Guía para desarrolladores

Amazon WorkDocs

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

Amazon WorkDocs: Guía para desarrolladores

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

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

..... iv
¿Qué es Amazon WorkDocs? 1
Accediendo WorkDocs 1
Precios 1
Recursos 2
Introducción 3
Conectarse WorkDocs con credenciales de usuario de IAM 3
Conectarse a WorkDocs mediante la asunción de un rol 5
Cargar un documento 8
Descargar un documento 10
Configuración de notificaciones 10
Creación de un usuario 13
Conceder a los usuarios permisos con respecto a un recurso 14
Autenticación y control de acceso para aplicaciones administrativas 16
Otorgar permisos a los desarrolladores para acceder a la API WorkDocs 16
Otorgar permiso a desarrolladores externos para WorkDocs APIs 17
Concesión de permisos de usuario para asumir un rol de IAM 19
Restringir el acceso a una instancia específica WorkDocs 19
Autenticación y control de acceso para aplicaciones de usuario 21
Otorgar permisos para llamar al WorkDocs APIs 21
Uso de la carpeta IDs en las llamadas a la API 23
Creación de una aplicación de 24
Ámbitos de la aplicación 24
Autorización 25
Invocando WorkDocs APIs 26
WorkDocs Gestor de contenido 28
Construyendo un administrador WorkDocs de contenido 28
Descarga de un documento 29
Carga de un documento 30

Aviso: las suscripciones de nuevos clientes y las actualizaciones de cuentas ya no están disponibles para Amazon. WorkDocs Obtén más información sobre los pasos de migración aquí: Cómo migrar datos desde. WorkDocs

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.

¿Qué es Amazon WorkDocs?

Amazon WorkDocs es un sistema de almacenamiento, colaboración e intercambio de documentos. WorkDocs es totalmente gestionado, seguro y de escala empresarial. Proporciona potentes controles administrativos, además de funcionalidades de comentarios que ayudan a mejorar la productividad de los usuarios. Los archivos se almacenan en la nube de forma segura. Los archivos de sus usuarios solo están visibles para ellos y los colaboradores y espectadores designados. Otros miembros de la organización no tienen acceso a ningún otro archivo de usuario, salvo que se les haya concedido acceso específicamente.

Los usuarios pueden compartir sus archivos con otros miembros de su organización para colaboraciones o revisiones. Las aplicaciones WorkDocs cliente se pueden utilizar para ver muchos tipos diferentes de archivos, según el tipo de soporte de Internet del archivo. WorkDocs admite todos los formatos de documentos e imágenes habituales, y se añade constantemente compatibilidad con otros tipos de medios.

Para obtener más información, consulta Amazon WorkDocs.

Accediendo WorkDocs

Los usuarios finales utilizan las aplicaciones cliente para obtener acceso a sus archivos. Los usuarios no administrativos nunca necesitan usar la WorkDocs consola o el panel de administración. WorkDocs ofrece varias aplicaciones y utilidades de cliente diferentes:

- Una aplicación web que se utiliza para la administración y revisión de documentos.
- Aplicaciones nativas para dispositivos móviles que se utilizan para la revisión de documentos.
- WorkDocs Unidad utilizada para sincronizar una carpeta del escritorio de tu Mac o Windows con tus WorkDocs archivos.

Precios

Con WorkDocs, no hay comisiones ni compromisos por adelantado. Solo se paga por las cuentas de usuario activas y por el almacenamiento que se utilice. Para obtener más información, consulte Precios.

Recursos

Los recursos relacionados siguientes pueden serle de ayuda cuando trabaje con este servicio.

- Clases y talleres: enlaces a cursos especializados y basados en roles, además de laboratorios a tu propio ritmo que te ayudarán a perfeccionar tus AWS habilidades y a adquirir experiencia práctica.
- AWS Centro para desarrolladores: explore los tutoriales, descargue herramientas y obtenga información sobre los eventos para desarrolladores. AWS
- AWS Herramientas para desarrolladores: enlaces a herramientas para desarrolladores SDKs, kits de herramientas IDE y herramientas de línea de comandos para desarrollar y administrar AWS aplicaciones.
- Centro de recursos de introducción: aprenda a configurar su aplicación Cuenta de AWS, a unirse a la AWS comunidad y a lanzar su primera aplicación.
- Tutoriales prácticos: sigue step-by-step los tutoriales para lanzar tu primera aplicación. AWS
- AWS Documentos técnicos: enlaces a una lista completa de AWS documentos técnicos, que abarcan temas como la arquitectura, la seguridad y la economía, redactados por arquitectos de AWS soluciones u otros expertos técnicos.
- AWS Support Center: el centro para crear y gestionar sus casos. AWS Support También incluye enlaces a otros recursos útiles, como foros, información técnica FAQs, estado del servicio y AWS Trusted Advisor.
- Soporte— La página web principal con información sobre Soporte un one-on-one canal de soporte de respuesta rápida que le ayudará a crear y ejecutar aplicaciones en la nube.
- Contacta con nosotros – Un punto central de contacto para las consultas relacionadas con la facturación AWS , cuentas, eventos, abuso y demás problemas.
- AWS Condiciones del sitio: información detallada sobre nuestros derechos de autor y marca comercial; su cuenta, licencia y acceso al sitio; y otros temas.

Introducción

Los siguientes fragmentos de código pueden ayudarte a empezar a usar el WorkDocs SDK.

> **Note**
>
> Para mayor seguridad, cree usuarios federados en lugar de usuarios de IAM siempre que sea posible.

Ejemplos

- Conéctese WorkDocs con las credenciales de usuario de IAM y consulte a los usuarios
- Conectarse a WorkDocs mediante la asunción de un rol
- Cargar un documento
- Descargar un documento
- Configuración de notificaciones
- Creación de un usuario
- Conceder a los usuarios permisos con respecto a un recurso

Conéctese WorkDocs con las credenciales de usuario de IAM y consulte a los usuarios

El siguiente código muestra cómo utilizar las credenciales de API de un usuario de IAM para realizar llamadas a la API. En este caso, el usuario de la API y el WorkDocs sitio pertenecen a la misma AWS cuenta.

> **Note**
>
> Para mayor seguridad, cree usuarios federados en lugar de usuarios de IAM siempre que sea posible.

Asegúrese de que se haya concedido al usuario de IAM el acceso a la WorkDocs API mediante una política de IAM adecuada.

El ejemplo de código utiliza la DescribeUsersAPI para buscar usuarios y obtener metadatos para los usuarios. Los metadatos de los usuarios proporcionan detalles como el nombre, los apellidos, el ID de usuario y el ID de carpeta raíz. El ID de carpeta raíz es especialmente útil si desea realizar operaciones de carga o descarga de contenido en nombre del usuario.

El código requiere que obtengas un identificador de WorkDocs organización.

Sigue estos pasos para obtener un identificador de WorkDocs organización desde la AWS consola:

Para obtener un ID de organización

1. En el panel de navegación de la consola de AWS Directory Service, elija Directorios.
2. Anote el valor del ID de directorio que corresponde a su WorkDocs sitio. Es el ID de organización del sitio.

El siguiente ejemplo muestra cómo utilizar las credenciales de IAM para realizar llamadas a la 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());
    }
  }
}
```

Conectarse a WorkDocs mediante la asunción de un rol

En este ejemplo, se utiliza el SDK de AWS Java para asumir un rol y se utilizan las credenciales de seguridad temporales del rol para acceder WorkDocs. El ejemplo de código usa la DescribeFolderContentsAPI para enumerar los elementos de la carpeta de un usuario.

```
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());
  }
}
```

Cargar un documento

> **Note**
>
> Debe ser un desarrollador de software para completar los pasos de esta sección. Para obtener información sobre cómo WorkDocs cargar archivos, consulte Carga de archivos en la Guía del WorkDocs usuario.

Utilice el siguiente procedimiento para cargar un documento en WorkDocs.

Para cargar un documento

1. Cree una instancia de `AmazonWorkDocsClient` como la siguiente:

   Si usa las credenciales de usuario de IAM, consulte Conéctese WorkDocs con las credenciales de usuario de IAM y consulte a los usuarios. Si asume un rol de IAM, consulte Conectarse a WorkDocs mediante la asunción de un rol para obtener más información.

   > **Note**
   >
   > Para mayor seguridad, cree usuarios federados en lugar de usuarios de IAM siempre que sea posible.

```
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. Obtenga una URL firmada para la carga como la siguiente:

```
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. Cargue el documento con la URL firmada como se muestra a continuación:

```
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. Complete el proceso de carga cambiando el estado del documento `ACTIVE` como se muestra a continuación:

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

Descargar un documento

> **Note**
>
> Debe ser un desarrollador de software para completar los pasos de esta sección. Para obtener información sobre WorkDocs cómo descargar archivos, consulte Descarga de archivos en la Guía del WorkDocs usuario.

Para descargar un documento WorkDocs, obtenga una URL para la descarga de la siguiente manera y, a continuación, utilice las acciones de la API proporcionadas por su plataforma de desarrollo para descargar el archivo mediante la 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());
```

Configuración de notificaciones

Sigue este proceso para configurar las notificaciones:

1. Configura los permisos de usuario o rol de IAM para permitir que las personas que llaman accedan a la gestión de suscripciones a las notificaciones. APIs
2. Llame a la suscripción de notificaciones APIs para activar o desactivar la publicación de mensajes de SNS en su terminal.

> **Note**
>
> Para mayor seguridad, cree usuarios federados en lugar de usuarios de IAM siempre que sea posible.

Para establecer los permisos de usuarios de IAM

- Utilice la consola IAM para configurar los siguientes permisos para el usuario:

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

Para habilitar las notificaciones

La activación de las notificaciones le permite llamar CreateNotificationSubscriptiondespués de suscribirse a las notificaciones.

1. Abre la WorkDocs consola en https://console.aws.amazon.com/zocalo/.
2. En la página Administre sus WorkDocs sitios, seleccione el directorio deseado y, a continuación, elija Acciones y, a continuación, Administrar notificaciones.
3. En la página Administrar notificaciones, elija Habilitar notificaciones.
4. Introduce el ARN del usuario o rol al que quieres permitir que reciba notificaciones de tu WorkDocs sitio.

Para obtener información sobre cómo WorkDocs habilitar el uso de notificaciones, consulte Uso de la WorkDocs API de Amazon con el SDK de AWS para Python y AWS Lambda. Una vez que habilite las notificaciones, usted y su usuario podrán suscribirse a ellas.

Para suscribirse a las notificaciones WorkDocs

1. Prepare el punto de conexión para procesar mensajes de Amazon SNS. Para obtener más información, consulte Fanout to HTTP/S endpoints en la Guía para desarrolladores de Amazon Simple Notification Service.

   > **Important**
   >
   > SNS envía un mensaje de confirmación al punto de conexión configurado. Debe confirmar este mensaje para recibir las notificaciones. Además, si necesita módulos criptográficos validados FIPS 140-2 al acceder a AWS a través de una interfaz de la línea de comandos o una API, utilice un punto de conexión de FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-2.

2. Haga lo siguiente:

   - Obtención de un ID de organización
     1. En el panel de navegación de la consola AWS Directory Service, elija Directorios.
     2. El ID de directorio correspondiente a tu WorkDocs sitio de Amazon también sirve como ID de organización para ese sitio.
   - Cree una solicitud de suscripción como la siguiente:

```
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());
```

Notificaciones de SNS

El mensaje incluye la siguiente información:

- `organizationId`: el ID de la organización.
- `parentEntityType`: el tipo de entidad principal (`Document` | `DocumentVersion` | `Folder`).
- `parentEntityId`: el ID de la entidad principal.
- `entityType`: el tipo de la entidad (`Document` | `DocumentVersion` | `Folder`).
- `entityId`: el ID de la entidad.
- action: la acción, que puede ser uno de los siguientes valores:
  - `delete_document`
  - `move_document`
  - `recycle_document`
  - `rename_document`
  - `revoke_share_document`
  - `share_document`
  - `upload_document_version`

Para deshabilitar las notificaciones

1. Abre la WorkDocs consola en https://console.aws.amazon.com/zocalo/.
2. En la página Administre sus WorkDocs sitios, seleccione el directorio deseado y, a continuación, elija Acciones y, a continuación, Administrar notificaciones.
3. En la página Administrar notificaciones, seleccione el ARN para el desea deshabilitar las notificaciones y elija Deshabilitar notificaciones.

Creación de un usuario

El siguiente ejemplo muestra cómo crear un usuario en. WorkDocs

> **Note**
>
> No es una operación válida para una configuración de Connected AD. Para crear un usuario en la configuración de Connected AD, el usuario ya debe estar presente en el directorio de la empresa. A continuación, debe realizar una llamada a la ActivateUserAPI para activar el usuario en WorkDocs.

En el ejemplo siguiente se muestra cómo crear un usuario con una cuota de almacenamiento de un 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);
```

Sigue estos pasos para obtener un ID de WorkDocs organización desde la AWS consola:

Para obtener un ID de organización

1. En el panel de navegación de la consola de AWS Directory Service, elija Directorios.
2. Anote el valor del ID de directorio que corresponde a su WorkDocs sitio. Es el ID de organización del sitio.

Conceder a los usuarios permisos con respecto a un recurso

El siguiente ejemplo muestra cómo usar la AddResourcePermissionsAPI para conceder `CONTRIBUTOR` permisos `USER` a un recurso. También puede usar la API para conceder permisos a un usuario o grupo con respecto a una carpeta o 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);
```

Autenticación y control de acceso para aplicaciones administrativas

WorkDocs APIs los administradores se autentican y autorizan mediante las políticas de IAM. Los administradores de IAM pueden crear una política de IAM y asociarla a un rol o usuario de IAM que pueda utilizar el desarrollador para acceder a la API.

Se proporcionan los siguientes ejemplos:

Tareas

- Otorgar permisos a los desarrolladores para acceder a la API WorkDocs
- Otorgar permiso a desarrolladores externos para WorkDocs APIs
- Concesión de permisos de usuario para asumir un rol de IAM
- Restringir el acceso a una instancia específica WorkDocs

Otorgar permisos a los desarrolladores para acceder a la API WorkDocs

> **Note**
>
> Para mayor seguridad, cree usuarios federados en lugar de usuarios de IAM siempre que sea posible.

Si es administrador de IAM, puede conceder acceso a la WorkDocs API a un usuario de IAM desde la misma AWS cuenta. Para ello, cree una política de permisos de WorkDocs API y adjúntela al usuario de IAM. La siguiente política de API concede permisos de solo lectura a las distintas. `Describe` APIs

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "WorkDocsAPIReadOnly",
```

```
            "Effect": "Allow",
            "Action": [
                "workdocs:Get*",
      "workdocs:Describe*"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Otorgar permiso a desarrolladores externos para WorkDocs APIs

Puedes conceder acceso a desarrolladores externos o a usuarios que utilicen una AWS cuenta diferente. Para ello, cree un rol de IAM y adjunte las políticas de autorización de la WorkDocs API.

Esta forma de acceso se requiere en los siguientes casos:

- El desarrollador pertenece a la misma organización, pero la AWS cuenta del desarrollador es diferente de la WorkDocs AWS cuenta.
- Cuando una empresa quiere conceder acceso a la WorkDocs API a desarrolladores de aplicaciones de terceros.

En ambos escenarios, hay dos AWS cuentas involucradas, una AWS cuenta de desarrollador y otra cuenta diferente que aloja un WorkDocs sitio.

El desarrollador tendrá que proporcionar la siguiente información para que el administrador de la cuenta pueda crear el rol de IAM:

- El ID AWS de tu cuenta
- Un `External ID` único que usará el cliente para identificarle. Para obtener más información, consulta Cómo utilizar un identificador externo al conceder acceso a tus AWS recursos a un tercero.
- Tienes que acceder a una lista de WorkDocs APIs las aplicaciones a las que debes acceder. El control basado en políticas de IAM proporciona un control granular y la capacidad para definir políticas que concedan o denieguen permisos en el nivel de API individual. Para ver la lista de WorkDocs APIs, consulte la referencia WorkDocs de la API.

En el procedimiento siguiente se describen los pasos que deben realizarse para configurar IAM para el acceso entre cuentas.

Para configurar IAM para el acceso entre cuentas

1. Cree una política WorkDocs de permisos de API, `WorkDocsAPIReadOnly` llámela política.
2. Cree un nuevo rol en la consola de IAM de la AWS cuenta que aloja el WorkDocs sitio:
   a. Inicie sesión en la consola de IAM AWS Management Console y ábrala en. https://console.aws.amazon.com/iam/
   b. En el panel de navegación de la consola, haga clic en Funciones y, a continuación, seleccione Crear nueva función.
   c. En Nombre de función, escriba un nombre de función que ayude a identificar el propósito de este rol (por ejemplo, `workdocs_app_role`). Los nombres de los roles deben ser únicos en su AWS cuenta. Cuando especifique el nombre haga clic en Paso siguiente.
   d. En la página Seleccionar tipo de función, seleccione la sección Función para acceso entre cuentas y, a continuación, elija el tipo de función que desea crear:
      - Seleccione Proporcionar acceso entre AWS las cuentas de su propiedad si es el administrador tanto de la cuenta de usuario como de la cuenta de recursos, o si ambas cuentas pertenecen a la misma empresa. Esta es también la opción que hay que elegir cuando los usuarios, la función y el recurso a los que se quiere obtener acceso se encuentran en la misma cuenta.
      - Seleccione Proporcionar acceso entre su AWS cuenta y una AWS cuenta de terceros si es el administrador de la cuenta propietaria del WorkDocs sitio y desea conceder permisos a los usuarios desde una cuenta de desarrollador de aplicaciones. Esta opción requiere que se especifique un ID externo (que el tercero debe proporcionarle) para obtener control adicional sobre las circunstancias en las que el tercero puede utilizar la función para obtener acceso a sus recursos. Para obtener más información, consulte Cómo utilizar un ID externo al otorgar acceso a los recursos de AWS a terceros.
   e. En la página siguiente, especifique el ID de AWS cuenta al que desea conceder acceso a sus recursos e introduzca también el ID externo en caso de que un tercero acceda a él.
   f. Haga clic en Siguiente paso para asociar una política.
3. En la página Adjuntar política, busca la política de permisos de la WorkDocs API que se creó anteriormente, selecciona la casilla situada junto a la política y haz clic en Siguiente paso.

4. Revise los detalles, copie el ARN de la función como referencia futura y haga clic en Crear función para terminar de crear la función.
5. Comparta el ARN de la función con el desarrollador. A continuación se muestra un ejemplo de un ARN de función:

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

Concesión de permisos de usuario para asumir un rol de IAM

Un desarrollador con una AWS cuenta administrativa puede permitir que un usuario asuma una función de IAM. Para ello, debe crear una política nueva y asociarla a ese usuario.

La política debe incluir una instrucción con el efecto `Allow` en la acción `sts:AssumeRole` y el Nombre de recurso de Amazon (ARN) del rol en un elemento `Resource`, tal y como se muestra en el siguiente ejemplo. Los usuarios que obtienen la política, ya sea a través de la pertenencia a un grupo o porque está asociada directamente, pueden cambiar al rol especificado.

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

Restringir el acceso a una instancia específica WorkDocs

Si tienes varios WorkDocs sitios en una AWS cuenta y quieres conceder acceso mediante API a un sitio específico, puedes definir un `Condition` elemento. El elemento `Condition` permite especificar condiciones que se aplican al entrar en vigor una política.

A continuación se muestra un ejemplo de un elemento de condición:

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

```
                }
    }
```

Con la condición anterior implementada en una política, los usuarios solo pueden acceder a la WorkDocs instancia con el ID `ded-123456789c5`. WorkDocs El ID de instancia a veces se denomina ID de organización o ID de directorio. Para obtener más información, consulte Restringir el acceso a una instancia específica WorkDocs .

Sigue estos pasos para obtener un ID de WorkDocs organización desde la AWS consola:

Para obtener un ID de organización

1. En el panel de navegación de la consola de AWS Directory Service, elija Directorios.
2. Anote el valor del ID de directorio que corresponde a su WorkDocs sitio. Es el ID de organización del sitio.

Autenticación y control de acceso para aplicaciones de usuario

WorkDocs las aplicaciones a nivel de usuario se registran y administran a través de la WorkDocs consola. Los desarrolladores deben registrar sus aplicaciones en la `My Applications` página de la WorkDocs consola, que será única IDs para cada aplicación. Durante el registro, los desarrolladores deben especificar la redirección URIs en la que recibirán los tokens de acceso, así como el alcance de las aplicaciones.

Actualmente, las aplicaciones solo pueden acceder a WorkDocs los sitios de la misma AWS cuenta en la que están registradas.

Contenido

- Otorgar permisos para llamar al WorkDocs APIs
- Uso de la carpeta IDs en las llamadas a la API
- Creación de una aplicación de
- Ámbitos de la aplicación
- Autorización
- Invocando WorkDocs APIs

Otorgar permisos para llamar al WorkDocs APIs

Los usuarios de la interfaz de línea de comandos deben tener permisos completos para WorkDocs y AWS Directory Service. Sin los permisos, cualquier llamada a la API devuelve UnauthorizedResourceAccessExceptionmensajes. La siguiente política concede permisos completos.

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

Si quiere conceder permisos de solo lectura, utilice esta política.

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

En la política, la primera acción otorga acceso a todas las WorkDocs `Describe` operaciones. La `DescribeDirectories` acción obtiene información sobre sus AWS Directory Service directorios. Las EC2 operaciones de Amazon permiten WorkDocs obtener una lista de sus subredes VPCs y sus subredes.

Uso de la carpeta IDs en las llamadas a la API

Siempre que una llamada a la API acceda a una carpeta, debe usar el ID de la carpeta, no el nombre de la carpeta. Por ejemplo, si lo `client.get_folder(FolderId='MyDocs')` apruebas, la llamada a la API devuelve un UnauthorizedResourceAccessExceptionmensaje y el siguiente mensaje 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.
```

Para evitarlo, use el ID de la URL de la carpeta.

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

Al pasar ese ID, se obtiene un resultado correcto.

```
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'}}
```

Creación de una aplicación de

Como WorkDocs administrador, crea tu aplicación siguiendo estos pasos.

Para crear una aplicación

1. Abra la WorkDocs consola en https://console.aws.amazon.com/zocalo/.
2. Elija Mis aplicaciones, Crear una aplicación.
3. Escriba los siguientes valores:

   Nombre de la aplicación

   Nombre de la aplicación.

   Correo electrónico

   Dirección de correo electrónico que se va a asociar a la aplicación.

   Descripción de la aplicación

   Descripción de la aplicación.

   Redirigir URIs

   La ubicación a la que WorkDocs quieres redirigir el tráfico.

   Ámbitos de la aplicación

   El ámbito (lectura o escritura) que desea que tenga la aplicación. Para obtener más información, consulta Ámbitos de la aplicación.
4. Seleccione Crear.

Ámbitos de la aplicación

WorkDocs admite los siguientes ámbitos de aplicación:

- Content Read (`workdocs.content.read`), que permite a la aplicación acceder a lo siguiente: WorkDocs APIs
  - Get*

  - Describe*
- Content Write (`workdocs.content.write`), que permite a tu aplicación acceder a lo siguiente WorkDocs APIs:
  - Create*
  - Update*
  - Delete*
  - Initiate*
  - Abort*
  - Add*
  - Remove*

Autorización

Una vez completado el registro de la solicitud, ésta puede solicitar la autorización en nombre de cualquier WorkDocs usuario. Para ello, la aplicación debe visitar el WorkDocs OAuth punto final y proporcionar los siguientes parámetros de consulta: `https://auth.amazonworkdocs.com/oauth`

- [Obligatorio] `app_id`: el ID de la aplicación generado cuando se registra una aplicación.
- [Obligatorio`oauth_type`]: el OAuth tipo de solicitud. El valor admitido es `ImplicitGrant`.
- [Obligatorio] `redirect_uri`: el URI de redirección registrado para una aplicación para recibir un token de acceso.
- [Opcional] `scopes`: una lista de ámbitos delimitada por comas. Si no se especifica, se usa la lista de ámbitos seleccionados durante el registro.
- [Opcional] `state`: una cadena que se devuelve junto con un token de acceso.

> **Note**
>
> Si necesita módulos criptográficos validados FIPS 140-2 al acceder a AWS a través de una interfaz de la línea de comandos o una API, utilice un punto de conexión de FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-2.

Un ejemplo de solicitud GET para iniciar el OAuth flujo y obtener un token de acceso:

```
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 el flujo de OAuth autorización, ocurre lo siguiente:

1. Se solicita al usuario de la aplicación que introduzca el nombre del WorkDocs sitio.
2. Se redirige al usuario a la página de WorkDocs autenticación para que introduzca sus credenciales.
3. Una vez realizada la autenticación, se le presenta al usuario la pantalla de consentimiento en la que puede conceder o denegar a la aplicación la autorización para obtener acceso a WorkDocs.
4. Cuando el usuario elige `Accept` en la pantalla de consentimiento, el navegador se redirige a la URL de devolución de llamada de la aplicación con el token de acceso y la información de la región como parámetros de consulta.

Un ejemplo de solicitud GET de WorkDocs:

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

Además del token de acceso, el WorkDocs OAuth servicio `region` también devuelve un parámetro de consulta para el WorkDocs sitio seleccionado. Las aplicaciones externas deben usar el `region` parámetro para determinar el punto final del WorkDocs servicio.

Si necesita módulos criptográficos validados FIPS 140-2 al acceder a AWS a través de una interfaz de la línea de comandos o una API, utilice un punto de conexión de FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulte Estándar de procesamiento de la información federal (FIPS) 140-2.

Invocando WorkDocs APIs

Después de obtener el token de acceso, la aplicación puede realizar llamadas a la API a servicios de WorkDocs.

> **Important**

En este ejemplo, se muestra cómo utilizar una solicitud GET de curl para obtener los metadatos de un documento.

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

Un ejemplo de JavaScript función para describir las carpetas raíz de un usuario:

```
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);
    });
}
```

A continuación se describe una invocación a una API basada en Java de ejemplo:

```
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 Gestor de contenido

WorkDocs El administrador de contenido es una herramienta de utilidad de alto nivel que carga contenido o lo descarga de un WorkDocs sitio.

Temas

- Construyendo un administrador WorkDocs de contenido
- Descarga de un documento
- Carga de un documento

Construyendo un administrador WorkDocs de contenido

Puede usar WorkDocs Content Manager para aplicaciones administrativas y de usuario.

En el caso de las aplicaciones de usuario, el desarrollador debe crear el gestor de WorkDocs contenido con AWS credenciales anónimas y un token de autenticación.

En el caso de las aplicaciones administrativas, el WorkDocs cliente debe inicializarse con credenciales AWS Identity and Access Management (IAM). Además, el token de autenticación debe omitirse en llamadas posteriores a la API.

El siguiente código muestra cómo inicializar el Gestor de WorkDocs contenido para aplicaciones de usuario mediante 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);
```

Descarga de un documento

Los desarrolladores pueden usar WorkDocs Content Manager para descargar una versión específica o la última versión de un documento. WorkDocs En los siguientes ejemplos se demuestra cómo descargar una versión específica de un documento mediante Java y C#.

> **Note**
>
> Para descargar la última versión de un documento, no especifique `VersionId` al crear al solicitud `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();
```

Carga de un documento

WorkDocs El administrador de contenido proporciona una API para cargar contenido a un sitio. WorkDocs En los siguientes ejemplos se demuestra cómo cargar un documento utilizando Java y 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();
```