MIGRATING DATA TO DRUPAL 8

8.2.x Ready!

about.me

Ignacio Sánchez

Drupal developer @ bluespark

@isholgueras

nacho@bluespark.com

drupal.org/u/isholgueras

http://www.isholgueras.com

MIEMBRO
DE LA
ASOCIACIÓN
ESPAÑOLA
DE DRUPAL

Drupal Association
Individual Member

STEPS

1. Requirements
2. Anatomy of a migration
3. Migration Framework
4. Performance tips

1.

REQUIREMENTS

Migrate is in core!

Three new modules

- **Migrate**
  Handles migrations. Framework.

- **Migrate Drupal**
  Contains migrations from D6 & D7.

- **Migrate Drupal UI**
  The older migrate update (new in 8.1.x).

Migrate is in core!

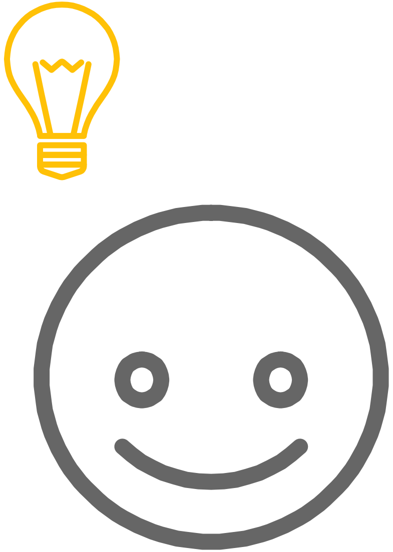
But... how can I execute my migration.

UI is not ready?

No Drush command?

So…

Contrib!



Needed

- Drupal 8.1.x (or superior)
- Drush 8
- Migrate tools (contrib)
- Migrate Plus (contrib)

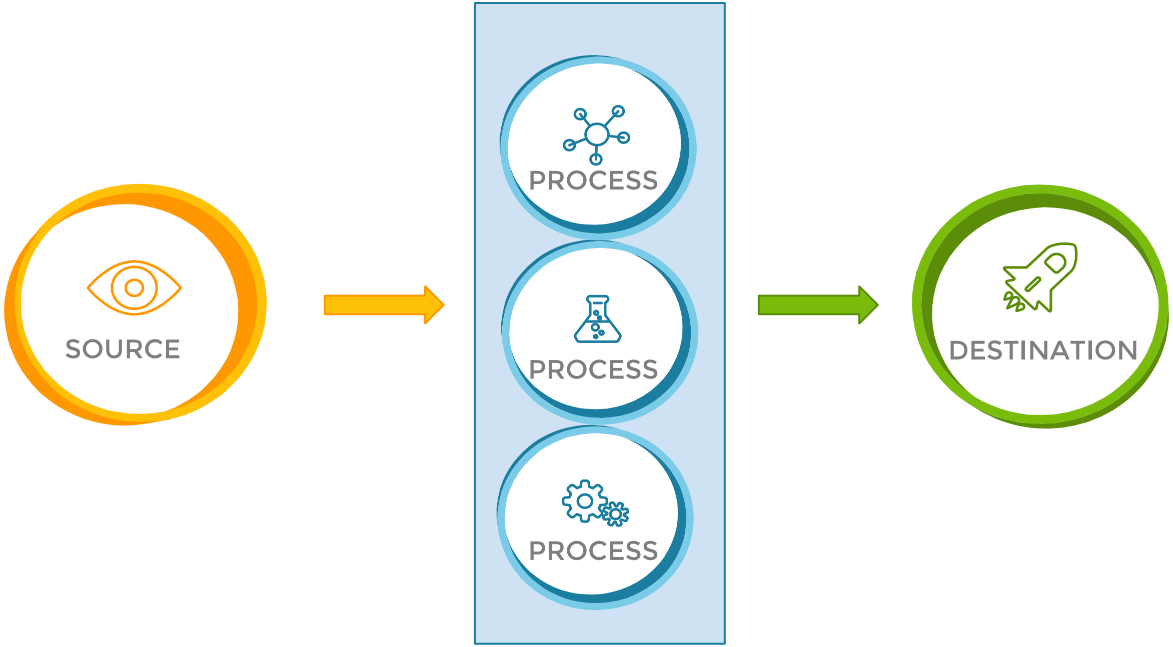
2.

ANATOMY OF A MIGRATION

Workflow

2\.

ANATOMY OF
A MIGRATION



In files

DEFINITIONS

Yaml Files.

Custom files

PLUGINS

PHP Files

Core or custom files

Types:

- Source
- Process
- Destination
- Builder
- ID Map

The easiest example

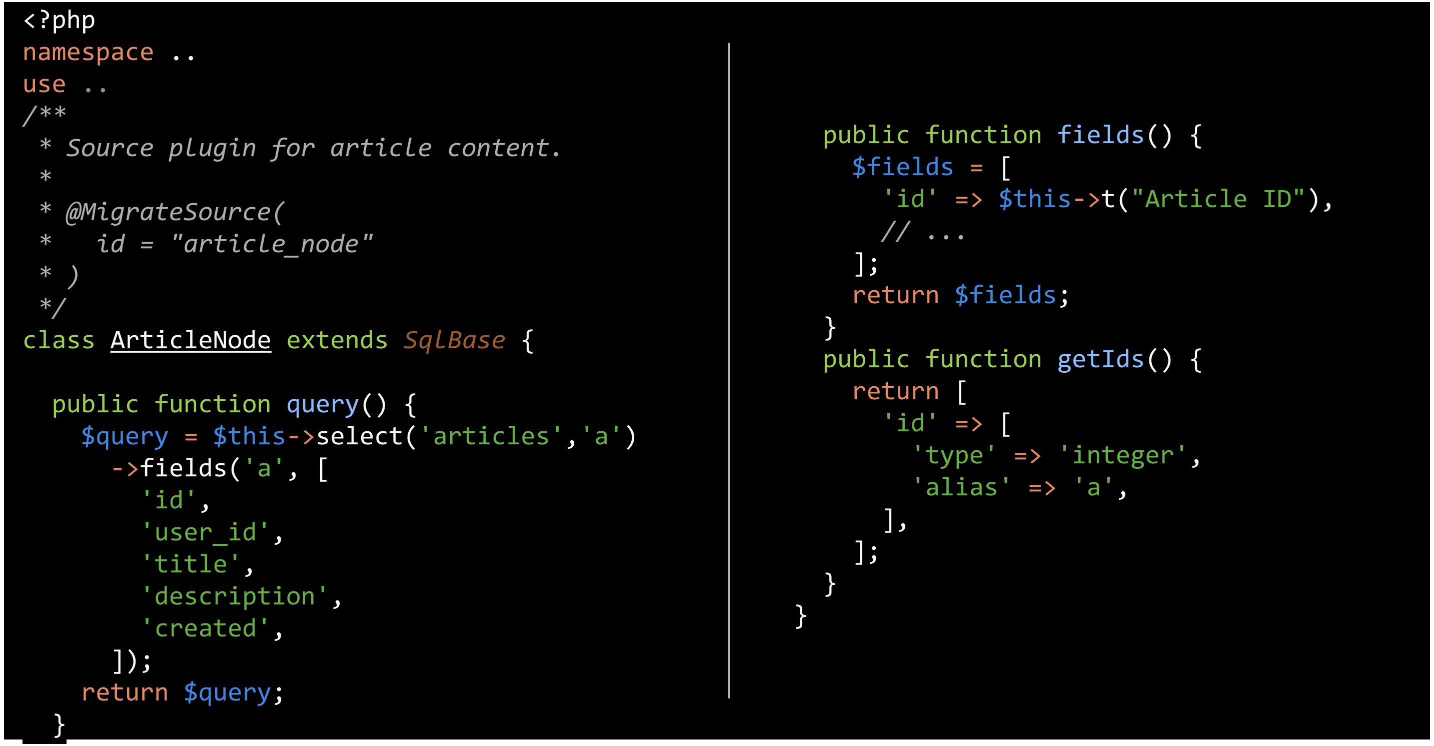
DEFINITION

config/install/migrate_plus.migration.article_node.yml

```
id: article_node
label: Migrate posts from CakePHP to Drupal 8

source:
  plugin: article_node
  key: legacy
destination:
  plugin: entity:node
process:
  type:
    plugin: default_value
    default_value: article
  nid: id
  title: title
  'body/value': description
  uid: user_id
  status:
    plugin: default_value
    default_value: true
  created:
    plugin: callback
    source: created
    callable: strtotime
migration_dependencies: {} #new in 8.1.x
```

PLUGINS

src/Plugin/migrate/source/ArticleNode.php



```
<?php
namespace ..
use ..
/**
 * Source plugin for article content.
 *
 * @MigrateSource(
 *   id = "article_node"
 * )
 */
class ArticleNode extends SqlBase {

 public function query() {
  $query = $this->select('articles','a')
   ->fields('a', [
    'id',
    'user_id',
    'title',
    'description',
    'created',
   ]);
  return $query;
 }
```

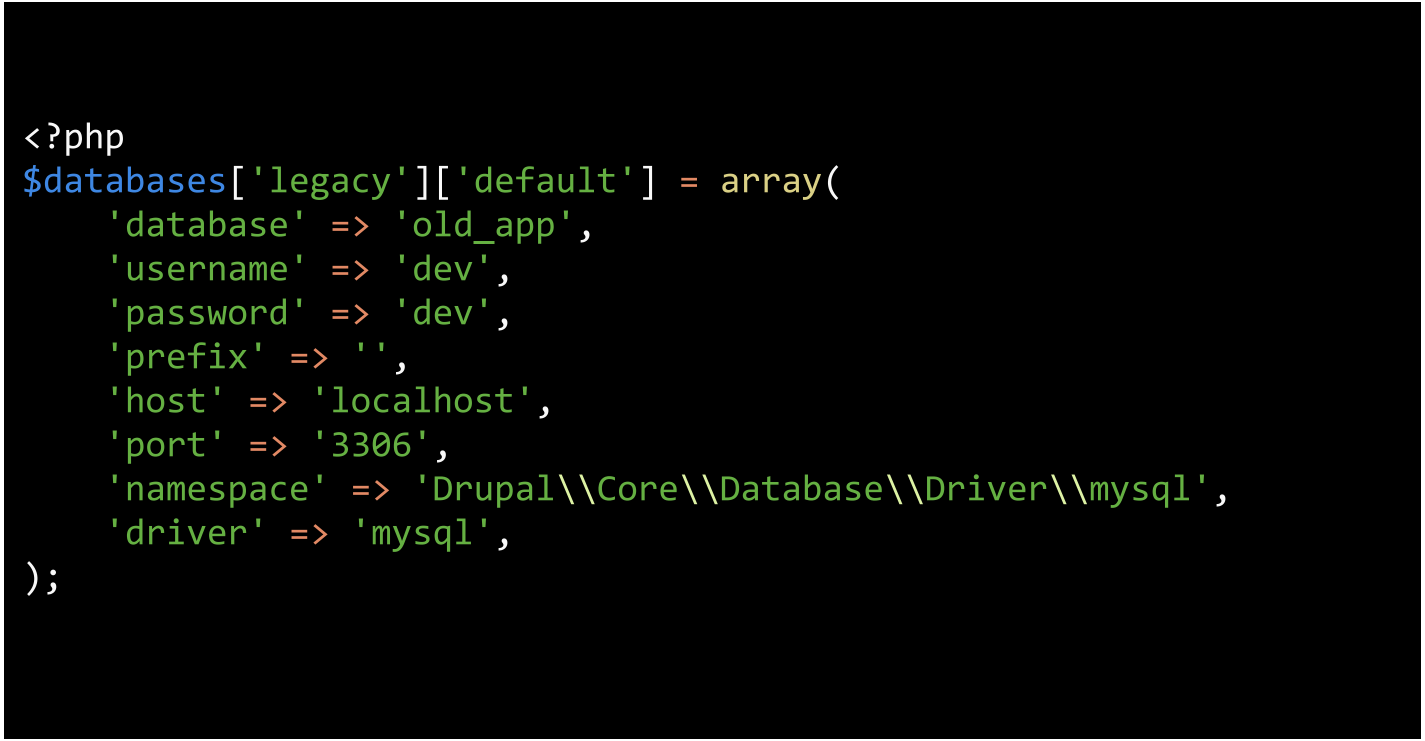
```
 public function fields() {
  $fields = [
   'id' => $this->t("Article ID"),
   // ...
  ];
  return $fields;
 }
 public function getIds() {
  return [
   'id' => [
    'type' => 'integer',
    'alias' => 'a',
   ],
  ];
 }
}
```

2.

ANATOMY OF A MIGRATION

CONFIGURATION

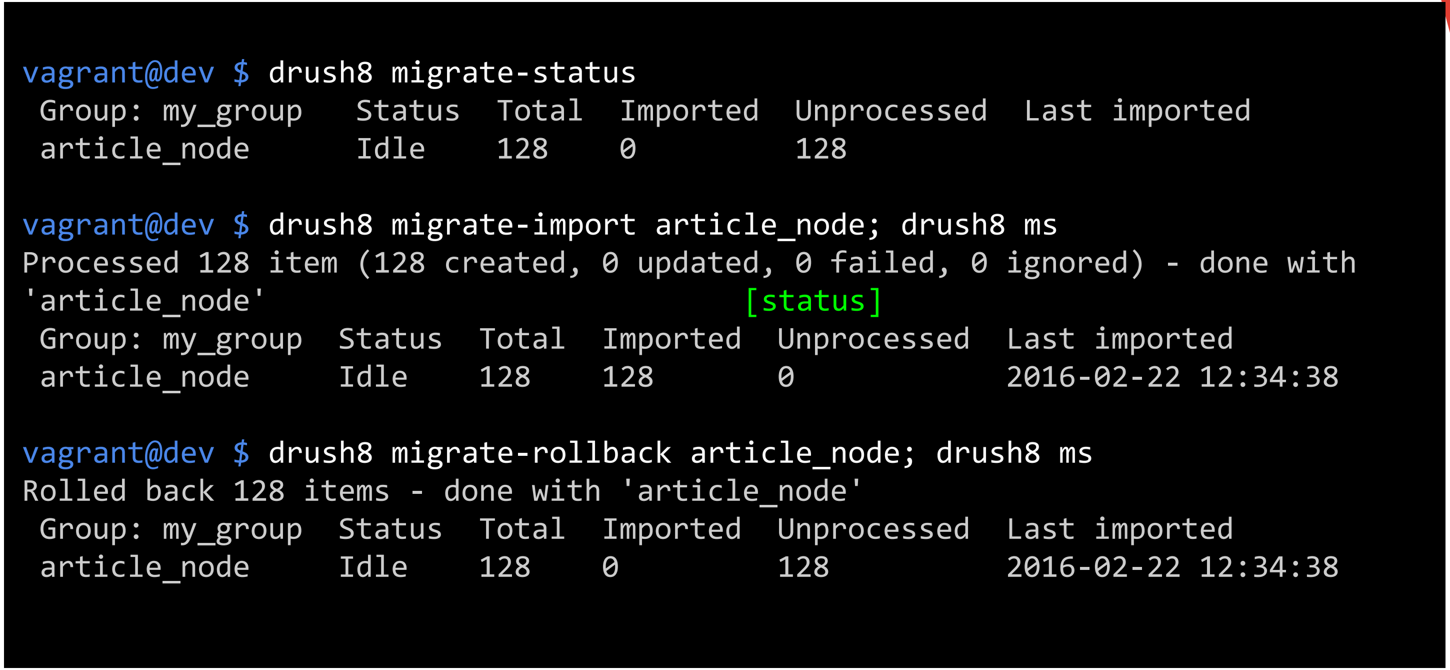
sites/local/settings.php



```php
<?php
$databases['legacy']['default'] = array(
    'database' => 'old_app',
    'username' => 'dev',
    'password' => 'dev',
    'prefix' => '',
    'host' => 'localhost',
    'port' => '3306',
    'namespace' => 'Drupal\\Core\\Database\\Driver\\mysql',
    'driver' => 'mysql',
);
```

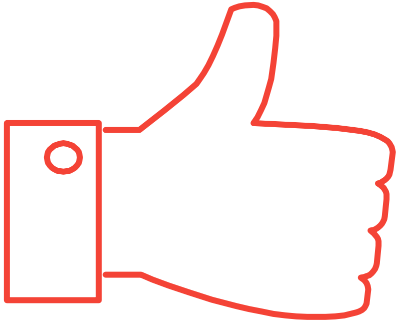
EXECUTION

Only with **Drush8 (8.1-dev)** and **migrate_tools** enabled



```
vagrant@dev $ drush8 migrate-status
 Group: my_group   Status  Total  Imported  Unprocessed  Last imported
 article_node      Idle    128    0         128

vagrant@dev $ drush8 migrate-import article_node; drush8 ms
Processed 128 item (128 created, 0 updated, 0 failed, 0 ignored) - done with
'article_node'                           [status]
 Group: my_group  Status  Total  Imported  Unprocessed  Last imported
 article_node     Idle    128    128       0            2016-02-22 12:34:38

vagrant@dev $ drush8 migrate-rollback article_node; drush8 ms
Rolled back 128 items - done with 'article_node'
 Group: my_group  Status  Total  Imported  Unprocessed  Last imported
 article_node     Idle    128    0         128          2016-02-22 12:34:38
```

Too easy!



3.

MIGRATION FRAMEWORK

0.- KNOWLEDGE

Source Plugin:

- SqlBase

Process Plugin:

- ProcessPluginBase

Destination Plugin:

- DestinationBase

Most used

More plugins in: `core/modules/migrate/src/Plugin/migrate`

1.- SOURCE

Here we tell SqlBase:

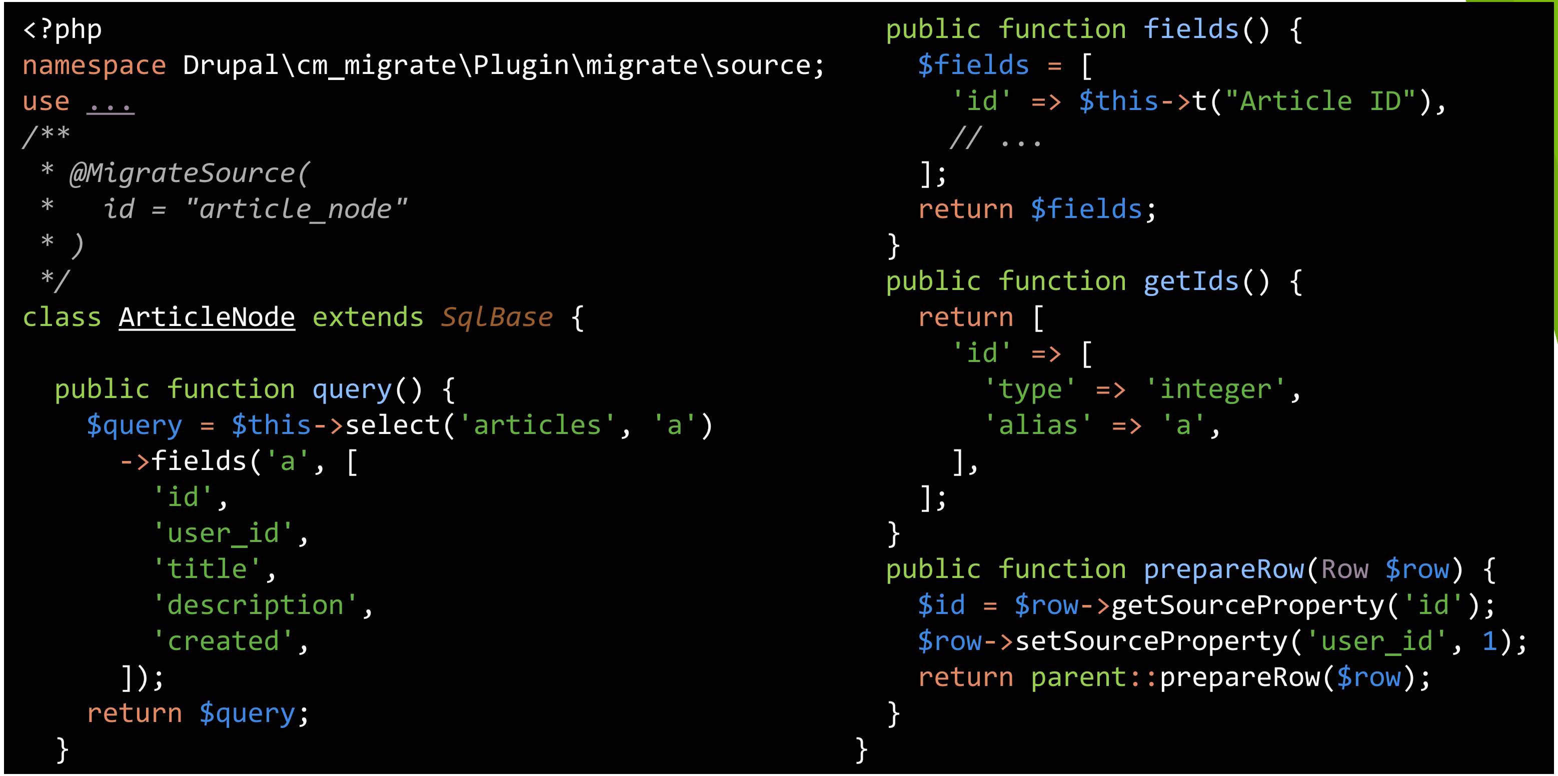
- Which **Database** is the **Source**.

```
#migrate_plus.migration.article_node.yml
source:
  plugin: article_node
  key: legacy
  #target: default
```

```
#settings.php
//By default [‘default’][‘migration’]
$databases['legacy']['default'] = array(
//           key      target
```

1.- SOURCE

- And which **Plugin** will make the **Query**

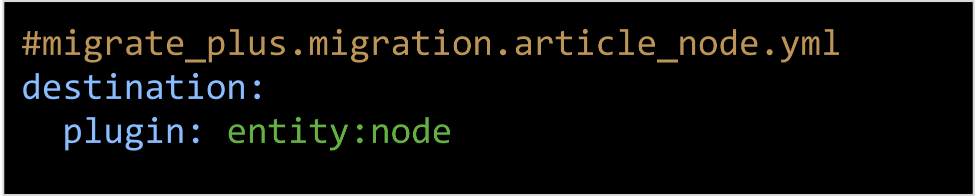


```php
<?php
namespace Drupal\cm_migrate\Plugin\migrate\source;
use ...
/**
 * @MigrateSource(
 *   id = "article_node"
 * )
 */
class ArticleNode extends SqlBase {

  public function query() {
    $query = $this->select('articles', 'a')
      ->fields('a', [
        'id',
        'user_id',
        'title',
        'description',
        'created',
      ]);
    return $query;
  }
```

```php
  public function fields() {
    $fields = [
      'id' => $this->t("Article ID"),
      // ...
    ];
    return $fields;
  }
  public function getIds() {
    return [
      'id' => [
        'type' => 'integer',
        'alias' => 'a',
      ],
    ];
  }
  public function prepareRow(Row $row) {
    $id = $row->getSourceProperty('id');
    $row->setSourceProperty('user_id', 1);
    return parent::prepareRow($row);
  }
}
```

2.- DESTINATION

How and where to store the data



```
#migrate_plus.migration.article_node.yml
destination:
  plugin: entity:node
```

- entity:<place-here-an-entity>

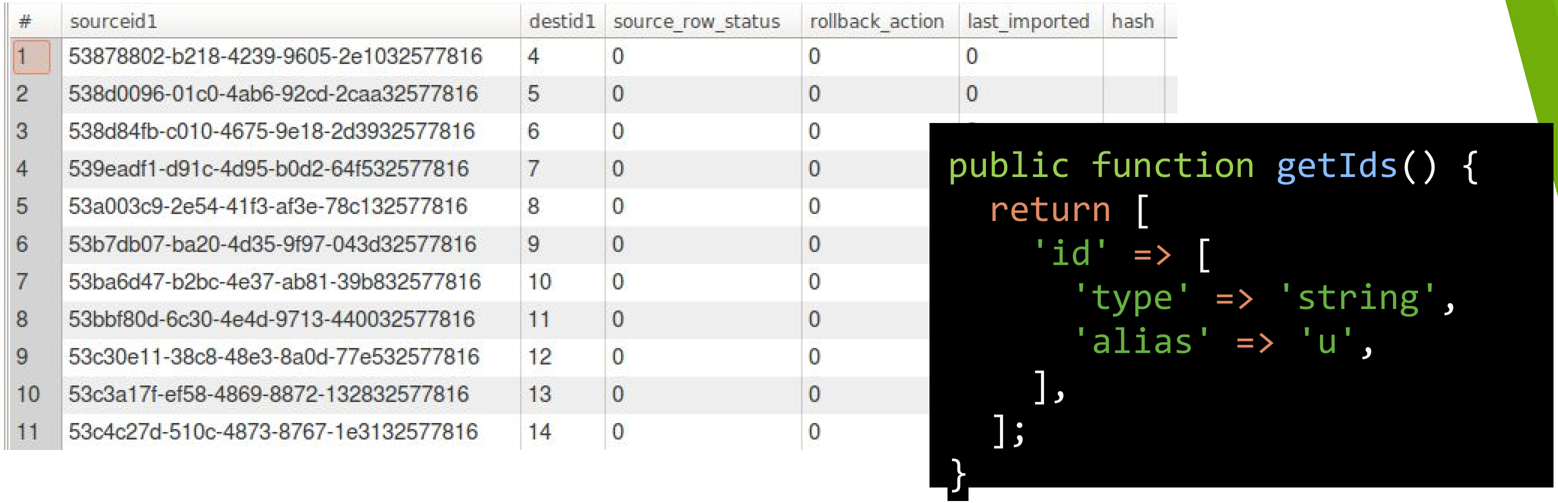
2.- DESTINATION

Need more destination plugins?

Search for “destination:” in core

3.- ID MAPPING

How Migrate associates old rows with new rows.



#	sourceid1	destid1	source_row_status	rollback_action	last_imported	hash
1	53878802-b218-4239-9605-2e1032577816	4	0	0	0	
2	538d0096-01c0-4ab6-92cd-2caa32577816	5	0	0	0	
3	538d84fb-c010-4675-9e18-2d3932577816	6	0	0		
4	539eadf1-d91c-4d95-b0d2-64f532577816	7	0	0		
5	53a003c9-2e54-41f3-af3e-78c132577816	8	0	0		
6	53b7db07-ba20-4d35-9f97-043d32577816	9	0	0		
7	53ba6d47-b2bc-4e37-ab81-39b832577816	10	0	0		
8	53bbf80d-6c30-4e4d-9713-440032577816	11	0	0		
9	53c30e11-38c8-48e3-8a0d-77e532577816	12	0	0		
10	53c3a17f-ef58-4869-8872-132832577816	13	0	0		
11	53c4c27d-510c-4873-8767-1e3132577816	14	0	0		

```php
public function getIds() {
  return [
    'id' => [
      'type' => 'string',
      'alias' => 'u',
    ],
  ];
}
```

4.- PROCESS

How we **transform** each field, each file or data.

You are able to:

- **Map fields**: Same value as origin.
- **Modify**: Change or process the value.
- **Add**: Create new fields from other fields or calculate these fields.

4.- PROCESS

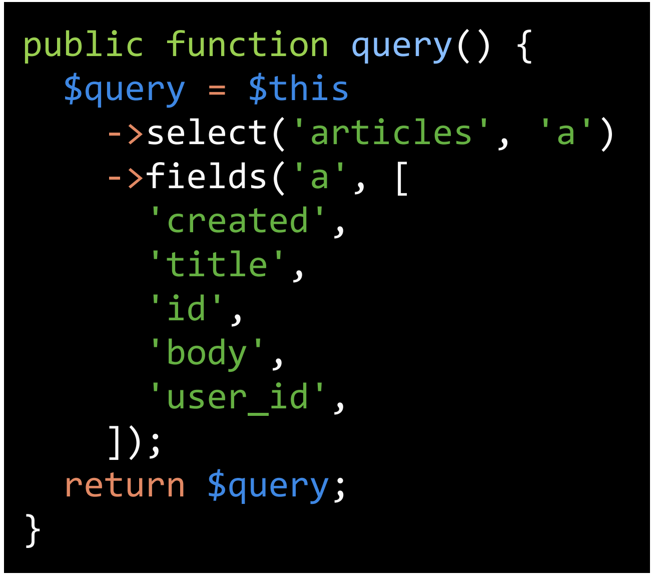
Map fields. Values are equal in both sides

```
public function query() {
  $query = $this
    ->select('articles', 'a')
    ->fields('a', [
      'created',
--->  'title',
      'id',
      'body',
      'user_id',
    ]);
  return $query;
}
```

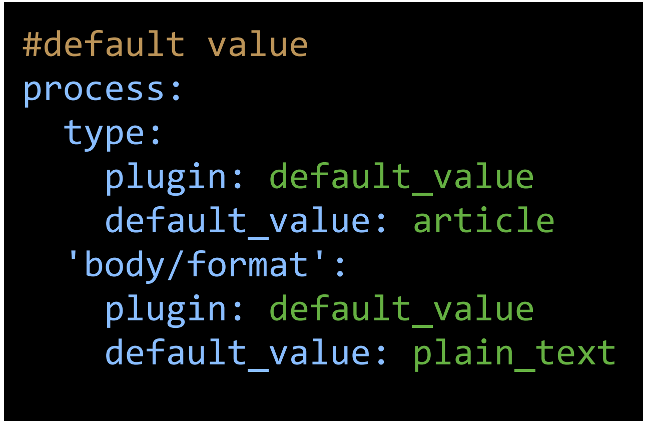
```
#common mapping
process:
  title: title <---
```

4.- PROCESS

DefaultValue. Add a default value



```
public function query() {
  $query = $this
    ->select('articles', 'a')
    ->fields('a', [
      'created',
      'title',
      'id',
      'body',
      'user_id',
    ]);
  return $query;
}
```



```
#default value
process:
  type:
    plugin: default_value
    default_value: article
  'body/format':
    plugin: default_value
    default_value: plain_text
```

4.- PROCESS

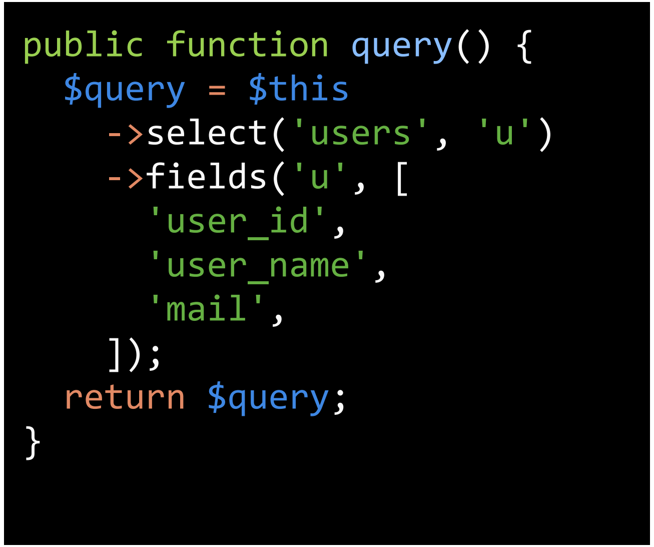
Callable. Values are related but a process is needed with a **function**

```
public function query() {
  $query = $this
    ->select('articles', 'a')
    ->fields('a', [
      'created_date',
      'title',
      'id',
      'body',
      'user_id',
    ]);
  return $query;
}
```

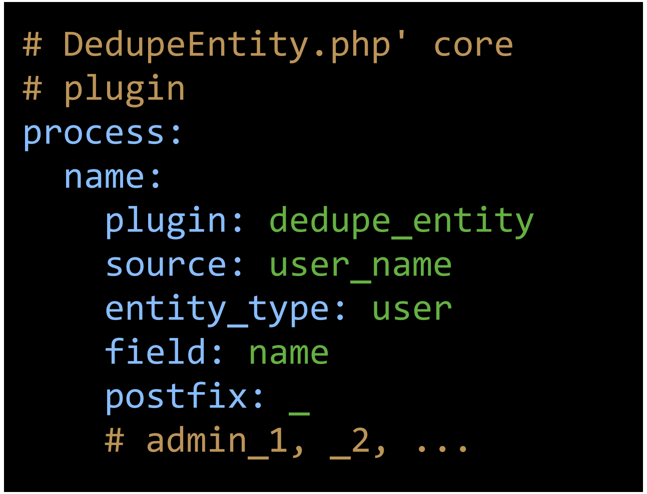
```
#'Callable.php' core plugin
process:
  created:
    plugin: callback
    source: created_date
    callable: strtotime
```

4.- PROCESS

DedupeEntity. Values in destination cannot be equal, but in origin could be.



```
public function query() {
  $query = $this
    ->select('users', 'u')
    ->fields('u', [
      'user_id',
      'user_name',
      'mail',
    ]);
  return $query;
}
```

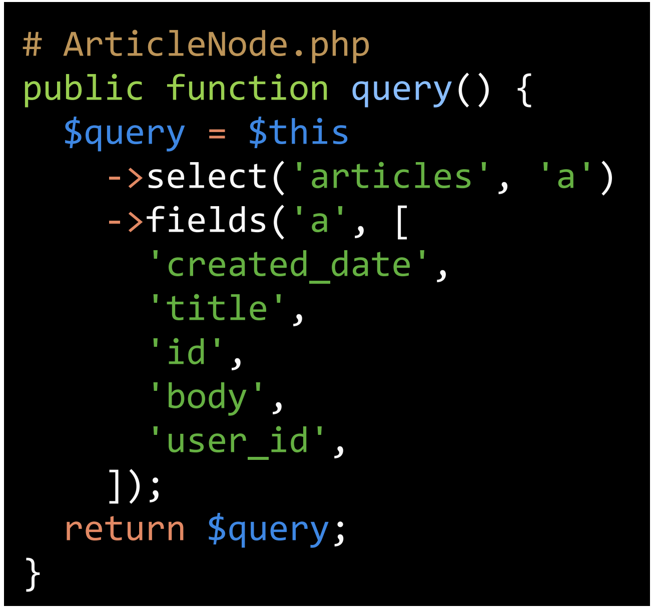


```
# DedupeEntity.php' core
# plugin
process:
  name:
    plugin: dedupe_entity
    source: user_name
    entity_type: user
    field: name
    postfix: _
    # admin_1, _2, ...
```

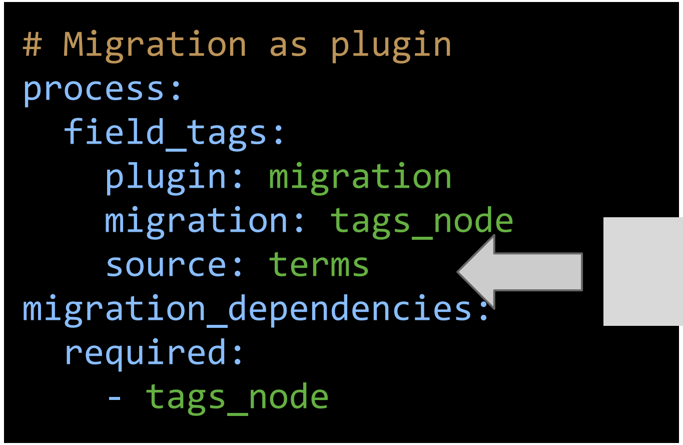
4.- PROCESS

Migration. Values from another migration.

Use it!   PLEASE!



```
# ArticleNode.php
public function query() {
  $query = $this
    ->select('articles', 'a')
    ->fields('a', [
      'created_date',
      'title',
      'id',
      'body',
      'user_id',
    ]);
  return $query;
}
```



```
# Migration as plugin
process:
  field_tags:
    plugin: migration
    migration: tags_node
    source: terms
migration_dependencies:
  required:
    - tags_node
```

fields()

4.- PROCESS

Migration. Values from another migration.

```
# ArticleNode.php
public function fields() {
    $fields = [
      'terms' =>
$this->t("New field terms"),
      // ...
    ];
    return $fields;
  }
```

```
# ArticleNode.php
public function prepareRow(Row $row) {
    $terms = $this->select('terms')
          //...
          ->fetchCol();
    $row->setSourceProperty('terms',
$terms);
    return parent::prepareRow($row);
  }
}
```

4.- PROCESS

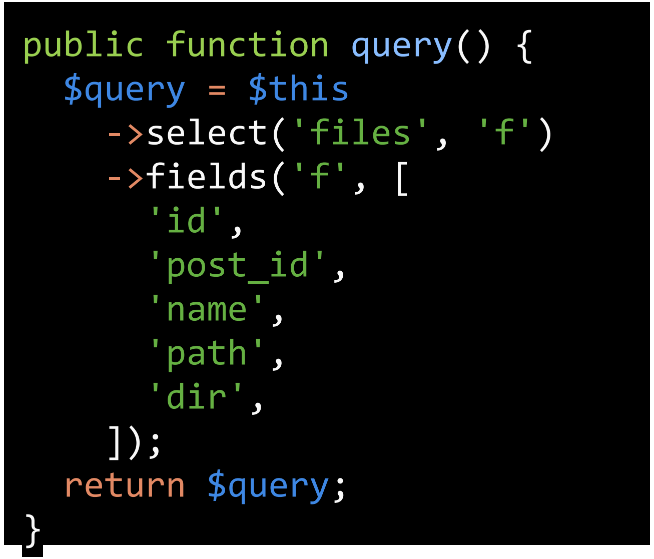
Migration. Values from another migration.

```
# TagsNode.php
public function query() {
     // code
    ->fields('terms', [
      'parent_term',
      'term']);
}
```

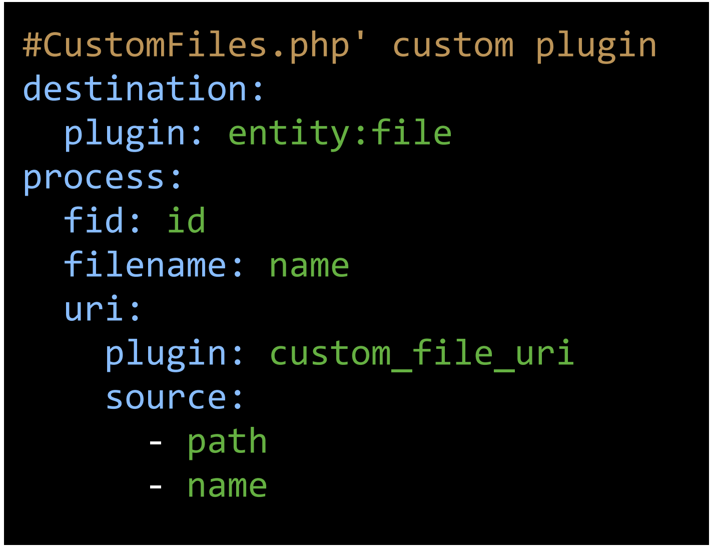
```
#migrate_plus.migration.tags_node
.yml
source:
  plugin: tags_node
destination:
  plugin: entity:taxonomy_term
process:
  name: term
  parent:
    plugin: migration
    migration: tags_node
    source: parent_term
```

4.- PROCESS

CustomPlugins. We need to copy some files

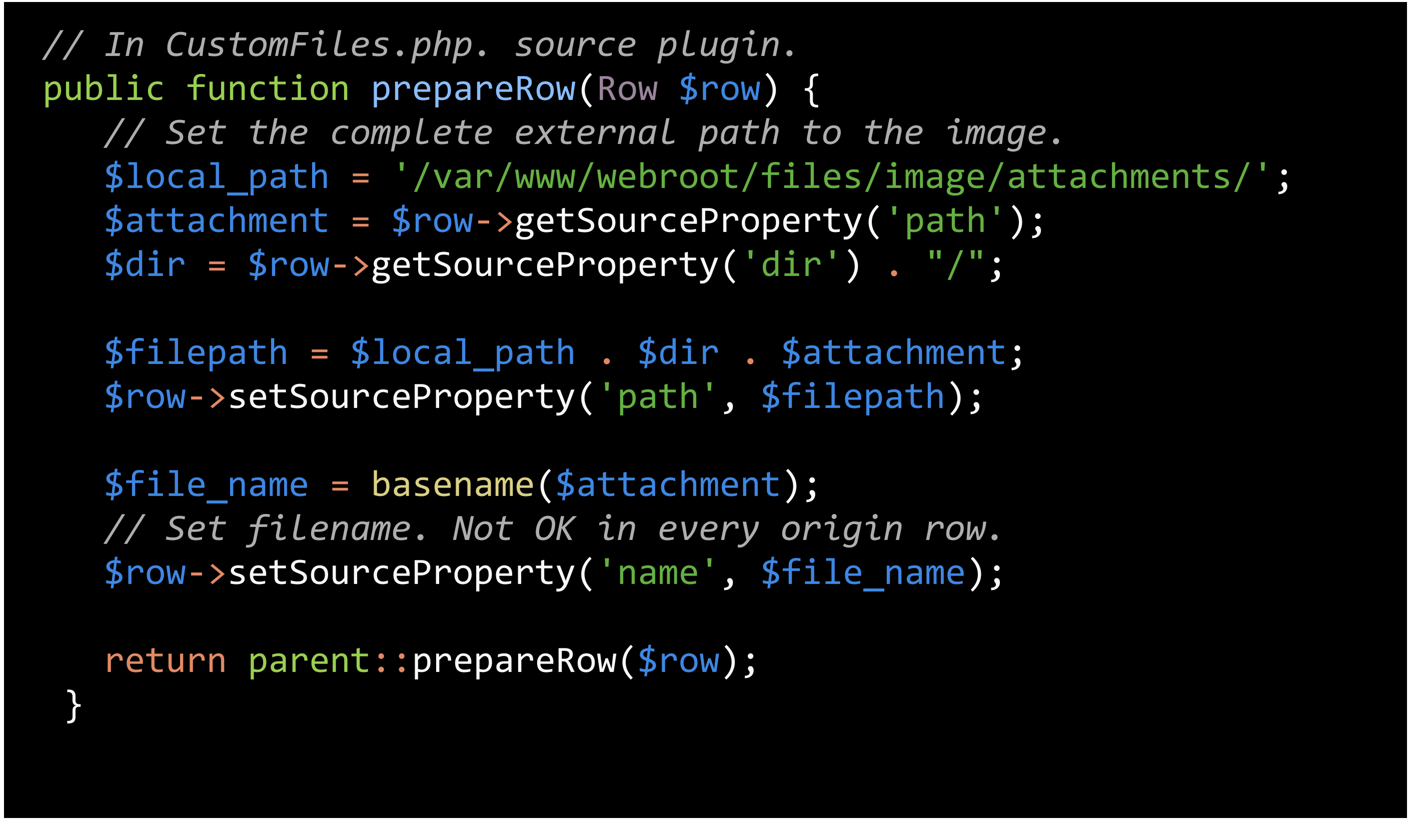


```
public function query() {
  $query = $this
    ->select('files', 'f')
    ->fields('f', [
      'id',
      'post_id',
      'name',
      'path',
      'dir',
    ]);
  return $query;
}
```



```
#CustomFiles.php' custom plugin
destination:
  plugin: entity:file
process:
  fid: id
  filename: name
  uri:
    plugin: custom_file_uri
    source:
      - path
      - name
```

4.- PROCESS

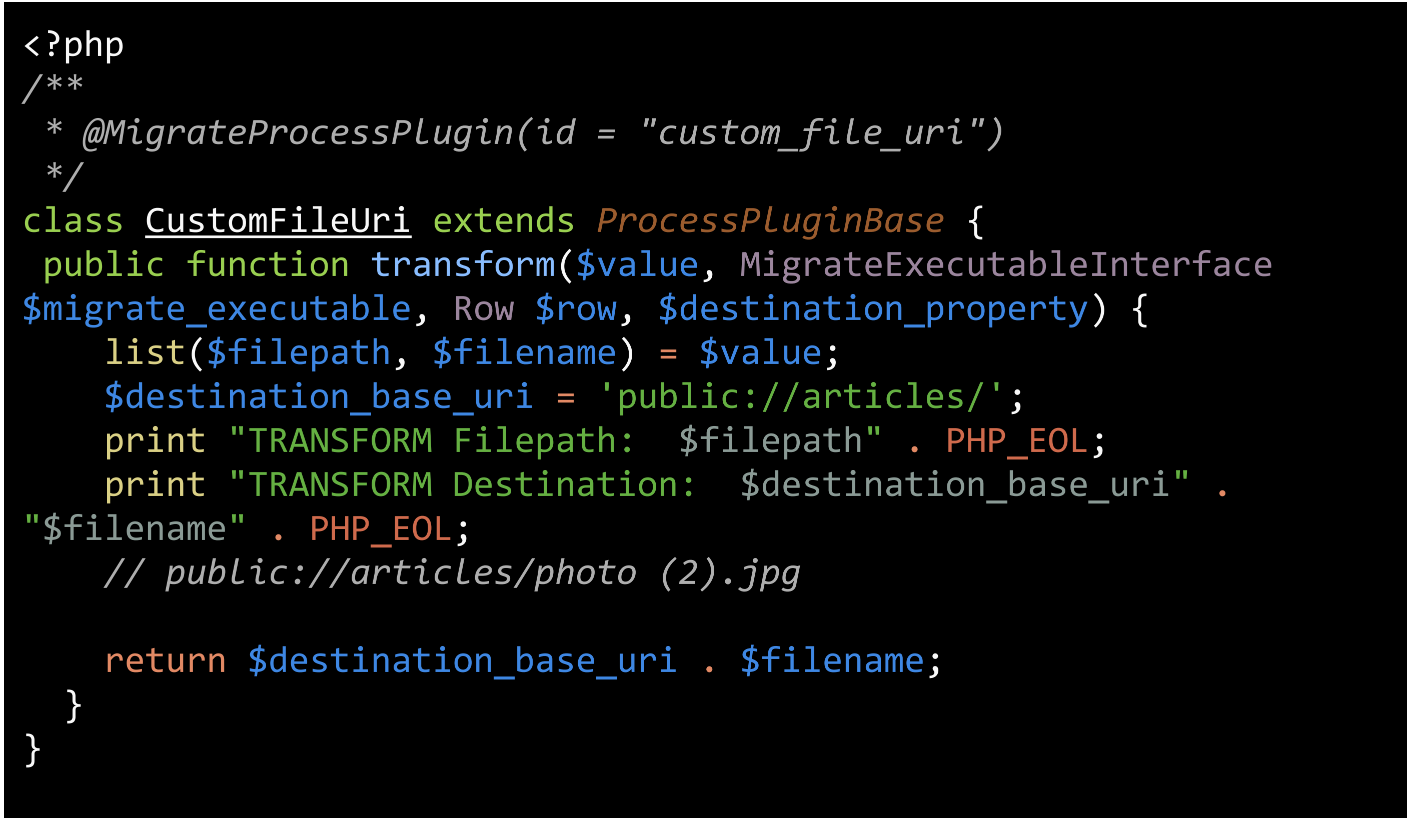


```
// In CustomFiles.php. source plugin.
public function prepareRow(Row $row) {
   // Set the complete external path to the image.
   $local_path = '/var/www/webroot/files/image/attachments/';
   $attachment = $row->getSourceProperty('path');
   $dir = $row->getSourceProperty('dir') . "/";

   $filepath = $local_path . $dir . $attachment;
   $row->setSourceProperty('path', $filepath);

   $file_name = basename($attachment);
   // Set filename. Not OK in every origin row.
   $row->setSourceProperty('name', $file_name);

   return parent::prepareRow($row);
 }
```

3.

MIGRATION FRAMEWORK

4.- PROCESS



```php
<?php
/**
 * @MigrateProcessPlugin(id = "custom_file_uri")
 */
class CustomFileUri extends ProcessPluginBase {
 public function transform($value, MigrateExecutableInterface
$migrate_executable, Row $row, $destination_property) {
    list($filepath, $filename) = $value;
    $destination_base_uri = 'public://articles/';
    print "TRANSFORM Filepath:  $filepath" . PHP_EOL;
    print "TRANSFORM Destination:  $destination_base_uri" .
"$filename" . PHP_EOL;
    // public://articles/photo (2).jpg

    return $destination_base_uri . $filename;
  }
}
```

3.
MIGRATION FRAMEWORK

4.- PROCESS

More core plugins:

- concat
- extract
- flatten
- get
- iterator
- machine_name
- menu_link_parent
- route
- skip_on_empty
- skip_row_if_not_set
- ...

Do you need an example?

Core is full of examples!

Search in core!

- "plugin: concat"
- "plugin: get"
- ...

3.
MIGRATION
FRAMEWORK

5.- DO YOU WANT MORE?

You can take a look at:

- core/modules/migrate/src/Plugin/migrate
- core/modules/<any>/src/Plugin/migrate
- core/modules/<any>/config/migration_templates

Or search in core:

- “@MigrateSource(“
- “destination:”
- “migration_tags:”

Drupal 8 core is

full of examples

Really!

4.

PERFORMANCE TIPS

0. ASK YOU FIRST

Do I have to migrate everything?

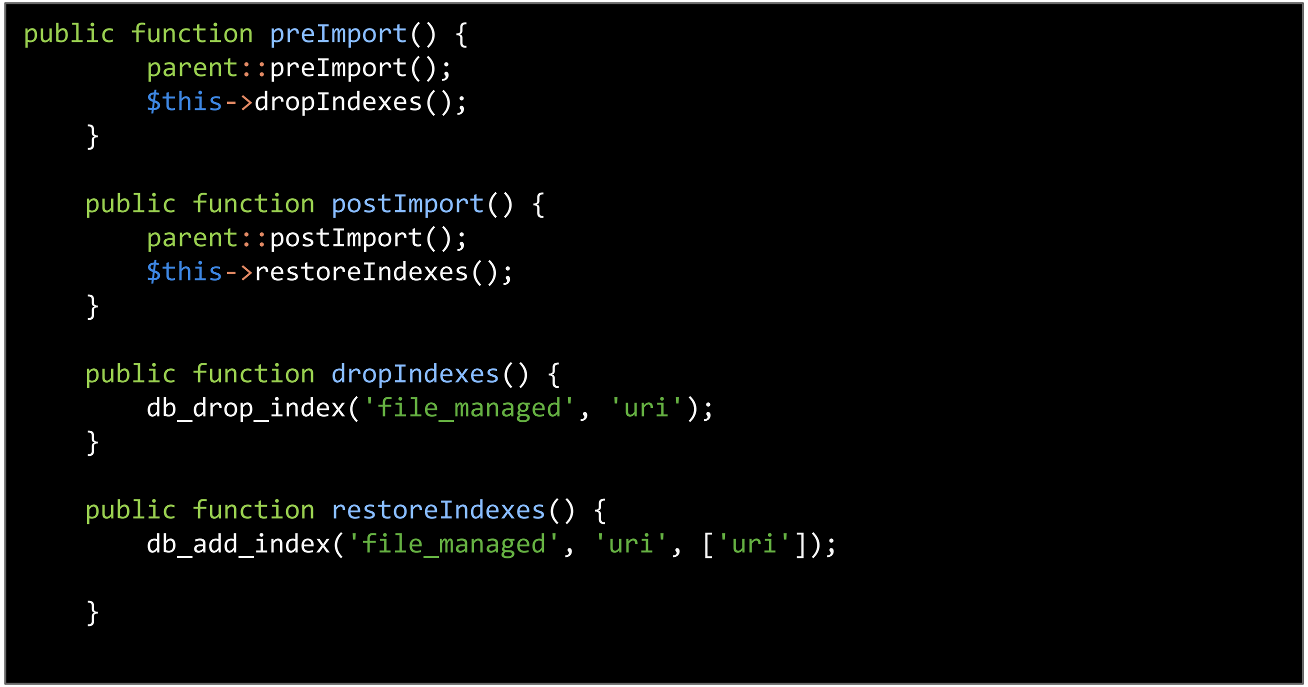
Is this migration fast enough?

1. DROPPING INDEXES

4.

PERF TIPS

- On every INSERT, indexes are rebuilt.



```
public function preImport() {
        parent::preImport();
        $this->dropIndexes();
    }

    public function postImport() {
        parent::postImport();
        $this->restoreIndexes();
    }

    public function dropIndexes() {
        db_drop_index('file_managed', 'uri');
    }

    public function restoreIndexes() {
        db_add_index('file_managed', 'uri', ['uri']);

    }
```

2. HARDWARE IMPROVEMENTS

- SSD.
- TMPFS.
- RAID 10.
- Better CPU.

4.

PERF TIPS

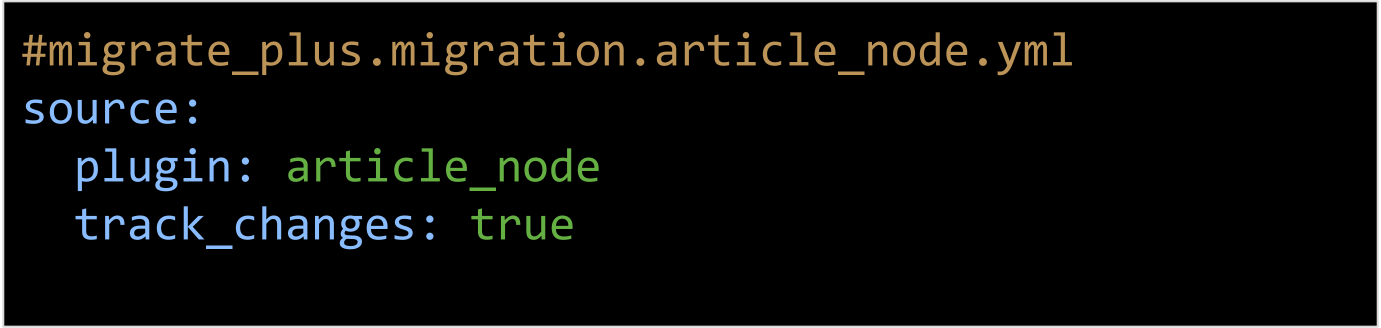
2. DISABLING FUNCTIONALITIES

Can you disable

- Modules?
- Hooks?

3. INCREMENTAL MIGRATION

track_changes: true



```
#migrate_plus.migration.article_node.yml
source:
  plugin: article_node
  track_changes: true
```

4. MULTITHREADED MIGRATION

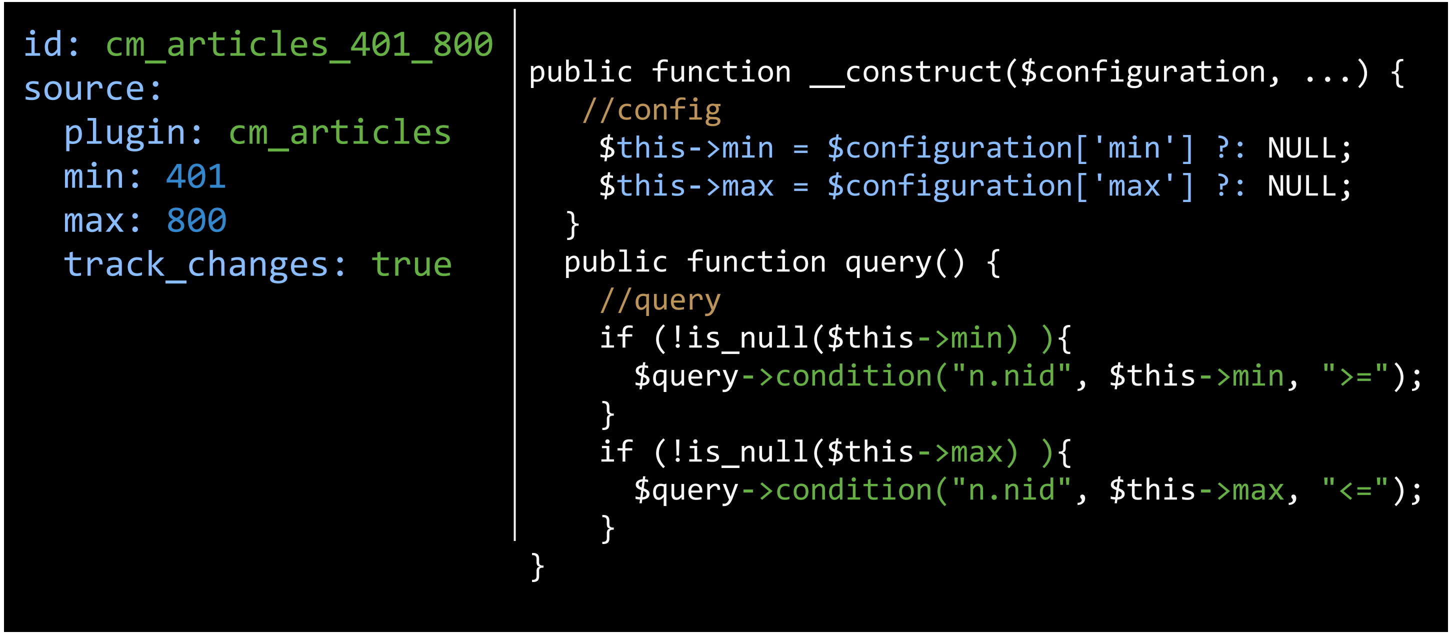
Only 1 CPU at 100%, but I have more idle

- Multithreaded php to execute drush
- “Multi-divided migration”
    - article_1_400
    - article_401_800
    - ...

4. MULTITHREADED MIGRATION

4. PERF TIPS

- “Multi-divided migration”



```yaml
id: cm_articles_401_800
source:
  plugin: cm_articles
  min: 401
  max: 800
  track_changes: true
```

```php
public function __construct($configuration, ...) {
   //config
    $this->min = $configuration['min'] ?: NULL;
    $this->max = $configuration['max'] ?: NULL;
  }
  public function query() {
    //query
    if (!is_null($this->min) ){
      $query->condition("n.nid", $this->min, ">=");
    }
    if (!is_null($this->max) ){
      $query->condition("n.nid", $this->max, "<=");
    }
}
```

You can try by clone `https://github.com/isholgueras/custom_migration`

4. MULTITHREADED MIGRATION

4.

PERF TIPS

```
$ drush mi article_1_1000 && \
  drush mi article_1001_2000 &&
...
```

You can try by clone `https://github.com/isholgueras/custom_migration`

5. MOST IMPORTANT

COMMON SENSE

QUESTIONS?

THANKS!

Helpful?
Thanks for **evaluating.**

http://bit.do/migrateDublin2016

You can contact me by @isholgueras or nacho@bluespark.com