Strategies & Learnings from a Drupal 7 to Drupal 9 migration

In this blog we look at what are the aspects to consider and factor in when doing a Drupal 7 to Drupal 9 migration? How do our choices help the Administrators, Editors and End users?

Article Type

- Narrative

Problem Statement

- Since there are multiple choices that appear in front of us when doing a Drupal 7 to Drupal 9 migration, how do we decide which is best for our Administrators, Editors, End users ?

Persona Questions

1. Who's this for?
   For Project Managers, Business Analysts and Tech Leads, Developers
2. What's the value?
   This blog will help us to,
   - Better perform the Drupal 7 to Drupal 9 migration by having clear understanding of the site and convert it into equivalent Drupal 9 site using a proper documented approach
   - Develop a generic template for understanding the source site and converting into equivalent target site
   - Apply the decisions that need to be made between different contributed modules and the why aspect of that decision

Draft

Drupal 7 to Drupal 9 migration - the big leap

In one of our earlier blogs Picking The Right Drupal MIgration Strategy we discussed what are the different migration approaches and how this is a good chance to refactor the content structure.

In this blog, we will see practical examples of the above steps in a way that would not be overwhelming for all the stakeholders.

Handling the Content types, Taxonomy and Modules first

Content types

We have highlighted the importance of handling the content when doing an migration in our earlier blogs Performing Drupal 9 Content Migration

To understand the content in our site better, it helps to look at the content with a fresh perspective. One of the ways we can do this is by categorizing the available content types into those which

- Rely on the Body field as primary source of information
- Rich in fields with different data types.

Categorization such as above can help us optimize the target site. For example:,

- Merging content types which rely on body field into Basic page with a mapping field to Drupal 7 content type
- We also need to decide which content types are not used and avoid them

Each content type can be documented through,

➢ Field level requirements
➢ Content Retention policy (what to migrate and what not?)
➢ What are the aggregate pages for the content type and type of filtering required?
➢ What does the content view page look like and what should be displayed?
➢ MIgration considerations specific to content type and assumptions made

A migration like this is also a good time to revisit the taxonomies, number of users, roles used in the site and add/remove items as needed.

Modules

This is the area where we need to be careful not to assume that everything is needed in the new site. During the course of a Drupal site it is natural for a lot of modules to be added but not actually used.

We recommend classifying the modules based on MoSCoW rules of prioritization.

Required modules (must haves and should haves)

Below fall in this category.

- ☐ Core modules in Drupal 7 and contributed modules in Drupal 7 which are now in Drupal 9 core
- ☐ Contributed modules in Drupal 7 which are actively used by administrators and editors. Examples include performance modules like memcache, basic utility modules like admin menu, XML sitemap, etc…

Optional modules (could haves)

Modules which are used in Drupal 7 but are not needed in initial phases of the site and can be included only if needed (like Features, Rules, Panels), fall in this category. This is because a lot can be achieved with Drupal 9 which was only possible via a contributed module in Drupal 7.

It is important to validate the need for each module (contributed and custom) and try to have only the minimal modules in Drupal 9 as it will ease the maintenance of the site in the long run.

Bringing the Roles and Content Workflow to work together

As far as Roles are concerned, below are the recommended steps.

- ☐ Validate existing roles and redefine them on the basis of actions done by the user.
- ☐ Use combinations of roles for a user to achieve the actions they did on Drupal 7 site
- ☐ If access to site content needs to be based on their categorisation, Permissions by Term module might be helpful
- ☐ Remove permissions which are duplicate or not any more required

- ☐ If the control of the site needs to be decentralized, then Group module is recommended
- ☐ Check for active users and migrate only them.
- ☐ If the number of users are less, then it is better to do the new role mapping manually, as achieving them via migration scripts would be time consuming considering the mapping process is a one time migration activity.

For the content workflow,

- ☐ Define clear content moderation states which each role can perform
- ☐ For sites with a simple workflow Drupal core's Content moderation module is recommended
- ☐ For sites which used the Workflow module in Drupal 7 using the same in Drupal 9 would make the migration easy as there is an upgrade path
- ☐ If the site needs a workflow dashboard, email notifications Workbench suite with Workbench, Workbench Moderation, Workbench Email serves good.

How are we going to manage the media?

The need for media depends on following:

- ☐ Whether the media in Drupal 7 site is restricted to a specific type like only images and video or is it diverse enough like Files, Audio scripts and any other custom media type which are specific to the site?
- ☐ What is the future scope of media in the site in Drupal 9 ? Are we expecting the site to be media rich with lots of metadata?
- ☐ Do we need the same field to store different types of files?

Based on the above we can decide if we need to have simple Image and Video fields to store the media or should we leverage the media module in core which allows us to store different types of files in the same field and also create custom media types with metadata.

It's worth noting that the community has often thought of moving to media as part of the default Drupal install. That said, it is extremely unlikely that simple file fields will disappear from the core in the foreseeable future. But if you are planning to pick media, you're in good company.

Check references for more detailed approaches for migrating media

Based on the path we choose we have some supporting modules.

- Simple Image/Video/File field path - File Entity Browser to browse files
- Media field in Drupal core - organize media with Media Directories module, Bulk Upload to upload a lot of files in one go.

NOTE: Media Directories module provides only virtual directories and are not physical mapping to actual directories in the file system. Work in progress for the same. There is also work going on to support drag and drop for the media library.

This is a good time to do some field re-engineering

FIeld re-engineering goal is to look at the data types of each field and validate if we are doing justice to them.

Below were some wrong examples of field types which is common in Drupal 7.,

➢ Plain text being used where it should have been Date field
➢ CkEditor being used where it should have been plain text

We can either do the following in destination site,

- change the field types and move the data (we might need to check if the data type between source and destination is compatible in this case)
  (or)
- move them to the same type and then do migration from old to new field type on a case by case basis

This will improve the editorial experience a lot.

Other than the above aspects, it is important to also consider the different integrations available in the current site and how are we going to achieve them in the migrated site? If these are custom integrations you have written, then you would have to rewrite these modules to be compatible with Drupal 9. That is a much larger topic and beyond the scope of this article.

Conclusion

It is important we document the above aspects as it clearly says the decision and route we take which would provide clarity for the team working on the migration.

References

- Migrating Files and Images, Migrate File, External reference
- Migrating to files then media, Migrate File Entities to Media Entities, External

reference