Guia do Desenvolvedor

Amazon WorkDocs

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

Amazon WorkDocs: Guia do Desenvolvedor

Copyright © 2025 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 desprestigie 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 a Amazon WorkDocs? 1
Acessando WorkDocs 1
Preços 1
Recursos 1
Conceitos básicos 3
Connect to WorkDocs com as credenciais de usuário do IAM 3
WorkDocs Conectando-se assumindo uma função 5
Fazer upload de um documento 8
Fazer download de um documento 10
Configuração de notificações 10
Criação de um usuário 13
Conceder permissões aos usuários para um recurso 14
Controle de acesso e autenticação de aplicativos administrativos 16
Conceder permissões aos desenvolvedores para a API WorkDocs 16
Conceder permissão a desenvolvedores terceirizados para o WorkDocs APIs 17
Conceder permissão aos usuários para assumirem um perfil do IAM 19
Restringindo o acesso a uma instância específica WorkDocs 19
Controle de acesso e autenticação de aplicativos de usuário 21
Concedendo permissões para chamar o WorkDocs APIs 21
Usando a pasta IDs em chamadas de API 23
Criar uma aplicação do 24
Escopos do aplicativo 24
Autorização 25
Invocando WorkDocs APIs 26
WorkDocs Gerenciador de conteúdo 28
Construindo o Gerenciador de WorkDocs Conteúdo 28
Fazer download de um documento 29
Fazer upload de um documento 30

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 de 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 a Amazon WorkDocs?

A Amazon WorkDocs é um sistema de armazenamento, colaboração e compartilhamento de documentos. WorkDocs é totalmente gerenciado, seguro e em escala corporativa. 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 WorkDocs cliente podem ser usados para visualizar vários tipos diferentes de arquivos, dependendo do tipo de mídia da Internet do arquivo. WorkDocs suporta 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.

Acessando WorkDocs

Os usuários finais usam os aplicativos cliente acessar seus arquivos. Usuários não administrativos nunca precisam usar o WorkDocs console ou o painel de administração. 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.
- WorkDocs Unidade usada para sincronizar uma pasta na área de trabalho do Mac ou Windows com seus WorkDocs arquivos.

Preços

Com WorkDocs, não há taxas ou compromissos iniciais. 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 especializados e baseados em funções, além de laboratórios individualizados para ajudar a aprimorar suas AWS habilidades e ganhar experiência prática.
- AWS Centro do desenvolvedor — explore tutoriais, baixe ferramentas e saiba mais sobre eventos para AWS desenvolvedores.
- AWS Ferramentas do desenvolvedor — Links para ferramentas do desenvolvedor SDKs, kits de ferramentas do IDE e ferramentas de linha de comando para desenvolver e gerenciar AWS aplicativos.
- Centro de recursos de introdução — Saiba como configurar seu aplicativo Conta da AWS, participar da AWS comunidade e lançar seu primeiro aplicativo.
- Tutoriais práticos — Siga os tutoriais para iniciar seu step-by-step primeiro aplicativo no. AWS
- AWS Whitepapers — Links para uma lista abrangente de AWS white papers técnicos, abrangendo tópicos como arquitetura, segurança e economia e criados por arquitetos de AWS soluções ou outros especialistas técnicos.
- AWS Support Center — O hub para criar e gerenciar seus AWS Support casos. Também inclui links para outros recursos úteis, como fóruns, informações técnicas FAQs, status de integridade do serviço AWS Trusted Advisor e.
- Suporte— A principal página da web com informações sobre Suporte um one-on-one canal de suporte de resposta rápida para ajudá-lo a criar e 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.
- AWS Termos do site — Informações detalhadas sobre nossos direitos autorais e nossa marca registrada; sua conta, licença e acesso ao site; e outros tópicos.

Conceitos básicos

Os trechos de código a seguir podem ajudar você a começar a usar o WorkDocs SDK.

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

Exemplos

- Conecte-se WorkDocs com as credenciais de usuário do IAM e consulte os usuários
- WorkDocs Conectando-se assumindo uma função
- Fazer upload de um documento
- Fazer download de um documento
- Configuração de notificações
- Criação de um usuário
- Conceder permissões aos usuários para um recurso

Conecte-se WorkDocs com as credenciais de usuário do IAM e consulte 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 WorkDocs site pertencem à mesma AWS conta.

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

Certifique-se de que o usuário do IAM tenha recebido acesso à WorkDocs API por meio de uma política apropriada do IAM.

A amostra de código usa a DescribeUsersAPI para pesquisar usuários e obter metadados para os usuários. 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 obtenha uma ID WorkDocs da organização.

Siga estas etapas para obter uma ID da WorkDocs organização no AWS console:

Como obter um ID de organização

1. No painel de navegação do console do AWS Directory Service, escolha Diretórios.
2. Anote o valor do ID do diretório que corresponde ao seu WorkDocs site. 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());
    }
  }
}
```

WorkDocs Conectando-se assumindo uma função

Este exemplo usa o AWS Java SDK para assumir uma função e usar as credenciais de segurança temporárias da função para acessar. WorkDocs A amostra de código usa a DescribeFolderContentsAPI 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

> **Note**
>
> Você deve ser um desenvolvedor de software para concluir as etapas desta seção. Para obter informações sobre como usar WorkDocs para fazer upload de arquivos, consulte Carregar arquivos no Guia do WorkDocs usuário.

Use o procedimento a seguir para carregar um documento no 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 WorkDocs com as credenciais de usuário do IAM e consulte os usuários. Se você assumir um perfil do IAM, consulte WorkDocs Conectando-se 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);
HttpURLConnection connection = (HttpURLConnection) url.openConnection();
connection.setDoOutput(true);
connection.setRequestMethod("PUT");
// Content-Type supplied here should match with the Content-Type set
// in the InitiateDocumentVersionUpload request.
connection.setRequestProperty("Content-Type","application/octet-stream");
connection.setRequestProperty("x-amz-server-side-encryption", "AES256");
File file = new File("/path/to/file.txt");
FileInputStream fileInputStream = new FileInputStream(file);
OutputStream outputStream = connection.getOutputStream();
com.amazonaws.util.IOUtils.copy(fileInputStream, outputStream);
connection.getResponseCode();
```

4. 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

> **Note**
>
> Você deve ser um desenvolvedor de software para concluir as etapas desta seção. Para obter informações sobre como usar WorkDocs para baixar arquivos, consulte Como baixar arquivos no Guia WorkDocs do usuário.

Para baixar um documento de WorkDocs, obtenha uma URL para o download da seguinte forma e, em seguida, use as ações de API fornecidas pela sua plataforma de desenvolvimento para baixar o arquivo usando a 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

Você segue esse processo para configurar as notificações:

1. Defina permissões de usuário ou função do IAM para permitir que o chamador acesse o gerenciamento de assinaturas de APIs notificações.
2. Ligue para a assinatura de notificação APIs para ativar ou desativar a publicação de mensagens do SNS em seu endpoint.

> **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 ligue CreateNotificationSubscriptiondepois de assinar as notificações.

1. Abra o WorkDocs console em https://console.aws.amazon.com/zocalo/.
2. Na página Gerenciar seus WorkDocs sites, selecione o diretório desejado e escolha Ações e, em seguida, Gerenciar notificações.
3. Na página Manage Notifications (Gerenciar notificações), selecione Enable Notifications (Ativar notificações).
4. Insira o ARN do usuário ou função que você deseja permitir que receba notificações do seu WorkDocs site.

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

Para se inscrever para WorkDocs receber notificações

1. Prepare o endpoint para processar mensagens do Amazon SNS. Para obter mais informações, consulte Fanout para 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 disponíveis, 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 do diretório correspondente ao seu WorkDocs site da Amazon também serve como ID da organização desse 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 WorkDocs console em https://console.aws.amazon.com/zocalo/.
2. Na página Gerenciar seus WorkDocs sites, selecione o diretório desejado e escolha Ações e, em seguida, 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 em 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 para a ActivateUserAPI para ativar o usuário 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 uma ID da WorkDocs organização no AWS console:

Como obter um ID de organização

1. No painel de navegação do console do AWS Directory Service, escolha Diretórios.
2. Anote o valor do ID do diretório que corresponde ao seu WorkDocs site. Essa é a ID da organização do site.

Conceder permissões aos usuários para um recurso

O exemplo a seguir mostra como usar a AddResourcePermissionsAPI para conceder `CONTRIBUTOR` permissões 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

WorkDocs APIs os administrativos são autenticados e autorizados 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 WorkDocs
- Conceder permissão a desenvolvedores terceirizados para o WorkDocs APIs
- Conceder permissão aos usuários para assumirem um perfil do IAM
- Restringindo o acesso a uma instância específica WorkDocs

Conceder permissões aos desenvolvedores para a API WorkDocs

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

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

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

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

Conceder permissão a desenvolvedores terceirizados para o WorkDocs APIs

Você pode conceder acesso a desenvolvedores terceirizados ou a usuários que estão usando uma AWS conta diferente. Para fazer isso, crie uma função do IAM e anexe políticas de permissão WorkDocs da API.

Essa forma de acesso é necessária nos seguintes cenários:

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

Em ambos os cenários, há duas AWS contas envolvidas, uma AWS conta de desenvolvedor e uma conta diferente que hospeda um WorkDocs site.

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

- ID AWS da sua conta
- Um `External ID` exclusivo que o cliente usará para identificar você. Para obter mais informações, consulte Como usar uma ID externa ao conceder acesso aos seus AWS recursos a terceiros.
- Uma lista do WorkDocs APIs seu aplicativo precisa ser acessada. 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 WorkDocs APIs, consulte Referência WorkDocs da API.

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 de WorkDocs API, chame-a `WorkDocsAPIReadOnly` de política.
2. Crie uma nova função no console do IAM da AWS conta que hospeda o WorkDocs site:
   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 das funções devem ser exclusivos em sua AWS conta. 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 Fornecer acesso entre AWS contas que você possui se você for o administrador da conta do usuário e da conta do recurso, ou se 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 Fornecer acesso entre sua AWS conta e uma AWS conta de terceiros se você for o administrador da conta proprietária do WorkDocs site e quiser conceder permissões aos usuários a partir de uma conta de desenvolvedor de aplicativos. 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 próxima página, especifique o ID da AWS conta à qual você deseja conceder acesso aos seus recursos e também insira o ID externo no caso de acesso de terceiros.
   f. Clique em Next Step (Próxima etapa) para anexar uma política.

3. Na página Anexar política, pesquise a política de permissão da WorkDocs API que foi criada anteriormente, selecione a caixa ao lado da política e 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 AWS conta administrativa pode permitir que um usuário assuma uma função 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"
  }
}
```

Restringindo o acesso a uma instância específica WorkDocs

Se você tiver vários WorkDocs sites em uma AWS conta e quiser conceder acesso à API a um site específico, poderá definir um `Condition` elemento. 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 só podem acessar a WorkDocs instância com o ID `ded-123456789c5`. WorkDocs Às vezes, o ID da instância é chamado de ID da organização ou ID do diretório. Para obter mais informações, consulte Restringindo o acesso a uma instância específica WorkDocs .

Siga estas etapas para obter uma ID da WorkDocs organização no AWS console:

Como obter um ID de organização

1. No painel de navegação do console do AWS Directory Service, escolha Diretórios.
2. Anote o valor do ID do diretório que corresponde ao seu WorkDocs site. Essa é a ID da organização do site.

Controle de acesso e autenticação de aplicativos de usuário

WorkDocs os aplicativos em nível de usuário são registrados e gerenciados por meio do WorkDocs console. Os desenvolvedores devem registrar seus aplicativos na `My Applications` página do WorkDocs console, que fornecerá informações exclusivas IDs para cada aplicativo. Durante o registro, os desenvolvedores devem especificar o redirecionamento para URIs onde receberão os tokens de acesso, bem como os escopos do aplicativo.

Atualmente, os aplicativos só podem acessar WorkDocs sites dentro da mesma AWS conta em que estão registrados.

Conteúdo

- Concedendo permissões para chamar o WorkDocs APIs
- Usando a pasta IDs em chamadas de API
- Criar uma aplicação do
- Escopos do aplicativo
- Autorização
- Invocando WorkDocs APIs

Concedendo permissões para chamar o WorkDocs APIs

Os usuários da interface de linha de comando devem ter permissões completas para WorkDocs AWS Directory Service e. Sem as permissões, todas as chamadas de API retornam UnauthorizedResourceAccessException mensagens. 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 WorkDocs `Describe` operações. A `DescribeDirectories` ação obtém informações sobre seus AWS Directory Service diretórios. As EC2 operações da Amazon permitem WorkDocs obter uma lista de suas VPCs e de suas sub-redes.

Usando a pasta IDs 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 for aprovado`client.get_folder(FolderId='MyDocs')`, a chamada da API retornará uma UnauthorizedResourceAccessExceptionmensagem 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/abc123def456ghi789jkl789mno4be7024df198736472dd50ca970eb22796082e3d489577`.

Passar esse ID retorna um resultado correto.

```
client.get_folder(FolderId='abc123def456ghi789jkl789mno4be7024df198736472dd50ca970eb22796082e3d
{'ResponseMetadata': {'RequestId': 'f8341d4e-4047-11e7-9e70-afa8d465756c',
 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': 'f234564e-1234-56e7-89e7-
a10fa45t789c', 'cache-control': 'private, no-cache, no-store, max-age=0',
 'content-type': 'application/json', 'content-length': '733', 'date':
 'Wed, 24 May 2017 06:12:30 GMT'}, 'RetryAttempts': 0}, 'Metadata': {'Id':
 'abc123def456ghi789jkl789mno4be7024df198736472dd50ca970eb22796082e3d489577', 'Name':
 'sentences', 'CreatorId':
 'S-1-5-21-2125721135-1643952666-3011040551-2105&d-906724f1ce', 'ParentFolderId':
 '0a811a922403ae8e1d3c180f4975f38f94372c3d6a2656c50851c7fb76677363',
 'CreatedTimestamp': datetime.datetime(2017, 5, 23, 12, 59, 13, 8000,
 tzinfo=tzlocal()), 'ModifiedTimestamp': datetime.datetime(2017, 5, 23, 13,
 13, 9, 565000, tzinfo=tzlocal()), 'ResourceState': 'ACTIVE', 'Signature':
 'b7f54963d60ae1d6b9ded476f5d20511'}}
```

Criar uma aplicação do

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

Para criar uma aplicação

1. Abra o WorkDocs console 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.

   Redirecionar URIs

   O local para o qual você WorkDocs deseja redirecionar o tráfego.

   Escopos do aplicativo

   O escopo, de leitura ou gravação, que você deseja que o aplicativo tenha. Consulte mais detalhes em Escopos do aplicativo.
4. Escolha Criar.

Escopos do aplicativo

WorkDocs suporta os seguintes escopos de aplicativos:

- Content Read (`workdocs.content.read`), que dá ao seu aplicativo acesso ao seguinte WorkDocs APIs:
  - Get*

  - Describe*
- Content Write (`workdocs.content.write`), que dá ao seu aplicativo acesso ao seguinte WorkDocs APIs:
  - Create*
  - Update*
  - Delete*
  - Initiate*
  - Abort*
  - Adicionar*
  - Remover*

Autorização

Depois que o registro do aplicativo for concluído, um aplicativo poderá solicitar autorização em nome de qualquer WorkDocs usuário. Para fazer isso, o aplicativo deve visitar o WorkDocs OAuth endpoint e fornecer os seguintes parâmetros de consulta: `https://auth.amazonworkdocs.com/oauth`

- [Obrigatório] `app_id`—ID de aplicativo gerado quando um aplicativo é registrado.
- [Obrigatório] `auth_type` — O OAuth tipo 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 disponíveis, consulte Federal Information Processing Standard (FIPS) 140-2.

Um exemplo de solicitação GET para iniciar o OAuth fluxo e 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 seguinte ocorre durante o fluxo OAuth de autorização:

1. O usuário do aplicativo é solicitado a inserir o nome do WorkDocs site.
2. O usuário é redirecionado para a página de WorkDocs autenticação para inserir suas credenciais.
3. Após a autenticação ser bem-sucedia, o usuário visualizará a tela de consentimento na qual ele pode conceder ou negar a seu aplicativo a autorização para acessar o 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 de WorkDocs:

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

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

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 disponíveis, consulte Federal Information Processing Standard (FIPS) 140-2.

Invocando WorkDocs APIs

Após obter o token de acesso, o aplicativo poderá fazer chamadas de API para os serviços do 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"
```

Um exemplo de JavaScript função 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());
}
```

WorkDocs Gerenciador de conteúdo

WorkDocs O Content Manager é uma ferramenta utilitária de alto nível que carrega ou baixa conteúdo de um WorkDocs site.

Tópicos

- Construindo o Gerenciador de WorkDocs Conteúdo
- Fazer download de um documento
- Fazer upload de um documento

Construindo o Gerenciador de WorkDocs Conteúdo

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

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

Para aplicativos administrativos, o WorkDocs cliente 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 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(param);
```

Fazer download de um documento

Os desenvolvedores podem usar WorkDocs o Gerenciador de conteúdo para baixar uma versão específica ou a versão mais recente de um documento do 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

WorkDocs O Content Manager fornece uma API para carregar conteúdo em um WorkDocs site. 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();
```