

# Delegation of mailbox access on M365

## Table of Contents

[Delegate mailbox access on Microsoft 365 \(has to be repeated for every user whose mailbox content are to be migrated\)](#)

[Overview](#)

[OPTION 1: Delegate access one user at a time](#)

[OPTION 2: Delegate access for all or selected users](#)

[Enable IMAP access \(has to be repeated for every user whose mailbox content are to be migrated\)](#)

[Bulk Operations using the power shell](#)

[Delegating access for all](#)

[Removing delegation access](#)

## Delegate mailbox access on Microsoft 365 (has to be repeated for every user whose mailbox content are to be migrated)

### Overview

In the previous step, we created Graph API access keys using the admin id to access data in user's mailbox or drive on M365.

In order for LegacyFlo to be able to access contents of a mailbox or drive, access to that mailbox has to be delegated to the admin id. This step is required to be repeated for all users whose content has to be copied to Vaultastic Store.

### NOTES:

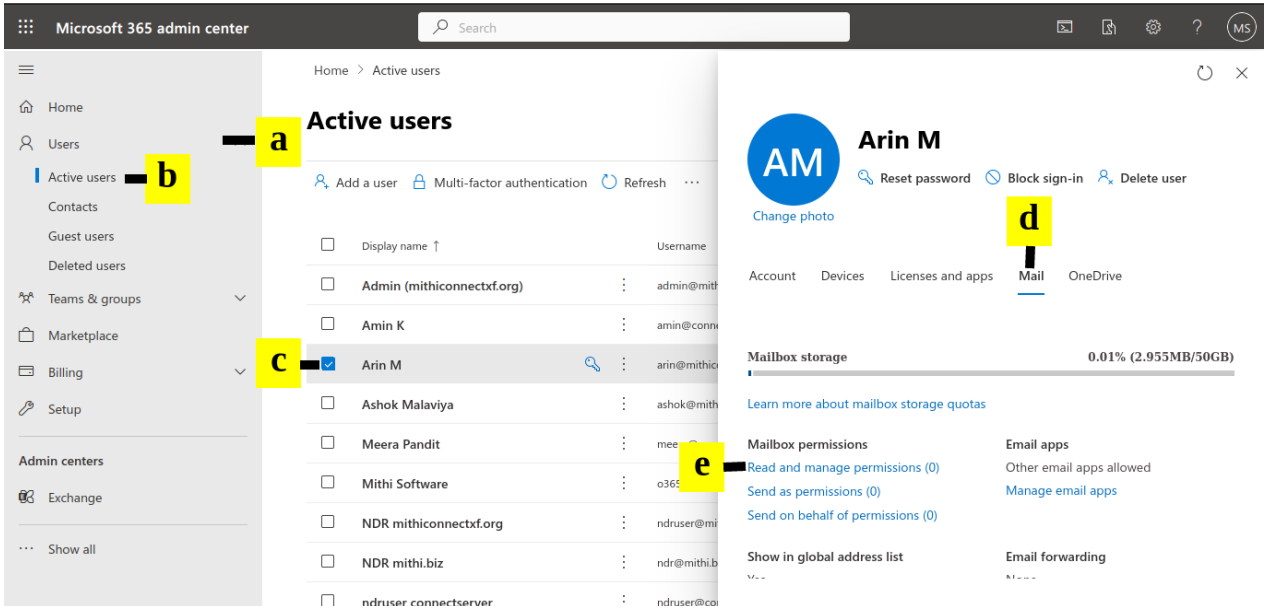
- 1. The admin ID must have a valid license - a basic license is sufficient. (License application takes about an hour to come into effect)**
- 2. This step is required for all request types connecting to the mailboxes on O365. Those which connect via API (e.g. M365API-S3-ZIP) AND those which connect via IMAP (e.g. O365-S3-ZIP)**

### OPTION 1: Delegate access one user at a time

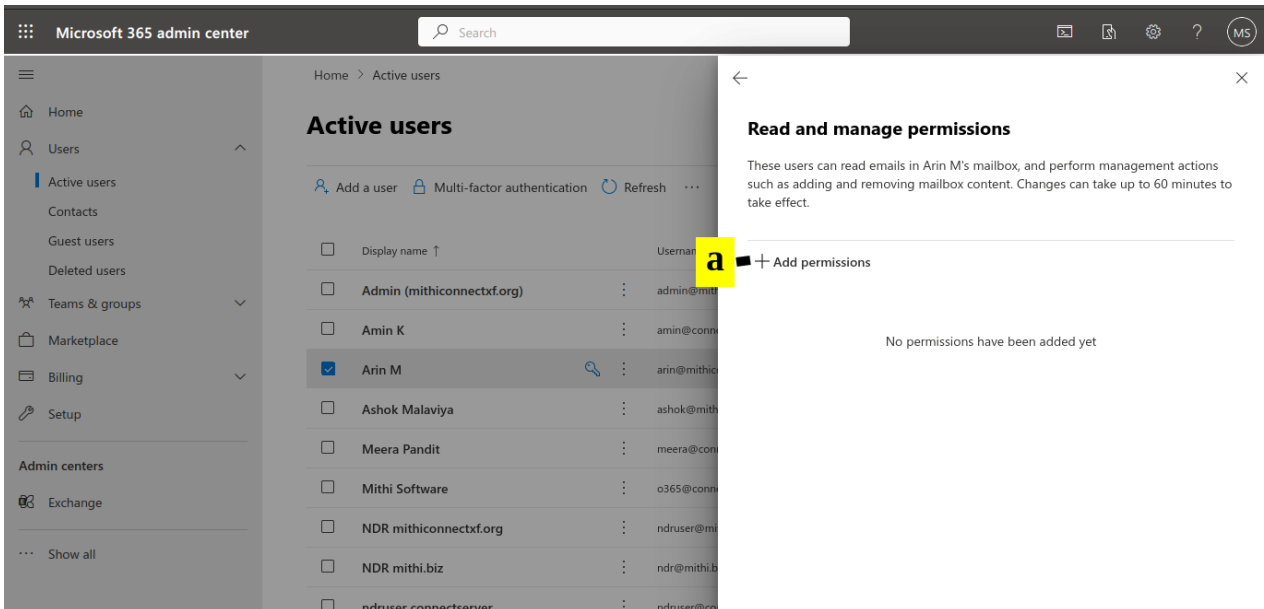
Login to Microsoft 365 admin center account and navigate to <https://admin.microsoft.com> (<https://admin.microsoft.com/>).

Http link: <https://admin.microsoft.com> (<https://admin.microsoft.com/>)

- (a) Select **Users**
- (b) Click on **Active users**
- (c) Select any licensed user whose account access has to be delegated
- (d) Click on the **Mail** tab for the selected user
- (e) Click on **Read and manage permissions**

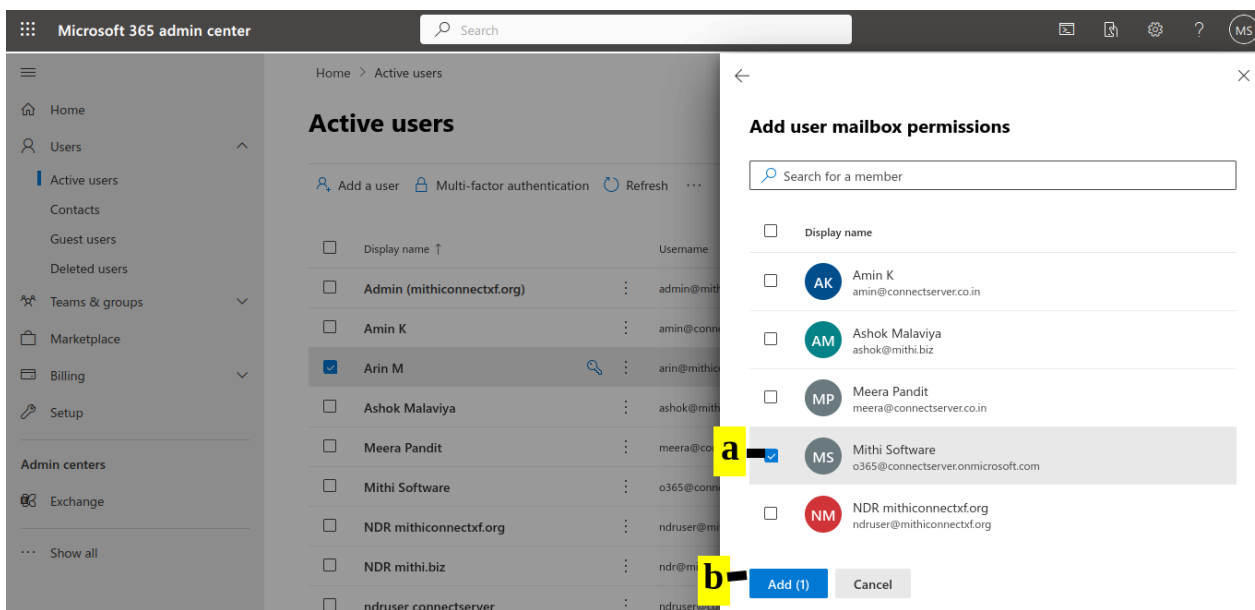


- (a) Click on **Add permissions**



(a) Select the user who will get delegated access

(b) Click on Add



**NOTE: Delegation takes about an hour to come into effect.**

## OPTION 2: Delegate access for all or selected users

Login to the M365 Admin Panel

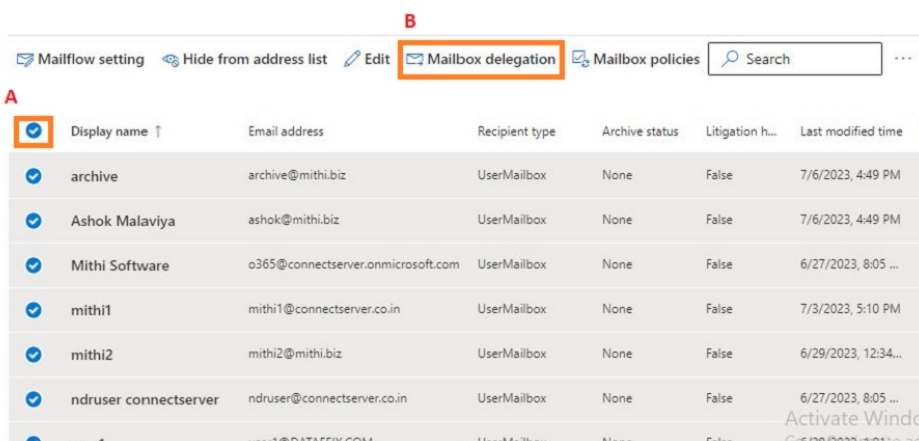
Navigate to **Exchange > Recipients > Mailboxes**

(A) Select All users

(B) Click on Mailbox delegation

### Manage mailboxes

Create and manage settings for shared mailboxes. You can also manage settings for user mailboxes, but to add or delete them you must go to the [Microsoft 365 admin center](#) and do this on the [active users](#) page. [Learn more about mailboxes](#)

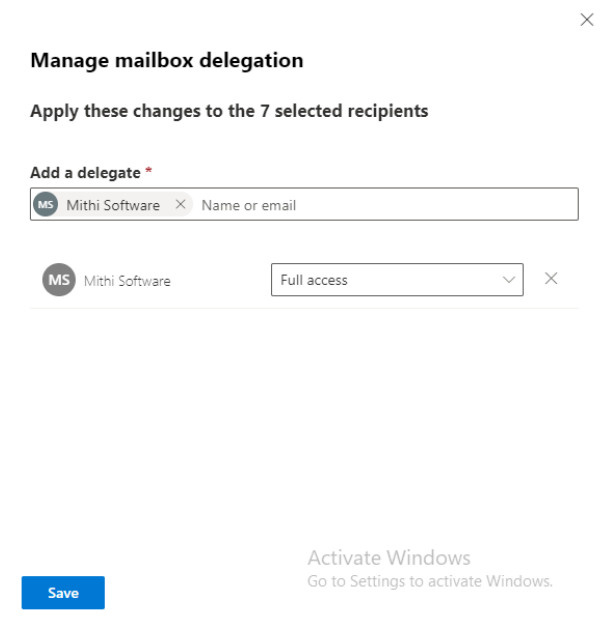


(C) Select the user ID for which we have generated the Graph API credentials

(D) Select Full access

(E) Save

**Note:** Post operation, it can take upto an hour for the delegation to be completed.



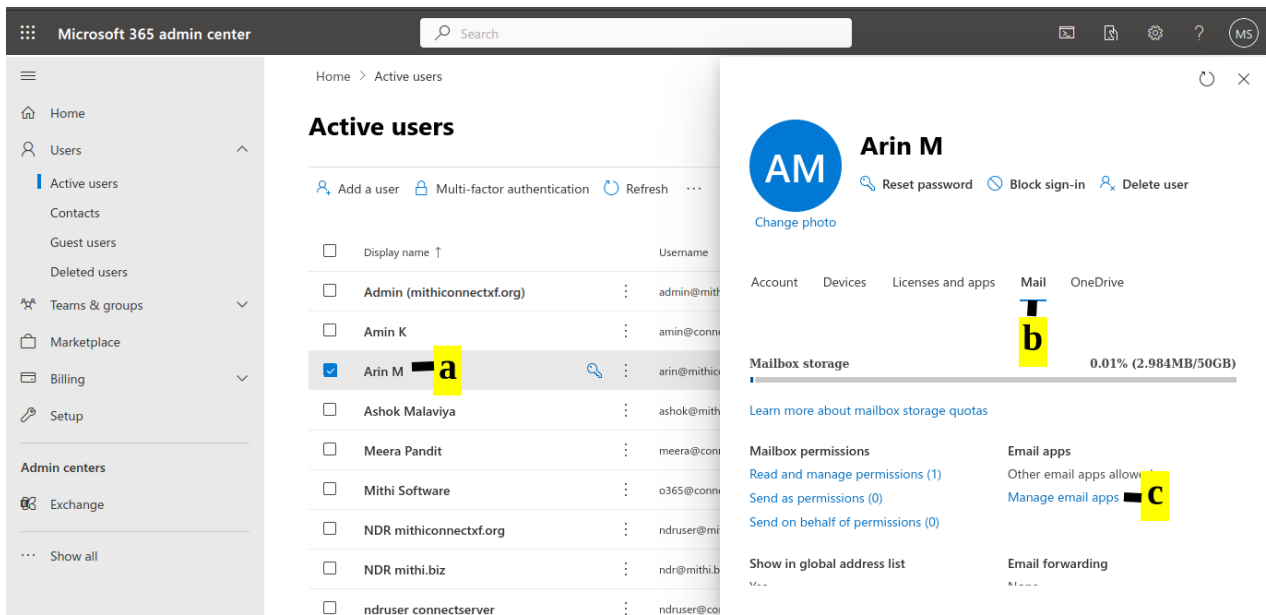
## Enable IMAP access (has to be repeated for every user whose mailbox content are to be migrated)

**NOTE:** This step has to be done only for the LegacyFlo request types which connect to the source using IMAP. For example jobs such as O365-S3-ZIP and O365-S3-PST.

(a) Select the user whose IMAP access has to be enabled (must have a valid license)

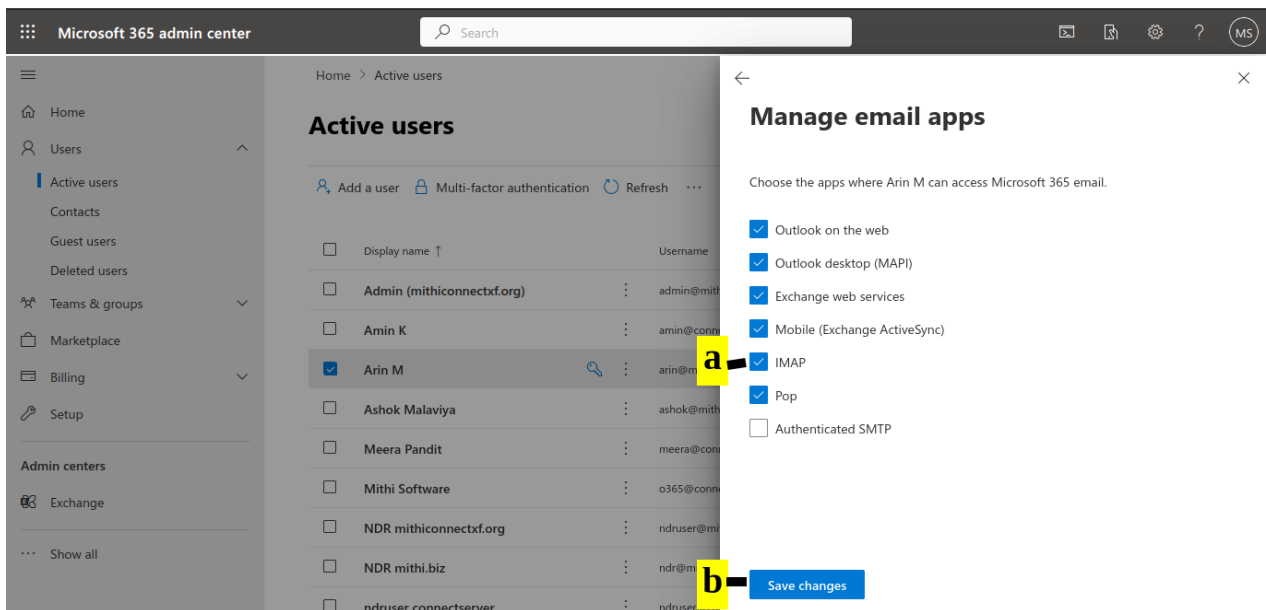
(b) Select the **Mail** tab

(c) Click on **Manage email apps**



(a) Select **IMAP**

(b) Click on **Save changes**



## Bulk Operations using the power shell

### Delegating access for all

- 1) Open Window power shell with administrative mode
- 2) Download packets of Connect-ExchangeOnline

(link to refer ;- <https://www.sharepointdiary.com/2021/11/connect-to-exchange-online-using-powershell.html>)

- Type Command- **Install-Module -Name ExchangeOnlineManagement**
- Now Type command- **Import-Module ExchangeOnlineManagement** (Module will be imported)

### To connect on exchange Admin ID

```
Connect-ExchangeOnline -UserPrincipalName o365@connectserver.onmicrosoft.com ()
```

### Assing delegate user from which we have generated Microsoft graph API key.

```
$DelegateUser = "o365@connectserver.onmicrosoft.com ()"
```

### Get all mailboxes username list from tenant environment in filed string name mailbox.

```
$Mailboxes = Get-Mailbox -ResultSize Unlimited
```

### Run below CMD to Loop through each mailbox and add full access permission.

Note :- Keep CMD terminal on if user list is more till it get completed and network connectivity should be proper other wise terminal will get disconnect and script will get kill in between.

```
foreach ($Mailbox in $Mailboxes) {
    try {
        Add-MailboxPermission -Identity $Mailbox.PrimarySmtpAddress -User $DelegateUser -AccessRights
FullAccess -InheritanceType All -AutoMapping $false
        Write-Host "Full access granted to $DelegateUser for mailbox $($Mailbox.PrimarySmtpAddress)"
    }
    catch {
        Write-Host "Error granting access to mailbox $($Mailbox.PrimarySmtpAddress): $_"
    }
}
```

### Disconnect Exchange

```
Disconnect from Exchange Online
```

```

PS C:\Users\Dell> Connect-ExchangeOnline -UserPrincipalName o365@connectserver.onmicrosoft.com

-----
This V3 EXO PowerShell module contains new REST API backed Exchange Online cmdlets which doesn't require WinRM for Client-Server communication. You can now run these cmdlets af
ter turning off WinRM Basic Auth in your client machine thus making it more secure.

Unlike the EXO* prefixed cmdlets, the cmdlets in this module support full functional parity with the RPS (V1) cmdlets.

V3 cmdlets in the downloaded module are resilient to transient failures, handling retries and throttling errors inherently.

REST backed EOP and SCC cmdlets are also available in the V3 module. Similar to EXO, the cmdlets can be run without WinRM basic auth enabled.

For more information check https://aka.ms/exov3-module
-----

PS C:\Users\Dell> $DelegateUser = "o365@connectserver.onmicrosoft.com"
PS C:\Users\Dell>
PS C:\Users\Dell>
PS C:\Users\Dell> $Mailboxes = Get-Mailbox -ResultSize Unlimited
PS C:\Users\Dell>
PS C:\Users\Dell>
PS C:\Users\Dell> foreach ($Mailbox in $Mailboxes) {
>> try {
>>     Add-MailboxPermission -Identity $Mailbox.PrimarySmtpAddress -User $DelegateUser -AccessRights FullAccess -InheritanceType All -AutoMapping $false
>>     Write-Host "Full access granted to $DelegateUser for mailbox $($Mailbox.PrimarySmtpAddress)"
>> }
>> catch {
>>     Write-Host "Error granting access to mailbox $($Mailbox.PrimarySmtpAddress): $_"
>> }
>> }

Identity            User                AccessRights
-----
c20819c7-7a2e-4ee... 5-1-5-21-40166369... {FullAccess}
WARNING: The appropriate access control entry is already present on the object "CN=c20819c7-7a2e-4eec-b22a-a80d37ad7022,OU=connectserver.onmicrosoft.com,OU=Microsoft Exchange
Hosted Organizations,DC=INDPR01A001,DC=PROD,DC=OUTLOOK,DC=COM" for account "5-1-5-21-4016636904-1928851825-4262717397-12347600".
Full access granted to o365@connectserver.onmicrosoft.com for mailbox archive@mithi.biz
ashok                5-1-5-21-40166369... {FullAccess}
WARNING: The appropriate access control entry is already present on the object "CN=ashok,OU=connectserver.onmicrosoft.com,OU=Microsoft Exchange Hosted
Organizations,DC=INDPR01A001,DC=PROD,DC=OUTLOOK,DC=COM" for account "5-1-5-21-4016636904-1928851825-4262717397-12347600".
Full access granted to o365@connectserver.onmicrosoft.com for mailbox ashok@mithi.biz

```

Choose File No file selected

### Removing delegation access

- 1) Open Window power shell with administrative mode
- 2) Download packets of Connect-ExchangeOnline  
(link to refer ;- <https://www.sharepointdiary.com/2021/11/connect-to-exchange-online-using-powershell.html>)

- Type Command- **Install-Module -Name ExchangeOnlineManagement**
- Now Type command- **Import-Module ExchangeOnlineManagement** (Module will be imported)
- Now type the command: **Connect-ExchangeOnline**
- Given the credentials of the admin id.

3) Define the power shell script as follows

```

$mailboxes = "User1","User2"
$mailboxes | ForEach-Object{
$mailbox = $_
Remove-MailboxPermission -Identity $mailbox -User "User to whom delegation has been given" -AccessRights
FullAccess -InheritanceType All}

```

**Notes:**

- **User ids have to be enclosed in double quotes**
- **"User1", "User2", etc. are the ids from which the delegation has to be removed**
- **"User to whom delegation has been given" is the user to whom the access has been given**

Example:

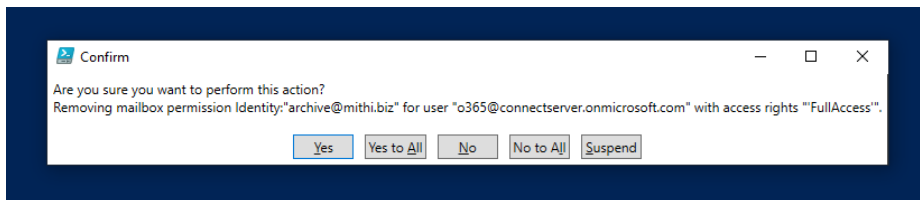
```
$mailboxes = "archive@mithi.biz ()","user@mithi.biz ()"
```

```
$mailboxes | ForEach-Object{
```

```
$mailbox = $_
```

```
Remove-MailboxPermission -Identity $mailbox -User "admin@connectserver.onmicrosoft.com ()" -AccessRights  
FullAccess -InheritanceType All}
```

4) On execution, you will be prompted to confirm the operation. **Click Yes**



5) Verify that the access has been removed using the Admin Center

---