



## Red Hat build of Keycloak 22.0

### Upgrading Guide





## Legal Notice

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

This book is a guide to upgrading your application from a previous version of Red Hat build of Keycloak 22.0.

---

## Table of Contents

<b>MAKING OPEN SOURCE MORE INCLUSIVE</b> .....	<b>3</b>
<b>CHAPTER 1. UPGRADING RED HAT BUILD OF KEYCLOAK</b> .....	<b>4</b>
<b>CHAPTER 2. UPGRADING THE RED HAT BUILD OF KEYCLOAK SERVER</b> .....	<b>5</b>
2.1. PREPARING FOR UPGRADING	5
2.2. DOWNLOADING THE SERVER ZIP FILE	5
2.3. DATABASE MIGRATION	5
2.3.1. Automatic relational database migration	5
2.3.2. Manual relational database migration	6
<b>CHAPTER 3. UPGRADING RED HAT BUILD OF KEYCLOAK ADAPTERS</b> .....	<b>7</b>
3.1. COMPATIBILITY WITH OLDER ADAPTERS	7
3.2. UPGRADING THE EAP ADAPTER	7
3.3. UPGRADING THE JAVASCRIPT ADAPTER	7
3.4. UPGRADING THE NODE.JS ADAPTER	7



## MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

## CHAPTER 1. UPGRADING RED HAT BUILD OF KEYCLOAK

Red Hat build of Keycloak is based on the Keycloak project, which enables you to secure your web applications by providing Web SSO capabilities based on popular standards such as OpenID Connect, OAuth 2.0, and SAML 2.0. The Red Hat build of Keycloak server acts as an OpenID Connect or SAML-based identity provider (IdP), allowing your enterprise user directory or third-party IdP to secure your applications by using standards-based security tokens.

This guide applies to version 22.0.11, which is a minor version of Red Hat build of Keycloak. To upgrade Red Hat build of Keycloak to version 22.0.11, use the following procedures in this order:

1. Upgrade the Red Hat build of Keycloak server
2. Upgrade Red Hat build of Keycloak adapters



## CHAPTER 2. UPGRADING THE RED HAT BUILD OF KEYCLOAK SERVER

### 2.1. PREPARING FOR UPGRADING

#### Procedure

1. Back up the old installation, such as configuration, themes, and so on.
2. Back up the database using instructions in the documentation for your relational database. The database will no longer be compatible with the old server after the upgrade. If you need to revert the upgrade, first restore the old installation, and then restore the database from the backup copy.
3. Handle any open transactions and delete the `data/tx-object-store/` transaction directory.



#### NOTE

In a minor upgrade of Red Hat build of Keycloak, all user sessions are lost. After the upgrade, all users will have to log in again.

### 2.2. DOWNLOADING THE SERVER ZIP FILE

#### Procedure

1. Download Red Hat build of Keycloak from the [Red Hat website](#) and extract it.
2. Move the ZIP file to the desired location.
3. Extract the archive.  
After extracting this file, you should have a directory that is named `rhbk-22.0.11`.
4. Copy `conf/`, `providers/` and `themes/` from the previous installation to the new installation.

### 2.3. DATABASE MIGRATION

Red Hat build of Keycloak can automatically migrate the database schema, or you can choose to do it manually. By default, the database is automatically migrated when you start the new installation for the first time.

#### 2.3.1. Automatic relational database migration

To perform an automatic migration, start the server connected to the desired database. If the database schema has changed for the new version of the server, it will be migrated.

Creating an index on huge tables with millions of records can be very time-consuming and possibly cause a major service disruption. Therefore, a threshold exists to limit the number of records allowed for the index. A warning message appears in the server logs with SQL commands you can apply later manually.

To change the threshold, set the `index-creation-threshold` property, value for the default `connections-liquibase` provider:

```
kc.[sh|bat] start --spi-connections-liquibase-default-index-creation-threshold=300000
```

### 2.3.2. Manual relational database migration

To enable manual upgrading of the database schema, set the **migration-strategy** property value to "manual" for the default **connections-jpa** provider:

```
kc.[sh|bat] start --spi-connections-jpa-legacy-migration-strategy=manual
```

When you start the server with this configuration, the server checks if the database needs to be migrated. The required changes are written to the **bin/keycloak-database-update.sql** SQL file that you can review and manually run against the database.

If you want to change the path and name of the exported SQL file, set the **migration-export** property for the default **connections-jpa** provider:

```
kc.[sh|bat] start --spi-connections-jpa-legacy-migration-export=<path>/<file.sql>
```

For more details on applying this file to the database, see the documentation for your relational database. After the changes have been written to the file, the server exits.

## CHAPTER 3. UPGRADING RED HAT BUILD OF KEYCLOAK ADAPTERS

It is important that you upgrade Red Hat build of Keycloak server first, and then upgrade the adapters. Earlier versions of the adapter might work with later versions of Red Hat build of Keycloak server, but earlier versions of Red Hat build of Keycloak server might not work with later versions of the adapter.

### 3.1. COMPATIBILITY WITH OLDER ADAPTERS

As mentioned above, we try to support newer release versions of Red Hat build of Keycloak server working with older release versions of the adapters. However, in some cases we need to include fixes on the Red Hat build of Keycloak server side which may break compatibility with older versions of the adapters. For example, when we implement new aspects of the OpenID Connect specification, which older client adapter versions were not aware of.

In those cases, we added Compatibility modes. For OpenID Connect clients, there is a section named **OpenID Connect Compatibility Modes** in the Red Hat build of Keycloak admin console, on the page with client details. Here, you can disable some new aspects of the Red Hat build of Keycloak server to preserve compatibility with older client adapters. More details are available in the tool tips of individual switches.

### 3.2. UPGRADING THE EAP ADAPTER

To upgrade an EAP adapter that has been copied to your web application, perform the following procedure.

#### Procedure

1. Download the new adapter archive.
2. Remove the previous adapter modules by deleting the **EAP\_HOME/modules/system/add-ons/keycloak/** directory.
3. Unzip the downloaded archive into **EAP\_HOME**.

### 3.3. UPGRADING THE JAVASCRIPT ADAPTER

To upgrade a JavaScript adapter that has been copied to your web application, perform the following procedure.

#### Procedure

1. Download the new adapter archive.
2. Overwrite the `keycloak.js` file in your application with the `keycloak.js` file from the downloaded archive.

### 3.4. UPGRADING THE NODE.JS ADAPTER

To upgrade a Node.js adapter that has been copied to your web application, perform the following procedure.

## Procedure

1. Download the new adapter archive.
2. Remove the existing Node.js adapter directory
3. Unzip the updated file into its place
4. Change the dependency for keycloak-connect in the package.json of your application