

Keycloak in Docker #1 – How to run Keycloak in a Docker container

📅 20th August 2021 / 👤 little_pinecone / 📁 Tools

Keycloak is an open source project that is a convenient option for delegating authentication and user management. It allows us to focus more on delivering business value to our projects. A proper Docker configuration for this service will come in handy whenever we need to run it locally.

Prerequisites

- **Docker** installed on your machine. You can check the installed version with the `docker --version` command. The command's output on my environment at the moment of writing this post is as follows:

```
Docker version 20.10.7
```

- The starting point for the work presented in this post is contained in the [commit 34f8178e23b3bb05da30a7b247cd5f3dc4110c4d](https://github.com/littlepinecone/keycloak-docker/commit/34f8178e23b3bb05da30a7b247cd5f3dc4110c4d).

How to configure Docker containers for Keycloak

We're going to create two services: **keycloak** from the [jboss/keycloak image](#) and **keycloakdb** from the [postgres image](#). First, we need to create the `docker-compose-keycloak.yml` file:

```
1 version: '3.3'
2 services:
3   keycloak:
4     image: jboss/keycloak:${KEYCLOAK_VERSION}
5     ports:
```



```
11     - DB_USER=${KEYCLOAK_DATABASE_USER}
12     - DB_PASSWORD=${KEYCLOAK_DATABASE_PASSWORD}
13     - DB_ADDR=${KEYCLOAK_DATABASE_HOST}
14     - DB_VENDOR=${KEYCLOAK_DATABASE_VENDOR}
15   networks:
16     internal:
17   depends_on:
18     - keycloakdb
19
20   keycloakdb:
21     image: postgres:${POSTGRES_VERSION}
22     ports:
23       - "5433:5432"
24     environment:
25       - POSTGRES_USER=${KEYCLOAK_DATABASE_USER}
26       - POSTGRES_PASSWORD=${KEYCLOAK_DATABASE_PASSWORD}
27       - POSTGRES_DB=${KEYCLOAK_DATABASE_NAME}
28     volumes:
29       - keycloak-postgres:/var/lib/postgresql/data
30     networks:
31       internal:
32
33   volumes:
34     keycloak-postgres:
35
36   networks:
37     internal:
```

Below you'll find the content of my `.env` file specifying the default environment variables for those services:

```
1 COMPOSE_PROJECT_NAME=keycloakindocker
2
3 POSTGRES_VERSION=14.1-alpine
4
5 KEYCLOAK_VERSION=15.0.2
```



```
11 KEYCLOAK_DATABASE_PASSWORD=keycloakdb
12 KEYCLOAK_DATABASE_HOST=keycloakdb
13 KEYCLOAK_DATABASE_VENDOR=postgres
```

The `COMPOSE_PROJECT_NAME` variable holds a prefix for the services names. You'll see the resulting names later in this post. You can provide the name of your project as a value for this variable.

Keycloak container configuration explained

Let's take a look at the properties configured in the `docker-compose-keycloak.yml` file.

image

The `keycloak` service will use the `jboss/keycloak` image. You can find [all available tags on the image page](#).

ports

On my machine, the `port 8080` is already taken by another service. Therefore, I'm going to [expose the keycloak service on port 8024](#) using the `HOST:CONTAINER` format.

environment

Below you'll find a short explanation for the variables that we're going to set.

KEYCLOAK_USER and KEYCLOAK_PASSWORD

We're specifying credentials to create an `admin` account in the `keycloak` service. As we can read in the image documentation:

By default there is no admin user created so you won't be able to login to the admin console. To create an admin account you need to use environment variables to pass in an initial username and password.



keep the values in files.

DB_DATABASE

An optional variable for specifying name of the database. Defaults to **keycloak**.

DB_USER

An optional variable for specifying a user used to authenticate to the database. Defaults to ''.

DB_PASSWORD

An optional variable for specifying a user password used to authenticate to the database. Defaults to ''.

DB_ADDR

An optional variable for specifying a **hostname** for the database. Because we're going to run the **keycloakdb** and **keycloak** services in a common **internal** network in Docker, we need to provide the service name here – **keycloakdb** in my example. Keycloak will add the default port 5432 to that hostname. So the resulting host will be **keycloakdb:5432**. On the other hand, you can specify a custom port by setting the **DB_PORT** variable as well.

DB_VENDOR

An optional variable for specifying a database vendor. If absent, Keycloak tries to determine the database on its own. If this process fails, Keycloak defaults to using H2. Consult the [image documentation](#) to see what values are supported. For our PostgreSQL database I'm going to use the **postgres** value.

Database container

You can find a detailed explanation on how to run a PostgreSQL database as a dockerized service in the [Set up a PostgreSQL database with Docker](#) post. Therefore, I won't be repeating the configuration and variables description in this



This image supports using H2, MySQL, PostgreSQL, MariaDB, Oracle or Microsoft SQL Server as the database.

(...) If the DB can't be detected it will default to the embedded H2 database.

<https://hub.docker.com/r/jboss/keycloak>

ports

On my machine, the **port 5432** is already taken by another service. Therefore, I'm going to **expose the keycloak database on port 5433** using the HOST:CONTAINER format.

volumes

We're going to create the named **keycloak-postgres volume** on our machine that will store content of the **/var/lib/postgresql/data** directory in the container.

Start Keycloak service as Docker containers

Finally, we're going to run the following command:

```
1 | docker-compose -f docker-compose-keycloak.yml up -d
```

The **-f option for docker-compose** is for specifying an alternate compose file (default: **docker-compose.yml**). The **-d option for docker-compose up** is for running the containers in the background.

As a result, we can see two Docker services running on our machine:

```
1 | docker ps
2 | CONTAINER ID   IMAGE                                PORTS
3 | 774a1d259bf9   jboss/keycloak:15.0.2              8443/tcp, 0.0.0.0:8024->8080/tcp, :::8
4 | 6f10181d012b   postgres:14.1-alpine              0.0.0.0:5433->5432/tcp, :::5433->5432/
```



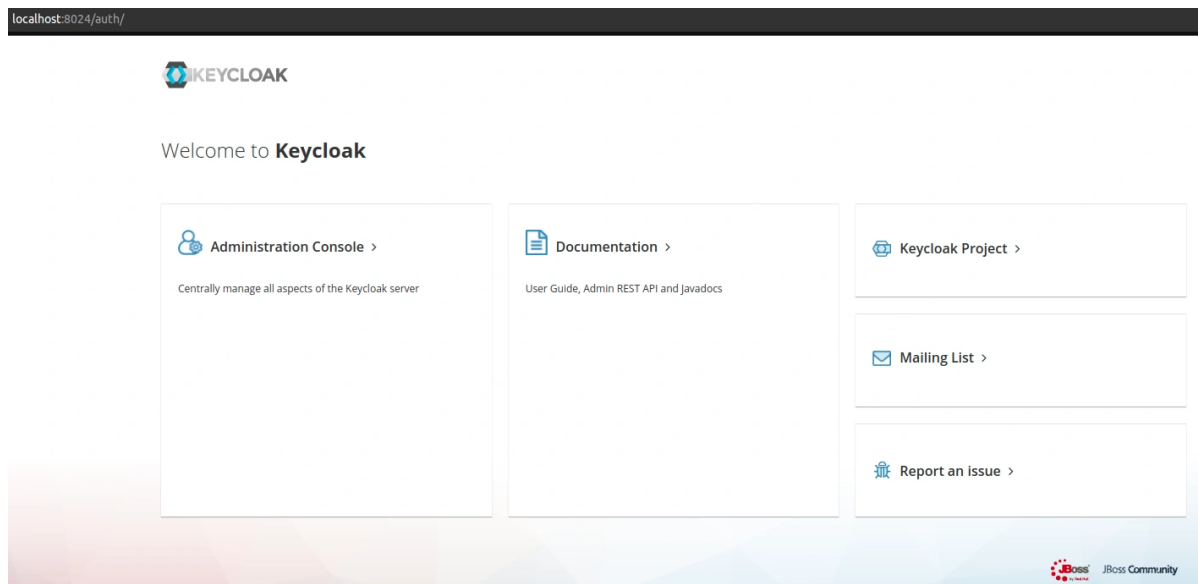
```
Log Properties Environment Variables Port Bindings Volume Bindings Files
Added 'keycloak' to '/opt/jboss/keycloak/standalone/configuration/keycloak-add-user.json', restart server to load user
-b 0.0.0.0
=====

Using PostgreSQL database

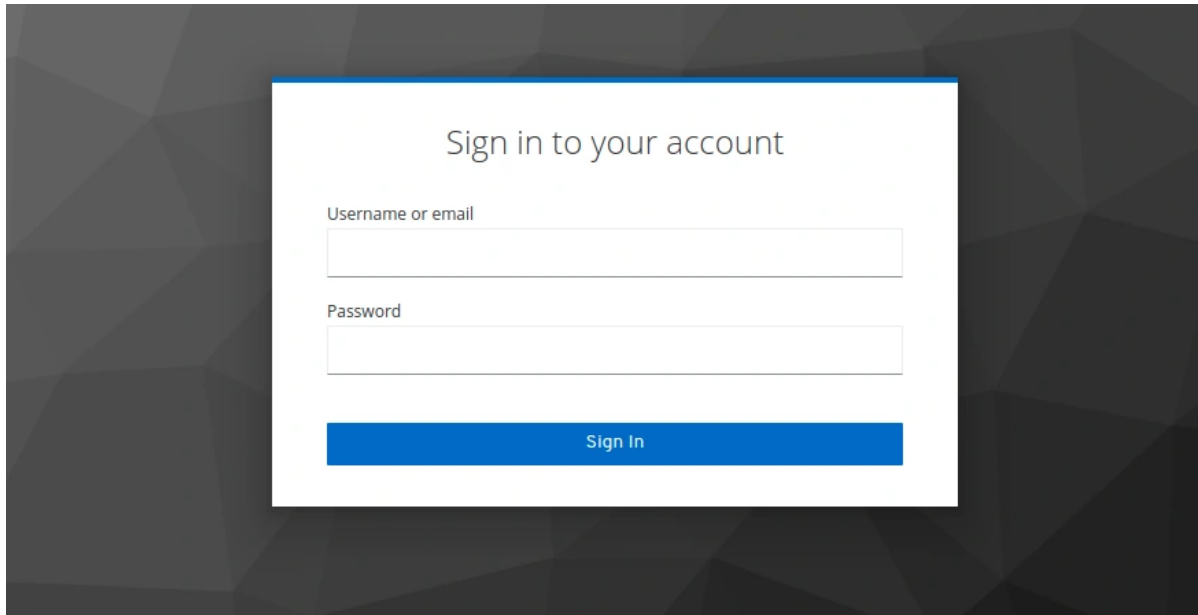
=====

08:05:56,908 INFO [org.jboss.modules] (CLI command executor) JBoss Modules version 1.11.0.Final
08:05:56,969 INFO [org.jboss.msc] (CLI command executor) JBoss MSC version 1.4.12.Final
08:05:56,976 INFO [org.jboss.threads] (CLI command executor) JBoss Threads version 2.4.0.Final
```

First, we're going to visit the <http://localhost:8024/auth/> url where we'll find the Keycloak **W**elcome screen:



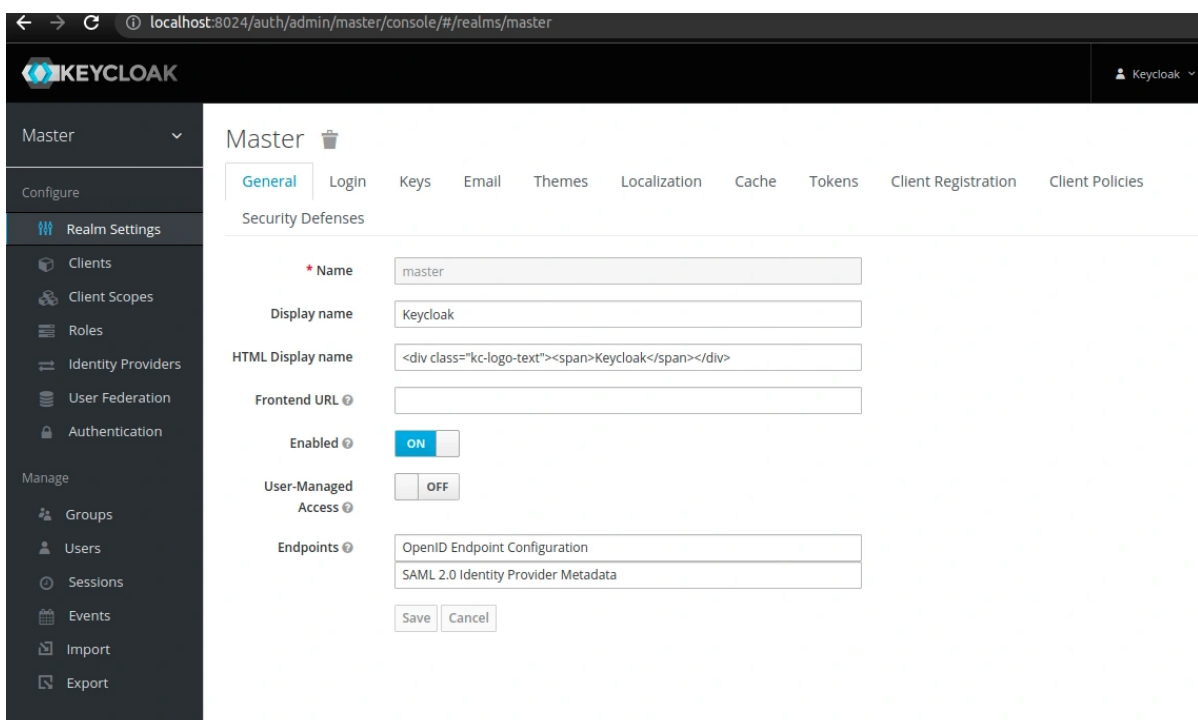
Next, we're going to select the **Administration Console** option. In order to log in, we'll need to enter our default credentials in the following form:



According to the default environment variables from my `.env` file, my credentials are:

- username: **keycloak**
- password: **keycloak**

Finally, after a successful authentication, Keycloak redirects us to the **master realm**, a pre-defined realm that Keycloak created for us:





Docker





















- [How to import a Keycloak realm](#)
- [How to customise Keycloak themes](#)

Photo by [Anna Shvets](#) from [Pexels](#)

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

Leave a Reply

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

B *I* U                    

Name *

Email *

Website

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

POST COMMENT



[JavaTools](#)[Taking care](#)[Angular](#)

About [in](#) [on](#) [with](#)

[Q](#) Search ...

[Privacy Policy](#)