**Magento Migration Tool**

**Definition:**

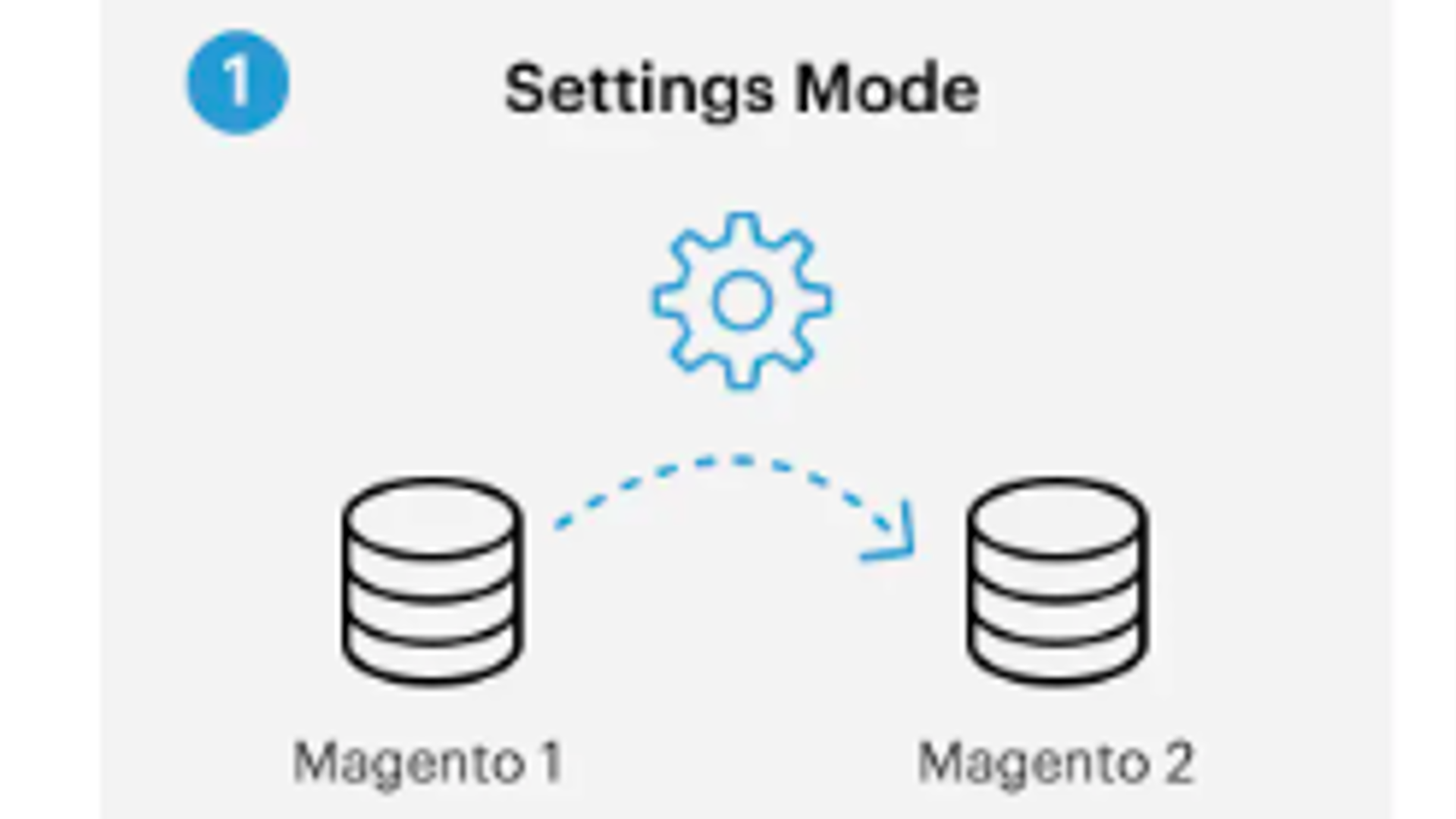
* The Data Migration Tool is a command-line interface (CLI) tool used for transferring data from Magento 1 to Magento 2.
* The Tool verifies consistency between Magento 1 and 2 database structures (tables and fields), tracks the data transfer progress, creates logs, and runs data verification tests.

**Terms:**

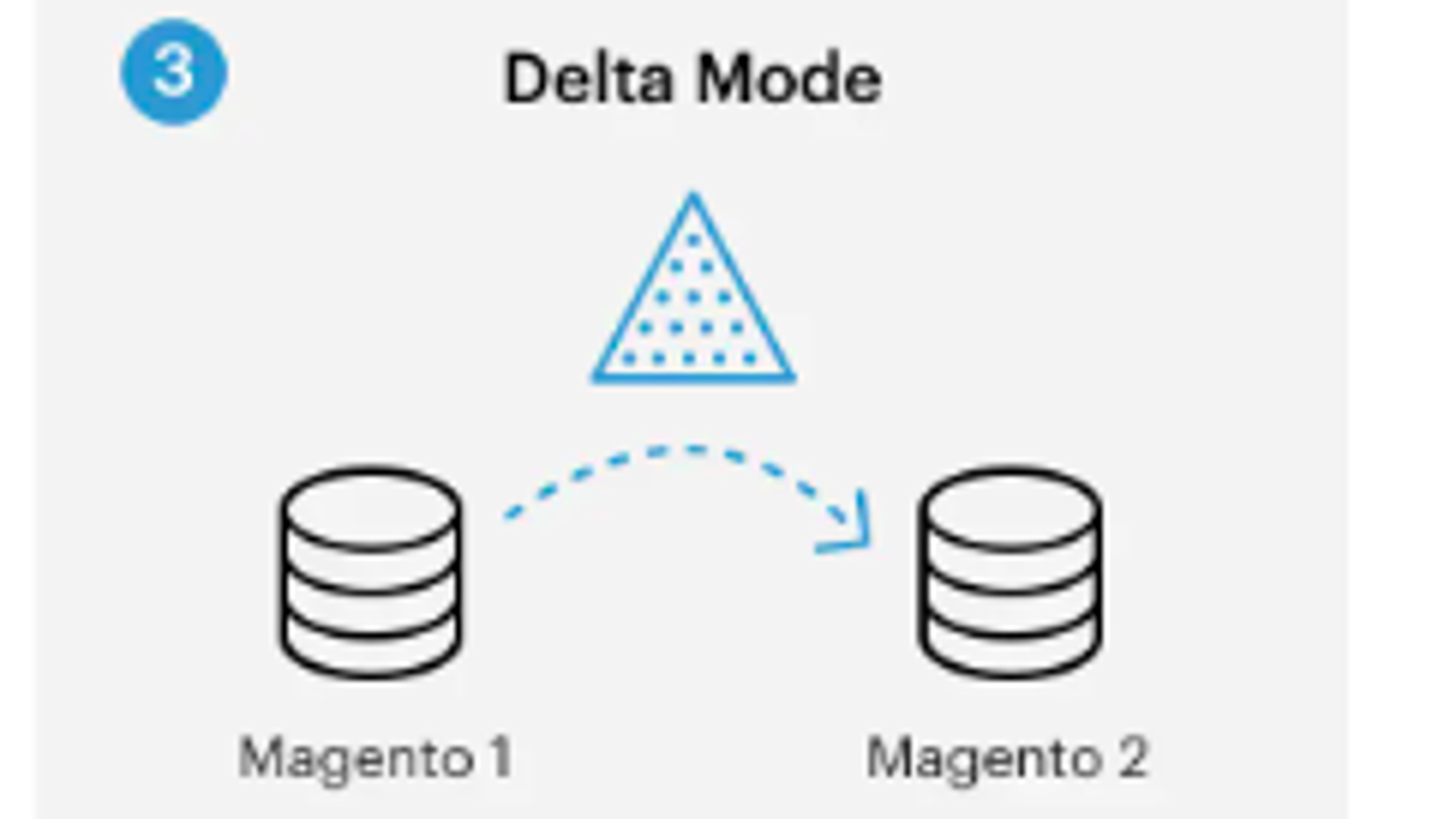
* Modes: An ordered set of operations for migrating data from Magento 1.X to Magento 2.X
* Steps: The tasks in the mode define the kinds of data to migrate
* Stages: The tasks in the step that validate, transfer, and verify the data.
* Map files: XML files that define the rules and connections between Magento 1.X and Magento 2.X data structures for completing the stages.

**Modes:**

* They are 3 types of modes in order to transfer data and adapt from Magento 1.X to Magento 2.X. The three modes are listed here
* **Setting mode:** Migration the system configuration and website related settings
* **Data Mode:** Migrates database assets in bulk
* **Delta Mode:** Migrates incremental changes (changes since the last run) and such as new customers and orders

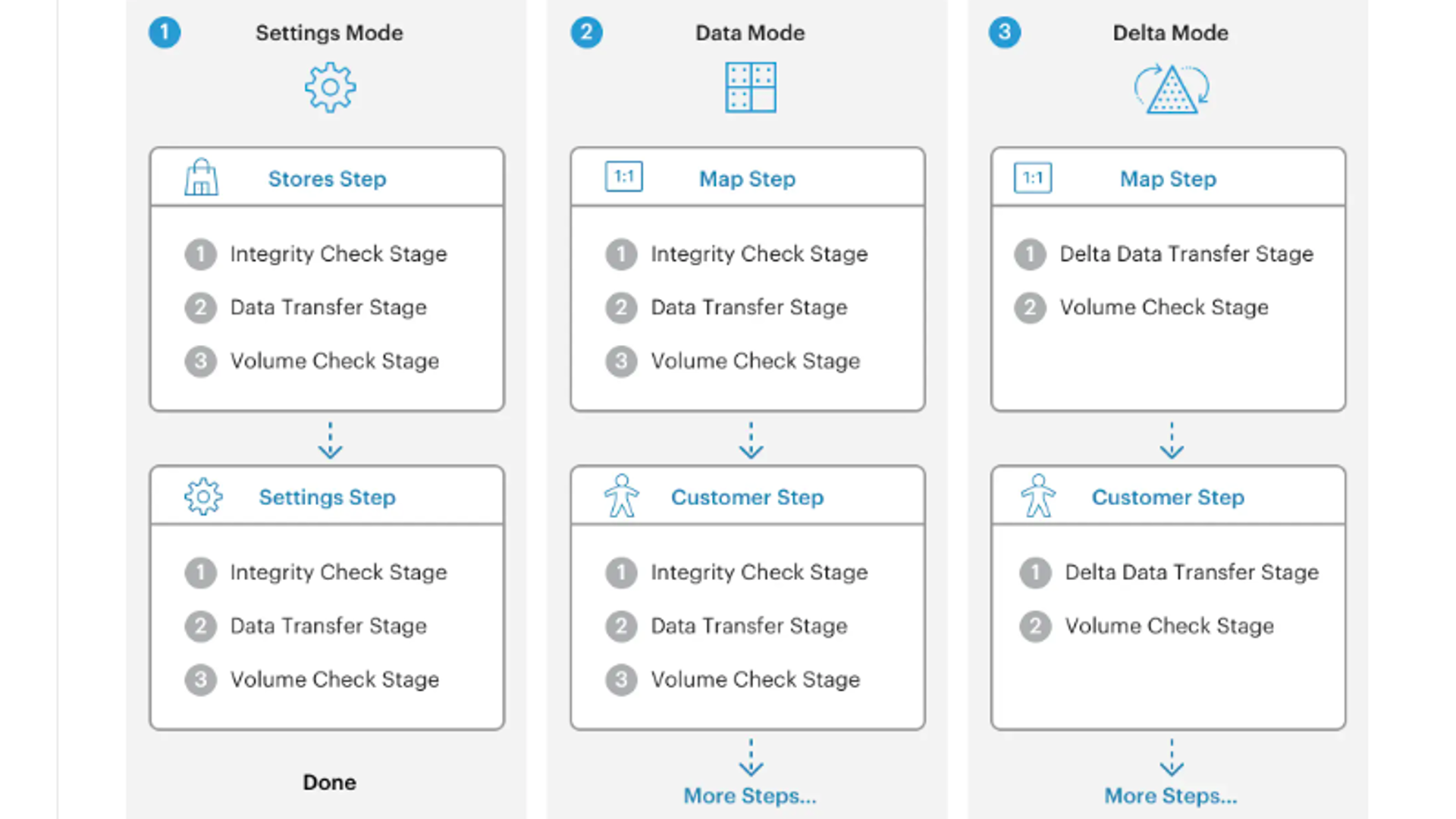






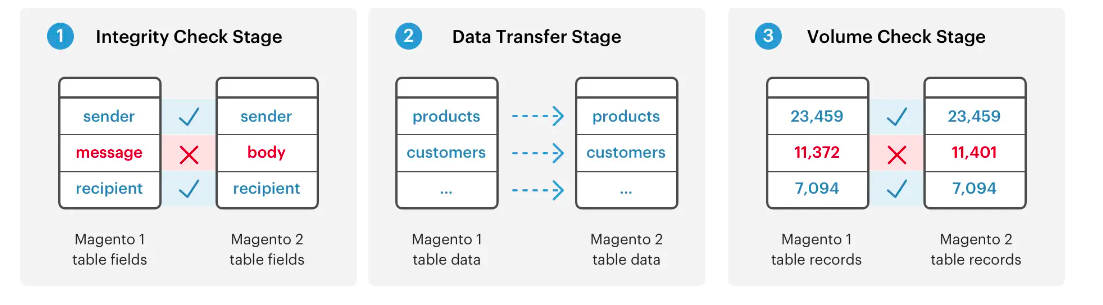
Steps:

In setting mode, it follows two steps,



**Stages:**

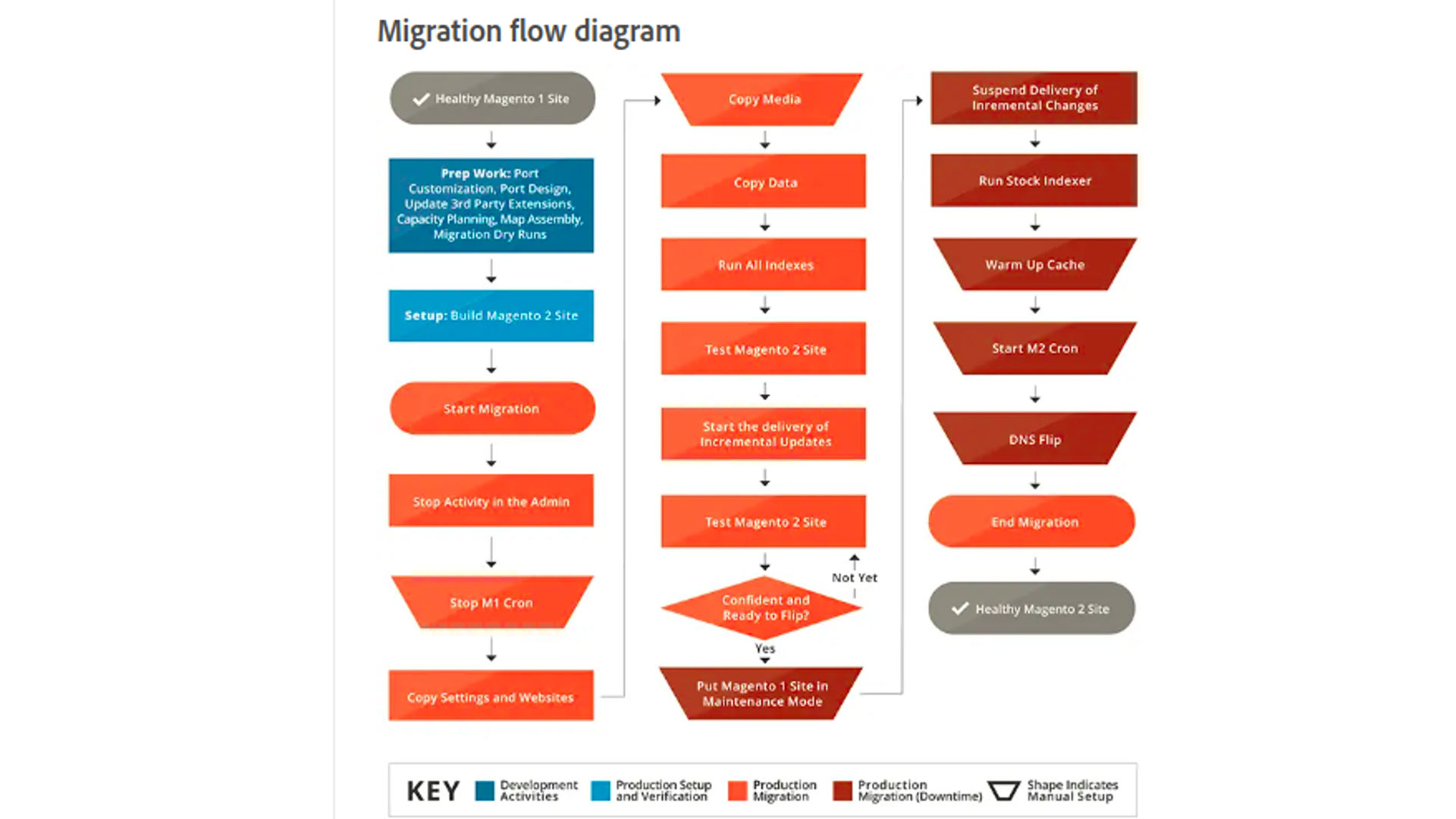
* **Integrity Check:** Compares the table field names, types, and other info to verify compatibility between Magento 1 and 2 data structures.
* **Data Transfer:** Transfers the data table by table from Magento 1 and 2
* **Volume Check:** Compares the number of records between tables to verify that the transfer was successful.



**Map Files:**

* At the lowest level of the migration processes are the XML *map files*.
* The Data Migration Tool uses map files within the stages of a step to transform different data structures between the Magento 1.x and 2.x tables.
* If there are no differences in data structure or data format, the Data Migration Tool transfers it as-is, including data from tables created by extensions to the Magento 2 database.
* When differences are not declared in map files, then the Data Migration Tool displays an error and does not start.

**Migration flow diagram**



**Migration Components:**

* Data
* Extension and custom code
* Themes and customization

**Data Migration best practices:**

* Use a copy of the database from a Magento 1 instance when performing migration testing. Do not use the production instance of your Magento 1 store database.
* Remove outdated and redundant data from your Magento 1 database before migration.
* Such data may include logs, order quotes, recently viewed or compared products, visitors, event-specific categories, and promotional rulesTo boost performance, enable the direct\_document\_copy option in your config.xml file:

**Supported versions**

* Adobe Commerce:1.11.x, 1.12.x, 1.13.x, 1.14.x
* Magento Open source: 1.6x, 1.7x, 1.8x, 1.9x
* If we migrate from Magento Open source to Adobe commerce the supported versions are 1.6x, 1.7x, 1.8x, 1.9x

**Migration tool prerequirements:**

**Magento system requirement:**

* Set up your Magento 2 system so that it meets the system requirements
* Use a topology and design that at least matches your existing Magento 1 system.
* Install Magento 2

**Cron:**

* Do not start Magento 2 cron jobs.

**Database:**

* After Installation backup your Magento 2 database.
* This allows you to restore the initial database state if migration is not successful
* Verify the data migration tool has network to access to connect the Magento 1 and Magento 2 databases
* If Binary Logging is enabled for your Magento 1 database, set the global [log\_bin\_trust\_function\_creators](https://dev.mysql.com/doc/refman/5.7/en/server-system-variables.html#sysvar_log_bin_trust_function_creators) MySQL system variable to 1, or grant the [SUPER privilege](https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#priv_super) to your account.
* We do not recommend creating new entities (products, categories, and attributes) in your Magento 2 store before migration because the Data Migration Tool overwrites such new entities with the old ones from Magento 1.

**Install Migration tool**

* Make sure you are using *the same released version* of both Magento 2 and the Data Migration Tool. For example, for Magento version 2.2.0, you must also use the Data Migration Tool version 2.2.0.

**Types of installation:**

* Git repository
* Composer

**Install by Composer**

* To install the Data Migration Tool, you must update composer.json in the Magento root installation directory to provide the location of the Data Migration Tool package.

1. Log in to your application server as, or switch to, the file system owner.
2. Change to the application root directory.
3. Enter the following commands:

**For 2.4.6 version**

* composer config repositories.magento composer <https://repo.magento.com>
* composer require magento/data-migration-tool:2.4.6
* When prompted, enter your authentication keys. Your public key is your username; your private key is your password.

**Install from GitHub**

* Log in to your application server as, or switch to, the file system owner.
* Change to the application root directory.
* Enter the following commands:

composer config repositories.data-migration-tool git https://github.com/magento/data-migration-tool

-------------------------------------------------------

composer require magento/data-migration-tool:<version>

----------------------------------------------------

For example, version 2.4.6

composer config repositories.data-migration-tool git https://github.com/magento/data-migration-tool

---------------------------------------------------------

composer require magento/data-migration-tool:2.4.6

**Check versions of installed migration tool**

* Change to your Data Migration Tool directory: <vendor>/magento/data-migration-tool.
* Open composer.json in a text editor.
* The version entry in that file is the version of the Data Migration Tool.

**Upgrade Magento Migration tool**

* Before you upgrade the Data Migration Tool, back up at least the vendor/magento/data-migration-tool directory.
* During upgrade, it could be deleted and replaced by the updated code.
* You can also back up the entire Magento codebase and database using the following command:

php <magento\_root>/bin/magento setup:backup --code --db

**WARNING**

The vendor/magento/data-migration-tool directory contains your custom code. Failure to back it up means you can lose your customizations during upgrade.

* Log in to your application server as, or switch to, the file system owner.
* Change to the application root directory.
* Enter the following commands:

composer require magento/data-migration-tool:<version>

**Configure Data Migration tool:**

There are two ways to configure the Data Migration Tool:

* Configure the Data Migration Tool in a separate module (recommended)
* Change the Data Migration Tool configuration in the <your Magento 2 install dir>/vendor/magento/data-migration-tool/etc/ directory.
* To use source control to manage your migration configuration and use it for deployment, you must create a separate module.
* If you plan to run the Data Migration Tool locally only, you can edit files in the <your Magento 2 install dir>/vendor/magento/data-migration-tool/ directory directly.

**Configure migration in separate module:**

Create a Magento 2 module.

<your Magento 2 install dir>/app/code/Vendor/Migration/composer.json

{

"name": "vendor/migration",

"description": "Providing config for migration",

"config": {

"sort-packages": true

},

"require": {

"magento/framework": "\*",

"magento/data-migration-tool": "\*"

},

"type": "magento2-module",

"autoload": {

"files": [

"registration.php"

],

"psr-4": {

"Vendor\\Migration\\": ""

}

},

"version": "1.0.0"

}

* <your Magento 2 install dir>/app/code/Vendor/Migration/registration.php
* <your Magento 2 install dir>/app/code/Vendor/Migration/etc/module.xml
* Copy the config.xml.dist configuration file from the appropriate directory of the Data Migration Tool (<your Magento 2 install dir>/vendor/magento/data-migration-tool/etc/<migration edition>/<ce or version>) into the <your Magento 2 install dir>/app/code/Vendor/Migration/etc/<migration edition>/<ce or version>/config.xml file.
* cd <your Magento 2 install dir>
* cp vendor/magento/data-migration-tool/etc/opensource-to-opensource/1.9.3.6/config.xml.dist app/code/Vendor/Migration/etc/opensource-to-opensource/1.9.3.6/config.xml
* In the config.xml file, you must set access details to M1 and M2 databases and encryption key.
* If your M1 store has custom changes, you should map the rest of your configuration files to your Magento 1 store customizations. See Work with configuration and mapping files.

Configure migration in vendor folder:

<your Magento 2 install dir>/vendor/magento/data-migration-tool/etc/<migration edition>/<ce or version>

cp config.xml.dist config.xml

* Open config.xml in a text editor.
* At a minimum, the config.xml file must contain access details to M1 and M2 databases and encryption keys.

Work with configuration and mapping files:

* Changing table names
* Changing field names
* Ignoring tables or fields
* Adapt transferring data of a field to Magento 2 format

Mapping file name with is description

