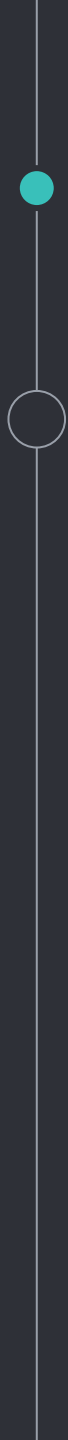


- Migration is hard,
but it works

● So, what's the problem with migration?

- It's complicated!
- Core provides full site migration only. That does not take into account that you may have learned something!
- Examples refer to all kinds of data migration but leave out the real world drupal stuff
- Documentation is sketchy, things are still experimental (=changing)



● Plan for this session

- Understand partial migration
- Introduce the main tools
- Move a content type over to D8
- Move the content over
 - Standard content
 - Referenced content
 - Build relationships
- Describe the migration workflow
- Live demo the full process.

● Modules/Tools you will need

- Drush
- Migrate
- Migrate Drupal
- Migrate plus (tool & examples)
- Migrate tools (drush extensions)
- (Drupal Console)
- A custom Module
 - Migration configurations.

● What can be migrated

○ Pretty much everything!

- Users
- Entities
- Taxonomies
- Fields (definitions, instances, formatter settings, widget settings etc.)
- Blocks, block content
- Comments
- Etc. etc....

● How to set up for a migration

- Best to have both sites set up locally
- You need access to the databases
- Understand tables and structures

Step 1: Add migrate DB to settings.php

```
$databases['migrate']['default'] = array (  
  'database' => 'polysite',  
  'username' => 'root',  
  'password' => 'password',  
  'prefix' => '',  
  'host' => 'localhost',  
  'port' => '3306',  
  'namespace' => 'Drupal\\Core\\Database\\Driver\\mysql',  
  'driver' => 'mysql',  
);
```

Migrations are defined as Configurations!

Module

config

install

migrate_plus.migration.xyz.yml

Install module

Drupal

Configuration

migrate_plus.migration.xyz

Drush

- migrate-status
- migrate-import
- migrate-rollback
- migrate-stop
- migrate-reset-status
- migrate-messages

Find out more:

Core/modules/xyz/migration_templates

A Configuration File defining a migration

Information

Where from?

new : old

What to do?

Where to?

Dependencies!

```
id: d7_node_type
label: Node type configuration
migration_group: polytechna
migration_tags:
  - Drupal 7
source:
  plugin: d7_node_type
  constants:
    preview: 1 # DRUPAL_OPTIONAL
process:
  type: type
  name: name
  description: description
  help: help
  title_label: title_label
  preview_mode: 'constants/preview'
  display_submitted: display_submitted
  new_revision: 'options/revision'
  create_body: create_body
  create_body_label: body_label
destination:
  plugin: entity:node_type
migration_dependencies: {}
dependencies:
  module:
    - custom_migrate
  enforced:
    module:
      - custom_migrate
```


- Handling configurations

- For the simple minded:
Write -> install -> test
Edit -> uninstall -> install

More elegant (and more dangerous)
Drush cedit-> drush cr ->test

Remove config (if stuff goes wrong)
Drupal config:delete (console).

- Migrating an Image Field

- 1. Move the files (entites)
 2. Migrate the entity references

Move the files

```
id: d7_file
label: Files
migration_tags:
  - Drupal 7
source:
  plugin: d7_file
  constants:
    source_base_path: 'http://ubuntudev/polysite/'
  process:
    fid: fid
    filename: filename
    source_full_path:
      -
        plugin: concat
        delimiter: /
        source:
          - constants/source_base_path
          - filepath
      -
        plugin: urlencode
    uri:
      plugin: file_copy
      source:
        - '@source_full_path'
        - uri
    filemime: filemime
    status: status
    created: timestamp
    changed: timestamp
    uid: uid
destination:
  plugin: entity:file
```

Migrate the content entity

```
source:
  plugin: d7_node
  node_type: produktgruppe
process:
  type:
    plugin: default_value
    default_value: produktgruppe
    langcode:
      plugin: default_value
      source: language
      default_value: "und"
  title: title
  uid: node_uid
  status: status
  created: created
  changed: changed
  promote: promote
  sticky: sticky
  field_beschreibung: field_beschreibung
  field_bild:
    plugin: iterator
    source: field_bild
    process:
      target_id:
        plugin: migration
        migration: d7_file
        source: fid
  field_filter_string: field_filter_string
destination:
  plugin: entity:node
```

Migrate entity references

Field_bild is filled by

Iterator on all instances

The value of target_id is filled by

The migration plugin

The relation generated in d7_file

The value fid in 'old' data

```
field_bild:  
  plugin: iterator  
  source: field_bild  
  process:  
    target_id:  
      plugin: migration  
      migration: d7_file  
      source: fid
```

● Migration Process

- Setup and test the migrations
- Run again
 - Only new content will be migrated
- Run with --update
 - All content will be reimported
- Add 'track_changes = true'
 - Sources will be compared to last import
 - If the hash is different, will be reimported
- Use group to run all
- Do the final run when switching over.

● More possibilities

○ There are plugins available for

- Xml
- Csv
- Json

-> see examples in migrate_plus

○ You can write your own plugin

- Source
- Process
- Destination

-> see https://www.drupal.org/project/migrate_drupal_d8



More....

Migration API

<https://www.drupal.org/node/2127611>

Drush migrate commands

<https://www.drupal.org/node/1561820>