
Amazon WorkDocs

Guía para desarrolladores



Amazon WorkDocs: Guía para desarrolladores

Copyright © 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 comerciales que no sean propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es Amazon WorkDocs?	1
Acceso	1
Precios	1
Recursos	1
Introducción	3
Connect a Amazon WorkDocs con credenciales de usuario de IAM	3
Connect a Amazon WorkDocs asumiendo un rol	4
Cargar un documento	6
Descargar un documento	7
Configuración de notificaciones para un usuario o función de IAM	7
Crear un usuario.	9
Conceder permisos a los usuarios de un recurso	10
Autenticación y control de acceso para aplicaciones administrativas	11
Conceder permiso a la API de Amazon WorkDocs a un desarrollador	11
Conceder permiso a un desarrollador externo	11
Conceder permiso a un desarrollador para que asuma una función de IAM	13
Restricción del acceso a una instancia de Amazon WorkDocs específica	13
Autenticación y control de acceso para aplicaciones de usuario	15
Cree una aplicación	15
Ámbitos de la aplicación	16
Autorización	16
Invocación de API de Amazon WorkDocs	17
Administrador de contenido de Amazon WorkDocs	19
Creación de Amazon WorkDocs Content Manager	19
Descarga de un documento	20
Carga de un documento	20
Glosario de AWS	22
.....	xxiii

¿Qué es Amazon WorkDocs?

Amazon WorkDocs es un servicio empresarial seguro de almacenamiento y uso compartido completamente administrado con controles administrativos estrictos y funciones de comentarios que mejoran 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 cliente de Amazon WorkDocs se pueden utilizar para ver muchos tipos de archivos diferentes, que dependen del tipo de medios de Internet del archivo. Amazon WorkDocs admite todos los formatos de imagen y documentos más comunes y constantemente se van agregando más tipos de medios.

Para obtener más información, consulte [Amazon WorkDocs](#).

Acceso

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

- 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.
- Amazon WorkDocs Drive se utiliza para sincronizar una carpeta del escritorio de Mac o de Windows con los archivos de Amazon WorkDocs.

Precios

Con Amazon WorkDocs, no hay cuotas de pago iniciales ni compromisos. Solo se paga por las cuentas de usuario activas y por el almacenamiento que se utiliza. 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 basados en roles y especializados, además de laboratorios autoguiados para ayudarlo a desarrollar sus conocimientos sobre AWS y obtener experiencia práctica.
- [Herramientas para desarrolladores de AWS](#): enlaces a herramientas para desarrolladores, SDK, conjuntos de herramientas de IDE y herramientas de línea de comandos para desarrollar y administrar aplicaciones de AWS.
- [Documentos técnicos de AWS](#): enlaces a una lista completa de documentos técnicos de AWS que tratan una gran variedad de temas técnicos, como arquitecturas, seguridad y economía de la nube, escritos por arquitectos de soluciones de AWS o expertos técnicos.
- [AWS Support Centro](#): el centro para crear y administrar los casos de AWS Support. También incluye enlaces a otros recursos útiles como foros, preguntas técnicas frecuentes, estado de los servicios y AWS Trusted Advisor.

- [AWS Support](#): la página web principal para obtener información acerca de AWS Support, un canal de soporte individualizado y de respuesta rápida que le ayudará a crear y ejecutar aplicaciones en la nube.
- [Contacto](#): un punto central de contacto para las consultas relacionadas con la facturación, cuentas, eventos, abuso y demás problemas relacionados con AWS.
- [Términos del sitio de AWS](#): información detallada acerca de nuestros derechos de autor y marca comercial, su cuenta, licencia y acceso al sitio, entre otros temas.

Introducción

Los siguientes fragmentos de código pueden ayudarle a empezar a utilizar el SDK de Amazon WorkDocs.

Ejemplos

- [Connect a Amazon WorkDocs con credenciales de usuario de IAM y consultar usuarios \(p. 3\)](#)
- [Connect a Amazon WorkDocs asumiendo un rol y examinar la carpeta raíz de un usuario. \(p. 4\)](#)
- [Cargar un documento \(p. 6\)](#)
- [Descargar un documento \(p. 7\)](#)
- [Configuración de notificaciones para un usuario o función de IAM \(p. 7\)](#)
- [Crear un usuario. \(p. 9\)](#)
- [Conceder permisos a los usuarios de un recurso \(p. 10\)](#)

Connect a Amazon WorkDocs con credenciales de usuario de IAM y consultar usuarios

El siguiente código, utilizando elAWSSDK, ilustra los pasos para realizar llamadas a la API mediante las credenciales de API de un usuario de IAM. En este caso, el usuario de la API y el sitio de Amazon WorkDocs pertenecen al mismo.AWSSaccount.

Asegúrese de que al usuario de IAM se le ha concedido acceso a la API de Amazon WorkDocs a través de una política de IAM adecuada.

En el ejemplo de código se usa la API describeUsers para buscar usuarios y obtener los metadatos de 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 se obtenga un ID de organización de Amazon WorkDocs.

Siga estos pasos para obtener un ID de organización de Amazon WorkDocs de laAWSConsole de :

Para obtener un ID de organización

1. En el panel de navegación de la [consola de AWS Directory Service](#), elija Directories (Directorios).
2. Tenga en cuenta elID de directoriovalor que corresponde a su sitio de Amazon WorkDocs. Es el ID de organización del sitio.

Este es el ejemplo de código:

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

Connect a Amazon WorkDocs asumiendo un rol y examinar la carpeta raíz de un usuario.

Este código de muestra, que usa elAWSJava SDK ilustra los distintos pasos para asumir un rol y usar las credenciales de seguridad temporales de la función para obtener acceso a Amazon WorkDocs. En el código de muestra se utiliza la API describeFolderContents para crear una lista de elementos presentes en 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, then ExternalId should
        // be set 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 temporary security credentials obtained
        // for 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

Utilice el siguiente procedimiento para cargar un documento en Amazon WorkDocs.

Para cargar un documento

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

Si va a usar las credenciales de usuario de IAM, consulte [Connect a Amazon WorkDocs con credenciales de usuario de IAM y consultar usuarios \(p. 3\)](#). Si asume una función de IAM, consulte [Connect a Amazon WorkDocs asumiendo un rol y examinar la carpeta raíz de un usuario. \(p. 4\)](#) para obtener más información.

```
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);
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. 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

Para descargar un documento de Amazon WorkDocs, obtenga una URL para la descarga como se indica a continuación y después utilice las acciones API proporcionadas por la 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 para un usuario o función de IAM

Para crear y administrar notificaciones en Amazon WorkDocs, los administradores utilizan las consolas IAM y Amazon WorkDocs. Utiliza la consola de IAM para establecer permisos de usuario y utiliza la consola de Amazon WorkDocs para habilitar las notificaciones. Una vez habilitadas las notificaciones, te suscribes a ellas. Siga estos pasos.

Para configurar los permisos de usuario

- Utilice la consola de IAM para establecer 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

Habilitar notificaciones le permite llamar `CreateNotificationSubscription` después de suscribirse a las notificaciones de.

1. Abra la consola de Amazon WorkDocs en <https://console.aws.amazon.com/zocalo/>.
2. En la página Administrar sus sitios de WorkDocs, seleccione el directorio que desee, y elija Acciones y Administrar notificaciones.
3. En la página Administrar notificaciones, elija Habilitar notificaciones.
4. Escriba el ARN del usuario o función al que desea permitir recibir notificaciones desde el sitio de Amazon WorkDocs.

Para obtener información sobre cómo habilitar Amazon WorkDocs para utilizar notificaciones, consulte [Uso de la API de Amazon WorkDocs con AWS SDK para Python y AWS Lambda](#). Una vez habilitadas las notificaciones, usted y su usuario pueden suscribirse a ellas.

Para suscribirse a las notificaciones de WorkDocs

1. Prepare el punto de enlace para procesar mensajes de Amazon SNS. Para obtener más información, consulte [Distribución ramificada a puntos de enlace HTTP/S](#) en la Guía para desarrolladores de Amazon Simple Notification Service.

Important

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

2. Haga lo siguiente:
 - Obtener un ID de organización
 1. En el navegador [Consola de AWS Directory Service](#) panel de navegación, seleccione Directorios.
 2. La ID de directorio correspondiente a su sitio de Amazon WorkDocs también sirve como ID de organización de ese sitio de.
 - 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 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. Abra la consola de Amazon WorkDocs en <https://console.aws.amazon.com/zocalo/>.
2. En la página Administrar sus sitios de WorkDocs, seleccione el directorio que desee, y elija Acciones y Administrar notificaciones.
3. En la página Administrar notificaciones, seleccione el ARN para el desea deshabilitar las notificaciones y elija Deshabilitar notificaciones.

Crear un usuario.

En el siguiente fragmento de código se muestra cómo crear solicitudes para crear un usuario en Amazon 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 API `ActivateUser` para activar el usuario en Amazon WorkDocs.

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

Siga estos pasos para obtener un ID de organización de Amazon WorkDocs de la AWS Consola de :

Para obtener un ID de organización

1. En el panel de navegación de la [consola de AWS Directory Service](#), elija Directories (Directorios).
2. Tenga en cuenta el ID de directorio o valor que corresponde a su sitio de Amazon WorkDocs. Es el ID de organización del sitio.

Conceder permisos a los usuarios de un recurso

En el siguiente fragmento de código se ilustra cómo crear una solicitud para añadir permisos a un usuario en un recurso. En este ejemplo vamos a añadir permisos CONTRIBUTOR a USER en un recurso. Esta API se puede usar también para conceder permisos a un usuario o grupo para 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

Las API administrativas de Amazon WorkDocs se autentican y autorizan mediante políticas de IAM. Los administradores de IAM pueden crear una política de IAM y asociarla a un rol o usuario de IAM que el desarrollador puede utilizar para obtener acceso a la API de.

Se proporcionan los siguientes ejemplos:

Tareas

- [Conceder permiso a la API de Amazon WorkDocs para un desarrollador en laAWSuenta \(p. 11\)](#)
- [Conceder permiso a la API de Amazon WorkDocs para desarrolladores de tercerosAWSuenta \(p. 11\)](#)
- [Conceder permiso a un desarrollador para que asuma una función de IAM de asignada por un cliente de Amazon WorkDocs \(p. 13\)](#)
- [Restricción del acceso a una instancia de Amazon WorkDocs específica \(p. 13\)](#)

Conceder permiso a la API de Amazon WorkDocs para un desarrollador en laAWSuenta

Si es un administrador de IAM, puede conceder acceso a la API de Amazon WorkDocs a un usuario de IAM desde el mismoAWSaccount. Para ello, cree una política de permisos de API de Amazon WorkDocs y asóciela al usuario de IAM. A continuación se incluye un ejemplo de política de API de Amazon WorkDocs que concede permiso a las API de solo lectura (API List y Describe).

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

Conceder permiso a la API de Amazon WorkDocs para desarrolladores de tercerosAWSuenta

Puede conceder acceso a desarrolladores de terceros o a usuarios que usen una cuenta de AWS diferente. Para ello, cree un rol de IAM y asocie políticas de permiso de API de Amazon WorkDocs.

Esta forma de acceso se requiere en los siguientes casos:

- El desarrollador pertenece a la misma organización, pero el desarrollador de AWS cuenta es diferente de Amazon WorkDocs AWS account.
- Cuando una compañía desea conceder acceso a la API de Amazon WorkDocs a desarrolladores de aplicaciones de terceros.

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

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

- Su ID de cuenta de AWS
- Un `External ID` único que usará el cliente para identificarle. Para obtener más información, consulte [Cómo utilizar un ID externo al otorgar acceso a los recursos de AWS a terceros](#).
- Una lista de las API de Amazon WorkDocs a las que necesita acceder la aplicación. El control basado en IAM proporciona un control granular y la capacidad para definir políticas de permitir o denegar en el nivel de API individual. Para ver la lista de API de Amazon WorkDocs, consulte [Referencia de API de Amazon WorkDocs](#).

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

Para configurar IAM para acceso entre cuentas

1. Cree una política de permisos de API de Amazon WorkDocs y denomínela `WorkDocsAPIReadOnly` política.
2. Cree un nuevo rol en la consola de IAM de la AWS cuenta que aloja el sitio de Amazon WorkDocs:
 - a. Inicie sesión en la AWS Management Console y abra la consola de IAM 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 rol deben ser únicos en su cuenta de AWS. 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 cuentas de AWS de su propiedad si es el administrador de la cuenta de usuario y de la cuenta del recurso, o si ambas cuentas pertenecen a la misma compañía. 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.
 - Select Proporcionar acceso entre su AWS cuenta y un tercero AWS cuenta si es el administrador de la cuenta que posee el sitio de Amazon WorkDocs 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 siguiente página, especifique el ID de cuenta de AWS a la que desea conceder acceso a sus recursos y escriba también el ID externo si se trata del acceso de un tercero.
 - f. Haga clic en Siguiente paso para asociar una política.

3. En la página **Asociar política**, busque la política de permisos de API de Amazon WorkDocs de que creó anteriormente, active la casilla situada junto a la política y haga clic en **Paso siguiente**.
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
```

Conceder permiso a un desarrollador para que asuma una función de IAM de asignada por un cliente de Amazon WorkDocs

Si es un desarrollador con una cuenta de AWS administrativa, puede conceder a un usuario permiso para cambiar a un rol mediante la creación de una nueva política y asociándola al usuario.

Una política que concede a un usuario permiso para asumir un rol 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"
  }
}
```

Restricción del acceso a una instancia de Amazon WorkDocs específica

Si tiene varios sitios de Amazon WorkDocs en un AWS y desea conceder acceso a la API de a un sitio específico, puede definir un elemento de condición. El elemento `Condition` permite especificar condiciones que se aplican al entrar en vigor una política.

A continuación se muestra un ejemplo de un elemento `condition`:

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


Con la condición anterior aplicada a una política, se permite a los usuarios tener acceso solo a la instancia de Amazon WorkDocs con ID.d-123456789c5. El ID de instancia de Amazon WorkDocs a veces se denomina ID de organización o ID de directorio.

Un ID de organización se denomina también ID de directorio o ID de instancia. Se puede utilizar para restringir el acceso a uno o varios sitios de Amazon WorkDocs de una cuenta. Para obtener más información, consulte [Restricción del acceso a una instancia de Amazon WorkDocs específica \(p. 13\)](#).

Siga estos pasos para obtener un ID de organización de Amazon WorkDocs delAWSConsola de :

Para obtener un ID de organización

1. En el panel de navegación de la [consola de AWS Directory Service](#), elija Directories (Directorios).
2. Tenga en cuenta laID de directoriovalor que corresponde a su sitio de Amazon WorkDocs. Es el ID de organización del sitio.

Autenticación y control de acceso para aplicaciones de usuario

Las aplicaciones de nivel de usuario de Amazon WorkDocs se registran y administran mediante la consola de Amazon WorkDocs. Los desarrolladores deben registrar sus aplicaciones en el `My Applications` en la consola de Amazon WorkDocs en la que se proporcionan ID únicos para cada aplicación. Durante el registro, los desarrolladores deben especificar los URI de redirección en los que recibirán los tokens de acceso, así como el ámbito de la aplicación.

Actualmente, las aplicaciones solo pueden obtener acceso a los sitios de Amazon WorkDocs en el mismo `AWS` cuenta en la que están registradas.

Contenido

- [Cree una aplicación \(p. 15\)](#)
- [Ámbitos de la aplicación \(p. 16\)](#)
- [Autorización \(p. 16\)](#)
- [Invocación de API de Amazon WorkDocs \(p. 17\)](#)

Cree una aplicación

Como administrador de Amazon WorkDocs, cree su aplicación mediante los siguientes pasos.

Para crear una aplicación

1. Abra la consola de Amazon WorkDocs 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.

URI de redirección

La ubicación en la que desea que Amazon WorkDocs redirija el tráfico.

Ámbitos de la aplicación

El ámbito (lectura o escritura) que desea que tenga la aplicación. Para obtener más información, consulte [Ámbitos de la aplicación \(p. 16\)](#).

4. Elija Create (Crear).

Ámbitos de la aplicación

Amazon WorkDocs admite los siguientes ámbitos de aplicación:

- Lectura de contenido (`workdocs.content.read`), que proporciona a la aplicación acceso a las siguientes API de Amazon WorkDocs siguientes:
 - Get*
 - Describe*
- Escritura de contenido (`workdocs.content.write`), que proporciona a la aplicación acceso a las siguientes API de Amazon WorkDocs siguientes:
 - Create*
 - Update*
 - Delete*
 - Initiate*
 - Abort*
 - Add*
 - Remove*

Autorización

Una vez completado el registro de la aplicación, una aplicación puede solicitar autorización en nombre de cualquier usuario de Amazon WorkDocs. Para ello, la aplicación debe visitar el punto de enlace OAuth de Amazon WorkDocs, <https://auth.amazonworkdocs.com/oauth>, y proporciona los siguientes parámetros de consulta:

- [Obligatorio]`app_id`: ID de aplicación generada cuando se registra una aplicación.
- [Obligatorio]`auth_type`: el tipo OAuth de la 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 línea de comandos o una API, utilice un punto de enlace de FIPS. Para obtener más información sobre los puntos de enlace de FIPS disponibles, consulte [Estándar de procesamiento de la información federal \(FIPS\) 140-2](#).

Una solicitud GET de ejemplo para iniciar el flujo de OAuth para 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
```

Estas son las operaciones que tienen lugar durante el flujo de autorización de OAuth:

1. Se pide al usuario de la aplicación que escriba el nombre de sitio de Amazon WorkDocs.

2. El usuario es redirigido a la página de autenticación de Amazon WorkDocs para escribir 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 Amazon 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.

Una solicitud GET de ejemplo desde Amazon WorkDocs:

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

Además del token de acceso, el servicio OAuth de Amazon WorkDocs devuelve también `region` como parámetro de consulta para el sitio de Amazon WorkDocs seleccionado. Las aplicaciones externas deben utilizar `region` para determinar el punto de enlace del servicio de Amazon WorkDocs.

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

Invocación de API de Amazon WorkDocs

Una vez obtenido el token de acceso, la aplicación puede realizar llamadas a la API de a los servicios de Amazon WorkDocs.

Una solicitud curl GET de ejemplo 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 acesstoken"
```

Una función JavaScript de ejemplo para describir las carpetas raíz del 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());
}
```

Administrador de contenido de Amazon WorkDocs

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

Temas

- [Creación de Amazon WorkDocs Content Manager \(p. 19\)](#)
- [Descarga de un documento \(p. 20\)](#)
- [Carga de un documento \(p. 20\)](#)

Creación de Amazon WorkDocs Content Manager

Amazon WorkDocs de contenido de puede usarse para las aplicaciones administrativas y de usuario.

Para las aplicaciones de usuario, el desarrollador de debe crear un administrador de contenido de Amazon WorkDocs con anónimasAWScredenciales y un token de autenticación.

Para las aplicaciones administrativas, el cliente de Amazon WorkDocs debe inicializarse conAWS Identity and Access Managementcredenciales (IAM). Además, el token de autenticación debe omitirse en llamadas posteriores a la API.

El siguiente código muestra cómo inicializar Amazon WorkDocs Content Manager para las aplicaciones de los usuarios 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").build();
```

C#:

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

Descarga de un documento

Los desarrolladores pueden usar Amazon WorkDocs de contenido de para descargar una versión específica o la última versión de un documento desde Amazon 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

Amazon WorkDocs de contenido de proporciona una API para cargar contenido en un sitio de Amazon 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();
```


Glosario de AWS

Para ver la terminología más reciente de AWS, consulte el [Glosario de AWS](#) en la Referencia general de AWS.

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