Red Hat build of Keycloak 24.0

Release Notes

Last Updated: 2024-06-27

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 ® is the registered trademark of Linus Torvalds in the United States and other countries.

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

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

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

Node.js ® 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 ® 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 guide consists of release notes for Red Hat build of Keycloak.

Table of Contents

CHAPTER 1. RED HAT BUILD OF KEYCLOAK 24.0 4
1.1. OVERVIEW 4
1.2. UPDATES FOR 24.0.5 4
1.3. UPDATES FOR 24.0.4 4
1.4. UPDATES FOR 22.0 4
1.5. NEW FEATURES AND ENHANCEMENTS 4
1.5.1. User profile and progressive profiling 4
1.5.1.1. Breaking changes to the User Profile SPI 5
1.5.1.2. Changes to Freemarker templates to render pages based on the user profile and realm 5
1.5.1.3. New Freemarker template for the update profile page at first login through a broker 5
1.5.2. Multi-site active-passive deployments 5
1.5.3. Account Console version 3 6
1.5.4. Welcome Page redesign 6
1.5.5. Enhanced reverse proxy settings 6
1.5.6. OAuth/OIDC related improvements 6
1.5.6.1. Lightweight access tokens support 6
1.5.6.2. OAuth 2.1 support 7
1.5.6.3. Scope parameter supported in the refresh token flow 7
1.5.6.4. Client policy executor for secure redirect URIs 7
1.5.6.5. Client policy executor for enforcing DPoP 7
1.5.6.6. Supporting EdDSA 7
1.5.6.7. EC Keys supported by JavaKeystore provider 7
1.5.6.8. Option to add X509 thumbprint to JWT when using private_key_jwt authentication for identity providers 8
1.5.6.9. OAuth Grant Type SPI 8
1.5.6.10. FAPI 2 drafts support 8
1.5.6.11. DPoP preview support 8
1.5.6.12. Feature flag for OAuth 2.0 device authorization grant flow 8
1.5.7. Authentication 8
1.5.7.1. Passkeys support 8
1.5.7.2. WebAuthn improvements 9
1.5.7.3. You are already logged-in 9
1.5.7.4. Password policy for specify Maximum authentication time 9
1.5.8. Server distribution 9
1.5.8.1. Load Shedding support 9
1.5.8.2. RESTEasy Reactive 9
1.5.9. Keycloak CR 9
1.5.9.1. Keycloak CR Optimized Field 9
1.5.9.2. Keycloak CR resources options 9
1.5.9.3. Keycloak CR cache-config-file option 10
1.5.10. Versioned Features 10
1.5.10.1. Keycloak CR Truststores 10
1.5.10.2. Trust Kubernetes CA 11
1.5.11. Group scalability 11
1.5.12. Keycloak JS 11
1.5.12.1. Using exports field in package.json 11
1.5.12.2. PKCE enabled by default 11
1.5.13. Changes to Password Hashing 11
1.5.14. Truststore improvements 11
1.5.15. More changes 12
1.5.15.1. Automatic certificate management for SAML identity providers 12

1.5.15.2. Non-blocking health check for load balancers 12
1.5.15.3. Changes to the user representation in both Admin API and Account contexts 12
1.5.15.4. Partial update to user attributes when updating users through the Admin User API is no longer supported 12
1.5.15.5. Sequential loading of offline sessions and remote sessions 12
1.5.15.6. Performing actions on behalf of another already authenticated user is not longer possible 13
1.5.15.7. Changes to the email verification flow 13
1.5.15.8. Localization files for themes default to UTF-8 encoding 13
1.5.15.9. Configuration option for offline session lifespan override in memory 13
1.5.15.10. Infinispan metrics use labels for cache manager and cache names 13
1.5.15.11. User attribute value length extension 13
1.5.15.12. Brute Force Protection changes 13
1.5.15.13. Authorization Policy 14
1.5.15.14. Temporary lockout log replaced with event 14
1.5.15.15. Updates to cookies 14
1.5.15.16. SAML User Attribute Mapper For NameID now suggests only valid NameID formats 14
1.5.15.17. Different JVM memory settings when running in container 14
1.5.15.18. Deprecated offline session preloading 15
1.6. FIXED ISSUES 15
1.7. KNOWN ISSUES 15
1.8. SUPPORTED CONFIGURATIONS 16
1.9. COMPONENT DETAILS 16

CHAPTER 1. RED HAT BUILD OF KEYCLOAK 24.0

1.1. OVERVIEW

Red Hat is proud to introduce a new era of identity and access management named 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.

While preserving the power and functionality of Red Hat Single Sign-on, Red Hat build of Keycloak is faster, more flexible, and efficient. Red Hat build of Keycloak is an application built with Quarkus, which provides developers with flexibility and modularity. Quarkus provides a framework that is optimized for a container-first approach and provides many features for developing cloud-native applications.

1.2. UPDATES FOR 24.0.5

This release contains several fixed issues including a fix for CVE-2024-4540. This fix is for a security issue affecting some OIDC confidential clients using PAR (Pushed authorization request). In case you use OIDC confidential clients together with PAR and you use client authentication based on **client_id** and **client_secret** sent as parameters in the HTTP request body (method **client_secret_post** specified in the OIDC specification), it is highly encouraged to rotate the client secrets of your clients after upgrading to this version.

1.3. UPDATES FOR 24.0.4

This release includes Fixed issues.

1.4. UPDATES FOR 22.0

If you are migrating from Red Hat Single Sign-On 7.6, other new features were added at Red Hat build of Keycloak version 22. For details, see the version 22 Release Notes.

1.5. NEW FEATURES AND ENHANCEMENTS

The following release notes apply to Red Hat build of Keycloak 24.0.3, the first 24.0 release of the product.

1.5.1. User profile and progressive profiling

The user profile preview feature is promoted to be fully supported and user profile is enabled by default.

The following are a few highlights of this feature;

- Fine-grained control over the attributes that users and administrators can manage so that you can prevent unexpected attributes and values from being set.
- Ability to specify what user attributes are managed and should be displayed on the forms to regular users or administrators.
- Dynamic forms - Previously, the forms where users created or updated their profiles, contain four basic attributes like username, email, first name and last name. The addition of any

attributes (or removing some default attributes) required you to create a custom theme. Now custom themes may not be needed because users see exactly the requested attributes based on the requirement of the particular deployment.

- Validations - Ability to specify validators for the user attributes including built-in validators that you can use to specify a maximum or minimum length, a specific regex, or limiting a particular attribute to be a URL or number.

- Annotations - Ability to specify that a particular attribute should be rendered for instance as a text area, an HTML select with specified options, or calendar or many other options. You can also bind JavaScript code to a specific field to change how an attribute is rendered and customize its behavior.

- Progressive profiling - Ability to specify that some fields are required or available on the forms just for particular values of **scope** parameter. This effectively allow progressive profiling. You no longer need to ask the user for twenty attributes during registration; you can instead ask the user to fill in attributes incrementally according to the requirements of the individual client applications that are used by the user.

- Migration from previous versions - The user profile is now always enabled, but it operates as before for those who did not use this feature. You can benefit from the user profile capabilities, but you are not required to use them. For migration instructions, see the Upgrading Guide.

The first release of the user profile as a supported feature is just the starting point and the baseline for delivering many more capabilities around identity management.

For more details about user profile capabilities, see the Server Administration Guide.

1.5.1.1. Breaking changes to the User Profile SPI

In this release, changes to the User Profile SPI might impact existing implementations based on this SPI. For more details, see the Upgrading Guide.

1.5.1.2. Changes to Freemarker templates to render pages based on the user profile and realm

In this release, the following templates were updated to make it possible to dynamically render attributes based on the user profile configuration set to a realm:

- **login-update-profile.ftl**

- **register.ftl**

- **update-email.ftl**

For more details, see the Upgrading Guide.

1.5.1.3. New Freemarker template for the update profile page at first login through a broker

In this release, the server renders the update profile page when the user is authenticating through a broker for the first time using the **idp-review-user-profile.ftl** template.

For more details, see the Upgrading Guide.

1.5.2. Multi-site active-passive deployments

Deploying Red Hat build of Keycloak to multiple independent sites is essential for some environments to provide high availability and a speedy recovery from failures. This release supports active-passive deployments for Red Hat build of Keycloak.

To get started, use the High Availability Guide which also includes a comprehensive blueprint to deploy a highly available Red Hat build of Keycloak to a cloud environment.

1.5.3. Account Console version 3

Account Console version 3 has built-in support for the user profile feature, which allows administrators to configure which attributes are available to users in the Account Console, and lands a user directly on their personal account page after logging in.

If you are using or extending the customization features of this theme, you may need to perform additional migrations. For more details, see the Upgrading Guide.

Account Console version 2 is deprecated and will be removed in a subsequent release.

1.5.4. Welcome Page redesign

The Welcome page that appears at the first use of Red Hat build of Keycloak is redesigned. It provides a better setup experience and conforms to the latest version of PatternFly. The simplified page layout includes only a form to register the first administrative user. After completing the registration, the user is sent directly to the Admin Console.

If you use a custom theme, you may need to update it to support the new welcome page. For details, see the Upgrading Guide.

1.5.5. Enhanced reverse proxy settings

It is now possible to separately enable parsing of either **Forwarded** or **X-Forwarded-*** headers by using the new **--proxy-headers** option. For details, see Using a reverse proxy. The original **--proxy** option is now deprecated and will be removed in a future release. For migration instructions, see the Upgrading Guide.

1.5.6. OAuth/OIDC related improvements

1.5.6.1. Lightweight access tokens support

This release contains support for Lightweight access tokens. As a result, you can have smaller access tokens for specified clients. These tokens have only a few claims, which is why they are smaller. Note that lightweight access token is still JWT signed by the realm key by default and still contains some very basic claims.

This release introduces an **Add to lightweight access token**flag that is available on some OIDC protocol mappers. Use this flag to specify if a particular claim should be added to a lightweight access token. It is **OFF** by default, which means that most claims are not added.

Also, a client policy executor exists. Use it to specify if a particular client request should use lightweight access tokens or regular access tokens. An alternative to the executor is to use an **Always use lightweight access token** flag on client advanced settings, which causes that client to always use lightweight access tokens. An executor can be an alternative if you need more flexibility. For instance, you may choose to use lightweight access tokens by default but use regular tokens only for the specified **scope** parameter.

In previous versions, introspection endpoint automatically returned most claims, which were available in the access token. Now most of protocol mappers include a new **Add to token introspection** switch . This addition allows more flexibility because an introspection endpoint can return different claims than an access token. This change is a first step towards "Lightweight access tokens" support because access tokens can omit lots of the claims, which would be still returned by the introspection endpoint. When migrating from previous versions, the introspection endpoint should return the same claims that are returned from access token, so the behavior should be effectively the same by default after the upgrade.

For more details, see Using lightweight access tokens.

1.5.6.2. OAuth 2.1 support

This release contains optional OAuth 2.1 support. New client policy profiles were introduced in this release, which administrators can use to make sure that clients and particular client requests comply with the OAuth 2.1 specification. This release includes a dedicated client profile for confidential clients and a dedicated profile for public clients.

For more details, see OAuth 2.1 support.

1.5.6.3. Scope parameter supported in the refresh token flow

Starting with this release, the **scope** parameter in the OAuth2/OIDC endpoint for token refresh is supported. Use this parameter to request access tokens with a smaller amount of scopes than originally granted, which means you cannot increase access token scope. This scope limitation does not affect the scope of the refreshed refresh token. This function works as described in the OAuth2 specification.

For more details, see the Server Administration Guide.

1.5.6.4. Client policy executor for secure redirect URIs

A new client policy executor **secure-redirect-uris-enforcer** is introduced. Use it to restrict which redirect URIs can be used by the clients. For instance, you can specify that client redirect URIs cannot have wildcards, should be just from specific domain, must be OAuth 2.1 compliant, and so on.

For more details, see Client Policies.

1.5.6.5. Client policy executor for enforcing DPoP

A new client policy executor **dpop-bind-enforcer** is introduced. You can use it to enforce DPoP for a particular client if **dpop** preview is enabled.

For more details, see Client Policies.

1.5.6.6. Supporting EdDSA

You can create EdDSA realm keys and use them as signature algorithms for various clients. For instance, you can use these keys to sign tokens or for client authentication with signed JWT. This feature includes identity brokering where Red Hat build of Keycloak itself signs client assertions that are used for **private_key_jwt** authentication to third party identity providers.

For more details, see Configuring Realm keys

1.5.6.7. EC Keys supported by JavaKeystore provider

The provider **JavaKeystoreProvider** for providing realm keys now supports EC keys in addition to previously supported RSA keys.

For more details, see Configuring Realm keys

1.5.6.8. Option to add X509 thumbprint to JWT when using private_key_jwt authentication for identity providers

OIDC identity providers now have the **Add X.509 Headers to the JWT**option for the situation when client authentication with JWT signed by private key is used. This option can be useful for interoperability with some identity providers such as Azure AD, which require the thumbprint to be present on the JWT.

For more details, see Integrating identity providers.

1.5.6.9. OAuth Grant Type SPI

The Red Hat build of Keycloak codebase includes an internal update to introduce the OAuth Grant Type SPI. This update allows additional flexibility when introducing custom grant types supported by the Red Hat build of Keycloak OAuth 2 token endpoint.

For more details, see Authorization services.

1.5.6.10. FAPI 2 drafts support

Red Hat build of Keycloak has new client profiles **fapi-2-security-profile** and **fapi-2-message-signing**, which ensure Red Hat build of Keycloak enforces compliance with the latest FAPI 2 draft specifications when communicating with your clients.

For more details, see Client Policies.

1.5.6.11. DPoP preview support

Red Hat build of Keycloak has preview for support for OAuth 2.0 Demonstrating Proof-of-Possession at the Application Layer (DPoP).

1.5.6.12. Feature flag for OAuth 2.0 device authorization grant flow

The OAuth 2.0 device authorization grant flow now includes a feature flag, so you can easily disable this feature. This feature is still enabled by default.

For more details, see Device authorization grant.

1.5.7. Authentication

1.5.7.1. Passkeys support

Red Hat build of Keycloak has preview support for Passkeys.

Passkey registration and authentication are realized by the features of WebAuthn. Therefore, users of Red Hat build of Keycloak can do Passkey registration and authentication by existing WebAuthn registration and authentication.

Both synced Passkeys and device-bound Passkeys can be used for both Same-Device and Cross-Device Authentication. However, Passkeys operations success depends on the user's environment. Make sure which operations can succeed in the environment.

1.5.7.2. WebAuthn improvements

WebAuthn policy includes a new field: **Extra Origins**. It provides better interoperability with non-Web platforms (for example, native mobile applications).

1.5.7.3. You are already logged-in

This release addresses an issue concerning when a user has a login page open in multiple browser tabs and is authenticated in one browser tab. When the user tried to authenticate in another browser tab, a message appeared: **You are already logged-in**. This situation is improved now as other browser tabs automatically authenticate the user after authentication in the first tab. However, more improvements are needed. For example, when an authentication session expires and is restarted in one browser tab, other browser tabs do not follow automatically with the login.

1.5.7.4. Password policy for specify Maximum authentication time

Red Hat build of Keycloak supports a new password policy that allows you to specify the maximum age of an authentication with which a password may be changed by a user without re-authentication. When this password policy is set to 0, the user is required to re-authenticate to change the password in the Account Console or by other means. You can also specify a lower or higher value than the default value of 5 minutes.

1.5.8. Server distribution

1.5.8.1. Load Shedding support

Red Hat build of Keycloak now features the **http-max-queued-requests** option to allow proper rejection of incoming requests under high load. For details, see the Server Guide.

1.5.8.2. RESTEasy Reactive

Red Hat build of Keycloak has switched to RESTEasy Reactive. Applications using **quarkus-resteasy-reactive** should still benefit from a better startup time, runtime performance, and memory footprint, even though not using reactive style/semantics. SPIs that depend directly on JAX-RS API should be compatible with this change. SPIs that depend on RESTEasy Classic including **ResteasyClientBuilder** will not be compatible and will require an update. This update will also be needed for other implementation of the JAX-RS API like Jersey.

1.5.9. Keycloak CR

1.5.9.1. Keycloak CR Optimized Field

The Keycloak CR now includes an **startOptimized** field, which may be used to override the default assumption about whether to use the **--optimized** flag for the start command. As a result, you can use the CR to configure build time options also when a custom Keycloak image is used.

1.5.9.2. Keycloak CR resources options

The Keycloak CR now allows for specifying the **resources** options for managing compute resources for the Keycloak container. It provides the ability to request and limit resources independently for the main Red Hat build of Keycloak deployment by using the Keycloak CR, and for the realm import Job by using the Realm Import CR.

When no values are specified, the default **requests** memory is set to **1700MiB**, and the **limits** memory is set to **2GiB**.

You can specify your custom values based on your requirements as follows:

```
apiVersion: k8s.keycloak.org/v2alpha1
kind: Keycloak
metadata:
  name: example-kc
spec:
  ...
  resources:
    requests:
      cpu: 1200m
      memory: 896Mi
    limits:
      cpu: 6
      memory: 3Gi
```

For more details, see the Operator Guide.

1.5.9.3. Keycloak CR cache-config-file option

The Keycloak CR now allows for specifying the **cache-config-file** option by using the **cache** spec **configMapFile** field, for example:

```
apiVersion: k8s.keycloak.org/v2alpha1
kind: Keycloak
metadata:
  name: example-kc
spec:
  ...
  cache:
    configMapFile:
      name: my-configmap
      key: config.xml
```

1.5.10. Versioned Features

Features now support versioning. To preserve backward compatibility, all existing features (including **account2** and **account3**) are marked as version 1. Newly introduced features will use versioning, which means that users can select between different implementations of desired features.

For details, see the Server Guide.

1.5.10.1. Keycloak CR Truststores

You may also take advantage of the new server-side handling of truststores by using the Keycloak CR, for example:

```
spec:
  truststores:
    mystore:
      secret:
        name: mystore-secret
    myotherstore:
      secret:
        name: myotherstore-secret
```

Currently only Secrets are supported.

1.5.10.2. Trust Kubernetes CA

The cert for the Kubernetes CA is added automatically to your Red Hat build of Keycloak Pods managed by the Operator.

1.5.11. Group scalability

Performance around searching of groups is improved for the use-cases with many groups and subgroups. There are improvements, which allow paginated lookup of subgroups.

1.5.12. Keycloak JS

1.5.12.1. Using exports field in package.json

The Red Hat build of Keycloak JS adapter now uses the **exports** field in its **package.json**. This change improves support for more modern bundlers like Webpack 5 and Vite, but comes with some unavoidable breaking changes. See the Upgrading Guide for more details.

1.5.12.2. PKCE enabled by default

The Red Hat build of Keycloak JS adapter now sets the **pkceMethod** option to **S256** by default. This change enables Proof Key Code Exchange (PKCE) for all applications using the adapter. If you use the adapter on a system that does not support PKCE, you can set the **pkceMethod** option to **false** to disable it.

1.5.13. Changes to Password Hashing

In this release, we adapted the password hashing defaults to match the OWASP recommendations for Password Storage.

As part of this change, the default password hashing provider has changed from **pbkdf2-sha256** to **pbkdf2-sha512**. Also, the number of default hash iterations for **pbkdf2** based password hashing algorithms changed. This change means better security aligned with latest recommendations, but it has impact on performance. It is possible to stick to the old behavior by adding password policies **hashAlgorithm** and **hashIterations** to your realm. For more details, see the Upgrading Guide.

1.5.14. Truststore improvements

Red Hat build of Keycloak introduces improved truststores configuration options. The Red Hat build of Keycloak truststore is now used across the server, including outgoing connections, mTLS, and database drivers. You no longer need to configure separate truststores for individual areas. To configure the truststore, you can put your truststores files or certificates in the default **conf/truststores**, or use the new **truststore-paths** config option.

For details, see the Server Guide.

1.5.15. More changes

1.5.15.1. Automatic certificate management for SAML identity providers

The SAML identity providers can now be configured to automatically download the signing certificates from the IDP entity metadata descriptor endpoint. In order to use the new feature, configure the **Metadata descriptor URL** option in the provider (the URL where the IDP metadata information with the certificates is published) and set **Use metadata descriptor URL** to **ON**. The certificates are automatically downloaded and cached in the **public-key-storage** SPI from that URL. The certificates can also be reloaded or imported from the Admin Console, using the action combo in the provider page.

See the Server Administration Guide for more details about the new options.

1.5.15.2. Non-blocking health check for load balancers

A new health check endpoint available at **/lb-check** was added. The execution is running in the event loop, which means this check is responsive also in overloaded situations when Red Hat build of Keycloak needs to handle many requests waiting in request queue. This behavior is useful, for example, in multi-site deployment to avoid failing over to another site that is under heavy load. The endpoint is currently checking availability of the embedded and external Infinispan caches. Other checks may be added later.

This endpoint is not available by default. To enable it, run Keyloak with the **multi-site** feature. For more details, see Enabling and disabling features.

1.5.15.3. Changes to the user representation in both Admin API and Account contexts

In this release, we are encapsulating the root user attributes (such as **username**, **email**, **firstName**, **lastName**, and **locale**) by moving them to a base/abstract class in order to align how these attributes are marshalled and unmarshalled when using both Admin and Account REST APIs.

This strategy provides consistency in how attributes are managed by clients and makes sure they conform to the user profile configuration set to a realm.

For more details, see the Upgrading Guide.

1.5.15.4. Partial update to user attributes when updating users through the Admin User API is no longer supported

When updating user attributes through the Admin User API, you cannot execute partial updates when updating the user attributes, including the root attributes such as **username**, **email**, **firstName**, and **lastName**.

For more details, see the Upgrading Guide.

1.5.15.5. Sequential loading of offline sessions and remote sessions

Starting with this release, the first member of a Red Hat build of Keycloak cluster will load remote sessions sequentially instead of in parallel. If offline session preloading is enabled, those will be loaded sequentially as well.

For more details, see the Upgrading Guide.

1.5.15.6. Performing actions on behalf of another already authenticated user is not longer possible

In this release, you can no longer perform actions such as email verification if the user is already authenticated and the action is bound to another user. For instance, a user can not complete the verification email flow if the email link is bound to a different account.

1.5.15.7. Changes to the email verification flow

In this release, if a user tries to follow the link to verify the email and the email was previously verified, a proper message will be shown.

In addition to that, a new error (**EMAIL_ALREADY_VERIFIED**) event will be fired to indicate an attempt to verify an already verified email. You can use this event to track possible attempts to hijack user accounts in case the link has leaked or to alert users if they do not recognize the action.

1.5.15.8. Localization files for themes default to UTF-8 encoding

Message properties files for themes are now read in UTF-8 encoding, with an automatic fallback to ISO-8859-1 encoding.

See the Upgrading Guide for more details.

1.5.15.9. Configuration option for offline session lifespan override in memory

To reduce memory requirements, we introduced a configuration option to shorten lifespan for offline sessions imported into the Infinispan caches. Currently, the offline session lifespan override is disabled by default.

For more details, see the Server Administration Guide.

1.5.15.10. Infinispan metrics use labels for cache manager and cache names

When enabling metrics for Red Hat build of Keycloak's embedded caches, the metrics now use labels for the cache manager and the cache names.

For more details, see the Upgrading Guide.

1.5.15.11. User attribute value length extension

As of this release, Red Hat build of Keycloak supports storing and searching by user attribute values longer than 255 characters, which was previously a limitation.

For more details, see the Upgrading Guide.

1.5.15.12. Brute Force Protection changes

There have been a couple of enhancements to the Brute Protection:

1. When an attempt to authenticate with an OTP or Recovery Code fails due to Brute Force Protection, the active Authentication Session is invalidated. Any further attempts to authenticate with that session will fail.

2. In previous versions of Red Hat build of Keycloak, the administrator had to choose between disabling users temporarily or permanently due to a Brute Force attack on their accounts. The administrator can now permanently disable a user after a given number of temporary lockouts.

3. The property **failedLoginNotBefore** has been added to the **brute-force/users/{userId}** endpoint

1.5.15.13. Authorization Policy

In previous versions of Red Hat build of Keycloak, when the last member of a User, Group, or Client policy was deleted then that policy would also be deleted. Unfortunately this could lead to an escalation of privileges if the policy was used in an aggregate policy. To avoid privilege escalation, the effect policies are no longer deleted and an administrator will need to update those policies.

1.5.15.14. Temporary lockout log replaced with event

There is now a new event **USER_DISABLED_BY_TEMPORARY_LOCKOUT** when a user is temporarily locked out by the brute force protector. The log with ID **KC-SERVICES0053** has been removed as the new event offers the information in a structured form.

For more details, see the Upgrading Guide.

1.5.15.15. Updates to cookies

Cookie handling code has been refactored and improved, including a new Cookie Provider. This provides better consistency for cookies handled by Red Hat build of Keycloak, and the ability to introduce configuration options around cookies if needed.

1.5.15.16. SAML User Attribute Mapper For NameID now suggests only valid NameID formats

User Attribute Mapper For NameID allowed setting **Name ID Format** option to the following values:

- **urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName**
- **urn:oasis:names:tc:SAML:1.1:nameid-format:WindowsDomainQualifiedName**
- **urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos**
- **urn:oasis:names:tc:SAML:2.0:nameid-format:entity**

However, Red Hat build of Keycloak does not support receiving **AuthnRequest** document with one of these **NameIDPolicy**, therefore these mappers would never be used. The supported options were updated to only include the following Name ID Formats:

- **urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress**
- **urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified**
- **urn:oasis:names:tc:SAML:2.0:nameid-format:persistent**
- **urn:oasis:names:tc:SAML:2.0:nameid-format:transient**

1.5.15.17. Different JVM memory settings when running in container

Instead of specifying hardcoded values for the initial and maximum heap size, Red Hat build of Keycloak uses relative values to the total memory of a container. The JVM options **-Xms** and **-Xmx** were replaced by **-XX:InitialRAMPercentage** and **-XX:MaxRAMPercentage**.

> **WARNING**
>
> It can significantly impact memory consumption, so executing particular actions might be required.

For more details, see the Upgrading Guide.

1.5.15.18. Deprecated offline session preloading

The default behavior of Red Hat build of Keycloak is to load offline sessions on demand. The old behavior to preload them at startup is now deprecated, as pre-loading them at startup does not scale well with a growing number of sessions, and increases Red Hat build of Keycloak memory usage. The old behavior will be removed in a future release.

For more details, see the Upgrading Guide.

1.6. FIXED ISSUES

Each release includes fixed issues:

- Red Hat build of Keycloak 24.0.5 Fixed Issues
- Red Hat build of Keycloak 24.0.4 Fixed Issues
- Red Hat build of Keycloak 24.0.3 Fixed Issues

1.7. KNOWN ISSUES

Red Hat Single Sign-On 7.6 OIDC adapters do not work by default with Red Hat build of Keycloak 24.0.

When running Red Hat Single Sign-On 7.6 OIDC adapters with Red Hat build of Keycloak 24.0, the log shows a CODE_TO_TOKEN_ERROR event. To work around this issue, make this change for each Red Hat build of Keycloak client that points to an application secured by Red Hat Single Sign-On 7.6 adapters.

1. In the Admin Console, select the affected client.
2. Go to the **Advanced** tab.
3. Locate the **OpenID Connect Compatibility Modes** section.
4. Toggle **Exclude Issuer From Authentication Response**to **ON**.

For more information, see https://issues.redhat.com/browse/RHSSO-3030.

1.8. SUPPORTED CONFIGURATIONS

For the supported configurations for Red Hat build of Keycloak 24.0, see Supported configurations.

1.9. COMPONENT DETAILS

For the list of supported component versions for Red Hat build of Keycloak 24.0, see Component details.