



Content Matrix Performance Optimization Guide

January 2018

Copyright

© 2018 Copyright Metalogix Corporation International GmbH.

All rights reserved. No part or section of the contents of this material may be reproduced or transmitted in any form or by any means without the written permission of Metalogix Software Corporation.

Metalogix Content Matrix Consoles™ are trademarks of Metalogix Software Corporation.

Windows SharePoint Services is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries. Other product and company names mentioned herein may be the trademarks of their respective owners.

Technical Support

For information about Metalogix Technical Support, please visit <http://metalogix.com/support>.

Technical support specialists can be reached by email at support@metalogix.com or by phone at 1.202.609.9100.

We want to hear from you. Please send any comments regarding this document to support@metalogix.com.

The level of technical support provided depends upon the support package that you have purchased. Contact us to discuss your support requirements.

Contents

Introduction	4
Requirements	4
Special Considerations for Migrations to a SharePoint On Premises Target	4
Scaling Hardware	4
Scaling Out	4
Dedicated Migration Servers	4
Installation on WFE Servers	6
Scaling Up	6
Hardware Best Practices	7
Non-WFE server/agent	7
WFE server/agent	7
Connection Adapters	7
Database (DB)	7
Local Object Model (OM)	8
Metalogix Extensions Web Service (MEWS)	8
Native Web Services and CSOM (NWS)	9
Special Considerations for SharePoint Online Targets	9
Functional and API Differences between SPO and SharePoint that impact migrations	10
Different Application Programming Interfaces available for SPO	10
Performance of Migrations to SPO with the new Migration API	11
Additional Best Practices unique to Content Matrix that can improve SPO migration performance even more	13
Performance Expectations	15
Recommendations	15
Setup an Azure Blob Storage Account	15
Use Microsoft Provided Containers	19
Use Azure Based Virtual Machines	21

Use Azure Active Directory Setup.....	24
To achieve performance levels of 10GB/hour or higher	25
Disabling Unnecessary Features.....	26
Multithreading and Batch Jobs.....	27
Single Jobs.....	27
Configuring Single Job Multithreading	27
Optimizing Single Job Multithreading	28
Concurrent Jobs.....	28
Configuring Concurrent Jobs	29
Optimizing Concurrent Jobs	29
Batched Jobs.....	30
Distributed Migration	31
Configuration	31
PowerShell.....	32
PowerShell	33
Configuring PowerShell.....	33
Optimizing PowerShell	33

Introduction

This document covers a number of different topics to help you with optimizing the performance for migrating your content into SharePoint with Content Matrix Console. These topics include: hardware best practices and scaling, SharePoint connection adapters, PowerShell, and special considerations for migrating to SharePoint Online.

These migration optimizations are only a general set of guideline to help you with your migration. For more information on any of the specific features and abilities of Content Matrix Console, please see the appropriate Edition's Help file (located within the client application, or they can be found online at: <https://www.metalogix.com/documentation>).

Requirements

The only requirement that is necessary to use this document is to be in the process of installing, or have already installed, Content Matrix Console into your environment, for migrating your content into SharePoint. Since appropriate installation of Content Matrix can have a major impact on performance, it is preferable to modify your installation if necessary based on the guidance in this document.

Please refer to the below documents for details on installing Content Matrix see [Content Matrix Console Quick Start Guide.pdf](#).

Special Considerations for Migrations to a SharePoint On Premises Target

Scaling Hardware

Content Matrix can take advantage of scaling hardware within the migration environment.

Scaling Out

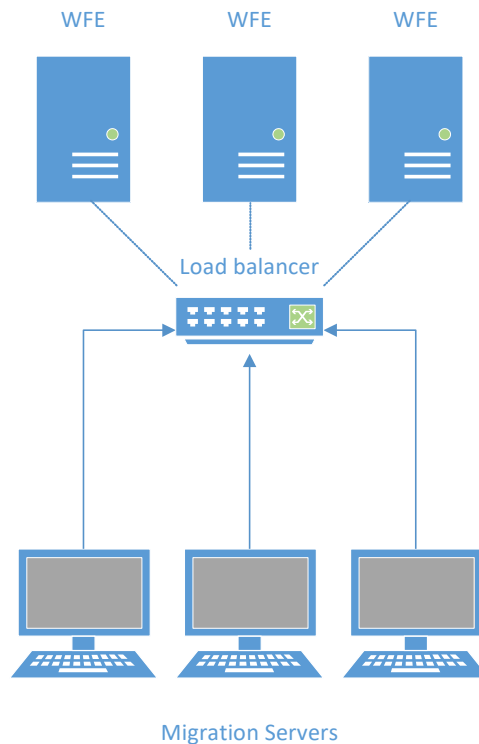
Scaling out is the process of adding servers with Content Matrix installed to perform migrations concurrently to different Web Front End (WFE) servers within the target on premises SharePoint farm.

Dedicated Migration Servers

Content Matrix can be installed on dedicated migration servers to facilitate maximum resource utilization in migrations whenever possible. In this configuration, Content Matrix can then be remotely pointed at the target SharePoint farm.

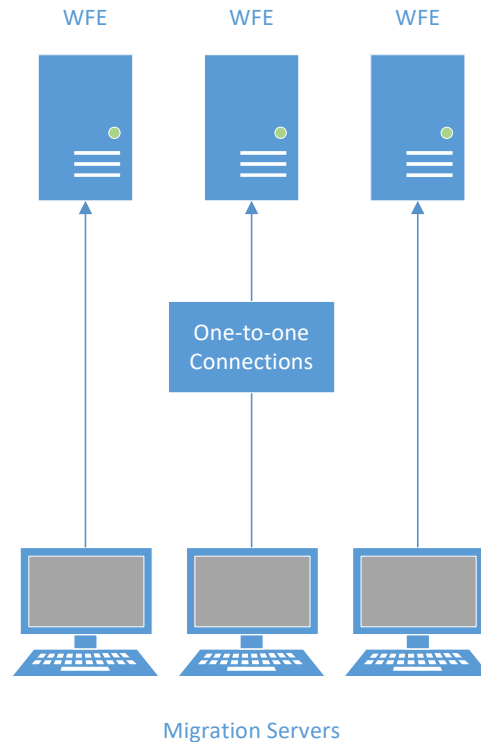
Note: This deployment scenario will require the installation of the Metalogix Extensions Web Service adapter on the target SharePoint server (as defined in the [Connection Adapters](#) section of this document). This will allow for remote connections to the SharePoint Object Model (OM).

Load balancing can be effectively used in this case to ensure that migration requests from Content Matrix will be properly balanced to WFE servers in the target farm based on utilization.



The number of migration servers that can be utilized should be managed to ensure that the target SharePoint server can effectively handle all of the migration requests. Generally a one to one correlation with WFE servers is recommended.

If load balancing is not being used on the target farm, migration servers can be directly pointed at specific WFE servers to ensure maximum concurrent throughput. See below image:



Installation on WFE Servers

When dedicated migration servers are unavailable, Content Matrix can be directly installed on WFE servers. In this configuration, Content Matrix can utilize the local OM adapter for connections.

In this configuration, load balancing will be bypassed due to Content Matrix utilizing only the local WFE OM for migrations.

It is recommended that Content Matrix be installed on the maximum number of idle WFE servers in order to maximize throughput.

Note: If WFE servers are already servicing user requests in a production environment, extreme care should be taken to ensure that the user experience is not impacted. Due to the potential resource utilization of Content Matrix on the host server, it is recommended that migrations via the application never be run on a production WFE **outside** of a maintenance window.

Scaling Up

Scaling up is the process of adding additional hardware to a single migration machine in order to facilitate faster copies.

In general, scaling up is only recommended to the point specified in the hardware best practices section below. It is more efficient to scale out the migration infrastructure and parallelize migration instead of overly increasing resources on a single machine.

Hardware Best Practices

Please consult the following document for minimum requirements when installing Metalogix Content Matrix:

[Content Matrix Console Advanced Installation Guide.pdf](#)

In addition to the minimum requirements laid out in the prior document, migration servers where Content Matrix will be installed will perform optimally with the following hardware configurations:

Non-WFE server/agent

- 4 core CPU or higher
- 8gb of RAM or higher
- 20gb of free disk space

WFE server/agent

- 8 core CPU
- 16gb of RAM or higher
- 50gb of free disk space

Note: Please avoid installation of Content Matrix on a disk that also hosts the SQL server for the target SharePoint installation. This can result in significant resource utilization, and will eliminate many of the benefits of parallelization.

Connection Adapters

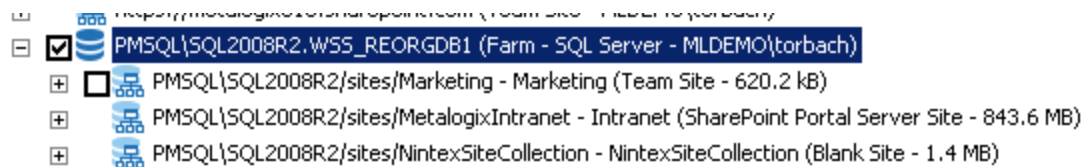
There are four primary adapter types available in Metalogix Content Matrix. They are defined below:

Database (DB)

The DB adapter is a **read-only** adapter designed for source side connections. If possible, **always** use the database adapter as the source for migrations, as it is the fastest adapter available in the application. It is also particularly useful for WAN based migrations or migrations to Office 365. This is because you can take database backups of your Content DBs, and move the resulting .bak files to a SQL Server on the target LAN (or Azure VMs in the case of O365 migrations), and then do a local migration.

DB connections look like this in the Content Matrix UI once connected:



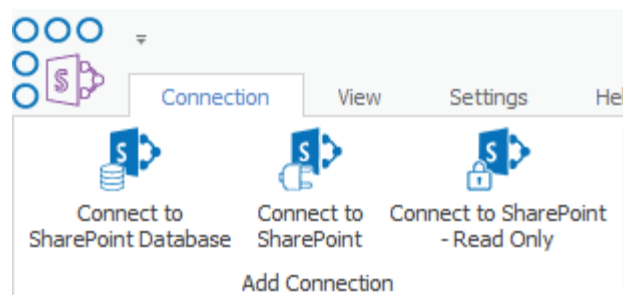
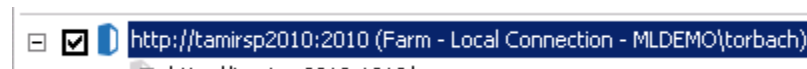


While the Database adapter does have some limitations, such as the inability to migrate Managed Metadata Services from one farm to another, users can use MEWS for that part of the migration, and the DB Adapter for the rest of the migration for best performance.

Local Object Model (OM)

The OM adapter is the second fastest adapter available in the application, and the fastest adapter for writing content. It can only be accessed when Content Matrix is directly installed on a WFE server in the SharePoint farm. This adapter is preferred if Content Matrix is installed on a WFE server in the **target** SharePoint farm (as of Version 8.3.0.1 of Content Matrix, this is only available in SharePoint 2013 and SharePoint 2016).

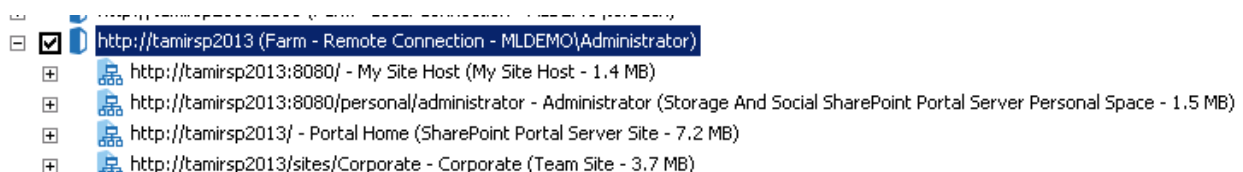
OM farm connections look like this in the Content Matrix UI once connected:



Metalogix Extensions Web Service (MEWS)

The MEWS adapter is a very fast adapter, but requires installation of a solution on the source or target SharePoint farm which it is being used in. The solution is included in the Content Matrix installer, and can be installed by running the installer on any WFE in the target SharePoint farm and selecting the "Install Content Matrix SharePoint WSPs" button, then checking the box to install the "SharePoint Extensions Web Service." If the target SharePoint farm will be connected to remotely, this adapter is preferred.

MEWS farm connections look like this in the Content Matrix UI once connected:



Note: The distinguishing factor between an OM connection and a MEWS connection is the “Local Connection” or “Remote Connection” string to the right of the connection.

Native Web Services and CSOM (NWS)

The NWS adapter should be avoided at all costs if optimizing for performance or functionality. Please utilize the DB adapter for source connections, and the OM or MEWS adapter for target connections wherever possible as defined above.

NWS connections look like this in the Content Matrix UI once connected (Note the globe behind the site icon. If this icon is showing on any of the currently connected nodes in Content Matrix, immediately attempt to remedy the problem by connecting to a different adapter type):

  <http://tamirsp2010/sites/contoso2> (MLDEMO\torbach)

Special Considerations for SharePoint Online Targets

Migrations from SharePoint on-premises (on-prem) to SharePoint Online (referenced as SPO in this document), have a number of significant factors that make them more complex and time consuming than on-premises to on-premises SharePoint migrations. These factors include:

- Functional and Application Programming Interface (API) differences between SharePoint on-prem and SPO that result in some on-prem elements not being easy or possible to migrate to SPO.
- Performance based limitations as a result of Microsoft’s protection mechanisms to ensure that no single SPO tenant/client can cause an entire multi-tenant farm to be unstable.
- Performance impact of the physical distance between a client’s on-prem SharePoint environment(s) and their SPO farm location.

This section explains why these limitations exist, and helps prepare Content Matrix clients to plan a migration to SPO. This includes:

- Listing and explaining the functional and API limitations and their significance on migrations
- Explaining the migration API leveraging Azure Storage to enhance the performance of content migrated to SPO
- Ensuring that customers allocate the appropriate time and consideration required when migrating their on-premises farms to SPO.

This document is essential reading for anyone migrating from SharePoint on-prem to SPO.

Functional and API Differences between SPO and SharePoint that impact migrations

Different Application Programming Interfaces available for SPO

For on-prem SharePoint, Microsoft offers a very rich API known as the SharePoint Server Object Model. This has been the most frequently used and richest API to access SharePoint on-prem.

However, due to the multi-tenant nature of SPO, Microsoft has to ensure that no single tenant can cause other tenants' environments to break. In light of that, Microsoft cannot expose the full SharePoint Object Model in SPO. They currently expose five API's that are useful for migrations:

- The Client Side Object Model (CSOM)
- The SharePoint REST API
- SharePoint Online Remote PowerShell (subset of CSOM)
- Office 365 Remote PowerShell (subset of CSOM)
- A Migration focused API that leverages Azure Blob Storage and batching to migrate list items and documents significantly more quickly to SPO (available both from CSOM and REST)

While the CSOM API is becoming close to the SharePoint Server Object Model in terms of feature parity, it is very slow for migrations. In light of that, Microsoft launched the new Migration API in May 2015. Today, the combination of the CSOM API and the new migration API is used by Content Matrix to write content to SPO. SPO and the CSOM API has some limitations that make Content Matrix and other migration tools unable to migrate all on-prem SharePoint content. For a full set of SPO limitations, see [this MS web page](#) listing. Another important page to look at is [SharePoint 2013 deprecated functionality](#). This applies to both SharePoint on-premises and SPO.

The bottom line is that our experience at Metalogix serving many clients who have migrated to SPO is that it is absolutely crucial to understand your SharePoint on-prem Inventory, whether any of it cannot be migrated to SPO, etc. before you start migrating to SPO. [Metalogix Expert](#) is a free tool that can help you identify some, but not all of the issues that you can expect.

Performance of Migrations to SPO with the new Migration API

There are 3 primary factors that until now have impacted performance of migrations to SPO dramatically:

1. The physical distance between your on-prem SharePoint farm and your SPO tenant, and the fact that in most cases, this Wide Area Network (WAN), is across the Internet, making it very unpredictable.

On-Premises SharePoint



SharePoint Online/
OneDrive For Business
(OD4B) Final Destination



Long distance between on-premises and Office 365

2. The protection mechanisms and throttling that Microsoft has in place to protect their multi-tenant clients from abuse by other tenants on the same farm.

On-Premises SharePoint



SharePoint Online/
OD4B Final Destination



- User and Farm based Throttling
- Virus Scanning
- Load Balancing of Web Server Connections
- Distributed Denial of Service Monitoring

3. The very chatty nature of the CSOM API that is the primary API used for migrations to SharePoint Online, compounded by the distance required for each API call to travel between on-prem environments and SPO.

On-Premises SharePoint



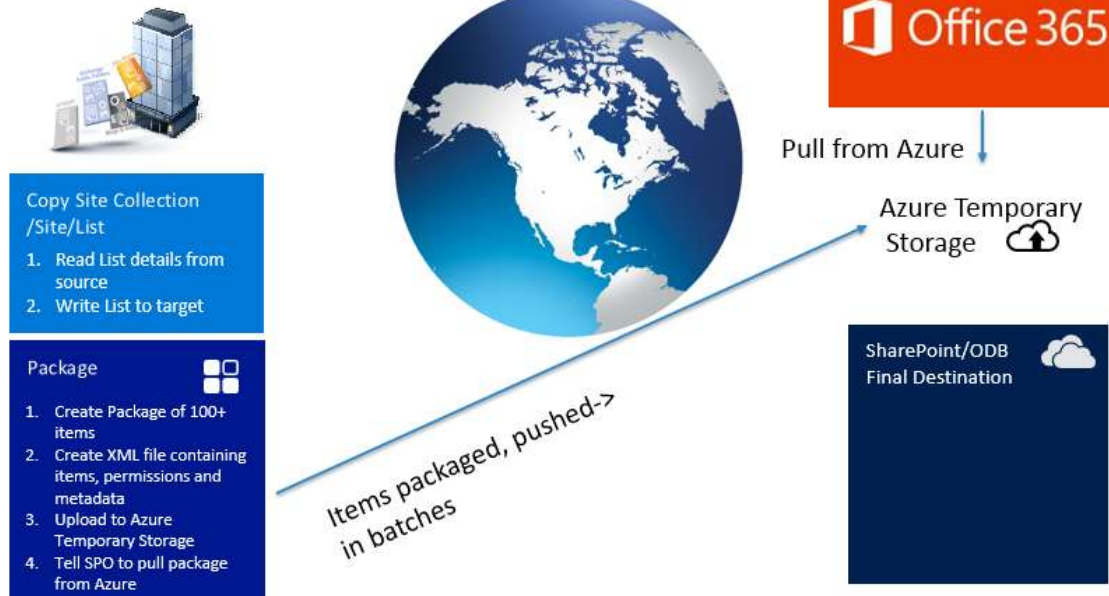
In order to alleviate the 2nd and 3rd aspects that slow down migrations, Microsoft has worked closely with Metalogix and other migration vendors to create a new Migration API that was introduced in May 2015 during Microsoft's Ignite conference.

The new API uses a combination of factors to alleviate the throttling and protection mechanisms and the chattiness of the CSOM API:

1. Site Collection structure, Site Structure, and List structure are still migrated using CSOM as they were before, and this happens prior to list items being migrated.
2. Instead of items being migrated with one API call at a time, they are now packaged locally into a batch with a corresponding manifest XML file to store permissions, metadata, etc. before being sent to SPO.
3. Instead of items/batches being sent directly to SPO, they are instead sent to an Azure Storage location (provided by either Microsoft or the user).
4. Once a batch has been successfully uploaded to the Azure Storage location, Content Matrix uses a CSOM request to ask SPO to pull content from the Azure Storage location into the previously created list in SPO.

Please see image below:

On-Premises SharePoint



The net result is a significant increase in performance of list items, that can be anywhere between 5 to 30 times faster than these migrations were before depending on a number of factors.

In light of this, Metalogix strongly recommends that our clients leverage the new Migration API for all SPO migrations. Users can either provide their own Azure Storage locations (as detailed later in this document), or leverage ones provided by Microsoft. While providing an account is the faster of the two options, it is also the only option with a cost.

Additional Best Practices unique to Content Matrix that can improve SPO migration performance even more

In addition to the above mentioned migration improvements, Content Matrix has a number of unique capabilities that make migrations to SPO potentially much faster than they are with the new Migration API, and we encourage clients who are doing large migrations (1TB or more) to take advantage of these capabilities:

1. Content Matrix has the unique ability to connect directly to SharePoint Content Databases for 99% of the migration from SharePoint on-prem to SPO.
2. This allows our clients to:
 - a. Backup their Content DBs on-premises
 - b. Copy their Content DB .bak files to an Azure VM running SQL Server in the same geographical region as their SPO tenancy (if the content DBs involved are very large, you

can also choose to ship physical disks to the Azure team, and that team will load them into your Azure VM).

- c. Restore the Content DBs to SQL Server in Azure VM.
- d. Use Content Matrix' "Connect to SharePoint Database" functionality to migrate directly from the Content DBs in the Azure VM to SPO:

On-Premises SharePoint



3. As per the above diagram, by moving the Content DBs to an Azure VM in the same geographical region as your SPO tenancy, you have now moved the bulk of the objects involved in the migration to what is effectively a LAN in the SPO geographical region. The Azure VMs are frequently in the same datacenter as the SPO infrastructure.
4. In addition to migrating directly from the Content DBs, Content Matrix allows you to:
 - a. Migrate many items in parallel without a steep reduction in performance. Content Matrix has significant parallelism and multi-threading capabilities that allow you for example to migrate 3 or 4 Site Collections at the same time from a single Azure VM, and then spin up additional Azure VMs in which you can migrate additional Site Collections in parallel.
 - b. Leverage the unique Distributed Migration capabilities of Content Matrix to distribute the load of migration across a multitude of different servers thereby increasing the level of parallelism.
 - c. Leverage significant automation capabilities with the ability to use PowerShell to control any operation available in the Content Matrix UI programmatically. This opens up many possibilities including the ability to create a list or CSV file that contains source Site Collections, Target Site Collections, Site Collection Job Configuration files, and Status, and then have the migration machine start the next available migration in this list. This is a recommended configuration in tandem with Distributed Migration to automate

migration actions across a multitude of different servers. Please reference the document “PowerShell Distributed Migration.pdf” for more information.

- i. An example of a list like this could be:

	Source Site Collection	Target Site Collection	Job Configuration File	Status
Finance	http://barnap3010.1813/	https://metalogssoftware.sharepoint.com/finance	Standard	Not Started
Marketing	http://barnap3010.205/sites/MetalogixIntranet	https://metalogssoftware.sharepoint.com/sites/MetalogixIntranet	Incremental	In Progress
Human	http://barnap3010.300/sites/finance	https://metalogssoftware.sharepoint.com/sites/finance	Standard with 3 year item filter	Not Started

Performance Expectations

There are a significant number of factors that will impact the performance of migrations using the new API. Many of those are beyond any migration vendor’s control. For example, the SPO tenancy has a service that runs to pull data from Azure Blob Storage accounts. If that service is being called on by many clients at the same time, it could significantly slow down migrations.

Another example is the workflow. While the new API is significantly faster for list items/documents, it is not faster for complex site collection/sub site structures with very complex and large Managed Metadata Trees, workflows, content types, etc. With that in mind, we’ve seen migration speeds vary between 2GB/hour and 40GB/hour depending on workload when using reasonable hardware for the migration machines involved.

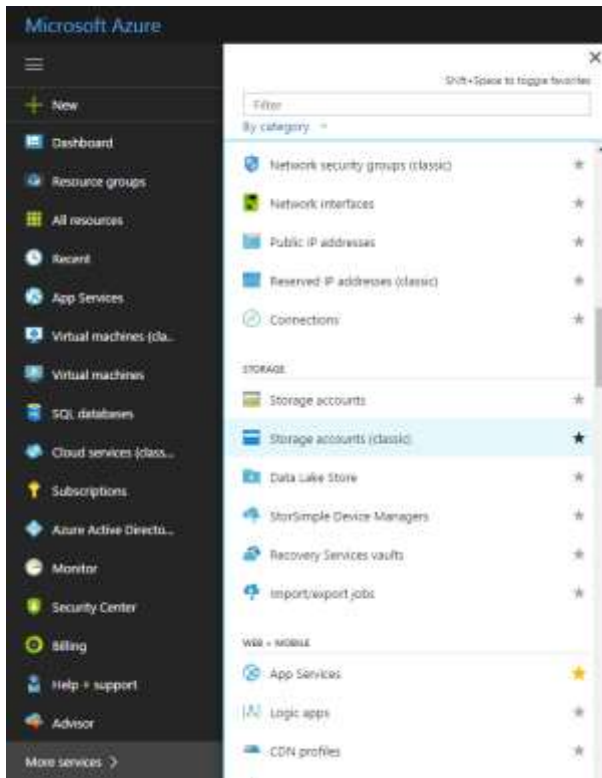
As compared to speeds of between 200MB and 550MB/hour on a single on-premises machine, and 400MB/hour to 10GB/hour on Azure VMs, this is a very significant increase. When trying to determine how long a project will take, we still recommend a conservative approach. Test several workload types using the new API, and preferable using Azure VMs, and see what you get. Run multiple jobs in parallel, and if you see no difference, and can afford the additional RAM/CPU, increase that, and then see what you get. Once you have those speeds, factor in that you will not maintain those speeds 24 hours/day due to variability in SPO. You will also have more remediation work to do post migration due to the differences between SP on-prem and SPO.

Recommendations

Setup an Azure Blob Storage Account

In order to use the new API, you can either provide your own Azure Storage location (for faster performance) or use one provided by Microsoft. This will detail how to generate your own once you have an Azure Login:

1. Log into the Azure portal by going to <https://manage.windowsazure.com>.
2. In the Azure Portal, choose the Storage accounts (classic) option:



3. Select “+ ADD”:

Storage accounts (classic)

Metalogix Software

+

+

Columns


Refresh

Subscriptions: Metalogix IT Production – Don't see a subscription? [Switch directories](#)

Filter by name...

0 items

NAME



No Storage accounts (classic) to display

Try changing your filters if you don't see what you're looking for. To learn more about the different services Azure has to offer, check out the [products listing on the Azure homepage](#). [Learn more](#)

Create Storage accounts (classic)

Create storage account

The cost of your storage account depends on the usage and the options you choose below. [Learn more](#)

Name

.core.windows.net

Deployment model

Resource manager

Classic

Account kind

General purpose

Performance

Standard

Premium

Replication

Read-access geo-redundant storage (RA-...

Subscription

Metalogix IT Production

Resource group

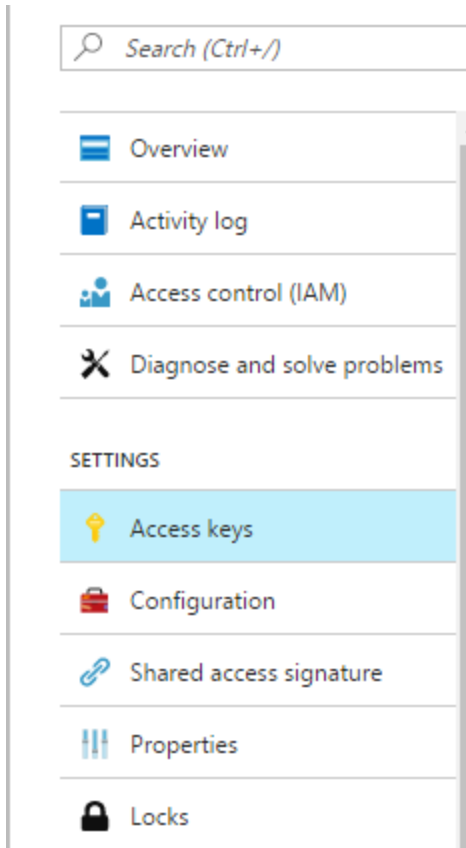
Create new

Use existing

Location

East Asia

- In the URL field, enter any name that can be verified.
- In the Location, make sure you put the location of your SPO tenancy. This is **critical** to achieve maximum performance. If you do not know the region your SPO tenancy is in, please contact your MS Technical Account Manager (TAM) if your company has one, or contact MS Support.
- For Replication, select "Locally Redundant" and then select "Create".
- Once the storage account has completed creation, select "Access Keys" from the settings section of the screen:



- Copy the Primary Connection String and store this for use later during configuration of Content Matrix:

NAME	KEY	CONNECTION STRING
Primary	BW5d4D4aQhMyOj8t8wK0CpQpTyy + z8t19t5d2D5d + E1U5yP9G3Z85H9q...	DefaultEndpointsProtocol=http;AccountName=primaryconnectiontest;AccountKey=Z...
Secondary	T8nTw8t9t4nRnUD + c3qy8t8k3d4q9g6 / D0w9yK0CpQpTyy1885 + M4dW...	DefaultEndpointsProtocol=http;AccountName=primaryconnectiontest;AccountKey=Tr...

- On the machine(s) where you have installed Content Matrix SharePoint Edition, open the file `c:\ProgramData\Metalogix\EnvironmentSettings.xml`, and scroll to the bottom of this file.

10. Find the XML parameter below, and populate the connection string from the “CONNECTION STRING” field in step 8:

```
<XmlableEntry>
  <Key>UploadManagerAzureStorageConnectionString</Key>
  <Value>DefaultEndpointsProtocol=https;AccountName=acme123;AccountKey=Tk9zp+o4NUnx
C7JCcdYzIU4qYq0BA6nvGs2YGFBTZARvtOrIO/RkzapBR7mMdlz9w2tmP/er5JfXGpLWUL0
JA==</Value>
</XmlableEntry>
```

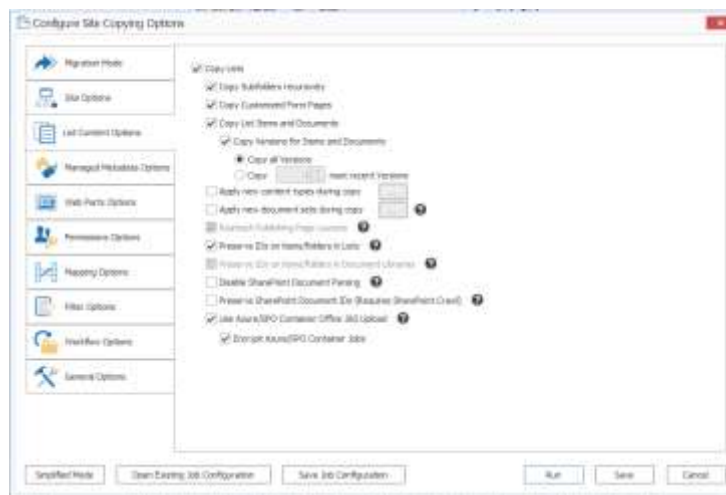
- Note some of the other parameters related to the use of Azure Blob Storage. You can set the batch size in number of items, or MB, and determine how many items or how many MB should be in each batch. For large lists with smaller items, we recommend starting with batch sizes of between 100 to 200 items. For smaller or larger lists with larger items, we recommend setting the parameter “UploadManagerDetermineBatchSizesByMB” to True, and then setting the

parameter “UploadManagerBatchSizeThresholdInMB” to between 300MB and 500MB. We will update this guidance as we learn more about what works best.

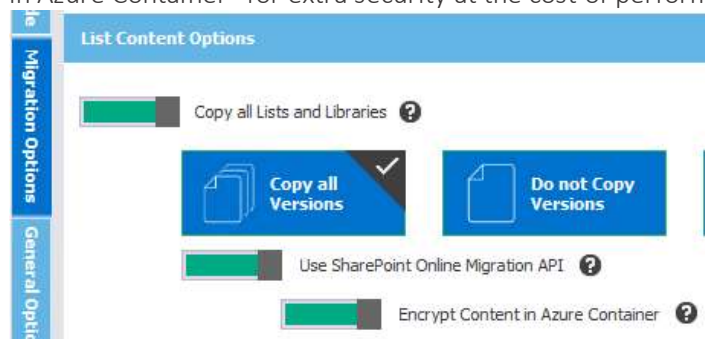
12. Find or add the parameter UploadManagerLocalTemporaryStorageLocation and set it to a location on your hard drive where processing of items and files can take place. It is important to have at least 40GB of free disk space for this processing. In the example below, I’m using C:\AzureAPI as the location for processing:

```
<XmlableEntry>
  <Key>UploadManagerLocalTemporaryStorageLocation</Key>
  <Value>C:\AzureAPI</Value>
</XmlableEntry>
```

13. Start Content Matrix as you normally would.
14. For any migration actions, in the List Content Options tab in Advanced Mode, select the “Use Azure/SPO Container Office 365 Upload” option. You may optionally choose to select “Encrypt Azure/SPO Container Jobs” for extra security at the cost of performance:



In Simplified Mode, under Migration Options and List Content Options, toggle the switch for “Use SharePoint Online Migration API”. You may optionally choose to select “Encrypt Content in Azure Container” for extra security at the cost of performance.



Use Microsoft Provided Containers

The Microsoft Provided Containers are a set of Azure Storage containers that are associated with a user’s tenant. This container is provided to a user at no cost from Microsoft and is associated with the

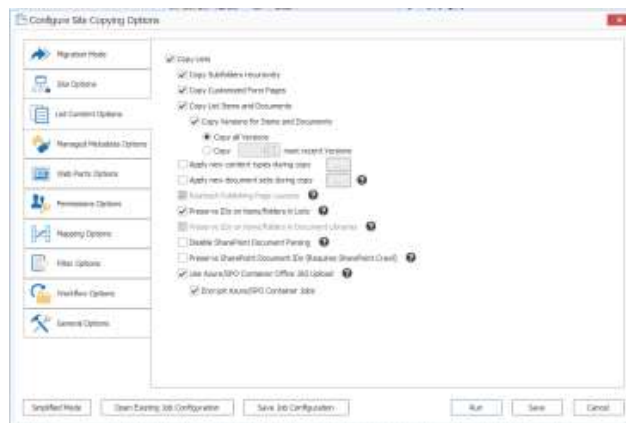
infrastructure the user's tenancy exists within. Content can only be uploaded to this location when it is also encrypted, making encryption a requirement for this action.

This action requires no work from the user outside of the selection of the appropriate checkboxes/toggles within Content Matrix.

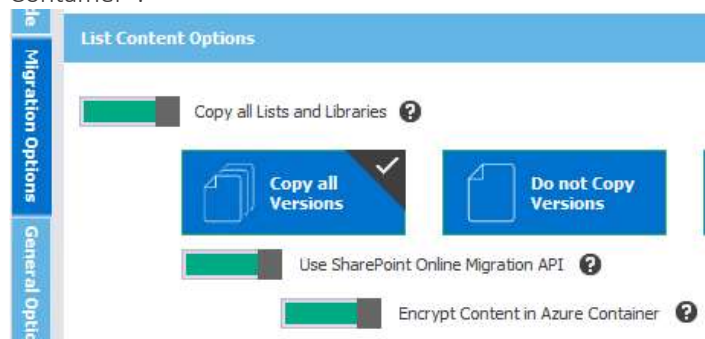
1. On the machine(s) where you have Content Matrix SharePoint Edition installed, open the file c:\ProgramData\Metalogix\EnvironmentSettings.xml, and scroll to the bottom of this file.
2. Find the XML parameter below and ensure the value of it is empty:

```
<XmlableEntry>
  <Key>UploadManagerAzureStorageConnectionString</Key>
  <Value />
</XmlableEntry>
```

3. For any migration actions, in the List Content Options tab in Advanced Mode, select both the "Use Azure/SPO Container Office 365 Upload" option and the "Encrypt Azure/SPO Container Jobs" option.



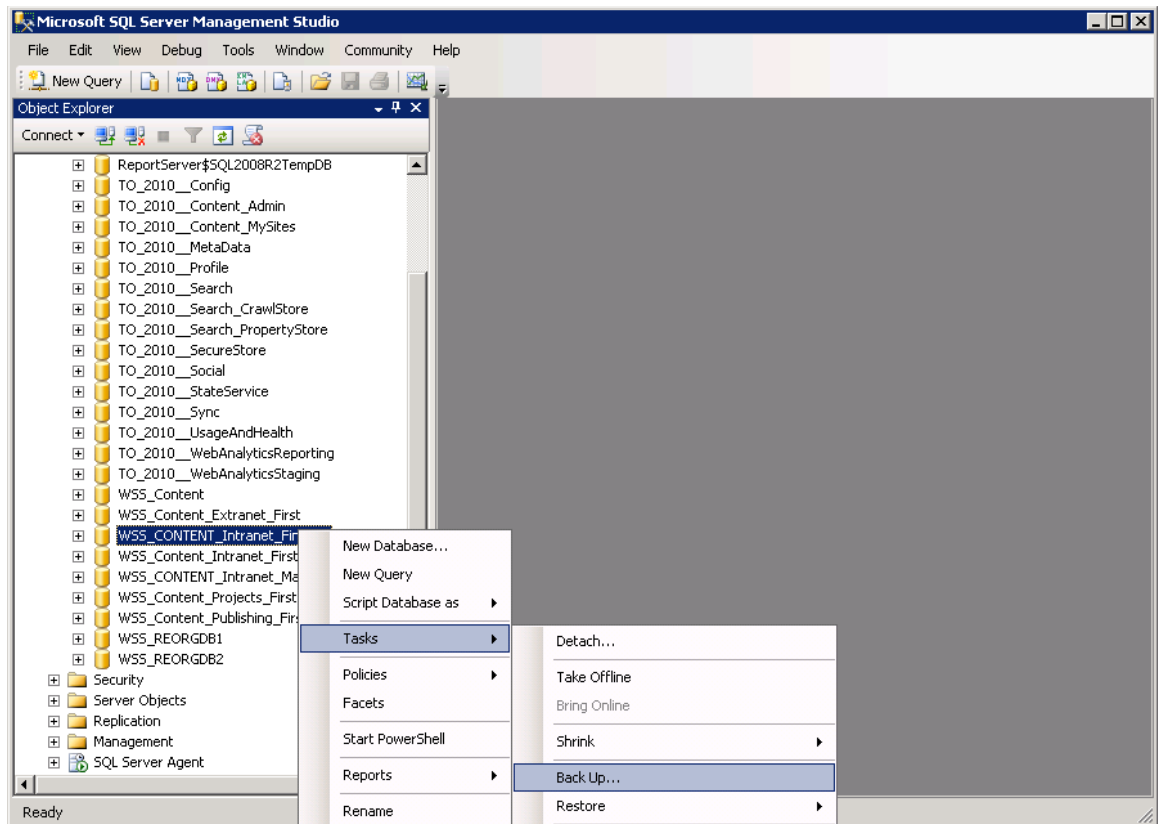
In Simplified Mode, under Migration Options and List Content Options, toggle both the switch for "Use SharePoint Online Migration API" and the switch for "Encrypt Content in Azure Container".



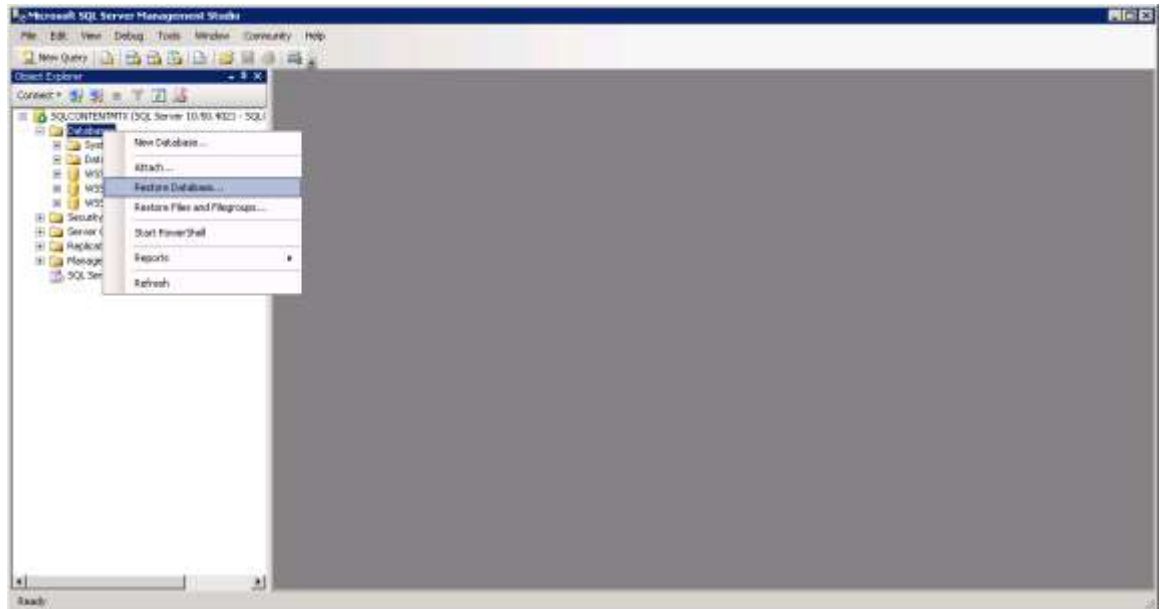
Use Azure Based Virtual Machines

As mentioned above, we have found that migrations from Azure Virtual Machines (VM) in the same region as the SPO instance being migrated to, increase performance of the migration. The process that we recommend for any migration to SPO is:

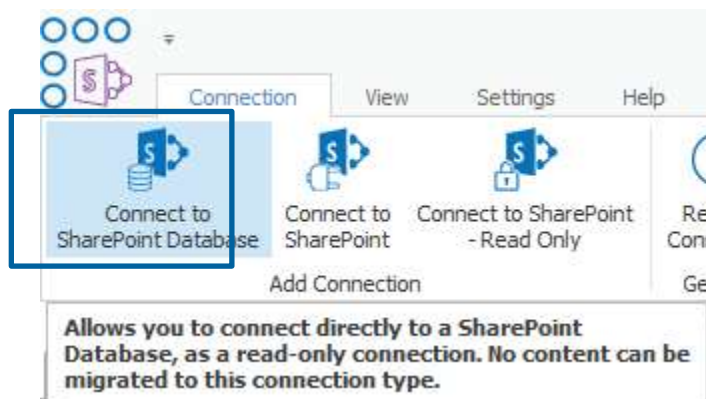
1. If you do not know the region your SPO tenancy is in, please contact your MS Technical Account Manager (TAM) if your company has one, or contact MS Support. If MS Support does not give you this information, please contact Metalogix, and we will get you the information.
2. Create an Azure VM in the same region as your O365 tenancy with Windows Server 2008 or 2012, and SQL Server 2008 or 2012 (please note that if you have the bits and rights to use SQL Server 2008 or 2012, it is cheaper to create a Windows only VM, and then install SQL Server on it).
3. Using SQL Server Management Studio Back up your source Content DBs to .bak files:



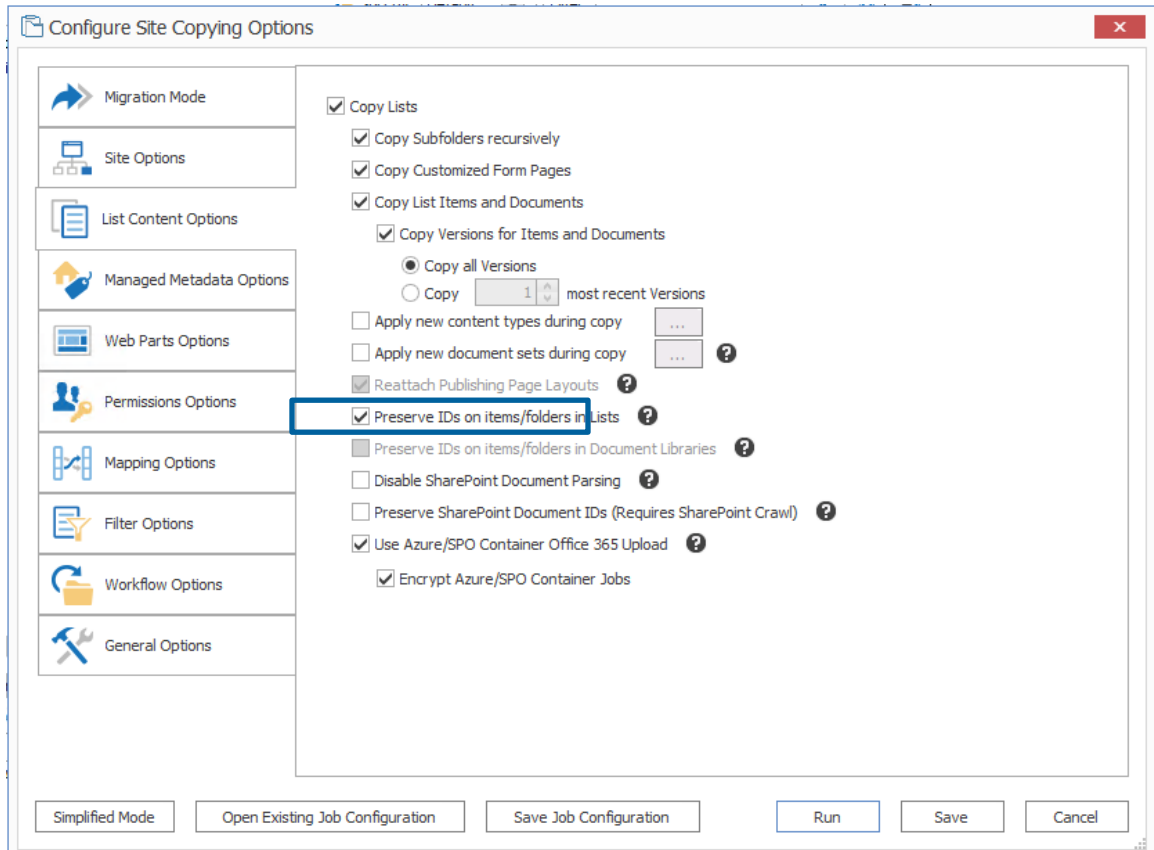
4. Use ftp, or other mechanism to copy the .bak file to your Azure VM. You can ship a physical hard disk to the Azure team if needed. For more information how to ship a physical hard disk, please see Microsoft's guidance [here](#).
5. In your Azure VM, restore the .bak files to a new databases:



6. Once these databases have been restored, give your Azure VM admin dbo access to each of these databases
7. Install Content Matrix SharePoint Edition Console on the Azure VM (you can install the console on multiple other Azure VMs that do not have SQL Server on them, but must be in the same internal network in order to enable parallel, multi username actions).
8. In Content Matrix, use the Connect to SharePoint Database option:



9. Enter the name of the database server (at this point most likely (local))
10. Choose the Content DB you just restored, and the Site Collection within that Content DB that you would like to connect to.
11. Migrate content as you normally would with Content Matrix
12. For any operations that require farm level connections (such as copying the Managed Metadata Services Term Store), migrate that from the source farm instead of Azure.
13. For the final Incremental Copy, there is no need to re-backup your Content DBs and restore them in Azure. You can perform an incremental copy from the actual source as long as you selected 'Preserve Item ID' in Advanced Mode for all List copy or Site copy operations that include lists:



Configure Site Copying Options

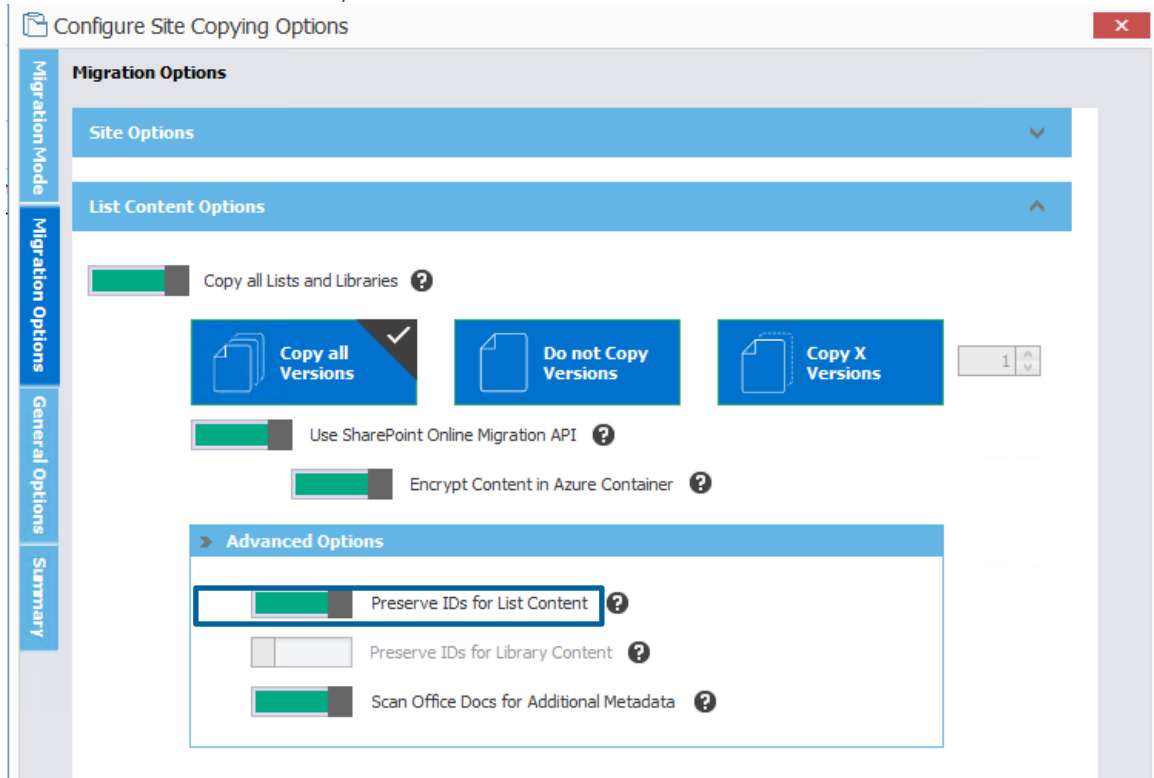
- Migration Mode
- Site Options
- List Content Options
- Managed Metadata Options
- Web Parts Options
- Permissions Options
- Mapping Options
- Filter Options
- Workflow Options
- General Options

Copy Lists

- ☒ Copy Lists
- ☒ Copy Subfolders recursively
- ☒ Copy Customized Form Pages
- ☒ Copy List Items and Documents
- ☒ Copy Versions for Items and Documents
 - ☒ Copy all Versions
 - ☐ Copy most recent Versions
- ☐ Apply new content types during copy
- ☐ Apply new document sets during copy
- ☒ Reattach Publishing Page Layouts
- ☒ Preserve IDs on items/folders in Lists
- ☐ Preserve IDs on items/folders in Document Libraries
- ☐ Disable SharePoint Document Parsing
- ☐ Preserve SharePoint Document IDs (Requires SharePoint Crawl)
- ☒ Use Azure/SPO Container Office 365 Upload
- ☒ Encrypt Azure/SPO Container Jobs

Simplified Mode | Open Existing Job Configuration | Save Job Configuration | Run | Save | Cancel

If using Simplified Mode toggle “Preserve IDs for List Content” in the Advanced Options section of the List Content Options tab:



Configure Site Copying Options

Migration Options

- Site Options
- List Content Options

Copy all Lists and Libraries

Copy all Versions | **Do not Copy Versions** | **Copy X Versions**

Use SharePoint Online Migration API

Encrypt Content in Azure Container

Advanced Options

- ☒ Preserve IDs for List Content
- ☐ Preserve IDs for Library Content
- ☒ Scan Office Docs for Additional Metadata

Use Azure Active Directory Setup

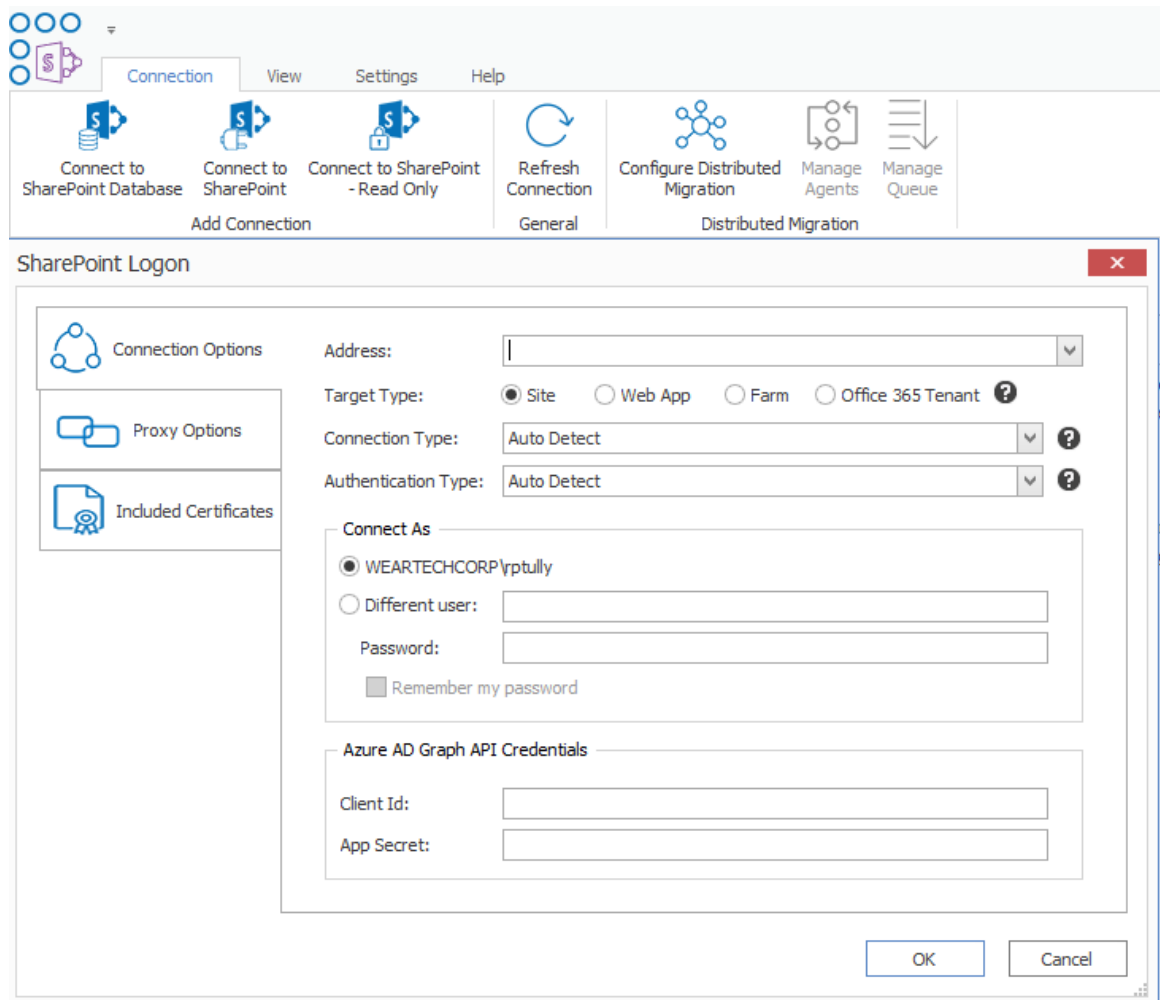
For a very small performance gain (potentially negligible), Content Matrix SharePoint Edition can communicate using a Client ID and Application Secret when connecting to SPO. This requires the following steps:

1. Register your application with Azure AD. Instructions on how to do so can be found [here](#).
2. On the machine(s) where you have Content Matrix SharePoint Edition installed, open the file c:\ProgramData\Metalogix\EnvironmentSettings.xml, and scroll to the bottom of this file.
3. Find the XML parameter below set the value to "Graph":

```
<XmlableEntry>  
<Key>ResolvePrincipalsMethod</Key>  
<Value>Graph</Value>  
</XmlableEntry>
```

Note: the only options of this parameter are "Graph" and "People". "People" is the default behavior.

4. Connect to SharePoint using the "Connect to SharePoint" dialog. The screen will have additional fields. Enter the predefined Client Id and App Secret from Step One into these fields and connect as normal:



To achieve performance levels of 10GB/hour or higher

1. Create multiple migration users in your SPO tenancy, each with Global Admin permissions (MS recently rolled out SPO Admin roles. If those are available in your tenancy, you should be able to use those instead of Global Admin roles).
2. Spin up multiple Azure VMs in the same domain as the SQL Server mentioned above.
3. Ensure that the SQL Server VM has sufficient RAM to deal with multiple simultaneous jobs.
4. Plan your migration so that each VM is migrating several jobs at a time, with larger Site Collections or Sites being migrated on separate machines (i.e.. Balance the workload across multiple VMs).
5. To achieve optimal performance, first migrate the structure of your SharePoint farm, and then the content.
6. Use Distributed Migration to shift the load across multiple VMs.

7. Use PowerShell to control migration jobs. This allows you to programmatically control migrations, check against an Excel file, DB, or SharePoint list for a list of jobs that are still required, and perform migrations in the order you would like without manning the consoles frequently.
8. PowerShell also uses far fewer UI resources, and therefore results in faster migrations. PowerShell is also able to take advantage of the Distributed Migration features.

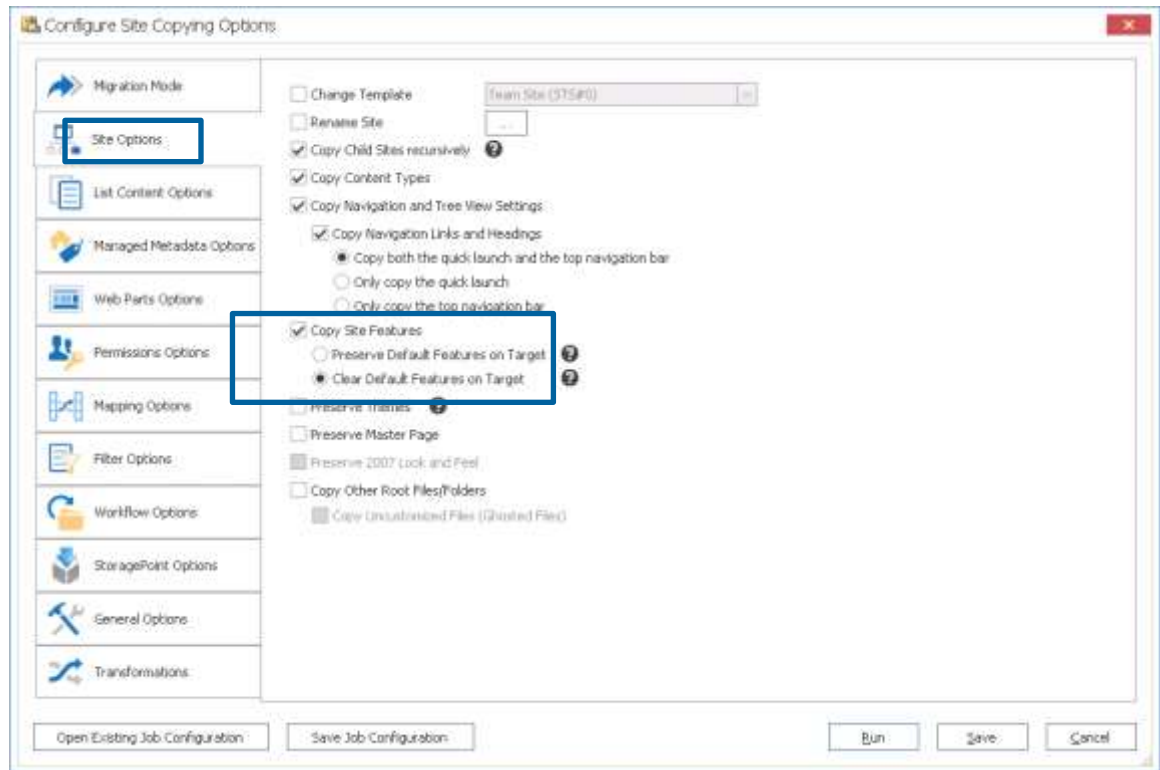
Disabling Unnecessary Features

The following options can significantly increase the time taken to migrate content, and should be turned off whenever possible:

- Verbose Logging
 - **Location:** In the configuration dialog, under the "General Options" tab. A checkbox labeled "Verbose".
 - **Function:** Turns on verbose logging. Is unnecessary unless full XML of all objects is required in the job listing.
- Item level permissions
 - **Location:** In the configuration dialog, under the "Permission Options" tab. A checkbox labeled "Copy Item Permissions"
 - **Function:** Copies item level permissions. As a best practice these objects should never be copied unless absolutely necessary.
- Copy form web parts
 - **Location:** Under the "Web Parts Options" tab, a checkbox labeled "Copy web parts on form pages".
 - **Function:** Copies web parts on form pages. In most cases these forms are not customized and this option will simply add checking overhead.

Following Content (SPO only)

- This feature is enabled by default when a new site (but not site collections) is created. When it is disabled, performance of Site copies improves by approximately 20% to 25% for document heavy workloads.
- Since Content Matrix does not currently give you the option of turning off only a particular feature on the target site, the only way to achieve this currently is to specify in the Site Options tab of the Paste Configuration screen to 'Clear Default Features on Target':



Multithreading and Batch Jobs

Multithreading in Content Matrix can be configured in two places.

- Multithreading within a single job.
- Multithreading in the form of multiple jobs started concurrently.

Jobs can additionally be batched together to more efficiently migrate larger sets of data.

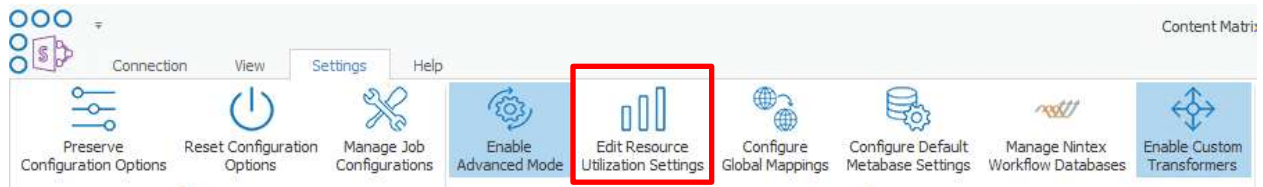
Note: This is more efficient when using Distributed Migration and multithreading.

Single Jobs

Within a single job, Content Matrix can utilize multithreading to copy site data concurrently.

Configuring Single Job Multithreading

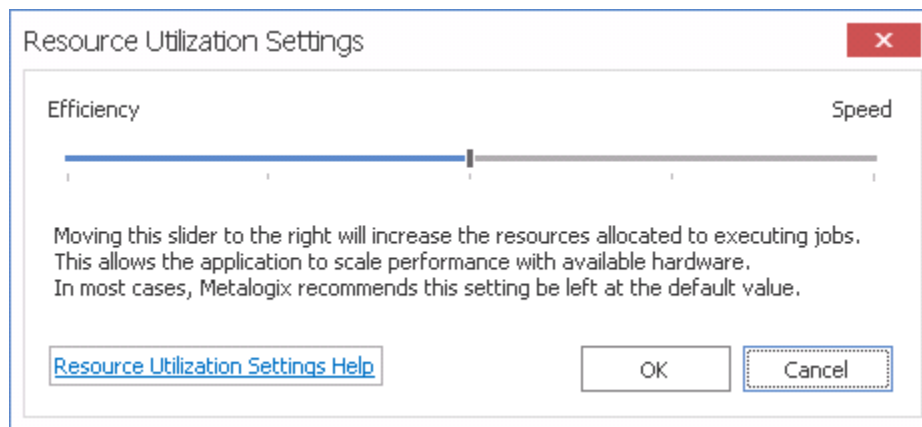
How many threads can be used simultaneously per job in Content Matrix is controlled by the "Edit Resource Utilization Settings" button within the "Settings" ribbon at the top of the Console UI



If clicked, a dialog containing a slider will appear that will allow control of the number of threads available within a single action. If the slider is moved all the way to the left, this will turn off multithreading and only allow a single thread to be used during the action.

Note: Turning off multithreading in this way can be a valuable way of troubleshooting whether multithreading is causing issues.

If the slider is left in the middle, it will allow twice the number of processors in threads to be created per action. For example, if the machine running Content Matrix has a two core CPU, four threads will be able to be used if the slider is in the below state:



Moving the slider farther to the right will allow more threads to be used, but can potentially overwhelm system resources.

Optimizing Single Job Multithreading

Content Matrix will only spawn new threads when there are multiple SharePoint site nodes to be migrated in the current location of the job. This means that in order to optimize single job multithreading, multiple sites should be migrated within a single job.

Concurrent Jobs

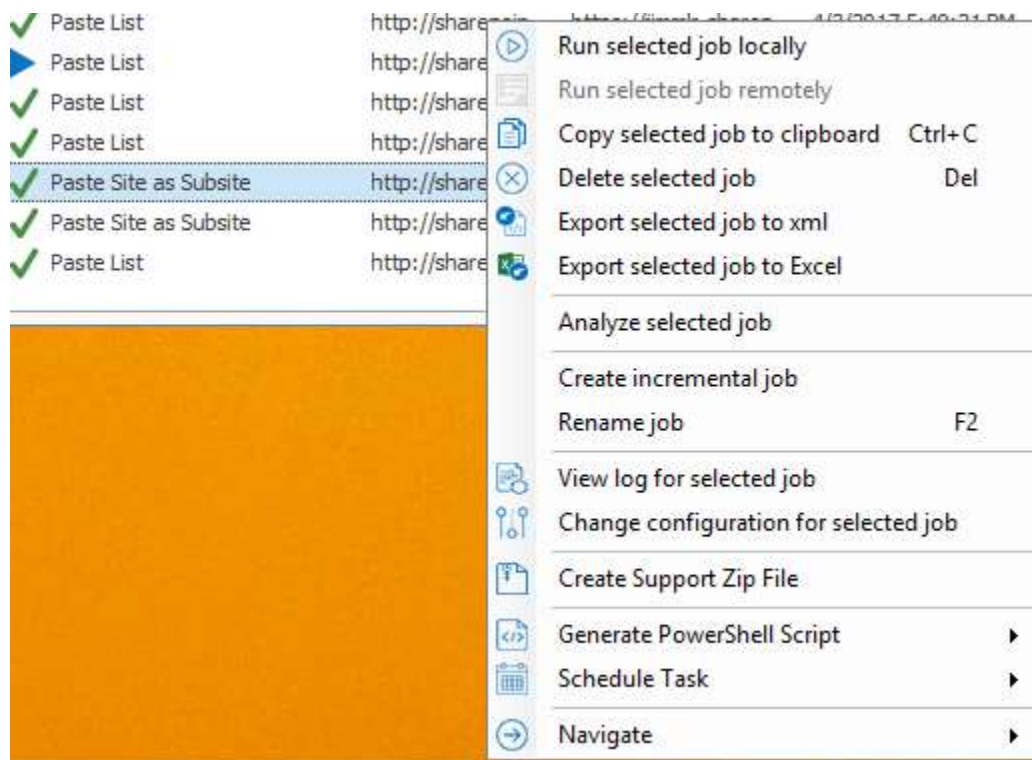
Content Matrix has the ability to concurrently run jobs.

Configuring Concurrent Jobs

To achieve this, pre-configure two different migrations. In the job listing, there should be two separate entries, as displayed below:

Paste Site as Subsite	http://sharepoin...	https://jimmlr.sharep...	4/7/2017 6:10:26 PM	Done	Lists Copied: 10, Items Copie...	Bytes Copied: 44.3 MB
Paste Site as Subsite	http://sharepoin...	https://jimmlr.sharep...	4/8/2017 12:08:34 AM	Done	Lists Copied: 10, Items Copie...	Bytes Copied: 44.2 MB

To run these jobs concurrently, for each job, right click and choose "Run selected job locally"



After both jobs have been run separately, there will be two status dialogs tracking the status of their respective jobs. You also have the ability to schedule these jobs to run at the same time later in the day by multi-selecting the jobs, right clicking and selecting 'Schedule Task'.

Note: An alternate method of running two jobs concurrently would be to generate a PowerShell script for each job and run the scripts simultaneously, or to leverage Distributed Migration.

Optimizing Concurrent Jobs

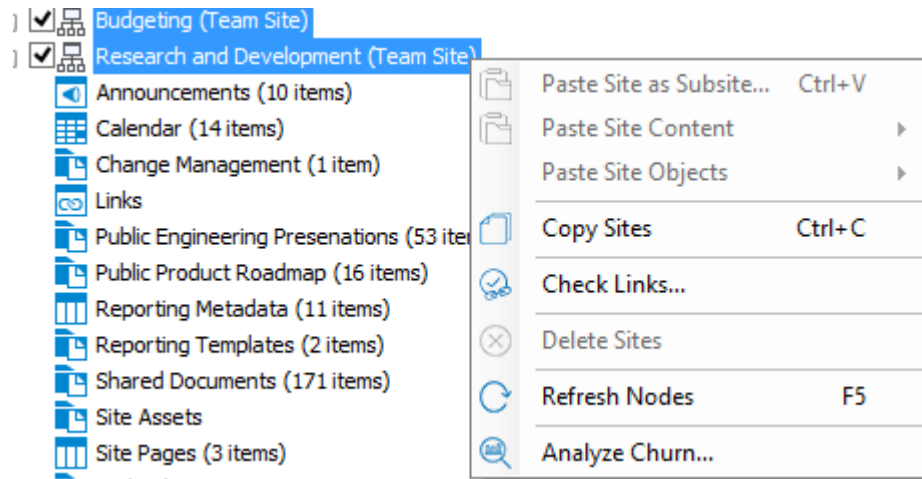
To ensure data integrity, it is important to enforce the following rules when running concurrent jobs:

- Multiple jobs should never migrate data into the same list simultaneously
- Multiple jobs should take care to avoid migrating from the same source data simultaneously

Batched Jobs

Jobs can be batched into groups, and then run. There are two primary ways of batching data into runnable jobs.

- Multi-selecting same-type nodes (sites can only be multi selected with other sites, lists with other lists, etc.)



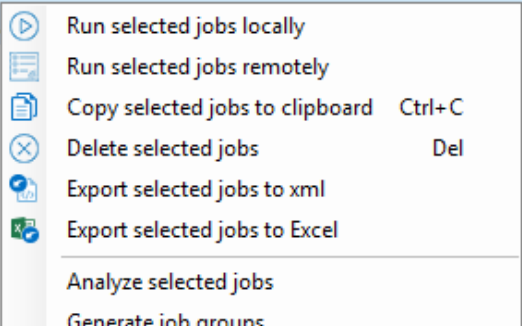
- Multi-selecting previously configured (or run) jobs

Paste List	http://sharepoin...	https://jimmlr.sharep...	4/3/2017 10:11:52 AM	Aborted	Run selected jobs locally
Paste List	http://sharepoin...	https://jimmlr.sharep...	4/3/2017 10:19:10 AM	Done	Run selected jobs remotely
Paste List	http://sharepoin...	https://jimmlr.sharep...	4/3/2017 11:08:40 AM	Done	Copy selected jobs to clipboard Ctrl+C
Paste List	http://sharepoin...	https://jimmlr.sharep...	4/3/2017 1:30:11 PM	Done	Delete selected jobs Del

When batching data in either fashion, migration of nodes within the batch will be run sequentially. This means that if two sites are selected, they will be migrated in order. Similarly, selecting and running multiple jobs will run those jobs from top to bottom.

Batched jobs can be run concurrently with other batched jobs. Previously defined settings on multithreading and concurrent jobs also apply to all jobs run in a batch.

Job Name	Source	Target Container	Started	Status	Log Si
Paste Site Collection	http://weartech...	https://metalogixsof...			
Paste Site Collection	http://weartech...	https://metalogixsof...			

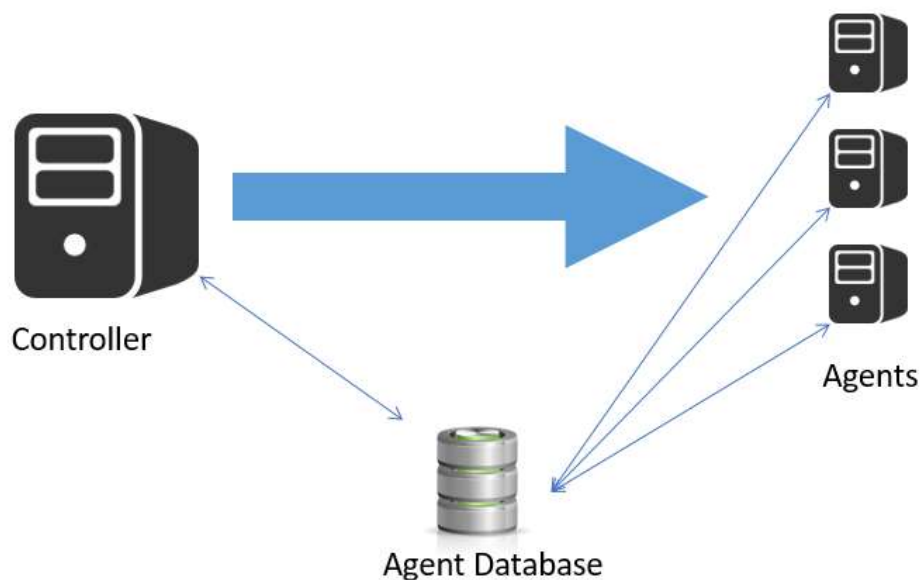


Distributed Migration

Content Matrix is able to leverage a unique feature called Distributed Migration in order to shift the load of multiple jobs to multiple machines using an Agent-Controller model for content distribution.

In the Agent-Controller model, one installation functions as the primary source for the distribution of migration jobs across multiple machines. The Controller does not (by default) participate in the migration process and only functions as a conduit for generating, configuring, and distributing jobs across the multitude of Agents. This does not prevent the Controller from executing jobs locally and independently.

On the Controller a Configuration Wizard will guide through the appropriate steps to generate an architecture similar to below:



In this architecture, there is a shared database across all the listed machines. This database is known as the "Agent Database" and contains a shared list of all jobs configured for migration in this model, as well as a listing of all Agents and (if configured) a shared grouping of settings for all consoles.

In the previously picture scenario, the Controller generates all jobs and writes these into the Agent Database. The Controller then queues up the listed jobs, identifies any Agents that are not currently performing a migration, and pushes that job to that Agent. Should there be no free Agents, jobs will remain queued until an Agent becomes available. As an Agent migrates content it will write job information back to the Agent Database. This information can then be viewed from the context of the Controller.

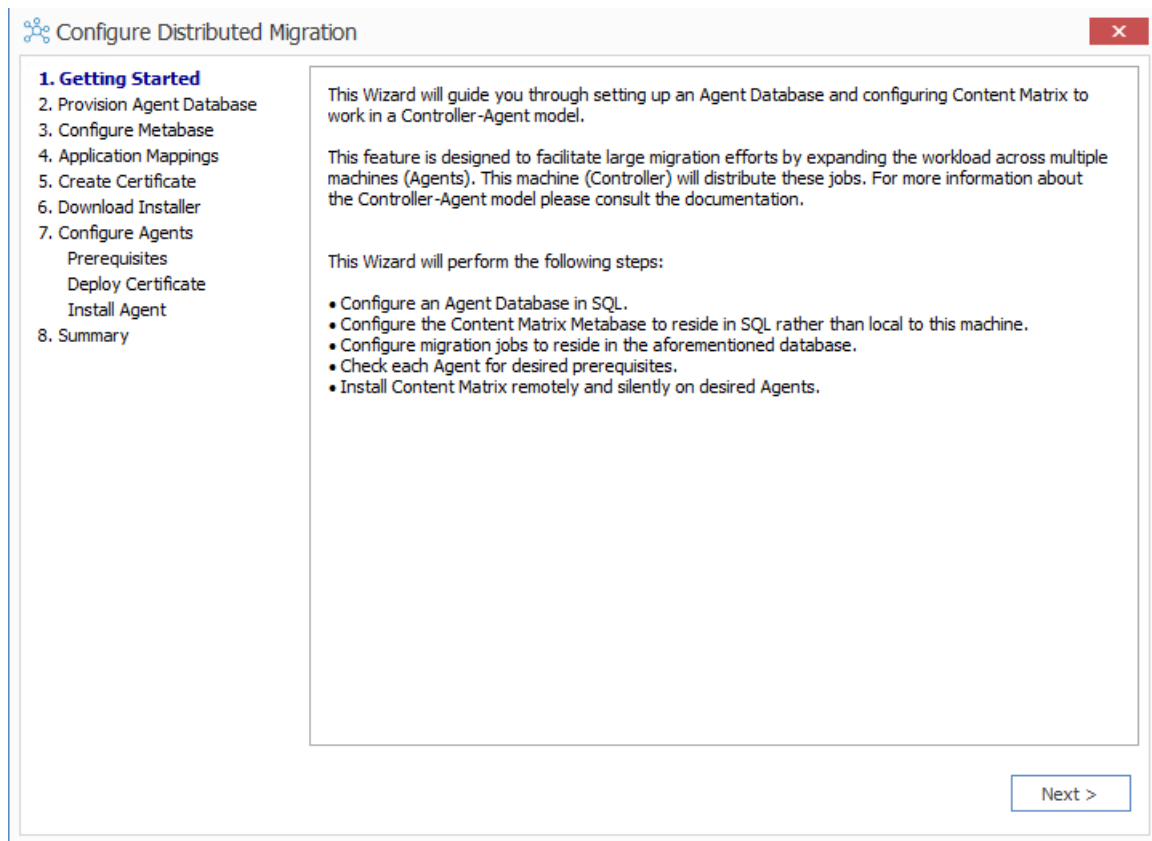
Configuration

The following steps should be followed for leveraging Distributed Migration appropriately:

- Launch Content Matrix SharePoint Edition.
- Within the top ribbon of the console is a button labeled “Configure Distributed Migration”:



- Upon clicking that button, a Wizard will pop-up walking the user through the configuration. The Wizard will provide information for all actions being performed. Please consult the documentation should there be concerns with any particular screen or configuration.



PowerShell

As PowerShell extensibility is one of the major tenants of Content Matrix, that extensibility has been brought to the concept of Distributed Migration. Please reference the document “PowerShell Distributed Migration” for further details of how to leverage PowerShell for an Agent-Controller model.

PowerShell

Metalogix Content Matrix has the ability to run jobs from a PowerShell context. Running jobs in this way provides some notable advantages:

- Minimizes resource consumption (no requirements to render or cache the UI)
- Separate job lists can be used for each distinct job (job lists are the storage medium for jobs, usually a SQL CE database)

Configuring PowerShell

To take advantage of PowerShell functionality in Content Matrix, simply configure a job via the UI in the normal fashion, and click the “Save” button at the bottom of the configuration dialog.

This will create a configuration entry for the job in the job listing that can be used to generate a PowerShell script. To generate a script, highlight the desired job and click the “Generate PowerShell Script” button at the top of the job listing, or right click and select Generate PowerShell:



Job Name	Source	Target Container	Started	Status	Log Summary	Data Migrated	Finished
Paste Site Collection	http://weartech...	https://metalogixof...				Bytes Copied: 0 B	
Paste Site Collection	http://weartech...	https://metalogixof...				Bytes Copied: 0 B	

This will create a script of the configured job that can be copied and pasted into an existing script, or saved or run as desired.

Note: It is more efficient to pre-configure jobs in the UI instead of trying to write them initially in PowerShell. This will allow configuration of desired job functionality without having to learn all of the PowerShell parameters for the Metalogix actions.

Optimizing PowerShell

Optimization of Metalogix PowerShell cmdlets:

- Use the `-quiet` parameter
 - This parameter will disable all visual logging to the PowerShell console (all logs are still tracked in the job listing), and will increase the speed of the migration significantly.
- Use the `-joblist` parameter to specify multiple job listings
 - This parameter will specify which job list to write log entries to when the PowerShell job is running. Providing multiple job lists for logging purposes will help to spread load and decrease bottlenecks during the migration.

Note: You can configure PowerShell to leverage Distributed Migration. Please see “PowerShell Distributed Migration” for appropriate usage.

