OpenShift Sandboxed Containers 1.4

OpenShift Sandboxed Containers リリースノート

Red Hat OpenShift 向け

Last Updated: 2024-02-01

OpenShift Sandboxed Containers 1.4 OpenShift Sandboxed Containers リリースノート

Red Hat OpenShift 向け

法律上の通知

概要

このリリースノートには、すべての新機能と拡張機能、注目すべき技術的変更、以前のバージョンからの主要な修正、および一般提供時の既知のバグがまとめられています。

目次

はじめに

第1章 OPENSHIFT SANDBOXED CONTAINERS 1.4 リリースノート

1.1. このリリースについて

これらのリリースノートでは、Red Hat Red Hat OpenShift 4.13 と並行して OpenShift Sandboxed Containers 1.4 の開発を追跡します。

この製品は、Red Hat OpenShift 4.10 の時点で完全にサポートされ、デフォルトで有効になっています。

1.2. 新機能および拡張機能

1.2.1. OpenShift Sandboxed Containers のピア Pod のサポート (テクノロジープレビュー)

AWS または Microsoft Azure のピア Pod を使用して、OpenShift Sandboxed Containers のワークロードをデプロイできるようになりました。これにより、ネストされた仮想化の必要性がなくなります。この機能はテクノロジープレビューであるため、完全にはサポートされていません。詳細は、ピア Pod を使用した OpenShift Sandboxed Containers ワークロードのデプロイ を参照してください。

1.2.2. QEMU エラーログ収集

QEMU の警告ログとエラーログは、ノードジャーナル、Kata ランタイムログ、および CRI-O ログの両方に出力されるようになりました。詳細は、OpenShift Sandboxed Containers のデバッグログの表示 を参照してください。

1.2.3. OpenShift Sandboxed Containers Operator をインストールするための更新チャネル

OpenShift Sandboxed Containers Operator をインストールするときのサブスクリプションチャネルは、一貫性を確保するために、**stable-<version>** ではなく、常に **stable** になりました。

1.3. バグ修正

- 以前は、OpenShift Sandboxed Containers をアップグレードしても、既存の **KataConfig** CR は自動的に更新されませんでした。その結果、以前のデプロイメントのモニター Pod は再起動されず、古い **kataMonitor** イメージで実行され続けました。
  リリース 1.3.2 以降、**kataMonitorImage はKataConfig** CR から削除され、すべてのモニター Pod のアップグレードは Operator によって内部処理されます。

  (KATA-1650)

- 以前のリリースでは、ネットワーク接続のないクラスターに OpenShift Sandboxed Containers をインストールできませんでした。kata-monitor コンテナーイメージのプル仕様では、ダイジェストの代わりにタグが使用されていました。これにより、イメージが **ImageContentSourcePolicy** リソースでミラーリングされなくなっていました。
  今回のリリースでは、OpenShift Sandboxed Containers Operator 内のすべてのコンテナーイメージが確実に含まれるように、CSV **spec.relativeImages** セクションが更新されました。その結果、すべてのコンテナーのプル仕様ではタグの代わりにダイジェストが利用されるようになり、切断された環境でも OpenShift Sandboxed Containers をインストールできるようになりました。

(KATA-2038)

- 以前は、汚染されたノード上で実行している OpenShift サンドボックスコンテナーではメトリックを利用できませんでした。このリリースでは、**kata-monitor** Pod に容認機能が追加され、汚染されたノードを含む任意のノードで Pod を実行してメトリックを収集できるようになりました。(KATA-2121)

- 以前は、OpenShift サンドボックスコンテナーの Operator のベースイメージは **ubi8/ubi-mimimal** イメージを使用していました。このリリースでは、RHEL 9 クラスターと Red Hat OpenShift 4.13 との互換性を確保するために、ベースイメージが ubi9 /**ubi** イメージを使用するように更新されました。(KATA-2212)

1.4. 既知の問題

- OpenShift Sandboxed Containers を使用している場合は、Red Hat OpenShift クラスター内の **hostPath** ボリュームからマウントされたファイルまたはディレクトリーにアクセスすると、SELinux が拒否することがありました。特権 Sandboxed Container は SELinux チェックを無効にしないため、特権 Sandboxed Container を実行している場合でも、このように拒否される可能性があります。
ホストで SELinux ポリシーに従うことで、デフォルトでサンドボックス化されたワークロードからホストファイルシステムを完全に分離することが保証されます。これにより、**virtiofsd** デーモンまたは QEMU の潜在的なセキュリティー上の欠陥に対する保護も強化されます。

  マウントされたファイルまたはディレクトリーにホスト上の特定の SELinux 要件がない場合は、代わりにローカル永続ボリュームを使用できます。ファイルは、コンテナーランタイムの SELinux ポリシーに従って、自動的に **container_file_t** に再ラベル付けされます。ローカルボリュームを使用した永続ストレージ を参照してください。

  マウントされたファイルまたはディレクトリーがホスト上で特定の SELinux ラベルを持つことが予想される場合、自動再ラベル付けはオプションではありません。代わりに、ホストでカスタム SELinux ルールを設定して、**virtiofsd** デーモンがこれらの特定のラベルにアクセスできるようにすることができます。(KATA-469)

- 一部の OpenShift Sandboxed Containers Operator Pod は、コンテナーの CPU リソース制限を使用して、Pod で使用可能な CPU の数を増やします。これらの Pod は、要求されたよりも少ない CPU を受け取る可能性があります。コンテナー内で機能が利用可能な場合は、**oc rsh <pod>** を使用して Pod にアクセスし、**lscpu** コマンドを実行することで、CPU リソースの問題を診断できます。

  ```
  $ lscpu
  ```

  出力例

  ```
  CPU(s):                16
  On-line CPU(s) list:   0-12,14,15
  Off-line CPU(s) list:  13
  ```

  オフライン CPU のリストは、実行ごとに予期せず変更される可能性があります。

  回避策として、CPU 制限を設定するのではなく、Pod アノテーションを使用して追加の CPU をリクエストできます。Pod アノテーションを使用する CPU リクエストは、プロセッサーの割り当て方法が異なるため、この問題の影響を受けません。CPU 制限を設定するのではなく、Pod のメタデータに次の **annotation** を追加する必要があります。

```
metadata:
  annotations:
    io.katacontainers.config.hypervisor.default_vcpus: "16"
```

(KATA-1376)

- ランタイムインストールの進行状況は、**kataConfig** カスタムリソース (CR) の **status** セクションに表示されます。ただし、次の条件がすべて当てはまる場合、進行状況は表示されません。
  - ワーカーノードが定義されていません。**oc get machineconfigpool** を実行して、マシン設定プール内のワーカーノードの数を確認できます。
  - インストールするノードを選択するための **kataConfigPoolSelector** が指定されていません。

  この場合、Operator はノードがコントロールプレーンとワーカーの両方のロールを持つコンバージドクラスターであると想定するため、コントロールプレーンノードでインストールが開始されます。**kataConfig** CR の **status** セクションは、インストール中に更新されません。(KATA-1017)

- Web コンソールの **KataConfig** タブで、**YAML view** で **Create KataConfig** をクリックすると、**KataConfig** YAML に **spec** フィールドがありません。**Form view** に切り替えてから **YAML view** に戻ると、この問題が修正され、完全な YAML が表示されます。(KATA-1372)

- Web コンソールの **KataConfig** タブに、**KataConfig** CR がすでに存在するかどうかにかかわらず、**404: Not found** エラーメッセージが表示されます。既存の **KataConfig** CR にアクセスするには、**Home** > **Search** に移動します。**Resources** リストから、**KataConfig** を選択します。(KATA-1605)

- **KataConfig** CR のインストール中に、最初のノードが再起動する前に **KataConfig** CR の削除が開始されると、ノードのステータスが正しくなくなります。これが発生すると、Operator は **KataConfig** CR の削除とインストールを同時に試行する状態でスタックします。想定される動作として、インストールが停止し、**KataConfig** CR が削除されます。(KATA-1851)

- コンテナーのセキュリティーコンテキストで SELinux Multi-Category Security (MCS) ラベルを設定すると、Pod が起動せず、次のエラーが出力されます。

  ```
  Error: CreateContainer failed: EACCES: Permission denied: unknown
  ```

  ランタイムは、Sandboxed Containers の作成時にコンテナーのセキュリティーコンテキストにアクセスできません。これは、**virtiofsd** が適切な SELinux ラベルで実行されず、コンテナーのホストファイルにアクセスできないことを意味します。その結果、MCS ラベルを利用して Sandboxed Containers 内のファイルをコンテナーごとに分離できません。つまり、すべてのコンテナーがSandboxed Containers 内のすべてのファイルにアクセスできることになります。現在、この問題に対する回避策はありません。

  (KATA-1875)

- Sandboxed Containers ワークロードを停止すると、次の QEMU エラーメッセージがワーカーノードシステムジャーナルに記録されます。

  ```
  qemu-kvm: Failed to write msg.
  qemu-kvm: Failed to set msg fds.
  qemu-kvm: vhost VQ 0 ring restore failed
  qqemu-kvm: vhost_set_vring_call failed
  ```

これらのエラーは無害なので無視してかまいません。

システムジャーナルログにアクセスする方法の詳細は、Red Hat サポートの OpenShift Sandboxed Containers データの収集 を参照してください。

(KATA-2133)

- Web コンソールを使用して OpenShift Sandboxed Containers Operator をインストールした場合は、**Install** をクリックした後に UI に間違った Operator バージョンが表示されることがあります。
バージョンが正しくない場合は、インストールウィンドウに灰色のテキストで次のように表示されます。

  <Version number> provided by Red Hat

  正しい Operator をインストールします。**Operators → Installed Operators** に移動すると、OpenShift Sandboxed Containers Operator の下に正しいバージョンがリストされていることがわかります。

  (KATA-2161)

- OpenShift Sandboxed Containers でピア Pod を使用する場合は、**KataConfig** CR を作成し、**enablePeerPods** フィールドを **true** に設定すると、**kata-remote-cc** ランタイムクラスが作成されます。結果として、**KataConfig** CR に **kata** ランタイムクラスに加えて、**kata-remote-cc** ランタイムクラスが表示され、技術的には、標準の Kata Pod とピア Pod Kata Pod の両方を同じクラスター上で実行できるはずです。
クラスター管理者として **KataConfig** CR を調べると、**Status.runtimeClass** フィールドに **kata** のみが表示されます。ランタイムクラス **kata-remote-cc** が表示されません。現在、この問題に対する回避策はありません。

  (KATA-2164)

- OpenShift Sandboxed Containers の FIPS コンプライアンスは、**kata** ランタイムクラスにのみ適用されます。新しいピア Pod ランタイムクラス **kata-remote-cc** はまだ完全にはサポートされておらず、FIPS コンプライアンスについてはテストされていません。(KATA-2166)

1.5. 制限

- OpenShift Sandboxed Containers で古いバージョンの Buildah ツールを使用すると、ビルドが次のエラーで失敗します。

  ```
  process exited with error: fork/exec /bin/sh: no such file or directory

  subprocess exited with status 1
  ```

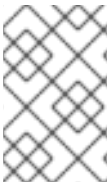
  quay.io で入手可能な Buildah の最新バージョンを使用する必要があります。

  (KATA-1278)

1.6. エラータの非同期更新

OpenShift サンドボックスコンテナー 4.13 のセキュリティー、バグ修正、および拡張機能の更新は、Red Hat Network を通じて非同期エラータとして発表されます。すべての Red Hat OpenShift 4.13 エラータは Red Hat カスタマーポータル で提供されています。非同期エラータの詳細は、Red Hat OpenShift ライフサイクル を参照してください。

Red Hat カスタマーポータルのユーザーは、Red Hat Subscription Management (RHSM) のアカウント設定でエラータの通知を有効にできます。エラータの通知を有効にすると、登録しているシステムに関連するエラータが新たに発表されるたびに、メールで通知が送信されます。



注記

Red Hat Customer Portal のユーザーアカウントには、システムが登録されていて、Red Hat OpenShift エラータ通知メールを生成するための Red Hat OpenShift エンタイトルメントを使用している必要があります。

以下のセクションは、これからも継続して更新され、今後の OpenShift sandboxed containers 1.4 の非同期リリースで発表されるエラータの拡張機能およびバグ修正に関する情報を追加していきます。

1.6.1. RHBA-2023:3529 - OpenShift Sandboxed Containers 1.4.0 イメージのリリース、バグ修正、および機能強化に関するアドバイザリー

発行日: 2023-06-08

OpenShift Sandboxed Containers リリース 1.4.0 が利用可能になりました。このアドバイザリーには、機能強化とバグ修正を含む OpenShift Sandboxed Containers の更新が含まれています。

更新に含まれるバグ修正のリストは、RHBA-2023:3529 アドバイザリーに記載されています。

1.6.2. RHSA-2023:4290 - OpenShift Sandboxed Containers 1.4.1 イメージのリリース、バグ修正、およびセキュリティーアドバイザリー

発行日: 2023-07-27

OpenShift サンドボックスコンテナーリリース 1.4.1 が利用可能になりました。このアドバイザリーには、セキュリティーおよびバグ修正を含む OpenShift Sandboxed Containers の更新が含まれています。

更新に含まれるバグ修正のリストは、RHSA-2023:4290 アドバイザリーに記載されています。