**The Guide v1.0**

First, here’s the setup, I was trying to model the LFP – SPL situation as precisely as possible. In our example **agzsolt.com** is used to be the destination tenant (SPL), and **jd0e.com** is the organization to be moved (LFP)

**The setup**

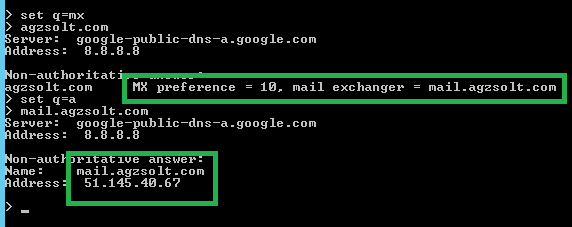
**Destination:**

**agzsolt.com**

Hybrid

O365: agzsolt.onmicrosoft.com

Onprem: 51.145.40.67



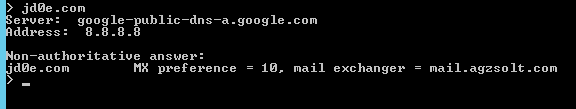
**Source:**

**Jd0e.com**

Hybrid

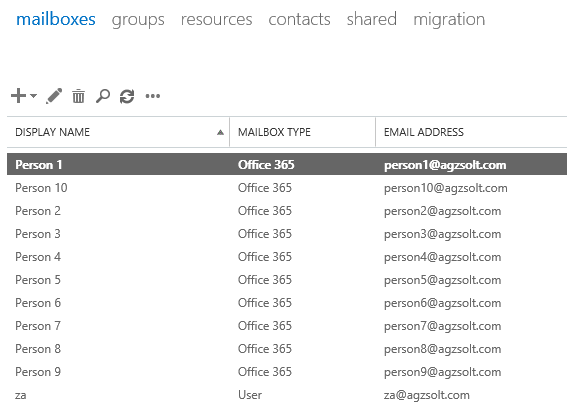
O365: jdoe.onmicrosoft.com

Onprem: 51.143.157.86

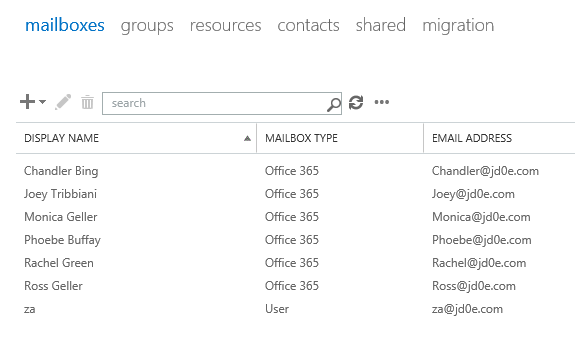


**Mailbox situation:**

Destination – agzsolt.com:

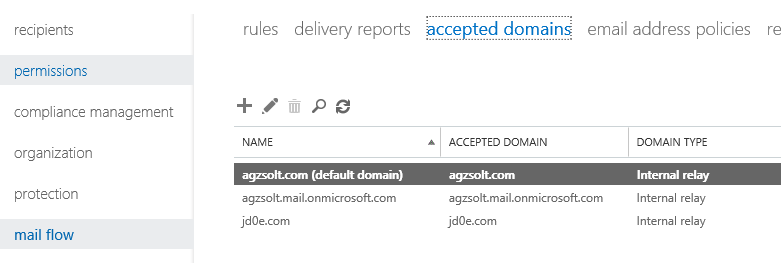
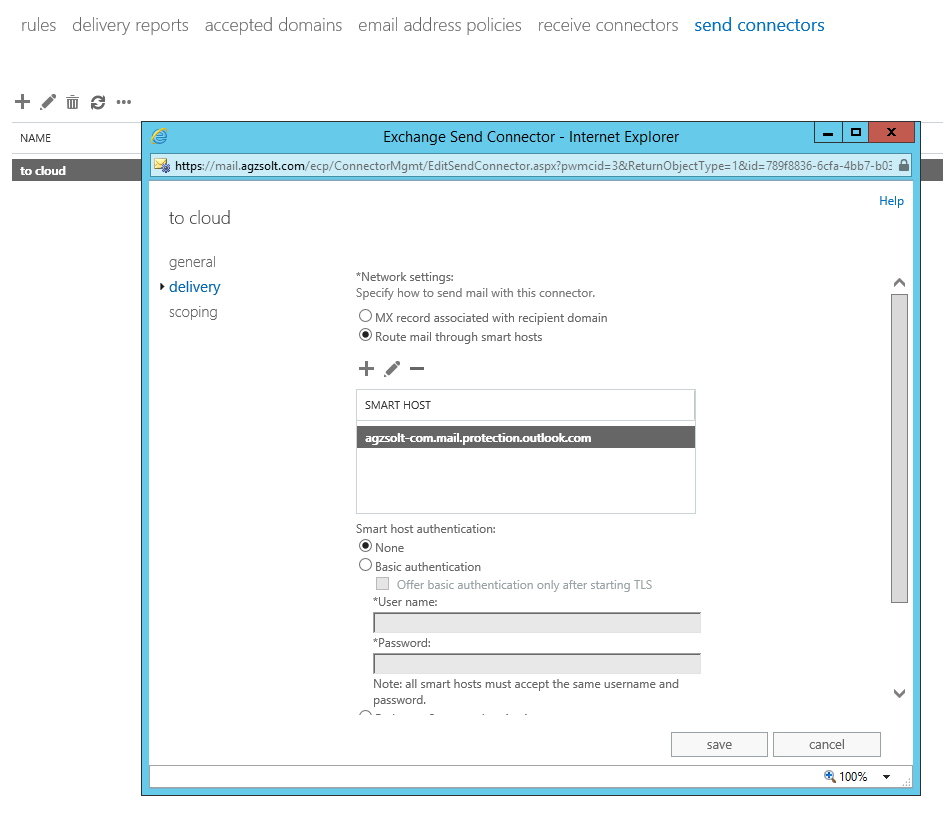


Source – jd0e.com:

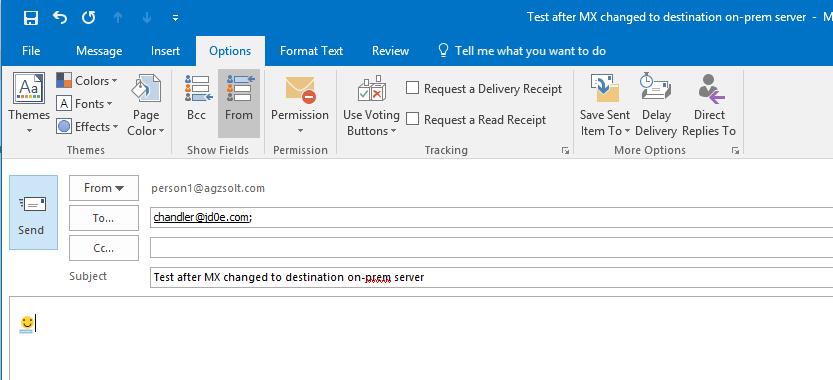


To test the mail flow, we send an email from gmail to jd0e. Note the jd0e MX record has already been directed to the destination on-prem server. Also the SPF record is updated accordingly to prevent the sent emails to be put in the recipient’s junk folder.

**Prepare the destination on-prem server**

1. Put jd0e.com in the accepted domain list  
   
2. If not already done, create a send connector to the cloud  
   
3. Set the jd0e.com MX record and SPF record accordingly

|  |
| --- |
| Spf for jd0e:  **v=spf1 include:mail.agzsolt.com include:spf.protection.outlook.com –all**  jd0e.com **MX** record to **mail.agzsolt.com** |





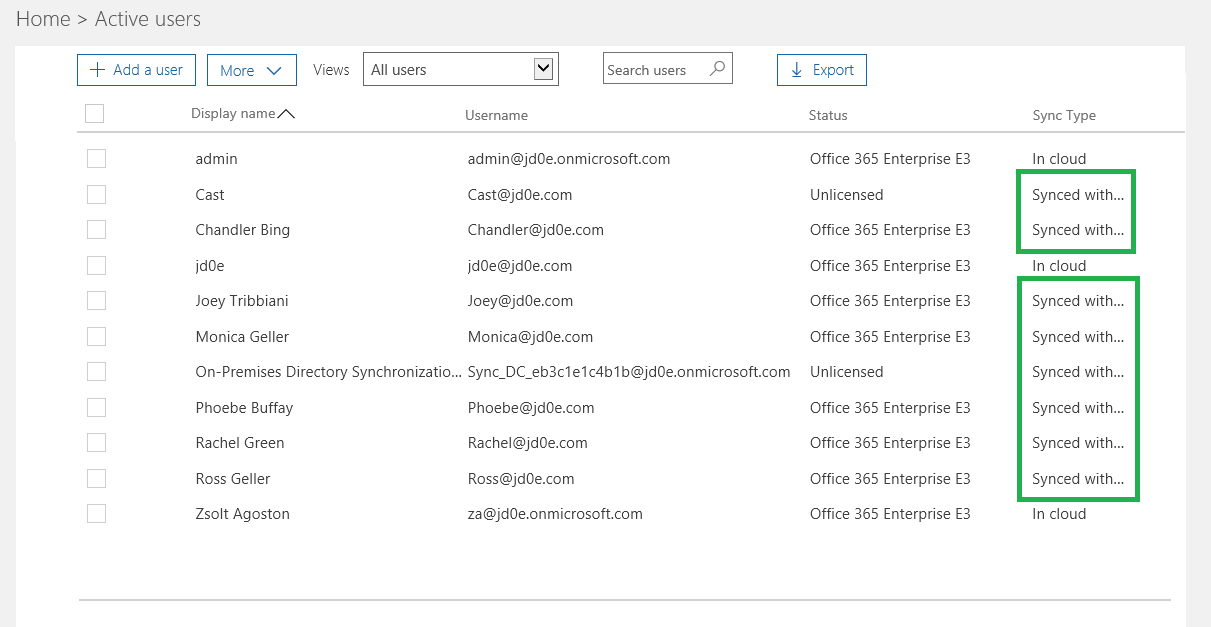
The email has arrived without an issue!

Ok, so let’s start the process!

**Cut jd0e.com dirsync (can take up to 72 hours to complete)**

(this converts accounts to cloud account while keeping their original password)

Before:



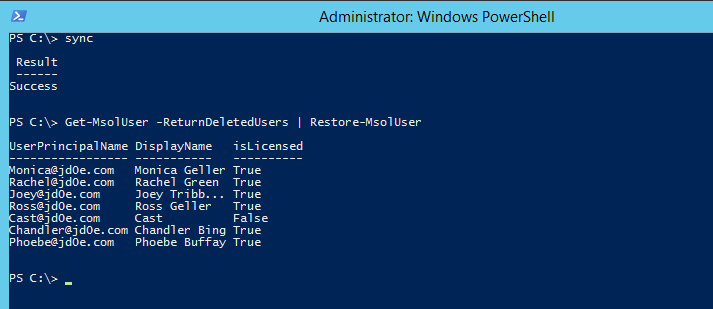
|  |
| --- |
| Set-MsolDirSyncEnabled -EnableDirSync:$false |

Check if the process has run:

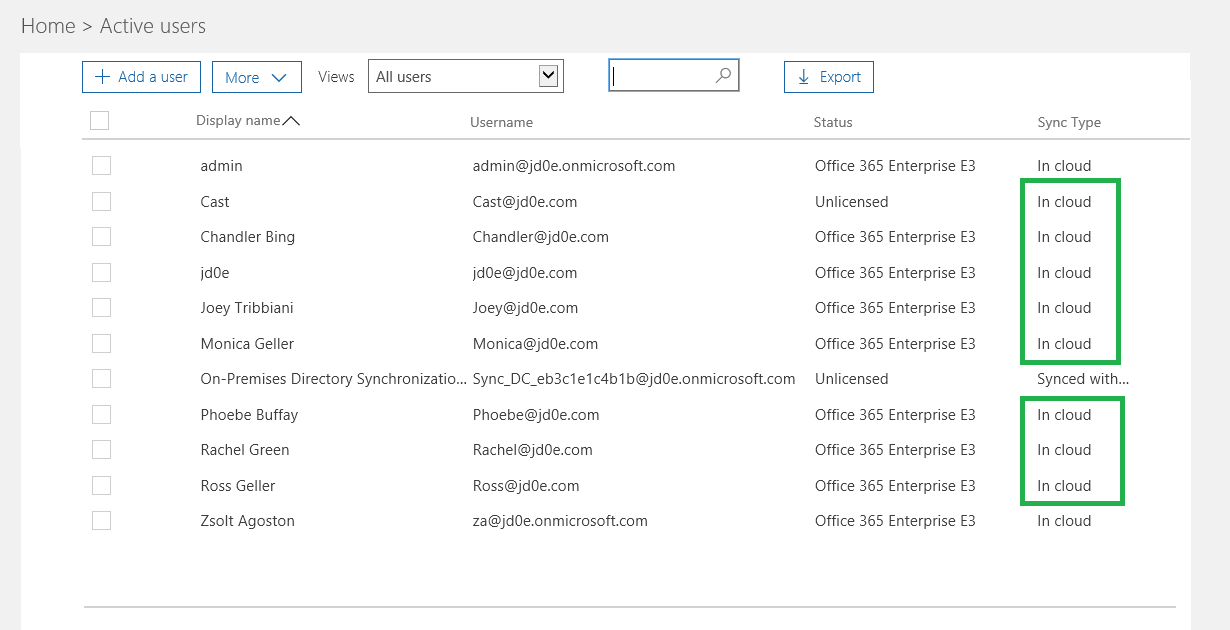
|  |
| --- |
| (Get-MSOLCompanyInformation).DirectorySynchronizationEnabled |

Note, if the process takes very long there’s another way: simply move the user accounts to an OU that is not synced to the cloud, and wait for or force a sync cycle. That will soft delete the cloud accounts, after which the can be restored using the following command. They cloud system will restore them as cloud accounts, preserving old passwords, permission settings as well.

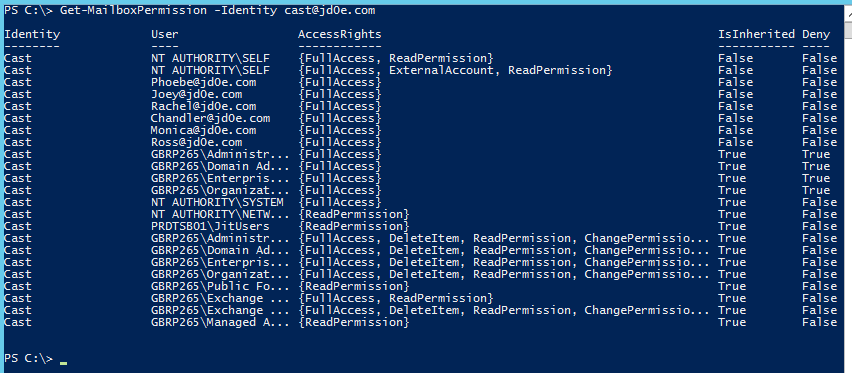
|  |
| --- |
| Get-MsolUser -ReturnDeletedUsers | Restore-MsolUser |

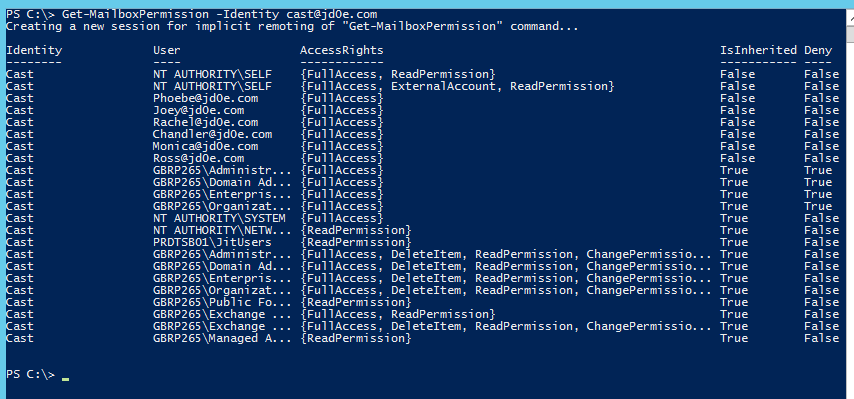


After:



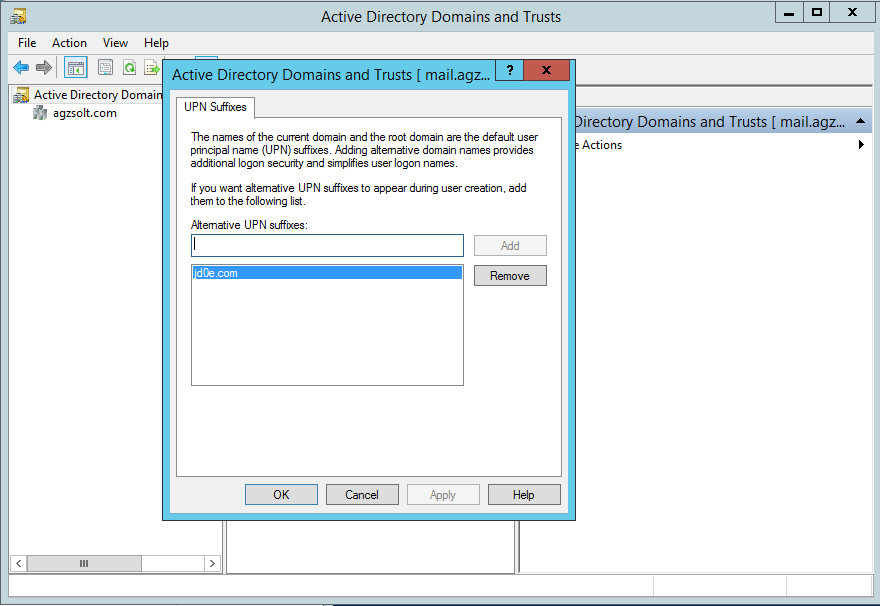
Now we check the permissions on the shared mailbox to make sure they are not lost like after a license unassign-reassign scenario. As seen below the situation before and after match so we are good to go!





**Create jd0e.com users in the agzsolt.com local AD in a non-synced OU**

First, we add the jd0e.com domain to the Active Directory Domains and Trusts temporarily to make the transition simpler for the users. This way they will be able to log in using the underlying Kerberos ticketing system – menaing no password prompts ☺



We run the following script to create the users, which will be created from users.csv

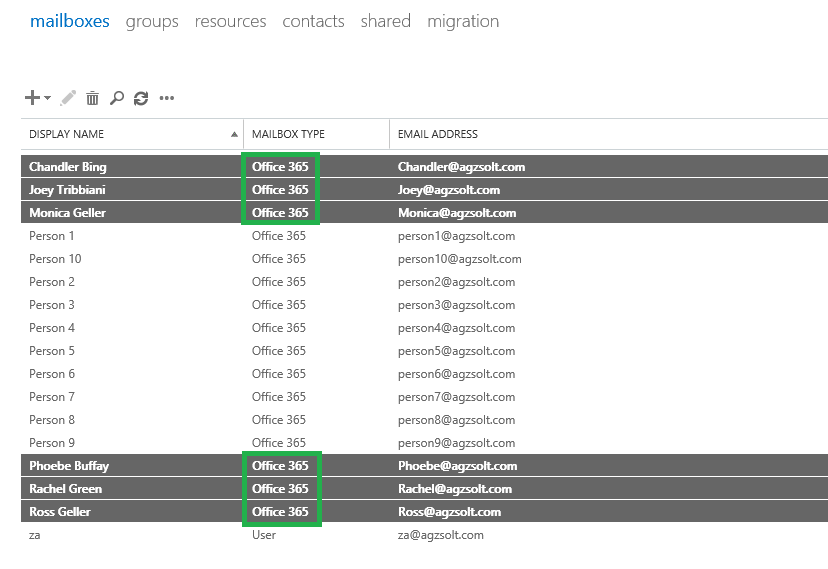
**Users.csv**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **FirstName** | **LastName** | | Ross | Geller | | Joey | Tribbiani | | Monica | Geller | | Rachel | Green | | Chandler | Bing | | Phoebe | Buffay | |

**Script:**

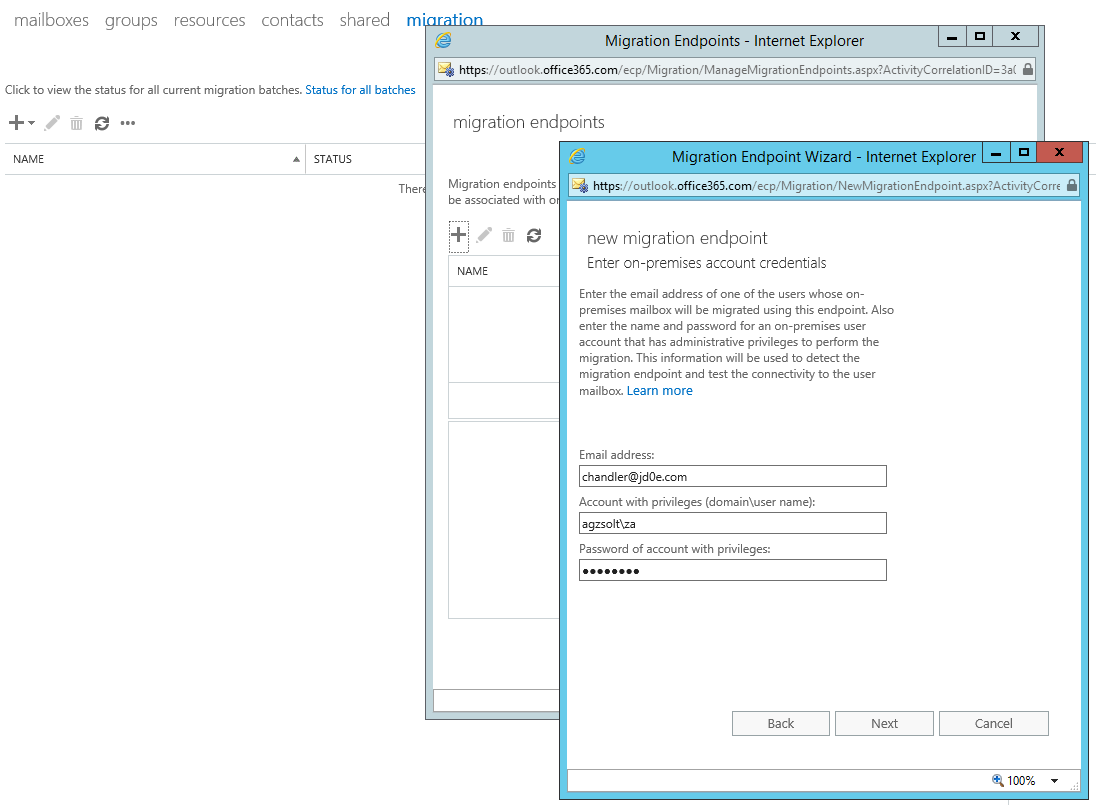
|  |
| --- |
| New-ADOrganizationalUnit -Name "jd0e" -Path "OU=My Business,DC=agzsolt,DC=com" –Verbose  import-csv users.csv | foreach {  $fn=$\_.FirstName  $ln=$\_.LastName  New-ADUser -Name "$fn $ln" -DisplayName "$fn $ln" -GivenName "$fn" -Surname "$ln" -UserPrincipalName $fn@jd0e.com -Path "OU=jd0e,OU=My Business,DC=agzsolt,DC=com" -Enabled:$true -EmailAddress "$fn@jd0e.com" -AccountPassword(ConvertTo-SecureString "Password12345!" -AsPlainText -Force)  Enable-RemoteMailbox -Identity $fn@jd0e.com -RemoteRoutingAddress $fn@jd0e.onmicrosoft.com  } |

After the commands being run we check the results on the agzsolt.com server:

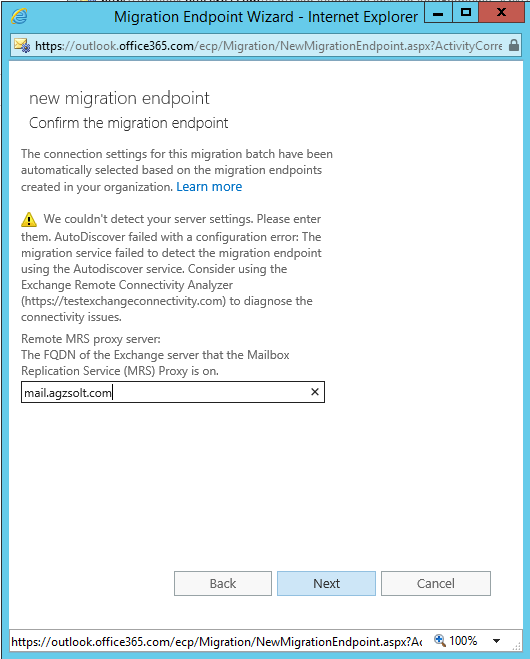
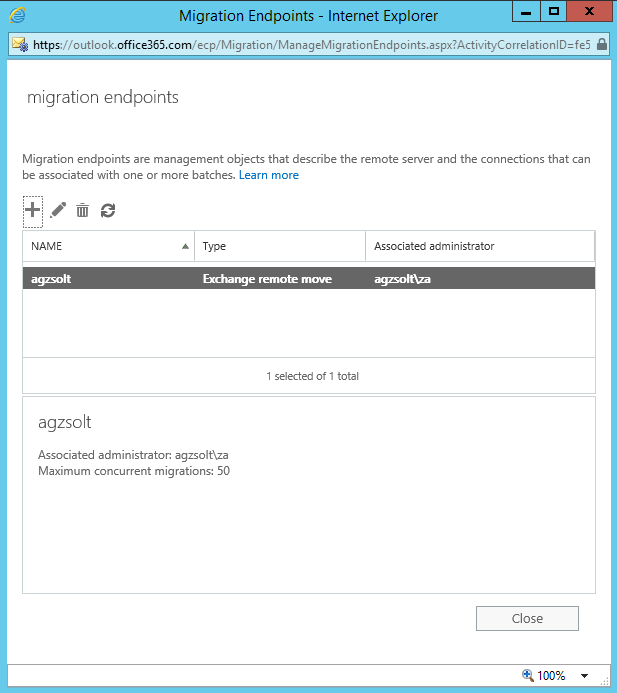


**Configure the cross-forest hybrid environment**

To make the servers able to migrate jd0e.com mailboxes to the agzsolt.com server we need to create a migration endpoint in the jd0e.com **cloud** server. It is done in **recipients/migration/migration endpoints**, as the new endpoint’s type we use **“exchange remote"**



It will fail because the server tries to determine the destination FQDN using autodiscover which points to the wrong location of course. We put the right server manually:

In our example we will call the connector “**agzsolt**”

**Sync the MSOL attributes into the agzsolt.com local AD accounts**

Here the most important thing is that the **ExchangeGUID** attribute of the accounts on the destination on-prem server must match the **ExchangeGUID** attribute of the actual cloud mailboxes. We can sync it doing the following:

1. On the cloud server run the following command that will create a file called **mailboxes.xml**, with all the mailboxes and all attributes we need:

|  |
| --- |
| Get-Mailbox -ResultSize Unlimited | Export-Clixml mailboxes.xml |

Now we import the important attributes in the on-prem server for compliance:

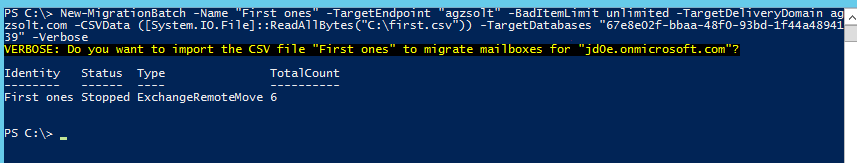
|  |
| --- |
| import-clixml mailboxes.xml | foreach {  $name=$\_.name  $alias=$\_.alias  $guid=$\_.exchangeGUID  Set-RemoteMailbox -Identity $name -ExchangeGuid $guid -EmailAddresses @{add="$alias@jd0e.com"}  write-host "$name has given GUID: $guid"  } |

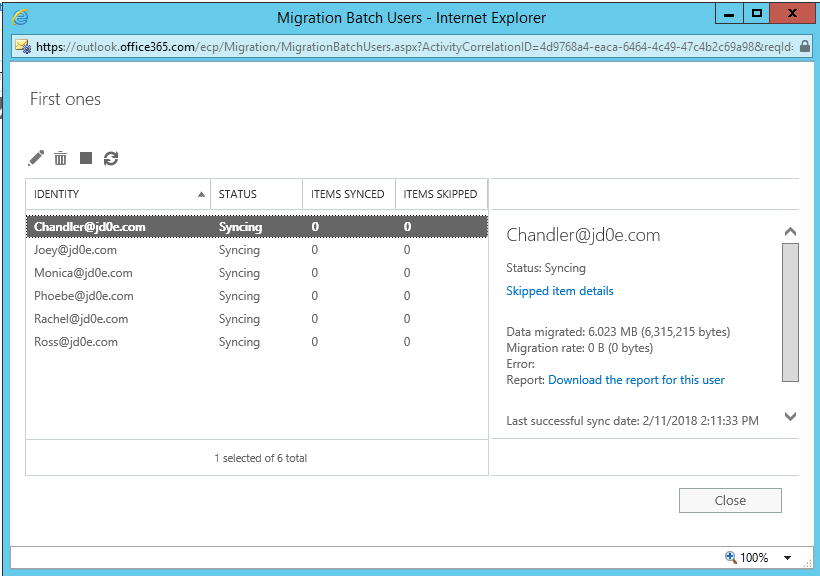
1. Now we create a migration batch that will synchronize and move the jd0e.com mailboxes to the on-prem endpoint. We create our first batch called “**First ones**” from **first.csv**:  
     
   **first.csv**

|  |
| --- |
| EmailAddress  Ross@jd0e.com  Joey@jd0e.com  Monica@jd0e.com  Rachel@jd0e.com  Chandler@jd0e.com  Phoebe@jd0e.com |

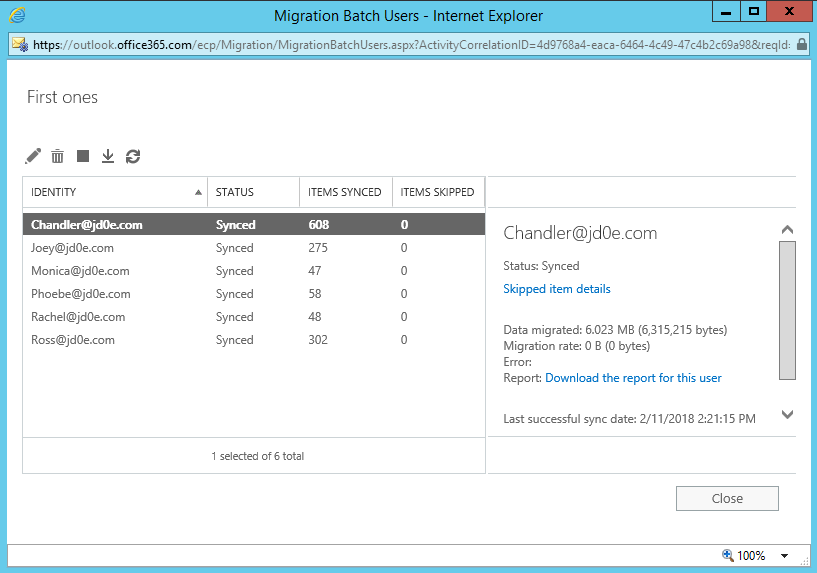
**Script**

|  |
| --- |
| New-MigrationBatch -Name "First ones" -TargetEndpoint "agzsolt" -BadItemLimit unlimited -TargetDeliveryDomain agzsolt.com -CSVData ([System.IO.File]::ReadAllBytes("C:\first.csv")) -TargetDatabases "67e8e02f-bbaa-48f0-93bd-1f44a4894139" -Verbose |

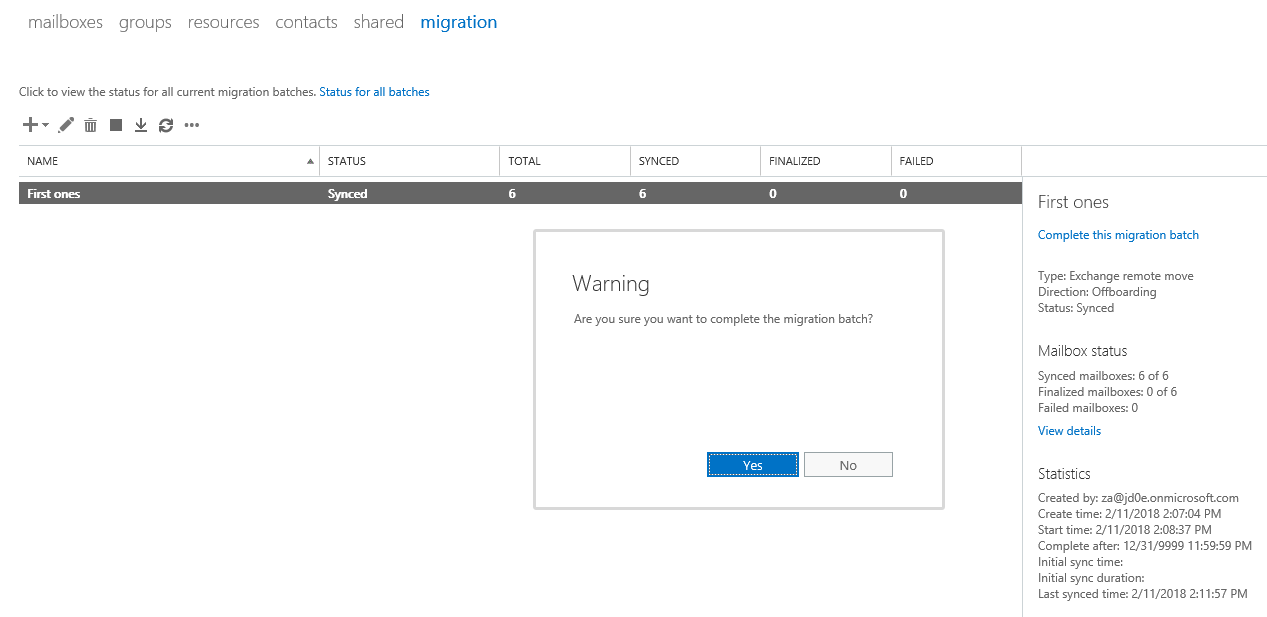




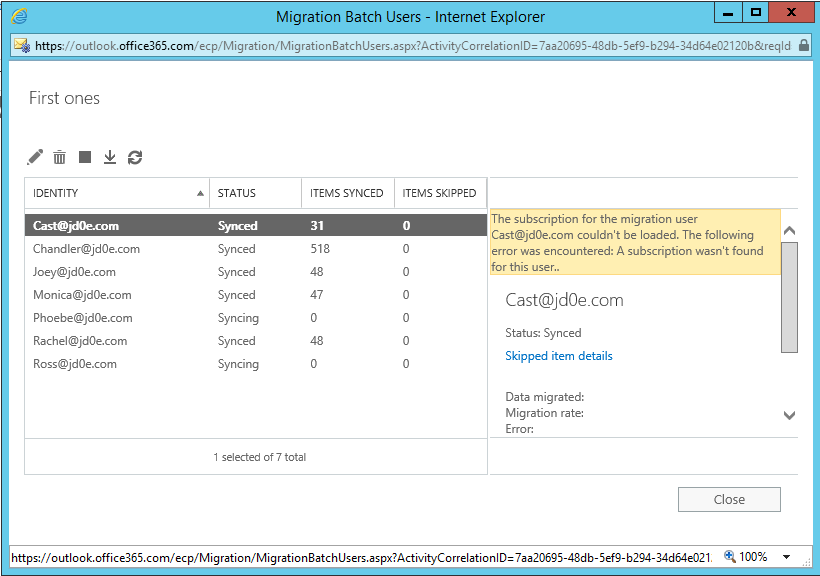
When it’s done we are ready to finish the whole batch or just specific mailboxes, it depends on our needs



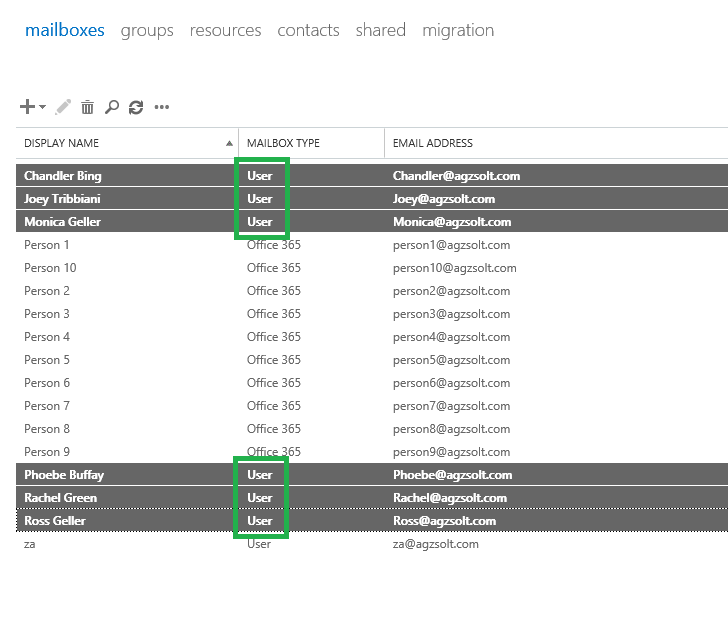
We complete the migration batch and make sure the users can access their mailboxes and they are functional



Just to mention: the shared mailboxes need a license assigned to them

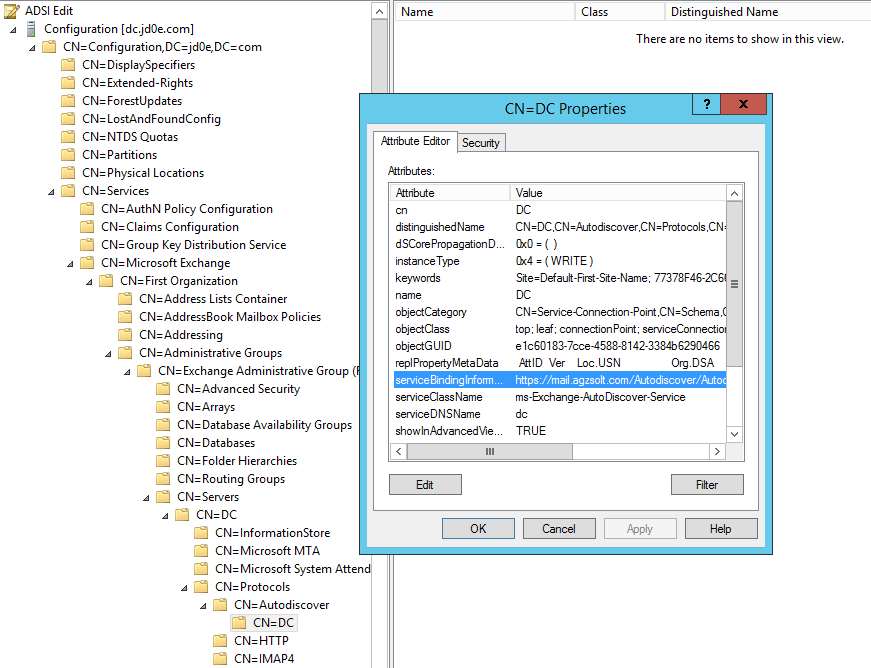


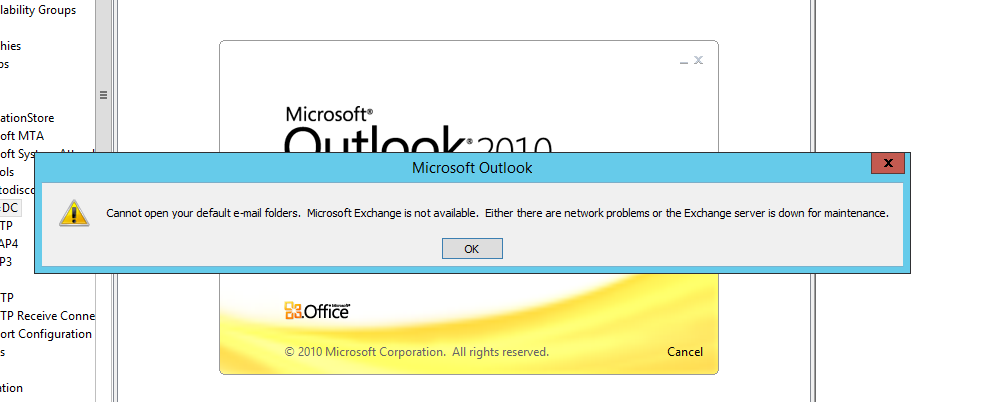
After the migration, we see that the on-prem server handles the mailboxes as local mailboxes:



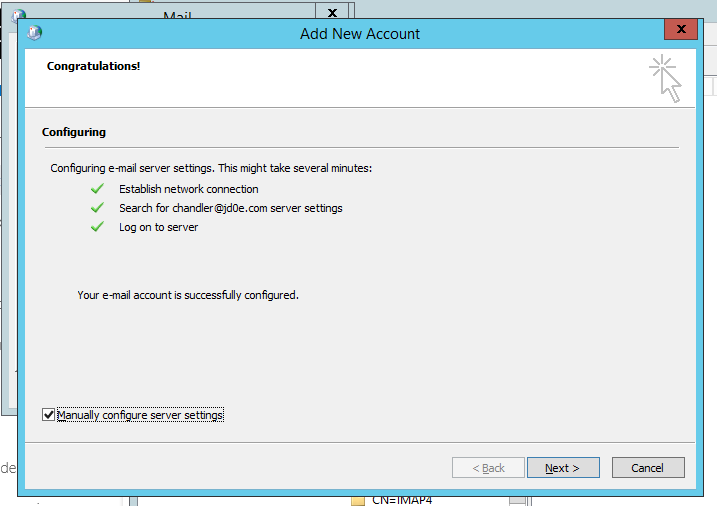
Now we change the local **SCP** record for **autodiscover** on the jd0e.com (source) **domain controller**, because this is the first place the server is looking for autodiscover.xml data which will not return the right values, we change that to point to the agzsolt.com (destination) autodiscover datafile:

|  |
| --- |
| https://mail.agzsolt.com/Autodiscover/Autodiscover.xml |

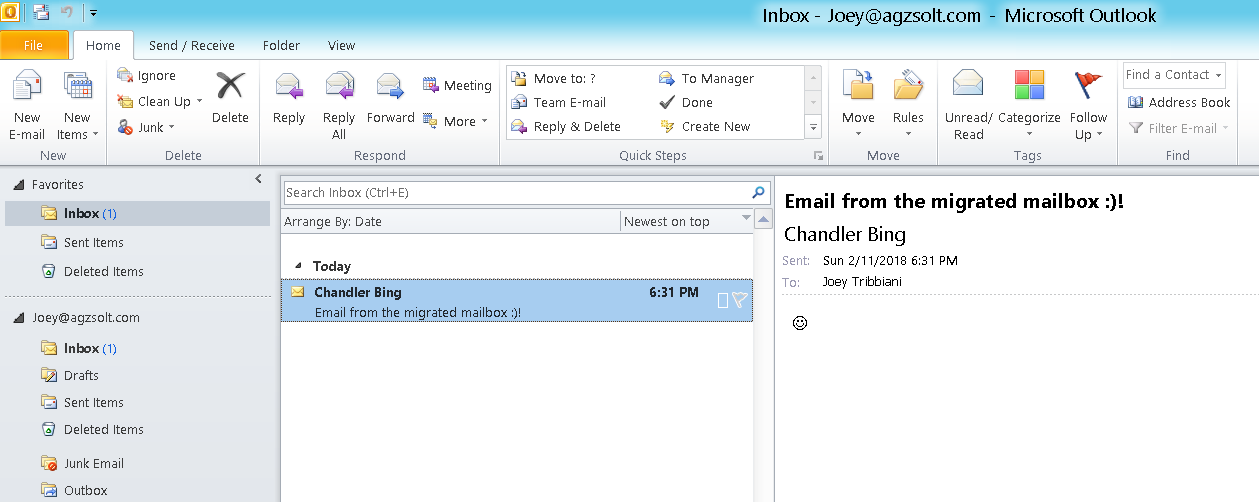


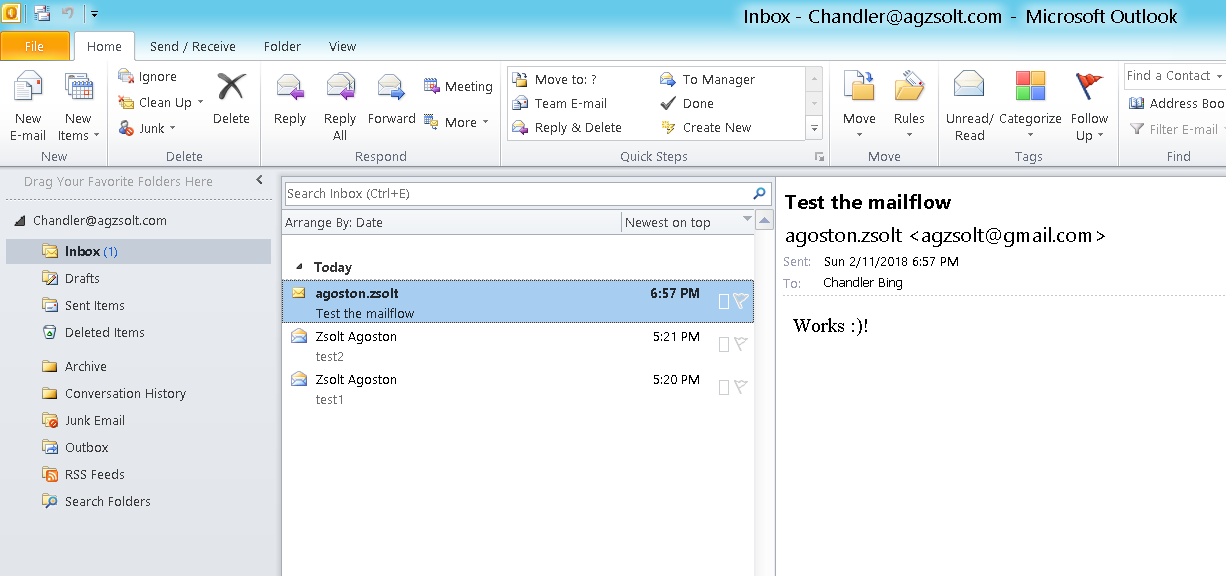
Once the users try to log in, Outlook will throw an error message:

A solution is to recreate the profile on their client computers. In fact, because of the on-prem kerberos system, they won’t be prompted for their passwords (as long as they match on both organizations), which makes the transition much easier



After logging in, we test the mail flow:





Excellent.

We simply need to remove the old jd0e.onmicrosoft.com STMP addresses from the mailboxes.

|  |
| --- |
| import-csv users.csv | foreach {  $name=$\_.FirstName  Set-Mailbox -Identity $name -EmailAddresses @{remove="$name@jd0e.onmicrosoft.com" }  } |

Now we remove jd0e.com domain from the jd0e.onmicrosoft.com tenant and add that to agzsolt.onmicrosoft.com

From this point it’s a normal migration to the cloud scenario.

☺

Z.