amazon WorkDocs Gestore di contenuto	28
Costruzione di Amazon WorkDocs Gestore di contenuto	28
Download di un documento	29
Caricamento di un documento	30
Glossario AWS	31
.....	xxxii

Che cos'è Amazon WorkDocs?

Amazon WorkDocs è un sistema di archiviazione, collaborazione e condivisione di documenti. Amazon WorkDocs è completamente gestito, sicuro e scalabile a livello aziendale. Fornisce solidi controlli amministrativi e funzionalità di feedback che aiutano a migliorare la produttività degli utenti. I tuoi file vengono archiviati nel [cloud](#), in modo sicuro. I file dei tuoi utenti sono visibili solo a loro e ai collaboratori e visualizzatori designati. Gli altri membri dell'organizzazione non hanno accesso ai file degli altri utenti, a meno che non gli venga concesso l'accesso specificamente.

Gli utenti possono condividere i loro file con altri membri dell'organizzazione a scopi di collaborazione o revisione. Le applicazioni WorkDocs client Amazon possono essere utilizzate per visualizzare molti tipi diversi di file, a seconda del tipo di supporto Internet del file. Amazon WorkDocs supporta tutti i formati di documenti e immagini più comuni e il supporto per altri tipi di file multimediali viene costantemente aggiunto.

Per ulteriori informazioni, consulta [Amazon WorkDocs](#).

Accedere ad Amazon WorkDocs

Gli utenti finali usano le applicazioni client per accedere ai file. Gli utenti non amministrativi non devono mai utilizzare la WorkDocs console Amazon o la dashboard di amministrazione. Amazon WorkDocs offre diverse applicazioni e utilità per i client:

- Un'applicazione Web usata per la gestione e la revisione dei documenti.
- App native per dispositivi mobili usate per la revisione dei documenti.
- Amazon WorkDocs Drive viene utilizzato per sincronizzare una cartella sul desktop Mac o Windows con i WorkDocs file Amazon.

Prezzi

Con Amazon WorkDocs, non sono previsti costi o impegni anticipati. Paghi solo in base agli account utente attivi e allo spazio di archiviazione che utilizzi. Per ulteriori informazioni, [consulta](#).

Risorse

Le seguenti risorse correlate possono rivelarsi utili durante l'utilizzo di questo servizio.

- [Corsi e workshop](#): collegamenti a corsi basati su ruoli e di specializzazione nonché a corsi gestiti dall'utente per affinare le proprie competenze su AWS e acquisire esperienza pratica.
- [Centro sviluppatori AWS](#): esplora i tutorial, scarica gli strumenti e scopri gli eventi destinati agli sviluppatori AWS.
- [Strumenti per sviluppatori AWS](#): collegamenti a strumenti per sviluppatori, SDK, kit di strumenti IDE e strumenti a riga di comando per lo sviluppo e la gestione delle applicazioni AWS.
- [Centro risorse per le nozioni di base](#): scopri come configurare il tuo Account AWS, unisciti alla community AWS e lancia la tua prima applicazione.
- [Tutorial pratici](#): segui i step-by-step tutorial per avviare la tua prima applicazione su AWS.
- [Whitepaper AWS](#): collegamenti a un elenco completo di whitepaper tecnici AWS, relativi ad argomenti come architettura, sicurezza ed economia, creati da AWS Solutions Architect o da altri esperti tecnici.
- [AWS SupportCentro](#) : il centro in cui creare e gestire i tuoi casi AWS Support. Include inoltre link ad altre risorse utili, quali forum, domande frequenti di tipo tecnico, stato d'integrità del servizio e AWS Trusted Advisor.
- [AWS Support](#): pagina Web principale che include le informazioni su AWS Support one-on-one, un canale di assistenza rapida che aiuta a creare ed eseguire applicazioni nel cloud.
- [Contatti](#) - Un punto di contatto centrale per richieste relative a fatturazione, account, eventi, uso illecito e altre questioni relative ad AWS.
- [AWS Termini di utilizzo del sito](#): informazioni dettagliate sul copyright e i marchi, l'account, la licenza, l'accesso al sito e altri argomenti.

Nozioni di base

I seguenti frammenti di codice possono aiutarti a iniziare a utilizzare Amazon WorkDocs SDK.

Note

Per una maggiore sicurezza, crea utenti federati anziché utenti IAM quando possibile.

Esempi

- [Connettiti ad Amazon WorkDocs con le credenziali utente IAM e le query per gli utenti](#)
- [Connettersi ad Amazon WorkDocs assumendo un ruolo](#)
- [Caricamento di un documento](#)
- [Download di un documento](#)
- [Configurazione delle notifiche](#)
- [Creazione di un utente](#)
- [Concessione delle autorizzazioni agli utenti per una risorsa](#)

Connettiti ad Amazon WorkDocs con le credenziali utente IAM e le query per gli utenti

Il codice seguente mostra come utilizzare le credenziali API di un utente IAM per effettuare chiamate API. In questo caso l'utente API e il WorkDocs sito Amazon appartengono allo stesso AWS account.

Note

Per una maggiore sicurezza, crea utenti federati anziché utenti IAM quando possibile.

Assicurati che all'utente IAM sia stato concesso l'accesso all' WorkDocs API Amazon tramite una policy IAM appropriata.

L'esempio di codice utilizza l'[DescribeUsers](#) API per cercare utenti e ottenere metadati per gli utenti. I metadati utente forniscono dettagli come nome, cognome, ID utente e ID della cartella principale.

L'ID della cartella principale è particolarmente utile se si desidera eseguire operazioni di caricamento o download di contenuti per conto dell'utente.

Il codice richiede l'ottenimento di un Amazon WorkDocs Organization ID.

Segui questi passaggi per ottenere un ID WorkDocs dell'organizzazione Amazon dalla AWS console:

Per ottenere un ID organizzazione

1. Nel riquadro di navigazione [AWS Directory Service console](#), scegliere Directories (Directory).
2. Annota il valore dell'ID di directory che corrisponde al tuo WorkDocs sito Amazon. Questo è l'ID dell'organizzazione del sito.

L'esempio seguente mostra come utilizzare le credenziali IAM per effettuare chiamate API.

```
import java.util.ArrayList;
import java.util.List;

import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.workdocs.AmazonWorkDocs;
import com.amazonaws.services.workdocs.AmazonWorkDocsClient;
import com.amazonaws.services.workdocs.model.DescribeUsersRequest;
import com.amazonaws.services.workdocs.model.DescribeUsersResult;
import com.amazonaws.services.workdocs.model.User;

public class GetUserDemo {

    public static void main(String[] args) throws Exception {
        AWSCredentials longTermCredentials =
            new BasicAWSCredentials("accessKey", "secretKey");
        AWSStaticCredentialsProvider staticCredentialProvider =
            new AWSStaticCredentialsProvider(longTermCredentials);

        AmazonWorkDocs workDocs =
            AmazonWorkDocsClient.builder().withCredentials(staticCredentialProvider)
                .withRegion(Regions.US_WEST_2).build();

        List<User> wdUsers = new ArrayList<>();
        DescribeUsersRequest request = new DescribeUsersRequest();
```

```
// The OrganizationId used here is an example and it should be replaced
// with the OrganizationId of your WorkDocs site.
request.setOrganizationId("d-123456789c");
request.setQuery("joe");

String marker = null;
do {
    request.setMarker(marker);
    DescribeUsersResult result = workDocs.describeUsers(request);
    wdUsers.addAll(result.getUsers());
    marker = result.getMarker();
} while (marker != null);

System.out.println("List of users matching the query string: joe ");

for (User wdUser : wdUsers) {
    System.out.printf("Firstname:%s | Lastname:%s | Email:%s | root-folder-id:%s\n",
        wdUser.getGivenName(), wdUser.getSurname(), wdUser.getEmailAddress(),
        wdUser.getRootFolderId());
}
}
```

Connettersi ad Amazon WorkDocs assumendo un ruolo

Questo esempio utilizza l'SDK AWS Java per assumere un ruolo e utilizzare le credenziali di sicurezza temporanee del ruolo per accedere ad Amazon. WorkDocs L'esempio di codice utilizza l'[DescribeFolderContents](#) API per elencare gli elementi nella cartella di un utente.

```
import java.util.ArrayList;
import java.util.List;

import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.auth.BasicSessionCredentials;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.securitytoken.AWSSecurityTokenService;
import com.amazonaws.services.securitytoken.AWSSecurityTokenServiceClientBuilder;
import com.amazonaws.services.securitytoken.model.AssumeRoleRequest;
import com.amazonaws.services.securitytoken.model.AssumeRoleResult;
```



```
import com.amazonaws.services.workdocs.AmazonWorkDocs;
import com.amazonaws.services.workdocs.AmazonWorkDocsClient;
import com.amazonaws.services.workdocs.model.DescribeFolderContentsRequest;
import com.amazonaws.services.workdocs.model.DescribeFolderContentsResult;
import com.amazonaws.services.workdocs.model.DocumentMetadata;
import com.amazonaws.services.workdocs.model.FolderMetadata;

public class AssumeRoleDemo {
    private static final String DEMO_ROLE_ARN = "arn:aws:iam::111122223333:role/workdocs-readonly-role";
    private static AmazonWorkDocs workDocs;

    public static void main(String[] args) throws Exception {

        AWSCredentials longTermCredentials =
            new BasicAWSCredentials("accessKey", "secretKey");

        // Use developer's long-term credentials to call the AWS Security Token Service (STS)
        // AssumeRole API, specifying the ARN for the role workdocs-readonly-role in
        // 3rd party AWS account.

        AWSSecurityTokenService stsClient =
            AWSSecurityTokenServiceClientBuilder.standard()
                .withCredentials(new AWSStaticCredentialsProvider(longTermCredentials))
                .withRegion(Regions.DEFAULT_REGION.getName()).build();

        // If you are accessing a 3rd party account, set ExternalId
        // on assumeRequest using the withExternalId() function.
        AssumeRoleRequest assumeRequest =
            new AssumeRoleRequest().withRoleArn(DEMO_ROLE_ARN).withDurationSeconds(3600)
                .withRoleSessionName("demo");

        AssumeRoleResult assumeResult = stsClient.assumeRole(assumeRequest);

        // AssumeRole returns temporary security credentials for the
        // workdocs-readonly-role

        BasicSessionCredentials temporaryCredentials =
            new BasicSessionCredentials(assumeResult.getCredentials().getAccessKeyId(),
            assumeResult
                .getCredentials().getSecretAccessKey(),
            assumeResult.getCredentials().getSessionToken());
```

```
// Build WorkDocs client using the temporary credentials.
workDocs =
    AmazonWorkDocsClient.builder()
        .withCredentials(new AWSStaticCredentialsProvider(temporaryCredentials))
        .withRegion(Regions.US_WEST_2).build();

// Invoke WorkDocs service calls using the temporary security credentials
// obtained for workdocs-readonly-role. In this case a call has been made
// to get metadata of Folders and Documents present in a user's root folder.

describeFolder("root-folder-id");
}

private static void describeFolder(String folderId) {
    DescribeFolderContentsRequest request = new DescribeFolderContentsRequest();
    request.setFolderId(folderId);
    request.setLimit(2);
    List<DocumentMetadata> documents = new ArrayList<>();
    List<FolderMetadata> folders = new ArrayList<>();

    String marker = null;

    do {
        request.setMarker(marker);
        DescribeFolderContentsResult result = workDocs.describeFolderContents(request);
        documents.addAll(result.getDocuments());
        folders.addAll(result.getFolders());
        marker = result.getMarker();
    } while (marker != null);

    for (FolderMetadata folder : folders)
        System.out.println("Folder:" + folder.getName());
    for (DocumentMetadata document : documents)
        System.out.println("Document:" + document.getLatestVersionMetadata().getName());
}
}
```

Caricamento di un documento

Note

Devi essere uno sviluppatore di software per completare i passaggi descritti in questa sezione. Per informazioni sull'utilizzo di Amazon WorkDocs per caricare file, consulta [Uploading files](#) nella Amazon WorkDocs User Guide.

Utilizza la seguente procedura per caricare un documento su Amazon WorkDocs.

Per caricare un documento

1. Creare un'istanza della classe `AmazonWorkDocsClient` come segue:

Se utilizzi le credenziali utente IAM, consulta. [Connettiti ad Amazon WorkDocs con le credenziali utente IAM e le query per gli utenti](#) Se assumi un ruolo IAM, consulta [Connettersi ad Amazon WorkDocs assumendo un ruolo](#) per ulteriori informazioni.

Note

Per una maggiore sicurezza, crea utenti federati anziché utenti IAM quando possibile.

```
AWSCredentials longTermCredentials =
    new BasicAWSCredentials("accessKey", "secretKey");
AWSStaticCredentialsProvider staticCredentialProvider =
    new AWSStaticCredentialsProvider(longTermCredentials);

// Use the region specific to your WorkDocs site.
AmazonWorkDocs amazonWorkDocsClient =
    AmazonWorkDocsClient.builder().withCredentials(staticCredentialProvider)
        .withRegion(Regions.US_WEST_2).build();
```

2. Ottenere l'URL firmato per il caricamento come segue:

```
InitiateDocumentVersionUploadRequest request = new
    InitiateDocumentVersionUploadRequest();
request.setParentFolderId("parent-folder-id");
request.setName("my-document-name");
```

```
request.setContentType("application/octet-stream");
InitiateDocumentVersionUploadResult result =
    amazonWorkDocsClient.initiateDocumentVersionUpload(request);
UploadMetadata uploadMetadata = result.getUploadMetadata();
String documentId = result.getMetadata().getId();
String documentVersionId = result.getMetadata().getLatestVersionMetadata().getId();
String uploadUrl = uploadMetadata.getUploadUrl();
```

3. Caricare il documento utilizzando l'URL firmato come segue:

```
URL url = new URL(uploadUrl);
URLConnection connection = (URLConnection) url.openConnection();
connection.setDoOutput(true);
connection.setRequestMethod("PUT");
// Content-Type supplied here should match with the Content-Type set
// in the InitiateDocumentVersionUpload request.
connection.setRequestProperty("Content-Type", "application/octet-stream");
connection.setRequestProperty("x-amz-server-side-encryption", "AES256");
File file = new File("/path/to/file.txt");
FileInputStream fileInputStream = new FileInputStream(file);
OutputStream outputStream = connection.getOutputStream();
com.amazonaws.util.IOUtils.copy(fileInputStream, outputStream);
connection.getResponseCode();
```

4. Completare il processo di caricamento modificando lo stato del documento in ACTIVE come segue:

```
UpdateDocumentVersionRequest request = new UpdateDocumentVersionRequest();
request.setDocumentId("document-id");
request.setVersionId("document-version-id");
request.setVersionStatus(DocumentVersionStatus.ACTIVE);
amazonWorkDocsClient.updateDocumentVersion(request);
```

Download di un documento

Note

Devi essere uno sviluppatore di software per completare i passaggi descritti in questa sezione. Per informazioni sull'utilizzo di Amazon WorkDocs per scaricare file, consulta [Downloading files](#) nella Amazon WorkDocs User Guide.

Per scaricare un documento da Amazon WorkDocs, ottieni un URL per il download come segue, quindi utilizza le azioni API fornite dalla tua piattaforma di sviluppo per scaricare il file utilizzando l'URL.

```
GetDocumentVersionRequest request = new GetDocumentVersionRequest();
request.setDocumentId("document-id");
request.setVersionId("document-version-id");
request.setFields("SOURCE");
GetDocumentVersionResult result = amazonWorkDocsClient.getDocumentVersion(request);
String downloadUrl =
    result.getMetadata().getSource().get(DocumentSourceType.ORIGINAL.name());
```

Configurazione delle notifiche

Segui questa procedura per configurare le notifiche:

1. Imposta le autorizzazioni di utente o ruolo IAM per consentire al chiamante l'accesso alle API di gestione delle sottoscrizioni alle notifiche.
2. Chiama le API di sottoscrizione alle notifiche per abilitare o disabilitare la pubblicazione di messaggi SNS sul tuo endpoint.

Note

Per una maggiore sicurezza, crea utenti federati anziché utenti IAM quando possibile.

Per impostare le autorizzazioni degli utenti IAM

- Utilizza la console IAM per impostare le seguenti autorizzazioni per l'utente:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "workdocs:CreateNotificationSubscription",
        "workdocs>DeleteNotificationSubscription",
        "workdocs:DescribeNotificationSubscriptions"
      ]
    }
  ]
}
```

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

Per abilitare le notifiche

L'abilitazione delle notifiche ti consente di chiamare [CreateNotificationSubscription](#) dopo esserti abbonato alle notifiche.

1. Apri la WorkDocs console Amazon all'[indirizzo https://console.aws.amazon.com/zocalo/](https://console.aws.amazon.com/zocalo/).
2. Nella pagina Gestisci i tuoi WorkDocs siti, seleziona la directory desiderata e scegli Azioni, quindi Gestisci notifiche.
3. Nella pagina Manage Notifications (Gestisci notifiche) scegliere Modify (Modifica).
4. Inserisci l'ARN per l'utente o il ruolo a cui desideri consentire la ricezione di notifiche dal tuo sito Amazon WorkDocs .

Per informazioni su come abilitare Amazon WorkDocs a utilizzare le notifiche, consulta [Using the Amazon WorkDocs API with the AWS SDK for Python e AWS Lambda](#). Dopo aver abilitato le notifiche, tu e il tuo utente potete abbonarvi ad esse.

Per sottoscrivere WorkDocs le notifiche

1. Prepara il tuo endpoint per elaborare i messaggi Amazon SNS. Per ulteriori informazioni, consulta [Fanout to HTTP/S endpoint](#) nella Amazon Simple Notification Service Developer Guide.

 Important

SNS invia un messaggio di conferma all'endpoint configurato. È necessario confermare questo messaggio per ricevere le notifiche. Inoltre, se hai bisogno di moduli crittografici convalidati FIPS 140-2 per accedere ad 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 \(FIPS\) 140-2](#).

2. Esegui questa operazione:

- Ottieni un ID dell'organizzazione

1. Nel riquadro di navigazione della [console AWS Directory Service](#), seleziona Directories.
 2. L'ID di directory corrispondente al tuo WorkDocs sito Amazon funge anche da ID dell'organizzazione per quel sito.
- Creare la richiesta di sottoscrizione come segue:

```
CreateNotificationSubscriptionRequest request = new
    CreateNotificationSubscriptionRequest();
request.setOrganizationId("d-1234567890");
request.setProtocol(SubscriptionProtocolType.Https);
request.setEndpoint("https://my-webhook-service.com/webhook");
request.setSubscriptionType(SubscriptionType.ALL);
CreateNotificationSubscriptionResult result =
    amazonWorkDocsClient.createNotificationSubscription(request);
System.out.println("WorkDocs notifications subscription-id: "
    result.getSubscription().getSubscriptionId());
```

Notifiche SNS

Il messaggio include le informazioni seguenti:

- `organizationId`— L'ID dell'organizzazione.
- `parentEntityType`— Il tipo di genitore (Document| DocumentVersion |Folder).
- `parentEntityId`— L'ID del genitore.
- `entityType`— Il tipo di entità (Document| DocumentVersion |Folder).
- `entityId`— L'ID dell'entità.
- azione: l'azione, che può corrispondere a uno dei seguenti valori:
 - `delete_document`
 - `move_document`
 - `recycle_document`
 - `rename_document`
 - `revoke_share_document`
 - `share_document`
 - `upload_document_version`

Per disabilitare le notifiche

1. Apri la WorkDocs console Amazon all'[indirizzo https://console.aws.amazon.com/zocalo/](https://console.aws.amazon.com/zocalo/).
2. Nella pagina Gestisci i tuoi WorkDocs siti, seleziona la directory desiderata e scegli Azioni, quindi Gestisci notifiche.
3. Nella pagina Manage Notifications (Gestisci notifiche) selezionare l'ARN per cui si desidera disabilitare le notifiche e scegliere Disable Notifications (Disabilita notifiche).

Creazione di un utente

L'esempio seguente mostra come creare un utente in Amazon WorkDocs.

Note

Questa non è un'operazione valida per una configurazione con AD Connector. Per creare un utente nella configurazione Connected AD, l'utente deve essere già presente nella directory aziendale. Quindi, devi effettuare una chiamata all'[ActivateUser](#) API per attivare l'utente in Amazon WorkDocs.

L'esempio seguente mostra come creare un utente con una quota di archiviazione di 1 gigabyte.

```
CreateUserRequest request = new CreateUserRequest();
    request.setGivenName("GivenName");
    request.setOrganizationId("d-12345678c4");
    // Passwords should:
    //   Be between 8 and 64 characters
    //   Contain three of the four below:
    //   A Lowercase Character
    //   An Uppercase Character
    //   A Number
    //   A Special Character
    request.setPassword("Badpa$$w0rd");
    request.setSurname("surname");
    request.setUsername("UserName");
    StorageRuleType storageRule = new StorageRuleType();
    storageRule.setStorageType(StorageType.QUOTA);
    storageRule.setStorageAllocatedInBytes(new Long(1048576L));
    request.setStorageRule(storageRule);
```



```
CreateUserResult result = workDocsClient.createUser(request);
```

Segui questi passaggi per ottenere un ID WorkDocs dell'organizzazione Amazon dalla AWS console:

Per ottenere un ID organizzazione

1. Nel riquadro di navigazione [AWS Directory Service console](#), scegliere Directories (Directory).
2. Annota il valore dell'ID di directory che corrisponde al tuo WorkDocs sito Amazon. Questo è l'ID dell'organizzazione del sito.

Concessione delle autorizzazioni agli utenti per una risorsa

L'esempio seguente mostra come utilizzare l'[AddResourcePermissions](#) API per concedere CONTRIBUTOR le autorizzazioni a una USER risorsa. Puoi anche utilizzare l'API per concedere autorizzazioni a un utente o a un gruppo su una cartella o un documento.

```
AddResourcePermissionsRequest request = new AddResourcePermissionsRequest();
    request.setResourceId("resource-id");
    Collection<SharePrincipal> principals = new ArrayList<>();
    SharePrincipal principal = new SharePrincipal();
    principal.setId("user-id");
    principal.setType(PrincipalType.USER);
    principal.setRole(RoleType.CONTRIBUTOR);
    principals.add(principal);
    request.setPrincipals(principals);
    AddResourcePermissionsResult result =
workDocsClient.addResourcePermissions(request);
```

Autenticazione e controllo degli accessi per le applicazioni amministrative

Le API WorkDocs amministrative di Amazon sono autenticate e autorizzate tramite policy IAM. Gli amministratori IAM possono creare una policy IAM e collegarla a un ruolo o utente IAM.

Di seguito vengono riportati degli esempi:

Processi

- [Concessione delle autorizzazioni agli sviluppatori per l' WorkDocs API Amazon](#)
- [Concedere agli sviluppatori di terze parti l'autorizzazione per le WorkDocs API di Amazon](#)
- [Concedere agli utenti il permesso di assumere un ruolo IAM](#)
- [Limitazione dell'accesso a una WorkDocs istanza Amazon specifica](#)

Concessione delle autorizzazioni agli sviluppatori per l' WorkDocs API Amazon

Note

Per una maggiore sicurezza, crea utenti federati anziché utenti IAM quando possibile.

Se sei un amministratore IAM, puoi concedere l'accesso all' WorkDocs API Amazon a un utente IAM dallo stesso AWS account. A tale scopo, crea una politica di autorizzazione delle WorkDocs API Amazon e allegala all'utente IAM. La seguente politica API concede l'autorizzazione di sola lettura alle varie Describe API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WorkDocsAPIReadOnly",
      "Effect": "Allow",
      "Action": [
        "workdocs:Get*"
      ]
    }
  ]
}
```

```
    "workdocs:Describe*"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Concedere agli sviluppatori di terze parti l'autorizzazione per le WorkDocs API di Amazon

È possibile concedere l'accesso agli sviluppatori di terze parti o agli utenti che usano un altro account AWS. A tale scopo, crea un ruolo IAM e allega le politiche di autorizzazione delle WorkDocs API Amazon.

Questo tipo di accesso è richiesto negli scenari seguenti:

- Lo sviluppatore appartiene alla stessa organizzazione ma l'AWSaccount dello sviluppatore è diverso dall' WorkDocs AWSaccount Amazon.
- Quando un'azienda desidera concedere l'accesso alle WorkDocs API Amazon a sviluppatori di applicazioni di terze parti.

In entrambi questi scenari, sono coinvolti dueAWS account, un accountAWS sviluppatore e un altro account che ospita un WorkDocs sito Amazon.

Lo sviluppatore dovrà fornire le seguenti informazioni in modo che l'amministratore dell'account possa creare il ruolo IAM:

- ID dell'account AWS
- External ID univoco che verrà utilizzato dal cliente per identificarti. Per ulteriori informazioni, vedi l'argomento su [come utilizzare un ID esterno quando si concede a terze parti l'accesso alle proprie risorseAWS](#).
- Un elenco di WorkDocs API Amazon a cui la tua applicazione deve accedere. Il controllo delle policy basato su IAM offre un controllo granulare, la possibilità di definire policy di autorizzazione o negazione a livello di singola API. Per l'elenco delle WorkDocs API Amazon, consulta [Amazon WorkDocs API Reference](#).

Di seguito viene descritta la procedura di configurazione di IAM per l'accesso multiaccount.

Per configurare IAM per l'accesso tra account

1. Crea una politica di autorizzazione WorkDocs delle API Amazon, chiamala `WorkDocsAPIReadOnly` politica.
2. Crea un nuovo ruolo nella console IAM dell'AWS account che ospita il WorkDocs sito Amazon:
 - a. Accedi alla AWS Management Console e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
 - b. Nel riquadro di navigazione della console fare clic su Roles (Ruoli), quindi su Create New Role (Crea nuovo ruolo).
 - c. In Role name (Nome ruolo) digitare un nome che identifichi lo scopo del ruolo, ad esempio `workdocs_app_role`. I nomi dei ruoli devono essere univoci all'interno dell'account AWS. Dopo avere inserito il nome, fare clic su Next Step (Fase successiva).
 - d. Nella pagina Select Role Type (Selezionare il tipo di ruolo) selezionare la sezione Role for Cross-Account Access (Ruolo per accesso multiaccount), quindi selezionare il tipo di ruolo che si desidera creare:
 - Seleziona Fornisci l'accesso tra AWS gli account di tua proprietà se sei l'amministratore sia dell'account utente che dell'account della risorsa o entrambi gli account appartengono alla stessa società. Questa è l'opzione da selezionare anche nei casi in cui gli utenti, il ruolo e la risorsa a cui si deve accedere si trovano tutti nello stesso account.
 - Seleziona Fornisci l'accesso tra il tuo AWS account e un AWS account di terze parti se sei l'amministratore dell'account proprietario del WorkDocs sito Amazon e desideri concedere le autorizzazioni agli utenti da un account sviluppatore dell'applicazione. Per questa opzione deve essere specificato un ID esterno (fornito dalla terza parte) per controllare le situazioni in cui la terza parte può utilizzare il ruolo per accedere alle risorse. Per ulteriori informazioni, vedi l'argomento su [come utilizzare un ID esterno quando si concede a terze parti l'accesso alle proprie risorse AWS](#).
 - e. Nella pagina successiva, digitare l'ID AWS account.
 - f. Fare clic su Next Step (Fase successiva) per collegare una policy.
3. Nella pagina Allega policy, cerca la politica di autorizzazione dell' WorkDocs API Amazon creata in precedenza, seleziona la casella accanto alla politica e fai clic su Next Step.
4. Rivedere i dettagli, copiare l'ARN del ruolo per riferimento futuro e fare clic su Create Role (Crea ruolo) per completare la creazione del ruolo.

5. Condividere l'ARN del ruolo con lo sviluppatore. Di seguito è riportato un esempio di ARN del ruolo:

```
arn:aws:iam::AWS-ACCOUNT-ID:role/workdocs_app_role
```

Concedere agli utenti il permesso di assumere un ruolo IAM

Uno sviluppatore con unAWS account amministrativo può consentire a un utente di assumere un ruolo IAM. A tale scopo, si crea una nuova policy e la si associa a quell'utente.

La policy deve contenere una dichiarazione con l'Alloweffetto sull'`sts:AssumeRole`azione, come mostrato nell'esempio che segue. Resource Agli utenti che ottengono la policy, mediante l'appartenenza a un gruppo o collegata direttamente.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::<aws_account_id>:role/workdocs_app_role"
  }
}
```

Limitazione dell'accesso a una WorkDocs istanza Amazon specifica

Se hai più WorkDocs siti Amazon su unAWS account e desideri concedere l'accesso tramite API a un sito specifico, puoi definire unCondition elemento. L'elemento Condition consente di specificare le condizioni che indicano quando è applicata una policy.

L'esempio seguente esempio mostra un elemento condizionale:

```
"Condition":
{
    "StringEquals": {
        "Resource.OrganizationId": "d-123456789c5"
    }
}
```

Con la condizione di cui sopra in una politica, gli utenti possono accedere all' WorkDocs istanza Amazon solo con l'ID did-123456789c5. L'ID dell' WorkDocs istanza Amazon viene talvolta denominato ID dell'organizzazione o ID directory. Per ulteriori informazioni, consulta [Limitazione dell'accesso a una WorkDocs istanza Amazon specifica](#).

Segui questi passaggi per ottenere un ID WorkDocs dell'organizzazione Amazon dallaAWS console:

Per ottenere un ID organizzazione

1. Nel riquadro di navigazione [AWS Directory Service console](#), scegliere Directories (Directory).
2. Prendi nota del valore dell'ID della directory che corrisponde al tuo WorkDocs sito Amazon. Questo è l'ID dell'organizzazione per il sito.

Autenticazione e controllo degli accessi per le applicazioni utente

Amazon WorkDocs Le applicazioni a livello di utente vengono registrate e gestite tramite Amazon WorkDocs Console. Gli sviluppatori devono registrare le proprie applicazioni sul My Applications pagina su Amazon WorkDocs Console, che fornisce ID univoci per ogni applicazione. Durante la registrazione, gli sviluppatori devono specificare gli URI di reindirizzamento in cui riceveranno i token di accesso, nonché gli ambiti delle applicazioni.

Attualmente, le applicazioni possono accedere solo ad Amazon WorkDocs siti all'interno dello stesso AWS account in cui sono registrati.

Indice

- [Concessione delle autorizzazioni per chiamare Amazon WorkDocs API](#)
- [Utilizzo degli ID delle cartelle nelle chiamate API](#)
- [Creazione di un'applicazione](#)
- [Ambiti delle applicazioni](#)
- [Autorizzazione](#)
- [Richiamo Amazon WorkDocs API](#)

Concessione delle autorizzazioni per chiamare Amazon WorkDocs API

Gli utenti dell'interfaccia a riga di comando devono disporre delle autorizzazioni complete per Amazon WorkDocs e AWS Directory Service. Senza le autorizzazioni, vengono restituite tutte le chiamate API `UnauthorizedResourceAccessException` messaggi. La policy seguente concede autorizzazioni complete.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "workdocs:*",
```

```

        "ds:*",
        "ec2:CreateVpc",
        "ec2:CreateSubnet",
        "ec2:CreateNetworkInterface",
        "ec2:CreateTags",
        "ec2:CreateSecurityGroup",
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeAvailabilityZones",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2>DeleteSecurityGroup",
        "ec2>DeleteNetworkInterface",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress"
    ],
    "Effect": "Allow",
    "Resource": "*"
}
]
}

```

Se vuoi concedere autorizzazioni di sola lettura, usa questa policy.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "workdocs:Describe*",
        "ds:DescribeDirectories",
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}

```

Nella politica, la prima azione garantisce l'accesso a tutta l'Amazzonia WorkDocs Describeoperazioni. LaDescribeDirectories action ottiene informazioni sul tuoAWS Directory

ServiceDirectory. Le operazioni di Amazon EC2 abilitano Amazon WorkDocs per ottenere un elenco di VPC e sottoreti.

Utilizzo degli ID delle cartelle nelle chiamate API

Ogni volta che una chiamata API accede a una cartella, è necessario utilizzare l'ID della cartella, non il nome della cartella. Ad esempio, se si passa `client.get_folder(FolderId='MyDocs')`, la chiamata API restituisce un `UnauthorizedResourceAccessException` messaggio e il seguente messaggio 404.

```
client.get_folder(FolderId='MyDocs')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "C:\Users\user-name\AppData\Local\Programs\Python\Python36-32\lib\site-packages\botocore\client.py", line 253, in _api_call
    return self._make_api_call(operation_name, kwargs)
  File "C:\Users\user-name\AppData\Local\Programs\Python\Python36-32\lib\site-packages\botocore\client.py", line 557, in _make_api_call
    raise error_class(parsed_response, operation_name)
botocore.errorfactory.UnauthorizedResourceAccessError: An error occurred (UnauthorizedResourceAccessError) when calling the GetFolder operation: Principal [arn:aws:iam:395162986870:user/Aman] is not allowed to execute [workdocs:GetFolder] on the resource.
```

Per evitarlo, usa l'ID nell'URL della cartella.

`site.workdocs/index.html#/folder/abc123def456ghi789jkl1789mno4be7024df198736472dd50ca970eb22796082e3d489577`.

Il passaggio di tale ID restituisce un risultato corretto.

```
client.get_folder(FolderId='abc123def456ghi789jkl1789mno4be7024df198736472dd50ca970eb22796082e3d489577')
{'ResponseMetadata': {'RequestId': 'f8341d4e-4047-11e7-9e70-afa8d465756c',
  'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': 'f234564e-1234-56e7-89e7-a10fa45t789c', 'cache-control': 'private, no-cache, no-store, max-age=0',
  'content-type': 'application/json', 'content-length': '733', 'date': 'Wed, 24 May 2017 06:12:30 GMT'}, 'RetryAttempts': 0}, 'Metadata': {'Id': 'abc123def456ghi789jkl1789mno4be7024df198736472dd50ca970eb22796082e3d489577', 'Name': 'sentences', 'CreatorId': 'S-1-5-21-2125721135-1643952666-3011040551-2105&d-906724f1ce', 'ParentFolderId': '0a811a922403ae8e1d3c180f4975f38f94372c3d6a2656c50851c7fb76677363',
```

```
'CreatedTimestamp': datetime.datetime(2017, 5, 23, 12, 59, 13, 8000, tzinfo=tzlocal()), 'ModifiedTimestamp': datetime.datetime(2017, 5, 23, 13, 13, 9, 565000, tzinfo=tzlocal()), 'ResourceState': 'ACTIVE', 'Signature': 'b7f54963d60ae1d6b9ded476f5d20511'}}}
```

Creazione di un'applicazione

Come Amazon WorkDocs , puoi creare la tua applicazione completando le fasi seguenti.

Per creare un'applicazione

1. Aprire Amazon WorkDocs Console presso <https://console.aws.amazon.com/zocalo/>.
2. Scegliere My Applications (Le mie applicazioni), Create an Application (Crea un'applicazione).
3. Immetti uno dei seguenti valori:

Nome applicazione

Nome dell'applicazione.

E-mail

Indirizzo e-mail da associare all'applicazione.

Application Description (Descrizione applicazione)

Descrizione per l'applicazione.

Redirect URIs (URI di reindirizzamento)

Posizione in cui vuoi Amazon WorkDocs per reindirizzare il traffico a.

Ambiti delle applicazioni

Ambito, ovvero lettura o scrittura, che vuoi assegnare all'applicazione. Per ulteriori dettagli, consulta [Ambiti delle applicazioni](#).

4. Scegli Create (Crea).

Ambiti delle applicazioni

Amazon WorkDocs supporta gli ambiti delle applicazioni seguenti:

- `Contenuto letto (workdocs.content.read)`, che permette all'applicazione di accedere al seguente WorkDocs API:
 - `Get*`
 - `Describe*`
- `Scrittura contenuto (workdocs.content.write)`, che permette all'applicazione di accedere al seguente WorkDocs API:
 - `Creare*`
 - `Aggiorna*`
 - `Elimina*`
 - `Initiate*`
 - `Abort*`
 - `Add*`
 - `Remove*`

Autorizzazione

Dopo la registrazione, un'applicazione può richiedere l'autorizzazione per conto di Amazon WorkDocs utente. A questo scopo, l'applicazione deve visitare Amazon WorkDocs endpoint OAuth, `https://auth.amazonworkdocs.com/oauth` e fornire i parametri di query seguenti:

- [Campo obbligatorio]`app_id`ID applicazione generato durante la registrazione di un'applicazione.
- [Campo obbligatorio]`auth_type`Tipo di OAuth per la richiesta. Il valore supportato è `ImplicitGrant`.
- [Campo obbligatorio]`redirect_uri`URI di reindirizzamento registrato per un'applicazione per la ricezione di un token di accesso.
- [Opzionale]`scopes`elenco di ambiti separati da virgole. Se non è specificato, viene usato l'elenco di ambiti selezionati durante la registrazione.
- [Opzionale]`state`stringa restituita insieme a un token di accesso.

Note

Se si richiedono moduli crittografici convalidati FIPS 140-2 quando si accede ad AWS tramite un'interfaccia a riga di comando o un'API, utilizzare un endpoint FIPS. Per ulteriori

informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-2](#).

Richiesta GET di esempio per avviare il flusso OAuth per ottenere un token di accesso:

```
GET https://auth.amazonworkdocs.com/oauth?app_id=my-app-id&auth_type=ImplicitGrant&redirect_uri=https://myapp.com/callback&scopes=workdocs.content.read&state=xyz
```

Durante il flusso di autorizzazione OAuth avvengono le attività seguenti:

1. All'utente dell'applicazione viene richiesto di entrare in Amazon WorkDocs nome del sito.
2. L'utente viene reindirizzato al Amazon WorkDocs pagina di autenticazione per inserire le proprie credenziali.
3. Dopo che l'autenticazione riesce, l'utente visualizza la schermata per il consenso, in cui può concedere o negare all'applicazione l'autorizzazione di accesso ad Amazon WorkDocs.
4. Dopo che l'utente sceglie Accept nella schermata per il consenso, il browser dell'utente viene reindirizzato all'URL di richiamata dell'applicazione, insieme al token di accesso e alle informazioni sulla regione come parametri di query.

Richiesta GET di esempio da Amazon WorkDocs:

```
GET https://myapp.com/callback?accessToken=accesstoken&region=us-east-1&state=xyz
```

Oltre al token di accesso, Amazon WorkDocs Restituisce anche il servizio OAuthregioncome parametro di query per l'Amazon selezionato WorkDocs sito. Le applicazioni esterne devono utilizzareregionparametro per determinare l'Amazzonia WorkDocs endpoint del servizio.

Se si richiedono moduli crittografici convalidati FIPS 140-2 quando si accede ad AWS tramite un'interfaccia a riga di comando o un'API, utilizzare un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-2](#).

Richiamo Amazon WorkDocs API

Dopo aver ottenuto il token di accesso, la tua applicazione può effettuare chiamate API ad Amazon WorkDocsServizi .

⚠ Important

Questo esempio mostra come utilizzare una richiesta GET arrotolata per ottenere i metadati di un documento.

```
Curl "https://workdocs.us-east-1.amazonaws.com/api/v1/documents/{document-id}" -H
"Accept: application/json" -H "Authentication: Bearer accesstoken"
```

Esempio di JavaScript funzione per descrivere le cartelle root di un utente:

```
function printRootFolders(accessToken, siteRegion) {
    var workdocs = new AWS.WorkDocs({region: siteRegion});
    workdocs.makeUnauthenticatedRequest("describeRootFolders", {AuthenticationToken:
    accessToken}, function (err, folders) {
        if (err) console.log(err);
        else console.log(folders);
    });
}
```

Di seguito viene descritta una chiamata API di esempio basata su Java:

```
AWSCredentialsProvider credentialsProvider = new AWSCredentialsProvider() {
    @Override
    public void refresh() {}

    @Override
    public AWSCredentials getCredentials() {
        new AnonymousAWSCredentials();
    }
};

// Set the correct region obtained during OAuth flow.
workDocs =
    AmazonWorkDocsClient.builder().withCredentials(credentialsProvider)
        .withRegion(Regions.US_EAST_1).build();

DescribeRootFoldersRequest request = new DescribeRootFoldersRequest();
request.setAuthenticationToken("access-token-obtained-through-workdocs-oauth");
DescribeRootFoldersResult result = workDocs.describeRootFolders(request);
```

```
for (FolderMetadata folder : result.getFolders()) {  
    System.out.printf("Folder name=%s, Id=%s \n", folder.getName(), folder.getId());  
}
```

Amazon WorkDocs Gestore di contenuto

Amazon WorkDocs Content Manager è uno strumento di utilità di alto livello che carica contenuti o li scarica da Amazon WorkDocs sito.

Argomenti

- [Costruzione di Amazon WorkDocs Gestore di contenuto](#)
- [Download di un documento](#)
- [Caricamento di un documento](#)

Costruzione di Amazon WorkDocs Gestore di contenuto

Puoi usare Amazon WorkDocs Content Manager per applicazioni amministrative e utente.

Per le applicazioni utente, uno sviluppatore deve creare Amazon WorkDocs Content Manager con anonimatoAWScredenziali e token di autenticazione.

Per le applicazioni amministrative, Amazon WorkDocs il client deve essere inizializzato conAWS Identity and Access Managementcredenziali (IAM). Inoltre, il token di autenticazione più essere omesso nelle successive chiamate API.

Il codice seguente mostra come inizializzare Amazon WorkDocs Content Manager per applicazioni utente che utilizzano Java o C#.

Java:

```
AWSStaticCredentialsProvider credentialsProvider = new AWSStaticCredentialsProvider(new
    AnonymousAWSCredentials());

AmazonWorkDocs client =
    AmazonWorkDocsClient.builder().withCredentials(credentialsProvider).withRegion("region").build

ContentManager contentManager =
    ContentManagerBuilder.standard().withWorkDocsClient(client).withAuthenticationToken("token").b
```

C#:

```
AmazonWorkDocsClient client = new AmazonWorkDocsClient(new AnonymousAWSCredentials(),
    "region");
```

```
ContentManagerParams params = new ContentManagerParams
{
    WorkDocsClient = client,
    AuthenticationToken = "token"
};
IContentManager workDocsContentManager = new ContentManager(param);
```

Download di un documento

Gli sviluppatori possono usare Amazon WorkDocs Content Manager per scaricare una versione specifica o l'ultima versione di un documento da Amazon WorkDocs. I seguenti esempi illustrano come effettuare il download di una versione specifica di un documento con Java e C#.

Note

Per scaricare l'ultima versione di un documento, non specificare il `VersionId` durante la costruzione di una richiesta `GetDocumentStream`.

Java

```
ContentManager contentManager =
    ContentManagerBuilder.standard().withWorkDocsClient(client).withAuthenticationToken("auth-
token").build();

// Download document.
GetDocumentStreamRequest request = new GetDocumentStreamRequest();
request.setDocumentId("document-id");
request.setVersionId("version-id");

// stream contains the content of the document version.
InputStream stream = contentManager.getDocumentStream(request).getStream();
```

C#

```
ContentManager contentManager =
    ContentManagerBuilder.standard().withWorkDocsClient(client).withAuthenticationToken("auth-
token").build();

// Download document.
```



```
GetDocumentStreamRequest request = new GetDocumentStreamRequest();
request.setDocumentId("document-id");
request.setVersionId("version-id");

// stream contains the content of the document version.
InputStream stream = contentManager.getDocumentStream(request).getStream();
```

Caricamento di un documento

Amazon WorkDocs Content Manager fornisce un'API per il caricamento di contenuti su Amazon WorkDocs sito. I seguenti esempi illustrano come caricare un documento utilizzando Java e C #.

Java

```
File file = new File("file-path");
InputStream stream = new FileInputStream(file);
UploadDocumentStreamRequest request = new UploadDocumentStreamRequest();
request.setParentFolderId("destination-folder-id");
request.setContentType("content-type");
request.setStream(stream);
request.setDocumentName("document-name");
contentManager.uploadDocumentStream(request);
```

C#

```
var stream = new FileStream("file-path", FileMode.Open);

UploadDocumentStreamRequest uploadDocumentStreamRequest = new
    UploadDocumentStreamRequest()
{
    ParentFolderId = "destination-id",
    DocumentName = "document-name",
    ContentType = "content-type",
    Stream = stream
};

workDocsContentManager.UploadDocumentStreamAsync(uploadDocumentStreamRequest).Wait();
```

Glossario AWS

Per la terminologia AWS più recente, consultare il [glossario AWS](#) nella documentazione di riferimento per Glossario AWS.

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.