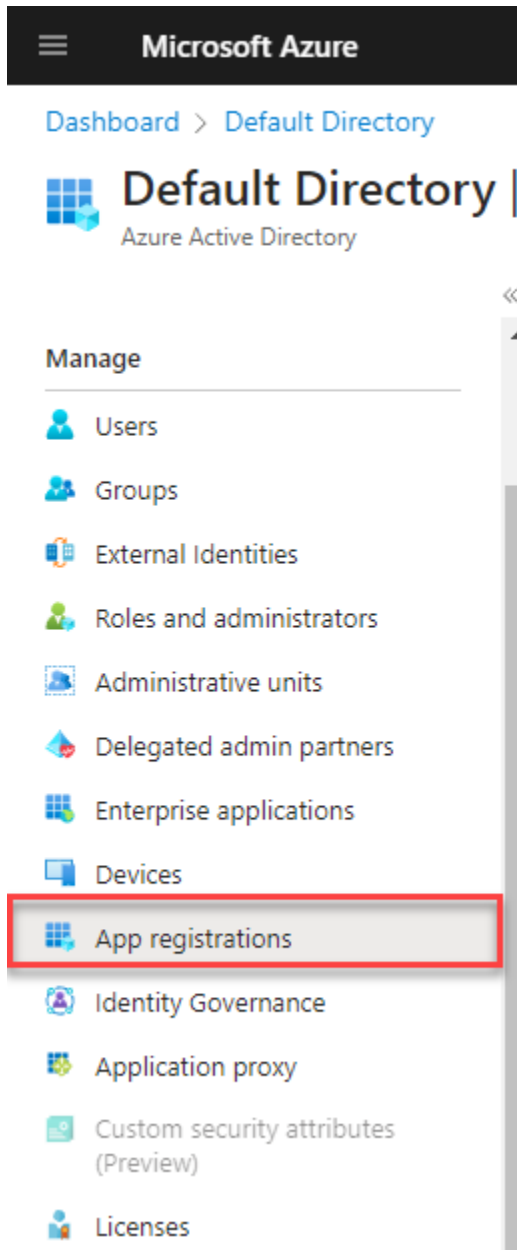


Document Converter Deployment Instructions

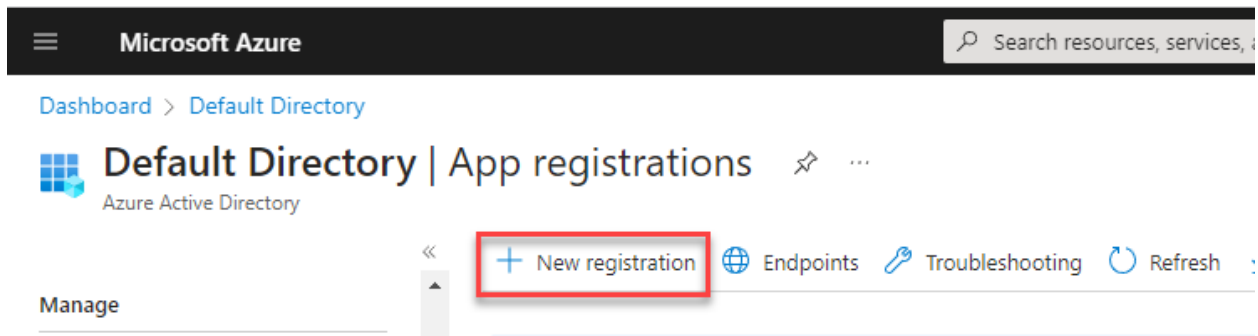
Azure configuration

* Create an App registration

1. Select App Registrations tab in Azure AD



2. Select New Registration



3. Type application name, select Accounts in this organization (single tenant), and select register

The screenshot shows the 'Register an application' page in the Microsoft Azure portal. The page has a heading 'Register an application' and a subheading 'The user-facing display name for this application (this can be changed later)'. Below this, there's a text input field for the application name, which contains 'DocumentToPDFFunction'. A red box and a red circle with the number 1 highlight this field. Below the name field, there's a section titled 'Supported account types' with the question 'Who can use this application or access this API?'. There are four radio button options: 'Accounts in this organizational directory only (Default Directory only - Single tenant)' (selected), 'Accounts in any organizational directory (Any Azure AD directory - Multitenant)', 'Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)', and 'Personal Microsoft accounts only'. A red box and a red circle with the number 2 highlight the first option. Below the account types section, there's a link 'Help me choose...'. Further down, there's a section titled 'Redirect URI (optional)' with a description: 'We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.' Below this, there's a dropdown menu for 'Select a platform' and a text input field for the redirect URI, which contains 'e.g. https://example.com/auth'. At the bottom of the page, there's a link 'Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from Enterprise applications.' and a link 'By proceeding, you agree to the Microsoft Platform Policies'. At the very bottom, there's a blue 'Register' button highlighted with a red box and a red circle with the number 3.

4. After Registration is created, select the new registration. Select Certificates and Secrets. Select the Federated Credentials Tab and then select add credential.

DocumentToPDFFunction | Certificates & secrets

Search

Got feedback?

Overview

Quickstart

Integration assistant

Manage

Branding & properties

Authentication

Certificates & secrets

Token configuration

API permissions

Expose an API

App roles

Owners

Roles and administrators

Manifest

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

Certificates (0)

Client secrets (0)

Federated credentials (0)

Allow other identities to impersonate this application by establishing a trust with an external OpenID Connect (OIDC) identity provider. This federation allows you to get tokens to access Azure AD protected resources that this application has access to like Azure and Microsoft graph. [Learn more](#)

+ Add credential

Name	Description	Subject Identifier
------	-------------	--------------------

No federated identity credentials have been added for this application.

5. Select GitHub Actions Deploying Azure resources

Microsoft Azure

Search resources, services, and docs (G+)

[Dashboard](#) > [Default Directory](#) | [App registrations](#) > [test](#) | [Certificates & secrets](#) >

Edit a credential ...

Configure an Azure AD managed identity or an identity from an external OpenID Connect Provider to get tokens as this application and access Azure resources.

Federated credential scenario *

GitHub Actions deploying Azure resources

Connect your GitHub account

Please enter the details of your GitHub Actions workflow that you want to connect with Azure Active Directory. These values will be used by Azure AD to validate the connection and should match your GitHub OIDC configuration. Issuer has a limit of 600 characters. Subject Identifier is a calculated field with a 600 character limit.

Issuer ⓘ

https://token.actions.githubusercontent.com

[Edit \(optional\)](#)

Organization *

Your Github Username

Repository *

Document-Converter-Azure-Function

Entity type *

Environment

Based on selection *

production

Subject identifier ⓘ

repo:zacaryfettig/Document-Converter-Azure-Function:environment:production

This value is generated based on the GitHub account details provided. [Edit \(optional\)](#)

Credential details

Provide a name and description for this credential and review other details.

Name ⓘ

Document-Converter-Azure-Function

Description ⓘ

Limit of 600 characters

Audience ⓘ

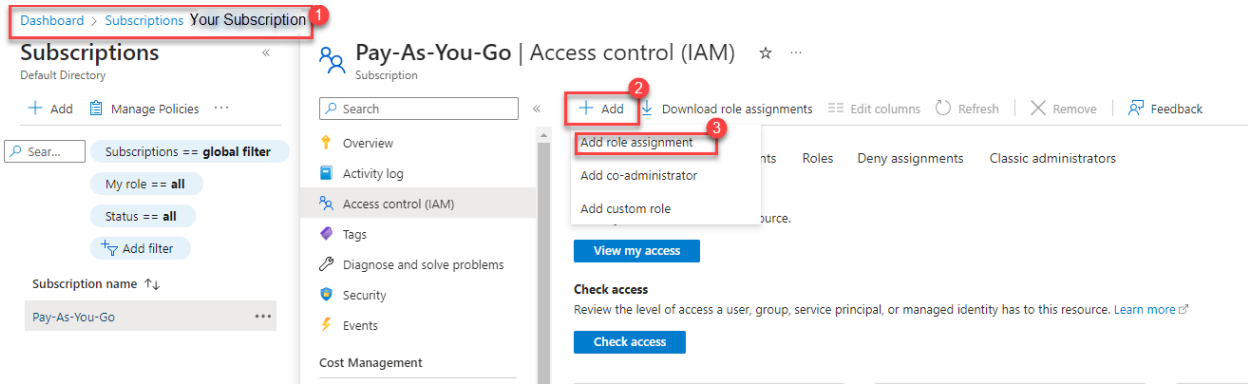
api://AzureADTokenExchange

[Edit \(optional\)](#)

Update

Cancel

6. Go to your subscription. Select Add. Add Role Assignment.



6. Select Assignment Type. Select Privileged administrator roles

Dashboard > Subscriptions > Pay-As-You-Go | Access control (IAM) >

Add role assignment

Assignment type Role Members Review + assign

Select the type of role to assign. [Learn more](#)

Assignment type

- ☐ Job function roles
Grant access to Azure resources based on job function, such as the ability to create virtual machines.
- ☒ Privileged administrator roles
Grant privileged administrator access, such as the ability to assign roles to other users.

⚠ Can a job function role with less access be used instead?

7. Select Contributor for the role



8. Under the Members tab select Assign access to user, group, or service principle. Select Members. Type name of app registration and select from the list. Select the select button and continue to next screen.

Add role assignment

Assignment type: Role **Members** Review + assign

Selected role: Contributor

Assign access to: ☒ User, group, or service principal ☐ Managed identity

Members: **+ Select members**

Name	Object ID	Type
No members selected		

Description: Optional

Review + assign Previous **Next**

Select members

Select document

DocumentToPDFFunction

Selected members:
No members selected. Search for and add one or more members you want to assign to the role for this resource.
[Learn more about RBAC](#)

Select Close

9. Review and Assign

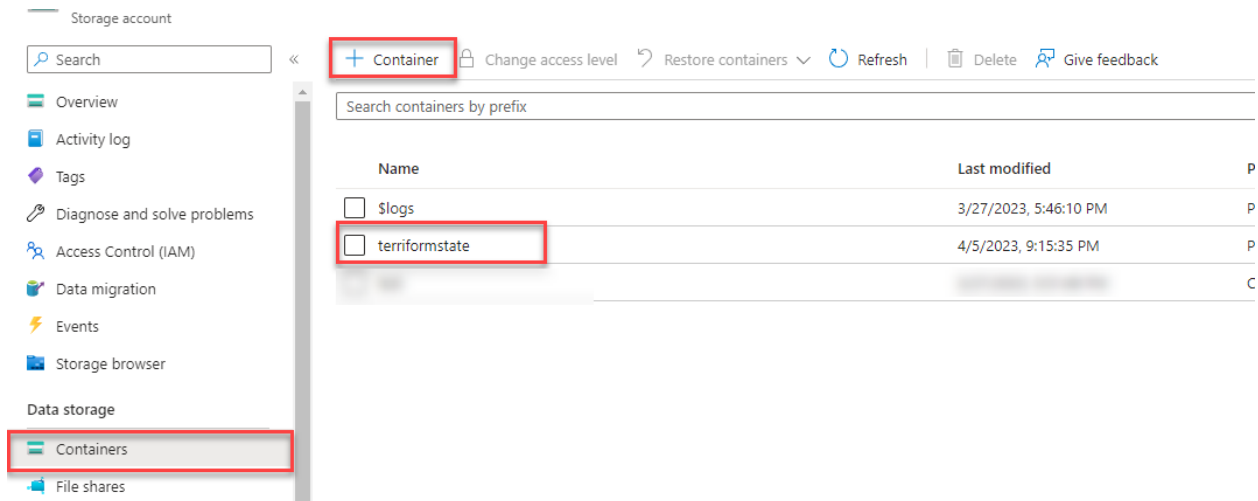
Add role assignment

Assignment type	Role	Members	Review + assign
Role	Contributor		
Scope	/subscriptions/		
Members			
	Name	Object ID	Type
	DocumentToPDFFunction	9ffb8cc8-9abe-427b-8350-8b85ecd6b77b	App
Description	No description		

Review + assign Previous Next

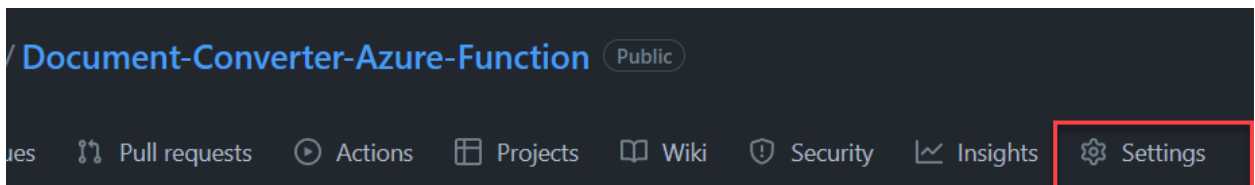
10. Create a resource group to store Terraform State files for deployment

11. Create a storage account to store the State Files. Create a container inside the Storage account used to store the Terraform State Files.

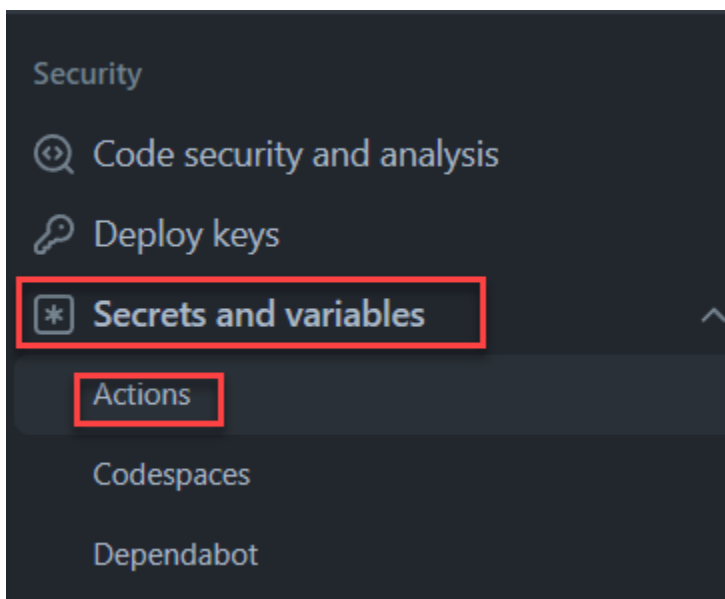


GitHub Configuration

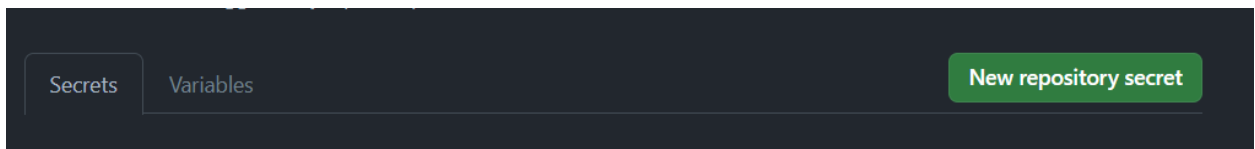
1. In the Document Converter Azure Function repository select Settings



2. Under the security section of settings select secrets and variables. Select Actions.



3. Select the secrets tab and New repository secret.



4. Create the secrets that will be used to identify your Azure account to enable Azure deployment

--Azure Client ID: Can be found on the overview page of the DocumentToPDFFunction App registration

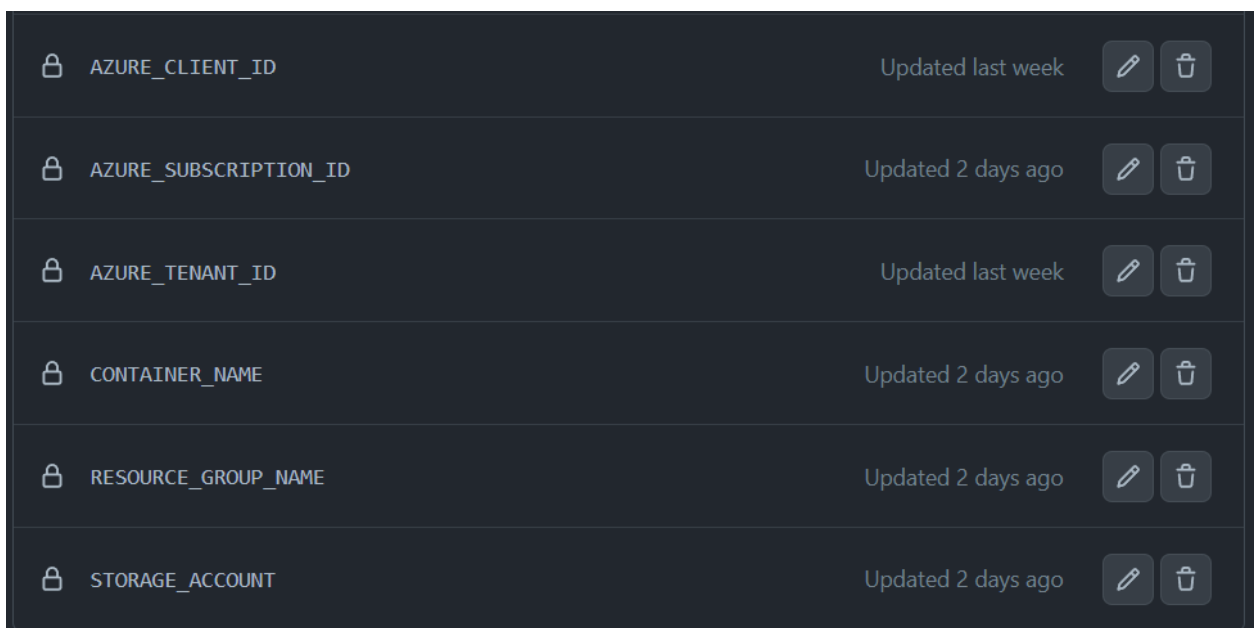
--Subscription ID: Found on the overview page of your subscription

--Tenant ID: Found on the overview page of the main Azure AD page

--Container Name: Name of the container inside of the storage account

--Resource Group: resource group used to hold the Terriform State Files

--Storage Account: Storage account name of the account used to store the Terriform State Files



Setup for Deployment is Finished

Run deployment through GitHub Actions

##PDF Program How To and accessing