

Export and import keycloak realm with users

Simon Scholz

2022-09-24

3 min read

 Feedback? Requests?

Table of Contents

- Starting a keycloak 19 instance (based on quarkus)
- Exporting a keycloak realm
- Exporting a keycloak realm including users and roles
- Copy exported realm file to host
- Import keycloak realm using docker compose
- Conclusion
- Sources

Starting a keycloak 19 instance (based on quarkus)

The following docker compose file can be used to start a keycloak instance:

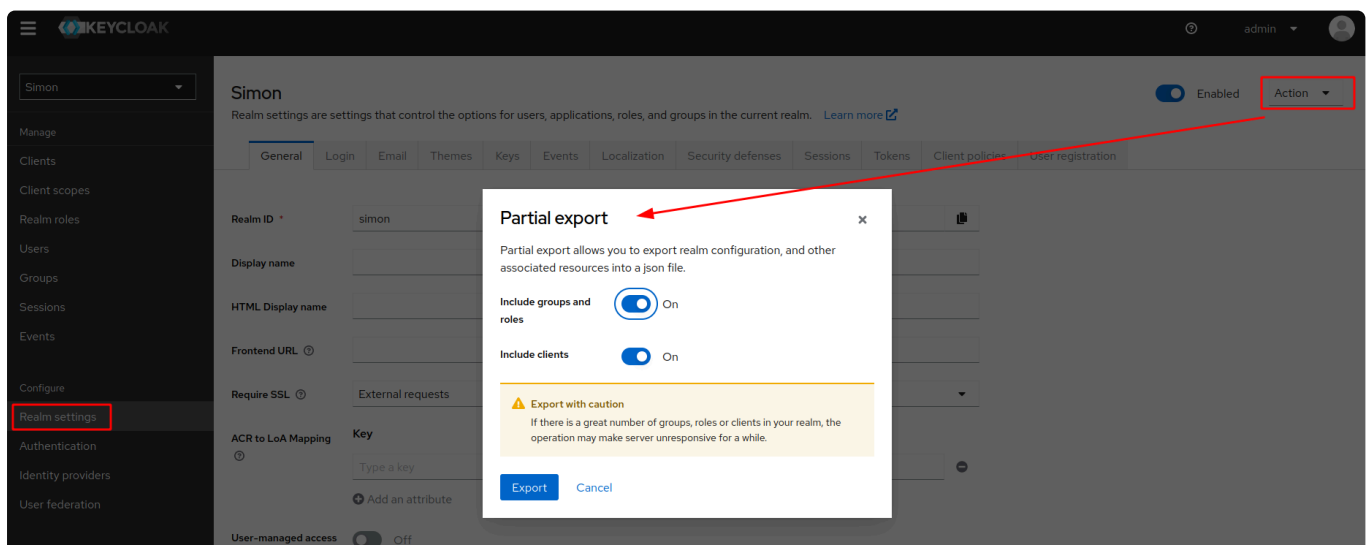
```
version: "3.7"

services:
  keycloak:
    image: "quay.io/keycloak/keycloak:19.0.2"
    container_name: quarkus-keycloak
    restart: "no"
    ports:
      - "8180:8080" # Changing this port because 8080 is usually the default for
    environment:
      - KEYCLOAK_ADMIN=admin
      - KEYCLOAK_ADMIN_PASSWORD=admin
    command:
      - start-dev
      - --import-realm
```

When saving this as `docker-compose.yml` file and running `docker-compose up` within the file's folder, keycloak will be started on <http://localhost:8180> and you can login with the `admin` user and `admin` password.

Exporting a keycloak realm

A keycloak realm can easily be exported by using the user interface of keycloak, **but** this won't contain the users and roles within that realm for security reasons.



When clicking the `Export` button you can have a look at the exported json and will find out that the users are missing.

Exporting a keycloak realm including users and roles

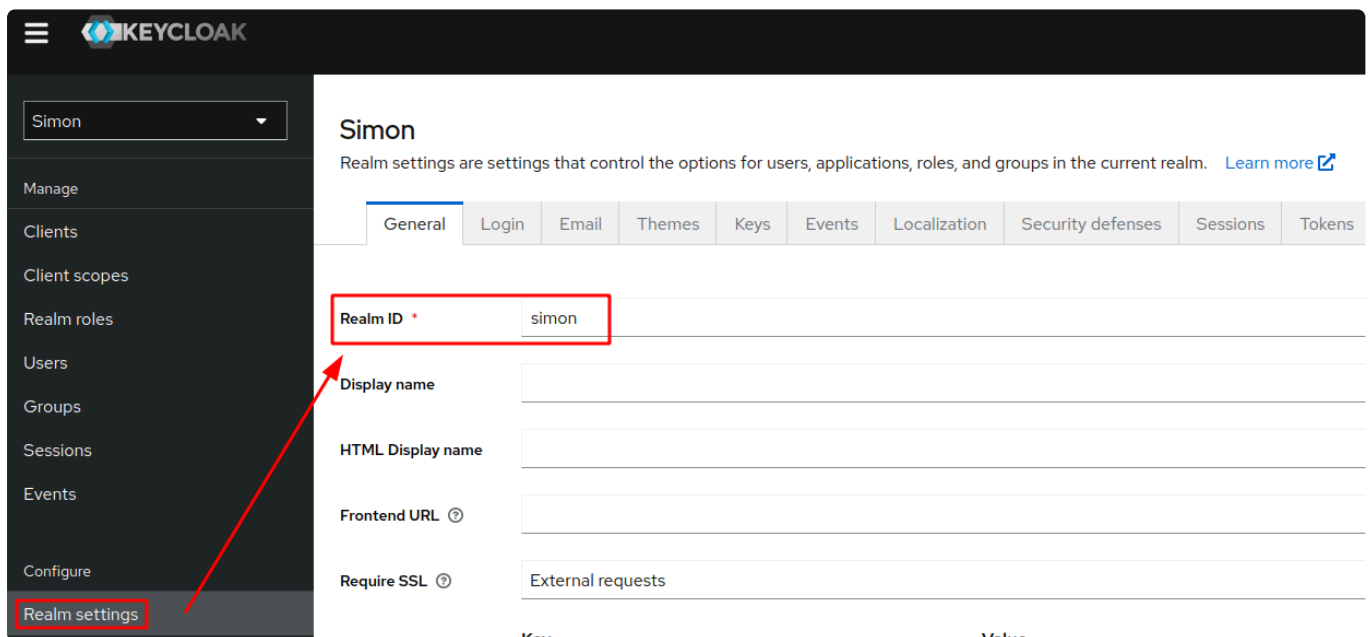
To also get user and role information the keycloak CLI has to be used.

By using the `docker ps` command we can find out the container id and connect to the container by using `docker exec -it {your-containers-id} bash`. Once the bash within the docker container is available the following command can be used to export the realm including users and roles:

```
> /opt/keycloak/bin/kc.sh export --dir /opt/keycloak/data/import --users realm_
```

With `--dir` or `--file` you can specify the export folder or file. `--users realm_file` will cause that the users will be put into the realm file. `--realm {your-realms-id}` is used to specify which realm is supposed to be exported.

The id of the realm can be found here:



This is how it looks like on my machine:

```

simon@simon:~$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS          PORTS                                                                                               NAMES
07c059321ed8  grafana/grafana:9.0.9               "/run.sh"               56 minutes ago  Up 56 minutes  0.0.0.0:3000->3000/tcp, :::3000->3000/tcp              quarkus-grafana
1d48ce2b50d9  quay.io/keycloak/keycloak:19.0.2   "/opt/keycloak/bin/k... 56 minutes ago  Up 56 minutes  8443/tcp, 0.0.0.0:8081->8080/tcp, :::8081->8080/tcp    quarkus-keycloak
7d240ca45440  mongo:5.0.12                       "docker-entrypoint.s..." 56 minutes ago  Up 56 minutes  0.0.0.0:27018->27017/tcp, :::27018->27017/tcp        quarkus-mongo

simon@simon:~$ docker exec -it 1d48ce2b50d9 bash
bash-4.45 /opt/keycloak/bin/kc.sh export --dir /opt/keycloak/data/import --users realm_file --realm simon
2022-09-22 01:17:21,215 INFO [org.keycloak.quarkus.runtime.hostname.DefaultHostnameProvider] (main) Hostname settings: Base URL: <unset>, Hostname: <request>, Strict HTTPS: false, Path: <request>, Strict
ackChannel: false, Admin URL: <unset>, Admin: <request>, Port: -1, Proxied: false
2022-09-22 01:17:22,064 INFO [org.keycloak.common.crypto.CryptoIntegration] (main) Detected crypto provider: org.keycloak.crypto.def.DefaultCryptoProvider
2022-09-22 01:17:23,072 WARN [org.infinispan.PERSISTENCE] (keycloak-cache-init) ISPN000554: jboss-marshalling is deprecated and planned for removal
2022-09-22 01:17:23,185 WARN [org.infinispan.CONFIG] (keycloak-cache-init) ISPN000569: Unable to persist Infinispan internal caches as no global state enabled
2022-09-22 01:17:23,234 INFO [org.infinispan.CONTAINER] (keycloak-cache-init) ISPN000556: Starting user marshaller 'org.infinispan.jboss-marshalling.core.jbossUserMarshaller'
2022-09-22 01:17:23,515 INFO [org.infinispan.CONTAINER] (keycloak-cache-init) ISPN000128: Infinispan version: Infinispan 'Triskaidekaphobia' 13.0.9.Final
2022-09-22 01:17:23,942 INFO [org.keycloak.connections.infinispan.DefaultInfinispanConnectionProviderFactory] (main) Node name: node_247759, Site name: null
2022-09-22 01:17:23,952 INFO [org.keycloak.exportimport.dir.DirExportProvider] (main) Exporting into directory /opt/keycloak/data/import
2022-09-22 01:17:24,254 INFO [org.keycloak.services] (main) KC-SERVICES0034: Export of realm 'simon' requested.
2022-09-22 01:17:24,811 INFO [org.keycloak.exportimport.dir.DirExportProvider] (main) Realm 'simon' - data exported
2022-09-22 01:17:24,915 INFO [org.keycloak.services] (main) KC-SERVICES0035: Export finished successfully
2022-09-22 01:17:25,091 ERROR [org.keycloak.quarkus.runtime.cli.ExecutionExceptionHandler] (main) ERROR: Failed to start server in (import_export) mode
2022-09-22 01:17:25,091 ERROR [org.keycloak.quarkus.runtime.cli.ExecutionExceptionHandler] (main) ERROR: Unable to start HTTP server
2022-09-22 01:17:25,092 ERROR [org.keycloak.quarkus.runtime.cli.ExecutionExceptionHandler] (main) ERROR: io.quarkus.runtime.QuarkusBindException
2022-09-22 01:17:25,092 ERROR [org.keycloak.quarkus.runtime.cli.ExecutionExceptionHandler] (main) For more details run the same command passing the '--verbose' option. Also you can use '--help' to see the
bash-4.45 cat /opt/keycloak/data/import/simon-realm.json
{
  "id": "6b182dc7-5e85-40fb-bf0a-8ee879a48222",
  "realm": "simon",
  "notBefore": 0,
  "defaultSignatureAlgorithm": "RS256",

```

To directly see the contents of the exported realm file you can invoke:

```
> cat /opt/keycloak/data/import/{your-realm-id}-realm.json
```

When looking at the output of the cat command, you'll see the users array:

```

"users" : [ {
  "id" : "b020aa3d-50b6-4a2c-8a3b-905d7d7a0e8d",
  "createdTimestamp" : 1663807231799,
  "username" : "kotlin",
  "enabled" : true,
  "totp" : false,
  "emailVerified" : true,
  "firstName" : "",
  "lastName" : "",
  "credentials" : [ {
    "id" : "9bad9713-fc2e-40c3-9d5f-69b3e4511383",
    "type" : "password",
    "userLabel" : "My password",
    "createdDate" : 1663807247584,
    "secretData" : "{\"value\":\"80gbnZD1kHikmCG0C76B3DsScKQD6oVXKdsrtxxPgD7BKKirGCAL5cuMsL...\",
    \"credentialData\" : \"{\\\"hashIterations\\\":27500,\\\"algorithm\\\":\\\"pbkdf2-sha256\\\",\\\"additio
  } ],
  "disableableCredentialTypes" : [ ],
  "requiredActions" : [ ],
  "realmRoles" : [ "default-roles-simon" ],
  "notBefore" : 0,
  "groups" : [ ]
}, {
  "id" : "337bd27b-fa84-4e3d-8c6c-6f098e4d68a1",
  "createdTimestamp" : 1663807259839,

```

Copy exported realm file to host

Using `cat` to view the contents of the realm file is nice, but here is the command to copy it over to your machine.

```
> docker cp {your-containers-id}:/opt/keycloak/data/import/{your-realm-id}-real
```

Import keycloak realm using docker compose

```
version: "3.7"

services:
  keycloak:
    image: "quay.io/keycloak/keycloak:19.0.2"
    container_name: quarkus-keycloak
    restart: "no"
    ports:
      - "8180:8080" # Changing this port because 8080 is usually the default for keycloak
    environment:
      - KEYCLOAK_ADMIN=admin
      - KEYCLOAK_ADMIN_PASSWORD=admin
      - KEYCLOAK_IMPORT=./{your-realm-id}-realm.json
    command:
      - start-dev
      - --import-realm
```

Adding `- KEYCLOAK_IMPORT=./{your-realm-id}-realm.json` to the `docker-compose.yaml` from the previous section will now also import the realm, which was exported earlier.

Conclusion

Of course this should be done with caution and my only reason to export also the users is to have a sample realm file in place for local development. My projects usually consist of a `docker-compose.yaml` file in the root of the project to easily ramp up services/technologies, which are used by the project, e.g., database, message broker, keycloak and more.

Sources

- <https://www.keycloak.org/guides#getting-started>
- https://www.keycloak.org/server/importExport#_exporting_a_realm_to_a_directory

- <https://www.redhat.com/sysadmin/compose-kubernetes-podman>
- <https://www.helikube.de/keycloak-18-export-and-import-feature/>

0 reactions



0 comments

Write Preview Aa

Sign in to comment

M+

Sign in with GitHub