Red Hat build of Keycloak 24.0

リリースノート

Last Updated: 2024-06-12

法律上の通知

概要

これは Red Hat Build of Keycloak のリリースノートです。

目次

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、Red Hat CTO である Chris Wright のメッセージ をご覧ください。

第1章 RED HAT BUILD OF KEYCLOAK 24.0

1.1. 概要

Red Hat は、新しい時代の ID およびアクセス管理である Red Hat build of Keycloak を導入できることを誇りに思います。Red Hat build of Keycloak は Keycloak プロジェクトをベースとしており、OpenID Connect、OAuth 2.0、SAML 2.0 などの一般的な標準仕様に基づいて Web SSO 機能を提供することで、Web アプリケーションのセキュリティーを保護します。Red Hat build of Keycloak サーバーは OpenID Connect または SAML ベースの ID プロバイダー (IdP) として機能し、エンタープライズユーザーディレクトリーまたはサードパーティー IdP が標準仕様ベースのセキュリティートークンを使用してアプリケーションを保護できるようにします。

Red Hat build of Keycloak は Red Hat Single Sign-on のパワーと機能性を維持しつつ、さらなるスピード、柔軟性、効率性を実現しました。Red Hat build of Keycloak は、Quarkus でビルドされたアプリケーションであり、開発者に柔軟性とモジュール性を提供します。Quarkus は、コンテナーファーストのアプローチに最適化されたフレームワークを提供し、クラウドネイティブアプリケーションを開発するための多くの機能を提供します。

1.2. 24.0.5 の更新

このリリースには、CVE-2024-4540 の修正など、複数の 修正された問題 が含まれています。この修正は、PAR (プッシュされた承認要求) を使用する一部の OIDC 機密クライアントに影響するセキュリティー問題に対して行われました。OIDC 機密クライアントを PAR と一緒に使用し、HTTP リクエスト本文のパラメーターとして送信された **client_id** と **client_secret** に基づくクライアント認証を使用する場合 (OIDC 仕様で指定されたメソッド **client_secret_post**)、このバージョンにアップグレードした後、クライアントのクライアントシークレットをローテーションすることを強く推奨します。

1.3. 24.0.4 の更新

このリリースには 修正された問題 が含まれています。

1.4. 新機能および機能拡張

次のリリースノートは、Red Hat build of Keycloak の最初の 24.0 リリースである 24.0.3 を対象とします。

1.4.1. ユーザープロファイルとプログレッシブプロファイリング

ユーザープロファイルのプレビュー機能が完全にサポートされるようになり、ユーザープロファイルがデフォルトで有効になりました。

この機能の主な特徴をいくつか紹介します。

- ユーザーと管理者が管理できる属性をきめ細かく制御し、予期しない属性や値が設定されるのを防ぐことができます。
- 通常のユーザーまたは管理者に対してフォームに表示される管理対象のユーザー属性を指定できます。
- 動的フォーム - 以前は、ユーザーがプロファイルを作成または更新するフォームに、ユーザー名、メールアドレス、名、姓などの 4 つの基本属性が含まれていました。属性を追加する (または一部のデフォルト属性を削除する) には、カスタムテーマを作成する必要がありました。現

在は、特定のデプロイメントの要件に基づいて、要求される属性が正確にユーザーに表示されるため、カスタムテーマは必要なくなる可能性があります。

- 検証 - ユーザー属性の検証を指定する機能。組み込みのバリデーターを使用して、最大長や最小長、特定の正規表現を指定したり、特定の属性を URL または数値に制限したりできます。
- アノテーション - 特定の属性を、たとえばテキスト領域、指定オプションを含む HTML の select、カレンダー、またはその他のオプションとしてレンダリングするよう指定する機能。また、JavaScript コードを特定のフィールドにバインドして、属性のレンダリング方法を変更し、その動作をカスタマイズすることもできます。
- プログレッシブプロファイリング - **scope** パラメーターの特定の値に対してのみ、フォーム上の一部のフィールドを必須または使用可能フィールドとして指定する機能。これにより、プログレッシブプロファイリングが効果的に可能になります。登録時にユーザーに 20 個の属性を尋ねる必要がなくなります。代わりに、ユーザーが使用する個々のクライアントアプリケーションの要件に応じて、段階的に属性を入力するようにユーザーに求めることができます。
- 以前のバージョンからの移行 - ユーザープロファイルが常に有効になりました。ただし、この機能を使用していないユーザーの場合は、以前と同じように動作します。ユーザープロファイルの機能を利用することもできますが、必ずしも使用する必要はありません。移行手順については、アップグレードガイド を参照してください。

ユーザープロファイルは、サポート対象の機能として初めてリリースされました。これは出発点であり、アイデンティティー管理に関する機能をさらに提供するためのベースラインです。

ユーザープロファイル機能の詳細は、サーバー管理ガイド を参照してください。

1.4.1.1. User Profile SPI の重大な変更

このリリースでは、User Profile SPI への変更により、この SPI に基づく既存の実装に影響が及ぶ可能性があります。詳細は、アップグレードガイド を参照してください。

1.4.1.2. ユーザープロファイルとレルムに基づいてページをレンダリングする Freemarker テンプレートの変更

このリリースでは、次のテンプレートが更新され、レルムに設定されたユーザープロファイル設定に基づいて属性を動的にレンダリングできるようになりました。

- **login-update-profile.ftl**
- **register.ftl**
- **update-email.ftl**

詳細は、アップグレードガイド を参照してください。

1.4.1.3. ブローカー経由で初めてログインしたときのプロファイル更新ページ用の新しい Freemarker テンプレート

このリリースでは、ユーザーが **idp-review-user-profile.ftl** テンプレートを使用して初めてブローカー経由で認証するときに、サーバーが更新プロファイルページをレンダリングします。

詳細は、アップグレードガイド を参照してください。

1.4.2. マルチサイトのアクティブ/パッシブデプロイメント

環境によっては、障害からの迅速な回復と高可用性を実現するために、複数の独立したサイトに Red Hat build of Keycloak をデプロイすることが不可欠です。このリリースでは、Red Hat build of Keycloak のアクティブ/パッシブデプロイメントがサポートされています。

使用を開始するには、高可用性ガイド を使用してください。このガイドには、高可用性 Red Hat build of Keycloak をクラウド環境にデプロイするための包括的なブループリントも含まれています。

1.4.3. アカウントコンソールバージョン 3

アカウントコンソールバージョン 3 には、ユーザープロファイル機能のサポートが組み込まれています。管理者はアカウントコンソールでユーザーが使用できる属性を設定し、ログイン後にユーザーを個人アカウントページに直接移動させることができます。

このテーマのカスタマイズ機能を使用または拡張する場合は、追加の移行を実行する必要がある場合があります。詳細は、アップグレードガイド を参照してください。

アカウントコンソールバージョン 2 は非推奨になりました。後続のリリースで削除される予定です。

1.4.4. Welcome ページの再設計

Red Hat build of Keycloak を初めて使用したときに表示される Welcome ページが再設計されました。新しいページは、より優れたセットアップエクスペリエンスを提供し、最新バージョンの PatternFly に準拠しています。ページのレイアウトが簡略化され、最初の管理ユーザーを登録するためのフォームのみが表示されます。登録が完了すると、ユーザーは管理コンソールに直接移動します。

カスタムテーマを使用している場合は、新しい Welcome ページをサポートするためにテーマを更新する必要がある場合があります。詳細は、アップグレードガイド を参照してください。

1.4.5. リバースプロキシー設定の強化

新しい **--proxy-headers** オプションを使用して、**Forwarded** ヘッダーまたは **X-Forwarded-*** ヘッダーのいずれかの解析を個別に有効にできるようになりました。詳細は、リバースプロキシーの使用 を参照してください。元の **--proxy** オプションは現在非推奨となっており、今後のリリースで削除される予定です。移行手順については、アップグレードガイド を参照してください。

1.4.6. OAuth/OIDC 関連の改善

1.4.6.1. 軽量アクセストークンのサポート

このリリースには、軽量アクセストークンのサポートが含まれています。そのため、指定のクライアントのアクセストークンを小さくすることができます。このトークンにはクレームがわずかしかないため、サイズが小さくなります。軽量アクセストークンは、デフォルトではレルムキーによって署名された JWT のままであり、非常に基本的なクレームをいくつか含んでいることに注意してください。

このリリースでは、**Add to lightweight access token** フラグが導入されました。これは一部の OIDC プロトコルマッパーで使用できます。このフラグを使用して、特定のクレームを軽量アクセストークンに追加する必要があるかどうかを指定します。デフォルトでは **OFF** なっているため、ほとんどのクレームは追加されません。

また、クライアントポリシーエグゼキューターも存在します。これを使用して、特定のクライアント要求で軽量アクセストークンを使用するか、通常のアクセストークンを使用するかを指定します。エグゼキューターの代替手段として、クライアントの詳細設定で **Always use lightweight access token** フラグを使用する方法があります。これを使用すると、クライアントが常に軽量アクセストークンを使用す

るようになります。より柔軟性が必要な場合は、代わりにエグゼキューターを利用できます。たとえば、デフォルトで軽量アクセストークンを使用し、指定の **scope** パラメーターに対してのみ通常のトークンを使用するように選択できます。

以前のバージョンでは、イントロスペクションエンドポイントが、アクセストークンで使用可能なほとんどのクレームを自動的に返していました。現在は、ほとんどのプロトコルマッパーに、新しい **Add to token introspection** スイッチが含まれています。これにより、イントロスペクションエンドポイントがアクセストークンとは異なるクレームを返すことができるため、柔軟性が向上します。この変更は、"軽量アクセストークン" のサポートに向けた第一歩です。アクセストークンでは多くのクレームを省略できますが、現在も、多くのクレームがイントロスペクションエンドポイントによって返されるためです。以前のバージョンから移行する場合、イントロスペクションエンドポイントはアクセストークンから返されるものと同じクレームを返す必要があります。そのため、アップグレード後の動作は、デフォルトで実質的に同じになるはずです。

詳細は、軽量アクセストークンの使用 を参照してください。

1.4.6.2. OAuth 2.1 のサポート

このリリースには、オプションの OAuth 2.1 サポートが含まれています。このリリースでは、新しいクライアントポリシープロファイルが導入されました。管理者はこれを使用して、クライアントと特定のクライアント要求が OAuth 2.1 仕様に準拠していることを確認できます。このリリースには、機密クライアント専用のクライアントプロファイルと、パブリッククライアント専用のプロファイルが含まれています。

詳細は、OAuth 2.1 のサポート を参照してください。

1.4.6.3. 更新トークンフローでサポートされる scope パラメーター

このリリースから、トークン更新用の OAuth2/OIDC エンドポイントの **scope** パラメーターがサポートされるようになりました。このパラメーターは、最初に付与されたスコープよりもスコープ数が少ないアクセストークンを要求するために使用します。つまり、アクセストークンのスコープ数を増やすことはできません。このスコープの制限は、更新された更新トークンのスコープには影響しません。この機能は、OAuth2 仕様で説明されているとおりに動作します。

詳細は、サーバー管理ガイド を参照してください。

1.4.6.4. セキュアなリダイレクト URI のクライアントポリシーエグゼキューター

新しいクライアントポリシーエグゼキューター **secure-redirect-uris-enforcer** が導入されました。これは、クライアントが使用できるリダイレクト URI を制限するために使用します。たとえば、クライアントのリダイレクト URI でワイルドカードを使用できないこと、URI が特定のドメインからのものであること、URI が OAuth 2.1 に準拠している必要があることなどを指定できます。

詳細は、クライアントポリシー を参照してください。

1.4.6.5. DPoP を適用するクライアントポリシーエグゼキューター

新しいクライアントポリシーエグゼキューター **dpop-bind-enforcer** が導入されました。プレビュー機能の **dpop** が有効になっている場合、このエグゼキューターを使用して、特定のクライアントに対して DPoP を適用できます。

詳細は、クライアントポリシー を参照してください。

1.4.6.6. EdDSA のサポート

EdDSA レルムキーを作成し、さまざまなクライアントの署名アルゴリズムとして使用できます。たとえば、このキーを使用してトークンに署名したり、署名済み JWT によるクライアント認証を行ったりすることができます。この機能には、サードパーティーのアイデンティティープロバイダーへの **private_key_jwt** 認証に使用されるクライアントアサーションに Red Hat build of Keycloak 自体が署名するアイデンティティーブローカリングが含まれています。

詳細は、レルムキーの設定 を参照してください。

1.4.6.7. JavaKeystore プロバイダーによってサポートされる EC キー

レルムキーを提供するためのプロバイダー **JavaKeystoreProvider** が、以前サポートされていた RSA キーに加えて、EC キーもサポートするようになりました。

詳細は、レルムキーの設定 を参照してください。

1.4.6.8. アイデンティティープロバイダーに private_key_jwt 認証を使用するときに、JWT に X509 サムプリントを追加するオプション

秘密鍵で署名された JWT を使用したクライアント認証が使用される状況に対応するために、OIDC アイデンティティープロバイダーに、**Add X.509 Headers to the JWT**オプションが追加されました。このオプションは、JWT にサムプリントが存在することを必須とする Azure AD など、一部のアイデンティティープロバイダーとの相互運用性を確保するのに役立ちます。

詳細は、アイデンティティープロバイダーの統合 を参照してください。

1.4.6.9. OAuth Grant Type SPI

Red Hat build of Keycloak のコードベースに、OAuth Grant Type SPI を導入するための内部更新が追加されました。この更新により、Red Hat build of Keycloak OAuth 2 トークンエンドポイントでサポートされているカスタムグラントタイプを導入する際の柔軟性が向上します。

詳細は、認可サービス を参照してください。

1.4.6.10. FAPI 2 ドラフトのサポート

Red Hat build of Keycloak に、新しいクライアントプロファイル **fapi-2-security-profile** と **fapi-2-message-signing** が追加されました。これにより、Red Hat build of Keycloak がクライアントと通信する際に最新の FAPI 2 ドラフト仕様への準拠が確実に適用されます。

詳細は、クライアントポリシー を参照してください。

1.4.6.11. DPoP プレビューのサポート

Red Hat build of Keycloak に、OAuth 2.0 Demonstrating Proof-of-Possession at the Application Layer (DPoP) のサポートのプレビューが追加されました。

1.4.6.12. OAuth 2.0 デバイス認可グラントフローの機能フラグ

OAuth 2.0 デバイス認可グラントフローに機能フラグが追加されたため、この機能を簡単に無効にすることができます。この機能はデフォルトで有効になっています。

詳細は、デバイス認可グラント を参照してください。

1.4.7. 認証

1.4.7.1. パスキーのサポート

Red Hat build of Keycloak は、パスキー のプレビューサポートを提供します。

パスキーの登録と認証は WebAuthn の機能によって実現されます。したがって、Red Hat build of Keycloak のユーザーは、既存の WebAuthn の登録と認証を使用して、パスキーの登録と認証を行うことができます。

同期されたパスキーとデバイスにバインドされたパスキーは、Same-Device Authentication と Cross-Device Authentication の両方に使用できます。ただし、パスキー操作の成功は、ユーザーの環境によって異なります。どの操作が 環境 で正常に実行されるかを確認してください。

1.4.7.2. WebAuthn の改善

WebAuthn ポリシーに、新しいフィールド **Extra Origins** が追加されました。これにより、非 Web プラットフォーム (ネイティブモバイルアプリケーションなど) との相互運用性が向上します。

1.4.7.3. You are already logged-in

このリリースでは、ユーザーが複数のブラウザータブでログインページを開いていて、1 つのブラウザータブでユーザーが認証するときに発生する問題に対処しています。ユーザーが別のブラウザータブで認証を試みると、**You are already logged-in** というメッセージが表示されていました。最初のタブで認証された後、他のブラウザータブでユーザーが自動的に認証されるようになったため、この問題は改善されました。しかし、さらなる改善が必要です。たとえば、あるブラウザータブで認証セッションが有効期限切れになった後に再開されても、他のブラウザータブではログインが自動的に行われません。

1.4.7.4. 最大認証時間を指定するためのパスワードポリシー

Red Hat build of Keycloak は、再認証なしでユーザーがパスワードを変更できる最大認証有効期間を指定できる新しいパスワードポリシーをサポートするようになりました。このパスワードポリシーが 0 に設定されている場合、ユーザーはアカウントコンソールまたはその他の手段でパスワードを変更するために再認証する必要があります。デフォルト値の 5 分よりも低い値または高い値を指定することもできます。

1.4.8. サーバー分散

1.4.8.1. 負荷制限のサポート

Red Hat build of Keycloak に、高負荷時に受信要求を適切に拒否できるようにする **http-max-queued-requests** オプションが追加されました。詳細は、サーバーガイド を参照してください。

1.4.8.2. RESTEasy Reactive

Red Hat build of Keycloak は RESTEasy Reactive に切り替わりました。**quarkus-resteasy-reactive** を使用するアプリケーションで、リアクティブスタイル/セマンティクスを使用していない場合でも、起動時間、実行時パフォーマンス、メモリーフットプリントの改善による利点が得られるはずです。JAX-RS API に直接依存する SPI は、通常、この変更に対して互換性があります。**ResteasyClientBuilder** など、RESTEasy Classic に依存する SPI は、互換性がないため、更新が必要になります。この更新は、Jersey のような JAX-RS API の他の実装にも必要になります。

1.4.9. Keycloak CR

1.4.9.1. Keycloak CR の Optimized フィールド

Keycloak CR に **startOptimized** フィールドが追加されました。これを使用すると、start コマンドに **--optimized** フラグを使用するかどうかに関するデフォルトの想定をオーバーライドできます。そのため、カスタム Keycloak イメージを使用する場合でも、CR を使用してビルド時のオプションを設定できます。

1.4.9.2. Keycloak CR の resources オプション

Keycloak CR で、Keycloak コンテナーのコンピュートリソースを管理するための **resources** オプションを指定できるようになりました。これにより、Keycloak CR を使用して Red Hat build of Keycloak に対して、および Realm Import CR を使用してレルムインポートジョブに対して、個別にリソースを要求および制限できます。

値が指定されていない場合、デフォルトの **requests** メモリーが **1700MiB** に設定され、**limits** メモリーが **2GiB** に設定されます。

次のように、要件に応じてカスタム値を指定できます。

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

詳細は、Operator ガイド を参照してください。

1.4.9.3. Keycloak CR の cache-config-file オプション

Keycloak CR で、**cache** 仕様の **configMapFile** フィールドを使用して **cache-config-file** オプションを指定できるようになりました。次に例を示します。

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

1.4.10. 機能のバージョン管理

機能のバージョン管理がサポートされるようになりました。下位互換性を維持するために、既存のすべ

ての機能 (**account2** および **account3** を含む) がバージョン 1 としてマークされています。新しく導入された機能ではバージョン管理が使用されるため、ユーザーは必要な機能のさまざまな実装を選択できます。

詳細は、サーバーガイド を参照してください。

1.4.10.1. Keycloak CR のトラストストア

Keycloak CR を使用して、トラストストアの新しいサーバー側処理を利用することもできます。次に例を示します。

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

現在はシークレットのみがサポートされています。

1.4.10.2. Kubernetes CA の信頼

Kubernetes CA の証明書は、Operator によって管理される Red Hat build of Keycloak Pod に自動的に追加されます。

1.4.11. グループのスケーラビリティー

多数のグループとサブグループを使用するユースケースで、グループの検索に関するパフォーマンスが向上します。ページ分割されたサブグループ検索を可能にする強化が行われました。

1.4.12. Keycloak JS

1.4.12.1. package.json の exports フィールドの使用

Red Hat build of Keycloak JS アダプターが、**package.json** 内の **exports** フィールド を使用するようになりました。この変更により、Webpack 5 や Vite などの最新のバンドラーのサポートが改善されますが、避けられない重大な変更がいくつか発生します。詳細は、アップグレードガイド を参照してください。

1.4.12.2. PKCE がデフォルトで有効

Red Hat build of Keycloak JS アダプターが、**pkceMethod** オプションをデフォルトで **S256** に設定するようになりました。この変更により、アダプターを使用するすべてのアプリケーションで Proof Key Code Exchange (PKCE) が有効になります。PKCE をサポートしていないシステムでアダプターを使用する場合は、**pkceMethod** オプションを **false** に設定すると、PKCE を無効にできます。

1.4.13. パスワードハッシュ化の変更

このリリースでは、パスワードハッシュ化のデフォルトを、パスワード保存に関する OWASP の推奨事項 に合わせて調整しました。

この変更の一環として、デフォルトのパスワードハッシュ化プロバイダーが **pbkdf2-sha256** から **pbkdf2-sha512** に変更されました。また、**pbkdf2** ベースのパスワードハッシュ化アルゴリズムのデフォルトハッシュ反復回数が変更されました。この変更により、最新の推奨事項に沿ったセキュリティー向上が実現しますが、パフォーマンスへの影響が生じます。パスワードポリシー **hashAlgorithm** と **hashIterations** をレルムに追加することで、古い動作を維持できます。詳細は、アップグレードガイド を参照してください。

1.4.14. トラストストアの改善

Red Hat build of Keycloak に、改善されたトラストストアの設定オプションが導入されました。Red Hat build of Keycloak のトラストストアが、送信接続、mTLS、データベースドライバーを含むサーバー全体で使用されるようになりました。個々の領域ごとに個別のトラストストアを設定する必要がなくなりました。トラストストアを設定するには、トラストストアファイルまたは証明書をデフォルトの **conf/truststores** に配置するか、新しい **truststore-paths** 設定オプションを使用します。

詳細は、サーバーガイド を参照してください。

1.4.15. その他の変更点

1.4.15.1. SAML アイデンティティープロバイダーの自動証明書管理

IDP エンティティーメタデータ記述子エンドポイントから署名証明書を自動的にダウンロードするように SAML アイデンティティープロバイダーを設定できるようになりました。新しい機能を使用するには、プロバイダーの **Metadata descriptor URL** オプション (証明書を含む IDP メタデータ情報が公開される URL) を設定し、**Use metadata descriptor URL** を **ON** に設定します。証明書はその URL から自動的にダウンロードされ、**public-key-storage** SPI にキャッシュされます。管理コンソールから、プロバイダーページのアクションコンボを使用して、証明書を再読み込みまたはインポートすることもできます。

新しいオプションの詳細は、サーバー管理ガイド を参照してください。

1.4.15.2. ロードバランサーの非ブロッキングヘルスチェック

/lb-check で利用可能な新しいヘルスチェックエンドポイントが追加されました。イベントループ内で実行が動作するため、このチェックは、リクエストキューで待機している多数の要求を Red Hat build of Keycloak が処理する必要がある過負荷の状況でも応答します。この動作は、たとえば、マルチサイトデプロイメントで、負荷の高い別のサイトへのフェイルオーバーを回避する場合に役立ちます。このエンドポイントは、現在、組み込みおよび外部 Infinispan キャッシュの可用性をチェックします。今後、他のチェックが追加される可能性があります。

このエンドポイントはデフォルトでは利用できません。有効にするには、**multi-site** 機能を使用して Keyloak を実行します。詳細は、機能の有効化と無効化 を参照してください。

1.4.15.3. Admin API コンテキストと Account コンテキストの両方におけるユーザー表現の変更

このリリースでは、Admin Account と Account REST API を使用するときにルートユーザー属性 (**username**、**email**、**firstName**、**lastName**、**locale** など) をマーシャリングおよびアンマーシャリングする方法を一致させるために、これらの属性を基底/抽象クラスに移動してカプセル化しています。

この方法により、クライアントによる属性の管理方法の一貫性が確保され、レルムに設定されたユーザープロファイル設定に属性が確実に準拠するようになります。

詳細は、アップグレードガイド を参照してください。

1.4.15.4. 管理ユーザー API を通じてユーザーを更新する際のユーザー属性の部分的な更新はサポートされなくなりました。

管理ユーザー API を介してユーザー属性を更新する場合、**username**、**email**、**firstName**、**lastName** などのルート属性を含むユーザー属性を更新するときに、部分的な更新はできません。

詳細は、アップグレードガイド を参照してください。

1.4.15.5. オフラインセッションとリモートセッションの順次ロード

このリリースから、Red Hat build of Keycloak クラスターの最初のメンバーが、リモートセッションを並行してではなく順番にロードするようになります。オフラインセッションのプリロードが有効になっている場合は、それらも順番にロードされます。

詳細は、アップグレードガイド を参照してください。

1.4.15.6. 認証済みの別のユーザーに代わってアクションを実行することが不可能

このリリースでは、ユーザーがすでに認証されていて、アクションが別のユーザーにバインドされている場合、メール検証などのアクションを実行できなくなりました。たとえば、メールリンクが別のアカウントにバインドされている場合、ユーザーは検証メールフローを完了できません。

1.4.15.7. メールアドレス検証フローの変更

このリリースでは、ユーザーがリンクをクリックしてメールアドレスを検証しようとしたときに、メールアドレスが以前に検証済みである場合、適切なメッセージが表示されます。

さらに、すでに検証済みのメールアドレスを検証しようとしていることを示すために、新しいエラー (**EMAIL_ALREADY_VERIFIED**) イベントが発生するようになりました。このイベントを使用すると、リンクが漏洩した場合にユーザーアカウントを乗っ取る可能性のある攻撃を追跡したり、ユーザーがアクションを認識していない場合に警告したりすることができます。

1.4.15.8. テーマのローカリゼーションファイルのデフォルト設定が UTF-8 エンコード

テーマのメッセージプロパティーファイルが UTF-8 エンコードで読み取られるようになり、自動的に ISO-8859-1 エンコードにフォールバックされます。

詳細は、アップグレードガイド を参照してください。

1.4.15.9. オフラインセッションの有効期間をメモリー内でオーバーライドするための設定オプション

メモリー要件を削減するために、Infinispan キャッシュにインポートされたオフラインセッションの有効期間を短縮する設定オプションが導入されました。現在、オフラインセッションの有効期間のオーバーライドはデフォルトで無効になっています。

詳細は、サーバー管理ガイド を参照してください。

1.4.15.10. Infinispan メトリクスがキャッシュマネージャーとキャッシュ名のラベルを使用

Red Hat build of Keycloak の組み込みキャッシュのメトリクスを有効にしたときに、メトリクスがキャッシュマネージャーとキャッシュ名のラベルを使用するようになりました。

詳細は、アップグレードガイド を参照してください。

1.4.15.11. ユーザー属性値の長さの拡張

このリリース以降、Red Hat build of Keycloak は、以前は制限されていた 255 文字を超えるユーザー属性値の保存と検索をサポートします。

詳細は、アップグレードガイド を参照してください。

1.4.15.12. ブルートフォース保護の変更

ブルートフォース保護にいくつかの機能拡張が加えられました。

1. ブルートフォース保護により OTP またはリカバリーコードによる認証の試行が失敗すると、アクティブな認証セッションが無効になります。そのセッションでさらに認証を試みても失敗します。

2. 以前のバージョンの Red Hat build of Keycloak では、アカウントに対するブルートフォース攻撃を受けた際に、ユーザーを一時的に無効にするか、永久に無効にするかを管理者が選択する必要がありました。管理者は、一定回数の一時的なロックアウト後にユーザーを永久に無効にできるようになりました。

3. プロパティー **failedLoginNotBefore** が **brute-force/users/{userId}** エンドポイントに追加されました。

1.4.15.13. 認可ポリシー

以前のバージョンの Red Hat build of Keycloak では、ユーザー、グループ、またはクライアントポリシーの最後のメンバーが削除されると、そのポリシーも削除されていました。そのため、このポリシーが集約ポリシーで使用された場合、権限の昇格が発生することがありました。権限の昇格を回避するために、有効なポリシーが削除されなくなりました。また、管理者がこのようなポリシーを更新する必要があります。

1.4.15.14. 一時的なロックアウトログのイベントへの置き換え

ブルートフォースプロテクターによってユーザーが一時的にロックアウトされた場合に、新しいイベント **USER_DISABLED_BY_TEMPORARY_LOCKOUT** が発生するようになりました。新しいイベントは情報を構造化された形式で提供するため、ID **KC-SERVICES0053** のログは削除されました。

詳細は、アップグレードガイド を参照してください。

1.4.15.15. Cookie の更新

新しい Cookie プロバイダーを含め、Cookie 処理コードがリファクタリングされ、改善されました。これにより、Red Hat build of Keycloak によって処理される Cookie の一貫性が向上し、必要に応じて Cookie に関する設定オプションを導入できるようになります。

1.4.15.16. SAML User Attribute Mapper For NameID で有効な NameID 形式のみを提案

以前は、User Attribute Mapper For NameID で、**Name ID Format** オプションを次の値に設定できました。

- **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**

しかし、Red Hat build of Keycloak は、これらの **NameIDPolicy** のいずれかを使用した **AuthnRequest** ドキュメントの受信をサポートしていないため、これらのマッパーは使用されません。サポートされるオプションが更新され、次の Name ID Formats だけが含まれるようになりました。

- **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**

1.4.15.17. コンテナー内で実行する場合の異なる JVM メモリー設定

Red Hat build of Keycloak では、初期ヒープサイズと最大ヒープサイズにハードコード値を指定せずに、コンテナーの合計メモリーに対する相対値を使用します。JVM オプション **-Xms** および **-Xmx** が、**-XX:InitialRAMPercentage** と **-XX:MaxRAMPercentage** に置き換えられました。

警告

メモリー消費に大きな影響を与える可能性があるため、特定のアクションが必要になる場合があります。

詳細は、アップグレードガイド を参照してください。

1.4.15.18. オフラインセッションのプリロードの非推奨化

Red Hat build of Keycloak のデフォルトの動作では、オンデマンドでオフラインセッションをロードします。起動時にオフラインセッションをプリロードするという従来の動作は、非推奨になりました。起動時にプリロードすると、セッション数の増加に応じて適切にスケールすることができず、Red Hat build of Keycloak のメモリー使用量が増加するためです。古い動作は今後のリリースで削除される予定です。

詳細は、アップグレードガイド を参照してください。

1.5. 修正された問題

各リリースには修正された問題が含まれています。

- Red Hat build of Keycloak 24.0.5 で修正された問題 。
- Red Hat build of Keycloak 24.0.4 で修正された問題 。
- Red Hat build of Keycloak 24.0.3 で修正された問題 。

1.6. 既知の問題

Red Hat build of Keycloak 24.0 では、Red Hat Single Sign-On 7.6 OIDC アダプターがデフォルトで動作しません。

Red Hat build of Keycloak 24.0 で Red Hat Single Sign-On 7.6 OIDC アダプターを実行すると、ログに CODE_TO_TOKEN_ERROR イベントが表示されます。この問題を回避するには、Red Hat Single Sign-On 7.6 アダプターによって保護されたアプリケーションを参照する各 Red Hat build of Keycloak クライアントに、以下の変更を加えます。

1. 管理コンソールで、該当するクライアントを選択します。
2. **Advanced** タブに移動します。
3. **OpenID Connect Compatibility Modes** セクションを見つけます。
4. **Exclude Issuer From Authentication Response** を **ON** に切り替えます。

詳細は、https://issues.redhat.com/browse/RHSSO-3030 を参照してください。

1.7. サポートされる構成

Red Hat build of Keycloak 24.0 でサポートされる構成については、 サポートされる構成 を参照してください。

1.8. コンポーネントの詳細

Red Hat build of Keycloak 24.0 でサポートされるコンポーネントバージョンのリストについては、コンポーネントの詳細 を参照してください。