aws

Guida per gli sviluppatori

Amazon WorkDocs

Amazon WorkDocs: Guida per gli sviluppatori

Table of Contents

Avviso: le registrazioni di nuovi clienti e gli upgrade degli account non sono più disponibili per Amazon. WorkDocs Scopri le fasi di migrazione qui: Come migrare i dati da. WorkDocs

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.

Che cos'è Amazon WorkDocs?

Amazon WorkDocs è un sistema di archiviazione, collaborazione e condivisione di documenti. WorkDocs è completamente gestito, sicuro e su scala aziendale. Fornisce solidi controlli amministrativi, oltre a 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 possono essere utilizzate per visualizzare diversi tipi di file, a seconda del tipo di supporto Internet del file. WorkDocs supporta tutti i formati di documenti e immagini più comuni e il supporto per tipi di file multimediali aggiuntivi viene costantemente aggiunto.

Per ulteriori informazioni, consulta Amazon WorkDocs.

Accedere WorkDocs

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

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

Prezzi

In questo modo WorkDocs, non sono previsti costi o impegni anticipati. Paghi solo per gli account utente attivi e lo spazio di archiviazione che utilizzi. Per ulteriori informazioni, consulta la pagina Prezzi.

Risorse

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

- Corsi e workshop: collegamenti a corsi specializzati e basati su ruoli, oltre a laboratori di autoapprendimento per aiutarti ad affinare le tue abilità e acquisire esperienza pratica. AWS
- AWS Developer Center: esplora i tutorial, scarica strumenti e scopri gli eventi per sviluppatori. AWS
- AWS Strumenti per sviluppatori: collegamenti a strumenti di sviluppo SDKs, toolkit IDE e strumenti da riga di comando per lo sviluppo e la gestione di applicazioni. AWS
- Centro risorse introduttivo: scopri come configurare Account AWS, unirti alla AWS community e lanciare la tua prima applicazione.
- Tutorial pratici: segui i tutorial per avviare la step-by-step tua prima applicazione su. AWS
- AWS Whitepaper: collegamenti a un elenco completo di AWS white paper tecnici, su argomenti quali architettura, sicurezza ed economia e redatti da Solutions Architects o altri esperti tecnici. AWS
- Supporto AWS Center: l'hub per la creazione e la gestione dei casi. Supporto AWS Include anche collegamenti ad altre risorse utili, come forum, informazioni tecniche FAQs, stato di salute del servizio e AWS Trusted Advisor.
- Supporto— La pagina web principale per informazioni su Supporto one-on-one, un canale di supporto a risposta rapida per aiutarti 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 del sito: informazioni dettagliate sul nostro copyright e marchio, sull'account, sulla licenza e sull'accesso al sito e altri argomenti.

Nozioni di base

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

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

Esempi

- Connect to WorkDocs with IAM: credenziali utente e query per gli utenti
- Connettersi a 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

Connect to WorkDocs with IAM: credenziali utente e 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 dell'API e il WorkDocs sito 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 tramite una policy IAM appropriata.

L'esempio di codice utilizza l'DescribeUsersAPI 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 ID WorkDocs dell'organizzazione.

Segui questi passaggi per ottenere un ID WorkDocs dell'organizzazione 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. 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 a WorkDocs assumendo un ruolo

Questo esempio utilizza AWS Java SDK per assumere un ruolo e utilizzare le credenziali di sicurezza temporanee del ruolo per accedere. WorkDocs L'esempio di codice utilizza l'DescribeFolderContentsAPI 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 per WorkDocs caricare file, consulta Caricamento di file nella Guida per l'WorkDocs utente.

Per caricare un documento in WorkDocs, usa la procedura seguente.

Per caricare un documento

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

   Se utilizzi le credenziali utente IAM, consulta. Connect to WorkDocs with IAM: credenziali utente e query per gli utenti Se assumi un ruolo IAM, consulta Connettersi a 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);
HttpURLConnection connection = (HttpURLConnection) 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 l 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**
>
> È necessario essere uno sviluppatore di software per completare i passaggi descritti in questa sezione. Per informazioni sull'utilizzo per WorkDocs scaricare file, consulta Download dei file nella Guida per l'WorkDocs utente.

Per scaricare un documento WorkDocs, recuperate un URL per il download nel modo seguente, quindi utilizzate le azioni API fornite dalla 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 di accedere alla gestione degli abbonamenti alle notifiche. APIs
2. Chiama l'abbonamento APIs alle notifiche per abilitare o disabilitare la pubblicazione dei 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 CreateNotificationSubscriptiondopo esserti abbonato alle notifiche.

1. Apri la WorkDocs console all'indirizzo 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 WorkDocs sito.

Per informazioni sull'abilitazione WorkDocs all'uso delle 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 endpoints 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 all'indirizzo 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. 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, è necessario effettuare una chiamata all'ActivateUserAPI per attivare l'utente in 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(10485761));
    request.setStorageRule(storageRule);
```

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

Segui questi passaggi per ottenere un ID WorkDocs dell'organizzazione 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. Questo è l'ID dell'organizzazione del sito.

Concessione delle autorizzazioni agli utenti per una risorsa

L'esempio seguente mostra come utilizzare l'AddResourcePermissionsAPI 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

WorkDocs APIs gli amministrativi sono autenticati e autorizzati tramite le policy IAM. Gli amministratori IAM possono creare una policy IAM e collegarla a un ruolo o utente IAM che può essere utilizzato dallo sviluppatore per accedere all'API.

Di seguito vengono riportati degli esempi:

Attività

- Concedere agli sviluppatori le autorizzazioni per l'API WorkDocs
- Concessione dell'autorizzazione agli sviluppatori di terze parti a WorkDocs APIs
- Concedere agli utenti l'autorizzazione ad assumere un ruolo IAM
- Limitazione dell'accesso a un'istanza specifica WorkDocs

Concedere agli sviluppatori le autorizzazioni per l'API WorkDocs

> **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 a un utente IAM dallo stesso AWS account. Per fare ciò, crea una policy di autorizzazione WorkDocs API e collegala all'utente IAM. La seguente politica API concede l'autorizzazione di sola lettura ai vari. `Describe` APIs

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

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

Concessione dell'autorizzazione agli sviluppatori di terze parti a WorkDocs APIs

Puoi concedere l'accesso a sviluppatori di terze parti o agli utenti che utilizzano un AWS account diverso. A tale scopo, crea un ruolo IAM e allega le politiche di autorizzazione delle WorkDocs API.

Questo tipo di accesso è richiesto negli scenari seguenti:

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

In entrambi questi scenari, sono coinvolti due AWS account, un account per sviluppatori e un altro account che ospita un WorkDocs sito. AWS

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

- L'ID AWS del tuo account
- `External ID` univoco che verrà utilizzato dal cliente per identificarti. Per ulteriori informazioni, vedi Come utilizzare un ID esterno per concedere l'accesso alle tue AWS risorse a terzi.
- È necessario accedere a un elenco delle WorkDocs APIs applicazioni a cui è necessario accedere. Il controllo delle policy basato su IAM fornisce un controllo granulare, ossia la capacità di definire policy di autorizzazione o rifiuto a livello di singola API. Per l'elenco di WorkDocs APIs, consulta WorkDocs API Reference.

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

Per configurare IAM per l'accesso tra più account

1. Crea una politica di autorizzazione WorkDocs API, chiamala `WorkDocsAPIReadOnly` policy.
2. Crea un nuovo ruolo nella console IAM dell' AWS account che ospita il WorkDocs sito:
   a. Accedi Console di gestione AWS 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 del tuo AWS account. 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 e desideri concedere le autorizzazioni agli utenti di un account sviluppatore di applicazioni. 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, specifica l'ID dell' AWS account a cui desideri concedere l'accesso alle tue risorse e inserisci anche l'ID esterno in caso di accesso da parte di terzi.
   f. Fare clic su Next Step (Fase successiva) per collegare una policy.
3. Nella pagina Allega policy, cerca la policy di autorizzazione dell' WorkDocs API creata in precedenza e seleziona la casella accanto alla policy e fai clic su Passaggio successivo.
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 l'autorizzazione ad assumere un ruolo IAM

Uno sviluppatore con un AWS account amministrativo può consentire a un utente di assumere un ruolo IAM. Per farlo, crei una nuova policy e la alleghi a quell'utente.

La policy deve includere una dichiarazione con l'`Allow`effetto sull'`sts:AssumeRole`azione, oltre all'Amazon Resource Name (ARN) del ruolo in un `Resource` elemento, come illustrato nell'esempio seguente. Gli utenti che ottengono la policy, tramite l'appartenenza al gruppo o tramite collegamento diretto, possono passare al ruolo specificato.

```
{
  "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 un'istanza specifica WorkDocs

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

L'esempio seguente mostra un elemento condizionale:

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

Con la condizione di cui sopra in una policy, gli utenti possono accedere all' WorkDocs istanza solo con l'ID did-123456789c5. WorkDocs L'ID dell'istanza viene talvolta indicato come ID dell'organizzazione o ID della directory. Per ulteriori informazioni, consulta Limitazione dell'accesso a un'istanza specifica WorkDocs .

Segui questi passaggi per ottenere un ID WorkDocs dell'organizzazione 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. Questo è l'ID dell'organizzazione del sito.

Autenticazione e controllo degli accessi per le applicazioni utente

WorkDocs le applicazioni a livello utente vengono registrate e gestite tramite la WorkDocs console. Gli sviluppatori devono registrare le proprie applicazioni sulla `My Applications` pagina della WorkDocs console che fornirà informazioni uniche IDs per ogni applicazione. Durante la registrazione, gli sviluppatori devono specificare il reindirizzamento URIs in cui riceveranno i token di accesso e gli ambiti delle applicazioni.

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

Indice

Concessione delle autorizzazioni per chiamare il WorkDocs APIs

Gli utenti dell'interfaccia a riga di comando devono disporre delle autorizzazioni complete per e. WorkDocs Directory Service Senza le autorizzazioni, tutte le chiamate API restituiscono UnauthorizedResourceAccessExceptionmessaggi. La seguente politica concede le 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 desideri concedere autorizzazioni di sola lettura, utilizza questo criterio.

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

Nella politica, la prima azione concede l'accesso a tutte le operazioni. WorkDocs `Describe` L'`DescribeDirectories` azione consente di ottenere informazioni sulle tue Directory Service directory. Le EC2 operazioni di Amazon consentono WorkDocs di ottenere un elenco delle tue VPCs sottoreti.

Utilizzo della cartella IDs 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 l'esito è `client.get_folder(FolderId='MyDocs')` positivo, la chiamata API restituisce un UnauthorizedResourceAccessExceptionmessaggio 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.UnauthorizedResourceAccessException: An error occurred
 (UnauthorizedResourceAccessException) when calling the GetFolder operation:
Principal [arn:aws:iam::395162986870:user/Aman] is not allowed to execute
 [workdocs:GetFolder] on the resource.
```

Per evitare che ciò accada, utilizza l'ID nell'URL della cartella.

site.workdocs/index.html#/folder/abc123def456ghi789jkl789mno4be7024df198736472dd50ca970eb22796082e3d489577.

Il passaggio di tale ID restituisce un risultato corretto.

```
client.get_folder(FolderId='abc123def456ghi789jkl789mno4be7024df198736472dd50ca970eb22796082e3d
{'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':
 'abc123def456ghi789jkl789mno4be7024df198736472dd50ca970eb22796082e3d489577', '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

In qualità di WorkDocs amministratore, crea la tua applicazione utilizzando i seguenti passaggi.

Per creare un'applicazione

1. Apri la WorkDocs console all'indirizzo 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.

   Reindirizzamento URIs

   La posizione verso cui desideri WorkDocs reindirizzare il traffico.

   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

WorkDocs supporta i seguenti ambiti applicativi:

- Content Read (`workdocs.content.read`), che consente all'applicazione di accedere a quanto segue: WorkDocs APIs
  - Get*

  - Describe*
- Content Write (`workdocs.content.write`), che consente all'applicazione di accedere a quanto segue WorkDocs APIs:
  - Creare*
  - Aggiorna*
  - Elimina*
  - Initiate*
  - Abort*
  - Add*
  - Remove*

Autorizzazione

Una volta completata la registrazione dell'applicazione, un'applicazione può richiedere l'autorizzazione per conto di qualsiasi WorkDocs utente. A tale scopo, l'applicazione deve visitare l' WorkDocs OAuthendpoint e fornire i seguenti parametri di interrogazione: `https://auth.amazonworkdocs.com/oauth`

- [Obbligatorio]`app_id`: ID dell'applicazione generato quando un'applicazione viene registrata.
- [Obbligatorio]`auth_type`: il OAuth tipo di richiesta. Il valore supportato è `ImplicitGrant`.
- [Obbligatorio]`redirect_uri`: l'URI di reindirizzamento registrato per consentire a un'applicazione di ricevere un token di accesso.
- [Facoltativo]`scopes`: un elenco di ambiti delimitato da virgole. Se non è specificato, viene usato l'elenco di ambiti selezionati durante la registrazione.
- [Facoltativo]`state`: una stringa che viene 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.

Un esempio di richiesta GET per avviare il OAuth flusso 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 OAuth autorizzazione si verifica quanto segue:

1. All'utente dell'applicazione viene richiesto di inserire il nome del WorkDocs sito.
2. L'utente viene reindirizzato alla pagina di WorkDocs 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 a 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.

Un esempio di richiesta GET da: WorkDocs

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

Oltre al token di accesso, il WorkDocs OAuth servizio viene restituito anche `region` come parametro di interrogazione per il WorkDocs sito selezionato. Le applicazioni esterne devono utilizzare il `region` parametro per determinare l'endpoint del WorkDocs 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.

Invocando WorkDocs APIs

Dopo aver ottenuto il token di accesso, l'applicazione può effettuare chiamate API a servizi WorkDocs.

> **Important**

Questo esempio mostra come utilizzare una richiesta GET curl 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"
```

Una JavaScript funzione di esempio per descrivere le cartelle principali 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());
}
```

WorkDocs Gestore dei contenuti

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

Argomenti

Costruire Content Manager WorkDocs

È possibile utilizzare WorkDocs Content Manager per applicazioni amministrative e utente.

Per le applicazioni utente, uno sviluppatore deve creare WorkDocs Content Manager con AWS credenziali anonime e un token di autenticazione.

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

Il codice seguente mostra come inizializzare WorkDocs Content Manager per le applicazioni utente utilizzando 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 utilizzare WorkDocs Content Manager per scaricare una versione specifica o l'ultima versione di un documento da 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

WorkDocs Content Manager fornisce un'API per il caricamento di contenuti su un sito. WorkDocs 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();
```