PDF Accessibility Remediation UI - Installation Guide

This guide provides detailed instructions for deploying the PDF Accessibility Remediation UI in your environment. This UI works in conjunction with the backend PDF Accessibility project to provide a complete solution for PDF accessibility remediation.

Prerequisites

Before you begin, ensure you have the following:

1. **AWS Account** with appropriate permissions to create resources
2. **Node.js** (version 14.x or later)
3. **AWS CLI** installed and configured with appropriate credentials
4. **AWS CDK CLI** installed (`npm install -g aws-cdk`)
5. **Git** for version control
6. **Python 3.9** for Lambda functions
7. **GitHub Personal Access Token** with repository access permissions
8. **Backend Deployment** - The PDF_Accessibility (https://github.com/ASUCICREPO/PDF_Accessibility) project must be deployed first

Step 1: Deploy the Backend PDF Accessibility Project

Before deploying this UI, you must first deploy the backend PDF Accessibility project:

1. Clone the backend repository:

```
git clone https://github.com/ASUCICREPO/PDF_Accessibility.git
cd PDF_Accessibility
```

2. Follow the installation instructions in the backend repository to deploy it completely.

3. **Important**: Take note of the S3 bucket name created during the backend deployment. You will need this for the UI deployment.

Step 2: Clone the UI Repository

1. Clone this repository:

```
git clone https://github.com/ASUCICREPO/PDF_accessability_UI.git
cd PDF_accessability_UI
```

Step 3: Install Dependencies

1. Install backend dependencies:

```
cd cdk_backend
npm install
```

2. Install frontend dependencies:

```
cd ../pdf_ui
npm install
```

Step 4: Configure and Deploy the CDK Backend

1. Return to the cdk_backend directory:

```
cd ../cdk_backend
```

2. Modify the domain prefix in the CDK stack:

   Open the file `cdk_backend/lib/cdk_backend-stack.ts` and locate this line:

```
const domainPrefix = 'pdf-ui-auth'; // must be globally unique in that region
```

   Change `pdf-ui-auth` to a unique name of your choice. This domain prefix must be globally unique within the AWS region you're deploying to.

3. Bootstrap your AWS environment (if not already done):

```
cdk bootstrap -c githubToken=<your-github-token> -c bucketName=<s3-bucket-name-from-backend-deployment>
```

4. Deploy the infrastructure:

```
cdk deploy -c githubToken=<your-github-token> -c bucketName=<s3-bucket-name-from-backend-deployment>
```

   Replace:

   - `<your-github-token>` with your GitHub Personal Access Token
   - `<s3-bucket-name-from-backend-deployment>` with the S3 bucket name from the backend deployment

5. The deployment will create several AWS resources including:

   - Cognito User Pool and Identity Pool
   - Amplify application for hosting the UI
   - IAM roles and policies
   - Lambda functions for user management

6. Take note of the outputs from the CDK deployment, especially the Amplify app URL.

Step 5: Configure S3 Bucket CORS Settings

After deployment, you need to configure CORS settings for the S3 bucket to allow cross-origin requests from the UI:

1. Go to the AWS Management Console
2. Navigate to S3 service
3. Select the S3 bucket that was created during the backend deployment
4. Click on the "Permissions" tab
5. Scroll down to the "Cross-origin resource sharing (CORS)" section
6. Click "Edit" and add the following CORS configuration:

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "GET",
            "HEAD",
            "PUT",
            "POST",
            "DELETE"
        ],
        "AllowedOrigins": [
            "*"
        ],
        "ExposeHeaders": []
    }
]
```

7. Click "Save changes"

Step 6: Configure AWS Amplify Redirect and Rewrite Rules

After the Amplify app is deployed, you need to configure redirect and rewrite rules:

1. Go to the AWS Management Console
2. Navigate to AWS Amplify service
3. Select your newly deployed application
4. Go to "Hosting" → "Rewrites and redirects"
5. Delete any existing rules
6. Add the following rules:

```
[
  {
    "source": "/</^[^.]+$|\\.(?!(css|gif|ico|jpg|js|png|txt|svg|woff|woff2|ttf|map|json)$)([^.]+$)/>",
    "status": "301",
    "target": "/index.html"
  },
  {
    "source": "/home",
    "status": "200",
    "target": "/index.html"
  },
  {
    "source": "/callback",
    "status": "200",
    "target": "/index.html"
  },
  {
    "source": "/app",
    "status": "200",
    "target": "/index.html"
  }
]
```

Important Note: These rules need to be added AFTER the application is done building. If you need to deploy a new update, you should:

1. Delete these rules
2. Deploy the update
3. Re-add these rules after the build is complete

Step 7: Testing the Deployment

1. Access your application using the Amplify app URL (from the CDK deployment outputs)
2. You should see the login page for the PDF Accessibility Remediation UI
3. Create a new account or sign in with existing credentials
4. Upload a PDF file to test the accessibility remediation process

Troubleshooting

Authentication Issues

- If you encounter "No matching state found" errors:
  - Clear browser cookies and cache
  - Ensure correct Cognito configuration in environment variables
  - Check redirect URIs in Cognito user pool client settings

Upload Failures

- Check file size limits in user attributes
- Verify S3 bucket permissions
- Check CORS configuration on the S3 bucket

PDF Processing Errors

- Check CloudWatch logs for Lambda functions
- Verify Adobe API credentials
- Monitor ECS task status

Amplify Deployment Issues

- If the application doesn't load correctly, verify the redirect and rewrite rules
- Check the Amplify build logs for any errors
- Ensure the GitHub token has the necessary permissions

Additional Configuration

Custom Domain (Optional)

If you want to use a custom domain for your application:

1. Go to the AWS Amplify console
2. Select your application
3. Go to "Domain management"
4. Follow the instructions to add and verify your domain

User Management

By default, the application creates three user groups:

- DefaultUsers: Regular users with standard permissions
- AmazonUsers: Users with Amazon email addresses
- AdminUsers: Users with administrative privileges

To manage users and their permissions:

1. Go to the AWS Cognito console
2. Select the user pool created for this application
3. Navigate to "Users and groups" to manage users and their group memberships

Security Considerations

- The CORS configuration provided allows requests from any origin (`*`). For production environments, you should restrict this to specific domains.
- Review the IAM roles and permissions created by the CDK deployment to ensure they follow the principle of least privilege.
- Consider implementing additional security measures such as WAF rules for the Amplify application.

Maintenance and Updates

When updating the application:

1. Make your changes to the code
2. For frontend changes:

```
cd pdf_ui
npm run build
```

3. For backend changes, redeploy the CDK stack:

```
cd cdk_backend
cdk deploy -c githubToken=<your-github-token> -c bucketName=<s3-bucket-name>
```

4. Remember to reconfigure the Amplify redirect and rewrite rules after each deployment

Support

If you encounter any issues or have questions, please open an issue on the GitHub repository.