



Guia do Desenvolvedor

Amazon WorkDocs



Amazon WorkDocs: Guia do Desenvolvedor

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

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

.....	iv
O que é o Amazon WorkDocs?	1
Acessar o Amazon WorkDocs	1
Preços	1
Recursos	2
Conceitos básicos	3
Conecte-se ao Amazon WorkDocs com credenciais de usuário do IAM	3
Conectando-se ao Amazon WorkDocs assumindo uma função	5
Fazer upload de um documento	8
Fazer download de um documento	9
Configuração de notificações para um usuário ou um perfil do IAM	10
Criação de um usuário	13
Conceder permissões aos usuários para um recurso	14
Controle de acesso e autenticação de aplicativos administrativos	15
Conceder permissões aos desenvolvedores para a API do Amazon WorkDocs	15
Conceder permissão a desenvolvedores terceirizados para as APIs do Amazon WorkDocs	16
Conceder permissão aos usuários para assumirem um perfil do IAM	18
Restringir o acesso a uma instancia do Amazon WorkDocs específica	19
Controle de acesso e autenticação de aplicativos de usuário	20
Conceder permissões para chamar as APIs do Amazon WorkDocs	20
Usando IDs de pasta em chamadas de API	22
Criar uma aplicação	23
Escopos do aplicativo	23
Autorização	24
Invocando as APIs do Amazon WorkDocs	25
Gerenciador de conteúdo do Amazon WorkDocs	27
Construindo o Amazon WorkDocs Content Manager	27
Fazer download de um documento	28
Fazer upload de um documento	29

Aviso: novas inscrições de clientes e atualizações de conta não estão mais disponíveis para a Amazon. WorkDocs Saiba mais sobre as etapas de migração aqui: [Como migrar dados da Amazon WorkDocs](#).

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.

O que é o Amazon WorkDocs?

O Amazon WorkDocs é um sistema de armazenamento, colaboração e compartilhamento de documentos. O Amazon WorkDocs é totalmente gerenciado, seguro e em escala empresarial. Ele fornece controles administrativos robustos, além de recursos de feedback que ajudam a melhorar a produtividade do usuário. Os arquivos são armazenados na [nuvem](#) com segurança. Os arquivos dos usuários ficam visíveis apenas para eles, seus colaboradores e visualizadores designados. Outros membros da sua organização não têm acesso a arquivos de outros usuários a não ser que você conceda o acesso a eles especificamente.

Os usuários podem compartilhar seus arquivos com outros membros da organização para colaboração ou revisão. Os aplicativos cliente do Amazon WorkDocs podem ser usados para visualizar vários tipos diferentes de arquivos, dependendo do tipo de mídia do arquivo na internet. O Amazon WorkDocs oferece suporte a todos os formatos comuns de documentos e imagens, e o suporte para outros tipos de mídia é constantemente adicionado.

Para obter mais informações, consulte [Amazon WorkDocs](#).

Acessar o Amazon WorkDocs

Os usuários finais usam os aplicativos cliente acessar seus arquivos. Usuários não administrativos nunca precisam usar o console do Amazon WorkDocs ou o painel de administração. O Amazon WorkDocs oferece vários aplicativos e utilitários clientes diferentes:

- Aplicativo web usado para gerenciamento e análise de documentos.
- Aplicativos nativos para dispositivos móveis usados para análise de documentos.
- O Amazon WorkDocs Drive é usado para sincronizar uma pasta em seu desktop Mac ou Windows com seus arquivos do Amazon WorkDocs.

Preços

Com o Amazon WorkDocs, não há taxas iniciais ou compromissos. Você paga somente pelas contas de usuário ativas e pelo armazenamento que você usa. Para obter mais informações, acesse a [Preços](#).

Recursos

Os recursos relacionados a seguir podem ajudar você à medida que trabalha com este serviço.

- [Aulas e workshops](#) — Links para cursos de especialidades e baseados em perfil, bem como laboratórios autoguiados para ajudar a aperfeiçoar suas habilidades na AWS e a obter experiência prática.
- [Centro dos desenvolvedores da AWS](#) — Explore tutoriais, baixe ferramentas e informe-se sobre eventos para desenvolvedores da AWS.
- [Ferramentas do desenvolvedor da AWS](#) — Links para ferramentas de desenvolvedor, SDKs, toolkits de IDE e ferramentas da linha de comando para desenvolver e gerenciar aplicativos da AWS.
- [Centro de recursos de conceitos básicos](#) — Saiba como configurar a Conta da AWS, participar da comunidade da AWS e lançar seu primeiro aplicativo.
- [Tutoriais práticos](#) — Siga os tutoriais passo a passo para iniciar seu primeiro aplicativo na AWS.
- [Whitepapers da AWS](#) — Links para uma lista abrangente de whitepapers técnicos da AWS que abrangem tópicos como arquitetura, segurança e economia, elaborados pelos arquitetos de soluções da AWS ou por outros especialistas técnicos.
- [AWS Support Center](#): a central para criar e gerenciar seus casos do AWS Support. Também inclui links para outros recursos úteis, como fóruns, perguntas frequentes técnicas, status de integridade do serviço e AWS Trusted Advisor.
- [AWS Support](#): a página web principal para obter informações sobre o AWS Support, um canal de suporte de resposta rápida e com atendimento individual para ajudar a construir e a executar aplicativos na nuvem.
- [Entrar em contato](#): um ponto central de contato para consultas relativas a faturas da AWS, contas, eventos, uso abusivo e outros problemas.
- [Termos do site da AWS](#): informações detalhadas sobre nossos direitos autorais e marca registrada; sua conta, licença e acesso ao site, entre outros tópicos.

Conceitos básicos

Os trechos do código a seguir ajudarão você a começar a usar o SDK do Amazon WorkDocs.

Note

Para maior segurança, crie usuários federados em vez de usuários do IAM sempre que possível.

Exemplos

- [Conecte-se ao Amazon WorkDocs com credenciais de usuário do IAM e as consultas para os usuários](#)
- [Conectando-se ao Amazon WorkDocs assumindo uma função](#)
- [Fazer upload de um documento](#)
- [Fazer download de um documento](#)
- [Configuração de notificações para um usuário ou um perfil do IAM](#)
- [Criação de um usuário](#)
- [Conceder permissões aos usuários para um recurso](#)

Conecte-se ao Amazon WorkDocs com credenciais de usuário do IAM e as consultas para os usuários

O código a seguir mostra como usar as credenciais de API de um usuário do IAM para fazer chamadas de API. Nesse caso, o usuário da API e o site do Amazon WorkDocs pertencem à mesma conta da AWS.

Note

Para maior segurança, crie usuários federados em vez de usuários do IAM sempre que possível.

Confirme se o usuário do IAM recebeu acesso à API do Amazon WorkDocs por meio de uma política apropriada do IAM.

O exemplo do código usa a API [DescribeUsers](#) para pesquisar usuários e coletar metadados para eles. Os metadados dos usuários fornecem detalhes como o primeiro nome, sobrenome, ID do usuário e ID da pasta raiz. O ID da pasta raiz é especialmente útil se você deseja executar qualquer operação de upload ou download de conteúdo em nome do usuário.

O código exige que você tenha um ID da organização do Amazon WorkDocs.

Siga estas etapas para obter um ID de organização do Amazon WorkDocs no Console da AWS:

Como obter um ID de organização

1. No painel de navegação do console do [AWS Directory Service](#), escolha Directories (Diretórios).
2. Observe o valor do ID do diretório que corresponde ao seu site do Amazon WorkDocs. Essa é a ID da organização do site.

O exemplo a seguir mostra como usar credenciais do IAM para fazer chamadas de 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());
}
}
```

Conectando-se ao Amazon WorkDocs assumindo uma função

Este exemplo usa o AWS SDK para Java para assumir uma função e usar as credenciais de segurança temporárias da função para acessar o Amazon WorkDocs. A amostra de código usa a API [DescribeFolderContents](#) para listar os itens na pasta do usuário.

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

Fazer upload de um documento

Use o procedimento a seguir para carregar um documento no Amazon WorkDocs.

Como fazer upload de um documento

1. Crie uma instância de `AmazonWorkDocsClient` conforme este exemplo:

Se você usar credenciais de usuário do IAM, consulte [Conecte-se ao Amazon WorkDocs com credenciais de usuário do IAM e as consultas para os usuários](#). Se você assumir um perfil do IAM, consulte [Conectando-se ao Amazon WorkDocs assumindo uma função](#) para obter mais informações.

Note

Para maior segurança, crie usuários federados em vez de usuários do IAM sempre que possível.

```
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. Obtenha o signed URL do upload conforme este exemplo:

```
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. Faça upload do documento usando o signed URL conforme este exemplo:

```
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. Conclua o processo do upload ao alterar o status do documento para ACTIVE conforme este exemplo:

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

Fazer download de um documento

Para fazer download de um documento a partir do Amazon WorkDocs, obtenha um URL do download conforme o exemplo abaixo e use as ações de API fornecidas pela plataforma de desenvolvimento para fazer download do arquivo usando o 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());
```

Configuração de notificações para um usuário ou um perfil do IAM

Para criar e gerenciar notificações no Amazon WorkDocs, os administradores usam os consoles IAM e Amazon WorkDocs. Você usa o console do IAM para definir as permissões do usuário e usa o console do Amazon WorkDocs para ativar as notificações. Depois de ativar as notificações, você as assina. Siga estas etapas.

Note

Para maior segurança, crie usuários federados em vez de usuários do IAM sempre que possível.

Para definir as permissões de um usuário IAM

- Use o console do IAM para definir as seguintes permissões para o usuário:

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

Como ativar notificações

A ativação das notificações permite que você chame [CreateNotificationSubscription](#) depois de assinar as notificações.

1. Abra o console do Amazon WorkDocs em <https://console.aws.amazon.com/zocalo/>.

2. Na página Manage Your WorkDocs Sites (Gerenciar os sites do WorkDocs), selecione o diretório desejado e escolha Actions (Ações) e Manage Notifications (Gerenciar notificações).
3. Na página Manage Notifications (Gerenciar notificações), selecione Enable Notifications (Ativar notificações).
4. Insira o ARN do usuário ou da função que terá permissão para receber notificações do seu site do Amazon WorkDocs.

Para obter informações sobre como habilitar o Amazon WorkDocs a usar notificações, consulte [Como usar a API do Amazon WorkDocs com o AWS SDK para Python e o AWS Lambda](#). Depois de ativar as notificações, você e seu usuário podem se inscrever nelas.

Como solicitar as notificações do WorkDocs

1. Prepare o endpoint para processar mensagens do Amazon SNS. Para obter mais informações, consulte [Fanout to HTTP/S endpoints](#) no Guia do desenvolvedor do Amazon Simple Notification Service.

Important

O SNS envia uma mensagem de confirmação ao seu endpoint configurado. Você deve confirmar essa mensagem para receber notificações. Além disso, se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar a AWS por meio de uma interface de linha de comando ou uma API, use um endpoint do FIPS. Para ter mais informações sobre endpoints do FIPS, consulte [Federal Information Processing Standard \(FIPS\) 140-2](#).

2. Faça o seguinte:
 - Obter um ID de organização
 1. No painel de navegação do [console do AWS Directory Service](#), escolha Diretórios.
 2. O ID de diretório correspondente ao seu site do Amazon WorkDocs também serve como ID de organização para esse site.
 - Crie a solicitação de recebimento conforme este exemplo:

```
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());
```

Notificações do SNS

A mensagem inclui as seguintes informações:

- `organizationId` — O ID da organização.
- `parentEntityType` — O tipo do pai (Document | DocumentVersion | Folder).
- `parentEntityId` — O ID do pai.
- `entityType` — O tipo da identidade (Document | DocumentVersion | Folder).
- `entityId` — O ID da entidade.
- `action` — A ação, que pode ter um dos seguintes valores:
 - `delete_document`
 - `move_document`
 - `recycle_document`
 - `rename_document`
 - `revoke_share_document`
 - `share_document`
 - `upload_document_version`

Como desativar notificações

1. Abra o console do Amazon WorkDocs em <https://console.aws.amazon.com/zocalo/>.
2. Na página Manage Your WorkDocs Sites (Gerenciar os sites do WorkDocs), selecione o diretório desejado e escolha Actions (Ações) e Manage Notifications (Gerenciar notificações).
3. Na página Manage Notifications (Gerenciar notificações) selecione o ARN para o qual você deseja desativar as notificações e clique em Disable Notifications (Desativar notificações).

Criação de um usuário

O exemplo a seguir mostra como criar um usuário no Amazon WorkDocs.

Note

Essa não é uma operação válida para uma configuração do Connected AD. Para criar um usuário na configuração do Connected AD, o usuário já deve estar presente no diretório corporativo. Em seguida, você deve fazer uma chamada à API [ActivateUser](#) para ativar o usuário no Amazon WorkDocs.

O exemplo a seguir demonstra como criar um usuário com uma cota de armazenamento 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$$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);
```

Siga estas etapas para obter um ID de organização do Amazon WorkDocs no Console da AWS:

Como obter um ID de organização

1. No painel de navegação do console do [AWS Directory Service](#), escolha Directories (Diretórios).
2. Observe o valor do ID do diretório que corresponde ao seu site do Amazon WorkDocs. Essa é a ID da organização do site.

Conceder permissões aos usuários para um recurso

O exemplo a seguir mostra como usar a API [AddResourcePermissions](#) para conceder permissões CONTRIBUTOR a um USER em um recurso. Você pode usar a API para conceder permissões para que um usuário ou grupo tenha acesso a uma pasta ou 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);
```

Controle de acesso e autenticação de aplicativos administrativos

As APIs administrativas do Amazon WorkDocs são autenticadas e autorizadas por meio de políticas do IAM. Os administradores do IAM podem criar uma política do IAM e anexá-la a um perfil do IAM ou usuário que possa ser usado pelo desenvolvedor para acessar a API.

Confira os exemplos a seguir:

Tarefas

- [Conceder permissões aos desenvolvedores para a API do Amazon WorkDocs](#)
- [Conceder permissão a desenvolvedores terceirizados para as APIs do Amazon WorkDocs](#)
- [Conceder permissão aos usuários para assumirem um perfil do IAM](#)
- [Restringir o acesso a uma instancia do Amazon WorkDocs específica](#)

Conceder permissões aos desenvolvedores para a API do Amazon WorkDocs

Note

Para maior segurança, crie usuários federados em vez de usuários do IAM sempre que possível.

Se você for um administrador do IAM, poderá conceder a um usuário do IAM acesso à API do Amazon WorkDocs na mesma conta da AWS. Para fazer isso, crie uma política de permissões da API do Amazon WorkDocs e anexe-a ao usuário do IAM. A política de API a seguir concede permissão somente de leitura às várias APIs Describe.

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

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

Conceder permissão a desenvolvedores terceirizados para as APIs do Amazon WorkDocs

Você pode conceder acesso a desenvolvedores de terceiros ou a usuários que estão usando uma conta diferente da AWS. Para fazer isso, crie um perfil do IAM e anexe políticas de permissões da API do Amazon WorkDocs.

Essa forma de acesso é necessária nos seguintes cenários:

- O desenvolvedor pertence à mesma organização, mas a conta da AWS do desenvolvedor é diferente da conta do Amazon WorkDocs da AWS.
- Quando uma empresa deseja conceder acesso à API do Amazon WorkDocs a desenvolvedores de aplicativos de terceiros.

Nesses dois cenários, há duas contas da AWS envolvidas: uma conta da AWS do desenvolvedor e uma conta diferente que hospeda um site do Amazon WorkDocs.

O desenvolvedor precisará fornecer as seguintes informações para que o administrador da conta possa criar o perfil do IAM:

- O ID da sua conta da AWS
- Um `External ID` exclusivo que o cliente usará para identificar você. Para obter mais informações, consulte [Como usar um ID externo ao conceder de acesso aos seus recursos da AWS a terceiros](#).
- Uma lista das APIs do Amazon WorkDocs às quais seu aplicativo precisa acessar. O controle de políticas do IAM fornece controle granular, a capacidade de definir políticas de permissão

ou negação no nível individual da API. Para ver a lista de APIs do Amazon WorkDocs, consulte [Referência da API do Amazon WorkDocs](#).

O procedimento a seguir descreve as etapas necessárias para configurar o IAM para acesso entre contas.

Como configurar o IAM para acesso entre contas

1. Crie uma política de permissão da API Amazon WorkDocs e a nomeie como política `WorkDocsAPIReadOnly`.
2. Crie uma nova função no console do IAM da conta da AWS que hospeda o site do Amazon WorkDocs:
 - a. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
 - b. No painel de navegação do console, clique em Roles (Funções) e depois em Create new role (Criar função).
 - c. No campo Role name (Nome da função), digite um nome de função para ajudar você a identificar a finalidade dela, por exemplo, `workdocs_app_role`. Os nomes de função devem ser exclusivos em sua conta da AWS. Depois de inserir o nome, clique em Next Step (Próxima etapa).
 - d. Na página Select Role Type (Selecionar tipo de função), selecione a seção Role for Cross-Account Access (Função para acesso entre contas) e, em seguida, escolha o tipo de função que você deseja criar:
 - Selecione Provide access between AWS accounts you own (Conceder acesso entre suas contas da) se você for o administrador da conta de usuário e da conta de recurso, ou ambas as contas pertencerem à mesma empresa. Também selecione essa opção quando os usuários, a função e o recurso a serem acessados estiverem na mesma conta.
 - Selecione Conceder acesso entre sua conta da AWS e uma conta da AWS de terceiros se você for o administrador da conta que possui o site do Amazon WorkDocs e desejar conceder permissões para usuários de uma conta de desenvolvedor do aplicativo. Nessa opção é necessário especificar um ID externo (que deve ser fornecido pelo terceiro) para que haja um controle adicional sobre as circunstâncias em que o terceiro pode usar a função para acessar seus recursos. Para obter mais informações, acesse [Como usar um ID externo ao conceder acesso a seus recursos da AWS a terceiros](#).

- e. Na página seguinte, especifique o ID da conta da AWS ao qual você deseja conceder acesso aos recursos e insira também o External ID (ID externo) no caso de acesso de terceiros.
 - f. Clique em Next Step (Próxima etapa) para anexar uma política.
3. Na página Vincular política, pesquise a política de permissão da API Amazon WorkDocs criada anteriormente e selecione a caixa ao lado dela. Em seguida, clique em Próxima etapa.
 4. Analise os detalhes, copie o ARN da função para usar como referência no futuro e clique em Create Role (Criar função) para concluir a criação.
 5. Compartilhe o ARN da função com o desenvolvedor. Este é um exemplo do ARN da função:

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

Conceder permissão aos usuários para assumirem um perfil do IAM

Um desenvolvedor com uma conta da AWS administrativa pode permitir que um usuário assuma um perfil do IAM. Para fazer isso, você cria uma nova política e a anexa a esse usuário.

Uma política deve incluir uma instrução com o efeito Allow sobre a ação `sts:AssumeRole` e o nome de recurso da Amazon (ARN) da função em um elemento `Resource`, conforme mostrado no exemplo a seguir. Os usuários que obtêm a política, por meio da associação do grupo ou diretamente anexada, podem alternar para a função especificada.

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

Restringir o acesso a uma instancia do Amazon WorkDocs específica

Se você tem vários sites do Amazon WorkDocs em uma conta da AWS deseja conceder acesso à API a um site específico, defina um elemento Condition. O elemento Condition permite que você especifique as condições para quando uma política está em vigor.

Este é um exemplo de um elemento de condição:

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

Com a condição acima em vigor em uma política, os usuários podem acessar somente a instância do Amazon WorkDocs com o ID da d-123456789c5. Às vezes, o ID da instância do Amazon WorkDocs é conhecido como ID da organização ou ID do diretório. Para obter mais informações, consulte [Restringir o acesso a uma instancia do Amazon WorkDocs específica](#).

Siga estas etapas para obter um ID de organização do Amazon WorkDocs no Console da AWS:

Como obter um ID de organização

1. No painel de navegação do console do [AWS Directory Service](#), escolha Directories (Diretórios).
2. Observe o valor do ID do diretório que corresponde ao seu site do Amazon WorkDocs. Essa é a ID da organização do site.

Controle de acesso e autenticação de aplicativos de usuário

Os aplicativos no nível do usuário do Amazon WorkDocs são registrados e gerenciados no console do Amazon WorkDocs. Os desenvolvedores devem registrar os aplicativos na página My Applications do console do Amazon WorkDocs. Essa página fornece IDs exclusivos para cada aplicativo. Durante o registro, os desenvolvedores devem especificar os URIs de redirecionamento nos quais eles receberão tokens de acesso e escopos de aplicativo.

Atualmente, os aplicativos só podem acessar os sites do Amazon WorkDocs que estão na mesma conta da AWS em que eles estão registrados.

Índice

- [Conceder permissões para chamar as APIs do Amazon WorkDocs](#)
- [Usando IDs de pasta em chamadas de API](#)
- [Criar uma aplicação](#)
- [Escopos do aplicativo](#)
- [Autorização](#)
- [Invocando as APIs do Amazon WorkDocs](#)

Conceder permissões para chamar as APIs do Amazon WorkDocs

Os usuários da interface de linha de comando devem ter permissões completas para o Amazon WorkDocs e o AWS Directory Service. Sem as permissões, todas as chamadas de API retornam mensagens UnauthorizedResourceAccessException. A política a seguir concede permissões totais.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "workdocs:*",
        "ds:*",
        "ec2:CreateVpc",
        "ec2:CreateSubnet",
        "ec2:CreateNetworkInterface",
        "ec2:CreateTags",
        "ec2:CreateSecurityGroup",
```



```

        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeAvailabilityZones",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2>DeleteSecurityGroup",
        "ec2>DeleteNetworkInterface",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress"
    ],
    "Effect": "Allow",
    "Resource": "*"
}
]
}

```

Se você deseja conceder permissões de acesso somente para leitura, use esta política.

```

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

```

Na política, a primeira ação concede acesso a todas as operações do Amazon WorkDocs Describe. A ação DescribeDirectories obtém informações sobre seus diretórios AWS Directory Service. As operações do Amazon EC2 permitem que o Amazon WorkDocs obtenha uma lista de suas VPCs e sub-redes.

Usando IDs de pasta em chamadas de API

Sempre que uma chamada de API acessa uma pasta, você deve usar o ID da pasta, não o nome da pasta. Por exemplo, se você receber aprovação de `client.get_folder(FolderId='MyDocs')`, a chamada da API retornará uma mensagem `UnauthorizedResourceAccessException` e a seguinte mensagem 404.

```
client.get_folder(FolderId='MyDocs')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "C:\Users\user-name\AppData\Local\Programs\Python\Python36-32\lib\site-packages\botocore\client.py", line 253, in _api_call
    return self._make_api_call(operation_name, kwargs)
  File "C:\Users\user-name\AppData\Local\Programs\Python\Python36-32\lib\site-packages\botocore\client.py", line 557, in _make_api_call
    raise error_class(parsed_response, operation_name)
botocore.errorfactory.UnauthorizedResourceAccessException: An error occurred
(UnauthorizedResourceAccessException) when calling the GetFolder operation:
Principal [arn:aws:iam::395162986870:user/Aman] is not allowed to execute
[workdocs:GetFolder] on the resource.
```

Para evitar isso, use o ID no URL da pasta.

`site.workdocs/index.html#/folder/abc123def456ghi789jkl1789mno4be7024df198736472dd50ca970eb22796082e3d489577`.

Passar esse ID retorna um resultado correto.

```
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'}}
```

Criar uma aplicação

Como um administrador do Amazon WorkDocs, crie seu aplicativo usando as etapas a seguir.

Para criar um aplicativo.

1. Abra o console do Amazon WorkDocs em <https://console.aws.amazon.com/zocalo/>.
2. Selecione My Applications (Meus aplicativos), Create an Application (Criar um aplicativo).
3. Insira os seguintes valores:

Application Name (Nome do aplicativo)

Nome do aplicativo.

E-mail

Endereço de e-mail para associar ao aplicativo.

Application Description (Descrição do aplicativo)

Descrição do aplicativo.

Redirect URIs (Redirecionar URIs)

O local para o qual você deseja que o Amazon WorkDocs redirecione o tráfego.

Escopos do aplicativo

O escopo, de leitura ou gravação, que você deseja que o aplicativo tenha. Para obter mais detalhes, consulte [Escopos do aplicativo](#).

4. Escolha Create (Criar).

Escopos do aplicativo

O Amazon WorkDocs oferece suporte a estes escopos do aplicativo:

- Leitura de conteúdo (`workdocs.content.read`), que fornece ao aplicativo acesso a estas APIs Amazon WorkDocs:
 - Comece*

- `Describe*`
- Gravação de conteúdo (`workdocs.content.write`), que fornece ao aplicativo acesso a estas APIs Amazon WorkDocs:
 - `Criar*`
 - `Atualização*`
 - `Excluir*`
 - `Initiate*`
 - `Abort*`
 - `Adicionar*`
 - `Remover*`

Autorização

Após o registro do aplicativo ser concluído, um aplicativo pode solicitar autorização em nome de qualquer usuário do Amazon WorkDocs. Para isso, o aplicativo deve acessar o endpoint OAuth do Amazon WorkDocs, `https://auth.amazonworkdocs.com/oauth` e fornecer os seguintes parâmetros de consulta:

- [Obrigatório] `app_id`—ID de aplicativo gerado quando um aplicativo é registrado.
- [Obrigatório] `auth_type`—O tipo de OAuth da solicitação. `ImplicitGrant` é o valor compatível.
- [Obrigatório] `redirect_uri`—O URI de redirecionamento registrado de um aplicativo para receber um token de acesso.
- [Opcional] `scopes`—Uma lista de escopos separada por vírgula. Se não houver especificação, a lista de escopos selecionada durante o registro será usada.
- [Opcional] `state`—Uma string que é retornada junto com um token de acesso.

Note

Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar a AWS por meio de uma interface de linha de comando ou uma API, use um endpoint do FIPS. Para ter mais informações sobre endpoints do FIPS, consulte [Federal Information Processing Standard \(FIPS\) 140-2](#).

Uma solicitação GET de exemplo para iniciar o fluxo OAuth para obter um token de acesso:

```
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
```

O exemplo a seguir é usado durante o fluxo de autorização OAuth:

1. O usuário do aplicativo é solicitado a inserir o nome do site do Amazon WorkDocs.
2. O usuário é redirecionado para a página de autenticação do Amazon WorkDocs para inserir suas credenciais.
3. Após a autenticação ser bem-sucedida, o usuário visualizará a tela de consentimento na qual ele pode conceder ou negar a seu aplicativo a autorização para acessar o Amazon WorkDocs.
4. Após o usuário escolher Accept na tela de consentimento, o navegador dele será redirecionado à URL de retorno de chamada de seu aplicativo junto com o token de acesso e as informações da região como parâmetros de consulta.

Um exemplo de solicitação GET do Amazon WorkDocs:


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

Além do token de acesso, o serviço OAuth do Amazon WorkDocs também retorna `region` como um parâmetro de consulta para o site do Amazon WorkDocs específico. Os aplicativos externos devem usar o parâmetro `region` para determinar o endpoint de serviço do Amazon WorkDocs.

Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar a AWS por meio de uma interface de linha de comando ou uma API, use um endpoint do FIPS. Para ter mais informações sobre endpoints do FIPS, consulte [Federal Information Processing Standard \(FIPS\) 140-2](#).

Invocando as APIs do Amazon WorkDocs

Após obter o token de acesso, o aplicativo poderá fazer chamadas de API para os serviços do Amazon WorkDocs.

 Important

Este exemplo mostra como usar uma solicitação GET curl para obter os metadados de um documento.

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

Uma função JavaScript de exemplo para descrever as pastas raiz de um usuário:

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

O quadro abaixo descreve uma invocação de exemplo de API baseada em Java:

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

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

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

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

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

Gerenciador de conteúdo do Amazon WorkDocs

O Amazon WorkDocs Content Manager é uma ferramenta do utilitário de alto nível que faz upload ou download do conteúdo de um site do Amazon WorkDocs.

Tópicos

- [Construindo o Amazon WorkDocs Content Manager](#)
- [Fazer download de um documento](#)
- [Fazer upload de um documento](#)

Construindo o Amazon WorkDocs Content Manager

Você pode usar o Amazon WorkDocs Content Manager pode ser usado para aplicativos administrativos e de usuário.

Para aplicativos de usuário, um desenvolvedor deve criar o Amazon WorkDocs Content Manager com credenciais anônimas da AWS e um token de autenticação.

Em aplicativos administrativos, o cliente do Amazon WorkDocs deve ser inicializado com credenciais AWS Identity and Access Management (IAM). Além disso, o token de autenticação deve ser omitido em chamadas de API subsequentes.

O código a seguir demonstra como inicializar o Amazon WorkDocs Content Manager para aplicativos de usuário usando Java ou 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);
```

Fazer download de um documento

Os desenvolvedores podem usar o Amazon WorkDocs Content Manager para fazer download de uma versão específica ou da versão mais recente de um documento no Amazon WorkDocs. Veja os exemplos de como fazer download de uma versão específica de um documento usando Java e C#.

Note

Para fazer download da versão mais recente de um documento, não especifique o `VersionId` ao criar a solicitação `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();
```

Fazer upload de um documento

O Amazon WorkDocs Content Manager fornece uma API para fazer upload de conteúdo para um site do Amazon WorkDocs. Os exemplos a seguir demonstram como fazer upload de um documento usando Java e C#.

Java

```
File file = new File("file-path");
InputStream stream = new FileInputStream(file);
UploadDocumentStreamRequest request = new UploadDocumentStreamRequest();
request.setParentFolderId("destination-folder-id");
request.setContentType("content-type");
request.setStream(stream);
request.setDocumentName("document-name");
contentManager.uploadDocumentStream(request);
```

C#

```
var stream = new FileStream("file-path", FileMode.Open);

UploadDocumentStreamRequest uploadDocumentStreamRequest = new
    UploadDocumentStreamRequest()
{
    ParentFolderId = "destination-id",
    DocumentName = "document-name",
    ContentType = "content-type",
    Stream = stream
};

workDocsContentManager.UploadDocumentStreamAsync(uploadDocumentStreamRequest).Wait();
```