How to use image-migrator

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Version | Date | Description | Modified By | Document Name |
| 1 | V0.1 | September 27 2023 | Draft version | Armando Ponce Soriano | How to use image-migrator |
| 2 | V0.2 | October 3  2023 | Added FAQ section | Armando Ponce Soriano | How to use image-migrator |
| 3 | V1.0 | October 30 2023 | Minor corrections | Armando Ponce Soriano | How to use image-migrator |

# Overview

There are migration scenarios where a customer that has saved all the images of their applications in repositories like Docker Hub wants to mobilize them to a cloud native solution, in these cases it is recommended the use of the following tool: **image-migrator.**

**image-migrator** is an image migration tool that can automatically migrate images from the Docker image repository built on Docker Registry v2 to SWR.

# Constraints

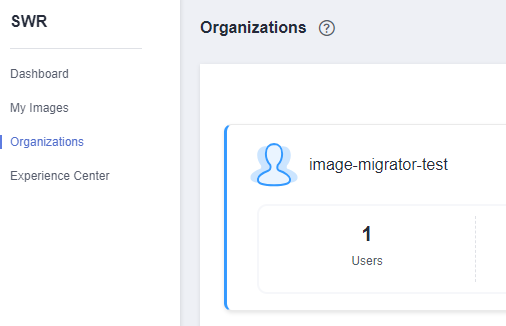
* Login credentials for SWR last only 24 hours by default, if you want to have longer credentials refer to the following: [Obtaining a Long-Term Valid Login Command\_Software Repository for Container\_User Guide\_Image Management\_Huawei Cloud](https://support.huaweicloud.com/intl/en-us/usermanual-swr/swr_01_1000.html)
* Image repositories and tagged images must comply with the Docker Registry v2 requisite, otherwise it won’t be possible to make the migration.

# Validation Steps

## **Prerequisites**

Before using image-migrator, there is one requisite that must be fulfilled, this is:

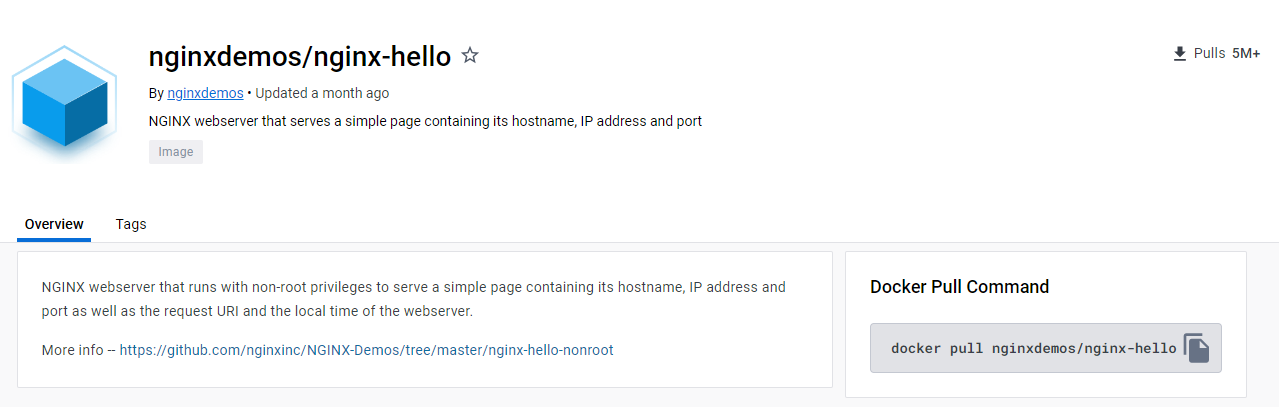
* Have at least one SWR organization created on HWC console



## **Using image-migrator**

1. Locate the image you want to migrate, for this guide we will use the **nginxdemos/nginx-hello** image from Docker Hub as an example

(<https://hub.docker.com/r/nginxdemos/nginx-hello>)



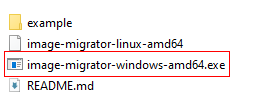
1. Download the image-migrator tool from the following link:

<https://ucs-migration-intl.obs.ap-southeast-3.myhuaweicloud.com/toolkits/image-migrator-23.3.0.0323215042.tar.gz>

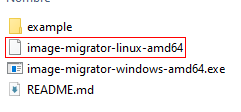
1. Uncompress the tool and access to the folder



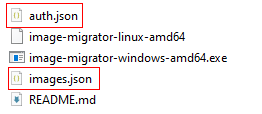
For Windows we will use **image-migrator-windows-amd64.exe**



For Linux we will use **image-migrator-linux-amd64**

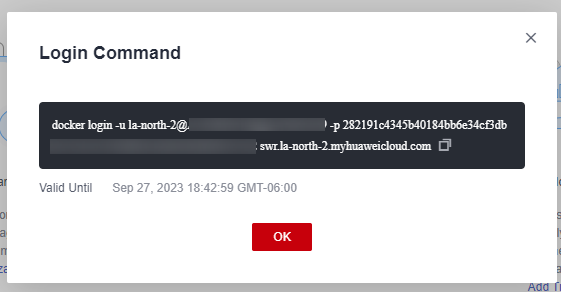


1. Create two files in the same directory named: **auth.json** & **images.json**:



1. On HWC console, generate a login command on SWR for credentials that will be used later on **auth.json**





1. Fill **auth.json** by following the template:

{

"Source image repository address": { },

"Destination image repository address": {

"username": "xxxxxx",

"password": "xxxxxx",

"insecure": true

}

}



****

**Image-migrator** already kwnows that the image repositories are on Docker Hub, so the source repository section has to follow the docker pull/push command format (**registry/namespace/repository**)

1. Fill **images.json** by follow the template:

{

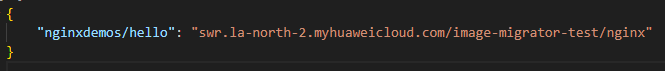
"Source image repository address": "SWR destination repository",

…

…

…

}



****

If source repository section has no tag attribute, **image-migrator** will try to migrate all the tagged images into the detination repository.

1. Once configurated the files, open a terminal and use the following command to start the migration process:

**For** **Windows**:

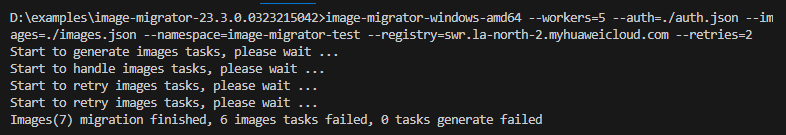
image-migrator-windows-amd64 --workers=<Number of concurrent processes> --auth=<Path to auth.json file> --images= <Path to images.json file> --namespace=<namespace of destination repository> --registry=<default registry of the destination repository> --retries=<number of retries in case of errors>

**For Linux**:

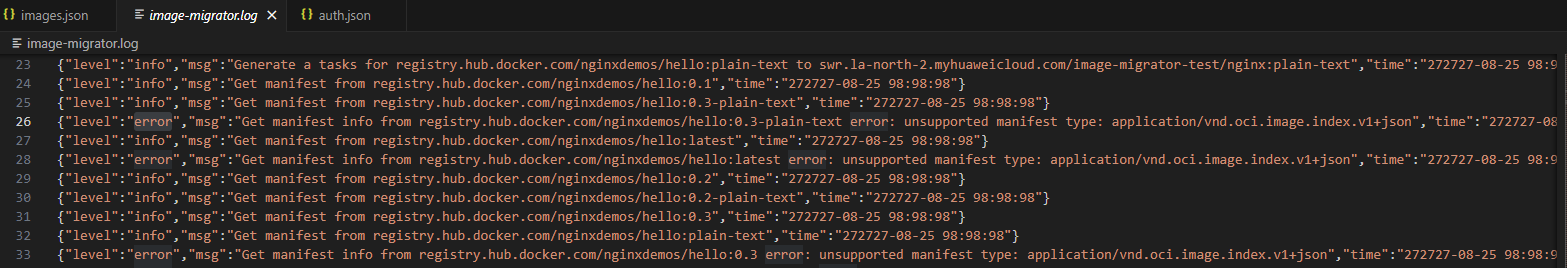
./image-migrator-linux-amd64 --workers=<Number of concurrent processes> --auth=<Path to auth.json file> --images= <Path to images.json file> --namespace=<namespace of destination repository> --registry=<default registry of the destination repository> --retries=<number of retries in case of errors>

**Where**:

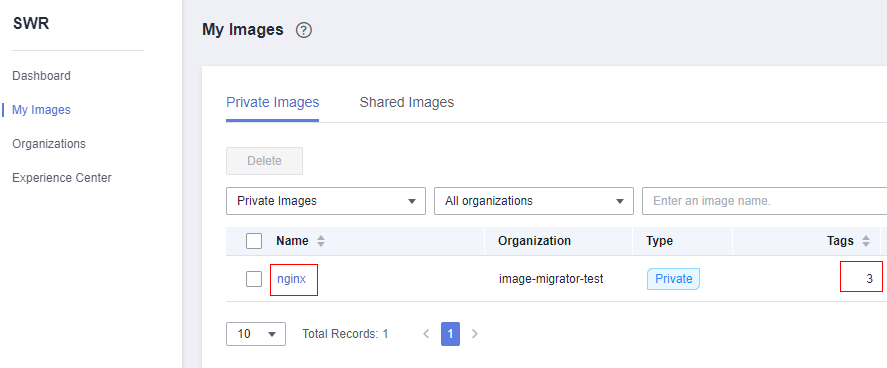
* --**auth**: specifies the path of auth.json.
* --**images**: specifies the path of images.json.
* --**namespace**: specifies the default namespace of the destination repository.
* --**registry**: specifies the default registry of the destination repository.
* --**retries**: specifies the number of retry times when the migration fails. The default value is 3.
* --**workers**: specifies the number of **goroutines** for image migration. The default value is 7.

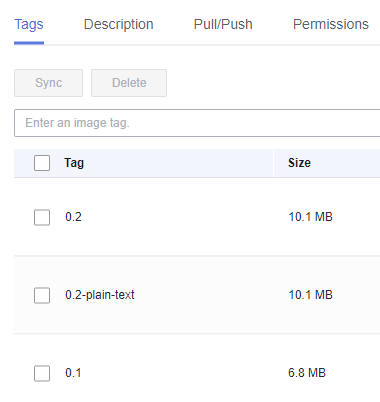


When an image migration fails, we can check the reason by entering the **image-migrator.log** file



1. Lastly, if we check our image repositories on **SWR** console, we will notice our successful process.





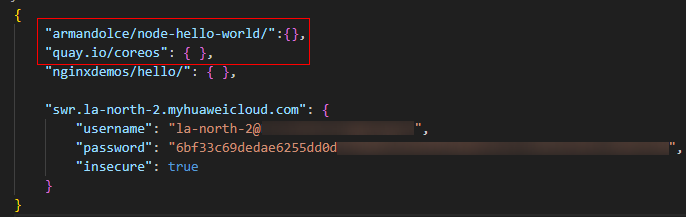
# FAQ

* **Q**: It is possible to migrate more than one image repository at one?

**A**: Yes, image-migrator supports multi-repository migrations. To achieve this, it’s necessary to modify our **Auth.json** and **Images.json** files:

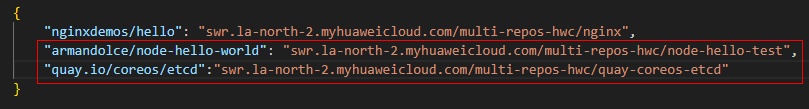
**auth.json:**

Add the source repositories you want to migrate

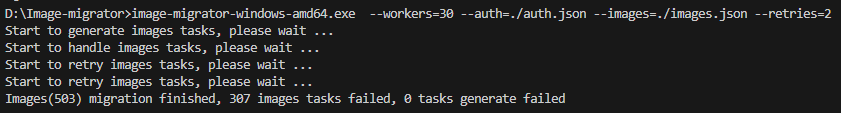


**images.json:**

Add the source repositories and their target repositories, always checking that the information is the same as the **auth.json** file.



Once finished, you can check the status of the migration task on terminal or on SWR.





* **Q**: If I increase the number on the workers parameter before execution, will my task be faster?

**A**: Theoretically, yes. But in a real-life scenario, the workers parameter only indicates the number of **goroutines** that the program will assign for the task. This number can be as large as you wish, but the performance may not increase significantly after a value greater than the number of CPU cores your machine has.

****

A **goroutine** is a thread implemented in the **Go** programming language.

# Appendix

If you need more information about **image-migrator**, refer to the following documentation:

[Image Migration\_Ubiquitous Cloud Native Service\_User Guide\_Container Migration\_Migration from Clusters in an On-premises IDC to the Cloud\_Huawei Cloud](https://support.huaweicloud.com/intl/en-us/usermanual-ucs/ucs_01_0175.html)