10 things to consider when planning your Mule 4 migration

a guide by Prathima Sanka and Sreekanth Cherukuri

+44 20 3475 7980 | UK
+1 646 568 9786 | US
+91 40 4265 6565 | India
info@whishworks.com
www.whishworks.com

CONTENTS

INTRODUCTION

Mule runtime is the engine for Anypoint Platform, combining data and application integration across legacy systems, SaaS applications, and APIs. Mule 4 is the latest and most advanced version of Mule runtime engine and has been designed to boost scalability and support performance optimisation and smooth upgrade.

According to MuleSoft, Mule 4 reduces management complexity and accelerates application development, with up to 50% fewer steps and concepts to learn.

At high level, the main advantages of Mule 4 are simplicity, performance and extensibility:

Simplicity: Implementing integrations with Mule 4 is simpler and faster. It takes less time and effort for projects to go live, resulting in reduced overall project costs.

Performance: Mule 4 runtime and respective connectors perform better both in terms of speed and sizing.

Extensibility: Compared to previous versions, connectors in Mule 4 are completely decoupled, which means that if you want to upgrade your connectors you don’t have to wait for your runtime engine to be upgraded.

WHEN TO MIGRATE

Better performance, faster pace of delivery and continued support

There can be many reasons for you to start considering upgrading to the newest version of Mule:

- The Mule version you're using reaches End-of-Support or End-of-Life
- You want to make significant updates to the existing applications
- You want to take advantage of key Mule 4 capabilities
- You decide to upgrade all of your apps to Mule 4 so that they are on one version

If you are on an older version, you are probably missing out on many of the new features introduced with Mule 4. Most importantly you may be facing significant impediments if your version has either reached, or is close to reaching, End-of-Support or End-of-Life (Table 1).

Table 1: Mule runtime End-of-Support / End-of-Life deadlines

	Standard Support	**Extended Support**	**End-of-Life Support (only for on premise)**
Mule 3.6	January 15, 2015	January 15, 2017	N/A
Mule 3.7	July 9, 2015	November 16, 2017	November 16, 2019
Mule 3.8	November 16, 2018	November 16, 2021	After November 16, 2021
Mule 3.9	March 20, 2021	March 20, 2024	After March 20, 2024

Note: *MuleSoft may change the deadlines for any of these versions, so make sure you check the status on the respective page: https://www.mulesoft.com/legal/versioning-back-support-policy#end-of-life-support*

WHAT CHANGES IN MULE 4

Easier to manage, greater automation, increased intuitiveness

Feature / Characteristic	Mule 3.x	Mule 4	Advantages
Runtime	**Not so smart**, explicit runtime tuning available depending on the scenario.	**Smart** – self-tuning, high performance, leverages the complete advantages of non-blocking I/O.	- Removes manual overhead and error-prone explicit runtime tuning configuration. - No need to explicitly configure thread pool. Runtime does that for you, resulting in optimum performance.
Message structure	**Complex message** structure with inbound, outbound properties and payload.	**Simplified message** model with payload and attributes.	- Easy access to a message's data and metadata. - Simplified message model allows users to update messages making it easier to send and receive data across connectors.
Exception / Error handling	The only way to deal with errors is by handling **java exceptions.**	**Improved error handling** with on-error-continue and on-error-propagate. Simplified with built-in errors and predefined errors, specific to connector operations. New Try Scope to handle errors specific to processor.	- Easy to handle errors. - With on-error-continue and on-error-propagate used in Try Scope, we can decide whether to continue the process or break out. - Improved readability of code with a meaningful name for error types (example : DB:QUERY_EXECUTION)
Connectors	**Tightly coupled** to runtime, both transport and operation oriented.	**Independent of runtime** with the introduction of isolated class loading, completely operation oriented.	- Connectors can be upgraded irrespective of the runtime. - Completely operation oriented improves developer experience with consistency and predictability of operations across connectors.

Feature / Characteristic	Mule 3.x	Mule 4	Advantages
Weaving payload	Can be updated by using Mule Expression Language, Dataweave, scripting components (using Javascript, Python, and Ruby) and **direct binding** with Java class implementing callable message.	Payload can only be **overwritten** rather than being partially updated.	- Immutability of payload and no direct binding with java class implementing callable interface, leads in improved performance.
Data handling	**Use both languages** - Mule Expression Language (MEL) and Dataweave 1.0.	**Dataweave 2.0** replaces both languages	- Easier to learn. - Completely function oriented. - Powerful with more built-in functions grouped into modules, which can then be imported into other flows. - Can package custom reusable functions into a custom module and use it across applications.
Streaming	Explicit **manual configuration** required to enable or disable streaming, non-repeatable streams.	Streaming concept is completely **abstracted and automatically handled** by runtime with the introduction of repeatable streams.	- No explicit, error prone configurations. - Very helpful when dealing with large payloads.
Enriching events	Need to use message enricher component.	Can be achieved by **configuring target variable** at the connector and component level	- Easier to use.

10 things to consider
Before, during and after the migration

BEFORE MIGRATION

Extensive assessment to estimate impact & effort

Where to begin?

Review your current installation to understand your obsolescence risks first

To understand your obsolescence risk, begin by conducting a lifecycle analysis of your components, modules and connectors. A thorough review of this information will help you understand and prioritise your modernisation needs and goals.

Therefore the first thing you need to do before migrating to Mule 4 is to assess and plan the migration.

1. ENVIRONMENT

The effort and cost required to migrate to Mule 4 would depend on the setup of your existing applications.

Starting with the environment, if you are on CloudHub then it is pretty straight forward, as CloudHub is Mule 4 ready. You don't need to assess anything; you just have to migrate and deploy your applications.

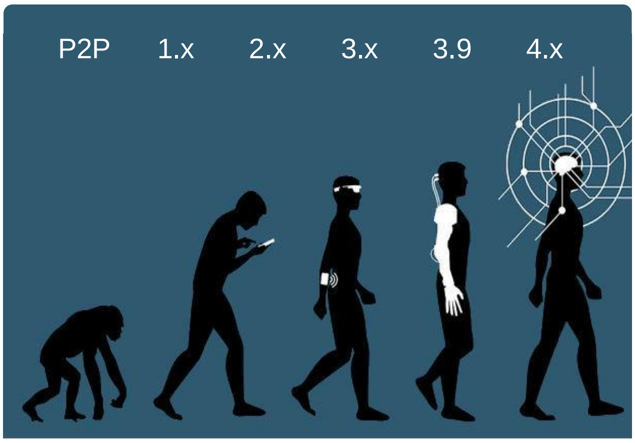
If you have an on-premise or hybrid environment, then you need to analyse the environment as well as the applications that need to be deployed. When doing so, you also need to consider the time and effort required to decommission your, now obsolete, applications.

Deployment Mechanism	Instruction
CloudHub	CloudHub is Mule 4 ready.
Hybrid Deployment	You must install the standalone version of Mule 4 to your servers.
Anypoint Platform Private Cloud Edition (PCE)	PCE supports Mule runtime, version 4.
Anypoint Platform for Pivotal Cloud Foundry (PCF)	PCF does not support Mule runtime, version 4.

2. CURRENT VERSION(S) AND SKILLSETS

As most of the components have been changed over the various Mule 3.x versions, the older the version you have the more effort will be required for the migration. Migrating from an older version of Mule, like 2.x or 3.5, would take more effort compared to 3.8 or 3.9.

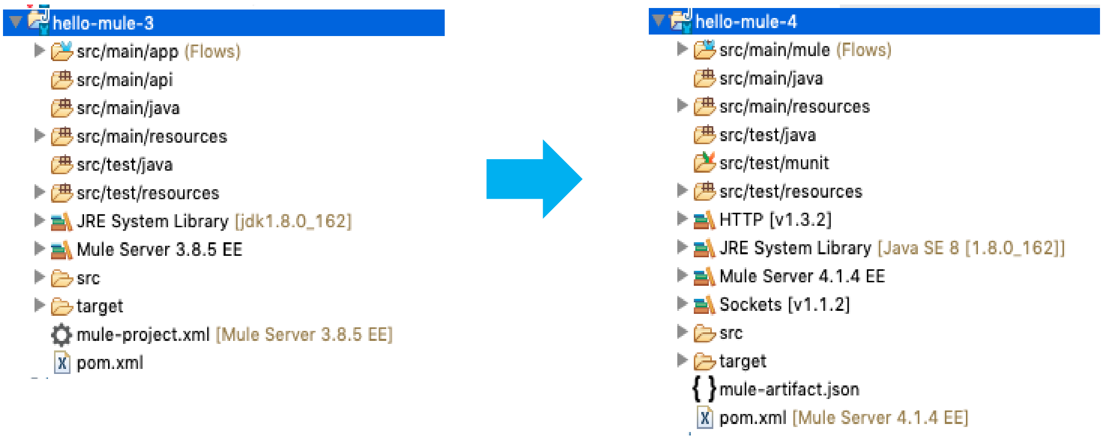
Depending on the estimated complexity, effort and criticality of your current environment, you may want to start by migrating only some applications to Mule 4. Then you can gradually continue to migrate everything else.



Another important factor at this stage is to plan for upskilling your in-house team to Mule 4, and/or ensuring that your professional services partner (if you are using one) is experienced with certified resources in the new version. MuleSoft is already providing a set of certifications for Mule 4.

3. NON-MAVEN PROJECTS NEED TO BE ‘MAVENISED’

There are two types of projects in **Mule 3**: ‘Mule Project’ and ‘Mule Application with Maven’. In Mule 4 however, this is simplified providing only one option. All **Mule 4** applications are now Maven based. Anypoint Studio 7.x embeds Maven as well as a default configuration.

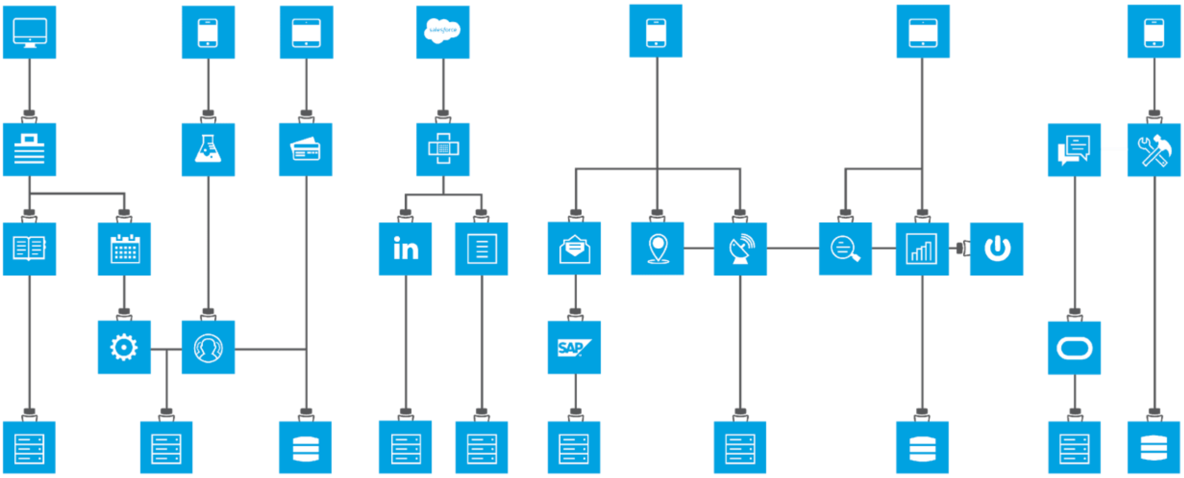


```
▼ hello-mule-3
    ▶ src/main/app (Flows)
      src/main/api
      src/main/java
    ▶ src/main/resources
      src/test/java
    ▶ src/test/resources
    ▶ JRE System Library [jdk1.8.0_162]
    ▶ Mule Server 3.8.5 EE
    ▶ src
    ▶ target
      mule-project.xml [Mule Server 3.8.5 EE]
      pom.xml
```

```
▼ hello-mule-4
    ▶ src/main/mule (Flows)
      src/main/java
    ▶ src/main/resources
      src/test/java
      src/test/munit
    ▶ src/test/resources
    ▶ HTTP [v1.3.2]
    ▶ JRE System Library [Java SE 8 [1.8.0_162]]
    ▶ Mule Server 4.1.4 EE
    ▶ Sockets [v1.1.2]
    ▶ src
    ▶ target
      mule-artifact.json
      pom.xml [Mule Server 4.1.4 EE]
```

Note: *This means that if any of your existing applications are not using Maven then you will need to ‘Mavenise’ the project first and then start looking at the migration.*

4. CONNECTORS USED IN THE APPLICATIONS



Connectors are a big topic, but the main thing to consider is whether the connectors you are using have been upgraded to Mule 4. Many connector developers like WHISHWORKS have already upgraded and certified their connectors and you can find them here: https://www.mulesoft.com/exchange/

If you are using any deprecated or custom connectors, you will need to redesign and implement for Mule 4. Keep in mind that even for connectors that are available for Mule 4, some modifications may be required. For example, you may have to modify DB queries for the latest DB connector, and/or the method of passing the HTTP Headers and query params.

Develop your migration plan

Determine where you will have the biggest impact

As you develop your migration plan you need to identify your most critical elements, dependencies and risks, as well as your migration options. You can then select the best migration plan for your specific Mule version to help you to continue enjoying the benefits of seamless API-led connectivity by paying only what you need to for your specific application needs, budget, and long-term goals.

DURING MIGRATION

Redesign to utilise Mule 4's enhanced capabilities

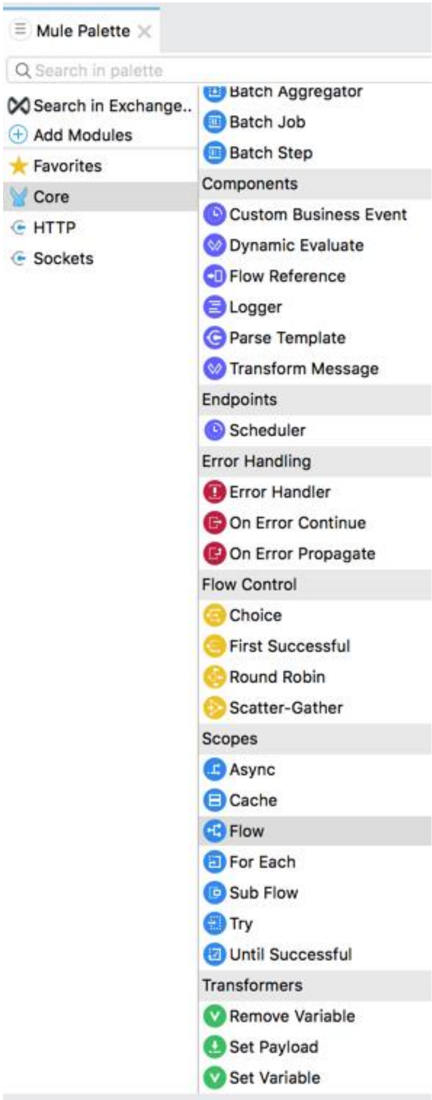
With your plan in place you are ready to start the migration. In doing so, you need to pay attention to the following three areas:

5. CORE COMPONENTS

In Mule 4 core components have changed considerably to improve performance and make it more readable and easier to implement. This means that you will have to rewrite all your core components to be able utilise the benefits presented with Mule 4.

For example if you have implemented your transformations and expressions with DataWeave 1.0 you will need to rewrite them in DataWeave 2.0. The way you set your properties has also changed and it is now a library instead of a plugin, so that you can use YAML. Similarly core components that have a lot of Java coding will need to be rewritten (the way Java classes have to be implemented in Mule 4 has changed drastically as "Callable" interface is no longer available).

Our advice is to use less Java coding in Mule 4 and instead use the built-in components provided.



6. MUNIT

MUnits are completely modified in Mule 4, and they will need to be rewritten. You will also have to replace assertion expressions with MunitTools::expressions.

7. REDESIGN TO UTILIZE MULE 4 CAPABILITIES

During development it may be worth redesigning some of your implementations to utilise the enhanced Mule 4 capabilities. For example, in Mule 4 batch processing has become part of the flow and therefore you don’t need to have batch components separately. Also with respect to filters, they are no longer available in Mule 4. In the latest version of Mule, you will have to create common DataWeave expressions that can be reused in all of your files.

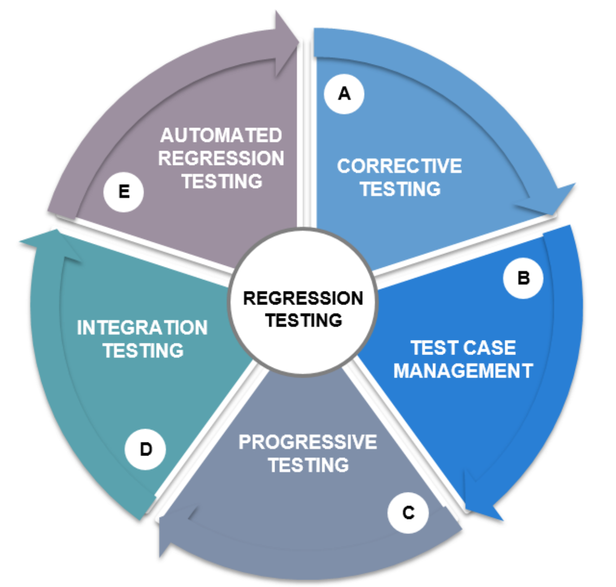
Finally, Message Enricher and Transformations have also been modified in Mule 4, so it is worth having a look into your current Mule 3.x implementation for components like these.

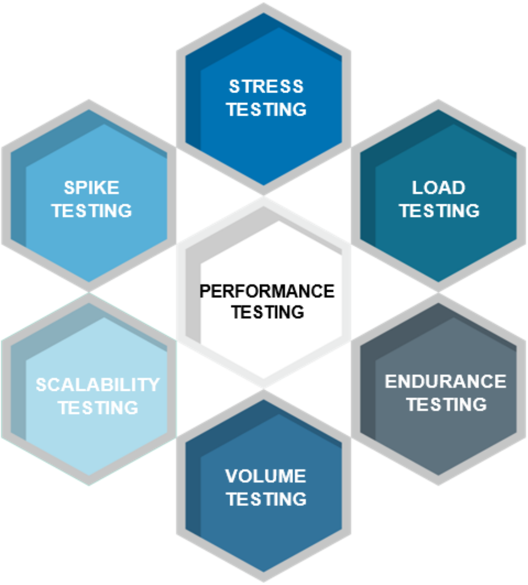
AFTER MIGRATION

Don't underestimate the importance of support enablement

8. REGRESSION / PERFORMANCE TESTING

The work doesn't stop after the migration. Testing is the most critical part of your migration before the go live. With regression testing we ensure that the changes we've implemented have not impacted functionality. With performance testing we ensure the efficiency of our applications and thereby create new baselines.





9. PHASED OR BIG BANG RELEASE

The new code must be saved in new repositories as the project structure has completely changed in Mule 4. All the CICD pipelines have to be updated accordingly, to build and release the applications to the required environment.

Once the testing is done and you are feeling confident with the results you can plan for the release. If you are migrating a large number of applications we suggest you release them with a phased approach. This way you will reduce the risk of having applications down in order to perform enhancements or further changes during the migration period. If you have a small set of applications to migrate then you can adopt a big bang approach to save on deployment and testing effort.

10. PLAN SUPPORT HANDOVER / HYPER CARE

Another important thing to consider is upskilling your support team so they are comfortable with Mule 4. Provision from 1 to 4 weeks for your hyper care planning depending on the size of your release. Once the release of your applications on Mule 4 is completed, you can plan for the decommissioning of your old applications.

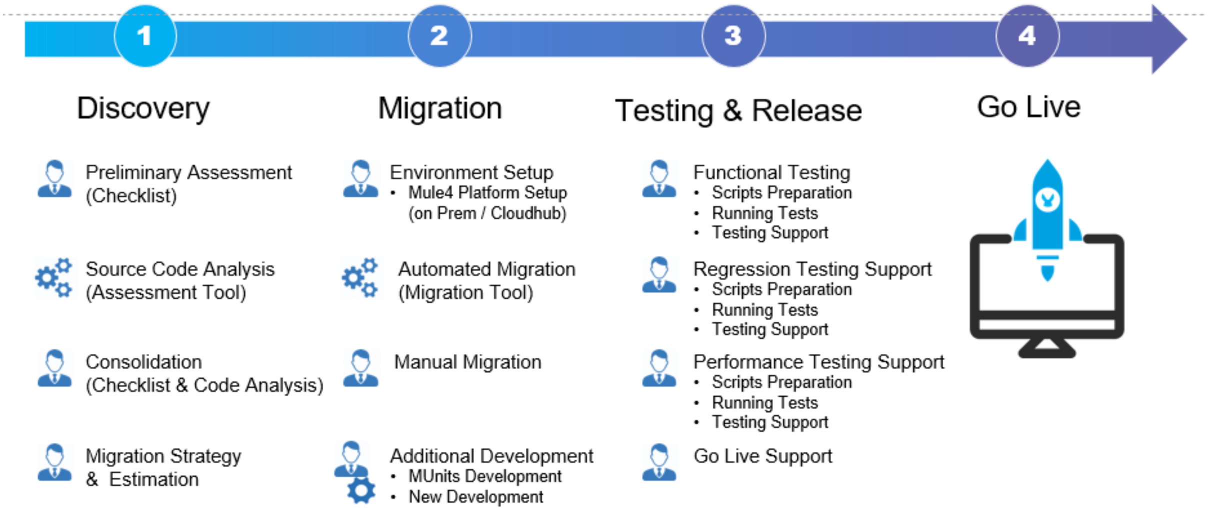
Mule 3 to Mule 4 Migration
An integrated approach

AN INTEGRATED APPROACH

Assess, plan, implement, test and release

Some of the points we have gone over in this guide may be new to you, or maybe you already have them in mind. To give you a framework to help you plan your migration, here is the approach which we follow with every migration project we undertake.

To accelerate speed of delivery, increase accuracy and eliminate errors, we have built various automation tools. Regardless of whether it is done manually or using a tool, every step is important and should be approached with care to avoid delays further down the line, and unwelcome surprises that will increase cost and effort.



1 Discovery

- Preliminary Assessment (Checklist)
- Source Code Analysis (Assessment Tool)
- Consolidation (Checklist & Code Analysis)
- Migration Strategy & Estimation

2 Migration

- Environment Setup
  - Mule4 Platform Setup (on Prem / Cloudhub)
- Automated Migration (Migration Tool)
- Manual Migration
- Additional Development
  - MUnits Development
  - New Development

3 Testing & Release

- Functional Testing
  - Scripts Preparation
  - Running Tests
  - Testing Support
- Regression Testing Support
  - Scripts Preparation
  - Running Tests
  - Testing Support
- Performance Testing Support
  - Scripts Preparation
  - Running Tests
  - Testing Support
- Go Live Support

4 Go Live

To summarise, we recommend starting with an initial discovery assessment of your environment. If you are on the cloud you are already half way through, but if your implementation is on-premise then you'll have to consider setting up your environment on Mule 4 while you have your Mule 3 application running in production as a part of BAU (business as usual).

Your migration strategy should consider the number and skillset of available resources and involve not just your IT team but also key business stakeholders who may play a role in deciding whether you want to go with a big bang approach or a phased approach.

Finally, remember the first step of the actual migration is to setup the platform and environment within Mule 4.

FINAL THOUGHTS

Running applications and data on unsupported versions can slow down the pace of innovation and delivery for your business. Whist the ramifications of ignoring end of support may seem less daunting than migrating, it is a great opportunity to transform your applications and infrastructure, take advantage of the features and improvements introduced in the latest versions of Mule, and benefit from:

- Reduced management complexity
- Higher performance
- Improved efficiency
- Faster pace of delivery
- Minimised risk
- Continued support

WHISHWORKS is a MuleSoft strategic partner and 5 times MuleSoft partner awards winner. With a dedicated MuleSoft Center of Excellence and over 500 MuleSoft certifications, we help customers across industries and geographies, accelerate transformation, improve outcomes and maximize ROI.

If you are thinking of migrating to Mule 4.x from a previous version and would like some help, feel free to contact us at:

+44 20 3475 7980 | UK
+1 646 568 9786 | US
+91 40 4265 6565 | India
info@whishworks.com | www.whishworks.com