



Panduan Developer

Amazon WorkDocs



Amazon WorkDocs: Panduan Developer

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan antara para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan properti dari masing-masing pemilik, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau tidak.

Table of Contents

Apa itu Amazon WorkDocs?	1
Mengakses Amazon WorkDocs	1
Harga	1
Sumber daya	1
Memulai	3
Connect ke Amazon WorkDocs dengan kredensi pengguna IAM	3
Menghubungkan ke Amazon WorkDocs dengan mengambil peran	5
Mengunggah dokumen	8
Mengunduh dokumen	10
Menyiapkan notifikasi	10
Membuat pengguna	13
Memberikan pengguna izin ke sumber daya	14
Autentikasi dan kontrol akses untuk aplikasi administratif	16
Memberikan izin kepada pengembang ke Amazon WorkDocs API	16
Memberikan izin kepada pengembang pihak ketiga ke WorkDocs API Amazon	17
Memberikan izin kepada pengguna untuk mengambil IAM role	19
Membatasi akses ke WorkDocs instans Amazon tertentu	19
Autentikasi dan kontrol akses untuk aplikasi pengguna	21
Memberikan izin untuk memanggil Amazon WorkDocs API	21
Menggunakan ID folder dalam panggilan API	23
Membuat aplikasi	24
Cakupan aplikasi	24
Otorisasi	25
Menginvokasi Amazon WorkDocs API	26
Amazon WorkDocs Manager	28
Membangun Amazon WorkDocs Manager	28
Mengunduh dokumen	29
Mengunggah dokumen	30
Glosarium AWS	31
.....	xxxii

Apa itu Amazon WorkDocs?

Amazon WorkDocs adalah penyimpanan dokumen, kolaborasi, dan sistem berbagi. Amazon WorkDocs dikelola sepenuhnya, aman, dan skala perusahaan. Ini memberikan kontrol administratif yang kuat, ditambah kemampuan umpan balik yang membantu meningkatkan produktivitas pengguna. File tersimpan di dalam [cloud](#), secara aman dan terlindungi. File pengguna Anda hanya terlihat oleh mereka, serta kontributor dan pemirsa mereka yang ditunjuk. Anggota lain dari organisasi Anda tidak memiliki akses ke file pengguna lain kecuali mereka secara khusus diberikan akses.

Pengguna dapat berbagi file mereka dengan anggota lain dari organisasi Anda untuk kolaborasi atau peninjauan. Aplikasi WorkDocs klien Amazon dapat digunakan untuk melihat berbagai jenis file, tergantung pada jenis media Internet file itu. Amazon WorkDocs mendukung semua format dokumen dan citra yang umum, dan dukungan untuk jenis media lainnya terus ditambahkan.

Untuk informasi lebih lanjut, lihat [Amazon WorkDocs](#).

Mengakses Amazon WorkDocs

Pengguna akhir menggunakan aplikasi klien untuk mengakses file mereka. Pengguna nonadministratif tidak perlu menggunakan WorkDocs konsol Amazon atau dasbor administrasi. Amazon WorkDocs menawarkan beragam aplikasi klien dan utilitas:

- Aplikasi web yang digunakan untuk pengelolaan dan peninjauan dokumen.
- Aplikasi native untuk perangkat seluler yang digunakan untuk peninjauan dokumen.
- Amazon WorkDocs Drive digunakan untuk menyinkronkan folder di desktop Mac atau Windows Anda dengan WorkDocs file Amazon Anda.

Harga

Dengan Amazon WorkDocs, tidak ada biaya di muka atau komitmen. Anda hanya membayar untuk akun pengguna aktif, dan penyimpanan yang Anda gunakan. Untuk informasi selengkapnya, lihat [Harga](#).

Sumber daya

Sumber daya terkait berikut dapat membantu Anda ketika bekerja dengan layanan ini.

- [Kelas & Lokakarya](#) — Tautan ke kursus specialty dan berbasis peran, selain lab mandiri untuk membantu mempertajam AWS keterampilan Anda dan mendapatkan pengalaman praktis.
- [AWS Pusat Pengembang](#) - Jelajahi tutorial, alat unduh, dan pelajari tentang acara AWS pengembang.
- [AWS Alat Pengembang](#) — Tautan ke alat pengembang, SDKs, toolkit IDE, dan alat baris perintah untuk mengembangkan dan mengelola AWS aplikasi.
- [Memulai Pusat Sumber Daya](#) — Pelajari cara mengatur Akun AWS, bergabung dengan AWS komunitas, dan meluncurkan aplikasi pertama Anda.
- [Tutorial Hands-On](#) - Ikuti step-by-step tutorial untuk meluncurkan aplikasi pertama Anda AWS.
- [AWS Laporan Resmi](#) — Tautan ke daftar lengkap AWS laporan resmi teknis, yang mencakup topik seperti arsitektur, keamanan, dan ekonomi dan ditulis oleh Arsitek AWS Solusi atau ahli teknis lainnya.
- [AWS Support Center](#) — Hub untuk membuat dan mengelola AWS Support kasus Anda. Juga mencakup tautan ke sumber daya yang bermanfaat lainnya, seperti forum, FAQ teknis, status kondisi layanan, dan AWS Trusted Advisor.
- [AWS Support](#) — Halaman web utama untuk informasi tentang AWS Support, saluran dukungan jawaban cepat untuk membantu Anda membangun dan menjalankan aplikasi di cloud. one-on-one
- [Kontak Kami](#) – Titik kontak pusat untuk pertanyaan tentang tagihan AWS, akun, peristiwa, penyalahgunaan, dan masalah lainnya.
- [AWS Persyaratan Situs](#) – Informasi detail tentang hak cipta dan merek dagang kami; akun, lisensi, dan akses situs Anda; serta topik lainnya.

Memulai

Cuplikan kode berikut dapat membantu Anda mulai menggunakan Amazon WorkDocs SDK.

Note

Untuk keamanan yang lebih besar, buat pengguna federasi alih-alih pengguna IAM bila memungkinkan.

Contoh-contoh

- [Connect ke Amazon WorkDocs dengan kredensi pengguna IAM dan kueri untuk pengguna](#)
- [Menghubungkan ke Amazon WorkDocs dengan mengambil peran](#)
- [Mengunggah dokumen](#)
- [Mengunduh dokumen](#)
- [Menyiapkan notifikasi](#)
- [Membuat pengguna](#)
- [Memberikan pengguna izin ke sumber daya](#)

Connect ke Amazon WorkDocs dengan kredensi pengguna IAM dan kueri untuk pengguna

Kode berikut menunjukkan cara menggunakan kredensial API pengguna IAM untuk melakukan panggilan API. Dalam hal ini pengguna API dan WorkDocs situs Amazon milik AWS akun yang sama.

Note

Untuk keamanan yang lebih besar, buat pengguna federasi alih-alih pengguna IAM bila memungkinkan.

Pastikan bahwa pengguna IAM telah diberikan akses Amazon WorkDocs API melalui kebijakan IAM yang sesuai.

Contoh kode menggunakan [DescribeUsers](#) API untuk mencari pengguna, dan mendapatkan metadata untuk pengguna. Metadata pengguna memberikan detail seperti nama depan, nama belakang, ID pengguna, dan ID Folder root. ID folder root sangat membantu jika Anda ingin melakukan operasi unggahan atau unduhan konten apa pun atas nama pengguna.

Kode tersebut mengharuskan Anda mendapatkan ID WorkDocs Organisasi Amazon.

Ikuti langkah-langkah berikut untuk mendapatkan ID WorkDocs organisasi Amazon dari AWS konsol:

Untuk mendapatkan ID organisasi

1. Pada panel navigasi [konsol AWS Directory Service](#), pilih Direktori.
2. Perhatikan nilai ID Direktori yang sesuai dengan WorkDocs situs Amazon Anda. Itu adalah ID Organisasi untuk situs tersebut.

Contoh berikut menunjukkan cara menggunakan kredensial IAM untuk melakukan panggilan 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());
}
}
```

Menghubungkan ke Amazon WorkDocs dengan mengambil peran

Contoh ini menggunakan AWS Java SDK untuk mengambil peran dan menggunakan kredensial keamanan sementara peran untuk mengakses Amazon. WorkDocs Contoh kode menggunakan [DescribeFolderContents](#) API untuk mencantumkan item dalam folder pengguna.

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

Mengunggah dokumen

Note

Anda harus menjadi pengembang perangkat lunak untuk menyelesaikan langkah-langkah di bagian ini. Untuk informasi tentang menggunakan Amazon WorkDocs untuk mengunggah file, lihat [Mengunggah file](#) di Panduan WorkDocs Pengguna Amazon.

Gunakan prosedur berikut untuk mengunggah dokumen ke Amazon WorkDocs.

Mengunggah dokumen

1. Buat instans dari `AmazonWorkDocsClient` dengan cara berikut:

Jika Anda menggunakan kredensi pengguna IAM, lihat [Connect ke Amazon WorkDocs dengan kredensi pengguna IAM dan kueri untuk pengguna](#). Jika Anda mengambil peran IAM, lihat [Menghubungkan ke Amazon WorkDocs dengan mengambil peran](#) untuk informasi lebih lanjut.

Note

Untuk keamanan yang lebih besar, buat pengguna federasi alih-alih pengguna IAM bila memungkinkan.

```
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. Dapatkan URL ditandatangani untuk unggahan dengan cara berikut:

```
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. Unggah dokumen menggunakan URL ditandatangani dengan cara berikut:

```
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. Selesaikan proses pengunggahan dengan mengubah status dokumen menjadi ACTIVE dengan cara berikut:

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

Mengunduh dokumen

Note

Anda harus menjadi pengembang perangkat lunak untuk menyelesaikan langkah-langkah di bagian ini. Untuk informasi tentang menggunakan Amazon WorkDocs untuk mengunduh file, lihat [Mengunduh file](#) di Panduan WorkDocs Pengguna Amazon.

Untuk mengunduh dokumen dari Amazon WorkDocs, dapatkan URL untuk unduhan sebagai berikut, lalu gunakan tindakan API yang disediakan oleh platform pengembangan Anda untuk mengunduh file menggunakan 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());
```

Menyiapkan notifikasi

Anda mengikuti proses ini untuk mengatur notifikasi:

1. Setel izin Pengguna atau Peran IAM untuk memungkinkan pemanggil mengakses API manajemen langganan notifikasi.
2. Panggil API langganan notifikasi untuk mengaktifkan atau menonaktifkan penerbitan pesan SNS ke titik akhir Anda.

Note

Untuk keamanan yang lebih besar, buat pengguna federasi alih-alih pengguna IAM bila memungkinkan.

Untuk mengatur izin pengguna IAM

- Menggunakan konsol IAM untuk mengatur izin berikut untuk pengguna:

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

Untuk mengaktifkan notifikasi

Mengaktifkan notifikasi memungkinkan Anda untuk menelepon [CreateNotificationSubscription](#) setelah Anda berlangganan ke notifikasi.

1. Buka WorkDocs konsol Amazon di <https://console.aws.amazon.com/zocalo/>.
2. Pada halaman Kelola WorkDocs Situs Anda, pilih direktori yang diinginkan dan pilih Tindakan dan kemudian Kelola Pemberitahuan.
3. Pada Kelola Notifikasi, pilih Aktifkan Notifikasi.
4. Masukkan ARN untuk pengguna atau peran yang ingin Anda izinkan untuk menerima pemberitahuan dari situs Amazon WorkDocs Anda.

Untuk informasi tentang mengaktifkan Amazon WorkDocs menggunakan notifikasi, lihat [Menggunakan Amazon WorkDocs API dengan AWS SDK untuk Python dan AWS Lambda](#). Setelah Anda mengaktifkan notifikasi, Anda dan pengguna Anda kemudian dapat berlangganan ke notifikasi tersebut.

Untuk berlangganan WorkDocs notifikasi

1. Siapkan titik akhir Anda untuk memproses pesan Amazon SNS. Untuk informasi selengkapnya, lihat [Fanout ke titik akhir HTTP/S](#) di Panduan Developer Amazon Simple Notification Service.

Important

SNS mengirimkan pesan konfirmasi ke titik akhir yang dikonfigurasi. Anda harus mengonfirmasi pesan ini agar dapat menerima notifikasi. Jika Anda memerlukan modul kriptografi yang divalidasi FIPS 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Untuk informasi selengkapnya tentang titik akhir FIPS yang tersedia, lihat [Federal Information Processing Standard \(FIPS\) 140-2](#).

2. Lakukan hal berikut:
 - Dapatkan ID organisasi
 1. Di panel navigasi [konsol AWS Directory Service](#), pilih Direktori.
 2. ID Direktori yang sesuai dengan WorkDocs situs Amazon Anda juga berfungsi sebagai ID Organisasi untuk situs tersebut.
 - Buat permintaan langganan sebagai berikut:

```
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());
```

Notifikasi SNS

Pesan ini mencakup informasi berikut:

- `organizationId` — ID organisasi.
- `parentEntityType` — Jenis induk (Document | DocumentVersion | Folder).

- `parentEntityId` — ID dari induk.
- `entityType` — Jenis entitas (`Document` | `DocumentVersion` | `Folder`).
- `entityId` — ID dari induk.
- tindakan — Tindakan, yang bisa jadi tertera dalam salah satu nilai berikut:
 - `delete_document`
 - `move_document`
 - `recycle_document`
 - `rename_document`
 - `revoke_share_document`
 - `share_document`
 - `upload_document_version`

Untuk menonaktifkan notifikasi

1. Buka WorkDocs konsol Amazon di <https://console.aws.amazon.com/zocalo/>.
2. Pada halaman Kelola WorkDocs Situs Anda, pilih direktori yang diinginkan dan pilih Tindakan dan kemudian Kelola Pemberitahuan.
3. Pada Kelola Notifikasi, pilih ARN yang ingin Anda nonaktifkan notifikasinya dan pilih Nonaktifkan Notifikasi.

Membuat pengguna

Contoh berikut menunjukkan untuk membuat pengguna di Amazon WorkDocs.

Note

Ini bukan operasi yang valid untuk konfigurasi AD Terhubung. Untuk membuat pengguna dalam konfigurasi AD Terhubung, pengguna harus sudah ada di direktori korporasi. Kemudian, Anda harus melakukan panggilan ke [ActivateUser](#) API untuk mengaktifkan pengguna di Amazon WorkDocs.

Contoh berikut menunjukkan cara membuat pengguna dengan kuota penyimpanan 1 gigabyte.


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

Ikuti langkah-langkah berikut untuk mendapatkan ID WorkDocs organisasi Amazon dari AWS konsol:

Untuk mendapatkan ID organisasi

1. Pada panel navigasi [konsol AWS Directory Service](#), pilih Direktori.
2. Perhatikan nilai ID Direktori yang sesuai dengan WorkDocs situs Amazon Anda. Itu adalah ID Organisasi untuk situs tersebut.

Memberikan pengguna izin ke sumber daya

Contoh berikut menunjukkan cara menggunakan [AddResourcePermissions](#) API untuk memberikan CONTRIBUTOR izin ke sumber USER daya. Anda juga dapat menggunakan API untuk memberikan izin kepada pengguna atau grup pada folder atau dokumen.

```
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);
```

Autentikasi dan kontrol akses untuk aplikasi administratif

API WorkDocs administratif Amazon diautentikasi dan diotorisasi melalui kebijakan IAM.

Administrator IAM dapat membuat kebijakan IAM dan menyematkannya ke IAM role atau pengguna yang dapat digunakan oleh developer untuk mengakses API.

Berikut ini disediakan sebagai contoh:

Tugas

- [Memberikan izin kepada pengembang ke Amazon WorkDocs API](#)
- [Memberikan izin kepada pengembang pihak ketiga ke WorkDocs API Amazon](#)
- [Memberikan izin kepada pengguna untuk mengambil IAM role](#)
- [Membatasi akses ke WorkDocs instans Amazon tertentu](#)

Memberikan izin kepada pengembang ke Amazon WorkDocs API

Note

Untuk keamanan yang lebih besar, buat pengguna federasi alih-alih pengguna IAM bila memungkinkan.

Jika Anda seorang administrator IAM, Anda dapat memberikan akses WorkDocs API Amazon ke pengguna IAM dari AWS akun yang sama. Untuk melakukannya, buat kebijakan izin WorkDocs API Amazon dan sekatkan ke pengguna IAM. Kebijakan API berikut memberikan izin baca-saja ke berbagai Describe API.

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

```
    " * "
  ]
}
}
```

Memberikan izin kepada pengembang pihak ketiga ke WorkDocs API Amazon

Anda dapat memberikan akses ke developer pihak ketiga atau pengguna yang menggunakan akun AWS berbeda. Untuk melakukannya, buat IAM role, dan sekatkan kebijakan WorkDocs izin Amazon API.

Bentuk akses ini diperlukan dalam skenario berikut:

- Developer merupakan bagian dari organisasi yang sama tetapi AWS akun developer berbeda dari WorkDocs AWS akun Amazon.
- Ketika korporasi ingin memberikan akses WorkDocs API Amazon ke developer aplikasi pihak ketiga.

Dalam kedua skenario ini, ada dua AWS akun yang terlibat, satu akun developer, dan AWS akun berbeda yang meng-hosting WorkDocs situs Amazon.

Developer harus memberikan informasi berikut sehingga administrator akun dapat membuat IAM role:

- ID akun AWS Anda
- External ID unik yang akan digunakan pelanggan untuk mengidentifikasi Anda. Untuk informasi selengkapnya, lihat [Cara Menggunakan ID Eksternal Saat Memberikan Akses ke Sumber Daya AWS Anda ke Pihak Ketiga](#).
- Daftar WorkDocs API Amazon yang aksesnya diperlukan oleh aplikasi Anda. Kontrol kebijakan berbasis IAM menyediakan kontrol terperinci, kemampuan untuk menentukan kebijakan izin atau penolakan pada tingkat masing-masing API. Untuk mengetahui daftar WorkDocs API Amazon, lihat [Referensi WorkDocs API Amazon](#).

Prosedur berikut menjelaskan langkah-langkah yang diperlukan dalam mengonfigurasi IAM untuk akses lintas akun.

Untuk mengonfigurasi IAM untuk akses lintas akun

1. Buat kebijakan izin WorkDocs API Amazon, sebut sebagai `WorkDocsAPIReadOnly` kebijakan.
2. Buat peran baru di konsol IAM pada AWS akun yang meng-hosting WorkDocs situs Amazon:
 - a. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
 - b. Pada panel navigasi konsol, klik Peran dan kemudian klik Buat Peran Baru.
 - c. Untuk Nama peran, ketik nama peran yang membantu Anda mengidentifikasi tujuan peran ini, contohnya `workdocs_app_role`. Nama peran harus unik di akun AWS Anda. Setelah Anda memasukkan nama, klik Langkah Selanjutnya.
 - d. Pada Pilih Jenis Peran, pilih bagian Peran untuk Akses Lintas Akun, dan kemudian pilih jenis peran yang ingin Anda buat:
 - Pilih Berikan akses pada akun AWS yang Anda miliki jika Anda adalah administrator baik untuk akun pengguna maupun akun sumber daya, atau jika kedua akun merupakan bagian dari perusahaan yang sama. Ini juga merupakan opsi yang perlu dipilih jika pengguna, peran, dan sumber daya yang akan diakses semua ada dalam akun yang sama.
 - Pilih Berikan akses pada AWS akun Anda dan AWS akun pihak ketiga jika Anda adalah administrator akun yang memiliki WorkDocs situs Amazon dan Anda ingin memberikan izin kepada pengguna dari akun developer Aplikasi. Pilihan ini mengharuskan Anda untuk menentukan ID eksternal (yang harus diberikan pihak ketiga kepada Anda) agar bisa memberikan kontrol tambahan atas keadaan di mana pihak ketiga dapat menggunakan peran tersebut untuk mengakses sumber daya Anda. Untuk informasi selengkapnya, lihat [Cara Menggunakan ID Eksternal Saat Memberikan Akses ke Sumber Daya AWS Anda ke Pihak Ketiga](#).
 - e. Di halaman berikutnya, tentukan ID akun AWS yang ingin Anda beri akses ke sumber daya Anda dan masukkan juga ID eksternal jika perlu memberi izin untuk akses pihak ketiga.
 - f. Klik Langkah Selanjutnya untuk menyematkan kebijakan.
3. Pada halaman Sematkan Kebijakan, cari kebijakan izin WorkDocs API Amazon yang dibuat sebelumnya, pilih kotak di samping kebijakan, dan klik Langkah Selanjutnya.
4. Tinjau detailnya, salin ARN peran untuk referensi di masa mendatang, dan klik Buat Peran untuk menyelesaikan pembuatan peran.
5. Bagikan ARN peran dengan developer. Berikut ini adalah contoh ARN peran:

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

Memberikan izin kepada pengguna untuk mengambil IAM role

Pengembang dengan AWS akun administratif dapat memungkinkan pengguna untuk mengambil peran IAM. Untuk melakukan itu, Anda membuat kebijakan baru dan melampirkannya ke pengguna tersebut.

Kebijakan harus memuat pernyataan dengan `Allow` efek pada `sts:AssumeRole` tindakan, ditambah Amazon Resource Name (ARN) peran tersebut dalam `Resource` elemen, seperti yang ditunjukkan dalam contoh berikut. Pengguna yang mendapatkan kebijakan, melalui keanggotaan kelompok atau lampiran langsung, dapat beralih ke peran yang ditentukan.

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

Membatasi akses ke WorkDocs instans Amazon tertentu

Jika Anda memiliki beberapa WorkDocs situs Amazon pada AWS akun dan Anda ingin memberikan akses API ke situs tertentu, Anda dapat menentukan `Condition` elemen. Elemen `Condition` memungkinkan Anda menentukan syarat agar suatu kebijakan dapat berlaku.

Contoh berikut menunjukkan elemen syarat:

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

Jika syarat di atas terdapat dalam kebijakan, pengguna hanya dapat mengakses WorkDocs instans Amazon dengan IDd-123456789c5. ID WorkDocs Instans Amazon kadang-kadang disebut sebagai ID Organisasi atau ID Direktori. Untuk informasi selengkapnya, lihat [Membatasi akses ke WorkDocs instans Amazon tertentu](#).

Ikuti langkah-langkah berikut untuk mendapatkan ID WorkDocs organisasi Amazon dari AWS konsol:

Untuk mendapatkan ID organisasi

1. Pada panel navigasi [konsol AWS Directory Service](#), pilih Direktori.
2. Perhatikan nilai ID Direktori yang sesuai dengan WorkDocs situs Amazon Anda. Itu adalah ID Organisasi untuk situs tersebut.

Autentikasi dan kontrol akses untuk aplikasi pengguna

Amazon WorkDocs aplikasi tingkat pengguna terdaftar dan dikelola melalui Amazon WorkDocs konsol. Pengembang harus mendaftarkan aplikasi mereka di My Applications halaman di Amazon WorkDocs konsol yang akan memberikan ID unik untuk setiap aplikasi. Saat pendaftaran, pengembang harus menentukan URI pengalihan untuk menerima token akses serta cakupan aplikasi.

Saat ini, aplikasi hanya dapat mengakses Amazon WorkDocs situs dalam yang sama AWS akun tempat mereka terdaftar.

Daftar Isi

- [Memberikan izin untuk memanggil Amazon WorkDocs API](#)
- [Menggunakan ID folder dalam panggilan API](#)
- [Membuat aplikasi](#)
- [Cakupan aplikasi](#)
- [Otorisasi](#)
- [Menginvokasi Amazon WorkDocs API](#)

Memberikan izin untuk memanggil Amazon WorkDocs API

Pengguna antarmuka baris perintah harus memiliki izin penuh ke Amazon WorkDocs dan AWS Directory Service. Tanpa izin, panggilan API apa pun akan kembali UnauthorizedResourceAccessException pesan. Kebijakan berikut memberikan izin penuh.

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

```

Jika Anda ingin memberikan izin hanya-baca, gunakan kebijakan ini.

```

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

```

Dalam kebijakan tersebut, tindakan pertama memberikan akses ke semua Amazon WorkDocs Describe operasi. Yang Describe Directories tindakan memperoleh informasi tentang tindakan Anda memperoleh informasi tentang tindakan Anda AWS Directory Service direktori. Operasi Amazon EC2 memungkinkan Amazon WorkDocs untuk mendapatkan daftar VPC dan subnet Anda.

Menggunakan ID folder dalam panggilan API

Setiap kali panggilan API mengakses folder, Anda harus menggunakan ID folder, bukan nama folder. Misalnya, jika Anda `lulusclient.get_folder(FolderId='MyDocs')`, panggilan API mengembalikan `UnauthorizedResourceAccessException` pesan dan pesan 404 berikut.

```
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.
```

Untuk menghindarinya, gunakan ID di URL folder.

<site.workdocs/index.html#/folder/abc123def456ghi789jkl1789mno4be7024df198736472dd50ca970eb22796082e3d489577>.

Melewati ID yang mengembalikan hasil yang benar.

```
client.get_folder(FolderId='abc123def456ghi789jkl1789mno4be7024df198736472dd50ca970eb22796082e3d489577')
{'ResponseMetadata': {'RequestId': 'f8341d4e-4047-11e7-9e70-afa8d465756c',
  'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': 'f234564e-1234-56e7-89e7-a10fa45t789c', 'cache-control': 'private, no-cache, no-store, max-age=0',
  'content-type': 'application/json', 'content-length': '733', 'date': 'Wed, 24 May 2017 06:12:30 GMT'}, 'RetryAttempts': 0}, 'Metadata': {'Id': 'abc123def456ghi789jkl1789mno4be7024df198736472dd50ca970eb22796082e3d489577', 'Name': 'sentences', 'CreatorId': 'S-1-5-21-2125721135-1643952666-3011040551-2105&d-906724f1ce', 'ParentFolderId': '0a811a922403ae8e1d3c180f4975f38f94372c3d6a2656c50851c7fb76677363',
  'CreatedTimestamp': datetime.datetime(2017, 5, 23, 12, 59, 13, 8000, tzinfo=tzlocal()), 'ModifiedTimestamp': datetime.datetime(2017, 5, 23, 13, 13, 9, 565000, tzinfo=tzlocal()), 'ResourceState': 'ACTIVE', 'Signature': 'b7f54963d60ae1d6b9ded476f5d20511'}}}
```

Membuat aplikasi

Sebagai Amazon WorkDocs administrator, buat aplikasi Anda menggunakan langkah-langkah berikut.

Untuk membuat aplikasi

1. Buka Amazon WorkDocs konsol di <https://console.aws.amazon.com/zocalo/>.
2. Pilih Aplikasi Saya, Buat Aplikasi.
3. Masukkan nilai berikut:

Nama Aplikasi

Nama untuk aplikasi.

Email

Alamat email untuk diasosiasikan dengan aplikasi.

Deskripsi Aplikasi

Deskripsi untuk aplikasi.

URI Pengalihan

Lokasi yang Anda ingin Amazon lokasi yang Anda ingin Amazon WorkDocs untuk mengarahkan lalu lintas ke.

Cakupan Aplikasi

Cakupan, baik baca atau tulis, yang Anda ingin atur untuk aplikasi. Untuk detail selengkapnya, lihat [Cakupan aplikasi](#).

4. Pilih Buat.

Cakupan aplikasi

Amazon WorkDocs mendukung cakupan aplikasi berikut:

- Baca Konten (`workdocs.content.read`), yang memberikan akses aplikasi Anda ke Amazon berikut WorkDocs API:
- **Dapatkan***

- Deskripsikan*
- Tulis konten (`workdocs.content.write`), yang memberikan akses aplikasi Anda ke Amazon berikut WorkDocs API:
 - Buat*
 - Perbarui*
 - Hapus*
 - Mulai*
 - Batalkan*
 - Tambahkan*
 - Hapus*

Otorisasi

Setelah pendaftaran aplikasi selesai, aplikasi dapat meminta otorisasi atas nama Amazon mana pun WorkDocs pengguna pengguna. Untuk melakukan ini, aplikasi harus mengunjungi Amazon WorkDocs titik akhir OAuth, `https://auth.amazonworkdocs.com/oauth`, dan memberikan parameter query berikut:

- [Diperlukan] `app_id`—ID aplikasi yang dihasilkan ketika aplikasi didaftarkan.
- [Diperlukan] `auth_type`—Tipe OAuth untuk permintaan. Nilai yang didukung adalah `ImplicitGrant`.
- [Diperlukan] `redirect_uri`—URI pengalihan yang didaftarkan untuk aplikasi untuk menerima token akses.
- [Opsional] `scopes`—Daftar cakupan yang dipisahkan koma. Jika tidak ditentukan, daftar cakupan yang dipilih saat pendaftaran akan digunakan.
- [Opsional] `state`—Sebuah string yang dihasilkan bersama dengan token akses.

Note

Jika Anda memerlukan modul kriptografi yang divalidasi FIPS 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Untuk informasi lebih lanjut tentang titik akhir FIPS yang tersedia, lihat [Standar Pemrosesan Informasi Federal \(FIPS\) 140-2](#).

Sampel permintaan GET untuk memulai aliran OAuth untuk mendapatkan token akses:

```
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
```

Berikut ini terjadi saat aliran otorisasi OAuth:

1. Pengguna aplikasi diminta untuk memasuki Amazon WorkDocs nama situs web.
2. Pengguna diarahkan ke Amazon WorkDocs laman autentikasi untuk memasukkan kredensialnya.
3. Setelah autentikasi berhasil, pengguna disajikan dengan layar persetujuan yang memungkinkan pengguna untuk memberikan atau menolak otorisasi untuk mengakses Amazon untuk aplikasi Anda WorkDocs.
4. Setelah pengguna memilih Accept pada layar persetujuan, peramban mereka diarahkan ke URL callback aplikasi Anda dengan token akses dan informasi wilayah sebagai parameter kuerinya.

Sampel permintaan GET sampel dari Amazon WorkDocs:


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

Selain token akses, Amazon WorkDocs Layanan OAuth juga kembali `region` sebagai parameter kueri untuk Amazon yang dipilih WorkDocs situs web. Aplikasi eksternal harus menggunakan `region` parameter untuk menentukan Amazon WorkDocs titik akhir layanan.

Jika Anda memerlukan modul kriptografi yang divalidasi FIPS 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Untuk informasi selengkapnya tentang titik akhir FIPS yang tersedia, lihat [Standar Pemrosesan Informasi Federal \(FIPS\) 140-2](#).

Menginvokasi Amazon WorkDocs API

Setelah mendapatkan token akses, aplikasi Anda dapat membuat panggilan API ke Amazon WorkDocslayanan.

 Important

Contoh ini menunjukkan bagaimana menggunakan permintaan GET curl untuk mendapatkan metadata dokumen.

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

Sampel sampel sampel JavaScript fungsi untuk mendeskripsikan folder root pengguna:

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

Sampel pengaktifan API berbasis Java dijelaskan di bawah ini:

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

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

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

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

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

Amazon WorkDocs Manager

Amazon WorkDocs Content Manager adalah alat utilitas tingkat tinggi yang mengunggah konten atau mengunduhnya dari Amazon WorkDocs Situs

Topik

- [Membangun Amazon WorkDocs Manager](#)
- [Mengunduh dokumen](#)
- [Mengunggah dokumen](#)

Membangun Amazon WorkDocs Manager

Anda dapat menggunakan Amazon WorkDocs Content Manager untuk aplikasi administrasi dan pengguna.

Untuk aplikasi pengguna, developer harus membangun Amazon WorkDocs Content ManagerAWSkredensial dan token autentikasi.

Untuk aplikasi administratif, Amazon WorkDocs klien harus diinisialisasi denganAWS Identity and Access Management(IAM) mandat. Selain itu, token autentikasi harus dihilangkan dalam panggilan API berikutnya.

Kode berikut menunjukkan cara menginisialisasi Amazon WorkDocs Content Manager untuk aplikasi pengguna menggunakan Java atau 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(params);
```

Mengunduh dokumen

Pengembang dapat menggunakan Amazon WorkDocs Content Manager untuk mengunduh versi tertentu atau versi terbaru dokumen dari Amazon WorkDocs. Contoh berikut menunjukkan cara mengunduh versi tertentu dari dokumen menggunakan Java dan C#.

Note

Untuk mengunduh versi terbaru dokumen, jangan tentukan `VersionId` ketika membuat permintaan `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();
```

Mengunggah dokumen

Amazon WorkDocs Content Manager menyediakan API untuk mengunggah konten ke Amazon WorkDocs Situs Contoh berikut menunjukkan cara mengunggah versi tertentu dokumen menggunakan Java dan 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();
```

Glosarium AWS

Lihat terminologi AWS terbaru di [AWS glosarium](#) dalam Referensi Glosarium AWS.

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.