



Panduan Developer

Amazon WorkDocs



Amazon WorkDocs: Panduan Developer

Copyright © 2026 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 di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

.....	iv
Apa itu Amazon WorkDocs?	1
Mengakses WorkDocs	1
Harga	1
Sumber daya	1
Memulai	3
Connect to WorkDocs dengan kredensi pengguna IAM	3
Menghubungkan WorkDocs dengan mengasumsikan 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 pengembang ke API WorkDocs	16
Memberikan izin pengembang pihak ketiga ke WorkDocs APIs	17
Memberikan izin kepada pengguna untuk mengambil peran IAM	19
Membatasi akses ke instance tertentu WorkDocs	19
Autentikasi dan kontrol akses untuk aplikasi pengguna	21
Memberikan izin untuk memanggil WorkDocs APIs	21
Menggunakan folder IDs dalam panggilan API	23
Membuat aplikasi	24
Cakupan aplikasi	24
Otorisasi	25
Memohon WorkDocs APIs	26
WorkDocs Pengelola Konten	28
Membangun Pengelola WorkDocs Konten	28
Mengunduh dokumen	29
Mengunggah dokumen	30

Pemberitahuan: Pendaftaran pelanggan baru dan peningkatan akun tidak lagi tersedia untuk Amazon. WorkDocs Pelajari tentang langkah-langkah migrasi di sini: [Cara memigrasi data dari WorkDocs](#).

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

Apa itu Amazon WorkDocs?

Amazon WorkDocs adalah penyimpanan dokumen, kolaborasi, dan sistem berbagi. WorkDocs sepenuhnya dikelola, 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 dapat digunakan untuk melihat berbagai jenis file, tergantung pada jenis media Internet dari file tersebut. WorkDocs mendukung semua format dokumen dan gambar umum, dan dukungan untuk jenis media tambahan terus ditambahkan.

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

Mengakses WorkDocs

Pengguna akhir menggunakan aplikasi klien untuk mengakses file mereka. Pengguna non-administratif tidak perlu menggunakan WorkDocs konsol atau dasbor administrasi. WorkDocs menawarkan beberapa aplikasi dan utilitas klien yang berbeda:

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

Harga

Dengan WorkDocs, tidak ada biaya atau komitmen di muka. 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 berbasis peran dan khusus, selain laboratorium mandiri untuk membantu mempertajam keterampilan Anda AWS dan mendapatkan pengalaman praktis.
- [AWS Pusat Pengembang](#) — Jelajahi tutorial, unduh alat, 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 menyiapkan Akun AWS, bergabung dengan AWS komunitas, dan meluncurkan aplikasi pertama Anda.
- [Hands-On Tutorial](#) - Ikuti step-by-step tutorial untuk meluncurkan aplikasi pertama Anda. AWS
- [AWS Whitepaper](#) — Tautan ke daftar lengkap AWS whitepaper teknis, yang mencakup topik-topik seperti arsitektur, keamanan, dan ekonomi dan ditulis oleh AWS Solutions Architects atau pakar teknis lainnya.
- [AWS Dukungan Pusat](#) — Hub untuk membuat dan mengelola AWS Dukungan kasus Anda. Juga termasuk tautan ke sumber daya bermanfaat lainnya, seperti forum, teknis FAQs, status kesehatan layanan, dan AWS Trusted Advisor.
- [Dukungan](#)— Halaman web utama untuk informasi tentang Dukungan, saluran dukungan respons cepat untuk membantu Anda membangun dan menjalankan aplikasi di cloud. one-on-one
- [Hubungi Kami](#) – Titik kontak pusat untuk pertanyaan tentang tandaihan AWS , akun, peristiwa, penyalahgunaan, dan masalah lainnya.
- [AWS Ketentuan Situs](#) — Informasi terperinci tentang hak cipta dan merek dagang kami; akun, lisensi, dan akses situs Anda; dan topik lainnya.

Memulai

Cuplikan kode berikut dapat membantu Anda mulai menggunakan SDK WorkDocs .

Note

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

Contoh

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

Connect to 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 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 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.

Ikuti langkah-langkah berikut untuk mendapatkan ID WorkDocs organisasi 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 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 WorkDocs dengan mengasumsikan peran

Contoh ini menggunakan AWS Java SDK untuk mengambil peran dan menggunakan kredensial keamanan sementara peran untuk mengakses 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 penggunaan WorkDocs untuk mengunggah file, lihat [Mengunggah file](#) di Panduan WorkDocs Pengguna.

Gunakan prosedur berikut untuk mengunggah dokumen ke WorkDocs.

Mengunggah dokumen

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

Jika Anda menggunakan kredensi pengguna IAM, lihat [Connect to WorkDocs dengan kredensi pengguna IAM dan kueri untuk pengguna](#). Jika Anda mengambil peran IAM, lihat [Menghubungkan WorkDocs dengan mengasumsikan 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 penggunaan WorkDocs untuk mengunduh file, lihat [Mengunduh file](#) di Panduan WorkDocs Pengguna.

Untuk mengunduh dokumen 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 manajemen langganan notifikasi. APIs
2. Panggil langganan notifikasi APIs 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 mengizinkan Anda untuk menelepon [CreateNotificationSubscription](#) setelah Anda berlangganan ke notifikasi.

1. Buka WorkDocs konsol 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 Anda WorkDocs .

Untuk informasi tentang mengaktifkan pemberitahuan, lihat [Menggunakan Amazon WorkDocs API dengan AWS SDK WorkDocs 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 HTTP/S titik akhir](#) di Panduan Pengembang Layanan Pemberitahuan Sederhana Amazon.

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 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 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 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(10485761));
request.setStorageRule(storageRule);
CreateUserResult result = workDocsClient.createUser(request);
```

Ikuti langkah-langkah berikut untuk mendapatkan ID WorkDocs organisasi 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 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

WorkDocs administratif APIs 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 pengembang ke API WorkDocs](#)
- [Memberikan izin pengembang pihak ketiga ke WorkDocs APIs](#)
- [Memberikan izin kepada pengguna untuk mengambil peran IAM](#)
- [Membatasi akses ke instance tertentu WorkDocs](#)

Memberikan izin pengembang ke API WorkDocs

Note

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

Jika Anda adalah administrator IAM, Anda dapat memberikan akses WorkDocs API ke pengguna IAM dari akun yang sama AWS . Untuk melakukan ini, buat kebijakan izin WorkDocs API dan lampirkan ke pengguna IAM. Kebijakan API berikut memberikan izin hanya-baca ke berbagai kebijakan.

Describe APIs

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

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

Memberikan izin pengembang pihak ketiga ke WorkDocs APIs

Anda dapat memberikan akses ke pengembang pihak ketiga, atau kepada pengguna yang menggunakan AWS akun lain. Untuk melakukannya, buat peran IAM, dan lampirkan kebijakan WorkDocs izin API.

Bentuk akses ini diperlukan dalam skenario berikut:

- Pengembang milik organisasi yang sama tetapi AWS akun pengembang berbeda dari WorkDocs AWS akun.
- Ketika perusahaan ingin memberikan akses WorkDocs API ke pengembang aplikasi pihak ketiga.

Dalam kedua skenario ini, ada dua AWS akun yang terlibat, AWS akun pengembang dan akun lain yang menghosting WorkDocs situs.

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

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

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, sebut saja `WorkDocsAPIReadOnly` kebijakan.

2. Buat peran baru di konsol IAM AWS akun hosting WorkDocs situs:
 - a. Masuk ke Konsol Manajemen AWS 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 dalam AWS akun 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 antar AWS akun yang Anda miliki jika Anda adalah administrator akun pengguna dan akun sumber daya, atau kedua akun milik 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 antara AWS akun Anda dan AWS akun pihak ketiga jika Anda adalah administrator akun yang memiliki WorkDocs situs dan Anda ingin memberikan izin kepada pengguna dari akun pengembang 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. Pada halaman berikutnya, tentukan ID AWS akun yang ingin Anda berikan akses ke sumber daya Anda dan juga masukkan ID Eksternal jika ada akses pihak ketiga.
 - f. Klik Langkah Selanjutnya untuk menyematkan kebijakan.
3. Pada halaman Lampirkan Kebijakan, cari kebijakan izin WorkDocs API yang dibuat sebelumnya, lalu pilih kotak di sebelah kebijakan, lalu klik Langkah Berikutnya.
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 peran IAM

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

Kebijakan harus menyertakan pernyataan dengan Allow efek pada `sts:AssumeRole` tindakan, ditambah Nama Sumber Daya Amazon (ARN) peran dalam Resource elemen, seperti yang ditunjukkan pada contoh berikut. Pengguna yang mendapatkan kebijakan, baik melalui keanggotaan grup 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 instance tertentu WorkDocs

Jika Anda memiliki beberapa WorkDocs situs di 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 kondisi:

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

Dengan kondisi di atas dalam kebijakan, pengguna hanya dapat mengakses WorkDocs instance dengan IDd-123456789c5. WorkDocs ID Instance kadang-kadang disebut sebagai ID Organisasi atau ID Direktori. Untuk informasi selengkapnya, lihat [Membatasi akses ke instance tertentu WorkDocs](#).

Ikuti langkah-langkah berikut untuk mendapatkan ID WorkDocs organisasi 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 Anda. Itu adalah ID Organisasi untuk situs tersebut.

Autentikasi dan kontrol akses untuk aplikasi pengguna

WorkDocs aplikasi tingkat pengguna terdaftar dan dikelola melalui WorkDocs konsol. Pengembang harus mendaftarkan aplikasi mereka pada My Applications halaman di WorkDocs konsol yang akan memberikan unik IDs untuk setiap aplikasi. Selama pendaftaran, pengembang harus menentukan pengalihan URIs di mana mereka akan menerima token akses serta cakupan aplikasi.

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

Daftar Isi

- [Memberikan izin untuk memanggil WorkDocs APIs](#)
- [Menggunakan folder IDs dalam panggilan API](#)
- [Membuat aplikasi](#)
- [Cakupan aplikasi](#)
- [Otorisasi](#)
- [Memohon WorkDocs APIs](#)

Memberikan izin untuk memanggil WorkDocs APIs

Pengguna antarmuka baris perintah harus memiliki izin penuh untuk WorkDocs dan Directory Service. Tanpa izin, API apa pun memanggil UnauthorizedResourceAccessException pesan balasan. 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 WorkDocs Describe operasi. DescribeDirectories Tindakan memperoleh informasi tentang Directory Service direktori Anda. EC2 Operasi Amazon memungkinkan WorkDocs untuk mendapatkan daftar subnet VPCs dan Anda.

Menggunakan folder IDs 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 akan menampilkan `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/abc123def456ghi789jkl789mno4be7024df198736472dd50ca970eb22796082e3d489577`.

Melewati ID itu mengembalikan hasil yang benar.

```
client.get_folder(FolderId='abc123def456ghi789jkl789mno4be7024df198736472dd50ca970eb22796082e3d489577')
{'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, 9, 565000, tzinfo=tzlocal()), 'ResourceState': 'ACTIVE', 'Signature': 'b7f54963d60ae1d6b9ded476f5d20511'}}
```

Membuat aplikasi

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

Untuk membuat aplikasi

1. Buka 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.

Mengalihkan URIs

Lokasi yang ingin WorkDocs Anda alihkan lalu lintas ke.

Lingkup Aplikasi

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

4. Pilih Buat.

Cakupan aplikasi

WorkDocs mendukung cakupan aplikasi berikut:

- Content Read (`workdocs.content.read`), yang memberikan akses aplikasi Anda ke hal-hal berikut WorkDocs APIs:
 - Dapatkan*
 - Deskripsikan*

- Content Write (`workdocs.content.write`), yang memberikan akses aplikasi Anda ke hal-hal berikut WorkDocs APIs:
 - Buat*
 - Perbarui*
 - Hapus*
 - Mulai*
 - Batalkan*
 - Tambahkan*
 - Hapus*

Otorisasi

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

- [Diperlukan] `app_id`—ID aplikasi yang dihasilkan ketika aplikasi didaftarkan.
- [Wajib] `auth_type` OAuth —Jenis 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 selengkapnya tentang titik akhir FIPS yang tersedia, lihat [Standar Pemrosesan Informasi Federal \(FIPS\) 140-2](#).

Contoh permintaan GET untuk memulai OAuth alur 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 selama aliran OAuth otorisasi:

1. Pengguna aplikasi diminta untuk memasukkan nama WorkDocs situs.
2. Pengguna diarahkan ke halaman WorkDocs otentikasi untuk memasukkan kredensialnya.
3. Setelah otentikasi berhasil, pengguna disajikan dengan layar persetujuan yang memungkinkan pengguna untuk memberikan atau menolak otorisasi aplikasi Anda untuk mengakses 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.

Contoh GET permintaan dari WorkDocs:

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

Selain token akses, WorkDocs OAuth layanan juga kembali `region` sebagai parameter kueri untuk WorkDocs situs yang dipilih. Aplikasi eksternal harus menggunakan `region` parameter untuk menentukan titik akhir WorkDocs 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](#).

Memohon WorkDocs APIs

Setelah mendapatkan token akses, aplikasi Anda dapat melakukan panggilan API ke WorkDocs layanan.

Important

Contoh ini menunjukkan cara 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"
```

Contoh 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());  
}
```

WorkDocs Pengelola Konten

WorkDocs Content Manager adalah alat utilitas tingkat tinggi yang mengunggah konten atau mengunduhnya dari situs. WorkDocs

Topik

- [Membangun Pengelola WorkDocs Konten](#)
- [Mengunduh dokumen](#)
- [Mengunggah dokumen](#)

Membangun Pengelola WorkDocs Konten

Anda dapat menggunakan WorkDocs Content Manager untuk aplikasi administratif dan pengguna.

Untuk aplikasi pengguna, pengembang harus membangun WorkDocs Content Manager dengan AWS kredensi anonim dan token otentikasi.

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

Kode berikut menunjukkan cara menginisialisasi 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").build()
```

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 Pengelola WorkDocs Konten untuk mengunduh versi tertentu atau versi terbaru dari dokumen 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

WorkDocs Content Manager menyediakan API untuk mengunggah konten ke 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();
```