

Keycloak in Docker #6 – How to import realms from a directory

 3rd February 2022 /  little_pinecone /  Tools

If we want to import multiple Keycloak realms, or realm resources are split into multiple files, we need to execute a directory import at boot time. Fortunately, running a Keycloak service with Docker makes this task easy.

Prerequisites

- [Docker Engine](#) and [Docker Compose](#) installed on your machine.
- If this is your first attempt to run Keycloak in Docker, I recommend reading the post [Keycloak in Docker # 1 – How to run Keycloak in a Docker container](#), as I explained the basic configuration there.
- You'll need at least one realm that was exported to a directory. You can learn how to do it in the [Keycloak in Docker #5 – How to export a realm with users and secrets](#) post.

Configure importing Keycloak realms from a directory

The example Docker Compose configuration used below is available in the <https://gist.github.com/little-pinecone/6b52ccd1fc0296b267e810d43fd01f3b> GitHub gist.

The importance of the naming convention

As we can read in the docs:



<REALM_NAME>-users-<INDEX>.json. For example, "acme-roadrunner-affairs-users-0.json" for the first user's file of the realm named "acme-roadrunner-affairs".

https://www.keycloak.org/docs/16.1/server_admin/#assembly-exporting-importing_server_administration_guide

Make sure the resources for your realms follow this naming convention. Otherwise, some or all of the files may be skipped completely from import.

Add a Docker volume for the imported resources

First, I'm going to create a Docker **volume** to make the import assets available in the **/tmp/import** directory in the container. Below you'll find the relevant configuration from my **docker-compose.yml** file:

```
1 | version: '3.3'
2 | services:
3 |   keycloak:
4 |     ...
5 |   volumes:
6 |     - ./keycloak/realms/import:/tmp/import
7 |     ...
```

Provide the required options when running the container

Next, I'm going to add the minimum configuration required to perform the directory import to my **docker-compose.yml** file:

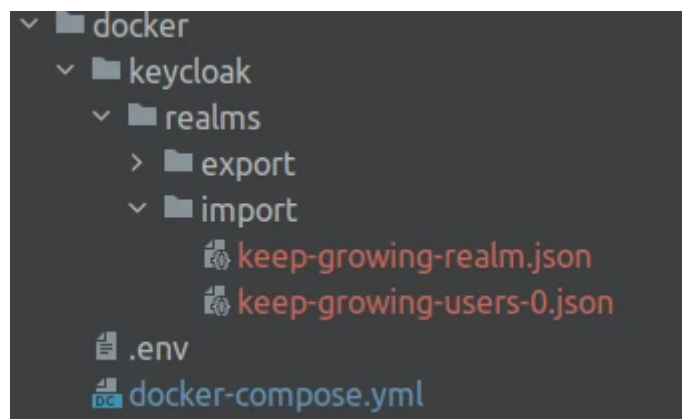
```
1 | services:
2 |   keycloak:
3 |     ...
4 |   volumes:
```



This configuration will overwrite existing realms by default. For other `keycloak.migration.X` options, see the official Keycloak documentation on [importing and exporting the database](#).

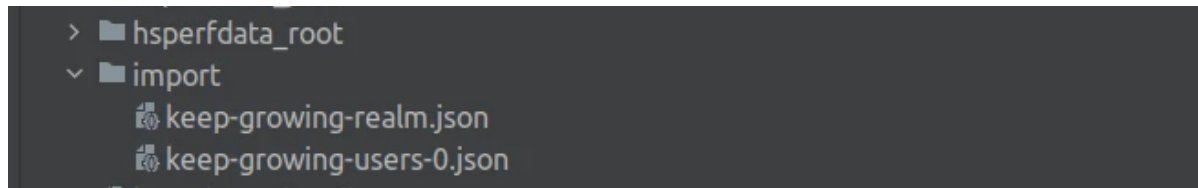
Import a Keycloak realm that has been exported in multiple files

Below you can see the starting point for importing my **keep-growing** realm:



To summarize, I have one file with the realm configuration and one with the associated users. With the Docker Compose configuration described in the previous section, I'm going to start the container using the **docker-compose up -d** command.

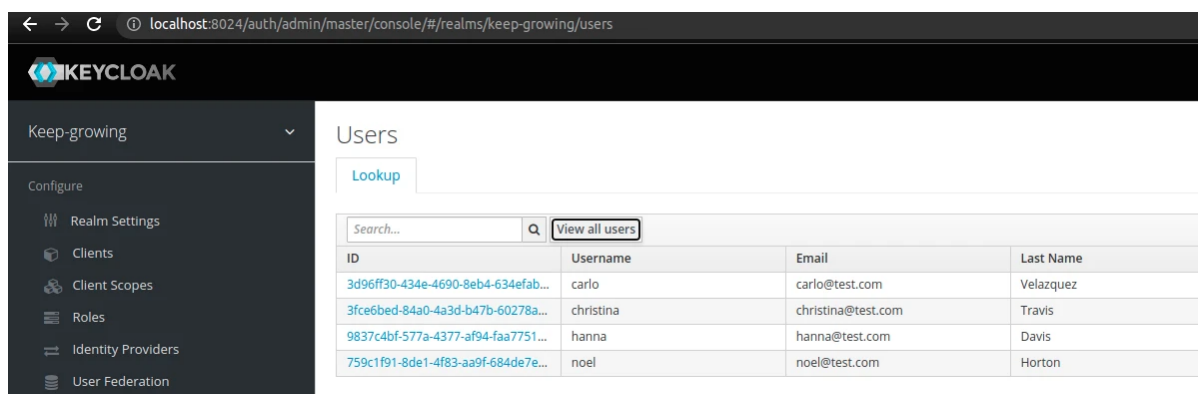
As a result, we can see in the screenshot below that the volume with realm resources was mapped properly:



Furthermore, the container logs contain entries documenting a successful import:

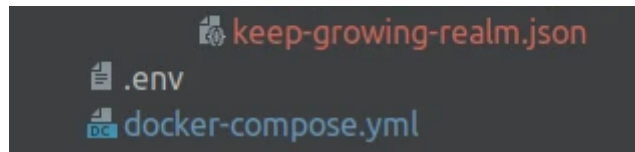
```
1 INFO [org.keycloak.exportimport.dir.DirImportProvider] (ServerService Thread
2 Pool -- 62) Importing from directory /tmp/import
3 ...
4 INFO [org.keycloak.services] (ServerService Thread Pool -- 62) KC-SERVICES005
5 0: Initializing master realm
6 INFO [org.keycloak.services] (ServerService Thread Pool -- 56) KC-SERVICES003
7 0: Full model import requested. Strategy: OVERWRITE_EXISTING
8 INFO [org.keycloak.exportimport.util.ImportUtils] (ServerService Thread Pool
9 -- 56) Realm 'keep-growing' imported
10 INFO [org.keycloak.exportimport.dir.DirImportProvider] (ServerService Thread
11 Pool -- 56) Imported users from /tmp/import/keep-growing-users-0.json
12 INFO [org.keycloak.services] (ServerService Thread Pool -- 56) KC-SERVICES003
13 2: Import finished successfully
```

At last, we can examine the imported realm in the [Keycloak Admin Console](#):



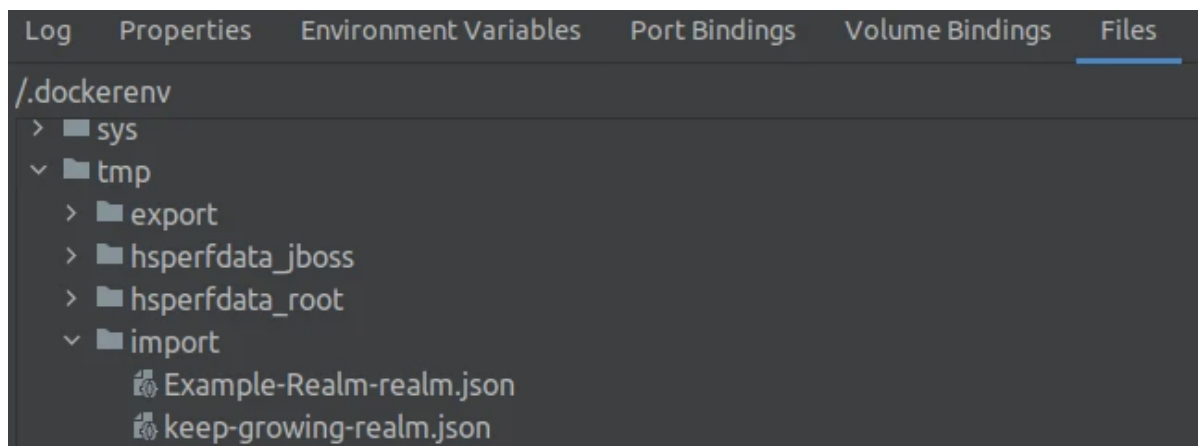
Import multiple Keycloak realms

Below you can see the starting point for importing my **keep-growing** and **Example-Realm** realms:



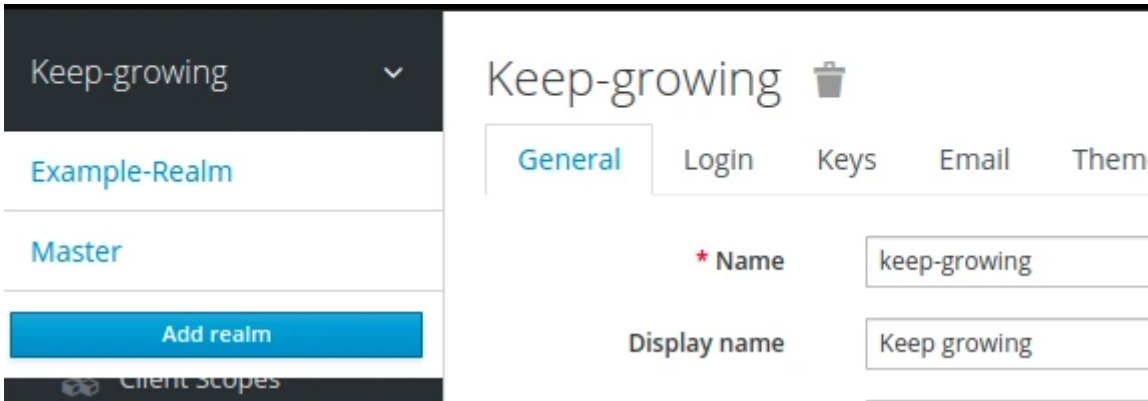
To summarize, I have two files, one for each realm. With the Docker Compose configuration described in the previous section, I'm going to start the container using the `docker-compose up -d` command.

Consequently, we can see in the screenshot below that the volume with realms is available in the container:



Furthermore, the container logs contain entries documenting successful imports:

```
1 INFO [org.keycloak.exportimport.dir.DirImportProvider] (ServerService Thread
2 Pool -- 62) Importing from directory /tmp/import
3 ...
4 INFO [org.keycloak.services] (ServerService Thread Pool -- 62) KC-SERVICES005
5 0: Initializing master realm
6 INFO [org.keycloak.services] (ServerService Thread Pool -- 62) KC-SERVICES003
7 0: Full model import requested. Strategy: OVERWRITE_EXISTING
  INFO [org.keycloak.exportimport.util.ImportUtils] (ServerService Thread Pool
  -- 62) Realm 'keep-growing' imported
  INFO [org.keycloak.exportimport.util.ImportUtils] (ServerService Thread Pool
  -- 62) Realm 'Example-Realm' imported
  INFO [org.keycloak.services] (ServerService Thread Pool -- 62) KC-SERVICES003
  2: Import finished successfully
```



Read more on Keycloak directory import

- In addition to this article, see the Keycloak documentation on [importing and exporting the database](#).
- Additionally, you can visit the [How to import Multiple realms in keycloak?](#) and [Missing users from Keycloak Realm after import](#) issues on StackOverflow.
- In case you need to import realm from a single file, see the [Keycloak in Docker #2 – How to import a Keycloak realm](#) post.

Photo by [Vlada Karpovich](#) from [Pexels](#)

[Docker](#) [Keycloak](#) [user management](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

B

I

U

:

1/3

↔

↔

“

✂

📄

📄

🔗

✖





Name *

Email *

Website

☐ By using this form you agree with the storage and handling of your data by this website. *

POST COMMENT

Created by [Marta Szymek](#)

[Privacy Policy](#)